Algorithmic Embeddings

by

Mihai Bădoiu

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2006

Author . . . .
Department of Electrical Engineering and Computer Science
May 25, 2006

Certified by . . . .
Piotr Indyk
Associate Professor
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Algorithmic Embeddings

by

Mihai Bădoiu

Submitted to the Department of Electrical Engineering and Computer Science
on May 25, 2006, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

Abstract

We present several computationally efficient algorithms, and complexity results on low distortion mappings between metric spaces. An embedding between two metric spaces is a mapping between the two metric spcaes and the distortion of the embedding is the factor by which the distances change. We have pioneered theoretical work on *relative* (or *approximation*) version of this problem. In this setting, the question is the following: for the class of metrics C, and a host metric M', what is the *smallest approximation factor* $a \geq 1$ of an efficient algorithm minimizing the distortion of an 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. 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.

We investingate several variants of the above problem, varying different host and target metrics, and definitions of distortion. We present results for different types of distortion: *multiplicative* versus *additive*, *worst-case* versus *average-case* and several types of target metrics, such as the line, the plane, d-dimensional Euclidean space, ultrametrics, and trees. We also present algorithms for ordinal embeddings and embedding with extra information.

Thesis Supervisor: Piotr Indyk
Title: Associate Professor

Acknowledgments

I would like to thank everybody that had a long-lasting influence on me, and everybody who helped me with my thesis. First and foremost I would like to thank my parents for having me, for raising me, for teaching me about life, and for giving me inspiration and freedom to work towards what I like. I could not have asked for a better childhood.

There are many people worth noting who helped in my computer science education: my teacher Ilie from primary school, my math teacher Trandafir from secondary school, my teachers Severius Moldovean (math), Mihai Boicu (computer science), Mihai Budiu (computer science), Mihai Bătraneanu (computer science) from high school, and numerous professors from which I had so much to learn: Ken Clarkson, Erik Demaine, Piotr Indyk, David Karger, Daniel Spielman, Santosh Vempala. Not only professors, but also many discussions with my peers helped shape my interest in computer science: during high-school Mihnea Galca, Ciprian Serbu, and all my other peers from the Informatics High School and the National Children's Palace; during college and my doctorate studies: Daniel Preda, Anastasios Sidiropoulos, and Adrian Soviani.

I'd like to thank all my friends for moral support during my graduate studies. I'd like to thank Adrian Soviani, Constantin Chiscanu, Bogdan Feledes and Viorel Costeanu for being my close friends, and for being able of discussing about virtually anything. I'd like to thank Radu Berinde and Viorel Costeanu for being my gym workout partners. I'd like to thank the Euro Club basketball team, the Theory volleyball team, and my friends from the Boston Foosball League for all the glorious moments. I actually took some incredibly interesting classes during my PhD and I'd like to thank Stephen M. Meyer for sharing his views on american modern foreign policy, Lee D. Perlman for his sharing his philosophical ideas, the ROTC intructors for giving me a view of the US military leadership, Howard Anderson and Ken Zolot for a course on entrepreneurship. (Who would have thought the most interesting non-technical classes I'd take during my PhD?) Finally, I'd like to thank all my friends from Ashdown, the Romanian Student Association, EuroClub, and the Computer Science and Artificial Intelligence Laboratory.

My co-authors helped a lot in the development of this thesis: Noga Alon, Julia Chuzhoy, Kenneth L. Clarkson, Artur Czumaj, Erik D. Demaine, Kedar Dhamdhere, Martin Farach-Colton, Anupam Gupta, Mohammad Taghi Hajiaghayi, Sariel Har-Peled, Piotr Indyk, Yuri Rabinovich, Harald Rdcke, R. Ravi, Anastasios Sidiropoulos, and Christian Sohler.

Contents

List of Figures

Chapter 1

Introduction

The problem of computing mappings between metric spaces started to receive more attention from the theoretical computer science community in the past twenty years. This problem has been found in many applications, and as theoretical tools used for approximation algorithms. Mathematical studies of such mappings helped establish the basis of this area in theoretical computer science. Classical results such as those of Bourgain and Johnson-Lindenstrauss have already found numerous applications. The main problem with such worst-case results, is that they cannot be applied to low-dimensional spaces, mainly because there are high lower-bounds. The only way around this shortcoming is to consider approximating the best distortion embedding.

These mappings between metric spaces have also been studied in the field called *multi-dimensional scaling* (MDS) and have their roots in work going back to the first half of the 20th century, and modern roots in work of Shepard [She62a, She62b], Kruskal [Kru64a, Kru64b], and others. This is a subject of extensive research [MDS]. However, despite significant practical interests, very few theoretical results exist in this area. The most commonly used algorithms are heuristic (e.g., gradient-based method, simulated annealing, etc.) and are often not satisfactory in terms of the running time and/or quality of the embeddings. The only theoretical results in this area [HIL98, Iva00, ABFC$^+$96, FCK96] have been a constant factor approximation algorithm for minimum distortion embedding into a *line*, into *ultrametrics*, and into *trees*, all using an *additive* notion of embeddings.

1.1 Preliminaries

A metric space is a pair (X, f), where X is the set of points and $f : X \times X \to \Re^+$. In this dissertation for the input metric, we only consider the case when the metric is finite, so X is a finite set. A metric has the following axioms:

- $\forall x, y \in X,\ f(x, y) = 0$ if and only if $x = y$.
- $\forall x, y \in X,\ f(x, y) = f(y, x)$
- $\forall x, y, z \in X,\ f(x, y) + f(y, z) \geq f(x, z)$.

The last axiom is called the *triangle inequality*.

An metric can be specified by the distance function f, which for an n-point set can be represented in an $n \times n$ table. The metric specified this way can have any geometric structure. This is why it's hard to use a metric if given this way. A better way to represent a metric would be to give each point coordinates in $\Re^d$, and define f as the distance between points (according to some norm). Yet another way would be to have a tree or a graph, where the vertices correspond to the points, and the distance between two points is defined as the shortest path distance on the tree/graph. This metric is called a *shortest path metric* on a graph.

We would like to extract the geometric structure of a given metric and represent the metric as a point set in $\Re^d$, a tree or a graph. Representing the metric this way would have its benefits, such as being easier to use for other algorithms, easier to visualize the data, and easier to store the data. Unfortunately it is not always possible to map any metric space into these representations without changing the distances. Thus, we want to change the distances as little as possible. The quantity that measures by how much the distances have changed is called distortion. There are several ways of defining the distortion of a mapping.

Definition 1 *Given two metric spaces (X, f) and (X', f'), a mapping $g : X \to X'$ is called* an embedding.

Definition 2 *An embedding* $g : X \to X'$ *is called an* isometric embedding*, if for any* x, y*,* $f(x,y) = f'(g(x), g(y))$*.*

Definition 3 *An embedding* $g : X \to X'$ *is called* non-contracting*, if for any* x, y*,* $f(x,y) \leq f'(g(x), g(y))$*.*

Definition 4 *An embedding* $g : X \to X'$ *is called* non-expanding*, if for any* x, y*,* $f(x,y) \geq f'(g(x), g(y))$*.*

1.2 Types of embeddings

We will consider two ways of computing the overall distortion of an embedding. One way is to consider the maximum distortion of any pair of points. This is the more established (standard) way of computing the distortion. Another way is to consider the average distortion over all the pairs. This will be called an *average distortion embedding*. We will consider two ways of measuring the distortion between two points. One way is to consider the ratio of the new distance over the old distance. This is the standard way of looking at distortion. To be clear, we will call this *multiplicative distortion embeddings*. Another way is to consider the absolute difference between the new ratio and the old ratio. This will be called an *additive distortion embedding*. Note that the distortion of additive embeddings change under scaling, i.e., if one forces the embedding to be non-contractive or non-expanding, one will get different results.

More formally, for the (classic) multiplicative worst-case embedding, the distortion is computed as follows:

$$\alpha = \frac{\max_{x,y \in X} f'(g(x), g(y))/f(x,y)}{\min_{x,y \in X} f'(g(x), g(y))/f(x,y)}$$

For an embedding problem we are interested in computing a low-distortion embedding, i.e., we are interested in minimizing the distortion of the embedding.

In this dissertation we will address the *relative* or *approximation* version of this problem. In this setting, the question is the following: for a class of metrics C, and

Paper	From	Into	Distortion	Comments
[FCK96]	general distance matrix	ultrametrics	c	
[ABFC$^+$96]	general distance matrix	tree metrics	$3c$	
			$\geq 9/8c$	Hard to 9/8-approximate
[HIL98]	general distance matrix	line	$2c$	
			$\geq 4/3c$	Hard to 4/3-approximate
[B $\breve{0}$ 3]	**general distance matrix**	**plane under l_1**	$O(c)$	
[BDHI04]	**general distance matrix**	**plane under l_2**	$O(c)$	**Time quasi-polynomial in Δ**
-	**general distance metrix**	**line**	$5c$	**Menger-type result**
				4-points criterion
-	**general distance matrix**	**line**	$O(c)$	**average additive distortion**

Figure 1-1: Work on relative embedding problems for maximum additive distortion. The rows in bold are presented in this thesis.

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.

1.3 Results

Our results will be partitioned into four categories: results about the additive distortion, multiplicative distortion, ordinal embeddings, and when extra information about the metric is available.

In general, minimizing an additive measure suffers from the "scale insensitivity" problem: local structures can be distorted in arbitrary way, while the global structure is highly over-constrained. Multiplicative distortion generally does not suffer from the scale insensitivity problem. Minimizing the multiplicative distortion seems to be a harder problem in general.

Table 1-1 summarizes the results known about the additive distortion. The results in bold will be presented in this dissertation, in the chapter on additive distortion.

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

Table 1-2 summarizes the results known about algorithmic embeddings in the case of multiplicative distortion. In this dissertation we will present several of these results, in the chapter on multiplicative distortion (the ones in bold).

We also present in this thesis results on ordinal embeddings and on embeddings with extra information.

Paper	From	Into	Distortion	Comments
[LLR95]	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	$O(c)$	
-	**general metrics**	**line**	$O(c)$	**for $c < 3/2$**

Figure 1-2: 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; The results in bold are presented in this thesis.

Chapter 2

Additive embeddings

In this chapter we present results using *additive distortion*. Using this notion, for an embedding g from (X, f) to (X', f'), the distortion is defined as

$$\alpha = max_{x,y \in X} |f'(g(x), g(y)) - f(x, y)|$$

The results might force the metric to be non-contracting or non-expanding, in which cases the notion of distortion varies, i.e., scaling changes the distortion of an additive embedding.

2.1 Embedding Into the Plane

Credits: The results in this section have appeared in SODA'03.

Embedding arbitrary distance matrices into the two dimensional plane is a fundamental problem occurring in many applications. In the context of data visualization, this approach allows the user to observe the structure of the data set and discover its interesting properties. In computational chemistry, this approach is used to recreate the geometric structure of the data from the distance information. Other application areas are discussed in [MDS].

In this section we present a polynomial-time algorithm that approximates a given distance matrix $D[\cdot,\cdot]$ by a matrix of distances induced by a set of points in a two-dimensional plane under l_1 norm. Specifically, consider $\epsilon = \min_f\{\max_{p,q} |D[p,q] - ||f(p) - f(q)||_1|\}$. Our algorithm computes f such that $\max_{p,q} |D[p,q] - ||f(p) - f(q)||_1| \le c\epsilon$. The constant[1] c guaranteed by our algorithm is equal to 30. However, it is likely that it can be made smaller by a more careful analysis.

To our knowledge, this is the first algorithm that finds an (approximately) optimal embedding of a given distance matrix into a fixed d-dimensional space, where $d > 1$ is low, under *any* standard definition of embedding (see Related Work, in chapter 1).

Overview of this section. In Section 2.1.2, we give an overview of the algorithm. In Section 2.1.3, we show how to solve the problem for a special case when we know the exact values of the x coordinates of the points and the value ϵ^* of the smallest error possible. In Section 2.1.5, we show how to reduce our problem to the special case.

2.1.1 Preliminaries

Assume we are given a set P of n points and an $n \times n$ symmetric, positive and all-zero on the diagonal distance matrix D, which also satisfies the triangle inequality. The goal is to find an embedding $f : P \to \Re^2$ of the points into the plane, which minimizes the difference between the distances given by D and the distances given by the embedding. The distances in the plane are computed using the l_1 norm (or l_∞, which is isomorphic to l_1 in two dimensions).

Let ϵ^* be the optimal additive distortion. We guess ϵ^* by doing a binary search and we can assume we know its value. Given ϵ, let $f : P \to \Re^2$ be an embedding such that $\forall p, q \in P$,

$$|D[p,q] - ||f(p) - f(q)||_\infty| \le \epsilon$$

Such an embedding exists for every $\epsilon \ge \epsilon^*$. To shorten the notation, we denote the x coordinate of $f(p)$ by p_1 and the y coordinate by p_2. Also, we write $||p-q||_\infty$ instead of $||f(p) - f(q)||_\infty$. [2]

[1]Different constant than the one in the intro.

[2]In the plane, l_1 and l_∞ are equivalent (by just rotating the point set by 90 degrees and scaling

2.1.2 Overview of the algorithm

The algorithm works in two parts. In the first part, we approximate the x-coordinates of the embedding within $O(\epsilon^*)$. In the second part, assuming we know the approximate values of the x coordinates, we find the y values approximately.

The solution for the first part is easy in the case of the l_2 norm: we guess the diameter p, q, guess their placement, rotate the plane such that p and q are horizontal. Then we know that all v belong to the intersection of $\text{Ball}(p, D[p,v]+\epsilon)-\text{Ball}(p, D[p,v]-\epsilon)$ with $\text{Ball}(q, D[q,v]+\epsilon)-\text{Ball}(q, D[q,v]-\epsilon)$. This intersection gives little freedom to the x-coordinate of v, and we can guess it within $c\epsilon$ for a constant c. Unfortunately, the l_1 norm requires more elaborate techniques along these lines.

To do the second part we first find certain combinatorial structure of the point-set and then solve the problem using linear programming. Here we use properties of the l_∞ norm in a crucial way. We do not yet know how to do the second part in the case of the l_2 norm, but we believe a very similar method should work. In particular, we can roughly prove every lemma for l_2 except (2.1.7).

2.1.3 A special case

In this section, we are going to solve a special case in which we know the exact x coordinates of the points and ϵ for which there exists an embedding with distortion of at most ϵ. More exactly, we will compute a 5-approximation solution, i.e., we compute f such that $\forall p, q \in P$,

$$|D[p,q] - ||f(p) - f(q)||_\infty| \leq 5\epsilon$$

In the following sections we are going to reduce the main problem to this special case.

Definition 2.1.1 We connect two points p, q with an edge if $D[p,q] > |p_1 - q_1| + 3\epsilon$. We call such an edge a "strong" edge. We connect two points p, q with a "weak" edge if there is no strong edge between p and q and $D[p,q] > |p_1 - q_1| + \epsilon$.

---

it by some factor).

Let E be the set of the strong edges,

$$E = \{(p, q) \mid D[p, q] > |p_1 - q_1| + 3\epsilon\}$$

Intuitively, the strong edges are the edges we care about. Our goal is eventually to reduce the problem to linear programming. If there is no strong edge between two points p and q, by adding the constraint $-D[p, q] - \epsilon \leq q_2 - p_2 \leq D[p, q] + \epsilon$ we can ensure that the distance between p and q in our solution is less than $D[p, q] + \epsilon$. Also, since there is no strong edge between p and q, the distance already given by $|p_1 - q_1|$ is good enough for a 3-approximation solution.

Let $G = (P, E)$. If p, q, w are vertices in the same connected component of G, we also add to E the weak edges between the points v and w if $p_1 \leq v_1 \leq q_1$ and if v is not part of the component:

$$E' = E \bigcup \{(v, w) \mid \exists p, q \text{ in the same connected component as } w \text{ of } G \text{ and } v \text{ is not in the same connected component as } w \text{ and } D[v, w] > |v_1 - w_1| + \epsilon \text{ and } p_1 \leq v_1 \leq q_1\}$$

Let $G' = (P, E')$.

Definition 2.1.2 For an edge $(p, q) \in E'$, $p_1 \leq q_1$, we have two cases: $p_2 - q_2 \geq 0$ or $p_2 - q_2 < 0$. [3] We say an edge is "oriented up" if it satisfies the first inequality and "down" if it satisfies the second inequality.

The main idea of the algorithm is the following: We partition the elements of P into connected components of G'. We first note that if we know the orientation of all the strong edges, we can compute an embedding with distortion of at most $3\epsilon^*$ via linear programming. We also note that within each connected component, if we fix the orientation of a single edge, we can determine the orientation of all the other edges. Finally, we observe that any relative orientation of the edges between the connected components suffices in computing an embedding with distortion of at most

[3]Note that the fact that $(p, q) \in E'$ and the first inequality implies $p_2 - q_2 > q_1 - p_1$. Also, $(p, q) \in E'$ and the second inequality implies $p_2 - q_2 > q_1 - p_1$.

$5\epsilon^*$.

Claim 2.1.3 *If p and q are in the same connected component of G, and $p_1 \leq q_1$, then every k for which $p_1 \leq k_1 \leq q_1$ is part of the same connected component of G' as p and q. Moreover, every weak edge in G' is adjacent to at least one strong edge.*

Proof: Since p and q are in the same connected component, there exists a path connecting p and q in G. Therefore, there exist v and w such that $(v, w) \in E$ and $v_1 \leq k_1 \leq w_1$. Since $(v, w) \in E$ we have

$$D[v, w] > |v_1 - w_1| + 3\epsilon$$

If $(v, k) \in E'$, then k is in the same connected component of G' as p and q. Otherwise, we have

$$D[v, k] \leq |k_1 - v_1| + \epsilon$$

By the triangle inequality we have $D[v, w] \leq D[v, k] + D[k, w]$. Combining these equations we get

$$D[k, w] \geq D[v, w] - D[v, k] > |v_1 - w_1| + 3\epsilon - |k_1 - v_1| - \epsilon = |k_1 - w_1| + 2\epsilon \quad (2.1)$$

which means $(k, w) \in E'$.

Moreover, every edge added has an adjacent edge from E, which is a strong edge. Note that we do not add all the weak edges to E'. ■

Definition 2.1.4 We say that two connected components of G' overlap if and only if a) there is no vertical line l that separates the elements of the first component from the elements of the second component, and b) l does not intersect any point.

Claim 2.1.3 reveals the structure of G'. More specifically, no two connected components overlap. This structure is exactly the desired one. We do not want to have strong edges between the connected components, and we want them not to overlap, such that we can guess the orientation of each component. Note that we do not care if we can have weak edges between the components.

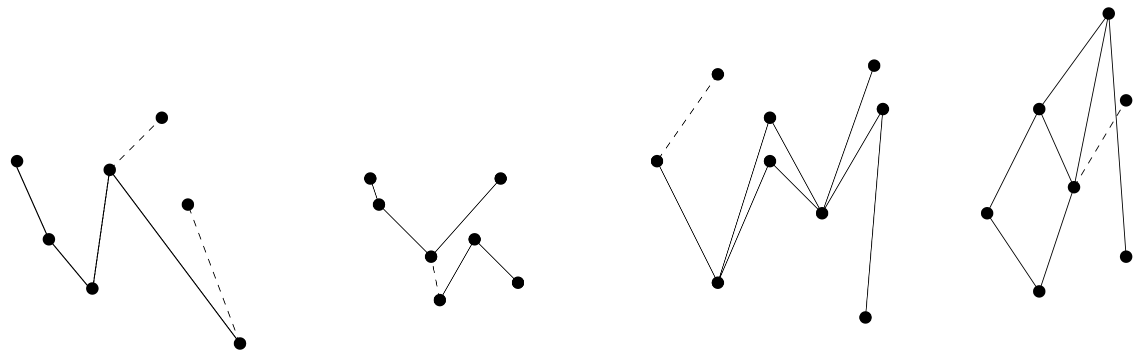


Figure 2-1: The structure of G': G' has 4 connected components. Strong edges are shown with solid lines, and weak edges with dotted lines. There are no strong edges between the connected components. Components do not "overlap." Each weak edge is adjacent to at least one strong edge.

For an edge $(v, w) \in E'$ which is oriented up, such that $v_1 \leq w_1$, we have

$$w_2 - v_2 + \epsilon \geq D[v, w] \geq w_2 - v_2 - \epsilon \tag{2.2}$$

Claim 2.1.5 *By fixing the orientation of an edge of G' we also fix the orientation of all the other edges in the same connected component of G'.*

Proof: We first show that if we know the orientation of an edge e, then we can also determine the orientation of any adjacent edge if both e and the adjacent edge are not weak edges. By setting the orientation of an edge and repeating this process we can determine the orientation of all the edges in the connected component.

Without loss of generality, we assume that the edge $(v, w) \in E'$ $(w_1 \geq v_1)$ is oriented up: $w_2 - v_2 > w_1 - v_1$. Also let $(w, t) \in E'$.

If (w, t) is a strong edge and (w, t) is oriented up, by using (2.2) multiple times we get

$$D[v, t] \;\geq\; |t_2 - v_2| - \epsilon \;\geq\; |w_2 - v_2| + |t_2 - w_2| - \epsilon \;\geq\; D[v, w] + D[w, t] - 3\epsilon$$

Since (w, t) is a strong edge, $D[w, t] > 3\epsilon$, therefore

$$D[v, t] > D[v, w] \tag{2.3}$$

Using the fact that (v, w) is an edge and (w, t) is a strong edge (i.e., $D[v, w] > w_1 - v_1 + \epsilon$ and $D[w, t] > |t_1 - w_1| + 3\epsilon$), we get

$$D[v, t] > (w_1 - v_1) + |t_1 - w_1| + \epsilon > |t_1 - v_1| + \epsilon \tag{2.4}$$

If (w, t) is a strong edge and (w, t) is oriented down, we have

$$D[v, t] \leq ||t - v||_\infty + \epsilon \leq \max\{|t_1 - v_1| + \epsilon, D[v, w] + \epsilon - D[w, t] + \epsilon\} \leq \max\{|t_1 - v_1| + \epsilon, D[v, w]\} \tag{2.5}$$

Since equations (2.3) and (2.4) contradict equation (2.5), we can determine whether (w, t) is oriented up or down.

If (w, t) is a weak edge and (v, w) is a strong edge, then by a similar argument we can determine the orientation of (w, t). If a connected component of G' contains only one connected component of G, we can determine the orientation of all the strong edges first and then the orientation of the weak edges. If a connected component of G' contains two connected components of G, then these two components must overlap (by the way we add weak edges), which means that there is a weak edge connecting them. This means that we can determine the orientation of the strong edges in the first component, then the orientation of a weak edge between the two components, then the orientation of the strong edges of the second component, and finally the orientation of all the remaining weak edges. The same argument applies to the case when the connected component of G' is composed of several connected components of G. ■

Claim 2.1.6 *Given the orientation of all the strong edges, we can compute a 3-approximation solution via linear programming.*

Proof: We construct the following linear program:

$$\text{Min} \quad \delta$$

subject to

$$D[p,q] - \delta \geq q_2 - p_2 \geq D[p,q] + \delta, \text{ if } (p,q) \in E \text{ is oriented up and } q_1 \geq p_1$$
$$D[p,q] - \delta \geq p_2 - q_2 \leq D[p,q] + \delta, \text{ if } (p,q) \in E \text{ is oriented down and } q_1 \geq p_1$$
$$-D[p,q] - \delta \leq q_2 - p_2 \leq D[p,q] + \delta, \text{ if } (p,q) \notin E$$

First note that, if we were to have all the edges (including the weak ones) in E, we would get an optimal solution. However, we know the orientation of only the strong edges, and this gives a 3-approximation solution: It is clear that a 3-approximation solution that satisfies the orientation given is a feasible solution for this linear program. It is also clear that any solution of this program is an embedding with error of at most $\max(\delta^*, 3\epsilon)$ and $\delta^* \leq 3\epsilon$. ■

By using Claims 2.1.5 and 2.1.6 we can get an approximate solution to the problem. But what if we have several connected components in G'? We will prove that no matter what relative orientation we take between the edges of the components, we will still get a constant approximation solution. So, the algorithm is as follows: we choose an arbitrary orientation to one edge from each connected component, and by using Claims 2.1.5 and 2.1.6 we get an approximate solution.

Claim 2.1.7 *There is a 5-approximation solution for every relative orientation between the edges of the components.*

Proof: Let f be an optimal embedding. Let $C_1, C_2, C_3, \ldots, C_k$ be the connected components of G', from the ones with the smallest x coordinate to the ones with higher x coordinate: $\forall i$, if $v \in C_i$ and $w \in C_{i+1}$, then $v_1 \leq w_1$.

Choose any relative orientation of the components. Let the function $s : \{1, 2, \ldots, k\} \to \{0, 1\}$ denote the relative orientation of our arbitrary choice to the optimal embedding f: $s(i) = 1$ if the orientation of the component C_i is different in f than in our arbitrary selection; $s(i) = 0$ otherwise.

We are going to start with an optimal solution and modify it to get a feasible solution that has the given relative orientation, with error of at most 5ϵ:

We are going to modify f incrementally from the first component to the last. Without loss of generality we can assume $s(1) = 0$. (If $s(1) = 1$ we can flip (or reflect) f by the x axis, flipping each $s(i)$ and still having an optimal embedding.)

We repeat the following steps for i from 2 to k:

If $s(i) = 0$, then we go to the next component C_{i+1}.

If $s(i) = 1$, then we will flip all the points in $\bigcup_{j=i}^{k} C_j$ by a certain line parallel to the x axis. This flip will change the values of $s(j)$ for all $j \geq i$. The line by which we are going to flip the points is computed as follows:

- Let p be a point in $\bigcup_{j=1}^{i-1} C_j$ that maximizes $p_1 + p_2$: $\forall v \in \bigcup_{j=1}^{i-1} C_j, p_1 + p_2 \geq v_1 + v_2$.
- Let q be a point in $\bigcup_{j=1}^{i-1} C_j$ that maximizes $q_1 - q_2$: $\forall v \in \bigcup_{j=1}^{i-1} C_j, q_1 - q_2 \geq v_1 - v_2$.
- Let l_1 be the line of slope $-\pi/4$ that passes through p, and let l_2 be the line of slope $\pi/4$ that passes through q. Let the point r denote the intersection of l_1 with l_2. Let l be the line through r that is parallel to the x axis.
- Flip the points in $\bigcup_{j=i}^{k} C_j$ by l: for all $v \in \bigcup_{j=i}^{k} C_j$, $v'_2 = 2r_2 - v_2$

It is easy to see that by performing this flip operation on f we change only the distances between v and w for $v \in \bigcup_{j=1}^{i-1} C_j$ and $w \in \bigcup_{j=i}^{k} C_j$. The question is: by how much? Since there are no strong edges between the components, for such v and w, we have:

$$D[v, w] \leq |v_1 - w_1| + 3\epsilon \tag{2.6}$$

If $v \in C_{i-1}$, then by the flips up to this step, the distance between v and w has remained the same as in the original embedding:

$$D[v, w] + \epsilon \geq ||v - w||_\infty \geq |w_2 - v_2| \tag{2.7}$$

Combining equations (2.6) and (2.7) we get

$$|w_2 - v_2| \leq D[v, w] + \epsilon \leq |w_1 - v_1| + 4\epsilon \tag{2.8}$$

Let v' be the point 4ϵ to the left of v: $v'_1 = v_1 - 4\epsilon$ and $v'_2 = v_2$. Let d' be the line of slope $\pi/4$ that goes through v' and d'' be the line of slope $-\pi/4$ that goes through v'.

It is easy to see that equation (2.8) implies w is located beneath d' and above d''. (See Figure 2-2)

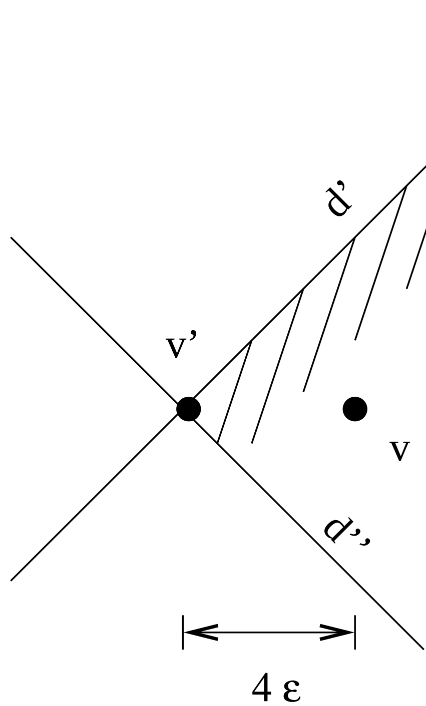


Figure 2-2: The points $w \in \bigcup_{j=i}^{k} C_j$ are located beneath d' and above d''.

This holds for each $v \in C_{i-1}$. Therefore each $w \in \bigcup_{j=i}^{k} C_j$ is located between these wedges. Now, our line l is chosen such that after we flip the points $w \in \bigcup_{j=i}^{k} C_j$, they will remain between these wedges, such that equation (2.8) remains true after the flip. [4] Since we do not change the x coordinates and the original f is a solution

[4]The intersection of the space between pairs of wedges has the same shape as the space between 2 wedges and l divides this intersection into 2 symmetrically equal pieces.

with error ϵ we know that

$$D[v,w] + \epsilon \geq |w_1 - v_1| \tag{2.9}$$

Combining equations (2.8) and (2.9), we get

$$|w_2 - v_2| \leq |w_1 - v_1| + 4\epsilon \leq D[v,w] + 5\epsilon$$

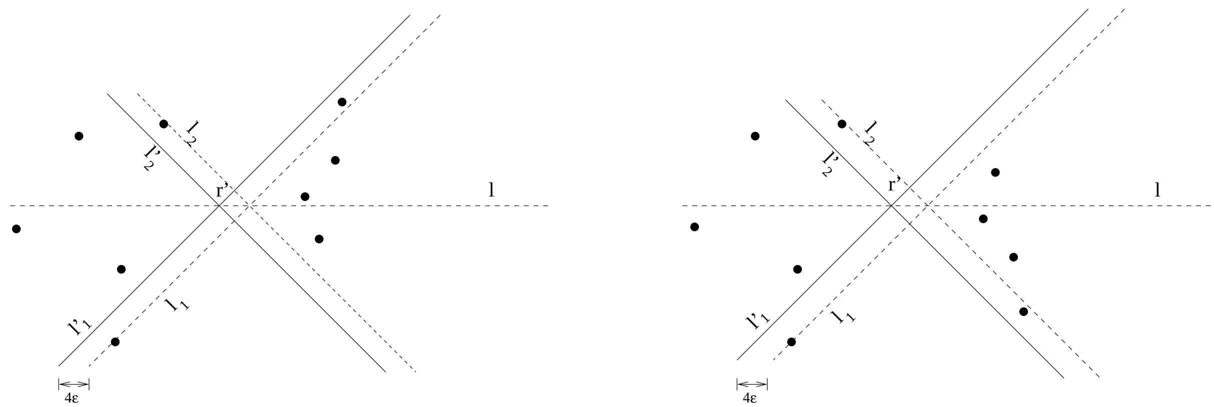


Figure 2-3: The points in $\bigcup_{j=i}^{k} C_j$ will remain between the two lines (wedges) l'_1 and l'_2 after the flip.

In addition, since for the next flips, the points that are being flipped are inside an even more restrictive space between two wedges, they will not leave the space between the two wedges and equation (2.8) will remain true after we have completed all the flips. Again, since equation (2.9) is also true, we can combine them and get that $|w_2 - v_2| \leq D[v,w] + 5\epsilon$ after all the flips are completed. By applying this argument to $v \in C_1, v \in C_2, \ldots$, we conclude that for all points v, w, $w_1 \geq v_1$, we have $|w_2 - v_2| \leq D[v,w] + 5\epsilon$. Since we have no strong edges between the components, for $v \in C_i$, $w \in C_j$, $i \neq j$, $|v_1 - w_1| \geq D[v,w] - 3\epsilon$. This implies that for two points from different components, the distortion in our construction is at most 5ϵ. Since we preserve the distances between points from the same component, we have constructed an embedding with distortion of at most 5ϵ for an arbitrary relative orientation of the edges between the connected components.

■

Finally, we apply Claim 2.1.6 to produce a 5-approximation solution. We know that for our orientation there is an embedding with error 5ϵ and this is a feasible solution for our linear program.

2.1.4 The Final Algorithm

So far, we assumed we know certain points (the pair of points that give the diameter, etc). To satisfy this assumption, we will iterate over all possible choices (a polynomial number of such choices). The total running time of the algorithm is $O(\log \frac{diam}{\epsilon^*} n^6 LP)$ where $diam$ is the value of the diameter of P and LP is the time to solve a linear program with n variables and $O(n)$ constraints. Thus, the running time is polynomial in n.

2.1.5 The general case

The main idea is to fix the x coordinates of the points and then to use the algorithm from the previous section. We do not need to guess the x coordinates exactly. If we guess them within $c\epsilon$ for a constant c, it will be enough to get a constant approximation algorithm for the general case.

Let p and q denote the diameter. Let f be an optimal embedding with error ϵ. Without loss of generality we have $p = (0, 0)$ and we assume that the diameter is given by $q_1 - p_1$.

Let A be the set of points $v \in P - \{p, q\}$ for which the following equation is true:

$$D[p, q] + k\epsilon \geq D[p, v] + D[v, q]$$

$$A = \{v \in P - \{p, q\} : D[p, q] + k\epsilon \geq D[p, v] + D[v, q]\}$$

for a fixed constant k which we will chose later.[5]

[5] k will be chosen to be any constant greater than 9.

Note that for $v \in A$, we have the following two inequalities:

$$v_1 \leq D[p, v] + \epsilon$$

$$v_1 \geq D[p, q] - D[v, q] - 2\epsilon \geq D[p, v] - (k + 2)\epsilon$$

It follows that by fixing $v_1 = \frac{2D[p,v]-(k+1)\epsilon}{2} = D[p, v] - \frac{(k+1)\epsilon}{2}$ we are within an additive factor of $\frac{(k+3)\epsilon}{2}$ from the value of v_1 in the optimal embedding f.

It is clear that if $A = P - \{p, q\}$ then we can guess the x coordinate of all the points in A and then by reducing the problem to the special case with $\epsilon' = \frac{(k+3)\epsilon}{2}$ we get a $\frac{5(k+3)}{2}$-approximation algorithm.

But what happens if $P - A \neq \emptyset$? We break our analysis into two cases:

Case 1

For this case we assume that for all points $v \in P - A - \{p, q\}$ we have either $||p - v||_\infty = |p_1 - v_1|$ or $||q - v||_\infty = |q_1 - v_1|$.

Partition the points of $P - A - \{p, q\}$ into three sets:

$$B = \{v \in P - A - \{p, q\} : ||p - v||_\infty = |p_1 - v_1|\}$$

$$C = \{v \in P - A - \{p, q\} : ||q - v||_\infty = |q_1 - v_1| \text{ and } v_2 - p_2 \geq 0\}$$

$$D = \{v \in P - A - \{p, q\} : ||q - v||_\infty = |q_1 - v_1| \text{ and } v_2 - p_2 < 0\}$$

Note that this is a partition: If $B \cap C \neq \emptyset$ then there exists $v \in B \cap C$, and we have $D[p, q] \geq q_1 - p_1 - \epsilon = q_1 - v_1 + v_1 - p_1 - \epsilon = ||q - v||_\infty + ||v - p||_\infty - \epsilon \geq D[q, v] + D[p, v] - 3\epsilon$ which implies $v \in A$, which is a contradiction.

The idea is to choose certain points to decide for each point $v \in P - A - \{p, q\}$ if $v \in B$ or $v \in C \cup D$. If we can decide that, we can approximate its x coordinate within an additive error of $c\epsilon$ for some constant c.

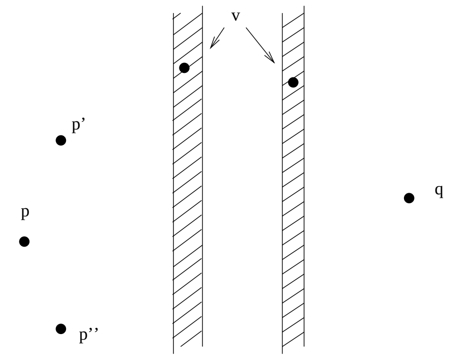


Figure 2-4: If $v \in B$ then v is restricted to the right stripe of width 2ϵ. If $v \in C \cup D$ then v is restricted to the left stripe of width 4ϵ. Since v is not in A, we know that the distance between the stripes is at least $(k-2)\epsilon$

Let the point $p' \in C$ be such that $p' = \min_{v \in C} v_1 + v_2$ and the point $p'' \in D$ such that $p'' = \min_{v \in D} v_1 - v_2$. Of course, p' or p'' may not even exist, but those cases are easier to handle and the proof is basically the same for them as well.

Since $p' \notin A$, we have

$$D[p', q] > D[p, q] + k\epsilon - D[p, p'] \tag{2.10}$$

Since $p' \in C$, we have the following inequalities

$$p'_2 - \epsilon \leq D[p', p] \leq p'_2 + \epsilon \tag{2.11}$$

$$D[p, q] - D[p', q] - 2\epsilon \leq p'_1 \leq D[p, q] - D[p', q] + 2\epsilon \tag{2.12}$$

By combining (2.10) with (2.12) we get that $p'_1 < D[p, p'] - (k-2)\epsilon$. Using (2.11) we get that

$$p'_1 < p'_2 - (k-3)\epsilon \tag{2.13}$$

For a point $v \in B$, we have

$$\begin{aligned} D[p', v] &\geq |v_1 - p'_1| - \epsilon \\ &\geq D[p, v] - 2\epsilon - p'_1 \\ &> D[p, v] - p'_2 + (k-5)\epsilon && \text{by (2.13)} \\ &\geq D[p, v] - D[p, p'] + (k-6)\epsilon && \text{by (2.11)} \end{aligned} \tag{2.14}$$

Also,

$$\begin{aligned} D[p', v] &\geq |v_1 - p'_1| - \epsilon \\ &\geq D[p, v] - 2\epsilon - p'_1 \\ &\geq D[p, v] - D[p, q] + D[p', q] - 4\epsilon && \text{by (2.12)} \end{aligned} \tag{2.15}$$

For a point $v \in C$, we have $D[p', v] \leq ||p', v||_\infty + \epsilon \leq \max(v_1 - p'_1, v_2 - p'_2) + \epsilon$ by the way p' was chosen.

If $||p', v||_\infty = v_2 - p'_2$, we have

$$\begin{aligned} D[p', v] &\leq v_2 - p'_2 + \epsilon \\ &\leq D[p, v] - p'_2 + 2\epsilon \\ &\leq D[p, v] - D[p, p'] + 3\epsilon && \text{by (2.11)} \end{aligned} \tag{2.16}$$

Note that if $k > 9$, equations (2.14) and (2.16) are contradictory.

If $D[p',v] = v_1 - p'_1 + \epsilon$, we have

$$\begin{aligned}
D[p',v] &= v_1 - p'_1 + \epsilon \\
&\leq D[p,q] - D[q,v] - p'_1 + 3\epsilon \\
&< D[p,v] - (k-3)\epsilon - p'_1 && \text{by (2.10)} \\
&\leq D[p,v] - (k-3)\epsilon \\
&- (D[p,q] - D[p',q] - 2\epsilon) && \text{by (2.12)} \\
&= D[p,v] - D[p,q] \\
&+ D[p',q] - (k-5)\epsilon
\end{aligned} \tag{2.17}$$

Note that if $k > 9$, equations (2.17) and (2.15) are contradictory.

We can also obtain similar equations for $v \in D$ by replacing p' with p'' in the above argument. We use these observations to prove the following claim:

Claim 2.1.8 *We can determine which points are in B and which ones are in $C \cup D$ if $k > 9$.*

Proof: If $v \in B$, by (2.14) and (2.15), the following equations are true:

$$D[p',v] \geq D[p,v] - D[p,p'] + (k-6)\epsilon$$

$$D[p',v] \geq D[p,v] - D[p,q] + D[p',q] - 4\epsilon$$

$$D[p'',v] \geq D[p,v] - D[p,p''] + (k-6)\epsilon$$

$$D[p'',v] \geq D[p,v] - D[p,q] + D[p'',q] - 4\epsilon$$

If $v \in C \cup D$ then either (2.16) or (2.17) is true (for p' if $v \in C$ or for p'' if $v \in D$) which implies that at least one of the above 4 equations is false. Therefore we say that $v \in B$ if all the about 4 equations are true, and $v \in C \cup D$ otherwise. ■

Case 2

For this case we assume that there exists a point $r \in P - \{p, q\}$ for which the following is true: $||p-r||_\infty > |p_1 - r_1|$ and $||q-r||_\infty > |q_1 - r_1|$. It follows that $||p-r||_\infty = |p_2 - r_2|$ and $||q - r||_\infty = |q_2 - r_2|$.

Let r be such a point that maximizes $|r_2 - p_2|$:

$$r = \max_{\substack{v \in P \\ ||p-r||_\infty = |p_2 - r_2| \\ ||q-r||_\infty = |q_2 - r_2|}} |r_2 - p_2|$$

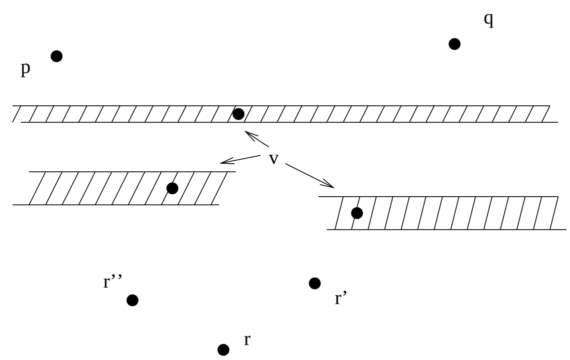


Figure 2-5: If $v \in B$ then v is restricted to the upper stripe of width 2ϵ. If $v \in C$ then v is restricted to the lower right stripe of width 4ϵ. If $v \in D$ then v is restricted to the lower left stripe of width 4ϵ

For this case, we will fix the y coordinate instead. The method we will use is going to be very similar to the method used for case 1. We partition the points of $P - \{p, q, r\}$ into four sets as follows:

$$A = \{v \in P - \{r, p, q\} : D[r, p] + k\epsilon \geq D[r, v] + D[v, p] \text{ or } D[r, q] + k\epsilon \geq D[r, v] + D[v, q]\}$$

$$B = \{v \in P - A - \{r, p, q\} : ||r - v||_\infty = |r_2 - v_2|\}$$

$$C = \{v \in P - A - \{r,p,q\} : ||r-v||_\infty = |r_1 - v_1| \text{ and } v_1 - r_1 \geq 0\}$$

$$D = \{v \in P - A - \{r,p,q\} : ||r-v||_\infty = |r_1 - v_1| \text{ and } v_1 - r_1 < 0\}$$

Claim 2.1.9 *If $v \in C$ we can determine its y coordinate within an additive factor of 2ϵ*

Proof: By the way of contradiction, we suppose that $||q-v||_\infty = q_1 - v_1$. Then $D[v,q] \geq q_1 - v_1 - \epsilon$. It follows that $D[r,q] \geq q_1 - r_1 - \epsilon \geq q_1 - v_1 + v_1 - q_1 - \epsilon \geq D[r,v] + D[v,q] - 3\epsilon$. But for $k \geq 3$, this implies $v \in A$, which contradicts the fact that $v \in C$.

Therefore, we know that $||q-v||_\infty = |q_2 - v_2|$. If $v_2 > q_2$ we have that $v_1 > v_2 > q_2 > q_1$ which means that the diameter is given by the $|p_1 - v_1|$, again impossible. Therefore,

$$||q-v||_\infty = q_2 - v_2$$

Using this observation, we have the following two bounds on v_2: $D[r,q]-D[r,v]-2\epsilon \leq v_2 \leq D[r,q]-D[r,v]+2\epsilon$ and we can guess v_2's real value within an additive distance of 2ϵ by setting $v_2 = D[r,q] - D[r,v]$. ■

Similarly by replacing q with p in the above proof, if $v \in D$ we can also determine its y coordinate within an additive factor of 2ϵ.

We shift everything and flip it by the y axis if necessary such that, $r = (0,0)$ and $p_2 > 0$. Note that this implies that $q_2 > 0$ (if $q_2 < 0$ then we would have $|p_2 - q_2| = |p_2 - r_2| + |r_2 - q_2| > |p_1 - r_1| + |r_1 - q_1| = |p_1 - q_1| = diam(P)$, which is a contradiction) and that every $v \in B$ has $v_2 \geq 0$ (because we chose r such that it maximizes $r_2 - p_2$). We proceed as before: we pick certain points to help us decide for each point in which set it belongs to. If we can decide that, we can approximate for each point its y coordinate within a constant times ϵ.

Let the point $r' \in C$ be such that $r' = \min_{v \in C} v_1 + v_2$ and the point $r'' \in D$ such that $r'' = \min_{v \in D} v_1 - v_2$. Of course, as before, r' or r'' may not even exist, but those cases are easier to handle and the proof is basically the same for them as well.

Since $r' \notin A$, we have

$$D[r', q] > D[r, q] + k\epsilon - D[r, r'] \tag{2.18}$$

Since $r' \in C$, we have the following inequalities

$$r'_1 - \epsilon \leq D[r', r] \leq r'_1 + \epsilon \tag{2.19}$$

$$D[r, q] - D[r', q] - 2\epsilon \leq r'_2 \leq D[r, q] - D[r', q] + 2\epsilon \tag{2.20}$$

By combining (2.18) with (2.20) we get that $r'_2 < D[r, r'] - (k-2)\epsilon$. Using (2.19) we get that

$$r'_2 < r'_1 - (k-3)\epsilon \tag{2.21}$$

For a point $v \in B$, we have

$$\begin{aligned} D[r', v] &\geq |v_2 - r'_2| - \epsilon \\ &\geq D[r, v] - 2\epsilon - r'_2 \\ &> D[r, v] - r'_1 + (k-5)\epsilon && \text{by (2.21)} \\ &\geq D[r, v] - D[r, r'] + (k-6)\epsilon && \text{by (2.19)} \end{aligned} \tag{2.22}$$

Also,

$$\begin{aligned} D[r', v] &\geq |v_2 - r'_2| - \epsilon \\ &\geq D[r, v] - 2\epsilon - r'_2 \\ &\geq D[r, v] - D[r, q] + D[r', q] - 4\epsilon && \text{by (2.20)} \end{aligned} \tag{2.23}$$

For a point $v \in C$, we have $D[r', v] \leq \max(v_2 - r'_2, v_1 - r'_1) + \epsilon$ by the way r' was chosen.

If $||r', v||_\infty = v_1 - r'_1$, we have

$$\begin{aligned} D[r', v] &\leq v_1 - r'_1 + \epsilon \\ &\leq D[r, v] - r'_1 + 2\epsilon \\ &\leq D[r, v] - D[r, r'] + 3\epsilon \qquad \text{by (2.19)} \end{aligned} \tag{2.24}$$

Note that if $k > 9$, equations (2.22) and (2.24) are contradictory.

If $||r', v||_\infty = v_2 - r'_2$, we have

$$\begin{aligned} D[r', v] &\leq v_2 - r'_2 + \epsilon \\ &\leq D[r, q] - D[q, v] - r'_2 + 3\epsilon \\ &< D[r, v] - (k-3)\epsilon - r'_2 \qquad \text{by (2.18)} \\ &\leq D[r, v] - (k-3)\epsilon \\ &- (D[r, q] - D[r', q] - 2\epsilon) \qquad \text{by (2.20)} \\ &= D[r, v] - D[r, q] + D[r', q] \\ &- (k-5)\epsilon \end{aligned} \tag{2.25}$$

Note that if $k > 9$, equations (2.25) and (2.23) are contradictory.

We also have very similar equations for $v \in D$ by using r''. We use these observations to prove the following claim.

As before, we can use these equations to determine which points are in B and which ones are in $C \cup D$. However, in this case, we should also distinguish between C and D. We make the following observation:

Claim 2.1.10 *If $D[q, v] + D[v, r'] \leq D[r', q] + 3\epsilon$ or $D[p, v] + D[v, r''] \leq D[r'', p] + 3\epsilon$ we can approximate v_2 within a factor of 3ϵ. Otherwise, if $v \in C$, then $D[r', v] < D[r'', v]$ and if $v \in D$ then $D[r', v] > D[r'', v]$.*

Proof: First note that $D[r', r'']$ has to be pretty large:

$$\begin{aligned} D[r', r''] &\geq r_1' - r_1 + r_1 - r_1'' - \epsilon \\ &\geq D[r', r] + D[r'', r] - 3\epsilon \\ &> D[r, q] - D[r', q] + D[r, p] \\ &- D[r'', p] + (2k-3)\epsilon \\ &\geq q_2 - r_2 - \epsilon - (q_2 - r_2' + \epsilon) \\ &+ p_2 - r_2 - \epsilon - (q_2 - r_2'' + \epsilon) + (2k-3)\epsilon \\ &= r_2' - r_2 + r_2'' - r_2 + (2k-7)\epsilon \\ &\geq (2k-7)\epsilon \end{aligned} \tag{2.26}$$

We break our proof into three cases:

- Case 1: $D[q, v] + D[v, r'] \leq D[r', q] + 3\epsilon$

  First note that we know that

  $$v_2 \geq q_2 - D[q, v] - \epsilon \geq D[r, q] - D[q, v] - 2\epsilon \tag{2.27}$$

  We also have that

  $$v_2 \leq r_2' + D[v, r'] + \epsilon \tag{2.28}$$

  $$\leq D[r, q] - D[q, r'] + D[v, r'] + 3\epsilon \tag{2.29}$$

  $$\leq D[r, q] - D[q, v] + 6\epsilon \tag{2.30}$$

  2 Therefore, as before, we can approximate v_2 within an additive error of 4ϵ by setting $v_2 = D[r, q] - D[q, v] + 2\epsilon$.

- Case 2: $D[p, v] + D[v, r''] \leq D[r'', p] + 3\epsilon$

  The analysis of this case is analogous to the one above. (replace r' by r'' and q by p).

- Case 3: If $||r' - v||_\infty = v_2 - r'_1$ we have that $D[q,v] + D[v,r'] \leq D[r',q] + 3\epsilon$, which falls into case 1. Therefore, we know that $||r' - v||_\infty = v_1 - r'_1$. In that case,

$$\begin{aligned} D[r'',v] &\geq v_1 - r'_1 + r'_1 - r''_1 - \epsilon \\ &\geq D[v,r'] + r'_1 - r''_1 - 2\epsilon \\ &\geq D[v,r'] + D[r',r''] - 3\epsilon \\ &> D[v,r'] + (2k-10)\epsilon \qquad \text{by (2.26)} \end{aligned} \tag{2.31}$$

Therefore, for $v \in C$ we have $D[r'',v] > D[v,r']$ if $k \geq 5$. Symmetrically, for $v \in D$ we have $D[r'',v] < D[v,r']$ for $k \geq 5$.

We conclude that it is either the case that we can approximate v_2 within 4ϵ (case 1 or 2) or we can compare $D[v,r'']$ with $D[v,r']$ to determine if $v \in C$ or $v \in D$ which implies we can approximate v_2 within additive error 2ϵ.

■

Using this observation we can easily distinguish between points in C and points in D. For the points in $v \in C$ we fix the y coordinate as $v_2 = D[r,q] - D[q,v]$ and for the points $v \in D$, $v_2 = D[r,p] - D[p,v]$.

2.1.6 Conclusions

In this section, we showed how to approximate within a constant factor an embedding of an arbitrary metric into a two-dimensional space where distances are computed using the l_1 norm with the notion of an additive error ϵ. Our constant is 30, but by combining the general case with the special case more carefully, we believe can get the constant down to 19, by just a tighter analysis in the constants.

Future Work. We believe the distortion and the running time can be improved further. We also believe that the same technique might be extended to get the same result for other norms (e.g., l_2) or multiplicative error. It would also be of interest to extend this result for higher dimensions - let's say three-dimensional space. It should also be noticed that in the case of the l_2 norm, while we don't know how to prove claim (2.1.7) with additive error, it is easy to prove it with multiplicative error and we can obtain an embedding $f : P \to \Re^2$, for which $D[p, q] \leq ||f(p) - f(q)||_2 \leq aD[p, q] + b\epsilon^*$, where a and b are absolute constants.

Acknowledgements. The author would like to thank Piotr Indyk and Sariel Har-Peled for numerous comments on early drafts of the results and in particular to Piotr Indyk for useful discussions.

2.2 Constant-factor approximation of the average distortion of embedding a metric into the line.

Credits: The results in this section is work done with Piotr Indyk and Yuri Rabinovich in the autumn of 2002. The results haven't been published yet.

The *average distance* of D, $\text{av}(D) = 1/n^2 \sum_{x,y \in P} D(x, y)$, is the average of distances of n^2 *ordered* pairs of points. In this section we consider non-contracting (or expanding) embeddings f of metric D into a host space Y, i.e., such that the induced submetric D' of Y dominates D (no distance decreases).

Let $\text{av}_Y(D)$ be the minimum of $\text{av}(D')$ over all such D'. We will show that, given D, one can $O(1)$-approximate $\text{av}_{\text{line}}(D)$, where the host space is $\Re$.

The *average distortion* of f is defined as $\text{av}(D')/\text{av}(D) \geq 1$. The *median* of the metric D on P be the point $p \in P$ minimizing the expression $1/n \sum_{x \in P} D(p, x)$; it will be denoted by med. Observe that for a set of points on a line, the standard order-median coincides with the metric-median.

We start with the following simple fact:

$$\text{av}(D) \;\geq\; \frac{1}{n} \sum_{x \in P} D(\text{med}, x) \;\geq\; 1/2 \cdot \text{av}(D) \; . \tag{2.32}$$

The first inequality simply says that the average value of $1/n \sum_{x \in P} D(p, x)$, is no less than its minimum. The second inequality is true in fact for any $p \in P$:

$$\sum_{x,y \in P} D(x,y) \;\le\; \sum_{x,y \in P} D(x,p) + D(p,y) \;=\; 2n \sum_{x \in P} D(p,x)\ . \qquad (2.33)$$

Thus, in order to approximate $\mathrm{av}_{\mathrm{line}}(D)$ it suffices to approximate the minimum possible value of $\sum_{x \in P} D'(\mathrm{med}, x)$ over all D-dominating line metrics D'.

Let *first* be the leftmost point of D'. It turns out that instead of working with the (relatively inconvenient) median, one can work with the "nice" *first*. This follows at once from the following claim:

Claim 2.2.1 *Let $S \subseteq \Re$ be a set of points $\{s_1, s_2, .., s_n\}$, in the left-to-right order. Then there exists, and is efficiently computable, a set $T = \{t_1, t_2, .., t_n\} \subseteq \Re$ (again in the left-to-right order) such that the induced metric δ of S is dominated by induced metric δ' of T (under the natural correspondence $s_i \mapsto t_i$), and*

$$\sum_i |s_i - s_{n/2}| \;\ge\; c \cdot \sum_i |t_i - t_1|$$

for some universal constant $c > 1/9$.

Proof: We shall use an idea from the solution of the so-called Lost Cow Problem [BCR93]. Assume for simplicity that $s_{n/2} = 0$. If $s_1 = 0$, S itself can serve as $\bar{S}$. Otherwise, assume w.l.o.g., that the distance between $s_{n/2}$ and the closest (but not identical) point of S lying to its left is 1. Consider a particle which starts at 0, and moves at a unit speed in the following manner:

$$0 \;\rightarrow\; -1 \;\rightarrow\; 2 \;\rightarrow\; -4 \;\rightarrow\; ... \;\rightarrow -2^{2k} \;\rightarrow\; 2^{2k+1} \;\rightarrow\; ...$$

Define t_i as the moment of time when the particle first sees s_i. E.g., $t_1 = 0$, $t_2 = 1$ and so on. We have obtained T whose induced metric dominates that of S: due to the unit speed, it is impossible get from s_i to s_j faster that in $|s_i - s_j|$ units of time. The (by now folklore) analysis of [BCR93] shows that T indeed has the other required property.

$\square$

Thus, in order to approximate $\text{av}_{\text{line}}(D)$, it suffices to approximate $\sum_{x \in P} D'(\text{first}, x)$ over all D-dominating line metrics D'.

Consider a complete graph G_n on P, such that the weight of the edge (i, j) is $D(s_i, s_j)$. A simple but important observation is that the optimal embedding is necessarily the shortest-path metric of some Hamiltonian path of G_n. Indeed, in the optimal embedding $f : (P, D) \mapsto \Re$ the distance between two neighboring points p, q is at least $D(p, q)$ by dominance, and no more that $D(p, q)$ by optimality. On the other hand, by triangle inequality, any such Hamiltonian path yields a legal noncontracting embedding.

In order to solve the problem, note that if we were after a Hamiltonian *tour* (rather than a path), aiming to minimize $(\sum_{x \in P} D'(\text{first}, x)) + D'(\text{last}, \text{first})$, then the corresponding problem were precisely the well-known *Minimum Latency problem*. A constant factor approximation for this problem were first obtained by Blum et al. [BCCPRS94], and subsequently improved by Goemans and Kleinberg [GK98]. However, the difference between a path and a tour is negligeable, and therefore these algorithms provides a solution to our path problem as well. To see that it suffices to observe that sum of weights rising from to a path is at least half the sum of weights rising from the corresponding tour.

To summarize, the approximation algorithm for our problem simply produces the embedding corresponding to the pseudo-optimal Hamiltonian path produced by the best known algorithm for the Minimum Latency problem. The approximation factor is the product of that of the latter, $\times 2$ for passing from tour to path, $\times 9$ for working with *first* instead of median, $\times 2$ for optimizing the average distance from the median instead the actual average distance $\text{av}(D)$. We have established the following theorem:

Theorem 2.2.2 *There is a (polynomial-time) $O(1)$-approximation algorithm for computing a non-contractive embedding f of a given metric into a line that minimizes the average distortion of f.*

2.3 The 4-points criterion for additive distortion into a line

Credits: The results in this section is work done with Piotr Indyk and Yuri Rabinovich in the autumn of 2002. The results haven't been published yet.

A classical result of Menger states that a metric space (X, D) embeds isometrically into Euclidean space R^m if and only if every subspace of (X, D) on at most $m + 3$ points embeds.

In this section we prove that given a metric space (X, D) for which every subspace of (X, D) on at most 4 points embeds into a line with distortion at most ϵ, then the whole metric embeds into a line with distortion at most 6ϵ. Since our proof is constructive, we automatically obtain an algorithm for computing such an embedding. Although the approximation ration is weeker than previously known, still this 4-points criterion appears to be very useful.

Definition 5 *A metric $M = (X, D)$ embeds into a line with additive distortion ϵ if there exist a mapping $f : X \to \Re$ such that $\forall x, y \in X$,*

$$D(x, y) - \epsilon \leq |f(x) - f(y)| \leq D(x, y) + \epsilon$$

Theorem 2.3.1 *Let $M = (X, D)$ be an arbitrary metric space. If every subspaces of M on at most 4 points embeds into a line with additive distortion at most ϵ, then there exists an embedding of M into a line with additive distortion 6ϵ.*

Proof: Let $x, y \in X$ such that $D(x, y) = \max_{p,q} D(p, q)$. If $\epsilon \geq D(x, y)$, then mapping all the points into a point is a good enough solution. Therefore, we can assume that $\epsilon < D(x, y)$. Consider the subspaces on 4 points that contain x and y. Without loss of generality we can assume that x is to the "left" of y, $x < y$.

It is easy to see that if $S = \{x, y, a, b\}$ such that a has lower coordinate than x in an optimal embedding of S into the line, then $|f^*(x) - f^*(a)| \leq 2\epsilon$ and $D[x,a] \leq 3\epsilon$:

$$|f(x)^* - f(a)^*| = f(x)^* - f(a)^* + f(y)^* - f(y)^* = |f(y)^* - f(a)^*| - |f(y)^* - f(x)^*| \leq$$

$$\leq D[y,a] - D[y,x] + 2\epsilon \leq 2\epsilon$$

and

$$D[x,a] \leq |f(x)^* - f(a)^*| + \epsilon \leq 3\epsilon$$

We construct the following embedding: start by placing x at the coordinate 0, and place the remaining points at the coordinate $D[x,p]$ if $D[x,p] > 3\epsilon$, and at the coordinate ϵ otherwise.

Formally, the embedding f is defined as follows:

- $f(x) = 0$
- $f(p) = D[x,p]$, if $D[x,p] > 3\epsilon$
- $f(p) = \epsilon$, if $D[x,p] \leq 3\epsilon$

All that one needs to prove is that for any 2 points $a, b \in X$, $D[a,b] - 4\epsilon \leq |f(a) - f(b)| \leq D[a,b] + 6\epsilon$. We split the analysis into 3 cases:

Case 1: $D[x,a] > 3\epsilon$ and $D[x,b] > 3\epsilon$

In the optimal embedding of $S = \{x, y, a, b\}$, $f^*(a) > 0$ and $f^*(b) > 0$, giving $|f^*(a) - D[x,a]| \leq \epsilon$ and $|f^*(b) - D[x,b]| \leq \epsilon$. Therefore $|f^*(a) - f(a)| \leq \epsilon$ and $|f^*(b) - f(b)| \leq \epsilon$ which gives that

$$|f(a) - f(b)| \leq |f^*(a) - f^*(b)| + 2\epsilon \leq D[a,b] + 3\epsilon$$

Similarly,

$$|f(a) - f(b)| \geq |f^*(a) - f^*(b)| - 2\epsilon \leq D[a,b] - 3\epsilon$$

Case 2: $D[x,a] \leq 3\epsilon$ and $D[x,b] \leq 3\epsilon$

By triangle inequality $D[a,b] \leq D[a,x] + D[x,b] \leq 6\epsilon$. Since $f(a) = f(b)$, we have $D[a,b] \geq |f(a) - f(b)| \geq D[a,b] - 6\epsilon$.

Case 3: $D[x, a] > 3\epsilon$ and $D[x, b] \leq 3\epsilon$

As before we have that

$$|f^*(a) - f(a)| \leq \epsilon \tag{2.34}$$

If $f^*(b) \geq 0$, we have $|f^*(b) - D[x, b]| \leq \epsilon$ and

$$|f^*(b) - f(b)| \leq 3\epsilon \tag{2.35}$$

If $f^*(b) \geq 0$, we have the same inequality $|f^*(b) - f(b)| = |f^*(b) - \epsilon| \leq 3\epsilon$

Combining equations (2.34) and (2.35) we get

$$|f(a) - f(b)| \leq |f^*(a) - f^*(b)| + 4\epsilon \leq D[a, b] + 5\epsilon$$

Similarly,

$$|f(a) - f(b)| \geq |f^*(a) - f^*(b)| - 4\epsilon \geq D[a, b] - 5\epsilon$$

■

2.4 Embedding with an Extremum Oracle

Credits: The results in this section is work done with Erik Demaine, Mohammad-Taghi Hajiaghayi, and Piotr Indyk, and has appeared in SoCG'04.

In this section, we describe an $O(1)$-approximation algorithm for minimizing the additive distortion in an embedding of a complete graph with distances specified by D into the Euclidean plane. Define the *spread* Δ of the metric by $\Delta = \text{diam}(D)/\epsilon$, where ϵ is the minimum additive distortion possible and $\text{diam}(D)$ is the diameter of D, i.e., the maximum distance in D. The algorithm runs in polynomial time, multiplied by a factor of $O(\lg \Delta)$ if ϵ is not approximately known, given an extremum oracle for a promised embedding f attaining minimum additive distortion ϵ. By exhaustive enumeration of the possible oracle answers, this algorithm can be converted into an algorithm without extra information having pseudo-quasipolynomial running time

$2^{O(\log n \cdot \log^2 \Delta)}$.

We view the algorithm as being given D and $\epsilon > 0$, and the goal is either to find an embedding of D into the plane with additive distortion $O(\epsilon)$ or to report that no embedding with additive distortion at most ϵ exists. Here we assume that $\epsilon > 0$ (and thus Δ is finite) because it is easy to test whether a complete graph of distances can be embedded without distortion. If ϵ is unknown, we can guess ϵ up to a constant factor in a standard way by trying values of the form diam$/2^i$ for $i = 0, 1, 2, \ldots$. This guessing multiplies the running time by $O(\lg \Delta)$, which is absorbed in the pseudo-quasipolynomial time bound.

We use a geometric annulus (the difference between two disks of the same radii) to represent approximately known distances. Define $R(p, r, \delta)$ to be the annulus centered at point p with inner radius $r - \delta$ and outer radius $r + \delta$. The next lemma shows how two such annuli can help isolate a point.

Lemma 2.4.1 *Consider two points a and b at a distance r on the x axis, and two radii r_a and r_b such that* $\max\{r_a, r_b\} \le 2r$. *Then, for any $\epsilon \le r$, the intersection $R = R(a, r_a, \epsilon) \cap R(b, r_b, \epsilon)$ is enclosed in a vertical slab $[x_0 - 4\epsilon, x_0 + 4\epsilon]$, where $x_0 = (r^2 + r_a^2 - r_b^2)/2r$.*

Proof: By a suitable translation, we may assume without loss of generality that $a = (0, 0)$ and $b = (r, 0)$. Any point $(x, y) \in R$ must satisfy $(r_a - \epsilon)^2 \le x^2 + y^2 \le (r_a + \epsilon)^2$ and $(r_b - \epsilon)^2 \le (x - r)^2 + y^2 \le (r_b + \epsilon)^2$. Subtracting these two bounds, the terms quadratic in x and y cancel out, and we obtain that any point $(x, y) \in R$ must satisfy $|x - (r^2 + r_a^2 - r_b^2)/2r| \le \epsilon |r_a - r_b|/r \le \epsilon(r_a + r_b)/r \le 4\epsilon$. Thus (x, y) is in the vertical slab $[x_0 - 4\epsilon, x_0 + 4\epsilon]$ where $x_0 = r^2 + r_a^2 - r_b^2$. ■

Using this tool, we show how to guess approximate x coordinates; the following lemma is also useful in Section 5.3.

Lemma 2.4.2 *Given a complete graph $G = (V, E)$ with distances specified by D, and given $0 < \epsilon <$ diam$(D)/2$, we can compute in polynomial time a set of guesses of the form $x : V \to \mathbb{R}$ such that, if there is an embedding f of G into the Euclidean plane of minimum additive distortion ϵ, at least one guess satisfies, for a suitable translation* *and rotation $\tilde{f}$ of f, $|\tilde{f}_x(v) - x(v)| \leq 5\epsilon$ for all $v \in V$. We can also ensure that the x coordinates are distinct in each guess.*

Proof: First we guess the diameter pair (a, b) in the embedding f, that is, the pair that maximizes $\|f(p) - f(q)\|$, by trying all pairs such that $D[a, b] \geq \mathrm{diam}(D) - 2\epsilon$. (The diameter pair must satisfy this property because f has additive distortion ϵ.) By suitable translation and rotation $\tilde{f}$ of f, we can assume that $\tilde{f}(a) = (0, 0)$ and $\tilde{f}_y(b) = 0$. Therefore we can assign $x(a) = 0$ and $x(b) = D[a, b]$, and we have that $x(a) = \tilde{f}_x(a)$ and $|x(b) - \tilde{f}_x(b)| \leq \epsilon$.

To guess the remaining x coordinates $\tilde{f}_x(v)$ for vertices $v \notin \{a, b\}$, we proceed as in Bădoiu's algorithm [B$\breve{0}$3]. For any such vertex v, define the region $R_v = R(a, D[a, v], \epsilon) \cap R(b, D[b, v], \epsilon)$. Because $D[a, v] \leq \mathrm{diam}(D) \leq D[a, b] + 2\epsilon < 2D[a, b]$, we can apply Lemma 2.4.1 and set $x(v)$ to the center x_0 of the vertical slab. Because $|x(b) - \tilde{f}_x(b)| \leq \epsilon$ and at worst the errors add, we have that $|x(v) - \tilde{f}_x(v)| < 5\epsilon$.

If two x coordinates are equal, we perturb them slightly, to guarantee that all x coordinates are distinct. By a sufficiently small perturbation, we preserve that $|x(v) - \tilde{f}_x(v)| < 5\epsilon$ for all vertices v. Therefore we obtain a suitable guess x. ■

We assume in the rest of this section that $\epsilon = 1$, by scaling the entires in D by $1/\epsilon$. Thus $\Delta = \mathrm{diam}(D)$.

We claim that it suffices to consider embeddings g with x coordinates given by a suitable guess of Lemma 2.4.2. Consider the translated and rotated optimal embedding $\tilde{f}$. Construct f' by setting $f'_x(v) = x(v)$ and $f'_y(v) = \tilde{f}_y(v)$ for all vertices v. By Lemma 2.4.2, $\|\tilde{f}(v) - f'(v)\| < 5\epsilon$ (for a suitable guess). By the triangle inequality, $|\,\|f'(v) - f'(w)\| - \|\tilde{f}(v) - \tilde{f}(w)\|\,| < 10\epsilon$, so the additive distortion of f' is at most $\epsilon + 10\epsilon = 11\epsilon$.

In addition, we require that each y coordinate in the embeddings we construct is a multiple of ϵ. By a similar argument as above, this assumption increases the additive error by at most ϵ, to $c'\epsilon = 12\epsilon$.

The algorithm uses the divide-and-conquer paradigm to compute the y coordinates in an embedding g (using the x coordinates given by the guess of Lemma 2.4.2). First, we compute the median x_m of the x coordinates of the vertices as mapped by g. Let

V^+ be the set of all points $p \in V$ such that $g(p)$ has x coordinate larger than the median x_m, and let $V^- = V - V^+$. The algorithm proceeds by creating the set of *constraints* on $g(V^+)$ and $g(V^-)$. The constraints have two properties:

1. The constraints are feasible; namely, f' satisfies them.

2. For any mapping g satisfying the constraints, we have $| \|g(p) - g(q)\| - D[p,q]| \leq c$, for all $p \in V^+$ and $q \in V^-$; here c is a certain global constant.

These properties allow us to compute $g(V^+)$ and $g(V^-)$ (while enforcing the constraints) recursively and independently from each other.

The constraints are of the form "$g_y(p) \in Y(p)$", where $Y(p)$ is a finite set of intervals. They are constructed as follows. For $i \geq 1$, define $I_i = (x_m + 2^{i-1} - 1, x_m + 2^i - 1]$; for $i \leq -1$, define $I_i = -I_{-i}$. For each I_i, the algorithm queries the extremum oracle to obtain a point $p^i_{up} \in V$, $f'_x(p^i_{up}) \in I_i$, such that $f'_y(p_{up})$ is maximum. Similarly, the algorithm obtains p^i_{down}. In addition, the algorithm obtains the values $f'_y(p^i_{up})$ and $f'_y(p^i_{down})$ for each i.

With the oracle's answers in hand, the algorithm imposes the following new constraints, for each i, $d \in \{up, down\}$, and $p \in V$:

1. "$g_y(p^i_d) = f'_y(p^i_d)$";

2. if $f'_x(p) \in I_i$, then "$g_y(p) \in [f'_y(p^i_{down}), f'_y(p^i_{up})]$"; and

3. "$g(p) \in R(f'(p^i_d), D[p^i_d, p], c')$". (This latter condition can be expressed as a restriction on $g_y(p)$.)

As mentioned above, after imposing the constraints, the algorithm recurses to find $g(V^+)$ and $g(V^-)$ independently. At the leaf level of recursion (i.e., when we are given only one point p), the algorithm sets $g_y(p)$ to be an arbitrary y coordinate satisfying all constraints (if it exists). If no such y coordinate exists, the algorithm concludes that there is no acceptable embedding for the guess of Lemma 2.4.2 and this set of oracle answers.

The oracle's answers can be implemented by trying all possible choices of the guessed variables. Each combination of a guess from Lemma 2.4.2 and the oracle answers leads to a different execution of the algorithm, ending with either a failure or a final embedding g whose additive distortion can be checked to be at most $c'\epsilon$. The total number of such choices is bounded by $2^{O(\log^2 \Delta)}$, because there are at most $O(\Delta)$ different potential values for the y coordinates of f'. The claimed bound for the running time $T(n)$ follows from the recursion $T(n) = 2^{O(\log^2 \Delta)}[T(n/2) + n^{O(1)}]$. Note that, if we could compute the oracle's answers in polynomial time, our algorithm would have polynomial running time as well.

It is easy to see that the constraints imposed at all stages are consistent with f'. It remains to show that, after $g(V^+)$ and $g(V^-)$ satisfying the constraints are found, then we have $| \|g(p) - g(q)\| - D[p,q]| \leq c$, for all $p \in V^+$, $q \in V^-$, and some global constant $c > 0$. This is done via the following two lemmas.

Lemma 2.4.3 *Consider any two points $a = (x, y)$ and $b = (x', y')$, such that $x' \geq x/2$. Define $b' = (x', y)$ and $I = \{0\} \times \mathbb{R}$. Then, for any r there exists r' such that $I \cap R(a, r, c') \subset R(b', r', c)$ for a fixed constant c.*

The interpretation (and usage) of this lemma is as follows. Consider the points $g(p)$ and $g(q)$ as above, and assume that $g_x(p) \in I_i$, $i < 0$, and $g_x(q) \in I_j$, $j > 0$, such that $(i, j) \neq (-1, 1)$. (We will take care of the case $(i, j) = (-1, 1)$ later.) In the procedure described above, we impose constraints on $g(p)$ of the form "$g(p) \in R(a, r, c')$", for $d \in \{down, up\}$, $r = D[p_d^j, p]$, and $a = f'(p_d^j)$. However, it will be more convenient to consider a different constraint, namely "$g(p) \in R(b', r', c)$", where $b' = (f'_x(q), f'_y(p_d^j))$, because in this way $f'(q)$ and b' have the same x coordinate, a property used in the next lemma. However, we do not know $f'(q)$, so we cannot impose the second constraint explicitly. Fortunately, Lemma 2.4.3 guarantees that the latter constraint is implied by the former. Note that the assumption $x' \geq x/2$ is satisfied by the construction of the intervals I_i and I_j.

Proof:[of Lemma 2.4.3] Without loss of generality, we can assume that $I \cap R(a, r, c')$ is nonempty. In addition, we assume that $I \cap R(a, r, c')$ consists of two disconnected components. (If it consists of only one component, the proof is similar.) Finally, without loss of generality, we can assume that $y = 0$. Denote the upper component (with larger y coordinates) by $Y = \{0\} \times [y_d, y_u]$. Let $q_d = (0, y_d)$, $q_u = (0, y_u)$. Note that $y_u^2 + x^2 = (r + c')^2$, and $y_d^2 + x^2 = (r - c')^2$. By symmetry, it suffices to ensure that $Y \subset R(b', r', c)$.

Define $r' = \|b' - q_u\| = {x'}^2 + y_u^2$. Consider any $(0, z) \in Y$. We need to show (1) $\|b' - (0, z)\|^2 \leq (r' + c)^2$ and (2) $\|b' - (0, z)\|^2 \geq (r' - c)^2$ or $r' < c$. First, $\|b' - (0, z)\|^2 = {x'}^2 + z^2 \leq {x'}^2 + y_u^2 = {r'}^2$. Second, $\|b' - (0, z)\|^2 \geq {x'}^2 + y_d^2 \geq {r'}^2 - 2r'c + c^2$.

By plugging in the expressions for y_d^2, ${r'}^2$, and then y_u^2, we obtain equivalently that

$$ {x'}^2 + (r - c')^2 - x^2 \geq [(r + c')^2 - x^2] + {x'}^2 - 2r'c + c^2, $$

which simplifies to $2r'c - c^2 \geq 2c'r$.

Because $r' \geq \max\{x', y_u\}$, $r' \leq x + y_u$, and (by the assumption) $x' \geq x/2$ and $r' \geq c$, it follows that the last expression is satisfied if $c \geq 4c'$. This proves the lemma.

■

The next lemma is about the following configuration of points: $a = (0, y_a)$, $b = (0, y_b)$, $c = (x, y_c)$, and $d = (x, y_d)$. For any r_a, r_b, r_c, r_d, and s, define two sets:

$$ \begin{aligned} S_1 &= \{(0, y) : y_a < y < y_b\} \cap R(c, r_c, s) \cap R(d, r_d, s), \\ S_2 &= \{(x, y) : y_c < y < y_d\} \cap R(a, r_a, s) \cap R(b, r_b, s). \end{aligned} $$

Lemma 2.4.4 *The difference* $\max_{u \in S_1, v \in S_2} \|u - v\| - \min_{u \in S_1, v \in S_2} \|u - v\|$ *is at most* $3s$*.*

Before we prove this lemma, we show how the two lemmas together imply that, for any two points $p \in V^-$ and $q \in V^+$ satisfying the imposed constraints, we have $\|g(p) - g(q)\| = \|f'(p) - f'(q)\| \pm O(1)$ as desired. To show this implication, we consider two cases. Let $f'_x(p) \in I_i$ and let $f'_x(q) \in I_j$.

Case 1: $i = -1, j = 1$. Let $y_{up} = \max[f'_y(p^{-1}_{up}), f'_y(p^{1}_{up})]$ and $y_{down} = \max[f'_y(p^{-1}_{down}), f'_y(p^{1}_{down})]$.

If $y_{up} - y_{down} \leq c_2$ for c_2 larger than, say, $10c'$, then the statement follows. Otherwise, if $y_{up} - y_{down} > 10c'$, then for any $u \in \{p, q\}$, the set

$$([-1,1] \times \mathbb{R}) \cap_{i\{-1,1\}, d \in \{up, down\}} R(f'(p_d^i), D[p_d^i, u], c')$$

has constant diameter. Thus the statement again follows.

Case 2: By Lemma 2.4.3 we can assume that the points p_{up}^i, p_{down}^i, and p (as well as p_{up}^j, p_{down}^j, and q) have the same x coordinates. Then we apply Lemma 2.4.4.

It remains only to prove Lemma 2.4.4.

Proof:[of Lemma 2.4.4] Let $z_1 \in S_1$ and $z_2 \in S_2$ be any two points such that $\|z_1 - z_2\| = \max\{\|u - v\| : u \in S_1, v \in S_2\}$. Similarly, let $t_1 \in S_1$ and $t_2 \in S_2$ be any two points such that $\|t_1 - t_2\| = \min\{\|u - v\| : u \in S_1, v \in S_2\}$. Let y_p denote the y coordinate of point p. Without loss of generality, we can assume that $y_{z_1} < y_{z_2} < y_d$.

We claim that, if $y_{t_2} \leq y_{z_1}$, then $z_1 = t_1$. If $y_{z_1} < y_{t_1}$, then by decreasing y_{t_1}, we decrease $\|t_1 - t_2\|$. If $y_{z_1} > y_{t_1}$, then by decreasing y_{z_1}, we increase $\|z_1 - z_2\|$. Thus, $z_1 = t_1$, and in this case, $\|z_1 - z_2\| - \|t_1 - t_2\| \leq 2s$.

It remains to analyze the case that $y_{t_2} > y_{z_1}$. In this case, it is easy to see that, as long as $y_a < y_{z_1}$, we can increase y_a and decrease r_a such that t_2 and z_2 continue to belong to S_2. Therefore, without loss of generality, we can assume that $a = z_1$ and $r_a + s = \|z_1 - z_2\|$.

Similarly, we apply the same idea to d and t_1: we note that $y_{t_1} < y_{z_2}$ and, by decreasing y_d, we can assume that $d = z_2$ and $r_d + s = r_a + s = \|z_1 - z_2\|$. It is easy to see that, in this case (see Figure 2-6), we have $\|t_1 - t_2\| \geq r_a - 3s = \|z_1 - z_2\| - 3s$. We conclude that $\|z_1 - z_2\| - \|t_1 - t_2\| \leq 3s$. ■

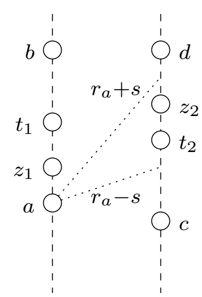


Figure 2-6: Proof illustration of Lemma 2.4.4.

Chapter 3

Multiplicative embeddings

In this chapter we present results on embedding using the more classical multiplicative notion of distortion. The results can be partitioned into two categories: results about unweighted shortest-path metrics on graphs, and results on the weighted version. The results on unweighted graphs are simpler, give better guarantees, and thus are more practical. The same algorithmic ideas can be extended with a lot more effort to the weighted problems.

3.1 Unweighted shortest path metrics into the line

Credits: The work in this section is a combined version of two earlier papers by Badoiu, Indyk, Rabinovich & Sidiropoulos, and by Dhamdhere, Gupta, Räcke & Ravi which obtained nearly identical results. The results have appeared in SODA'05.

In this section, we present several approximation algorithms for the problem of embedding metric spaces into a line, and into the two-dimensional plane. Among other results, we give an $O(\sqrt{n})$-approximation algorithm for the problem of finding a line embedding of a metric induced by a given unweighted graph, that minimizes the (standard) multiplicative distortion. We give an improved $\tilde{O}(n^{1/3})$ approximation for the case of metrics generated by unweighted trees. This is the first result of this type.

More formally, we present algorithms for the following fundamental embedding problem: given a graph $G = (V, E)$ inducing a shortest path metric $M = M(G) = (V, D)$, find a mapping f of V into a *line* that is non-contracting (i.e., $|f(u) - f(v)| \geq D(u, v)$ for all $u, v \in V$) which minimizes the distortion $c_{line}(M, f) = \max_{u,v \in V} \frac{|f(u)-f(v)|}{D(u,v)}$. That is, our goal is to find $c_{line}(M) = \min_f c_{line}(M, f)$. For the case when G is an *unweighted* graph, we show the following algorithms for this problem (denote $n = |V|$):

- A polynomial (in fact, $O(n^3c)$-time) c-approximation algorithm for metrics M for which $c_{line}(M) \leq c$. This also implies an $O(\sqrt{n})$-approximation algorithm for any M (Section 3.1.1).

- A polynomial-time $\tilde{O}(\sqrt{c})$ approximation algorithm for metrics generated by unweighted trees. This also implies an $\tilde{O}(n^{1/3})$-approximation algorithm for these metrics (Section 3.1.2).

- An exact algorithm, with running time $n^{O(c_{line}(M))}$ (Section 3.1.3).

We complement our algorithmic results by showing that a-approximating the value of $c_{line}(M)$ is NP-hard for certain $a > 1$ in Section 3.1.4. In particular, this justifies the exponential dependence on $c_{line}(M)$ in the running time bound for the exact algorithm.

Distortion vs Bandwidth. In the context of unweighted graphs, the notion of minimum distortion of an embedding into a line is closely related to the notion of a graph *bandwidth*. Specifically, if the non-contraction constraint $|f(u)-f(v)| \geq D(u, v)$ is replaced by a constraint $|f(u) - f(v)| \geq 1$ for $u \neq v$, then $c_1(M(G))$ becomes precisely the same as the bandwidth of the graph G.

There are several algorithms that approximate the bandwidth of a graph [Fei00, Gup00b]. Unfortunately, they do not seem applicable in our setting, since they do not enforce the non-contraction constraint for all node pairs. However, in the case of *exact* algorithms the situation is quite different. In particular, our exact algorithm for computing the distortion is based on the analogous algorithm for the bandwidth problem by Saxe [Sax80a].

3.1.1 A c-approximation algorithm

We start by stating an algorithmic version of a fact proved in [Mat90].

Lemma 3.1.1 *Any shortest path metric over an unweighted graph $G = (V, E)$ can be embedded into a line with distortion at most $2n - 1$ in time $O(|V| + |E|)$.*

Proof: Let T be a spanning tree of the graph. We replace every (undirected) edge of T with a pair of opposite directed edges. Since the resulting graph is Eulerian, we can consider an Euler tour C in T. Starting from an arbitrary node, we embed the nodes in T according to the order that they appear in C, ignoring multiple appearances of a node, and preserving the distances in C. Clearly, the resulting embedding is non-contracting, and since C has length $2n$, the distortion is at most $2n - 1$. ■

Note that the $O(n)$ bound is tight, e.g. when G is a star.

Let $G = (V, E)$ be a graph, such that there exists an embedding of G of distortion c. The algorithm for computing an embedding of distortion at most $O(c^2)$ is the following:

1. Let f_{OPT} be an optimal embedding of G (note that we just assume the existence of such an embedding, without computing it). Guess nodes $t_1, t_2 \in V$, such that $f_{OPT}(t_1) = \min_{v \in V} f_{OPT}(v)$, and $f_{OPT}(t_2) = \max_{v \in V} f_{OPT}(v)$.

2. Compute the shortest path $p = v_1, v_2, \ldots, v_L$ from t_1 to t_2.

3. Partition V into disjoint sets $V_1, V_2, \ldots V_L$, such that for each $u \in V_i$, $D(u, v_i) = \min_{1 \leq j \leq L} D(u, v_j)$. Break ties so that each V_i is connected.

4. For $i = 1 \ldots L$, compute a spanning tree T_i of the subgraph induced by V_i, rooted at v_i. Embed the nodes of V_i as in the proof of Lemma 3.1.1, leaving a space of length $|V_i|$ between the nodes of V_i and V_{i+1}.

Lemma 3.1.2 *For every $i, 1 \leq i \leq L$, and for every $x \in V_i$, we have $D(v_i, x) \leq c/2$.*

Proof: Assume that the assertion is not true. That is, there exists v_i, and $x \in V_i$, such that $D(x, v_i) > c/2$. Consider the optimal embedding f_{OPT}. By the fact that v_1

and v_L are the left-most and right-most embedded nodes in the embedding f_{OPT}, it follows that there exists $j, 1 \leq j < L$, such that $f_{OPT}(x)$ lies between $f_{OPT}(v_j)$, and $f_{OPT}(v_{j+1})$. W.l.o.g., assume that $f_{OPT}(v_j) < f_{OPT}(x) < f_{OPT}(v_{j+1})$. Since $x \in V_i$, we have $|f_{OPT}(v_{j+1}) - f_{OPT}(v_j)| = f_{OPT}(v_{j+1}) - f_{OPT}(x) + f_{OPT}(x) - f_{OPT}(v_j) \geq D(v_{j+1}, x) + D(x, v_j) \geq 2D(x, v_i) > c$. This is a contradiction, since the expansion of f_{OPT} is at most c. ■

Lemma 3.1.3 *For every i, $1 \leq i \leq L - c + 1$, we have $\sum_{j=i}^{i+c-1} |V_j| \leq 2c^2$.*

Proof: Assume that there exists i such that $\sum_{j=i}^{i+c-1} |V_j| > 2c^2$. Note that

$$\max_{i \leq j_1 < j_2 \leq i+c-1} |f_{OPT}(v_{j_1}) - f_{OPT}(v_{j_2})| \leq c(c-1).$$

Moreover, since $\sum_{j=i}^{i+c-1} |V_j| > 2c^2$, we have $\max_{u,w \in \bigcup_{j=i}^{i+c-1} V_j} |f_{OPT}(u) - f_{OPT}(w)| \geq 2c^2$. It follows that there exists $u \in V_l$, for some l, with $i \leq l \leq i + c - 1$, such that $|f_{OPT}(v_l) - f_{OPT}(u)| \geq \frac{2c^2 - c(c-1)}{2} > c^2/2$. Since the expansion is at most c, we have $D(v_l, u) > c/2$, contradicting Lemma 3.1.2. ■

Lemma 3.1.4 *The embedding computed by the algorithm is non-contracting.*

Proof: Let $x, y \in V$. If x and y are in the same set V_i, for some i, then clearly $|f(x) - f(y)| \geq D(x, y)$, since the distance between x and y produced by an traversal of the spanning tree of the graph induced by V_i is at least the distance of x and y on T_i, which is at least $D(x, y)$.

Assume now that $x \in V_i$ and $y \in V_j$, for some $i < j$. We have $|f(y) - f(x)| \geq |V_i| + 2\sum_{l=i+1}^{j-1} |V_l| + |V_j| \geq |V_i| + |V_j| + j - i > D(x, v_i) + D(y, v_j) + D(v_i, v_j) \geq D(x, y)$. ■

Lemma 3.1.5 *The distortion of the embedding computed by the algorithm is at most $4c^2$.*

Proof: It suffices to show that for each $\{x, y\} \in E$, $|f(x) - f(y)| \leq 4c^2$. Let $x \in V_i$, and $y \in V_j$. If $|i - j| \leq 2c$, then by Lemma 3.1.3 we obtain that $|f(x) - f(y)| \leq 4c^2$.

Assume now that there exist nodes $x \in V_i$ and $y \in V_j$, with $\{x, y\} \in E$, and $|i-j| > 2c$. By Lemma 3.1.2, we obtain that $D(v_i, x) \leq c/2$, and $D(y, v_j) \leq c/2$, and thus $|i-j| = D(v_i, v_j) \leq c+1$, a contradiction. ■

Theorem 3.1.6 *The described algorithm computes a non-contracting embedding of maximum distortion $O(c^2)$, in time $O(n^3 c)$.*

Proof: By Lemmata 3.1.4 and 3.1.5, it follows that the computed embedding is non-contracting and has distortion at most $O(c^2)$. In the beginning of the algorithm, we compute all-pairs shortest paths for the graph. Next, for each possible pair of nodes t_1 and t_2, the described embedding can be computed in linear time. Thus, the total running time is $O(n^2|E|) = O(n^3 c)$. ■

Theorem 3.1.7 *There exists a $O(\sqrt{n})$-approximation algorithm for the minimum distortion embedding problem.*

Proof: If the optimal distortion c is at most $\sqrt{n}$, then the described algorithm computes an embedding of distortion at most $O(c\sqrt{n})$. Otherwise, the algorithm described in Lemma 3.1.1, computes an embedding of distortion $O(n)$. Thus, by taking the best of the above two embeddings, we obtain an $O(\sqrt{n})$-approximation. ■

3.1.2 Better embeddings for unweighted trees

For the case of trees, we use a similar framework as for general graphs: we divide the tree along the path from t_1 to t_2 and obtain connected components $V_1, \ldots, V_L$ each with $\mathrm{diam}(V_i) \leq c$ and $\sum_{j=i}^{i+c-1} |V_j| \leq 2c^2$. Instead of a spanning tree on each V_i, we give a more sophisticated embedding. We consider all the vertices in $X_i = \cup_{j=i}^{i+c} V_j$ together. Lemma 3.1.2 gives the following bound on the diameter of the set X_i.

Lemma 3.1.8 *The diameter of the set X_j (for $j = 1, 2, \ldots$) is at most $2c$.*

We use the following straightforward lower bound on the distortion for embedding X_j.

The *local density* Δ of G is defined as

$$\Delta = \max_{v \in V, r \in \mathbb{R}_{>0}} \left\{ \frac{|B(v,r)| - 1}{2r} \right\} ,$$

where $|B(v,r)| = \{u \in V \mid d(u,v) \leq r\}$ denotes the ball of nodes within distance r from v. Intuitively, a high local density tells us that there are dense clusters in the graph, which will cause a large distortion. The following lemma formalizes this intuition.

Lemma 3.1.9 [Local Density] *Let G denote a graph with local density Δ. Then any map of G into the line has distortion at least Δ.*

Prefix Embeddings.

We first prove that it suffices to consider embeddings where each prefix of the associated tour forms a connected component of the tree; this will allow us to considerably simplify all our later arguments.

Lemma 3.1.10 [Prefix Embeddings] *Given any graph G, there exists an embedding of G into the real line with the following two properties:*

1. *Walk from left to right on the line, the set of points encountered up to a certain point forms a connected component of G.*

2. *The distortion of this map is at most twice the optimal distortion.*

Proof: Consider the optimal embedding f^*, and let $v_1, v_2, \ldots, v_n$ be the order of the points in this embedding. (We will blur the distinction between a vertex v and its image $f^*(v)$ on the line.) Without loss of generality, we can assume that the distance between any two adjacent points v_i and v_{i+1} in this embedding is their shortest path distance $D(v_i, v_{i+1})$.

Let i be the smallest index such that $\{v_1, v_2, \ldots, v_i\}$ does not form a connected subgraph; hence there exists some vertex on every v_{i-1}-v_i path that has not yet been laid out. We pick a shortest path P, take the vertex w in $P \setminus \{v_1, v_2, \ldots, v_{i-1}\}$ closest to v_{i-1}, and place it at distance $D(v_{i-1}, w)$ to the right of v_{i-1} in the embedding. We repeat this process until Property 1 is satisfied; it remains to bound the distortion we have introduced.

Note that the above process moves each vertex at most once, and only moves vertices to the left. We claim that each vertex is moved by at most distance c, where c is the optimal distortion. Indeed, consider a vertex w that is moved when addressing the v_{i-1}-v_i path, and let v_k be a neighbor of w among $v_1, \dots, v_{i-1}$. Note that the distance $|f^*(v_k) - f^*(w)|$ between these two vertices is at most c in the optimal embedding. Since w stays to the right of v_k, the distance by which w is moved is at most c.

In short, though the above alterations move vertices to the left, whilst keeping others at their original locations in f^*, the distance between the endpoints of an edge increases by at most c. Since the distance $|f^*(v) - f^*(u)|$ was at most c to begin with, we end up with an embedding with (multiplicative) distortion at most $2c$, proving the lemma. ■

Henceforth, we will only consider embeddings that satisfy the properties stated in Lemma 3.1.10. The bound on the increase in distortion is asymptotically best possible: for the case of the n-vertex star $K_{1,n-1}$, the optimal distortion is $\approx n/2$, but any prefix embedding has distortion at least $n - 2$.

The Embedding Algorithm.

In this section, we give an algorithm which embeds trees with distortion $g(c) = 2\Delta\sqrt{c \log c} + c$, where Δ is the local density and c the optimal distortion. The algorithm proceeds in rounds: in round i, we lay down a set Z_i with about $g(c)$ vertices. To ensure that the neighbors of vertices are not placed too far away from them, we enforce the condition that the vertices in Z_i include all the neighbors of vertices in $\cup_{j<i} Z_j$ that have not already been laid out.

It is this very tension between needing to lay out a lot of vertices and needing to ensure their neighbors can be laid out later on, that leads to the following algorithm. In fact, we will mentally separate the action of laying out the neighbors of previously embedded vertices (which we call the *BFS part* of the round) from that of laying out new vertices (which we call the *DFS part*).

We assume that we know the left-most vertex r in the prefix embedding; we can just run over all the possible values of r to handle this assumption. Let $N(X)$ denote the set of neighbors of vertices in a set $X \subseteq V$.

We define a *light path ordering* on the vertices of the tree T. The light path ordering is a DFS ordering which starts at root r and at each point enters the subtree with smallest number of vertices in it.

Algorithm Tree-Embed:

1. let $C \leftarrow \{r\}$ denote the set of vertices
   already visited. Set $i \leftarrow 1$.
2. while $C \neq V(T)$ do
   (Round i BFS)
3. Visit all vertices in $N(C) \setminus C$;
   let $C \leftarrow C \cup N(C)$
   (Round i DFS)
5. set B to be a set of $g(c)$ vertices
   of $V(T) \setminus C$ in the *light path ordering*.
   Visit all vertices in B; let $C \leftarrow C \cup B$.
6. endwhile

Lemma 3.1.11 [Number of rounds] *The algorithm Tree-Embed requires at most $\sqrt{c \log^{-1} c}$ iterations.*

Proof: By the very definition of the algorithm, the set C grows by at least $g(c)$ in every iteration. Note that the diameter of the tree is bounded by $2c$ and its local density is Δ. Therefore, the number of nodes in the tree is at most $2\Delta c$. Hence, within $(2\Delta c)/g(c) \leq \sqrt{c \log^{-1} c}$ iterations, all the vertices of the tree will be visited. ■

The heart of the proof is to show that visiting the vertices in Steps 3 and 5 does not incur too much distortion; it may be the case that the size of $N(C) \setminus C$ may be too large, or even that these vertices may be separated very far from each other.

Lemma 3.1.12 [Span of boundary] *The size of the induced spanning tree on the boundary $N(C) \setminus C$ is bounded by $g(c)$.*

Proof: Consider the set C_i of vertices that have been visited by round i. Consider a vertex x visited in round j of the DFS for some $j \leq i$. Note that the children of the vertex x will be visited *after* x. We say that x is a *branching point* if not all the children of x were visited in the same round as x. The branching point x is *active* after round i if at least one of the vertices below it has not been visited by round i; otherwise it is *inactive*. We claim that all the active branching points in C_i lie on some root-leaf path. This follows because the light path ordering is a DFS ordering. Therefore, if some vertices below a branching point x have not been visited, then the DFS part of the algorithm will not visit a different subtree.

Note that each active branching point (except possibly the lowest one) has at least two children and the algorithm visits the child which has a smaller number of vertices in its subtree. Recall that the size of the tree is bounded by $2c^2$ by Lemma 3.1.3. Therefore, the number of active branching points on a root to leaf path is at most $2 \log c + 1$.

We claim that every node in $N(C_i) \setminus C_i$ is within a distance of $i + 1$ of some active branching point. We prove this by induction on i. Before the first round, this property is true, since $C_0 = \{r\}$. Now assume the property for $i - 1$ and consider a vertex $v \in N(C_i) \setminus C_i$. Let u be the neighbor of v such that $u \in C_i$. If u was visited in the round i of the DFS, then u is an active branching point, since its child v has not been visited in the same round. Otherwise, if u was visited in round i of the BFS, then u is within distance i of some branching point x. Since v is below x and has not been visited after round i, the branching point x must be active. Therefore, v is within distance $i + 1$ from some active branching point.

Consider an active branching point x and let N_x contain the points from $N(C_i) \setminus C_i$

that are within distance $i + 1$ from x. Then, we can bound the span of the induced tree on N_x using the local density bound. The number of vertices in the induced tree on N_x is bounded by $(i + 1)\Delta$. Thus, for each active branching point, the number of vertices in the induced tree is bounded by $\Delta\sqrt{c \log^{-1} c}$. Since there are $2 \log c + 1$ branching points overall, the sum of spans over all the active branching points is at most $2\Delta\sqrt{c \log c}$. Note that, all the active branching points are on a single root-leaf path. Therefore, connecting all the branching points in $N(C_i) \setminus C_i$ requires only a path of length c. Hence, the total span of vertices in $N(C_i) \setminus C_i$ is bounded by $g(c)$.
■

Lemma 3.1.13 *The span of the tree induced on the vertices visited in any iteration is bounded by* $2g(c)$.

Proof: From Lemma 3.1.12, the span of the vertices visited in Step 3 of the algorithm is bounded by $g(c)$. The number of new vertices visited in Step 5 of the algorithm is bounded by $g(c)$. Since, we visit a set of connected components, their span is bounded by $g(c) + \text{span}(N(C) \setminus C)$. Therefore, the span of the vertices visited in each iteration is bounded by $2g(c)$. ■

Lemma 3.1.14 *The distortion of the embedding produced by Algorithm Tree-Embed is* $4g(c)$.

Proof: For a pair of vertices that are visited during the same iteration, the distance in the embedding is bounded by $2g(c)$ (from Lemma 3.1.13). Therefore, the distortion of such a pair is bounded by $4g(c)$. So, consider an edge (x, y) such that x and y were visited in different iterations. Note that, Step 1 of the algorithm ensures that if x is visited in iteration i, then y is visited in iteration $i + 1$. Therefore, the distance between x and y in the embedding is bounded by $4g(c)$. Hence, the distortion is bounded by $4g(c)$. ■

Concatenating the embeddings. In order to concatenate the embeddings of $X_1, X_2, \ldots$, it is enough to observe that since the input graph is a tree, there is only one edge connecting components X_i and X_{i+1} for all i. Consider the last vertex in X_i, viz.

v_{ic}. To produce an embedding of the component X_i using Algorithm Tree-Embed, we use a light path ordering of X_i assuming that the subtree containing v_{ic} is the heaviest subtree. Hence v_{ic} is last in the light path ordering of X_i and is visited in the last iteration of the Algorithm Tree-Embed. This makes sure that the distortion of the edge (v_{ic}, v_{ic+1}) is smaller than $2g(c)$. Changing the light path ordering in this way does not affect the bound on the distortion proved in Lemma 3.1.14. Thus we get the following result.

Theorem 3.1.15 *There is a polynomial time algorithm that finds an embedding of an unweighted tree with distortion* $8\Delta\sqrt{c \log c} + 4c$.

Corollary 3.1.16 *There is a polynomial time algorithm that finds an embedding of an unweighted tree with distortion within a factor* $O((n \log n)^{1/3})$ *of the optimal distortion.*

3.1.3 A dynamic programming algorithm for graphs of small distortion

Given a connected simple graph $G = (V, E)$ and an integer c, we consider the problem of deciding whether there exists a non-contracting embedding of G into the integer line with maximum distortion at most c.

Note that the maximum distance between any two points in an optimal embedding can be at most $c(n-1)$, and there always exists an optimal embedding with all the nodes embedded into integer coordinates. W.l.o.g., in the rest of this section, we will only consider embeddings of the form $f : V \to \{0, 1, \ldots, c(n-1)\}$. Furthermore, if G admits an embedding of distortion c, then the maximum degree of G is at most $2c$. Thus, we may also assume that G has maximum degree $2c$.

Definition 6 (Partial Embedding) *Let* $V' \subseteq V$. *A* partial embedding *on* V' *is a function* $g : V' \to \{0, 1, \ldots, c(n-1)\}$.

Definition 7 (Feasible Partial Embedding) *Let* f *be a partial embedding on* V'.

f is called feasible *if there exists an embedding g of distortion at most c, such that for each $v \in V'$, we have $g(v) = f(v)$, and for each $u \notin V'$, it is $g(u) > \max_{w \in V'} f(w)$.*

Definition 8 (Plausible Partial Embedding) *Let f be a partial embedding on V'. f is called* plausible *if*

- *For each $u, v \in V'$, we have $|f(u) - f(v)| \geq D(u, v)$.*
- *For each $u, v \in V'$, if $\{u, v\} \in E$, then $|f(u) - f(v)| \leq c$.*
- *Let $L = \max_{v \in V'} f(v)$. For each $u \in V'$, if $f(u) \leq L - c$, then for each $w \in V$ such that $\{u, w\} \in E$, we have $w \in V'$.*

Lemma 3.1.17 *If a partial embedding is feasible, then it is also plausible.*

Proof: Let f be a partial embedding over V', such that f is feasible, but not plausible, and let $L = \max_{v \in V'} f(v)$. It follows that there exists $\{u, w\} \in E$, with $u \in V'$, such that $f(u) \leq L - c$, and $w \notin V'$. Since f is feasible, there exists an embedding g of distortion at most c, satisfying $g(u) = f(u) \leq L - c$, and $g(w) > L$. Thus, $|g(u) - g(w)| > c$, a contradiction. ■

Definition 9 (Active Region) *Let f be a partial embedding over V'. The* active region *of f is a couple (X, Y), where $X = \{(u_1, f(u_1)), \ldots, (u_{|X|}, f(u_{|X|}))\}$ is a set of $\min\{2c + 1, |V'|\}$ couples, where $\{u_1, \ldots, u_{|X|}\}$ is a subset of V', such that $f(u_i) = \max_{u \in V' \setminus \{u_{i+1}, \ldots, u_{|X|}\}} f(u)$, and Y is the set of all edges in E having exactly one endpoint in V'.*

Lemma 3.1.18 *Let f_1 be a plausible partial embedding over V_1, and f_2 be a plausible partial embedding over V_2. If f_1 and f_2 have the same active region, then*

- *$V_1 = V_2$.*
- *f_1 is feasible if and only if f_2 is feasible.*

Proof: Let $L = \max_{v \in V'} f(v)$. To prove that $V_1 \subseteq V_2$, assume that there exists $v \in V_1 \setminus V_2$. Let p be a path starting at v, and terminating at some node in $V_1 \cap V_2$, and let v'' be the first node in $V_1 \cap V_2$ visited by p, and v' be the node visited exactly before v''. Clearly, $v' \in V_1 \setminus V_2$, and v' is not in the active region, thus $f_1(v') < L - 2c$. Furthermore, by the definition of a plausible partial embedding, since the edge $\{v'', v'\}$ has exactly one endpoint in V_2, it follows that $f_2(v'') > L - c$. Thus, $|f_1(v') - f_1(v'')| = |f_1(v') - f_2(v'')| > c$, contradicting the fact that f_1 is plausible. Similarly we can show that $V_2 \subseteq V_1$, and thus $V_1 = V_2$.

Assume now that f_1 is feasible, thus there exists an embedding g_1 of distortion at most s, such that for each $v \in V_1$, we have $f_1(v) = g_1(v)$, and for each $v \notin V_1$, we have $g_1(v) > L$. Consider the embedding g_2, where $g_2(u) = f_2(u)$, if $u \in V_2$, and $g_2(u) = g_1(u)$ otherwise. It suffices to show that g_2 is non-contracting and has distortion at most c.

If g_2 has distortion more than c, then since f_2 is a plausible partial embedding, and g_1 has distortion at most c, it follows that there exists an edge $\{u, w\}$, with $u \in V_2$ and $w \notin V_2$, such that $|g_2(u) - g_2(w)| > c$. Since the edge $\{u, w\}$ has exactly one endpoint in V_2, it follows that $f_2(u) > L - c$, and thus u is in the active region, and $f_2(u) = f_1(u)$. Thus, we obtain that $|g_1(u) - g_1(w)| = |g_2(u) - g_2(w)| > c$, a contradiction. Thus, g_2 has distortion at most c.

If g_2 is a contraction, then there exist nodes u and w such that $|g_2(u) - g_2(w)| < D(u, w)$. Since f_2 is plausible, and g_2 is non-contracting, we obtain that exactly one of the nodes u and w is in V_2. W.l.o.g., assume that $u \in V_2$ and $w \notin V_2$, and thus $f_2(u) > L - c$. Thus, u must be in the active region, and we obtain that $f_2(u) = f_1(u)$, and thus $|g_1(u) - g_1(w)| = |g_2(u) - g_2(w)| < D(u, w)$, a contradiction. We have shown that g_2 is non-contracting and has distortion at most c, thus f_2 is feasible. ■

Lemma 3.1.19 *For fixed values of c, the number of all possible active regions for all the plausible partial embeddings is at most $O(n^{4c+2})$.*

Proof: Let f be a plausible partial embedding, with active region (X, Y), such that $|X| = i$. It is easy to see that every edge in Y has exactly one endpoint in X. Since the degree of every node is at most $2c$, after fixing X, the number of possible values for Y is at most 2^{2ic}. Also, the number of possible different values for X is at most $\binom{n}{i}(nc)^i$. Thus, the number of possible active regions for all plausible partial embeddings is at most $\sum_{i=1}^{2c+1}\binom{n}{i}(nc)^i 2^{2ic} = O(n^{4c+2})$. ∎

Definition 10 (Successor of a Partial Embedding) *Let f_1 and f_2 be plausible partial embeddings on V_1 and V_2 respectively. f_2 is a* successor *of f_1 if and only if*

- *$V_2 = V_1 \cup \{u\}$, for some $u \notin V_1$.*
- *For each $u \in V_1 \cap V_2$, we have $f_1(u) = f_2(u)$.*
- *If $u \in V_2$ and $u \notin V_1$, then $f_2(u) = \max_{v \in V_2} f_2(v)$.*

Let P be the set of all plausible partial embeddings, and let $\hat{P}$ be the set of all active regions of the embeddings in P. Consider a directed graph H with $V(H) = \hat{P}$. For each $\hat{x}, \hat{y} \in V(H)$, $(\hat{x}, \hat{y}) \in E(H)$ if and only if there exist plausible embeddings x, y, such that $\hat{x}$ and $\hat{y}$ are the active regions of x and y respectively, and y is a successor of x.

Lemma 3.1.20 *Let x_0 be the active region of the empty partial embedding. G admits a non-contracting embedding of distortion at most c, if and only if there exists a directed path from x_0 to some node x in H, such that $x = (X, Y)$, with $X \neq \emptyset$ and $Y = \emptyset$.*

Proof: If there exists a path from x_0 to some node $x = (X, Y)$, with $X \neq \emptyset$ and $Y = \emptyset$, then since $X \neq \emptyset$, it follows that x is not the active region of the empty partial embedding. Furthermore, since G is connected and $Y = \emptyset$, it follows that x is the active region of a plausible embedding f of all the nodes of G. By the definition of a plausible embedding, it follows that f is a non-contracting embedding of G with distortion at most c.

If there exists a non-contracting embedding f of G, with distortion at most c, then we can construct a path in H, visiting nodes $y_0, y_1, \ldots, y_{|V|}$, as follows: For each i let f_i be the partial embedding obtained from f by considering only the i leftmost embedded nodes, and let y_i be the active region of f_i. Clearly, each f_i is a feasible embedding, and thus by Lemma 3.1.17, it is also plausible. Moreover, $y_0 = x_0$, and for each $0 < i \leq |V|$, it is easy to see that f_i is a successor of f_{i-1}, and thus $(y_{i-1}, y_i) \in E(H)$. Since, $f_{|V|}$ is an embedding of all the nodes of G, the active region $y_{|V|} = (X_{|V|}, Y_{|V|})$ satisfies $X_{|V|} \neq \emptyset$, and $Y_{|V|} = \emptyset$. ■

Using Lemma 3.1.20, we can decide whether there exists an embedding of G as follows: We begin at node x_0, and we repeatedly traverse edges of H, without repeating nodes. Note that we do not compute the whole H from the beginning, but we instead compute only the neighbors of the current node. This is done as follows: At each step i, we maintain a plausible partial embedding g_i, such that each partial embedding induced by the j leftmost embedded nodes in g_i, has active region equal to the jth node in the path from x_0 to the current node. We consider all the plausible embeddings obtained by adding a rightmost node in g_i. The key property is that by Lemma 3.1.18, the active regions of these embeddings are exactly the neighbors of the current node. This is because an active region completely determines the subset of embedded nodes, as well as the feasibility of such a plausible embedding. By Lemma 3.1.19, the above procedure runs in polynomial time when s is fixed.

Theorem 3.1.21 *For any fixed integer c, we can compute in polynomial time a non-contracting embedding of G, with distortion at most c, if one exists.*

3.1.4 Hardness of approximation

In this section we show that the problem of computing minimum distortion embedding of unweighted graphs is NP-hard to a-approximate for certain $a > 1$. This is done by a reduction from TSP over $(1, 2)$-metrics. Recall that the latter problem is NP-hard to approximate up to some constant $a > 1$.

Recall that a metric $M = (V, D)$ is a $(1, 2)$-metric, if for all $u, v \in V$, $u \neq v$, we have $D(u, v) \in \{1, 2\}$. Let $G(M)$ be a graph (V, E) where E contains all edges $\{u, v\}$ such that $D(u, v) = 1$.

The reduction F from the instances of TSP to the instances of the embedding problem is as follows. For a $(1, 2)$-metric M, we first compute $G = (V, E) = G(M)$. Then we construct a copy $G' = (V', E')$ of G, where V' is disjoint from V. Finally, we add a vertex o with an edge to all vertices in $V \cup V'$. In this way we obtain the graph $F(M)$.

The properties of the reduction are as follows.

Lemma 3.1.22 *If there is a tour in M of length t, then $F(M)$ can be embedded into a line with distortion at most t.*

Proof: The embedding $f : F(M) \to \Re$ is constructed as follows. Let $v_1, \ldots, v_n, v_1$ be the sequence of vertices visited by a tour T of length t. The embedding f is obtained by placing the vertices V in the order induced by T, followed by the vertex o and then the vertices V'. Formally:

- $f(v_1) = 0$, $f(v_i) = f(v_{i-1}) + D(v_{i-1}, v_i)$ for $i > 1$
- $f(o) = f(v_n) + 1$
- $f(v'_1) = f(o) + 1$, $f(v'_i) = f(v'_{i-1}) + D(v'_{i-1}, v'_i)$ for $i > 1$

It is immediate that f is non-contracting. In addition, the maximum distortion (of at most t) is achieved by the edges $\{o, v_1\}$ and $\{o, v'_n\}$. ■

Lemma 3.1.23 *If there is an embedding f of $F(M)$ into a line that has distortion s, then there is a tour in M of length at most $s + 1$.*

Proof: Let $H = F(M)$. Let $U = u_1 \ldots u_{2n}$ be the sequence of the vertices of $V \cup V'$ in the order induced by f. Partition the range $\{1 \ldots 2n\}$ into maximal intervals $\{i_0 \ldots i_1 - 1\}, \{i_1 \ldots i_2 - 1\}, \ldots, \{i_{k-1} \ldots i_k - 1\}$, such that for each interval I, the set $\{u_i : i \in I\}$ is either entirely contained in V, or entirely contained in V'. Recall that H has diameter 2. Since f has distortion s, it follows that $|f(u_1) - f(u_{2n})| \leq 2s$. Moreover, from non-contraction of f it follows that $|f(u_{i_j-1}) - f(u_{i_j})| = 2$ for all j. It follows that if we swap any two subsequences of U corresponding to different intervals I and I', then the resulting mapping of $V \cup V'$ into $\Re$ is still non-contracting (with respect to the metric induced by H). Therefore, there exists a mapping f' of $V \cup V'$ into $\Re$ which is non-contracting, in which all vertices of V precede all vertices of V', and such that the diameter of the set $f'(V \cup V')$ is at most $2s$. Without loss of generality, assume that the diameter Δ of $f'(v)$ is not greater than the diameter of $f'(V')$. This implies that $\Delta \leq (2s-2)/2 = s-1$. Therefore, the ordering of the vertices in V induced by f' corresponds to a tour in M of length at most $\Delta+2 \leq s+1$. ■

Corollary 3.1.24 *There exists a constant $a > 1$ such that a-approximating the minimum distortion embedding of an unweighted graph is NP-hard.*

3.2 Embedding into the line when the distortion is small

Credits: The results in this section is work done with Piotr Indyk and Yuri Rabinovich in the autumn of 2002. The results haven't been published yet.

For the case when G is a weighted graph and we want to embed it into the line, we obtain the following result. For induced metrics M such that $c_{line}(M) = 1 + \epsilon < 1.5$, we give an algorithm that finds a line embedding f such that $c_{line}(M, f) = 1 + O(\epsilon)$. In other words, the algorithm constructs a good embedding for metrics that are very well embeddable into a line. The algorithm proceeds by computing an MST T of M, and then ordering the nodes according to T. Thus, its running time is $O(n^2)$ in the worst case, and it is even more efficient for metric spaces that support faster MST computation. We also note that ordering the metric nodes using MST is a popular heuristic (e.g., see [BJDG+03]). To our knowledge, our result provide the first known provable guarantee for this heuristic.

The algorithm proceeds as follows: we start with every node of G being in its own component, keeping for each component an embedding of the points of the component into the line. We traverse the edges of G in increasing order of the distances D. If the endpoints of the edge $e = \{v, u\}$ are in different components, we merge these two components in the following way. Let $\Delta(C)$ be the diameter of a component C,

$$\Delta(C) = \max_{a,b \in C} D(a,b).$$

If $D(e) < \Delta(C)/(\epsilon(1+\epsilon))$, we can determine the order of the points of $C = A \bigcup B$ in an optimal embedding. If the distance $D(e) \geq \Delta(C)/(\epsilon(1+\epsilon))$, we choose any order.

3.2.1 A Special Case

We first show how to prove the correctness for the special case when we have no contractions, i.e., we have $D(e) < \Delta(C)/(\epsilon(1+\epsilon))$ whenever we merge 2 components. In this case we show how to get an $(1+\epsilon)$-approximation.

Claim 3.2.1 *Let $f : X \to \Re$ be an optimal embedding. At any step of the algorithm, for every 2 nodes $a, b \in C$ such that $f(a) > f(b)$ and any node $p \notin C$, either $f(p) > f(a)$ or $f(b) > f(p)$.*

Proof: Suppose we have $f(a) > f(p) > f(b)$. The nodes of C are linked by an MST. Let $\{b', a'\}$ be an edge of the MST such that $f(a') > f(p) > f(b')$. (such an edge exists because a, b exist) Then we have either $f(p) \geq \frac{f(a')+f(b')}{2}$ or $f(p) < \frac{f(a')+f(b')}{2}$. Without loss of generality, we have $f(p) \geq \frac{f(a')+f(b')}{2}$. Then, it must be the case that $D(a', b') > D(a', p)$, and it follows that p must have been added to the MST of C, therefore $p \in C$, which is a contradiction. ■

Claim 3.2.2 *At any step of the algorithm for any component, there is a unique ordering of the $(1+\epsilon)$-embeddings into the line, $f : X \to \Re$, not considering the reverse orderings.*

Proof: We prove the statement by induction. The claim is trivially verified for the base case, when the component contains only 1 node. When we merge 2 components, A and B, because of claim 3.2.1 and the inductive hypothesis, we have only 4 possibilities: we first place the embedding of A or the reverse of it and then we place the embedding of B or the reverse of it. Let e be the smallest edge between the 2 components. Since $D(e) < \Delta(A)/(\epsilon(1+\epsilon))$ and $D(e) < \Delta(B)/(\epsilon(1+\epsilon))$, we cannot have the case that both placing the embedding and the reverse are feasible solutions. Therefore, out of the 4 possibilities, at least 3 possibilities do not give an $(1+\epsilon)$-embedding. Since there exists an $(1+\epsilon)$-embedding, one of the 4 possibilities must give an $(1+\epsilon)$-embedding. Thus, this embedding is unique for the component $A \bigcup B$. ■

We can compute the ordering of the embedding of $A \bigcup B$, by just looking at the distances D between the extreme nodes of the embeddings of A and B. Given the embeddings of the 2 components, $f(A)$ and $f(B)$ we compute $f(A \bigcup B)$ by using the right ordering of the nodes, and such that the distance between the closest 2 points $a \in A, b \in B$, is exactly $D(a, b)$.

Claim 3.2.3 *f is an $(1+\epsilon)$-embedding of G.*

Proof: The ordering of the points is exactly the same as in an optimal solution, by the previous claim. The distance between every 2 consecutive nodes a, b is exactly $D(a, b)$, so f is contracted as much as possible. Thus, f cannot expand more than $1+\epsilon$. It remains to show that f is non-contracting. We show this by induction. In the beginning of the algorithm the map of each component is trivially non-contracting. Given 2 points a, b, consider the step of the algorithm when the 2 components A and B are merged. $(a \in A, b \in B)$ Let $v \in A, u \in B$ the closest 2 points in f. (v and u are extreme nodes in $f(A)$ and $f(B)$ respectively) By the triangle inequality and the inductive hypothesis, $D(a, b) \leq D(a, v) + D(v, u) + D(u, b) \leq f(a, v) + f(v, u) + f(u, b) = f(a, b)$. ■

The proof of the general case is based on the following structural theorem.

Theorem 3.2.4 *Let (X, D) be a metric $(1+\epsilon)$-embeddable into a line for $\epsilon < 1/2$. Let $G = (X, X \times X)$ be a complete graph with distances D. Then the shortest paths metric on the minimum spanning tree of G c-approximates (X, D), for $c = 2\sum_{i=1}^{\infty} \epsilon^i = O(\epsilon)$.*

Proof: Let M be the set of edges on a MST of G. Let $a_1, a_2, \ldots, a_k$ be a path on the MST. Let $f : X \to \Re$ be a non-contracting $(1+\epsilon)$-embedding of (X, D) into a line. Since the map f does not expand too much we have

$$\sum_{i=1}^{k-1}(1+\epsilon)D(a_i, a_{i+1}) \geq \sum_{i=1}^{k-1}|f(a_i)-f(a_{i+1})| \geq |f(a_1)-f(a_k)|. \tag{3.1}$$

Without loss of generality we can assume $f(a_1) < f(a_k)$.

Lemma 3.2.5 *If $f(a_1) < f(a_i) < f(a_k)$ for $i = 2, 3, \ldots, k-1$, then $\sum_{i=1}^{k-1} D(a_i, a_{i+1}) \leq |f(a_1) - f(a_k)|$.*

Proof: If $f(a_i) < f(a_{i+1})$ for all $1 \leq i < k$, then $\sum_{i=1}^{k-1} D(a_i, a_{i+1}) \leq \sum_{i=1}^{k-1} |f(a_i) - f(a_{i+1})| = |f(a_1) - f(a_k)|$. Otherwise there exists i, such that $1 < i < k$, and such that $f(a_i) > f(a_{i+1})$. It follows that there exist l, j such that $l < j$, and such that $f(a_l) < f(a_j) < f(a_{l+1}) < f(a_{j+1})$. Since $\{a_l, a_{l+1}\}$ is an edge in the MST and $\{a_l, a_j\}$ is not, we have $D(a_l, a_{l+1}) < D(a_l, a_j)$. By the same argument we have $D(a_j, a_{j+1}) < D(a_{l+1}, a_{j+1})$. We construct a new path P' by removing $\{a_l, a_l + 1\}$ and $\{a_j, a_{j+1}\}$ and adding $\{a_l, a_j\}$ and $\{a_{l+1}, a_{j+1}\}$. We also set $D(a_l, a_j) = D(a_l, a_{l+1})$ and $D(a_{l+1}, a_{j+1}) = D(a_j, a_{j+1})$. Note that we decrease the distances of these edges, which is OK. Since the degree of each node is 2, we will have a path from a_1 to a_k and the sum of the distances of the path will remain the same. We repeat the argument, as long as there exists i such that $f(a_i) > f(a_{i+1})$. Every such step performed decreases the number of edges that overlap in the embedding. Thus, after a finite number of these steps, we will end up with a path. Therefore, we have

$$\sum_{i=1}^{k-1} D'(a_i, a_{i+1}) = \sum_{i=1}^{k-1} D(a'_i, a'_{i+1}) \leq \sum_{i=1}^{k-1} D'(a'_i, a'_{i+1}) \leq \sum_{i=1}^{k-1} |f(a'_i) - f(a'_{i+1})| = |f(a_1) - f(a_k)|, \tag{3.2}$$

where $a'_1, a'_2, \ldots, a'_k$ are the points on the path we end up with, and D' is the original distance function. ■

We call applying lemma 3.2.5 "linearizing" a path. We will now proceed to prove the case when a_1 and a_k are not necessarily at the extremes of f. For the path $a_1, a_2, \ldots, a_k$, let s, t such that $f(a_s) < f(a_l) < f(a_t)$ for all $l \in \{1, \ldots, k\} -$

$\{s, t\}$. We linearize the path $a_s, a_{s+1}, \ldots, a_t$ by applying lemma 3.2.5. Let a_q be the rightmost point on the path $a_1, \ldots, a_i$. We linearize the path $a_q, \ldots, a_i$ and find the leftmost point p on the path $a_1, \ldots a_q$. We linearize $a_p, \ldots, a_q$ and recursively apply the previous argument to $a_1, \ldots, a_q$.

Claim 3.2.6 *Consider two consecutive sub-paths: $a_1, \ldots, a_i, \ldots, a_j$, such that $f(a_l) < f(a_{l+1})$ for $l \in \{1, \ldots, i-1\}$, and such that $f(a_l) > f(a_{l+1})$ for $l \in \{i, \ldots, j-1\}$. Then $f(a_{i+1}) < f(a_1)$.*

Proof: Assume $f(a_{i+1}) \geq f(a_1)$. Then there exists $l < i$, such that $f(a_l) < f(a_i) < f(a_{l+1})$. If $f(a_i) - f(a_l) \leq (f(a_{l+1}) - f(a_i))/2$ then it must be the case that $D(a_l, a_i) < D(a_l, a_{l+1})$ which implies $\{a_l, a_i\}$ must belong to the MST which is a contradiction. Respectively, if $f(a_{l+1}) - f(a_i) \leq (f(a_{l+1}) - f(a_i))/2$ implies $\{a_i, a_{l+1}\}$ is in the MST, contradiction. ∎

Applying claim 3.2.6 to the first two linearized paths $a_1, \ldots a_i, \ldots, a_j$, such that $f(a_l) < f(a_{l+1})$ for $l \in \{1, \ldots, i-1\}$, and such that $f(a_l) > f(a_{l+1})$ for $l \in \{i, \ldots, j-1\}$, we get that

$$\sum_{l=1}^{i-1} D(a_l, a_{l+1}) \leq |f(a_i) - f(a_1)| \leq \epsilon D(a_i, a_{i+1})$$

by using the fact that $D(a_i, a_{i+1}) < D(a_1, a_{i+1})$. By charging the cost of the small linearized paths to the bigger ones, i.e., using the previous argument for each linearized path and writing the length of the small path as ϵ times the size of the bigger path, we get the following

$$\sum_{i=1}^{k} D(a_i, a_{i+1}) \leq (\sum_{i=s}^{t} D(a_i, a_{i+1}))(1 + 2\sum_{i=1}^{\infty} \epsilon^i) \tag{3.3}$$

If $t = s + 1$ (the $a_s, \ldots, a_t$ path has only one edge) then $D(a_1, a_k) \geq D(a_s, a_t)$ since $\{a_s, a_t\}$ is not on the MST. By using 3.3 we get

$$\sum_{i=1}^{k} D(a_i, a_{i+1}) \leq D(a_1, a_k)(1 + 2\sum_{i=1}^{\infty} \epsilon^i) \leq f(a_1, a_k)(1 + 2\sum_{i=1}^{\infty} \epsilon^i)$$

If $t > s + 1$, then $D(a_1, a_{s+1}) \geq D(a_s, a_{s+1})$ and $D(a_k, a_{t-1}) \geq D(a_{t-1}, a_t)$. By using 3.3 we get

$$\sum_{i=1}^{k} D(a_i, a_{i+1}) \leq (D(a_1, a_{s+1}) + \sum_{i=s+1}^{t-2} D(a_i, a_{i+1}) + D(a_1, a_{s+1}))(1+2\sum_{i=1}^{\infty} \epsilon^i) \leq f(a_1, a_k)(1+2\sum_{i=1}^{\infty} \epsilon^i)$$

■

3.2.2 The general case of the algorithm

In this section we solve the case when we relax the condition of the special case. Consider the step of the algorithm when we merge 2 components, A, B and we have $D(e) \geq \Delta(A)/(\epsilon(1 + \epsilon))$. In this case we arbitrarily choose to place the embedding of A or the reverse.

Claim 3.2.7 *f is an $(1 + O(\epsilon))$-distortion embedding of G.*

Proof: The non-contracting part is exactly as the non-contracting proof of the claim 3.2.3. It remains to compute how much f can expand. The distance between every 2 consecutive nodes a, b is exactly $D(a, b)$, so f is contracting as much as possible. However, the ordering that we have computed might not be the same as in an optimal solution.

We show there exists a graph $G' = (V, E, D')$, such that for every 2 components A, B that are merged, the edge between the closest 2 points in the embedding f is part of the MST of G'. The distances d' of G' have the following property: $(1 + 2\epsilon + 2\epsilon^2 + \epsilon^3)D(v, w) \geq D'(v, w) \geq D(v, w)$. Our algorithm gives the same output on G' as on G. The edges of the MST of G' are non-expanding. Using these edges we can upper-bound $|f(v) - f(w)|$ by $(1 + O(\epsilon))D(v, w)$.

When we merge A and B, let $e = \{v, u\}$ be the smallest distance edge between A and B. Let $a \in A, b \in B$ such that

$$|f(a) - f(b)| = \min_{a \in A, b \in B} |f(a) - f(b)|.$$

If $D(v,u) < \Delta(A)/(\epsilon(1+\epsilon))$ and $D(v,u) < \Delta(B)/(\epsilon(1+\epsilon))$ then $D(a,b) < (1+\epsilon)D(v,u)$. If $D(v,u) \geq \Delta(A)/(\epsilon(1+\epsilon))$ and $D(v,u) < \Delta(B)/(\epsilon(1+\epsilon))$ then $\Delta(A) < D(v,u)\epsilon(1+\epsilon)$. By triangle inequality $D(a,b) < D(v,u)(1+\epsilon) + \Delta(A)(1+\epsilon) = (1+\epsilon)(D(v,u)+\Delta(A)) < (1+\epsilon)(D(v,u)(1+\epsilon(1+\epsilon))) = (1+2\epsilon+2\epsilon^2+\epsilon^3)D(v,u)$. The other 2 cases are similar to these ones. Next set the distances $D'(p,r) = \max(D(p,r), D(a,b)+\delta)$ for every $p \in A$ and $r \in B$, except for $D(a,b)$, for infinitesimally small $\delta > 0$ such that $D(p,r) > D(a,b)$. We set $D'(a,b) = D(a,b)$. It follows that the new distances are bigger by at most a multiplicative factor of $(1+2\epsilon+2\epsilon^2+\epsilon^3)$. We do this for every component A and B which have been merged. By theorem 3.2.4 we have that the MST approximates the metric within $1+2\sum_{i=1}^{\infty}\epsilon^i$. Let $\sigma(v,w) = \{(a,b) | (a,b)$ is on the MST path from v to $w\}$. For every pair of nodes $\{v,w\}$, using theorem 3.2.4,

$$\begin{aligned}
D'(v,w) &\geq \sum_{(p,r)\in\sigma(v,w)} D'(p,r)/(1+2\sum_{i=1}^{\infty}\epsilon^i) \\
&= \sum_{(p,r)\in\sigma(v,w)} |f(p)-f(r)|/(1+2\sum_{i=1}^{\infty}\epsilon^i) \\
&\geq |f(v)-f(w)|/(1+2\sum_{i=1}^{\infty}\epsilon^i).
\end{aligned}$$

We have $D(v,w) \geq D'(v,w)/(1+2\epsilon+2\epsilon^2+\epsilon^3)$. Therefore, $D(v,w) \geq |f(v) - f(w)|/((1+2\sum_{i=1}^{\infty}\epsilon^i)(1+2\epsilon+2\epsilon^2+\epsilon^3)) = |f(v)-f(w)|/(1+O(\epsilon))$. Therefore, f doesn't expand more than $1+O(\epsilon)$.

■

Theorem 3.2.8 *Let (X, D) be a metric $(1+\epsilon)$-embeddable into a line for $\epsilon < 1/2$. Then a map $f : X \rightarrow \Re$ can be computed in polynomial time such that f is an*

$(1 + O(\epsilon))$*-embedding of* (X, D) *into a line.*

3.3 Embedding spheres into the plane

Credits: The work in this section is a combined version of two earlier papers by Badoiu, Indyk, Rabinovich & Sidiropoulos, and by Dhamdhere, Gupta, Räcke & Ravi which obtained nearly identical results. The results have appeared in SODA'05.

In this section we study the problem of embedding metrics into the *plane*. In particular, we focus on embedding metrics $M = (X, D)$ which are induced by a set of points on a unit sphere S^2. Embedding such metrics is important, e.g., for the purpose of visualizing point-sets representing places on Earth or other planets, on a (planar) computer screen.[1] In general, we show that an n-point spherical metric can be embedded with distortion $O(\sqrt{n})$, and this bound is optimal in the worst case. (The lower bound is shown by resorting to the Borsuk-Ulam theorem [Bor33], which roughly states that any continuous mapping from S^2 into the plane maps two antipodes of S^2 to the same point.) For the algorithmic problem of embedding M into the plane, we give a 3.512-approximation algorithm, when D is the Euclidean distance in $\mathbb{R}^3$. For the case where D corresponds to the geodesic distance in S^2, our algorithm can be re-analyzed to give an approximation guarantee of 3.

To our knowledge, our results provide the first non-trivial approximation guarantees for the standard (multiplicative) notion of distortion for embeddings into low-dimensional spaces.

Let $M = (X, D)$ be a metric induced by a set X of n points on a unit sphere S^2, under the Euclidean distance in $\mathbb{R}^3$. Let $c_p^d(M)$ denote the minimum distortion of any embedding of M into l_p^d.

Theorem 3.3.1 *If* $M = (X, D)$ *is the metric induced by a set* X *of* n *points on a unit sphere* S^2*, under the Euclidean distance in* $\mathbb{R}^3$*, then* $c_2^2(M) = O(\sqrt{n})$.

[1] Indeed, the whole field of cartography is devoted to low-distortion representations of spherical maps in the plane.

Proof: Since the size of the surface of S^2 is constant, it follows that there exists a cap K in S^2, of size $\Omega(1/n)$, such that $X \cap K = \emptyset$. Let p_0 be the center of K on S^2, and p'_0 be its antipode. By rotating S^2, we may assume that $p_0 = (0,0,1)$, and thus $p'_0 = (0,0,-1)$.

For points $p, p' \in S^2$, let $\rho_S(p,p')$ be the geodesic distance between p and p' in S^2. Consider the mapping $f : X \to \mathbb{R}^2$, such that for every point $p \in X$, with $p = (x,y,z)$, we have $f(p) = \left(\rho_S(p,p'_0)\frac{x}{\sqrt{x^2+y^2}}, \rho_S(p,p'_0)\frac{y}{\sqrt{x^2+y^2}}\right)$, if $p \neq p'$, and $f(p) = (0,0)$, if $p = p'$. It is straightforward to verify that f is non-contracting.

Claim 3.3.2 *The expansion of f is maximized for points p, q, on the perimeter of K, which are antipodals with respect to K.*

Proof: Let $p, q \in S^2$. W.l.o.g., we assume that $p = (0, \sin\varphi_p, 1 + \cos\varphi_p)$, and $q = (\sin\varphi_q \sin\theta_q, \sin\varphi_q \cos\theta_q, 1 + \cos\varphi_q)$, for some $0 \leq \varphi_p, \varphi_q \leq \varphi$, and $0 \leq \theta_q \leq \pi$. The images of p and q are $f(p) = (0, \varphi_p)$, and $f(q) = (\varphi_q \sin\theta_q, \varphi_q \cos\theta_q)$, respectively. Let $h = \frac{\|f(p)-f(q)\|}{\|p-q\|}$, be the expansion of f in the pair p, q. We obtain:

$$h^2 = \frac{\varphi_q^2 + \varphi_p^2 - 2\varphi_q\varphi_p\cos\theta_q}{2 - 2\cos\varphi_p\cos\varphi_q - 2\sin\varphi_p\sin\varphi_q\cos\theta_q}$$

Observe that since $\sin\varphi_p \leq \varphi_p$, and $\sin\varphi_q \leq \varphi_q$, it follows that h^2 is maximized when $\cos\theta_q$ is minimized. That is, the expansion is maximized for $\theta_q = \pi$.

Thus, we can assume that the expansion of f is maximized for points $p, q \in S^2$, with $p = (0, \sin\varphi_p, 1 + \cos\varphi_p)$, and $q = (0, -\sin\varphi_q, 1 + \cos\varphi_q)$. For such points, the expansion is $\frac{\varphi_p + \varphi_q}{2\sin\frac{\varphi_p+\varphi_q}{2}}$. It follows that the expansion is maximized when $\varphi_p + \varphi_q$ is maximized, which happens when p and q are on the perimeter of K. ■

We pick p and q on the perimeter of K, such that p is the antipode of q w.r.to K. Let φ_K be the angle of K, and set $r_K = \varphi_K/2$. We have $r_K = \Omega(1/\sqrt{n})$, and $\|f(p) - f(q)\| = 2\pi - 2r_K$, while $\|p - q\| = 2\sin r_K$. Thus, the expansion is at most $\frac{\pi - r_K}{\sin r_K}$. W.l.o.g., we can assume that $r_K \leq \pi/2$, since otherwise we can simply consider a smaller cap K. Thus, $\frac{\pi - r_K}{\sin r_K} \leq 2\frac{\pi - r_K}{\pi r_K} < \frac{2}{r_K} = O(\sqrt{n})$. Since the embedding is non-contracting, it follows that the expansion is $O(\sqrt{n})$. ■

Theorem 3.3.3 *There exists a metric $M = (X, D)$, induced by a set X of n points on a unit sphere S^2, under the Euclidean distance in $\mathbb{R}^3$, such that any mapping $f : X \to \mathbb{R}^2$ has distortion $\Omega(\sqrt{n})$.*

Proof: Let $X \subset S^2$ be a set of n points, such that X is a $O(1/\sqrt{n})$-net of S^2, and let $f : X \to \mathbb{R}^2$ be a non-expanding embedding. Since $S^2 \subset \mathbb{R}^3$, by Kirszbraun's Theorem ([Kir34], see also [LN04a]), we obtain that f can be extended to a non-expanding mapping $f' : S^2 \to \mathbb{R}^2$. Also, by the Borsuk-Ulam Theorem, it follows that there exist antipodals $p, q \in S^2$, such that $f'(p) = f'(q)$. Since X is an $O(1/\sqrt{n})$-net, there exist points $p', q' \in X$, such that $\|p - p'\| = O(1/\sqrt{n})$, and $\|q - q'\| = O(1/\sqrt{n})$. Since f is non-expanding, it follows that $\|f(p') - f(q')\| = O(1/\sqrt{n})$. On the other hand, we have $\|p - q\| = 2$, and thus $\|p' - q'\| = \Omega(1)$. Thus, f has distortion $\Omega(\sqrt{n})$.
■

Theorem 3.3.4 *There exists a polynomial-time,* 3.512*-approximation algorithm, for the problem of embedding a finite sub-metric of S^2 into $\mathbb{R}^2$.*

Proof: We apply the embedding of Theorem 3.3.1, by choosing K to be the largest empty cap in S^2. Let r_K be the radius of K. By using an analysis similar to the one of Theorem 3.3.1, we obtain that the distortion of the embedding is at most $\frac{\pi - r_K}{\sin_{r_K}}$. Moreover, by using the analysis of Theorem 3.3.3, we can show that the distortion of an optimal embedding is at least $\max\{1, \frac{\cos r_K}{2\sin\frac{r_K}{2}}\}$. By simple calculations, we obtain that the distortion is maximized for $r_K = 2\tan^{-1}\frac{(\sqrt{3}-1)3^{3/4}\sqrt{2}}{6} \approx 0.749$, for which we obtain that the approximation ratio is less than 3.512. ■

For the case where the metric $M = (X, D)$ corresponds to the geodesic distances between the points of the sphere, we can show using the same techniques that the algorithm of Theorem 3.3.4, is in fact a 3-approximation.

3.4 Weighted shortest path metrics into the line

Credits: The results in this section is work done with Julia Chuzhoy, Piotr Indyk, and Anastasios Sidiropoulos, and has appeared in STOC'05.

From	Into	Distortion	Comments
general metrics	line	$O(\Delta^{4/5}c^{13/5})$	
weighted trees	line	$c^{O(1)}$	
weighted trees	line	$\Omega(n^{1/12}c)$	Hard to $O(n^{1/12})$-approximate even for $\Delta = n^{O(1)}$

Figure 3-1: Our results.

3.4.1 Introduction

In this section, we consider the problem of embedding metrics induced by *weighted* graphs into the line. The known algorithms were designed for *unweighted graphs* and thus provide only very weak guarantees for the problem. Specifically, assume that the minimum interpoint distance between the points is 1 and the maximum distance[2] is Δ. Then, by scaling, one can obtain algorithms for weighted graphs, with approximation factor multiplied by Δ.

Our results are presented in Figure 3-1. The first result is an algorithm that, given a general metric c-embeddable into the line, constructs an embedding with distortion $O(\Delta^{4/5}c^{13/5})$. The algorithm uses a novel method for traversing a weighted graph. It also uses a modification of the unweighted-graph algorithm from [BDG+05] as a subroutine, with a more general analysis.

Then, we consider the problem of embedding weighted tree metrics into the line. In this case we are able to get rid of the dependence on Δ from the approximation factor. Specifically, our algorithm produces an embedding with distortion $c^{O(1)}$.

We complement our upper bounds by a lower bound, which shows that the problem is hard to approximate up to a factor $a = \Omega(n^{1/12})$. This dramatically improves over the earlier result of [BDG+05], which only showed that the problem is hard for some constant $a > 1$ (note however that their result applies to unweighted graph metrics as well). Since the instances used to show our hardness result have spread $\Delta \leq n^{O(1)}$, it follows that approximating the distortion up to a factor of $\Delta^{\Omega(1)}$ is hard as well. In fact, the instances used to show hardness are metrics induced by (weighted) *trees*;

[2] We call the maximum/minimum interpoint distance ratio the *spread* of the metric.

thus the problem is hard for tree metrics as well. Our hardness proof is inspired by the ideas of Unger [Ung98].

3.4.2 Preliminaries

Consider an embedding of a set of vertices V into the line. We say that $U \subset V$ is embedded *continuously*, if there are no vertices $x, x' \in U$, and $y \in V - U$, such that $f(x) < f(y) < f(x')$.

We say that vertex set U is embedded *inside* vertex set U' iff the smallest interval containing the embedding of U also contains the embedding of U'. In particular, we say that vertex v is embedded inside edge $e = (x, y)$ for $v \neq x, v \neq y$, if either $f(x) < f(v) < f(y)$ or $f(y) < f(v) < f(x)$ hold.

Let $M = (X, D)$ be a metric, and $f : X \to \mathbb{R}$ be a non-contracting embedding of M into the line. Then, the *length* of f is $\max_{u \in X} f(u) - \min_{v \in X} f(v)$.

3.4.3 General metrics

In this section we will present a polynomial-time algorithm that given a metric $M = (X, D)$ of spread Δ that c-embeds into the line, computes an embedding of M into the line, with distortion $O(c^{11/4}\Delta^{3/4})$. Since it is known [Mat90] that any n-point metric embeds into the line with distortion $O(n)$, we can assume that $\Delta = O(n^{4/3})$.

We view metric M as a complete graph G defined on vertex set X, where the weight of each edge $e = \{u, v\}$ is $D(u, v)$. As a first step, our algorithm partitions the point set X into sub-sets $X_1, \ldots, X_\ell$, as follows. Let W be a large integer to be specified later. Remove all the edges of weight greater than W from G, and denote the resulting connected components by $C_1, \ldots, C_\ell$. Then for each $i : 1 \leq i \leq \ell$, X_i is the set of vertices of C_i. Let G_i be the subgraph of G induced by X_i. Our algorithm computes a low-distortion embedding for each G_i separately, and then concatenates the embeddings to obtain the final embedding of M. In order for the concatenation to have small distortion, we need the length of the embedding of each component to be sufficiently small (relatively to W). The following simple lemma, essentially shown in [Mat90], gives an embedding that will be used as a subroutine.

Lemma 3.4.1 *Let $M = (X, D)$ be a metric with minimum distance 1, and let T be a spanning tree of M. Then we can compute in polynomial time an embedding of M into the line, with distortion $O(cost(T))$, and length $O(cost(T))$.*

The embedding in the lemma is computed by taking an (pre-order) walk of the tree T. Since each edge is traversed only a constant number of times, the total length and distortion of the embedding follows.

Our algorithm proceeds as follows. For each $i : 1 \leq i \leq \ell$, we compute a spanning tree T_i of G_i, that has the following properties: the cost of T_i is low, and there exists a walk on T_i that gives a small distortion embedding of G_i. We can then view the concatenation of the embeddings of the components as if it is obtained by a walk on a spanning tree T of G. We show that the cost of T is small, and thus the total length of the embedding of G is also small. Since the minimum distance between components is large, the inter-component distortion is small.

Embedding the Components

In this section we concentrate on some component G_i, and we show how to embed it into a line.

Let H be the graph on vertex set X_i, obtained by removing all the edges of length at least W from G_i, and let H' be the graph obtained by removing all the edges of length at least cW from G_i. For any pair of vertices $x, y \in X_i$, let $D_H(x, y)$ and $D_{H'}(x, y)$ be the shortest-path distances between x and y in H and H', respectively. Recall that by the definition of X_i, H is a connected graph, and observe that $D_H(x, y) \geq D_{H'}(x, y) \geq D(x, y)$.

Lemma 3.4.2 *For any $x, y \in X_i$, $D_{H'}(x, y) \leq cD(x, y)$.*

Proof: Let f be an optimal non-contracting embedding of G_i, with distortion at most c. Consider any pair u, v of vertices that are embedded consecutively in f. We start by showing that $D(u, v) \leq cW$. Let T be the minimum spanning tree of H. If edge $\{u, v\}$ belongs to T, then $D(u, v) \leq W$. Otherwise, since T is connected, there is an edge $e = \{u', v'\}$ in tree T, such that both u and v are embedded inside e. But then $D(u', v') \leq W$, and since the embedding distortion is at most c, $|f(u) - f(v)| \leq |f(u') - f(v')| \leq cW$. As the embedding is non-contracting, $D(u, v) \leq cW$ must hold.

Consider now some pair $x, y \in X_i$ of vertices. If no vertex is embedded between x and y, then by the above argument, $D(x, y) \leq cW$, and thus the edge $\{x, y\}$ is in H' and $D_{H'}(x, y) = D(x, y)$. Otherwise, let $z_1, \dots, z_k$ be the vertices appearing in the embedding f between x and y (in this order). Then the edges $\{x, z_1\}, \{z_1, z_2\}, \dots, \{z_{k-1}, z_k\}, \{z_k, y\}$ all belong to H', and therefore

$$\begin{aligned}
D_{H'}(x, y) &\leq D_{H'}(x, z_1) + D_{H'}(z_1, z_2) + \dots D_{H'}(z_{k-1}, z_k) + D_{H'}(z_k, y) \\
&= D(x, z_1) + D(z_1, z_2) + \dots D(z_{k-1}, z_k) + D(z_k, y) \\
&\leq |f(x) - f(z_1)| + |f(z_1) - f(z_2)| + \dots + |f(z_{k-1}) - f(z_k)| + |f(z_k) - f(y)| \\
&= |f(x) - f(y)| \leq cD(x, y)
\end{aligned}$$

■

We can now concentrate on embedding graph H'. Since the weight of each edge in graph H' is bounded by $O(cW)$, we can use a modified version of the algorithm of [BDG+05] to embed each G_i. First, we need the following technical Claim.

Claim 3.4.3 *There exists a shortest path $p = v_1, \dots, v_k$, from u to u' in H', such that for any i, j, with $|i - j| > 1$, $D(v_i, v_j) = \Omega(W|i - j|)$.*

Proof: Pick an arbitrary shortest path, and repeat the following: while there exist consecutive vertices x_1, x_2, x_3 in p, with $D_{H'}(x_1, x_3) < cW$, remove x_2 from p, and add the edge $\{x_1, x_3\}$ in p. ■

The algorithm works as follows. We start with the graph H', and we guess points u, u', such that there exists an optimal embedding of G_i having u and u' as the left-most and right-most point respectively. Let $p = (v_1, \dots, v_k)$ be the shortest path from u to u' on H' (here $v_1 = u$ and $v_k = u'$), that is given by Claim 3.4.3. We partition X_i into clusters $V_1, \dots, V_k$, as follows. Each vertex $x \in X_i$ belongs to cluster V_j, that minimizes $D(x, v_j)$.

Our next step is constructing super-clusters $U_1, \dots, U_s$, where the partition induced by $\{V_j\}_{j=1}^k$ is a refinement of the partition induced by $\{U_j\}_{j=1}^s$, such that there is a small-cost spanning tree T' of G_i that "respects" the partition induced by $\{U_j\}_{j=1}^s$. More precisely, each edge of T' is either contained in a super-cluster U_i, or it is an edge of the path p. The final embedding of G_i is obtained by a walk on T', that traverses the super-clusters $U_1, \ldots, U_s$ in this order.

Note that there exist metrics over G_i for which any spanning tree that "respects" the partition induced by V_j's is much more expensive that the minimum spanning tree. Thus, we cannot simply use $U_j = V_j$.

We now show how to construct the super-clusters $U_1, \ldots, U_s$. We first need the following three technical claims, which constitute a natural extensions of similar claims from [BDG$^+$05] to the weighted case.

Claim 3.4.4 *For each $i : 1 \leq i \leq k$, $\max_{u \in V_i} \{D(u, v_i)\} \leq c^2 W/2$.*

Proof: Let $u \in V_i$. Consider the optimal embedding f. Since $f(v_1) = \min_{w \in X} f(w)$, and $f(v_k) = \max_{w \in X} f(w)$, it follows that there exists j, with $1 \leq j < k$, such that

$$\min\{f(v_j), f(v_{j+1})\} < f(u) < \max\{f(v_j), f(v_{j+1})\}.$$

Assume w.l.o.g., that $f(v_j) < f(u) < f(v_{j+1})$. We have $D(u, v_j) \geq D(u, v_i)$, since $u \in V_i$. Since f is non-contracting, we obtain $f(u) - f(v_j) \geq D(u, v_j) \geq D(u, v_i)$. Similarly, we have $f(v_{j+1}) - f(u) \geq D(u, v_i)$. Thus, $f(v_{j+1}) - f(v_j) \geq 2D(u, v_i)$. Since $\{v_j, v_{j+1}\} \in E(G')$, we have $D(v_j, v_{j+1}) \leq cW$. Thus, $c \geq \frac{f(v_{j+1}) - f(v_j)}{D(v_{j+1}, v_j)} \geq \frac{2D(u,v_i)}{cW}$. ■

Claim 3.4.5 *For each $r \geq 1$, and for each $i : 1 \leq i \leq k - r + 1$, $\sum_{j=i}^{i+r-1} |V_i| \leq c^2 W(c + r - 1) + 1$.*

Proof: Let $A = \bigcup_{j=1}^{i+r-1} V_i$. Let $x = \text{argmin}_{u \in A} f(u)$, and $y = \text{argmax}_{u \in A} f(u)$. Let also $x \in V_i$, and $y \in V_j$. Clearly, $|f(v_i) - f(v_j)| \leq cD(v_i, v_j) \leq cD_{G'}(v_i, v_j) \leq c^2 W |i - j| \leq c^2 W(r - 1)$. By Claim 3.4.4, we have $D(x, v_i) \leq c^2 W/2$, and $D(y, v_j) \leq c^2 W/2$. Thus, $|f(x) - f(v_i)| \leq cD(x, v_i) \leq c^3 W/2$, and similarly $|f(y) - f(v_j)| \leq c^3 W/2$. It follows that $|f(x) - f(y)| \leq |f(x) - f(v_i)| + |f(v_i) - f(v_j)| + |f(v_j) - f(y)| \leq c^3 W + c^2 W(r - 1)$. Note that by the choice of x, y, and since the minimum distance in M is 1, and f is non-contracting, we have $\sum_{j=i}^{i+r-1} |V_i| \leq |f(x) - f(y)| + 1$, and the assertion follows. ■

Claim 3.4.6 *If $\{x, y\} \in E(H')$, where $x \in V_i$, and $y \in V_j$, then $D(v_i, v_j) \leq cW + c^2W$, and $|i - j| = O(c^2)$.*

Proof: Since $\{x, y\} \in E(G')$, we have $D(x, y) \leq cW$. By Claim 3.4.4, we have $D(x, v_i) \leq c^2W/2$, and $D(y, v_j) \leq c^2W/2$. Thus, $D(v_i, v_j) \leq D(v_i, x) + D(x, y) + D(y, v_j) \leq cW + c^2W$.

By Lemma 3.4.2, we have that $D_{G'}(v_i, v_j) \leq cD(v_i, v_j) \leq c^2W + c^3W$. Since every edge of G' has length at least 1, we have $|i - j| \leq D_{G'}(v_i, v_j) \leq c^2W + c^3W$. ■

Let α be an integer with $0 \leq \alpha < c^4W$. We partition the set X_i into super-clusters $U_1, \ldots, U_s$, such that for each $l : 1 \leq l \leq s$, U_l is the union of c^4W consecutive clusters V_j, where the indexes j are shifted by α. We refer to the above partition as *α-shifted*.

Claim 3.4.7 *Let T be an MST of G_i. We can compute in polynomial time a spanning tree T' of G_i, with cost$(T') = O($cost$(T))$, and an α-shifted partition of X_i, such that for any edge $\{x, y\}$ of T', either both $x, y \in U_l$ for some $l : 1 \leq l \leq s$, or $x = v_j$ and $y = v_{j+1}$ for some $j : 1 \leq j < k$.*

Proof: Observe that since H is connected, all the edges of T can have length at most W, and thus T is a subgraph of both H and H'. Consider the α-shifted partition obtained by picking $\alpha \in \{0, \ldots, c^4W - 1\}$, uniformly at random. Let T' be the spanning tree obtained from T as follows: For all edges $\{x, y\}$ of T, such that $x \in V_i \subseteq U_{i'}$, and $y \in V_j \subseteq U_{j'}$, where $i' \neq j'$, we remove $\{x, y\}$ from T, and we add the edges $\{x, v_i\}$, $\{y, v_j\}$, and the edges on the subpath of p from v_i to v_j. Finally, if the resulting graph T' contains cycles, we remove edges in an arbitrary order, until T' becomes a tree. Note that although T' is a spanning tree of G_i, it is not necessarily a subtree of H'.

Clearly, since the edges $\{x, v_i\}$, and $\{y, v_j\}$ that we add at each iteration of the above procedure are contained in the sets $U_{i'}$, and $U_{j'}$ respectively, it follows that T' satisfies the condition of the Claim.

We will next show that the expectation of $\mathrm{cost}(T')$, taken over the random choice of α, is $O(\mathrm{cost}(T))$. For any edge $\{x, y\}$ that we remove from T, the cost of T' is increased by the sum of $D(x, v_i)$ and $D(y, v_j)$, plus the length of the shortest path from v_i to v_j in H'. Observe that the total increase of $\mathrm{cost}(T')$ due to the subpaths of p that we add, is at most $\mathrm{cost}(T)$. Thus, it suffices to bound the increase of $\mathrm{cost}(T')$ due to the edges $\{x, v_i\}$, and $\{y, v_j\}$.

By Claim 3.4.4, $D(x, v_i) \leq c^2W/2$, and $D(y, v_j) \leq c^2W/2$. Thus, for each edge $\{x, y\}$ that we remove from T, the cost of the resulting T' is increased by at most $O(c^2W)$.

For each i, the set $U_i \cup U_{i+1}$ contains $\Omega(c^4W)$ consecutive clusters V_j. Also, by Claim 3.4.6 the difference between the indexes of the clusters V_{t_1}, V_{t_2} containing the endpoints of an edge, is at most $|t_1 - t_2| = O(c^2)$. Thus, the probability that an edge of T is removed, is at most $O(\frac{1}{c^2W})$, and the expected total cost of the edges in $E(T') \setminus E(T)$ is $O(|X_i|) = O(\mathrm{cost}(T))$. Therefore, the expectation of $\mathrm{cost}(T')$, is at most $O(\mathrm{cost}(T))$. The Claim follows by the linearity of expectation, and by the fact that there are only few choices for α. ■

Let $U_1, \ldots, U_s$ be an α-shifted partition, satisfying the conditions of Claim 3.4.7, and let T' be the corresponding tree. Clearly, the subgraph $T'[U_i]$ induced by each U_i is a connected subtree of T'. For each U_i, we construct an embedding into the line by applying Lemma 3.4.1 on the spanning tree $T'[U_i]$. By Claim 3.4.5, $|U_i| = O(c^6W^2)$, and by Claim 3.4.4, the cost of the spanning tree $T'[U_i]$ of U_i is at most $O(|U_i|c^2W) = O(c^8W^3)$. Therefore, the embedding of each U_i, given by Lemma 3.4.1 has distortion $O(c^8W^3)$, and length $O(c^8W^3)$.

Finally, we construct an embedding for G_i by concatenating the embeddings computed for the sets $U_1, U_2, \ldots, U_s$, while leaving sufficient space between each consecutive pair of super-clusters, so that we satisfy non-contraction.

Lemma 3.4.8 *The above algorithm produces a non-contracting embedding of G_i with distortion $O(c^8W^3)$ and length $O(\mathit{cost}(MST(G_i)))$.*

Proof: Let g be the embedding produced by the algorithm. Clearly, g is non-contracting. Consider now a a pair of points $x, y \in X$, such that $x \in U_i$, and $y \in U_j$. If $|i - j| \leq 1$, then $|g(x) - g(y)| = O(c^8 W^3)$, and thus the distortion of $D(x, y)$ is at most $O(c^8 W^3)$.

Assume now that $|i - j| \geq 2$, and $x \in V_{i'}$, $y \in V_{j'}$. Then $|g(x) - g(y)| = O(|i - j| \cdot c^8 W^3)$. On the other hand, $D(x, y) \geq D(v_{i'}, v_{j'}) - D(v_{i'}, x) - D(v_{j'}, y) \geq D(v_{i'}, v_{j'}) - c^2 W \geq D_{H'}(v_{i'}, v_{j'})/c - c^2 W \geq |i' - j'|/c - c^2 W = \Omega(|i - j| c^4 W^2)$. Thus, the distortion on $\{x, y\}$ is $O(c^7 W^2)$. In total, the maximum distortion of the embedding g is $O(c^8 W^3)$.

In order to bound the length of the constructed embedding, consider a walk on T' that visits the vertices of T according to their appearance in the line, from left to right. It is easy to see that this walk traverses each edge at most 4 times. Thus, the length of the embedding, which is equal to the total length of the walk is at most $4\text{cost}(T') = O(\text{cost}(T))$. ■

The Final Embedding

We are now ready to give a detailed description of the final algorithm. Assume that the minimum distance in M is 1, and the diameter is Δ. Let $H = (X, E)$ be a graph, such that an edge $(u, v) \in E$ iff $D(u, v) \leq W$, for a threshold W, to be determined later. We use the algorithm presented above to embed every connected component $G_1, \ldots, G_k$ of H. Let $f_1, f_2, \ldots, f_k$ be the embeddings that we get for the components $G_1, G_2, \ldots G_k$ using the above algorithm, and let T be a minimum spanning tree of G. It is easy to see that T connects the components G_i using exactly $k - 1$ edges.[3] We compute our final embedding f as follows. Fix an arbitrary Eulerian walk of T. Let P be the permutation of $(G_1, G_2, \ldots, G_k)$ that corresponds to the order of the first occurrence of any node of G_i in our traversal. Compute embedding f by concatenating the embeddings f_i of components G_i in the order of this permutation. Let T_i be the minimum spanning tree of G_i. Between every 2 consecutive embeddings in the permutation f_i and f_j, leave space $\max_{u \in G_i, v \in G_j} \{D(u, v)\} = D(a, b) + O(\text{cost}(T_i)) +$

[3]Follows from correctness of Kruskal's algorithm. These $k - 1$ edges are exactly the last edges to be added because they are bigger than W and within components we have edges smaller than W

$O(cost(T_j))$, where $D(a, b)$ is the smallest distance between components G_i and G_j. This implies the next two Lemmata.

Lemma 3.4.9 *The length of f is at most $O(c\Delta)$.*

Proof: The length of f is the sum of the lengths of all f_i and the space that we leave between every 2 consecutive f_i, f_j's. Then, by Lemma 3.4.8, the length of f_i is $O(c \cdot \text{cost}(T_i))$. Thus, the sum of the lengths of all f_i's is $O(c \cdot \text{cost}(T))$. The total space that we leave between all pairs of consecutive embeddings f_i is $\text{cost}(T) + 2\sum_{i=1}^{k} O(\text{cost}(T_i)) = O(\text{cost}(T))$. Therefore the total length of the embedding f is $O(\text{cost}(T))$. At the same time, the cost of T is at most the length of the optimal embedding f, which is $O(c\Delta)$. The statement follows. ■

Lemma 3.4.10 *Let $a \in G_i, b \in G_j$ for $i \neq j$. Then $W \leq D(a, b) \leq |f(a) - f(b)| \leq O(c\Delta) \leq O(cD(a, b)\frac{\Delta}{W})$*

Proof: The first part $D(a, b) \leq |f(a) - f(b)|$ is trivial by construction, since we left enough space between components G_i and G_j. Since a and b are in difference connected components, we have $D(a, b) > W$. Using Lemma 3.4.9 we have that $|f(a) - f(b)| = O(c\Delta) = O(c\Delta\frac{D(a,b)}{W}) = O(cD(a, b)\frac{\Delta}{W})$. ■

Theorem 3.4.11 *Let $M = (X, D)$ be a metric with spread Δ, that embeds into the line with distortion c. Then, we can compute in polynomial time an embedding of M into the line, of distortion $O(c^{11/4}\Delta^{3/4})$.*

Proof: Consider any pair of points. If they belong to different components, their distance distortion is $O(c\Delta/W)$ (Lemma 3.4.10). If they belong to the same component, their distance distortion is $O(c^8W^3)$ (Lemma 3.4.8). Setting $W = \Delta^{1/4}c^{-7/4}$ gives the claimed distortion bound. ■

3.4.4 Hardness of Embedding Into the Line

In this section we show that even the problem of embedding weighted trees into the line is n^{β}-hard to approximate, for some constant $0 < \beta < 1$. Our reduction is performed from the 3SAT(5) problem, defined as follows. The input is a CNF formula φ, in which each clause consists of exactly 3 different literals and each variable participates in exactly 5 clauses, and the goal is to determine whether φ is satisfiable. Let $x_1, \ldots, x_n$, and $C_1, \ldots, C_m$, be the variables and the clauses of φ respectively, with $m = 5n/3$. Given an input formula φ, we construct a weighted tree G, such that if φ is satisfiable then there is an embedding of G into the line with distortion $O(b)$ (for some $b = \text{poly}(n)$) and if φ is not satisfiable, then the distortion of any embedding is at least $b\tau$, where $\tau = \text{poly}(n)$. The construction size is polynomial in τ, and hence the hardness result follows.

The construction

Our construction makes use of *caterpillar* graphs. A caterpillar graph consists of a path called *body*, and a collection of vertex disjoint paths, called *hairs*, while each hair is attached to a distinct vertex of the body, called the *base* of the hair. One of the endpoints of the caterpillar body is called the *first vertex* of the caterpillar, and the other endpoint is called the *last vertex*. We use two integer paremeters $b = \text{poly}(n)$ and $\tau = \text{poly}(n)$, whose exact value is determined later. We call a caterpillar graph a *canonical caterpillar*, if: (1) its body consists of integer-length edges, (2) the length of each hair is a multiple of b, and (3) each hair consists of edges of length $\frac{1}{b\tau}$. Our weighted tree G is a collection of canonical caterpillars, connected together in some way specified later. Notice that in any embedding of a canonical caterpillar with distortion less than $b\tau$, each hair must be embedded continuously (the formal proof appears below). Let $B_1, \ldots, B_t$ be caterpillars. A *concatenation* of $B_1, \ldots, B_t$ is a caterpillar obtained by connecting each pair of consecutive caterpillars B_i, B_{i+1} for $1 \leq i < t$ with a unit-length edge between the last vertex of B_i and the first vertex of B_{i+1}.

The building blocks of our graph G are literal caterpillars, variable caterpillars and clause caterpillars, that represent the literals, the variables and the clauses of the input formula φ. All these caterpillars are canonical. Let x_i be some variable in formula φ. We define two caterpillars called *literal caterpillars* w_i and w'_i, which represent the literals x_i and $\overline{x}_i$, respectively. Additionally, we have a *variable* caterpillar v_i representing variable x_i.

Let Y_L and Y_R be caterpillars whose bodies contain only one vertex (denoted by L and R respectively), with a hair of length $\tau^3 b$ (denoted by H_L and H_R respectively) attached to the body. The main part of our graph G is a canonical caterpillar W, defined as a concatenation of $Y_L, w_1, w'_1, w_2, w'_2, \ldots w_n, w'_n, Y_R$. The hairs of H_L and H_R are used as padding, to ensure that all the vertices of $G \backslash (H_L \cup H_R)$, are embedded between L and R. The length of the body of W is denoted by N, and is calculated later. Variable caterpillars v_i attach to W as follows. The first vertex of v_i connects by a unit-length edge to the first vertex of w'_i.

For every clause C_j in formula φ, our construction contains a canonical caterpillar k_j representing it, which is also called a *key*. Each key k_j is attached to vertex L by an edge of length N. Figure 3-3 (which appears in the Appendix) summarizes the above described construction.

We now provide the details on the structure of the literal caterpillars. Consider a literal ℓ, and let w be the caterpillar that represents it (i.e., if ℓ is x_i or $\overline{x}_i$, then w is w_i or $\overline{w}_i$). Assume that ℓ participates in (at most 5) clauses $C_1^\ell, C_2^\ell, \ldots$. Then w is the concatenation of at most 5 caterpillars, denoted by $h_1^\ell, h_2^\ell, \ldots$, that represent the participation of ℓ in these clauses (see Figure 3-2). Following [Ung98], we call these caterpillars *keyholes*. For convenience, we ensure that for each literal ℓ there are exactly 5 such keyholes $h_1^\ell, h_2^\ell, \ldots, h_5^\ell$, as follows. If the literal participates in less than 5 clauses, we use several copies of the same keyhole that corresponds to some clause in which ℓ participates. Thus, for each clause, for each literal participating in this clause, there is at least one keyhole. All the keyholes that correspond to the same clause C_j are copies of the same caterpillar $h(j)$, called *the keyhole* of C_j.

The main idea of the construction is as follows. First, the keys and the keyholes are designed in a special way, such that in order to avoid the distortion of $b\tau$, each key k_j has to be embedded inside one of the matching keyholes (copies of $h(j)$). The variable caterpillars are shaped in such a way that in any embedding with distortion less than $b\tau$, each variable caterpillar v_i is either embedded in w_i or w'_i. If v_i is

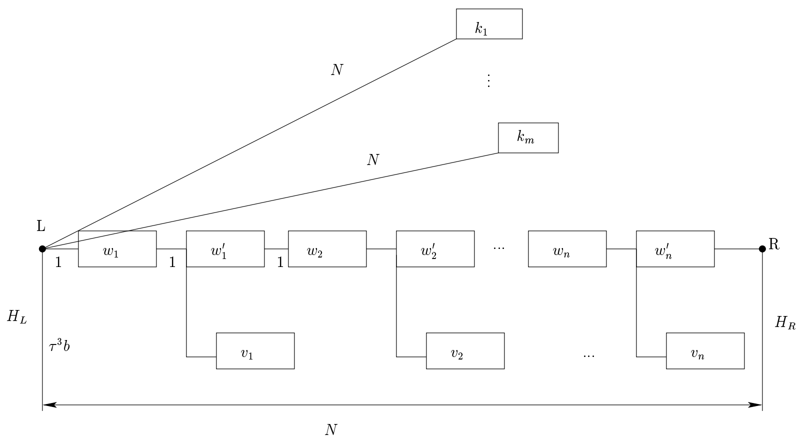


Figure 3-3: The high-level view of the construction.

embedded in w_i, then no key can be embedded inside any keyhole belonging to w_i without incurring the distortion of $b\tau$, and the same is true in case v_i is embedded into w_i'. Suppose formula φ is satisfiable. Then embedding of G with distortion $O(b)$ is obtained as follows. We first embed hair H_L (starting from the vertex furthest from L), then the body of W and then H_R (starting from the vertex closest to R). For each variable x_i, if the correct assignment to x_i is TRUE, then variable caterpillar v_i is embedded inside the literal caterpillar w_i', and otherwise it is embedded inside w_i. Given a clause C_j, if ℓ is the satisfied literal in this clause, we embed the key k_j in the copy of keyhole $h(j)$, that corresponds to literal ℓ. On the other hand, if φ is not satisfiable, we still need to embed each variable caterpillar v_i inside one of the two corresponding caterpillars w_i, w_i', thus defining an assignment to all the variables. For example, if v_i is embedded inside w_i, this corresponds to the assignment FALSE to variable x_i. Such embedding of v_i will block all the keyholes in the caterpillar w_i. Since the assignment is non-satisfying, for at least one of the keys k_j, all the corresponding keyholes (copies of $h(j)$) are blocked, and so in order to embed k_j, we will need to incur a distortion of $b\tau$.

Keys and Keyholes

We start with the following definition.

Definition 11 *For an integer α, a* barrier *caterpillar of length α consists of a body* *of α unit-length edges, and a hair of length b, attached to each one of the vertices of the body.*

Observe that the length of an embedding of a barrier of length α is at least αb. Intuitively, a barrier B of a "proper" length makes it impossible to embed a "short" edge (u, v) such that u and v are on the opposite sides of B, without incurring high distortion.

For a clause C_j, the corresponding keyhole $h(j)$ consists of three parts: *prefix*, *suffix* and the *main part*.

The prefix caterpillar, denoted by P, starts with a barrier of size τ^3, which is connected by an edge of length τ^2, called *large* edge, to vertex s which in turn is connected by a unit-length edge to a barrier of size $3\tau^4$. There is also a hair of length $b\tau^2$, called *large* hair, that attaches to vertex s.

The suffix caterpillar is denoted by S, and it is the mirror reflection of the prefix, where vertex s is denoted by t (see Figure 3-4).

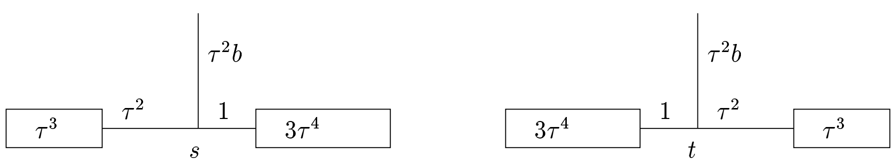


Figure 3-4: The prefix and the suffix.

The main part of keyhole $h(j)$ corresponding to clause C_j consists of m caterpillars $Q_1, Q_2, \ldots, Q_m$. Caterpillar Q_i, for $1 \le i \le j$ consists of a vertex z_i with a hair of length τb attached to it, which is referred to as a *small hair*. Vertex z_i connects with an edge of length τ (called a *small edge*) to a barrier of size τ^2. For $j < i \le m$, caterpillar Q_i is just a barrier of size τ^2. The keyhole h_j is defined to be the concatenation of $P, Q_1, \ldots, Q_m, S$.

We now proceed to define the keys. A key k_j is defined identically to the keyhole h_j, with the following changes:

- Observe that in the body of prefix P of $h(j)$, vertex s is adjacent to two edges, of sizes τ^2 and 1. We switch these two edges. We do the same with the two edges adjacent to vertex t in the body of suffix S. The resulting prefix and suffix are denoted by P' and S' respectively.

- Observe that each vertex z_i, $1 \leq i \leq j$ is attached in the body of $h(j)$ to two edges, of sizes 1 and τ. We switch these two edges.

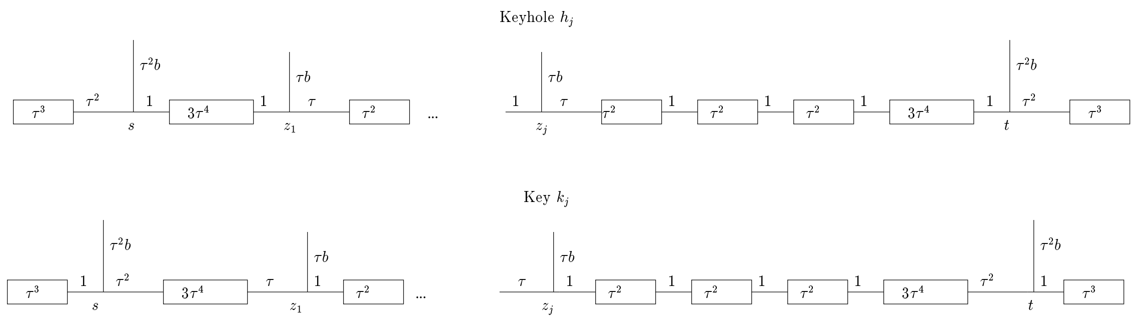


Figure 3-5: The key and the keyhole.

The intuition is that when any key is embedded into a keyhole, the two large hairs of the key have to be embedded inside the two large edges of the keyhole and vice versa, while the small hairs of both key and keyhole are embedded between the two long hairs. Similarly, the small hairs of the key have to be embedded inside the small edges of the keyhole and vice versa. Moreover, inside each small edge of a key (keyhole), at most one small hair of a keyhole (key) can be embedded, if the distortion is less than τb. Assume now that the key and the keyhole do not match, for example, we have key k_j and keyhole $h(i)$ where $j < i$. Then the number of small hairs in the keyhole is larger than the number of small edges in the key, and the distortion of embedding key k_j into keyhole $h(i)$ is large.

Variable caterpillars

We now define caterpillars v_i, representing variable x_i in formula φ.

Caterpillar v_i is a concatenation of five identical caterpillars $L_1, \ldots, L_5$. Caterpillar L_j for $1 \leq j \leq 5$ consists of three parts: The prefix P' and the suffix S' are identical to the prefix and the suffix of a key; the main part consists of m barriers of size τ^2 each, where each pair of consecutive barriers is connected by an edge of length τ.

The idea is that when v_i is embedded into w_i or w_i', then each one of the caterpillars $L_1, \ldots, L_5$ will be embedded into the 5 corresponding keyholes, thus blocking them. More precisely, the 10 large hairs of v_i will be embedded into the 10 large edges of $L_1, \ldots, L_5$, ensuring that no large hair of any key can be embedded there.

Construction Size

We fix $\tau = n^{\mu}$ for some large integer μ. Our first step is bounding the length N of the body of W. Recall that W consists of $2n$ literal caterpillars, each consisting of 5 keyholes. The length of a keyhole is at most $m(\tau^2+\tau+1)+6\tau^4+2\tau^3+2\tau^2+2 < 7\tau^4$. Therefore, $N = O(\tau^4 n)$. We set $b = 3N$.

One can easily see that the size of the construction is dominated by the number of vertices on the hairs H_R and H_L. The length of each one of these hairs is $\tau^3 b$, and the length of each edge on a hair is $\frac{1}{b\tau}$. Therefore, the construction size is $O(\tau^4 b^2) = O(\tau^{12} n^2)$.

Analysis

In the following, we consider an embedding f of our graph G with distortion less than τb. We start by showing several structural properties of this embedding.

Claim 3.4.12 *Each hair of each caterpillar is embedded continuously.*

Proof: Assume otherwise. Then there is an edge $e = (x, y)$ on some hair H, and a vertex v not belonging to H embedded inside e. But the length of e is only $\frac{1}{\tau b}$, while the distance $D(x, v)$ is at least 1, and thus the distortion is at least τb. ∎

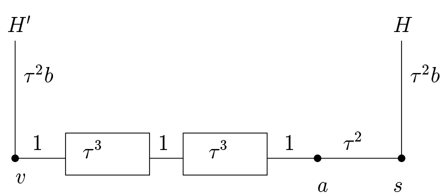
Claim 3.4.13 *The set of vertices in $G \setminus (H_L \cup H_R)$ is embedded continuously between the embeddings of L and R.*

Proof: By Claim 3.4.12, H_L and H_R are embedded continuously. Since the length of each H_L, and H_R is $\tau^3 b$, and the length of the longest edge of W is τ^2, it follows that $G \setminus (H_L \cup H_R)$ also has to be embedded continuously. Thus, in order to avoid distortion larger than τb, $G \setminus (H_L \cup H_R)$ has to be embedded between L and R. ∎

Our next goal is to prove that given some large edge $e = (u, v)$ on the body of W (which must belong to the prefix or the suffix of one of the keyholes), the only large hair of W that is embedded in it is the hair attached to u or v. The meaning of this claim is that the embedding of W has to be "nice", with the main part of each keyhole embedded between its prefix and suffix.

Claim 3.4.14 *Let h^i_j be any keyhole on caterpillar W, and let e be one of its large edges (assume w.l.o.g. that this edge is from its prefix). Let H be the large hair belonging to the prefix. Then H is the only hair belonging to W embedded inside e.*

Proof: We denote $e = (s, a)$, where s is the base of hair H. Recall that there is a barrier B_1 of size τ^3 attached to a. If h^i_j is not the first keyhole of W, then there is a suffix of another keyhole adjacent to B_1, with a barrier B_2 of size τ^3 attached to B_1 by a unit-length edge. The other endpoint of B_2 attaches by a unit-length edge to a base of a large hair H'. Clearly, H is embedded inside edge e continuously. Since the length of H is $\tau^2 b$, barriers B_1, B_2, and hair H' are embedded on the same side of H as vertex a.



Assume the claim is false, and let H'' be some other large hair belonging to some keyhole embedded inside e. Let x be the base of this hair. Since hair H'' is embedded inside edge e, so is its base x. Recall that vertex x attaches with a unit-length edge to a barrier B' of length $3\tau^4$. As the body of this barrier consists of unit-length edges, it has to be embedded completely between the embeddings of H and H'. The distance between s and the base of H' is only $2\tau^3 + \tau^2 + 3$, and thus the distance between their images in the embedding is at most $2\tau^4 b + \tau^2 b + 3b$. On the other hand, the size of the embedding of B' must be at least $3\tau^4 b$.

The only case we still need to consider is when h^i_j is the first keyhole on W. But then it is easy to see that the barrier B' has to be embedded between the embeddings of H and the hair H_L, which is again impossible.

■

The next corollary follows from Claim 3.4.14 and uses the fact that the main part of each keyhole only contains edges of length at most τ.

Corollary 3.4.15 *The main part of each keyhole is embedded between the two large hairs of the prefix and the suffix of the keyhole. Moreover, the large hairs of caterpillar W are embedded in the same order in which they appear on the body of W.*

Proof: Consider some keyhole h_j, and path P between s and t on its body. Recall that s and t serve as bases of large hairs whose length is $\tau^2 b$, and every edge on path P is of length at most τ. Therefore, all the vertices on path P and the hairs attached to them have to be embedded between the embeddings of these two large hairs.

Assume now that the large hairs on caterpillar W are not embedded in the same order in which they appear in W. Then there are three hairs H_1, H_2, H_3, such that H_1 and H_2 appear consecutively in W, but H_3 is embedded between H_1 and H_2. Let a and b be the bases of hairs H_1 and H_2. Then H_3 is embedded inside some edge e on the path (a, b). In order to avoid distortion τb, e has to be a large edge, and the only large edges between a and b are the two edges adjacent to a and b inside which the hairs H_1 and H_2 are embedded, which contradicts Claim 3.4.14 ■

We prove next that for any large edge on any keyhole, at most one large hair of any key or a variable caterpillar can be embedded inside it.

Claim 3.4.16 *Let h_i be some keyhole, and let e be one of its large edges. Then there is at most one large hair belonging to any key or a variable caterpillar embedded inside e.*

Proof: Denote the endpoints of e by $\{v, u\}$. From the construction, there is a large hair H attached to one of these vertices, assume it's u. Recall also that both v and u are connected to barriers of size at least τ^3. Clearly, hair H is embedded inside e right next to vertex u. Suppose there are two other large hairs, H' and H'' embedded inside e, and assume that H'' is embedded between H and H'. Denote the base of the hair H'' by v''. Recall that v'' is connected by unit-length edge to a barrier of length τ^3. It is impossible to embed this whole barrier inside edge e, since the total length of such an embedding would be $\tau^3 b$, while the length of edge e is only τ^2. Therefore, there is at least one unit-length edge e' (part of the barrier body), whose one endpoint is embedded next to H'' and whose other endpoint is embedded outside e. But then one of the hairs H', H is embedded inside e', so it is impossible that the distortion is less than τb.

■

Using the same reasoning, we can prove the following two claims:

Claim 3.4.17 *For each small edge in a keyhole, only one small hair belonging to any key or a variable caterpillar can be embedded inside it.*

Claim 3.4.18 *For every key, for each one of its large (small, respectively) edges, at most one large (small, respectively) hair of a keyhole can be embedded inside it.*

Additionally, observe that the main part of any key k_i must be embedded completely between the prefix and the suffix of some keyhole h_j^ℓ and the large hairs of k_i are embedded into large hairs of h_j^ℓ. In this case we say that key k_i is embedded inside keyhole h_j^ℓ.

Yes instance

Note that the distance between any two vertices on the bodies of any caterpillars in our construction is at most $3N = b$.

Claim 3.4.19 *For each j, with $1 \leq j \leq m$, key k_j can be embedded inside a copy of $h(j)$ with distortion $O(b)$.*

Proof: The embedding is as follows. We move from left to right. While embedding the barriers, we embed a hair from the key and then a hair from the keyhole interchangeably, as follows: let H be a hair from the key and H' be a hair from a keyhole. We first embed H starting from its base, then we embed H' starting from the vertex furthest from its base. The distance between the embeddings of H and H' is $3b$, and thus the maximum stretch of an edge on the bodies of the barriers is $O(b)$. The large and the small hairs are embedded inside the large and the small edges respectively as follows. Let the endpoints of the large (small) edge of the key be denoted by v, u (the hair is attached to v), and denote the endpoints of the large (small) edge of the keyhole by u', v', the hair being attached to v'. We first embed vertex u', then the large (small) hair of the key, starting from v, then the large (small) hair of the keyhole (starting from the endpoint opposite to v', so v' is embedded last), and then vertex u. In case H, H' are large, the distance between their embeddings is $2\tau^2 b + b$, and if they are small, the distance is $2\tau b + b$. In any case, the distortion of this embedding is at most $O(b)$. ■

For each variable caterpillar v_i, we can view its five sub-caterpillars $L_1, \ldots, L_5$ as "master keys" that can be embedded into any keyhole. We say that variable caterpillar v_i is embedded inside literal w iff the five sub-caterpillars of v_i are embedded into the five keyholes of w.

Similarly to Claim 3.4.19, we can prove the following claim.

Claim 3.4.20 *For each $i : 1 \leq i \leq n$, variable caterpillar v_i can be embedded inside each one of the literal caterpillars w_i or w_i' with distortion at most $O(b)$.*

Lemma 3.4.21 *If φ is satisfiable, then there exists an embedding of G into the line, with distortion at most $O(b)$.*

Proof: Consider the satisfying assignment to the variables, and assume the assignment to x_i is TRUE. Then, we embed v_i inside w_i'. Each clause contains at least one literal that satisfies it, so no variable caterpillar is embedded on this literal. We embed the key corresponding to the clause on the keyhole that belongs to that literal.

Finally, we embed H_L and H_R, to the left and to the right of the image of G, respectively. The maximum distortion of this embedding is at most $O(b)$. ■

Unsatisfiable instance

Claim 3.4.22 *Suppose we have any embedding with distortion less than τb. Then each key is embedded in one of its corresponding keyholes.*

Proof: Suppose key k_i is embedded inside some keyhole h_j and $i \neq j$ (w.l.o.g., let $i < j$). Since all the small edges of k_i and the small hairs of h_j are embedded between the long hairs of k_i, and the number of small edges of k_i is less than the number of small hairs of h_j, the distortion must be at least $b\tau$. ■

Claim 3.4.23 *Each variable caterpillar v_i is embedded inside either w_i, or w_i'. Moreover, once we embed v_i inside w_i or w_i', it is impossible to embed any keys inside keyholes of w_i or w_i', respectively, without incurring distortion τb.*

Proof: Let v_i be some variable caterpillar. Observe that there are 10 large hairs in v_i, which, in order to avoid distortion of τb, have to be embedded into 10 large edges of W. We prove that these have to be 10 consecutive large edges of w_i or of w_i'. Recall that the large hairs of W are embedded in the order in which they appear in W, each one of them is embedded into its adjacent large edge. The edge that attaches v_i to W is unit length, thus the first large hair of v_i has to be embedded into the hair of w_i' or w_i that lies closest to v_i. Observe also that large hairs of W can only be embedded inside large edges of v_i, and only one such hair is embedded into any large edge of v_i. Therefore, all the large hairs of v_i have to be embedded into the large edges of w_i or into the large edges of w_i'. Assume we embed v_i into w_i. Then inside each large edge of w_i, there is a large hair of v_i embedded in it. By Claim 3.4.16, it is impossible to embed additional large edge into this edge, thus none of the keys can be embedded into keyholes belonging to w_i. ■

Lemma 3.4.24 *If φ is not satisfiable, then any embedding of G into the line has distortion at least τb.*

Proof: Assume we have an embedding with distortion less than τb. Then by Claim 3.4.22, each variable must be embedded in one of its corresponding literals, which implies an assignment to the variables. This assignment is not a satisfying one, so for some clause, for each one of its literals, there is a variable caterpillar embedded inside them, so it is impossible to embed the key corresponding to the clause into one of its keyholes, and the distortion must be at least τb. ■

Theorem 3.4.25 *Given an M-point metric that c-embeds into the line, it is NP-hard to compute an embedding with distortion less than $\Omega(cM^{1/12-\epsilon})$ for arbitrarily small constant ϵ.*

Proof: Recall that our construction size is $M = \tau^{12}n^2$. If φ is satisfiable, then there is an embedding with distortion $O(b)$. Otherwise, any embedding has distortion at least τb. Since $\tau = n^{\mu}$ for a large enough constant μ, the theorem follows.

■

3.4.5 Approximation Algorithm for Weighted Trees

In this section we consider embedding of weighted trees into the line. Given a weighted tree T, let φ be its optimal embedding into the line, whose distortion is denoted by c (we assume that $c \geq 200$). We provide a poly(c)-approximation algorithm, which, combined with earlier work, implies $n^{1-\epsilon}$ approximation algorithm for weighted trees, for some constant $0 < \epsilon < 1$. The first step of our algorithm is guessing the optimal distortion c, and from now on we assume that we have guessed its value correctly.

We start with notation. Fix any vertex r of the tree to be the root. Given a vertex $v \neq r$, denote $d(v) = D(v, r)$. Consider any edge $e = (u, v)$. The length of e is denoted by w_e, and $d_e = \min\{d(u), d(v)\}$ is the distance of e from r. We say that e is a *large* edge if $w_e \geq \frac{d_e}{c}$, it is a *medium* edge if $\frac{d_e}{c} > w_e \geq \frac{d_e}{c^2}$, and otherwise e is a *small* edge.

Claim 3.4.26 *If $e = (u, v)$ is a medium or a small edge, then r is not embedded between u and v in the optimal solution.*

Proof: Assume otherwise. Then $|\varphi(u) - \varphi(v)| \geq d_e$. But $D(u, v) = w_e < \frac{d_e}{c}$, and edge e is stretched by a factor greater than c. ■

Let $\mathcal{C}$ be the collection of connected components, obtained by removing all the large edges from the graph. For each component $C \in \mathcal{C}$, let $r(C)$ denote its "root", i.e. the vertex of C closest to r in tree T. We also denote by $e(C)$ the unique edge incident on $r(C)$ on the path from $r(C)$ to r, and by $\alpha(C)$ the length of this edge.

PROCEDURE PARTITION

Let $\mathcal{C}$ be the current set of all the components.
While there is a large component $C \in \mathcal{C}$, with a medium-sized edge e on the path from $r(C)$ to $\ell(C)$, such that the removal of e splits C into two large components, do:

Let C' and C'' be the two large components obtained by removing e. Remove C from $\mathcal{C}$ and add C' and C'' to $\mathcal{C}$.

Clearly, in the optimal solution, the embedding of component C lies completely to the left or to the right of r.

Given some component $C \in \mathcal{C}$, let $\ell(C)$ be the vertex in C that maximizes $D(r(C), \ell(C))$, and let $P(C)$ be the path between $r(C)$ and $\ell(C)$ in tree T. We define the *radius* of C to be $s(C) = D(r(C), \ell(C))$. Component C is called *large* if $s(C) > c^4\alpha(C)$, otherwise the component is called *small*. We define a tree T' of components, whose vertex set is $\mathcal{C} \cup \{r\}$, and the edges connecting the components are the same as in the original graph, (i.e., $e(C)$ for all $C \in \mathcal{C}$.)

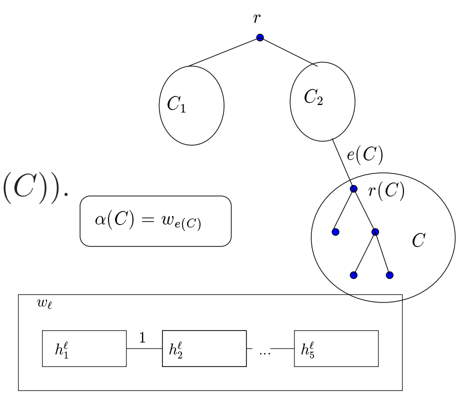


Figure 3-2: Caterpillar representing literal ℓ

The main idea of our algorithm is to find the embedding of each one of the components separately recursively, and then concatenate these embeddings in some carefully chosen order. However, there is a problem with this algorithm, which is illustrated by the following example. Consider a large component C, consisting of a very long path, and a small component C' attached to this path in the middle. In this case any small-distortion embedding has to interleave the vertices of C and C', and thus our algorithm fails. We note that as $e(C')$ is a large edge, vertices of component C' have to be embedded into medium-sized edges of C (formal proof of this fact is provided later). In order to solve the above problem, we perform PROCEDURE PARTITION, that further subdivides large components by removing some medium-size edges from them.

From now on we only consider the components after the application of the above procedure, and the component graph, the values $r(C), \ell(C), \alpha(C)$ and so on are defined with respect to these components. It is easy to see that if a medium size edge e is incident on some component C, then C is a large component.

In fact, it is more convenient for us to define and solve a slightly more general problem. In the modified problem, in addition to a weighted tree T, we are also given a threshold value H. Given any embedding of our tree into the line, we say that it satisfies the *root condition* if: (1) each component C is embedded completely to the right or to the left of r, and (2) no component C with $\alpha(C)+s(C) \geq cH$ is embedded to the right of r. Our goal is to find an embedding that satisfies the root condition, while minimizing its distortion. Even though the problem might look artificial at this point, it is easy to see that by setting $H = \infty$, it converts to our original problem. The reason for defining the problem this way is that our algorithm solves the problem recursively on each component $C \in \mathcal{C}$, and then concatenates their embeddings into the final solution. In order to avoid large distortion of the distance between r and $r(C)$, we need to impose the root condition on the sub-problem corresponding to C with threshold $H = D(r, r(C))$. We later claim that for each sub-problem there is an optimal embedding with distortion c that satisfies the corresponding root condition.

The Structure of the Optimal Solution

In this section we explore some structural properties of the optimal solution, on which our algorithm relies.

Definition 12 *Let C, C' be two large components. We say that these components are* incompatible *if $s(C) > 2c^3\alpha(C')$ and $s(C') > 2c^3\alpha(C)$.*

The proof of the following lemma appears in section 3.4.5.

Lemma 3.4.27 *If C and C' are large incompatible components, then in the optimal solution they are embedded on different sides of r.*

Definition 13 *Let C be a large component, and C' a small component. We say that there is a* conflict *between C and C' iff $2c^4\alpha(C) < \alpha(C') < s(C)/2c^4$.*

Lemma 3.4.28 *If C is a large component having a conflict with small component C', then C and C' are embedded on different sides of r in the optimal solution.*

The proof of the above lemma can be found in section 3.4.5.

Claim 3.4.29 *Let C, C' be large components and C'' a small component. Moreover, assume that there is a conflict between C and C'' and there is a conflict between C' and C''. Then C and C' are incompatible.*

Proof: Since there is a conflict between C and C'', $\alpha(C'') > 2c^4\alpha(C)$. A conflict between C' and C'' implies that $\alpha(C'') < s(C')/2c^4$. Therefore, $s(C') > 2c^3\alpha(C)$. Similarly, we can prove that $s(C) > 2c^3\alpha(C')$. ■

We subdivide the small components into types or subsets $\mathcal{M}_1, \mathcal{M}_2, \ldots$. We say that a small component C is of *type* i and denote $C \in \mathcal{M}_i$ iff $c^{i-1} \leq \alpha(C) < c^i$.

Claim 3.4.30 *For each i, $|\mathcal{M}_i| \leq 4c^4$.*

Proof: Consider some $i \geq 1$, and assume that $|\mathcal{M}_i| > 4c^4$. Then in the optimal solution, there are more than $2c^4$ components of type i embedded on one of the sides of r. Denote these components by $C_1^i, C_2^i, \ldots, C_k^i$, $k > 2c^4$, and assume that vertices $r(C_j^i)$ are embedded in the optimal solution in this order, where $r(C_1^i)$ is embedded closest to r. It is easy to see that for any pair C, C' of small components, the distance between $r(C)$ and $r(C')$ is at least $\frac{\alpha(C)}{c}$. As the optimal embedding is non-contracting, for every $j = 1, \ldots, k-1$, there is a distance of at least $\alpha(C_j^i)/c \geq c^{i-2}$ between the embedding of $r(C_j^i)$ and $r(C_{j+1}^i)$. Therefore, $r(C_k^i)$ is embedded at a distance at least $kc^{i-2} > 2c^{i+2}$ from r. However, $d(r(C_k^i)) \leq \alpha(C_k^i) + c\alpha(C_k^i) \leq 2c^{i+1}$, and thus this distance is distorted by more than a factor of c in the optimal embedding. ■

The Approximation Algorithm

Our algorithm consists of three major phases. In the first phase we compute the set $\mathcal{C}$ of components, after performing Procedure Partition. In the second phase, we solve the problem recursively for each one of the components $C \in \mathcal{C}$, where the threshold for the root condition becomes $H = D(r(C), r)$. In the final phase, we combine the recursive solutions to produce the final embedding.

Claim 3.4.31 *For each recursive call to our algorithm, there is an embedding of the corresponding instance with distortion c, that satisfies the root condition.*

Proof: Let C be a component, and let C' be a component obtained after decomposing C. We consider the recursive call in C'. Since C is just a subtree of T, it embeds into the line with distortion c. Let f be such an embedding of C with distortion c. W.l.o.g., we can assume that $r(C')$ is embedded to the left of $r(C)$. It suffices to show that f satisfies the root condition in component C'.

Observe that for the recursive call in C', the threshold value is $H = D(r(C), r(C'))$. All the edges of C' as not large w.r.to $r(C)$, thus all the vertices of C' are embedded to the left of $r(C)$. Assume now that the root condition is not satisfied for C'. This implies that there exists a component C'' that is obtained after decomposing C', such that $\alpha(C'') + s(C'') \geq cH$, and such that C'' is embedded to the left of $r(C')$. Thus, $f(r(C')) < f(l(C'')) < f(r(C))$. It follows that $|f(r(C')) - f(r(C))| > |f(r(C')) - f(l(C''))| \geq D(r(C'), l(C'')) = \alpha(C'') + s(C'') \geq cH = cD(r(C'), r(C))$, a contradiction. ■

The final embedding is produced as follows. First, partition the set $\mathcal{C}$ of components into two subsets $\mathcal{R}$, $\mathcal{L}$, containing the components to be embedded to the right and to the left of r, respectively. The partition procedure is explained below. The components in $\mathcal{L}$ are then embedded to the left of r, while the embedding of each component is determined by the recursive procedure call, and the embeddings of different components do not overlap. The order of components is determined as follows. For each small component C, let $f(C) = \alpha(C)$, and for each large component C', let $f(C') = s(C')/2c^4$. The order of embedding is according to $f(C)$, where the component C with smallest $f(C)$ is embedded closest to the root r. The embedding of components in $\mathcal{R}$ is performed similarly, except that the embedding of each component is the mirror image of the embedding returned by the recursive procedure call (so that the root condition holds in the right direction). We put enough empty space between the embeddings of different components to ensure that the embedding is non-contracting. In the rest of this section we show how to partition $\mathcal{C}$ into the subsets $\mathcal{R}$ and $\mathcal{L}$.

We start with large components. We translate the problem into an instance of 2SAT, as follows. We have one variable $x(C)$ for each large cluster C. Embedding C to the left of r is equivalent to setting $x(C) = T$. If two components C and C' are incompatible, we ensure that variables $x(C)$ and $x(C')$ get different assignments, by adding clauses $x(C) \vee x(C')$ and $\overline{x(C)} \vee \overline{x(C')}$. Additionally, if $s(C) + \alpha(C) > cH$, then we ensure that C is not embedded to the right of r by adding a clause $x(C) \vee x(C)$. The optimal solution induces a satisfying assignment to the resulting 2SAT formula, and hence we can find a satisfying assignment in polynomial time. The clusters C with $x(C) = T$ are added to $\mathcal{L}$ and all other clusters are added to $\mathcal{R}$.

Consider now any small cluster C. If $s(C) + \alpha(C) > cH$, then we add C to $\mathcal{L}$. Otherwise, if $s(C) + \alpha(C) \leq cH$, then there is at most one large component C' that has conflict with C. If such a component C' exists, then we embed C on the side opposite to that where C' is embedded. Otherwise, C is embedded to the left of r. Clearly, in any embedding consistent with the above decision the root condition is satisfied.

The analysis of this phase of the algorithm appears in Section 3.4.5, together with the proof of the following theorem:

Theorem 3.4.32 *The algorithm produces a non-contracting embedding with distortion bounded by $c^{O(1)}$.*

Large Incompatible Components

The goal of this section is to prove Lemma 3.4.27

We start with the following claim:

Claim 3.4.33 *Let C and C' be two large incompatible components. Then in the optimal solution, vertex $\ell(C')$ is* **not** *embedded inside any edge of $P(C)$.*

Proof: Assume otherwise, and let $e = (u, v)$ be an edge of $P(C)$, with $d(u) < d(v)$, such that $\ell(C')$ is embedded between u and v. In order to finish our proof, it is enough to show that $D(u, \ell(C')) \geq d(u)$: in this case, if $\ell(C')$ is embedded between u and v, then $|\varphi(u) - \varphi(v)| \geq d(u)$, and as e is not a large edge, it is stretched by a factor greater than c in this embedding. It now only remains to prove that $D(u, \ell(C')) \geq d(u)$. For the sake of convenience, we denote $\ell = \ell(C')$.

We consider three cases. The first case is when the components C and C' are not the ancestor and descendant of one another in the tree of components. Let a be the least common ancestor of u and ℓ, note that $a \neq u, a \neq \ell$. Then $D(u, \ell) = D(a, u) + D(a, \ell)$. However, $D(a, \ell) \geq s(C') \geq c^4 \alpha(C') \geq c^2 d_{e(C')} \geq d(a)$ (we are using the facts that C' is a large component and so $s(C') \geq c^4 \alpha(C')$ and also that $e(C)$ is a large or a medium size edge, and therefore $\alpha(C') = w_{e(C')} \geq \frac{d_{e(C')}}{c^2}$). Thus, $D(u, \ell) \geq D(a, u) + d(a) \geq d(u)$ as desired.

The second case is when C' is a descendant of C in the tree of components. Let $a \in C$ be the least common ancestor of u and ℓ, note that $a = u$ is possible. Then $D(u, \ell) = D(u, a) + D(a, \ell)$. Again, $D(a, \ell) \geq s(C') \geq c^4 \alpha(C') \geq c^2 d_{e(C')} \geq d(a)$ holds, and thus $D(u, \ell) \geq D(a, u) + d(a) \geq d(u)$.

The third case is when C' is an ancestor of C in the component tree. Let $a \in C'$ be the least common ancestor of u and ℓ. Notice first that $D(a, r(C')) < s(C')/2$ must hold, since otherwise $d_{e(C)} \geq D(a, r(C')) \geq s(C')/2 > c^3 \alpha(C) = c^3 w_{e(C)}$, which is impossible since $e(C)$ is a large or a medium size edge. Assume now that $D(a, r(C')) < s(C')/2$ holds. But then $D(a, \ell) \geq s(C')/2 \geq c^3 \alpha(C)$. To finish the proof, observe that $D(u, \ell) = D(a, \ell) + D(a, u) \geq c^3 \alpha(C) + D(u, r(C)) \geq d(r(C)) + D(u, r(C)) \geq d(u)$. ■

Lemma 3.4.34 (Lemma 3.4.27) *If C and C' are large incompatible components, then in the optimal solution they are embedded on different sides of r.*

Proof: Assume C and C' are embedded on the same side of r. As Claim 3.4.33 holds in both directions, the only way for C and C' to be embedded on the same side of r is when $\ell(C)$ is embedded between $r(C')$ and r or when $\ell(C')$ is embedded between $r(C)$ and r.

Assume w.l.o.g. that $\ell(C)$ is embedded between $r(C')$ and r. Since $D(\ell(C), r) \geq s(C) \geq 2c^3\alpha(C')$, vertices $r(C')$ and r are embedded at a distance at least $2c^3\alpha(C')$ from one another. However, $d(r(C')) = \alpha(C') + d_{e(C')} \leq \alpha(C') + c^2\alpha(C') < 2c^2\alpha(C')$ and thus this distance is distorted by more than a factor of c.

■

Combining Large and Small Components

This section is devoted to proving Lemma 3.4.28.

Lemma 3.4.35 (Lemma 3.4.28) *If C is a large component having a conflict with small component C', then C and C' are embedded on different sides of r in the optimal solution.*

Proof: Our proof consists of three claims. In the first claim, we show that if C and C' are embedded on the same side of r, then $r(C')$ is embedded inside some edge e on path $P(C)$. The second claim shows that C' must be a descendant of C in the tree of components. Finally, in the third claim, we show that edge e on path $P(C)$ into which $r(C')$ is embedded is a medium-size edge, whose removal splits C into two large components, therefore e must have been removed by PROCEDURE PARTITION.

Claim 3.4.36 *Assume that C and C' are embedded on the same side of r. Then $r(C')$ is embedded inside some edge e on path $P(C)$.*

Proof: Assume otherwise. Then either $r(C')$ is embedded between r and $r(C)$, or all the vertices on path $P(C)$ are embedded between $r(C')$ and r. If the former case is true, then $|\varphi(r) - \varphi(r(C))| > d(r(C')) \geq \alpha(C') \geq 2c^4\alpha(C)$. But $d(r(C)) = \alpha(C) + d_{e(C)} \leq \alpha(C) + c^2\alpha(C) < 2c^2\alpha(C)$. Thus, the distance between r and $r(C)$ is distorted by a factor greater than c.

If the latter is true, then $|\varphi(r) - \varphi(r(C'))| > s(C) > 2c^4\alpha(C')$. However, this means that the distance between r and $r(C')$ is distorted by a factor greater than c, since $d(r(C')) = \alpha(C') + d_{e(C')} \leq \alpha(C') + c\alpha(C') \leq 2c\alpha(C')$. ■

Let $e=(u,v)$ denote the edge on path $P(C)$, such that $r(C')$ is embedded inside e, and assume w.l.o.g. that $d(u)<d(v)$.

Claim 3.4.37 *C' is a descendant of C in the tree of components.*

Proof: Assume otherwise. There are two cases to consider. If C is the descendant of C', then $d_{e(C)} \geq \alpha(C') \geq 2c^4\alpha(C)$, which is impossible since $e(C)$ is a large or a medium size edge.

The second case is when C and C' are not an ancestor-descendant pair. Let a be the least common ancestor of u and $r(C')$, and notice that $a \notin C'$. We show that $D(u,r(C')) \geq d(u)$, and thus $|\varphi(u)-\varphi(v)| \geq d(u)$ must hold, while $D(u,v) = w_e < d(u)/c$ since e is not large. Therefore, edge e is stretched by a factor greater than c, leading to a contradiction. To see that $D(u,r(C')) \geq d(u)$, Observe that $D(u,r(C')) \geq \alpha(C')+\alpha(C)+D(u,r(C))$. However, $\alpha(C') \geq 2c^4\alpha(C) \geq d_{e(C)}$ (we used the facts that C' and C have a conflict, and also that $e(C)$ is a large or a medium size edge). Therefore, $D(u,r(C')) \geq d(e(C))+\alpha(C)+D(u,r(C)) \geq d(u)$.

■

Claim 3.4.38 *Edge e is of medium size, and upon its removal component C splits into two large components.*

Proof: We first show that e is a medium size edge. Let a be the least common ancestor of $r(C')$ and u. Since C' is a descendant of C, $a \in C$. Then $D(u,r(C')) = D(u,a)+D(a,r(C'))$. However, $D(a,r(C')) \geq \alpha(C') \geq \frac{d(a)}{c}$, since $e(C')$ is a large edge, and a is on the path from $r(C')$ to the root. Altogether, we have that $D(u,r(C')) \geq D(u,a)+\frac{d(a)}{c} \geq \frac{d(u)}{c}$. Since $r(C')$ is embedded between u and v, $|\varphi(u)-\varphi(v)| \geq \frac{d(u)}{c}$, and thus $D(u,v)=w_e \geq \frac{d(u)}{c^2}$ must hold.

Consider now two components C_1, C_2 obtained from C by removing edge e, and assume w.l.o.g. that $r(C) \in C_1$. We show that both these components are large.

Assume for contradiction that C_1 is small. On one hand, since C and C' have a conflict, $2c^4\alpha(C) < \alpha(C')$. On the other hand, since $r(C')$ is embedded inside edge e, and $D(u,r(C')) \geq \alpha(C')$, then $\alpha(C') \leq cw_e$ must hold. Combining the two inequalities together, we have: $2c^3\alpha(C) < w_e$. But since e is not large, $d(u) = d_e > w_e \cdot c > 2c^4\alpha(C)$. Finally, recall that $d(u) \leq D(u, r(C)) + \alpha(C) + c^2\alpha(C)$, and thus $D(u, r(C)) > c^4\alpha(C)$ must hold. But $D(u, r(C)) \leq s(C_1)$, and thus C_1 is a large component.

We now prove that C_2 is a large component. The main part of the proof is showing that $d(u) \leq \left(1 - \frac{1}{c}\right) \frac{s(C)}{c^3}$. Assume that the above bound is true. Then since e is not large, $w_e < \frac{d(u)}{c} \leq \left(1 - \frac{1}{c}\right) \frac{s(C)}{c^4}$. On the other hand, we can show that $s(C_2)$ is sufficiently large relatively to w_e, as follows:

$$s(C_2) \geq s(C) - d(u) - w_e \geq s(C) - \left(1 - \frac{1}{c}\right) \frac{s(C)}{c^3} - \left(1 - \frac{1}{c}\right) \frac{s(C)}{c^4} \geq \left(1 - \frac{1}{c}\right) s(C)$$

Therefore, $s(C_2) \geq c^4 w_e$ holds, and C_2 is a large component.

It now only remains to prove that $d(u) \leq \left(1 - \frac{1}{c}\right) \frac{s(C)}{c^3}$. Recall that $r(C')$ is embedded between u and v, and thus the distance between the embeddings of u and v is at least:

$$D(u, r(C')) + D(v, r(C')) \geq 2D(u, r(C')) = 2[D(u, a) + D(a, r(C'))]$$

As the distortion is at most c,

$$w_e \geq 2\frac{D(u, a) + D(a, r(C'))}{c}$$

must hold. On the other hand, edge e is not large, and thus

$$w_e < \frac{d(u)}{c} = \frac{d(a) + D(u, a)}{c}$$

Combining the two inequalities together, we get:

$$d(a) \geq D(u, a) + 2D(a, r(C')) \geq D(u, a) + 2\alpha(C')$$

Since a is on the path from $r(C')$ to r and $e(C')$ is a large edge, $\alpha(C') \geq \frac{d(a)}{c}$. We thus have: $d(a)\left(1-\frac{2}{c}\right) \geq D(u,a)$.

Altogether,

$$d(u) = d(a) + D(u,a) \leq d(a)\left(2-\frac{2}{c}\right) \leq c\alpha(C')\left(2-\frac{2}{c}\right) \leq \frac{s(C)}{c^3}\left(1-\frac{1}{c}\right)$$

■

■

Analysis of the Algorithm

We start with the following simple observation.

Observation 3.4.39 *Let C be any component, and let r be the root of the current instance. Then $D(r(C),r) \leq 2c^2\alpha(C)$.*

Proof: It is easy to see that $D(r(C),r) = \alpha(C) + d_{e(C)}$. However, since $e(C)$ is a large or a medium size edge, $\alpha(C) \geq \frac{d_{e(C)}}{c^2}$. In total, $D(r(C),r) \leq \alpha(C)+c^2\alpha(C) \leq 2c^2\alpha(C)$.

■

We now bound the empty space we need to leave between each pair of components that are embedded next to each other. Consider some component C embedded to the left of r. Recall that in the recursive procedure call for C, we use threshold value $H = D(r(C),r)$ for the root condition. Let $v \in C$ be the rightmost vertex in the embedding of C.

We want to show $D(v,r)$ is "small".Assume w.l.o.g. that $v \neq r(C)$. Let C' be the component, obtained by the decomposition of C, that contains v. Note that due to Observation 3.4.39, $D(r(C'),r(C)) \leq 2c^2\alpha(C')$. Since v (and therefore C') lies on the right side of $r(C)$, it must satisfy the threshold condition $\alpha(C') + s(C') \leq cH = cD(r(C),r)$. We can now write

$$
\begin{aligned}
D(v,r) &\leq D(r(C),r) + [D(v,r(C')) + D(r(C'),r(C))] \\
&\leq D(r(C),r) + [s(C') + 2c^2\alpha(C')] \\
&\leq D(r(C),r) + 2c^3 H \\
&\leq 3c^3 D(r(C),r) \\
&\leq 6c^5\alpha(C)
\end{aligned}
$$

For each component C embedded to the left of r, we leave empty space of $6c^5\alpha(C)$ to the right of the embedding of C, and empty space of $s(C) + D(r,r(C)) \leq s(C) + 2c^2\alpha(C)$ to the left of the embedding of C, such that empty spaces belonging to different components do not overlap. The embedding of components in $\mathcal{R}$ is performed similarly. It is easy to see that the resulting embedding is non-contracting.

Consider now some component C. Let $\mathcal{L}(C), \mathcal{S}(C)$ denote the sets of large and small components, respectively, embedded between C and r by our algorithm. We define $L(C) = \sum_{C' \in \mathcal{L}(C)} s(C')$, and $S(C) = \sum_{C' \in \mathcal{S}(C)} \alpha(C')$. In order to bound the approximation ratio of our algorithm, it is helpful to bound first the values $L(C)$ and $S(C)$ in terms of $\alpha(C)$.

Lemma 3.4.40 *For any component C, $L(C) \leq 4c^4\alpha(C)$, and $S(C) \leq 24c^8\alpha(C)$.*

Proof:

We start by bounding $L(C)$. Consider any pair C_1, C_2 of large components, embedded on the same side of r. These components are compatible, and thus we can assume w.l.o.g. that $s(C_1) \leq 2c^3\alpha(C_2)$. However, since C_2 is large, $\alpha(C_2) \leq s(C_2)/c^4$, and therefore $s(C_1) \leq 2s(C_2)/c$, and C_1 is embedded closer than C_2 to the root.

Assume now that C is a large component, and let $C' \in \mathcal{L}(C)$ be the component that maximizes $s(C')$. Then $s(C') \leq 2c^3\alpha(C)$ (since otherwise C must be embedded closer to r than C'). Moreover, the values of $s(C'')$ for $C'' \in \mathcal{L}(C)$ constitute a geometric series with ratio $\frac{2}{c}$. Therefore, $L(C) \leq 4c^3\alpha(C)$.

If C is a small component, let $C' \in \mathcal{L}(C)$ be the component that maximizes $s(C')$. Due to the ordering of the components by our algorithm, $\frac{s(C')}{2c^4} \leq \alpha(C)$. The values of $s(C'')$ for $C'' \in \mathcal{L}(C)$ again form a geometric series, and thus $L(C) \leq 4c^4\alpha(C)$.

We now proceed to bound $S(C)$. Recall that there are at most $4c^4$ small components of each type. Assume first that C is a small component of type i. Then $S(C)$ contains at most $4c^4$ components of the same type (whose α is less than $\alpha(C)$), and at most $4c^4$ components for each one of the types $1, \ldots, i-1$. Thus, $S(C) \leq 12c^4\alpha(C)$.

Suppose now that C is a large component, and let $C' \in \mathcal{S}(C)$ be the component maximizing $\alpha(C')$. Then $\alpha(C') \leq \frac{s(C)}{2c^4}$. Since there is no conflict between C and C', $\alpha(C') \leq 2c^4\alpha(C)$ must hold. Again, we have at most $4c^4$ components of the same type as C', whose α-value is not greater than $\alpha(C')$, and at most $4c^4$ components of each one of the smaller types. Therefore, $S(C) \leq 12c^4 \cdot 2c^4\alpha(C) \leq 24c^8\alpha(C)$. ■

Definition 14 *Given a component C, its weight $W(C)$ is defined to be the sum of weights of its edges.*

Claim 3.4.41 $W(C) \leq 2cs(C)$

Proof: The length of any embedding of C is at least $W(C)$, while the maximum distance between any pair of points in C is $2s(C)$. Since the distortion of the optimal embedding is c, the claim holds. ■

The next theorem is the central theorem in the analysis of our algorithm.

Theorem 3.4.42 *Let C be the instance of our problem with threshold H for the root condition. Then the algorithm produces an embedding with the following properties:*

- *The length of the embedding is at most $c^{13}W(C)$.*
- *The length of the embedding to the right of the root r is at most $c^{28}H$.*
- *For any vertex $v \in C$, v is embedded within distance $c^{29}D(v,r)$ from r.*

Proof:

The proof is by induction on the instance size. Let $\mathcal{C}$ be the collection of components produced by our algorithm. We assume that the claim holds for each $C' \in \mathcal{C}$ and the corresponding threshold value, and prove it for C.

We start by bounding the embedding length. We first bound the length of the embedding to the left of r. Let C_L be the leftmost component embedded to the left of r (if such a component exists). The length of the embedding to the left of r consists of the following parts: (1) the lengths of the embeddings of all the components in $\mathcal{L}$: they are bounded by $c^{13} \sum_{C' \in \mathcal{L}} W(C')$ by the inductive hypothesis; (2) the additional space we need to leave between the components to ensure non-contraction.

We show that this additional space is at most $c^{13}\alpha(C_L)$. Observe that edge $e(C_L)$ does not participate in any of the recursive algorithm executions. Since we can bound the length of the embedding to the right of r in a similar fashion, this will finish the proof that the total length of the embedding is at most $c^{13}W(C)$.

We now bound the additional space we need to place between the components. Let $C' \in \mathcal{L} \setminus \{C_L\}$ be some large component. The empty space we need to leave due to C' is at most $2[s(C')+D(r(C'),r)] \leq 2[s(C')+2c^2\alpha(C')] \leq 3s(C')$ (since C' is large). Thus, in total, the large components in $\mathcal{L} \setminus \{C_L\}$ contribute at most $3L(C_L) \leq 12c^4\alpha(C_L)$. Consider now some small component $C' \in \mathcal{L} \setminus \{C_L\}$. The empty space due to C' is again bounded by $2[s(C') + D(r(C'),r)] \leq 2[s(C') + 2c^2\alpha(C')]$. However, since C' is small, $s(C') \leq c^4\alpha(C')$, and thus its contribution is at most $3c^4\alpha(C')$. In total, small components in $\mathcal{L} \setminus \{C_L\}$ contribute at most $3c^4S(C_L) \leq 72c^{12}\alpha(C_L)$. Finally, component C_L itself contributes at most $6c^5\alpha(C_L)$. The total additional empty space is thus at most:

$$12c^4\alpha(C_L) + 72c^{12}\alpha(C_L) + 6c^5\alpha(C_L) \leq c^{13}\alpha(C_L)$$

We now prove the second part of the theorem.

Let C_R be the rightmost component in our embedding. From the root condition, $\alpha(C_R) + s(C_R) \leq cH$. If C' is a large component embedded between C_R and r, then its embedding length is at most $c^{13}W(C') \leq 2c^{14}s(C')$. The amount of empty space we need to leave out for this component is at most $2[s(C') + D(r(C'), r)] \leq 2[s(C') + 2c^2\alpha(C')] \leq 3s(C')$. Thus, the total contribution of such components is at most $6c^{14}L(C_R) \leq 3c^{14} \cdot 4c^4\alpha(C_R) = 12c^{18}\alpha(C_R)$.

Similarly, the length of the embedding of a small component C' is at most $2c^{14}s(C') \leq 2c^{18}s(C')$, and the amount of free space we need to add due to C' is bounded by $2[s(C') + D(r(C'), r)] \leq 2[s(C') + 2c^2\alpha(C')] \leq 3c^4\alpha(C')$.The total contribution of small components is at most $3c^{18}S(C_R) \leq 3c^{18} \cdot 24c^8\alpha(C_R) \leq 72c^{26}\alpha(C_R)$. Finally, the length of the embedding of C_R is at most $2c^{14}s(C_R)$, and the empty space we need to leave to the left of it is at most $6c^5\alpha(C_R)$. The total size of the embedding to the right of r is at most:

$$12c^{18}\alpha(C_R) + 72c^{26}\alpha(C_R) + 6c^5\alpha(C_R) + 2c^{14}s(C_R) \leq c^{27}(\alpha(C_R) + s(C_R)) \leq c^{28}H$$

Finally, we prove the third part of the theorem. Consider some vertex v, belonging to some component C'. Let ψ be the embedding computed by the algorithm. Then $|\psi(v) - \psi(r)| \leq |\psi(v) - \psi(r(C'))| + |\psi(r) - \psi(r(C'))|$, while $D(v, r) = D(v, r(C')) + D(r, r(C'))$. By the inductive hypothesis, $|\psi(v) - \psi(r(C'))| \leq c^{30}D(v, r(C'))$. We now prove that $|\psi(r) - \psi(r(C'))| \leq c^{30}D(r, r(C'))$, thus finishing the proof.

The distance between the embeddings of $r(C')$ and r consists of three parts: (1) The length of the recursive embedding of component C' to the right of its root $r(C')$: bounded by $c^{28}D(r, r(C'))$ by the induction hypothesis; the empty space we need to leave between the embedding of C' and its neighbor that lies between C' and r: bounded by $6c^5\alpha(C')$; (3) the embeddings of all the components lying between C' and the root r, and their empty spaces. The last term can be bounded by the similar way we used to bound the distance between the embedding of C_R and the root, which is at most $c^{27}\alpha(C_R)$. Summing the three terms together, we get:

$$|\psi(r) - \psi(r(C'))| \leq c^{28}D(r, r(C')) + 6c^5\alpha(C) + c^{27}\alpha(C_R) \leq c^{29}D(r, r(C'))$$

■

Theorem 3.4.43 *(Theorem 3.4.32) The algorithm produces a non-contracting embedding with distortion bounded by $c^{O(1)}$.*

Proof: It is easy to see that the embedding produced by the algorithm is non-contracting. We now prove that the distortion is at most $4c^{32}$. Let $e = (u, v)$ be some edge in our original instance. Let C be the first instance in our recursive algorithm executions, where u and v are separated: i.e., $u, v \in C$, but there are two components $C_u, C_v \subseteq C$, such that $u \in C_u$, $v \in C_v$. Let r denote the root of the current instance.

Then edge e is a large or a medium-size edge, and thus $D(u, v) = w_e \geq \frac{d(u)}{c^2}$. Also, since $d(v) = d(u) + w_e \leq c^2 w_e + w_e \leq 2c^2 w_e$, we have that in total:

$$D(u, v) = w_e \geq \frac{d(u) + d(v)}{4c^2}$$

On the other hand, consider the embedding ψ produced by our algorithm. Then:

$$\begin{aligned} |\psi(u) - \psi(v)| &\leq |\psi(u) - \psi(r)| + |\psi(v) - \psi(r)| \\ &\leq c^{30}(d(u) + d(v)) \\ &\leq 4c^{32} \frac{d(u) + d(v)}{4c^2} \\ &\leq 4c^{32} w_e \end{aligned}$$

■

3.5 Embedding Ultrametrics Into Low-Dimensional Spaces

Credits: The results in this section is work done with Julia Chuzhoy, Piotr Indyk, and Anastasios Sidiropoulos, and has appeared in SoCG'06.

3.5.1 Introduction

In this section we consider embedding *ultrametrics* into the plane. Ultrametrics are a natural class of metrics, frequently occurring in applications involving hierarchical clustering. They are of particular interest in biology, where they can be used to represent evolutionary trees (cf. [FK99] or [DEKM98], p. 168). Visualizing such trees requires embedding them into the plane, which is exactly the problem we consider in this paper.

Our main result is an algorithm which receives as input an ultrametric and outputs its embedding into the plane. If the input ultrametric embeds into the plane with distortion c (under l_p norm for any $1 \leq p \leq \infty$[4]), then the embedding produced by the algorithm has distortion $O(c^3)$. In particular, for the case where the input ultrametric is embeddable into the plane with constant distortion, the distortion of the embedding produced by the algorithm is also constant. The running time of our algorithm is linear in the input size, assuming it is given the value of the optimum distortion c (or its approximation). The algorithm generalizes to embeddings into $\Re^d$ (equipped with the l_2 norm), and the distortion becomes $c^{O(d)}$, where c is the distortion of the optimal embedding of the ultrametric into $\Re^d$.

In our second result we prove that any ultrametric can be embedded into the plane with distortion $O(\sqrt{n})$. More generally, for any $d \geq 2$, we show how to embed any ultrametric into $\Re^d$ with distortion $d^{O(1)} n^{1/d}$. Notice that unlike the first result, this result relates to the absolute version of the distortion minimization problem. The proof is algorithmic - the embedding can be found in polynomial time. Combining the two results together, we obtain an $O(n^{1/3})$-approximation algorithm for embedding ultrametrics into the plane.

We also remark that for the case of embedding ultrametrics into low-dimensional spaces, it has been shown (cf. [BM04b]) that for any $\epsilon > 0$, any ultrametric can be embedded into $\ell_p^{O(\epsilon^{-2} \log n)}$, with distortion $1 + \epsilon$.

Finally, we investigate the hardness of embedding ultrametrics into the plane. We

[4]The algorithm is described for the case of the l_2 norm. However, since l_p norms for all $1 \leq p \leq \infty$ in $\Re^2$ are equivalent up to a factor of 2, the algorithm works for any l_p norm as well.

prove that the problem of finding the smallest-distortion embedding is strongly NP-hard, if the distance is measured according to the l_∞ norm. Interestingly, the problem of minimizing the distortion of embedding *into* ultrametrics can be solved exactly in polynomial time [ABD+05].

Our techniques

We use the well-known fact that any ultrametric $M = (X, D)$ can be well approximated by hierarchically well-separated trees (HST's) (see Section 3.5.2 for definitions). The corresponding HST T has the points of X as its leaves, and each vertex v of T has a label $l(v) \in \Re^+$. The distance of any pair of points $p, q \in X$ is exactly the label of their nearest common ancestor.

The hierarchical structure of HST's naturally enables constructing the embedding in a recursive manner. That is, the mapping is constructed by embedding (recursively and independently) the children of the root node, and then combining the embeddings. Implementing this idea, however, requires overcoming a few obstacles, which we discuss now. For simplicity, we focus on embeddings into the plane.

Distortion lower bound. The first issue is how to obtain a good lower bound for the distortion. It is not difficult to see that the distortion depends on both the number of nodes, and the structure of the ultrametric. For example, the full 2-HST of depth t, where all internal nodes have degree 4, requires $\Omega(t)$ distortion; at the same time, the full 4-HST of depth t, where all internal nodes have degree 4, can be embedded with constant distortion.

Our lower bound is obtained as follows. Consider any node v and its children $u_1 \dots u_k$. Let P_i be the set of leaves in the subtree of the node u_i, $P = P_1 \cup \dots \cup P_k$. By the definition of ultrametrics, the distances between any pair of points $p \in P_i$ and $q \in P_j$ for $i \neq j$, are equal to the same value, namely $l(v)$. Consider any non-contracting embedding $f : P \to \Re^2$. Construct a ball of radius $l(v)/2$ around each point $f(p)$, $p \in P$, and denote this ball by $B(p, l(v)/2)$. It is easy to see that the union of the interiors of the balls around points in P_i and the union of the interiors of the balls around points in P_j must be disjoint if $i \neq j$.

Our lower bound on distortion proceeds by estimating the total volume $C(v)$ of $\cup_{p \in P} B(p, l(v)/2)$. Specifically, by packing argument, one can observe that the distortion of the optimal embedding must be at least $\Omega(\sqrt{C(v)} - O(1))$. Thus, it suffices to have a good lower bound for the volume $C(v)$. It would appear that such lower bounds could be obtained by summing $C(u_i)$'s, since the balls around different sets P_i are disjoint. Unfortunately, $C(u_i)$ is the volume of the union of the balls of radius $l(u_i)/2$, not $l(v)/2$, so the above is not strictly true. However, $\cup_{p \in P_i} B(p, l(v)/2)$ can be expressed as a Minkowski sum of $\cup_{p \in P_i} B(p, l(u_i)/2)$ and a ball of radius $[l(v) - l(u_i)]/2$. Then the volume of that set can be bounded from below by using Brunn-Minkowski inequality, by a function of $C(u_i)$ and $l(v) - l(u_i)$. This enables us to obtain a recursive formula for $C(v)$ as a function of $C(u_i)$'s.

Distortion accumulation. The recursive formula for the lower bound suggests a recursive algorithm. Consider some vertex v of the HST, and let $u_1, \ldots, u_k$ be its children. For each u_i, $1 \leq i \leq k$, the leaves in the subtree of u_i are mapped into a square $R(u_i)$ whose volume is at most $C(u_i)$. Then the squares are re-arranged to form a square $R(v)$. The main difficulty with this approach is that the optimal way to pack the squares is difficult to find. In fact, the optimal embedding could, in principle, not pack the points into squares. To overcome this problem, we allow some limited stretching of the squares, to fit them into $R(v)$. However, stretching causes distortion, and thus we need to make sure that stretching done over different levels does not accumulate. In order to avoid such accumulation of distortion, we alternate between the horizontal and vertical stretchings of the squares. Specifically, we assign, for each vertex v of the HST, a bit $g(v)$ that determines whether the squares into which the sub-trees of the children of v are embedded will be stretched in the horizontal or the vertical direction before they are packed into the square $R(v)$. We calculate the values of the bits $g(v)$ in a top-down manner, starting with the leaves of the HST, to ensure that the final stretchings are balanced.

It appears that the need to compute a proper choice of stretching directions (which can also be viewed as rotations) at each level is not just an artifact of our algorithm, but it might be necessary to achieve low distortion. In particular, the only constant distortion embedding of a full 2-HST into the plane that we are aware of uses alternating rotations.

Higher dimensions. We show how to generalize the algorithm for embedding ultrametrics into the plane to higher dimensions. We show an algorithm that produces a $c^{O(d)}$-distortion embedding of the input ultrametric into $\Re^d$ under the l_2 norm, where c denotes the optimal distortion achievable when embedding the input ultrametric into $\Re^d$.

Hardness. We show NP-hardness of the embedding problem for the case of the plane under l_∞ norm. We use a reduction from the square packing problem. Since our algorithm also uses (a variant of) square packing, the packing problem appears to be intimately related to embeddings of ultrametrics.

3.5.2 Preliminaries and Definitions

A metric $M = (X, D)$ is an *ultrametric* if it can be represented by a *labeled tree* T whose set of leaves is X, in the following manner. Each non-leaf vertex v of T has a label $l(v) > 0$. If u is a child of v in tree T, then $l(u) \leq l(v)$. For any $x, y \in X$, the distance between x and y is defined to be the label of the nearest common ancestor of x and y, and this distance should be equal to $D(x, y)$.

We now proceed to define hierarchically well-separated trees (HST's). For any $\alpha \geq 1$, an α-HST is an ultrametric where for each parent-child pair of vertices (u, v), $l(u) = \alpha l(v)$. It is easy to see that for any $\alpha \geq 1$, any ultrametric can be α-approximated by an α-HST (cf. [Bar96]). Moreover, such an HST can be found in time linear in the input size. Therefore, if the input ultrametric M embeds into $\Re^d$ with distortion c, then the metric M' defined by the corresponding 2-HST embeds into $\Re^d$ with distortion $c' = 2c$. Any non-contracting embedding of M' into $\Re^d$ with distortion c'' represents a non-contracting embedding of M with distortion $O(c'')$. Therefore, from now on we will concentrate on embeddings of HST's into $\Re^d$.

Given a 2-HST T, we will use the following additional notation. Let r denote the root of the tree, and let h denote the tree height. We assume that r belongs to the first level of T, and all the leaves belong to level h. By scaling the underlying metric M, we can assume w.l.o.g., that for each vertex v at level $h-1$, $l(v) = 2$. For any non-leaf vertex v, we denote by X_v the set of leaves of the subtree of T rooted at v, and we denote the number of leaves in the subtree n_v.

We will use the Brunn-Minkowski inequality, defined as follows. Given any two sets $A, B \subseteq \mathbb{R}^d$, let $A \oplus B$ denote the Minkowski sum of A and B, i.e., $A \oplus B = \{a + b \mid a \in A, b \in B\}$.

Theorem 3.5.1 (Brunn-Minkowski inequality) *For any pair of sets $A, B \subseteq \mathbb{R}^d$,*

$$Vol(A \oplus B)^{1/d} \geq Vol(A)^{1/d} + Vol(B)^{1/d}.$$

3.5.3 A Lower Bound on the Distortion of Optimal Embedding

In this section we show a lower bound on the distortion of optimal embedding of a metric M' which is defined by a 2-HST denoted by T.

For any $r > 0$, let $B(r)$ denote the ball of radius r in ℓ_2^d centered at the origin. Let $V_d(r)$ denote the volume of a d-dimensional ball of radius r, $V_d(r) = \frac{\pi^{d/2} r^d}{\Gamma(1+d/2)}$. For each vertex v of T, we define a value $C(v)$, which intuitively is a lower-bound on the minimum volume embedding of X_v (the precise statement appears below). The values $C(v)$ are defined recursively, starting from the leaves. For each leaf v, we set $C(v) = V_d(1/2)$.

Consider now vertex v at level $j \in [h-1]$, and let $u_1, \ldots, u_k$ be the children of v in T. We define:

$$C(v) = \sum_{i=1}^{k} \left((C(u_i))^{1/d} + (V_d(l(v)/4))^{1/d} \right)^d$$

Given any embedding $\varphi : X \rightarrow \ell_2^d$, for any subset $X' \subseteq X$, let $\varphi(X')$ denote the image of points in X' under φ.

Lemma 3.5.2 *Let v be a non-leaf vertex of T, and let φ be any non-contracting embedding of X_v into ℓ_2^d. Then the volume of $\varphi(X_v) \oplus B\left(\frac{l(v)}{2}\right)$ is at least $C(v)$.*

Proof: Let $u_1, \ldots, u_k$ be the children of v. The proof is by induction. Assume first that v belongs to level $h-1$ of T, and consider $S = \varphi(X_v) \oplus B(l(v)/2)$. Recall that $l(v) = 2$. Since the embedding is non-contracting, for any $1 \leq i < j \leq k$, vertices u_i, u_j are embedded at a distance at least 2 from each other. Therefore, set S consists of k balls of disjoint interiors, of radius 1 each, and thus the volume of S is exactly $kV_d(1) = C(v)$.

Assume now that v belongs to some level $j \in [h-2]$. Let $S = \varphi(X_v) \oplus B(l(v)/2)$. Equivalently, S is the union of $S_i = \varphi(X_{u_i}) \oplus B(l(v)/2)$ for $i \in [k]$. Since the embedding is non-contracting, all the sets S_i have disjoint interiors. For each $i \in [k]$, let us denote $S'_i = \varphi(X_{u_i}) \oplus B(l(u_i)/2)$. Recall that $l(v) = 2l(u_i)$. Therefore, for each $i \in [k]$, $S_i = S'_i \oplus B(l(v)/4)$. Using the induction hypothesis, the volume of S'_i is at least $C(u_i)$. From the Brunn-Minkowski inequality, it follows that:

$$\begin{aligned}(\mathrm{Vol}(S_i))^{1/d} &\geq (\mathrm{Vol}(S'_i))^{1/d} + (V_d(l(v)/4))^{1/d} \\ &\geq (C(u_i))^{1/d} + (V_d(l(v)/4))^{1/d}\end{aligned}$$

Therefore, in total,

$$\begin{aligned}\mathrm{Vol}(S) = \sum_{i=1}^{k} \mathrm{Vol}(S_i) &\geq \sum_{i=1}^{k} \left((C(u_i))^{1/d} + (V_d(l(v)/4))^{1/d}\right)^d \\ &= C(v).\end{aligned}$$

■

Suppose we are given some set of points $S \subseteq \Re^d$, that has volume V. We define $\rho_d(V) = \left(\frac{V \cdot \Gamma(1+d/2)}{\pi^{d/2}}\right)^{1/d}$, i.e., $\rho_d(V)$ is the radius of the d-dimensional ball in $\Re^d$ that has volume V. Observe that S has two points at a distance at least $\rho_d(V)$ from each other (otherwise, S is contained in a ball of radius smaller than $\rho_d(V)$, which is impossible).

Corollary 3.5.3 *Let v be some non-leaf vertex of T, and let φ be any non-contracting* *embedding of M' into ℓ_2^d, with distortion at most c'. Then $c' \geq \rho_d(C(v))/l(v) - 1$.*

Proof: Consider $S = \varphi(X_v) \oplus B\left(l(v)/2\right)$. By Lemma 3.5.2, the volume of S is at least $C(v)$, and thus there are two points $x, y \in S$ within a distance at least $\rho = \rho_d(C(v))$ from each other. By the definition of S, it follows that there are two points $a, b \in X_v$, which are embedded at a distance of at least $\rho - l(v)$ from each other. As the distance between a, b in T is at most $l(v)$, the bound on the distortion follows. ■

3.5.4 Approximation Algorithm for Embedding Ultrametrics Into the Plane

Preliminaries and Intuition

Let $M = (X, D)$ be the input ultrametric that embeds into the plane with distortion c. Let $M' = (X, D')$ be the metric defined by the 2-HST T which 2-approximates M. Then M' embeds into the plane with distortion $c' \leq 2c$, and any non-contracting embedding of M' into the plane with distortion $O(c'^3)$ is also a non-contracting embedding of M with distortion at most $O(c^3)$. Therefore, from now on we concentrate on embedding M' into the plane.

Consider some non-leaf vertex u. We define $a_u = \sqrt{C(u)}$. If $u \neq r$, let v be its father. We define $b_u = a_u + \frac{\sqrt{\pi} l(v)}{4}$.

Our algorithm works in bottom-up fashion. Let v be some vertex. The goal of the algorithm is to embed all the vertices of X_v into a square Q of side a_v, incurring only small distortion. Let $u_1, \ldots, u_k$ be the children of v, and assume that for all $j : 1 \leq j \leq k$, we have already embedded $X(u_j)$ inside a square Q_j of side a_{u_j}. Recall that for any pair of vertices $x \in X_{u_j}$, $y \in X_{u_{j'}}$, where $1 \leq j \neq j' \leq k$, the distance between x and y in T is $l(v)$. Our first step is to ensure non-contraction (or more precisely small contraction), by adding empty strips of width $\frac{b_{u_j} - a_{u_j}}{2} = \frac{\sqrt{\pi} l(v)}{8}$ around the squares. Thus, we obtain a collection $Q'_1, \ldots, Q'_k$ of squares, of sides $b_{u_1}, \ldots, b_{u_k}$, respectively. Our goal now is to pack these squares into one large square Q of side a_v. Observe that from volume view point, $\mathrm{Vol}(Q) = \mathrm{Vol}(Q'_1) + \ldots + \mathrm{Vol}(Q'_k)$, since $a_v^2 = \sum_{j=1}^{k} b_{u_j}^2$, by the definition of C_v. However, it is not always possible to obtain such tight packing of squares. Instead, we convert each square Q'_j to rectangle R_j whose sides are $b_{u_j}s_{u_j}$, b_{u_j}/s_{u_j} for some $s_{u_j} = O(c')$. Observe that the volume of R_j is the same as that of Q'_j. This will enable us to pack all the rectangles $R_1, \ldots, R_k$ into Q. Recall that inside each square Q'_j, vertices of X_{u_j} are embedded. In order to convert square Q'_j into rectangle R_j, we contract all the distances along one axis, and expand all the distances along the other axis, by the same factor s_{u_j}.

Consider now two vertices u, v, and let z be their least common ancestor. The distance between u and v might thus be contracted or expanded when we calculate the embedding of X_z. However, for each vertex z' on the path from z to r, the distance between u and v might be contracted or expanded again, when calculating the embedding of $X_{z'}$. In order to avoid accumulation of distortion, we would like to alternate the contractions and expansions of this distance in an appropriate way. To this end, we calculate, for each vertex v, a value $g(v) \in \{-1, 1\}$. Let $u_1, \ldots, u_k$ be the children of v, and let $Q'_1, \ldots, Q'_k$ be their corresponding squares. If $g(v) = 1$, then when embedding squares $Q'_1, \ldots, Q'_k$ into square Q of side a_v, we expand them along axis x and contract along axis y. If $g(v) = -1$, we do the opposite. The values of $g(v)$ have to be computed in a top-bottom fashion. They are calculated in such a way that the total distortion of distance between any pair of points in X stays below $poly(c')$.

For any non-root vertex u in T, with parent a vertex v, we define $s_u = a_v/b_u$. Also, for the root r of T, let $s_r = 1$.

Lemma 3.5.4 *For each vertex u, $1 \leq s_u \leq 32c'$.*

Proof: If u is the root, then $s_u = 1$. Otherwise, let $u, v \in T$, such that v is the father of u. We have already observed that a_v^2 is the sum of $b_{u_j}^2$, for all children u_j of v. Thus, $s(u) \geq 1$ holds.

Recall now that by the definition of b_u, its value is at least $\frac{l(v)}{4}$. On the other hand, by Corollary 3.5.3, $c' \geq \frac{a_v}{l(v)\sqrt{\pi}} - 1$, and thus $a_v \leq (c'+1)\sqrt{\pi}l(v) \leq 8c'l(v)$. Therefore, $s_u = \frac{a_v}{b_u} \leq 32c'$. ■

Let v be some non-leaf vertex, and let $u_1, \ldots, u_k$ be its children. Let $Q'_1, \ldots, Q'_k$ be the squares of side $b_{u_1}, \ldots, b_{u_k}$, respectively, corresponding to the children. In order to pack these squares into a square of side a_v, we transform each square Q'_j into a rectangle with sides $b_{u_j} s_j, \frac{b_{u_j}}{s_j}$. The goal of the next lemma is to calculate the values $g(v) \in \{-1, 1\}$ for each $v \in V$, that will determine, along which axis we contract, and along which expand when embedding the subtree of v.

Suppose we have a function $g : V(T) \to \{-1, 1\}$. Consider some vertex $v \in V(T)$, and let $v_1, v_2, \ldots, v_k$ be the vertices on the path from v to r, where $v_1 = r$, $v_k = v$. We define $h(v) = \prod_{j=1}^{k-1} s_{v_{j+1}}^{g(v_j)}$.

Lemma 3.5.5 *We can calculate, in linear time, function $g : V(T) \to \{-1, 1\}$, such that for each $v \in V(T)$, $\frac{1}{32c'} \leq h(v) \leq 32c'$.*

Proof: Observe first that in order to be able to calculate $h(v)$ for any $v \in V$, it is enough to know the values of $g(v')$ of all the vertices v' on the path from r to v, not including v.

We traverse the tree in the top-bottom fashion. For root r, we set $g(r) = 1$. Since for all the values s_v, $1 \leq s_v \leq 32c'$ holds, we have that for each level-2 vertex v, $\frac{1}{32c'} \leq h(v) \leq 32c'$ holds, as required.

Consider now some vertex $v \in V$ at level k, where $k \geq 2$. Let $v_1, v_2, \ldots, v_k$ be the vertices on the path from r to v, where $v_1 = r$, and $v_k = v$, and assume we have calculated $g(v_1), \ldots, g(v_{k-1})$, such that for each $j : 2 \leq j \leq k$, $\frac{1}{32c'} \leq h(v_j) \leq 32c'$ holds. We set $g(v) = 1$ if $h(v_k) \leq 1$, and we set $g(v) = -1$ otherwise. Let u be a child of v. Since $h(u) = h_v \cdot s_u^{g(v)}$, and $s_u \leq 32c'$, the inequality $\frac{1}{32c'} \leq h(u) \leq 32c'$ holds.

It is easy to see that the running time of the above algorithm is linear, if the values $h(v)$ of the vertices calculated by the algorithm are stored in a table. The algorithm traverses each vertex only once, and for each vertex v the calculation of $h(v)$ and $g(v)$ takes only constant time. ■

Algorithm Description

The algorithm consists of two phases. The first phase is pre-processing, and the second phase is computing the embedding itself.

Phase 1: Preprocessing. In this phase we translate the input ultrametric M into a 2-HST T, and calculate the values a_v, b_v, s_v, $g(v)$ for each vertex $v \in T$. Each one of these operations takes time linear in the input size.

Phase 2: Computing the embedding. The algorithm works in a bottom-up fashion. For any vertex v in tree T, we produce an embedding of vertices X_v inside a square of side a_v. We start from level-h vertices (the leaves). Let v be such vertex. Then $a_v = \sqrt{C(v)} = \sqrt{\pi/4}$. We embed this point in the center of a square with a side of length $\sqrt{\pi/4}$.

Consider some level-i vertex v, for $1 \leq i < h$, and let $u_1, \ldots, u_k$ be its children. We assume that for each $j : 1 \leq j \leq k$, we have calculated the embeddings of u_j into a square Q_j of side a_{u_j}. We convert this square into a rectangle R_j, as follows. First, we add an empty strip of width $\frac{\sqrt{\pi}l(v)}{8}$ along the border of Q_j, so that now we have a new square Q'_j of side b_{u_j}. If $g(v) = 1$, then we expand the square along axis x and contract it along axis y by the factor of s_{u_j}. Otherwise, we expand square Q'_j along axis y and contract it along axis x by the factor of s_{u_j}. Notice that by the definition of s_{u_j}, the length of the longer side of R_j is precisely a_v. As the volume of R_j equals to the volume of Q'_j, and since $a_v^2 = \sum_{j=1}^{k} b_{u_j}^2$, we can pack all the rectangles next to each other inside a square Q of side a_v, with their longer side parallel to the x-axis if $g(v) = 1$, and to y-axis otherwise.

Analysis

The goal of this section is to bound the distortion produced by the algorithm. We first bound the maximum contraction, and then the maximum expansion of distances.

Lemma 3.5.6 *For any $u, u' \in X$, the distance between the images of u and u', is at least $\Omega(1/c')D(u, u')$.*

Proof: Let v be the least common ancestor of u, u'.

Let z, z' be the children of v, to whose subtrees vertices u, u' belong, respectively. Let Q, Q' be the squares into which X_z, and $X_{z'}$ are embedded, respectively, and let R, R' be the corresponding rectangles. Recall that we have added a strip of width at least $\frac{\sqrt{\pi}l(v)}{4}$ to squares Q, Q', and then stretched the new squares by a factors of $s(z), s(z')$, respectively. Without loss of generality, we can assume $s(z) \geq s(z')$. Therefore, immediately after computing the embedding for X_v, there is a strip S of width at least $\frac{l(v)}{4s(z)}$ between the rectangles R, R'. The width of strip S in the final embedding is a lower bound on the distance between the images of u and u'. Let $v_1, \ldots, v_k$ be the vertices on the path from r to v, where $v_1 = r, v_k = v$. Let $u_{k+1} = z$. If $g(v) = 1$, then strip S is horizontal, and thus for each $j : 1 \leq j \leq k-1$, if $g(v_j) = 1$ then its width decreases by the factor of $s(v_{j+1})$, and if $g(v_j) = -1$ then its width increases by the same factor. Thus, the final width of S is at least: $\frac{l(v)}{4s(z)^{g(v)}} \prod_{j=1}^{k-1} s(v_{j+1})^{-g(v_j)} = \frac{l(v)}{4} \prod_{j=1}^{k} s(v_{j+1})^{-g(v_j)} \geq \frac{l(v)}{4h(z)} \geq \frac{l(v)}{128c'}$.

If $g(v) = -1$, then strip S is vertical, and thus for each $j : 1 \leq j \leq k-1$, whenever $g(v_j) = 1$, the width of the strip grows by the factor of $s(v_{j+1})$, and whenever $g(v_j) = -1$, this width decreases by the same factor. Thus, in this case, the final width of S is at least: $\frac{l(v)}{4} s(z)^{g(v)} \prod_{j=1}^{k-1} s(v_{j+1})^{g(v_j)} = \frac{l(v)}{4} \prod_{j=1}^{k} s(v_{j+1})^{g(v_j)} \geq \frac{l(v)}{128c'}$.

As $D(u, u') = l(v)$, this concludes the proof of the lemma. ∎

Lemma 3.5.7 *For any $u, u' \in X$, the distance between the images of u and u', is at most $O(c'^2)D(u, u')$.*

Proof: Let v be the least common ancestor of u, u'. Then $D(u, u') = l(v)$. Following Corollary 3.5.3, $c' \geq \sqrt{\frac{C(v)}{\pi}}/l(v) - 1$, and thus $a_v \leq (c'+1)\sqrt{\pi}l(v) \leq 4c'l(v)$.

When calculating the embedding of X_v, all the vertices in X_v were embedded inside a square A whose side is $a_v \leq 4c'l(v) = O(c'D(u, u'))$.

After computing the final embedding, A is mapped to a rectangle A', which is obtained from A by expanding by a factor of γ along one axis, and by expanding by a factor of $1/\gamma$ along the other axis. If $v_1, \ldots, v_k$ are all the vertices along the path from the root $r = v_1$ to $v = v_k$, then $\gamma = \prod_{j=1}^{k-1} s(v_{j+1})^{g(v_j)} = h(v)$. Thus, by Lemma 3.5.5, γ is at least $\Omega(1/c')$, and at most $O(c')$. It follows that the diameter of A' is at most $O(c'^2 D(u, u'))$. Since the images of u and u' in the final embedding are contained inside A', the assertion follows. ∎

The following result is now immediate:

Theorem 3.5.8 *Given an ultrametric M that c-embeds into the Euclidean plane, we can compute in linear time an embedding of M into the Euclidean plane with distortion $O(c^3)$.*

3.5.5 Upper Bound on Absolute Distortion

In this section we show that for any $d \geq 2$, any n-point ultrametric can be embedded into ℓ_2^d with distortion $O(d^{1/2}n^{1/d})$.

Given an ultrametric M, we first compute an α-HST T that α-approximates M, for some constant $\alpha > 16$. Let M' be the metric associated with T. Observe that any embedding of M' into ℓ_2^d with distortion c, is also an embedding of M into ℓ_2^d, with distortion $O(c)$. Thus, it suffices to show that M' can be embedded into ℓ_2^d with distortion $O(d^{1/2}n^{1/d})$.

We will compute an embedding of M' into ℓ_2^d inductively, starting from the leaves of T. For every subtree of T rooted at a vertex u, we compute an embedding f_u of the submetric of M' induced by X_u, into ℓ_2^d. We maintain the following inductive properties of f_u:

- The contraction of f_u is at most 16.
- $f(X_u)$ is contained inside a hypercube of side length $l(u)n_u^{1/d}$.

We assume w.l.o.g. that for each leave v of T, $l(v) = 1$. Thus, we can embed each leave in a center of a hypercube of side 1. The following lemma shows how to compute the recursive embedding of inner vertices of T.

Lemma 3.5.9 *Let v be an internal vertex of T, whose children are $u_1, \ldots, u_k$. Assume that for each $i \in [k]$, we are given an embedding $f_{u_i} : X_{u_i} \to \mathbb{R}^d$, with contraction at most 16, such that $f_{u_i}(X_{u_i})$ is contained inside a d-dimensional hypercube S_{u_i}, with side length $l(u_i)n_{u_i}^{1/d}$. Then we can compute in polynomial time an embedding $f_v : X_v \to \mathbb{R}^d$, with contraction at most 16, such that $f_v(X_v)$ is contained inside a d-dimensional hypercube S_v, with side length $l(v)n_v^{1/d}$.*

Proof:

For each $i \in [k]$, let $r_i = l(u_i)n_{u_i}^{1/d}$ be the length of the side of the hypercube S_{v_i}. Let also S'_{u_i} be a hypercube of side length $r'_i = r_i + l(v)/16$, having the same center as S_{u_i}. We assume w.l.o.g. that $n_1 \geq n_2 \geq \cdots \geq n_k$ and thus $r'_1 \geq \cdots \geq r'_k$. We note that for each $i : 1 \leq i \leq k$, $r'_i \leq l(v)n_v^{1/d}/4$, since $r'_i = r_i + l(v)/16 = l(u_i)n_{u_i}^{1/d} + l(v)/16 \leq l(v)n_v^{1/d}/4$.

We first define a partition $\mathcal{R} = \{R_j\}_{j=1}^{\lambda}$, of the set $[k]$, which we will use to partition the set of hypercubes $\{S_{u_i}\}_{i=1}^{k}$, as follows. We will define $\lambda + 1$ integers $t_0, t_1, \ldots, t_\lambda$, where $t_0 = 0$, $t_\lambda = k$, and $t_0 < t_1 < \cdots < t_\lambda$, and then set R_j to contain all the indices $i : t_{j-1} + 1 \leq i \leq t_j$. This defines a partition of the hypercubes into λ sets $\mathcal{S}_1, \ldots, \mathcal{S}_\lambda$, where $\mathcal{S}_j$ contains the hypercubes S_{u_i} with $i \in R_j$. For each $j : 1 \leq j \leq \lambda$, let $\rho_j = r'_{t_{j-1}+1}$ denote the side of the largest hypercube in $\mathcal{S}_j$, and let $\rho'_j = r_{t_j}$ denote the side of the smallest hypercube in $\mathcal{S}_j$.

We now proceed to define the numbers t_j, for $j : 0 \leq j \leq \lambda$. Set $t_0 = 0$, and for each $j \geq 1$, if $t_{j-1} < k$, we inductively define t_j as

$$t_j \quad = \quad \min\{k, t_{j-1} + \lfloor l(v)n_v^{1/d}/r'_{t_{j-1}+1} \rfloor^{d-1}\}.$$

If $t_j = k$ then we set $\lambda = j$.

Note that for any $j \in [\lambda - 1]$,

$$|R_j| = \left\lfloor \frac{l(v)n_v^{1/d}}{\rho_j} \right\rfloor^{d-1}$$

We now define the embedding f_v by placing the hypercubes S'_{u_i} inside a hypercube of side length $l(v)n_v^{1/d}$, such that their interiors do not overlap, using the partition $\mathcal{R}$. For each $j \in [\lambda]$, we place the hypercubes in $\mathcal{S}_j$ inside a parallelepiped W_j having $d - 1$ sides of length $l(v)n_v^{1/d}$, and one side of length ρ_j, as follows. It is easy to see that we can pack $|R_j|$ d-dimensional hypercubes of side ρ_j inside W_j. Since each hypercube in $\mathcal{S}_j$ has side at most ρ_j, we can replace each hypercube embedded into W_j by a hypercube from $\mathcal{S}_j$, such that the centers of both hypercubes coincide.

Finally, we place the parallelepipeds W_j inside a parallelepiped W having $d-1$ sides of length $l(v)n_v^{1/d}$, and one side of length $\sum_{j=1}^{\lambda} \rho_j$. Observe first that the contraction of this embedding is at most 16: for any pair of vertices $x, y \in X(v)$, if x, y both belong to a subtree of the same child u_i of v, then by induction hypothesis the distance between them is contracted by at most 16. If $x \in X(u_i), y \in X(u_{i'})$ and $i \neq i'$, then the original distance is $D(x, y) = l(v)$. Since we add emty space of width $l(v)/32$ around the hypercubes $S(u_q)$ when they are transformed into hypercubes $S'(u_q)$, it is clear that the distance between the embeddings of x and y is at least $l(v)/16$.

It now only remains to show that $\sum_{j=1}^{\lambda} \rho_j \leq l(v)n_v^{1/d}$. We partition the parallelepipeds W_j into two types. The first type contains all the parallelepipeds W_j, where $\rho_j/\rho'_j \geq 2$. Additionally, the last parallelepiped W_k is also of the first type, regardless of the ratio ρ_k/ρ'_k. Let $\mathcal{M}_1 \subseteq [k]$ contain all the indices j where W_j is of the first type. All the other parallelepipeds belong to the second type, and let $\mathcal{M}_2 = [k] \setminus \mathcal{M}_1$ contain the indices of the parallelepipeds of the second type. Notice that for $j \in \mathcal{M}_1$, the values ρ_j form a geometric series with ratio $1/2$. Since the sides r'_i of the hypercubes S_{u_i} are bounded by $l(v)n_v^{1/d}/4$, it is easy to see that:

$$\sum_{j \in \mathcal{M}_1} \rho_j \leq \frac{l(v)n_v^{1/d}}{4}\left(1 + \frac{1}{2} + \frac{1}{4} + \cdots\right) \leq \frac{l(v)n_v^{1/d}}{2}$$

It now remains to bound $\sum_{j \in \mathcal{M}_2} \rho_j$. Fix some $j \in \mathcal{M}_2$, and consider some hypercube S'_{u_i} where $i \in R_j$. As W_j is of the second type, we know that $r'_i \geq \rho_j/2$. On the other hand,

$$\begin{aligned} r'_i = r_i + \frac{l(v)}{16} &= l(u_i)n_{u_i}^{1/d} + \frac{l(v)}{16} \\ &\leq \frac{l(v)}{16}\left(1 + n_{u_i}^{1/d}\right) \leq \frac{l(v)}{4} n_{u_i}^{1/d} \end{aligned}$$

Therefore, $n_{u_i} \geq \left(\frac{2\rho_j}{l(v)}\right)^d$. Recall that for $j : 1 \leq j < \lambda$, $|R_j| = \left\lfloor \frac{l(v)n_v^{1/d}}{\rho_j} \right\rfloor^{d-1} \geq \left(\frac{l(v)n_v^{1/d}}{2\rho_j}\right)^{d-1}$. Therefore, we have that

$$\sum_{i \in R_j} n_{u_i} \geq \left(\frac{l(v) n_v^{1/d}}{2\rho_j} \right)^{d-1} \cdot \left(\frac{2\rho_j}{l(v)} \right)^d \geq \frac{2\rho_j}{l(v)} n_v^{1-1/d}$$

Thus, $\rho_j \leq \frac{l(v) \sum_{i \in R_j} n_{u_i}}{2 n_v^{1-1/d}}$, and

$$\sum_{j \in \mathcal{M}_2} \rho_j \leq \frac{l(v) n_v}{2 n_v^{1-1/d}} \leq \frac{l(v) n_v^{1/d}}{2}$$

We have that in total, $\sum_j \rho_j = \sum_{j \in \mathcal{M}_1} \rho_j + \sum_{j \in \mathcal{M}_2} \rho_j \leq l(v) n_v^{1/d}$. ■

We are now ready to prove the main theorem of this section.

Theorem 3.5.10 *For any $d \geq 2$, any n-point ultrametric can be embedded into ℓ_2^d with distortion $O(d^{1/2} n^{1/d})$. Moreover, the embedding can be computed in polynomial time.*

Proof: Starting from the leaves of T, we inductively compute for each $v \in V(T)$ the embedding f_v as described above. By recursively applying Lemma 3.5.9 we can compute in polynomial time the embedding f_v, that also satisfies the inductive properties. Let f be the resulting embedding f_r.

Consider now two points $x, y \in X$, and let v be the nearest common ancestor of x and y. Since $f_v(X_v)$ is contained inside a hypercube of side length $l(v) n_v^{1/d}$, it follows that $\|f(x) - f(y)\|_2 \leq \left(d n_v^{2/d} l^2(v) \right)^{1/2} = d^{1/2} n^{1/d} D(x,y)$. Since the contraction of f_v is at most 16, it follows that the distortion of f is $O(d^{1/2} n^{1/d})$. ■

Observe that for $d = 2$, the algorithm provides an $O(\sqrt{n})$-distortion embedding. Combining this with the $O(c^3)$-distortion algorithm from Section 3.5.4, we obtain the following result:

Theorem 3.5.11 *There is an efficient $O(n^{1/3})$-approximation algorithm for minimum distortion embedding of ultrametrics into the plane.*

Proof: Let c be the optimal distortion achievable by any embedding of the input ultrametric into the plane. If $c > n^{1/6}$ then the above algorithm, which produces an $O(\sqrt{n})$-distortion embedding is an $O(n^{1/3})$-approximation. Otherwise, if $c \leq n^{1/6}$, then the algorithm from Section 3.5.4 gives $O(c^2) = O(n^{1/3})$-approximation. ■

We remark that Theorem 3.5.10 generalizes a result of Gupta [Gup00a], who shows that every n-point weighted star metric can be embedded into $\mathbb{R}^d$, with distortion $O(n^{1/d})$. This is a corollary of the following simple observation.

Claim 3.5.12 *Every n-point weighted star can be embedded into an ultrametric of size $O(n)$ with distortion at most 2.*

Proof: Consider a star S with root r, and leaves $x_1, \dots, x_n$, where for each $i \in [n]$, $D_S(r, x_i) = w_i$. Assume w.l.o.g. that $w_1 \leq w_2 \leq \dots \leq w_n$. We construct a tree T with root r' as follows. T contains a path $z_n, z_{n-1}, \dots, z_1$, where $z_n = r'$, and for each $i \in [n-1]$, $D_T(r', z_i) = w_n - w_i$. We now embed S into T as follows. For each $i \in [n]$, we add x_i to T, and we connect x_i to z_i with an edge of length w_i. Observe that the shortest-path metric on the leaves of T is an ultrametric, since all the leaves are on the same level. Moreover, for any $i < j \in [n]$, $D_T(x_i, x_j) = 2w_j$, while $D_S(x_i, x_j) = w_i + w_j$, and so the resulting embedding is non-contracting, and has expansion at most 2. ■

3.5.6 NP-hardness of Embedding Ultrametrics Into the Plane

In this section we consider embeddings into the plane under the ℓ_∞ norm. We say that a square $S \subset \mathbb{R}^2$ is *orthogonal* if the sides of S are parallel to the axes.

We will show that the problem of computing a minimum distortion embedding of an ultrametric into the plane under the ℓ_∞ norm is NP-hard. We perform a reduction from the following NP-complete problem (see [LTW+90]): Given a packing square S and a set of packed squares $L = \{s_1, \dots, s_n\}$, is there an orthogonal packing of L into S? We call this problem SquarePacking.

For a square s, let $a(s)$ denote the length of its side. Assume w.l.o.g. for each $i \in [n]$, $a(s_i) \in \mathbb{N}$, $a(S) \in \mathbb{N}$, and that $a(s_1) \leq a(s_2) \leq \dots \leq a(s_n)$. The SquarePacking problem is strongly NP-complete. Thus we can assume w.l.o.g. that there exists $N = \text{poly}(n)$, such that $1 \leq a(s_1) \leq \dots \leq a(s_n) \leq a(S) < N$.

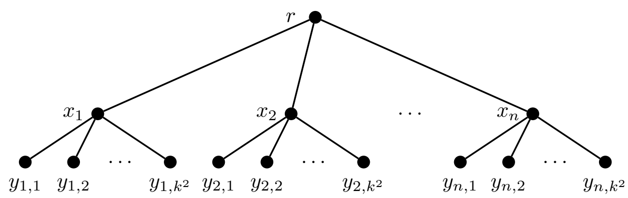


Figure 3-6: The constructed tree T. The labels of the vertices are: $l(r) = a(S)$ and $l(x_i) = a(s_i) - a(S)/(k-1)$.

The Construction

Consider an instance of the SquarePacking problem, where S is the packing square, and $L = \{s_1, \ldots s_n\}$ is the set of packed squares. We will define an ultrametric $M = (X, D)$ and an integer k, such that M embeds into the plane with distortion at most $k-1$ iff there exists an orthogonal packing of L into S. It is convenient to define M by constructing its associated labeled tree T, where each $v \in V(T)$ has a label $l(v) \in \mathbb{Q}$.

Let $k = N^{10}$. For each square $s_i \in L$, we introduce a set of k^2 leaves $y_{i,1}, \ldots y_{i,k^2}$ in T. We connect all of these leaves to a vertex x_i, and we set $l(x_i) = a(s_i) - a(S)/(k-1)$. Note that $l(x_i)$ is very close to $a(s_i)$. Next, we introduce a root vertex $r \in V(T)$, and for each $i \in [n]$, we connect x_i to r. We set $l(r) = a(S)$.

For a vertex $v \in V(T)$, we denote by X_v the set of leaves of T having v as an ancestor. Figure 3-6 depicts the described construction.

YES-Instance

Assume that there exists an orthogonal packing of L into S. We will show that there exists an embedding $f : X \to \mathbb{R}^2$ with distortion $k-1$.

As a first step, for each vertex $x_i : 1 \le i \le n$, we embed all the vertices of X_{x_i} in a square Q_i of side $(k-1)l(x_i)$. This is done by simply placing a $k \times k$ orthogonal grid with step $l(x_i)$ inside Q_i and embedding the vertices of X_{x_i} on the grid points. Next, we transform the squares Q_i into squares Q'_i by adding empty strips of width $a(S)/2$ around Q_i. Notice that the side of Q'_i is exactly $(k-1)l(x_i) + a(S) = (k-1)a(s_i)$. Finally, we embed the squares Q'_i into a square $\mathcal{S}$ of side $(k-1)a(S)$ according to the

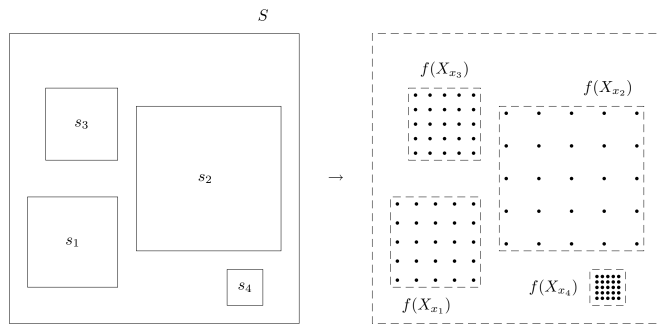


Figure 3-7: The embedding constructed for the YES instance.

packing of the input squares in S. Figure 3-7 depicts the resulting embedding f.

We now show that the distortion of the embedding f is at most $k-1$.

Let $u, v \in X$. We have to consider the following cases for u, v:

Case 1: $u, v \in X_{x_i}$ for some $i \in [n]$. Since the vertices of X_{x_i} are embedded on a grid of step $l(x_i)$, it follows that $\|f(u) - f(v)\|_\infty \geq l(x_i) = D(u, v)$. Thus, the contraction is at most 1. Moreover, since all the vertices of X_{x_i} are embedded inside a square Q_i of side $l(x_i)(k-1)$, the expansion is at most $k-1$.

Case 2: $u \in X_{x_i}$ and $v \in X_{x_j}$, for some $i \neq j$. Since we add empty strips of width $a(S)/2$ around the squares Q_i, Q_j, we have that $\|f(u) - f(v)\|_\infty \geq a(S) = l(r) = D(u, v)$. Thus, the contraction is 1. On the other hand, all the vertices are embedded inside a square $\mathcal{S}$ of side $l(r)(k-1) = a(S)(k-1)$, and therefore the expansion is at most $k-1$.

Thus, we have shown that the distortion is at most $k-1$.

NO-Instance

Assume that there is no orthogonal packing of L inside S. We show that the minimum distortion required to embed M into the plane is greater than $k-1$. Assume that there exists an embedding $f : X \to \mathbb{R}^2$, with distortion at most $k-1$. W.l.o.g. we can assume that f is non-contracting.

The following lemma will be useful in the analysis.

Lemma 3.5.13 *Let $M = (X, D)$ be a uniform metric on k^2 points, for some integer $k > 0$. Then, the minimum distortion for embedding M into the plane is $k-1$. Moreover, an embedding f has distortion $k-1$ iff $f(X)$ is an orthogonal grid.*

Proof: By scaling M, we can assume w.l.o.g. that for any $u, v \in X$, $D(u, v) = 1$. Consider an non-contracting embedding $f : X \to \mathbb{R}^2$. For any $v \in X$, let A_v be square of side length 1, centered at $f(v)$. Clearly, for any $u, v \in X$, with $u \neq v$, the interiors of squares A_u and A_v are disjoint. Let $A = \bigcup_{v \in X} A_v$. It follows that $\mathrm{Vol}(A) = |X|$. Thus, there exist $p_1, p_2 \in A$, such that $\|p_1 - p_2\|_\infty \geq |X|^{1/2} = k$. Let $v_1, v_2 \in X$ be the centers of the squares A_{v_1}, A_{v_2} to which p_1 and p_2 belong, respectively. Then $\|f(v_1) - p_1\|_\infty \leq 1/2$, and $\|f(v_2) - p_2\|_\infty \leq 1/2$. It follows that $\|f(v_1) - f(v_2)\|_\infty \geq k - 1$. Thus the distortion is at least $k - 1$.

Clearly, if f maps X onto a $k \times k$ orthogonal grid, the distortion of f is $k - 1$. It remains to show that this is the only possible optimal embedding.

Assume that an embedding f has distortion $k - 1$, and let f be non-contracting. Observe that since the diameter of $f(X)$ is at most $k - 1$, $f(X)$ must be contained inside a square K of side length $k - 1$. Let $\{A_v\}_{v \in X}$ be defined as above. It follows that A is contained inside a square K' of side length k. Since $\mathrm{Vol}(A) = \mathrm{Vol}(K')$, it easily follows that $f(X)$ is an orthogonal $k \times k$ grid. ■

Corollary 3.5.14 *For each $i \in [n]$, $f(X_{x_i})$ is an orthogonal $k \times k$ grid of side length $(k-1)l(x_i) = (k-1)a(s_i) - a(S)$.*

For each $i \in [n]$, let Q'_i be the square of side length $(k-1)a(s_i)$, that has the same center of mass as $f(X_{x_i})$.

Claim 3.5.15 *For each $i, j \in [n]$, $i \neq j$, the interiors of the squares Q'_i, Q'_j are disjoint.*

Proof: Assume that the assertion is not true. That is, there exist $i, j \in [n]$, with $i \neq j$, and $p \in \mathbb{R}^2$, such that p belongs to the interiors of both squares Q'_i, Q'_j. By the definition of Q'_i and Q'_j, there are points $v_1 \in X_{x_i}$, $v_2 \in X_{x_j}$ which are embedded within distance smaller than $a(S)/2$ from p. But then $\|f(v_1) - f(v_2)\|_\infty < a(S)$, contradicting the fact that the embedding is non-contracting. ■

Claim 3.5.16 $\bigcup_{i=1}^{n} Q'_i$ *is contained inside a square of side length* $ka(S)$.

Proof: Since f has expansion at most $k-1$, $f(X)$ is contained inside an orthogonal square $\mathcal{S}$ of side length $(k-1)l(r) = (k-1)a(S)$. Observe that for each $i \in [n]$, for each point $p \in Q_i$, there exists $v \in X_{x_i}$, such that $\|p - f(v)\|_\infty \leq a(S)/2$. Let $\mathcal{S}'$ be the square of side length $ka(S)$ that has the same center as $\mathcal{S}$. It follows that $\mathcal{S}'$ contains $\bigcup_{i=1}^{n} Q'_i$. ■

Lemma 3.5.17 *If M can be embedded into the plane with distortion at most $k-1$, then there exists an orthogonal packing of L inside S.*

Proof: If there exists an embedding $f : X \to \mathbb{R}^2$ with distortion $k-1$, by Claim 3.5.16 we obtain that $\bigcup_{i=1}^{n} Q_i$ is contained inside a square of side length $ka(S)$. Moreover, by Claim 3.5.15, the embeddings of squares Q'_i defines a feasible packing of these squares into the square $\mathcal{S}'$. Note that for each $i : 1 \leq i \leq n$, Q_i has side length $(k-1)a(s_i)$. That is, the squares $Q_1, \ldots, Q_n$ are just scaled copies of the squares $s_1, \ldots, s_n$. Thus, we obtain that there exists an orthogonal packing of L inside a square S' of side length $a(S)\frac{k}{k-1}$. Recall that $k = N^{10} > a(S)^{10}$. Thus, S' has side length less than $a(S) + 1/2$.

Since $a(S)$ and $a(s_i)$ for each $i \in [n]$ are integers, it follows that there is also an orthogonal packing of L into a square of side length $a(S)$. ■

The following theorem is now immediate.

Theorem 3.5.18 *The problem of minimum-distortion embedding of ultrametrics into the plane under the ℓ_∞ norm is NP-hard.*

3.5.7 Approximation Algorithm for Embedding Ultrametrics Into Higher Dimensions

In this section we extend the techniques used in Section 3.5.4, to obtain an approximation algorithm for embedding ultrametrics into ℓ_2^d.

Given an ultrametric $M = (X, D)$ that embeds into ℓ_2^d with distortion c, we first embed M into a 2-HST $M' = (X, D')$. Let T be the labeled tree associated with M', as in Section 3.5.4. Then M' embeds into ℓ_2^d with distortion $c' = O(c)$. We now focus on finding an embedding of M' into the ℓ_2^d with distortion at most $c'^{O(d)}$. The same embedding is an $c^{O(d)}$-distortion embedding of M into ℓ_2^d. We compute an embedding of M' into ℓ_2^d by recursively embedding the subtrees of vertices in a bottom-up fashion.

For any vertex u in the tree, let $a_u = (C(u))^{1/d}$. If u is a non-root vertex, let v be the father of u in T. We set $b_u = a_u + (V_d(l(v)/4))^{1/d}$, and $s_u = a_v/b_u$. If u is the root of the tree, we set $s_u = 1$.

Given a vertex v in the tree, we embed the vertices in X_v into a hypercube of side a_v, recursively. Let $u_1, \ldots, u_k$ be the children of v, and assume that for each $i \in [k]$, we are given an embedding of X_{u_i} into a d-dimensional hypercube Q_{u_i} of side length a_{u_i}. We define an additional hypercube Q'_{u_i} of side length b_{u_i} that has the same center as Q_{u_i} (i.e., Q'_{u_i} is obtained from Q_{u_i} by adding a "shell" of width $(V_d(l(v)/4))^{1/d}/2$ around Q_{u_i}). Let Q_v be a d-dimensional hypercube of side length a_v.

Note that the volume of Q_v equals the sum of volumes of Q'_{u_i}, for $1 \le i \le k$. This is since the volume of Q_v is $a_v^d = C(v)$, while the sum of volumes of $Q'_{u_i}, 1 \le i \le k$ is

$$\sum_{i=1}^{k} b_{u_i}^d = \sum_{i=1}^{k} \left((C(u_i))^{1/d} + (V_d(l(v)/4))^{1/d} \right)^d = C(v).$$

Fix one coordinate $j \in [d]$. We now show how to embed the hypercubes $Q'_{u_1}, \ldots, Q'_{u_k}$ into Q_v. Consider some hypercube $Q'_{u_i} : 1 \le i \le k$. For each dimension $j' \neq j$, we increase the length of the corresponding side of Q'_{u_i} by the factor of s_{u_i}. Additionally, we decrease the length of the side of Q'_{u_i} corresponding to the dimension j by the factor of $s_{u_i}^{d-1}$. Let R_i denote the resulting parallelepiped. Notice that for each dimension $j' \neq j$, the length of the corresponding side of parallelepiped R_i is exactly a_v. Moreover, the volume of R_i equals the volume of Q'_{u_i}. Therefore, we can easily pack the parallelepipeds R_i, $1 \leq i \leq k$, inside the hypercube Q_v, where the shortest side of R_i is placed along dimension j.

As in the algorithm for embedding ultrametrics into the plane, we need to ensure that these stretchings do not accumulate as we go up the tree. To ensure this, we calculate, for each vertex v a value $g(v) \in [d]$. When calculating the embedding of the hypercubes $Q'_{u_1}, \ldots, Q'_{u_k}$ into the hypercube Q_v, we contract the hypercubes $Q'_{u_1}, \ldots, Q'_{u_k}$ along the dimension $g(v)$ and expand them along all the other dimensions.

Our next goal is to prove an analogue of Lemma 3.5.5, that shows how to calculate the values $g(v)$ so that the total distortion is not accumulated.

We start with the following claim:

Claim 3.5.19 *For each vertex u of the tree, $1 \leq s_u \leq 8c'$.*

Proof: If u is the root of the tree, then $s_u = 1$ and the claim is trivially true. Assume now that u is not the root, and let v be its father. We denote the children of v by $u_1, \ldots, u_k$, and we assume that $u = u_i$ for some $i \in [k]$.

Recall that $s_u = a_v/b_u$, and that we have already observed that $a_v^d = \sum_{j=1}^k b_{u_j}^d$, and thus $s_u \geq 1$ clearly holds.

We now prove the second inequality. For the sake of convenience, we denote $V = (V_d(l(v)/4))^{1/d}$. Recall that $b_u = a_u + V \geq V$.

On the other hand, from Corollary 3.5.3,

$$c' \geq \rho_d(C(v))/l(v) - 1$$

Therefore, we have that

$$\rho_d(C(v)) = \left(\frac{C(v)\Gamma(1 + d/2)}{\pi^{d/2}}\right)^{1/d} \leq 2c'l(v)$$

and thus

$$a_v = C(v)^{1/d} \leq 2c' l(v) \left(\frac{\pi^{d/2}}{\Gamma(1+d/2)} \right)^{1/d} = 8c'V$$

Therefore, $s_u = a_v / b_u \leq 8c'V/V \leq 8c'$. ■

For each vertex u of the tree, for each dimension $j \in [d]$, we recursively define a value $h_j(u)$, as follows. If u is the root, then $h_j(u) = 1$ for all $j \in [d]$. Consider now some vertex u which is not the root, and let v be its father. Then we define $h_j(u) = h_j(v) \cdot s_u^{\alpha_j(v)}$, where $\alpha_j(v)$ is defined to be 1 if $j \neq g(v)$, and it is defined to be $-(d-1)$ if $i = g(v)$. Notice that $\prod_{j \in [d]} h_j(u) = 1$.

Fix any vertex $u \in V(T)$ and any dimension $j \in [d]$. Let Q_u be the hypercube of side a_u into which the vertices of X_u have been embedded when u was processed by the algorithm. Then the value $h_j(u)$ is precisely the stretch along the dimension j of Q_u in the final embedding. In other words, if we take a pair of points $x, y \in Q_u$ such that $x_j = y_j - 1$, and for all the other coordinates j', $x_{j'} = y_{j'}$, then $h_j(u)$ is precisely the distance between x and y in the final embedding. We next prove that we can calculate the values $g(v)$ in a way that ensures that that for each vertex u and for each dimension $j \in [d]$, $h_j(u)$ lies between $(O(1/c'))^d$ and $(O(c'))^d$.

Lemma 3.5.20 *We can compute in polynomial time values $g(u)$ for all $u \in V(T)$, such that for each $u \in V(T)$, for each dimension $j \in [d]$, $(O(1/c'))^d \leq h_j(u) \leq (O(c'))^d$.*

Proof: If u is the root, then we arbitrarily set $g(u) = 1$.

Consider now some non-root vertex u, and let v be its parent. Let $j \in [d]$ be the dimension for which $h_j(v)$ is maximized. Then we set $g(u) = j$.

Claim 3.5.21 *For every vertex u, $\frac{\max_i\{h_i(u)\}}{\min_i\{h_i(u)\}} \leq (8c')^d$.*

Proof: The claim is trivially true for the root r since $\frac{\max_i\{h_i(r)\}}{\min_i\{h_i(r)\}} = 1$. For any non-root vertex u, assume that the claim is true for its parent v. Assume w.l.o.g. that $h_1(v) \geq h_2(v) \geq \cdots \geq h_d(v)$, and $g(u) = 1$. Then $h_1(u) = h_1(v)/s_u^{d-1}$, and for each $i > 1$, $h_i(u) = h_i(v) \cdot s_u$. There are three cases to consider. If $h_1(u)$ equals the maximum value among $\{h_i(u)\}_{i=1}^d$, then clearly $\frac{\max_i\{h_i(u)\}}{\min_i\{h_i(u)\}} \leq \frac{\max_i\{h_i(v)\}}{\min_i\{h_i(v)\}} \leq (8c')^d$ by the induction hypothesis. If $h_1(u)$ equals the minimum value among $\{h_i(u)\}_{i=1}^d$, then $\frac{\max_i\{h_i(u)\}}{\min_i\{h_i(u)\}} = \frac{h_2(u)}{h_1(u)} = \frac{s_u^d h_2(v)}{h_1(v)} \leq s_u^d$. Finally, if neither of the above two cases happens, then $\frac{\max_i\{h_i(u)\}}{\min_i\{h_i(u)\}} = \frac{h_2(u)}{h_d(u)} = \frac{h_2(v)s_u}{h_d(v)s_u} \leq (8c')^d$ by the induction hypothesis. ■

Since $\prod_{i=1}^d h_i(u) = 1$, we get that $(O(c'))^{-d} \leq h_i(u) \leq (O(c'))^d$.

It is easy to see that the algorithm for computing the values $g(u)$, runs in polynomial time.

■

Let $f : X \to \Re^d$ denote the resulting embedding produced by the algorithm. The next two lemmas bound the maximum contraction and the maximum expansion of the distances in this embedding.

Lemma 3.5.22 *For any pair $u, u' \in X$ of points, $\|f(u)-f(u')\|_\infty \geq (O(c'))^{-d}\, D'(u, u')$.*

Proof: Fix any pair $u, u' \in X$ of vertices, and let v be their least common ancestor in the tree T. Thus, $D'(u, u') = l(v)$. Let z, z' be the children of v such that $u \in X_z$ and $u' \in X_{z'}$. Assume w.l.o.g. that $s_z > s_{z'}$. Recall that $Q'_z, Q'_{z'}$ contain empty shell of width $(V_d(l(v)/4))^{1/d}/2$ in which no vertices are embedded. When $Q'_z, Q'_{z'}$ are embedded inside Q_v, they are contracted by the factors $s_z, s_{z'}$ respectively along the ith dimension, where $i = g(v)$. Thus, in the embedding of X_v inside Q_v, the distance between the images of u and u' along the ith dimension is at least:

$$\frac{V_d(l(v)/4)}{s_u^{d-1}} = \frac{\sqrt{\pi}l(v)}{4(\Gamma(1+d/2))^{1/d}s_u^{d-1}} \geq \frac{l(v)}{2^{O(\log d)}s_u^{d-1}}$$

In the final embedding this distance is multiplied by the factor $h_i(v)$. Thus, the final distance is at least

$$\frac{l(v)}{2^{O(\log d)}s_u^{d-1}}h_i(v) = \frac{l(v)}{2^{O(\log d)}}h_i(u) \geq \frac{l(v)}{(O(c'))^d}$$

■

Lemma 3.5.23 *For any pair $u, u' \in X$ of points, $\|f(u)-f(u')\|_\infty \leq (O(c'))^{d+1}\, D'(u, u')$.*

Proof: Fix any pair $u, u' \in X$ of vertices, and let v be their least common ancestor in the tree T, so that $D'(u, u') = l(v)$.

Recall that Q_v is a hypercube of side a_v, and thus when the embedding of X_v has been computed, the distance between the images of u and u' was at most a_v. In the final embedding this distance increased by the factor of at most $\max_{i \in [d]} \{h_i(v)\} \leq (O(c'))^d$, and thus the final distance is at most $a_v (O(c'))^d$. From Corollary 3.5.3, using the same reasoning as in the proof of Claim 3.5.19, we have that

$$a_v \leq 2c' l(v) \frac{\sqrt{\pi}}{(\Gamma(1 + d/2))^{1/d}} \leq O(c') l(v)$$

Thus, $\|f(u) - f(u')\|_\infty \leq (O(c'))^{d+1} l(v)$. ■

Combining the results of Lemma 3.5.22 and Lemma 3.5.23, we obtain the following theorem.

Theorem 3.5.24 *For any $d > 2$, there is a polynomial time algorithm that embeds any input ultrametric M into ℓ_2^d with distortion $c^{O(d)}$, where c is the optimal distortion of embedding M into ℓ_2^d.*

3.5.8 Conclusions and Open Problems

In this section we investigated the problem of embedding ultrametrics into low-dimensional spaces $\Re^d$. In particular, for $d = 2$, we provided two results. The first one was relative: a linear-time algorithm which, given any ultrametric c-embeddable into the plane, produces an embedding with distortion $O(c^3)$. The second result was absolute: any n-point ultrametric can be embedded into the plane with distortion $\sqrt{n}$.

The key question left open by this work is: is it possible to generalize our results to a larger class of (weighted) metrics? In particular, it would be very interesting to design an algorithm for relative embeddings of (weighted) tree metrics. Such metrics are encountered in many applied areas, such as computational biology. Similarly, it would be interesting to obtain an $o(n)$-distortion embedding of weighted tree metrics into the plane (this problem has been posed already in [BMMV02]).

Finally, it remains to determine what is the best possible distortion of relative embeddings of ultrametrics into the plane that can be computed in polynomial time. Our results show that the answer is greater than c but smaller than $O(c^3)$, leaving a wide range of possibilities.

3.6 Approximation Algorithms for Embedding General Metrics Into Trees

Credits: The results in this section is work done with Piotr Indyk, and Anastasios Sidiropoulos, and it hasn't been published yet.

In this section, 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.

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 $\Omega(\log n / \log \log n)$.

3.6.1 Introduction

In this section 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.

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 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[5] 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

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

$(c\log n)^{O(\sqrt{\log\Delta})}$. 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].

Related Work

The study of the problem of approximating metrics by tree metrics has been initiated in [FCK96, 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.

Our techniques

In this section we give a brief overview of the main ideas behind our algorithms. For more in-depth descriptions, see the introductions to the individual sections.

We start from the unweighted graphs. The basic observation behind our algorithms is that if a metric M can be embedded into the a with "low" distortion, then M should "look" like a tree. That is, there exists a decomposition of M into clusters, which can be "connected" together in a manner resembling some tree (say, T). In this case, the embedding can be constructed by embedding each cluster separately, and combining the embeddings using T as an outline.

We mention that a similar general idea has been used in the aforementioned algorithm of [EP04], for $O(\log n)$-approximating the distortion of embedding unweighted graph metrics into subtrees. However, our algorithm for computing the decomposition is substantially different from theirs. In particular, [EP04] use a divide-and-conquer method for constructing the decomposition, using the existence of balanced separators for the input metric. In contrast, we employ a BFS-like approach. It appears that our method leads to a significantly simpler algorithm, even after combining the algorithm for embedding into trees with the "tree-to-subtree" conversion.

To extend the above approach to the weighted metrics, we need to apply the algorithm recursively within each cluster in the decomposition. This is because, unlike in the *unweighted* case, here the diameter of each cluster can be pretty large. To ensure that multiple recursive steps do not amplify the distortion, we need a stronger tree-like decomposition, in which any two clusters are relatively "far apart". We call it a *tree-like well-separated* decomposition.

The techniques of Emek and Peleg, start with a somewhat similar clustering to our algorithm for unweighted graphs. Their clustering requires to be balanced (ie, split the graph in parts at most half the original size), while our does not. The way we construct the tree for the base level and for the recursion level is radically different, mainly because we are not outputting a subtree for the $O(1)$-approximation algorithm for the unweighted case. Also, in the end, our recreating of their $O(\log n)$-approximation algorithm for spanning subtree for unweighted graphs is much simpler and shorter.

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 stated otherwise that the tree might contain steiner nodes. By a result of Gupta [Gup01], after computing 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 $D(u, v)$.

3.6.2 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 3.6.1 *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, with $0 \leq i < 4$, there exists a path p_i, with endpoints v_i, and $v_{i+1 \bmod 4}$, and*
- *for each i, with $0 \leq i < 4$, $D_G(p_i, p_{i+2 \bmod 4}) > \lambda W_G$,*

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 3.6.2 *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 3.6.3 *For each i, with $0 \leq i < 4$, we have $T_i \cap T_{i+2 \bmod 4} = \emptyset$.*

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

Case 1: *$T_i \cap T_{i+2 \bmod 4}$ contains a node from $V(p_i) \cup V(p_{i+2 \bmod 4})$.* W.l.o.g., we assume that there exists $w \in V(p_{i+2 \bmod 4})$, such that $w \in T_i \cap T_{i+2 \bmod 4}$. By Claim 3.6.2, 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 $D_G(p_i, p_{i+2 \bmod 4}) > \lambda W$, 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: *$T_i \cap T_{i+2 \bmod 4}$ does not contain nodes from $V(p_i) \cup V(p_{i+2 \bmod 4})$.* Let $w \in T_i \cap T_{i+2 \bmod 4}$. By Claim 3.6.2, there exist $\{u_1, v_1\} \in E(p_i)$, and $\{u_2, v_2\} \in E(p_{i+2 \bmod 4})$, such that w lies in both P_{u_1,v_1}, and P_{u_2,v_2}. We have $D_T(f(u_1), f(v_1)) + D_T(f(u_2), f(v_2)) = D_T(f(u_1), f(w)) + D_T(f(w), f(v_1)) + D_T(f(u_2), f(w)) + D_T(f(w), f(v_2)) \geq D_T(f(u_1), f(u_2)) + D_T(f(v_1), f(v_2)) \geq D_G(u_1, u_2) + D_G(v_1, v_2) \geq 2D_G(p_i, p_{i+2 \bmod 4}) > 2\lambda W$. 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 $p_{i+1 \bmod 4}$, share an end-point, we have $T_i \cap T_{i+1 \bmod 4} \neq \emptyset$. By Claim 3.6.3, it follows, that $\bigcup_{i=0}^{3} T_i \subseteq T$, contains a cycle, a contradiction. ■

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.

$T_{\mathcal{K}}^G$ 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 3-8 depicts the described decomposition.

Proposition 3.6.4 *$T_{\mathcal{K}}^G$ 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. ■

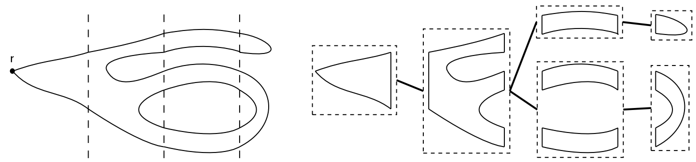


Figure 3-8: An example of a tree-like decomposition of a graph.

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 useful 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 3.6.1 in G. Applying lemma 3.6.1 we obtain a lower bound on the optimal distortion, contradicting the fact that G embeds into a tree with small distortion.

Lemma 3.6.5 *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 3.6.1. 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 3.6.6 *Let $G = (V,E)$ be a graph that γ-embeds into a tree, let $r \in V(G)$, and let $(T^G_{\mathcal{K}}, \mathcal{K}_G)$ be a (r,γ)-tree-like decomposition of G. Then, for any $K_1, K_2 \in \mathcal{K}_G$, and for any $x_1 \in K_1$, $x_2 \in K_2$,*

$$(D_{T^G_{\mathcal{K}}}(K_1,K_2) - 2)W_G\gamma \leq D_G(x_1,x_2) \leq (D_{T^G_{\mathcal{K}}}(K_1,K_2) + 2)20W_G\gamma.$$

Proof: Let $\delta = D_{T^G_{\mathcal{K}}}(K_1,K_2)$. 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 $T^G_{\mathcal{K}}$.* Let the path between K_1 and K_2 in $T^G_{\mathcal{K}}$ 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 ancestor of K_1 and K_2, it follows that K_a separates 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 3.6.5, the diameter of each $K \in \mathcal{K}_G$ is at most $20W_G\gamma$, it follows that $D_G(x_1, x_2) \leq \delta W_G + (\delta + 1)20W_G\gamma < (\delta + 2)20W_G\gamma$. ■

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 adjacent 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 $\mathcal{K}_G = \{K_1, \dots, K_t\}$. 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 $T^G_{\mathcal{K}}$, we add an edge $\{\rho_i, \rho_j\}$ in T.

By proposition 3.6.4, we know that the resulting graph T is indeed a tree, so we can focus of bounding the distortion of T. By lemma 3.6.5, 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 $D_T(x_1, x_2) = 2 + D_T(\rho_1, \rho_2) = 2 + D_{T^G_{\mathcal{K}}}(K_1, K_2)$. By lemma 3.6.6 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 3.6.7 *There exists a polynomial time, constant-factor approximation algorithm, for the problem of embedding an unweighted graph into a tree, with minimum multiplicative distortion.*

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 independently 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 allows 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 3.6.8 *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: Since $G[A]$ is connected, it suffices to show that any edge e of G is either contained in some $K \in \mathcal{K}_G$, or the end-points of e are contained in sets $K, K' \in \mathcal{K}_G$, such that there is an edge between K and K' in $T_{\mathcal{K}}^G$. Assume that this is not true, and pick an edge $\{v_1, v_2\} \in E(G)$, with $v_1 \in K_1$, and $v_2 \in K_2$, for some $K_1, K_2 \in \mathcal{K}_G$, such that there is no edge between K_1 and K_2 in $T_{\mathcal{K}}^G$.

Let $K_r \in \mathcal{K}_G$ be such that $r \in K_r$. Assume first that K_1 is on the path from K_2 to $K_r \in \mathcal{K}_G$ in $T_{\mathcal{K}}^G$. This implies however that $D(v_1, v_2) > W_G$, contradicting the fact that $\{v_1, v_2\} \in E(G)$.

It remains to consider the case where K_1 is not in the path from K_2 to K_r, and K_2 is not in the path from K_1 to K_r in $T_{\mathcal{K}}^G$. Then by the construction of $\mathcal{K}_G$ we know that any path from a vertex in K_1 to a vertex in K_2 in G has to pass through an ancestor of K_1, and K_2. Thus, there is not edge between K_1 and K_2 in G, a contradiction. ■

$\mathcal{A}_G$ 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.

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 is worth 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 3.6.9 *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. ■

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 3.6.10 *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 3.6.8 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 3.6.5, 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 3.6.6 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. ■

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

Proof: 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 $\mathcal{A}_G^0 = \bar{A}_G$.

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

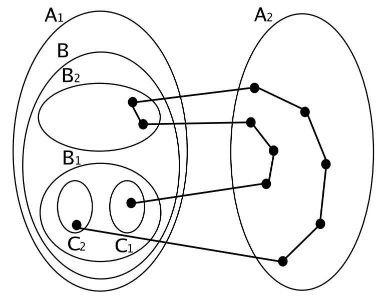


Figure 3-9: Case 2 of the proof of Lemma 3.6.11.

are connected. Thus, $D_G(A_1, A_2) > W_G/(\gamma^{1/2}\alpha)$.

By Lemma **??**, 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, $D_T(z_1, z_2) \geq 2D_G(A_1, A_2) > 2W_G/(\gamma^{1/2}\alpha) \geq 2\gamma D(z_1, z_2)$, 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 3-9. ∎

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.

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

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

Proof:

Claim 3.6.13 *For any $A_i, A_j \in \mathcal{A}_G$, with $A_i \neq A_j$, either $T_i \cap T_j = \emptyset$, or there exists $v \in V(T)$, and $v_1, \ldots, v_l$, for some $l \geq 0$, such that $v_1, \ldots, v_l$ are children of v, and $T_i \cap T_j = \{v, v_1, \ldots, v_l\}$.*

Proof: It follows immediately from the fact that for any $A_i, A_j \in \mathcal{A}_G$, the diameter of $T_{\mathcal{K}}^{G,A_i} \cap T_{\mathcal{K}}^{G,A_j}$ is at most $50c^2$. ■

Let r be the root of T. Initially, T' contains a single vertex r'. To simplify the discussion, we assume w.l.o.g., that r is a leaf vertex of T. We also assume that for every edge $\{u, v\} \in E(T)$, there is a tree T_i that contains $\{u, v\}$. This is because if there is no such tree, then we can simply introduce a new subtree T_i, that contains only the vertices u, and v.

For every T_i that visits r, we introduce in T' a copy $\varphi_i(T_i)$ of T_i, and we connect $\varphi_i(r)$ to r'.

We proceed by visiting the vertices of T in a top-down fashion. Assume that we are visiting a vertex $v \in V(T)$, with parent $p(v)$, and children $v_1, \ldots, v_t$. At this step, we are going to introduce in T' a copy $\varphi_i(T_i)$ of T_i, for every T_i that visits v, and we have not considered yet. We consider the following cases:

Case 1: There is no T_i that visits v, and $p(v)$.

Let T_a be a subtree that visits $p(v)$. For every T_b that visits v, and we have not considered yet, we introduce in T' a copy $\varphi_b(T_b)$ of T_b, and we connect $\varphi_b(v)$ to $\varphi_a(p(v))$.

Case 2: There exists T_i that visits v, and $p(p(v))$, and there is no $j \neq i$, such that T_j visits v, and $p(v)$.

For every T_b that visits v, and we have not considered yet, we introduce in T' a copy $\varphi_b(T_b)$ of T_b, and we connect $\varphi_b(v)$ to $\varphi_i(v)$.

Case 3: There is no T_i that visits v, and $p(p(v))$, and there exists T_j that visits v, and $p(v)$.

Let $a \in [k]$ be the minimum integer such that T_a visits v, and $p(v)$. For every T_b that visits v, and we have not considered yet, we introduce in T' a copy $\varphi_b(T_b)$ of T_b, and we connect $\varphi_b(v)$ to $\varphi_a(v)$.

Case 4: There exists T_i that visits v, and $p(p(v))$, and there exists T_j, with $i \neq j$, that visits v, and $p(v)$.

Let $a \in [k]$ be the minimum integer with $a \neq i$, such that T_a visits v, and $p(v)$. For every T_b that visits v, and we have not considered yet, we introduce in T' a copy $\varphi_b(T_b)$ of T_b. With probability $1/2$, we connect $\varphi_b(v)$ to $\varphi_i(v)$, and with probability $1/2$, we connect $\varphi_b(v)$ to $\varphi_a(v)$.

Claim 3.6.14 *T' is a tree.*

Proof: T' is a forest since each $\varphi_i(T_i)$ is a tree, and also each $\varphi_i(T_i)$ is connected to exactly one $\varphi_j(T_j)$, such that T_j was considered before i. Also, T' is connected since every vertex of T is contained in some subtree T_t. ■

Claim 3.6.15 *For any $v \in V(T)$, there exists at most one $i \in [k]$, such that T_i visits both v, and $p(p(v))$.*

Proof: Assume that the assertion is not true. Let T_i, T_j be subtrees that visit both v, and $p(p(v))$. Then, T_i and T_j also visit $p(v)$. This however contradicts the definition of the subtrees $T_1, \ldots, T_k$. ■

Claim 3.6.16 *Let $i, j \in [k]$, with $i \neq j$, be such that T_i, and T_j both visit a vertex $v \in V(T)$, but they do not visit $p(v)$. Then, with probability at least $1/2$, there exists $t \in [k]$, such that T_t visits v, and $p(v)$, and both $\varphi_i(v)$, and $\varphi_j(v)$ are connected to $\varphi_t(v)$.*

Proof: Recall the procedure for constructing T', described above. Consider the step in which we add to T' the subtrees that visit the vertex v, and v is their highest vertex in T. Clearly T_i, and T_j are both in this set of subtrees. Observe that in cases 1, 2, and 3, the first event of the assertion happens with probability 1. This is because all the trees that we consider are connected to the same subtree.

In the remaining case 4, there are subtrees $T_{i'}$, $T_{j'}$ such that each subtree that we consider is going to be connected to $T_{i'}$ with probability 1/2, and to $T_{j'}$ with probability 1/2. Thus, with probability 1/2, T_i and T_j are going to be connected to the same subtree. ■

Claim 3.6.17 *Let $i, j \in [k]$, with $i \neq j$, be such that T_i visits v, and does not visit $p(v)$, and T_j visits both v, and $p(v)$, for some $v \in V(T)$. Then, with probability at least 1/4, there exists $L \leq 4$, and $t(1), \ldots, t(L)$, such that*

- *$t(1) = i$, and $t(L) = j$,*
- *for each $l \in [L-1]$, $\varphi_{t(l)}(T_{t(l)})$ is connected to $\varphi_{t(l+1)}(T_{t(l+1)})$.*

Proof: We have to consider the following cases:

Case 1: T_j visits $p(p(v))$.

In this case, $\varphi_i(v)$ is connected to $\varphi_j(v)$ with probability at least 1/2.

Case 2: T_j does not visit $p(p(v))$.

Let w be the smallest integer, such that T_w visits v, and $p(v)$, but does not visit $p(p(v))$. If $w = j$, then $\varphi_i(v)$ is connected to $\varphi_j(v)$ with probability at least 1/2.

Otherwise, if $w \neq j$, then with probability at least 1/2, $\varphi_i(v)$ is connected to $\varphi_w(v)$. Moreover, by Claim 3.6.16, with probability at least 1/2, there exists $w' \in [k]$, such that both $\varphi_w(p(v))$, and $\varphi_j(p(v))$, are connected to $\varphi_{w'}(p(v))$. Observe that the above two events are independent. Thus, with with probability at least 1/4, the sequence of subtrees $T_i, T_w, T_{w'}, T_j$, satisfy the conditions of the assertion.

■

Claim 3.6.18 *Let T_i, T_j be two subtrees such that they both visit some vertex $v \in V(T)$. Then, with probability at least $1 - n^{-4}$, there exists $L = O(\log n)$, such that for any T_i, T_j, there exists a sequence of subtrees $T_{t(1)}, \ldots, T_{t(L)}$, with*

- *$t(1) = i$, and $t(L) = j$, and*
- *for any $l \in [L-1]$, $\varphi_{t(l)}(T_{t(l)})$ is connected to $\varphi_{t(l+1)}(T_{t(l+1)})$.*

Proof: By the previous claim, we know that with constant probability there exists a path of length at most 3 between $\varphi_i(T_i)$ and $\varphi_j(T_j)$ in T'. If this happens, then we have a small path between $\varphi_i(T_i)$ and $\varphi_j(T_j)$. Otherwise, we look at the trees $\varphi_{i'}(T_{i'})$ and $\varphi_{j'}(T_{j'})$ which are connected to $\varphi_i(T_i)$ and $\varphi_j(T_j)$ towards the root, and they visit the vertex $p(p(v))$. Note that with constant probability (by the previous claim again) there exists a path of length at most 4 between $\varphi_{i'}(T_{i'})$ and $\varphi_{j'}(T_{j'})$. By continuing this argument towards the root $6 \log n$ times, it follows that with probability $1-n^{-6}$ there exists a path of length at most $20 \log n$. By an union bound argument it follows that with probability $1 - n^{-4}$ every $\varphi_i(T_i)$ and $\varphi_j(T_j)$ which have a vertex in common are connected by a path of length at most $20 \log n$ in T'. ■

Claim 3.6.19 *Let T_i, T_j be two subtrees such that they both visit some vertex $v \in V(T)$. Then, with probability at least $1 - n^{-4}$, for any $v_i \in V(T_i)$, and for any $v_j \in V(T_j)$, $D_T(v_i, v_j) \leq D_{T'}(\varphi_i(v_i), \varphi_j(v_j)) \leq (D_T(v_i, v_j) + 1)O(\log n)$.*

Proof: Observe that since the diameter of the intersection of the two subtrees is at most 2, in order to approximate the distance between $\varphi_i(v_i)$ and $\varphi_j(v_j)$ for all v_i, v_j, it suffices to approximate the distance between $\varphi_i(v)$ and $\varphi_j(v)$. By the previous claim, it easily follows that there a path of length $20 \log n$ that connects $\varphi_i(v)$ to $\varphi_j(v)$. ■

In order to finish the proof, it suffices to consider pairs T_i, T_j that do not intersect. Let T_i, T_j be such a pair of subtrees, and let x_i, x_j be the closest pair of vertices between T_i, and T_j. Let p be the path between x_i to x_j in T. Assume that p visits the subtrees $T_i, T_{t(1)}, \ldots, T_{t(l)}, T_j$. We further assume w.l.o.g., that for each $T_{t(s)}$, p visits at least one edge from $T_{t(s)}$, that does not belong to any other $T_{t(s')}$, with $s \neq s'$. Assume that for each $s \in [l]$, p enters $T_{t(s)}$ in a vertex y_s, and leaves $T_{t(s)}$ at a vertex z_s. We have

$$\begin{aligned}
D_{T'}(\varphi_i(x_i), \varphi_j(x_j)) &= D_{T'}(\varphi_i(x_i), \varphi_{t(1)}(y_1)) + \sum_{s=1}^{l} D_{T'}(\varphi_{t(s)}(y_s), \varphi_{t(s)}(z_s)) + \\
&\quad \sum_{s=1}^{l-1} D_{T'}(\varphi_{t(s)}(z_s), \varphi_{t(s+1)}(y_{s+1})) + D_{T'}(\varphi_{t(l)}(z_l), \varphi_j(x_j)) \\
&\leq O(l \cdot \log n) + \sum_{s=1}^{l} D_T(y_s, z_s) \\
&= O(D_T(x_i, y_i) \log n)
\end{aligned}$$

Similarly to the proof of the above claim, we observe that since the intersection of any two trees is short, and we approximate the distance between the closest pair of T_i, and T_j, it follows that we also approximate the distance between any pair of vertices od T_i, and T_j. ■

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 $C_1 \in \varphi_1(T_1)$, 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 $\varphi_1(T_1)$ with $\varphi_2(T_2)$, 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 $\varphi_1(C_1)$, and $\varphi_2(C_2)$ 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 $(T_{\mathcal{K}}^G, \mathcal{K}_G, \mathcal{A}_G)$ is a (r, c^2)-well-separated decomposition, we can assume the properties of lemmata 3.6.9, and 3.6.11, for $\gamma = c^2$.

Lemma 3.6.20 *For any $A \in \mathcal{A}_G$, G_A c-approximates M.*

Proof: The next claim is similar to a lemma given in [BCIS05], modified for the case of embedding into trees.

Claim 3.6.21 *Let $\alpha > 0$. Let G be an α-restricted subgraph of M, and let G' be an αc-restricted subgraph of M, with $V(G) = V(G')$. If G is connected, then for any $u, v \in V(G)$, $D(u, v) \leq D_{G'}(u, v) \leq cD(u, v)$.*

Proof: Let M' be the restriction of M on $V(G)$. Consider a non-contracting embedding of M' into a tree T' with distortion at most c. Consider an edge $\{u, v\} \in E(T')$. We will first show that $D(u, v) \leq \alpha c$. Let S be a minimum spanning tree of G. If $\{u, v\} \in E(S)$, then since G is connected, it follows that $D(u, v) \leq \alpha$. Assume now that $\{u, v\} \notin E(S)$. Let T_u and T_v be the two subtrees of T', obtained after removing the edge $\{u, v\}$, and assume that T_u contains u, and T_v contains v. Let $p = x_1, \ldots, x_{|p|}$ be the unique path in S with $u = x_1$, and $v = x_{|p|}$. Observe that the sequence of vertices visited by p start from a vertex in T_v, and terminate at a vertex in T_u. Thus, there exists $i \in [|p| - 1]$, such that $v_i \in T_v$, while $v_{i+1} \in T_u$. It follows that the edge $\{u, v\}$ lies in the path from v_i to v_{i+1} in T', and thus $D_{T'}(u, v) \leq D_{T'}(v_i, v_{i+1})$. Since $\{v_i, v_{i+1}\}$ is an edge of S, we have by the above argument that $D(v_i, v_{i+1}) \leq \alpha$. Since the embedding in T has expansion at most c, it follows that $D_{T'}(v_i, v_{i+1}) \leq \alpha c$. Thus, $D_{T'}(u, v) \leq \alpha c$.

Consider now some pair $x, y \in V(G)$. If no vertex is embedded between x and y, then by the above argument, $D(x, y) \leq \alpha c$, and thus the edge $\{x, y\}$ is in G' and $D_{G'}(x, y) = D(x, y)$. Otherwise, let $z_1, \ldots, z_k$ be the vertices appearing in T' between x and y (in this order). Then the edges $\{x, z_1\}, \{z_1, z_2\}, \ldots, \{z_{k-1}, z_k\}, \{z_k, y\}$ all belong to G', and therefore

$$\begin{aligned}
D_{G'}(x, y) &\leq D_{G'}(x, z_1) + D_{G'}(z_1, z_2) + \ldots D_{G'}(z_{k-1}, z_k) + D_{G'}(z_k, y) \\
&= D(x, z_1) + D(z_1, z_2) + \ldots D(z_{k-1}, z_k) + D(z_k, y) \\
&\leq D_{T'}(x, z_1) + D_{T'}(z_1, z_2) + \ldots + D_{T'}(z_{k-1}, z_k) + D_{T'}(z_k, y) \\
&= D_{T'}(x, y) \leq cD(x, y)
\end{aligned}$$

■

By the construction of the set $\mathcal{A}_G$, it follows that a W_G/c^2-restricted subgraph with vertex set A, is connected. Thus, by claim 3.6.21, D_{G_A} c-approximates D. ■

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 incur 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 3.6.22 *The contraction of S is $O(c^7\alpha)$.*

Proof: In order to bound the contraction of S, it is sufficient to bound the contraction between pairs of vertices $x_1, x_2 \in V(G)$, such that either $\{x_1, x_2\} \in S$, or between x_1 and x_2 there are only steiner nodes in S.

We will prove the assertion by induction on the recursive steps of the algorithm. Consider an execution of the recursive procedure RecursiveTree, with input a graph G with maximum edge weight W_G. If G contains only one vertex, then assertion is trivially true. Otherwise, assume that all the recursively computed trees S^A satisfy the assertion.

Consider such a pair $x_1, x_2 \in V(G)$, and assume that in the path from x_1 to x_2 in S, there are $k \geq 0$ steiner nodes. If there exists $A \in \mathcal{A}_G$, such that $x_1, x_2 \in A$, then the assertion follows by the inductive hypothesis.

Assume now that there exist $A_1, A_2 \in \mathcal{A}_G$, with $A_1 \neq A_2$, such that $x_1 \in A_1$, and $x_2 \in A_2$. It follows that $D_S(x_1, x_2) = (k+1)W_G/(c^3\alpha)$. Pick $C_1, C_2 \in V(T)$, and $K_1, K_2 \in \mathcal{K}_G$, such that $x_1 \in K_1 \in C_1$, and $x_2 \in K_2 \in C_2$. We have $D_{T'}(\varphi_1(C_1), \varphi_2(C_2)) = k+1$. By Lemma 3.6.12, we obtain $D_T(C_1, C_2) \leq k+1$. Thus, $D_{T^G_{\mathcal{K}}}(K_1, K_2) \leq (k+2)50c^2$. By Lemma 3.6.6, $D(x_1, x_2) \leq ((k+2)50c^2 + 2)W_G c^2$. Thus, the contraction on x_1, x_2 is $\frac{D_S(x_1,x_2)}{D(x_1,x_2)} \leq \frac{((k+2)50c^2+2)W_G c^2}{(k+1)W_G/(c^3\alpha)} < 104c^7\alpha$. ■

Lemma 3.6.23 *The expansion of S is at most $(c^{O(1)} \log n)^{\log_\alpha \Delta}$.*

Proof: We will prove the assertion by induction on the recursive steps of the algorithm.

Consider an execution of the recursive procedure RecursiveTree, with input a graph G with maximum edge weight W_G. If G contains only one vertex, then the expansion of the computed tree is at most 1. Otherwise, at Step 2 we partition $V(G)$ into $\mathcal{A}_G$, and at Step 3, for each $A \in \mathcal{A}_G$ we define the graph G_A, and recursively execute RecursiveTree on G_A, obtaining an embedding of G_A into a tree S^A. Assume that for each $A \in \mathcal{A}_G$, the expansion on S^A is at most ξ.

Consider $x, y \in V(G)$. Assume that $x \in A_{i_x}$, and $y \in A_{i_y}$, for some $A_{i_x}, A_{i_y} \in \mathcal{A}_G$. If $A_{i_x} = A_{i_y}$, then the expansion is at most ξ, be the inductive hypothesis. We can thus assume that $A_{i_x} \neq A_{i_y}$. Pick $K_x, K_y \in \mathcal{K}_G$, and $C_x, C_y \in V(T)$, such that $x \in K_x \in C_x$, and $y \in K_y \in C_y$. Let p be the path between $\varphi_{i_x}(C_x)$, and $\varphi_{i_y}(C_y)$ in T'.

Let also q be the path from x to y in S. Assume that q visits the sets in $\mathcal{A}_G$ in the order $A_{t_1}, A_{t_2}, \ldots, A_{t_k}$. Let v_i, and v'_i be the first and the last respectively vertex of A_{t_i} visited by q Similarly, let $\varphi_{j_i}(C_i)$, $\varphi_{j_i}(C'_i)$ and be the first, and the last respectively vertex of $\varphi_{j_i}(T_{j_i})$ visited by p. For each $j \in [k]$, pick $K_i, K'_i \in \mathcal{K}_G$, such that $v_i \in K_i$, and $v'_i \in K'_i$.

Let $\delta = W_G/(c^3\alpha)$. We have:

$$\begin{aligned}
D_S(x,y) &= \sum_{j=1}^{k} D_S(v_j, v'_j) + \sum_{j=1}^{k-1} D_S(v_{j'}, v_{j+1}) \\
&\leq \xi \sum_{j=1}^{k} D(v_j, v'_j) + \delta \sum_{j=1}^{k-1} D_{T'}(\varphi_{j_i}(C'_i), \varphi_{j_{i+1}}(C_{i+1})) \\
&\leq \xi W_G c^2 \sum_{j=1}^{k} (2 + D_{T^G_{\mathcal{K}}}(K_j, K'_j)) + 20\delta \log n \sum_{j=1}^{k-1} (1 + D_T(C'_i, C_{i+1})) \\
&\leq \xi W_G c^2 \sum_{j=1}^{k} (2 + 100c^2 D_T(C_j, C'_j)) + 20\delta \log n \sum_{j=1}^{k-1} (1 + D_T(C'_i, C_{i+1})) \\
&\leq (102\xi W_G c^4 + 40\delta \log n) D_T(C_x, C_y) \\
&\leq (102\xi W_G c^4 + 40\delta \log n) D_{T^G_{\mathcal{K}}}(K_x, K_y) \\
&\leq (102\xi W_G c^4 + \frac{40 W_G \log n}{c^3\alpha})(\frac{D(x,y)}{W_G c} + 2)
\end{aligned}$$

Since $A_{i_x} \neq A_{i_y}$, it follows that $D(x,y) \geq \delta = W_G/(c^3\alpha)$. Thus,

$$\begin{aligned} D_S(x,y) &\leq (102\xi W_G c^4 + \frac{40 W_G \log n}{c^3\alpha})(\frac{D(x,y)}{W_G c} + 2c^3\alpha\frac{D(x,y)}{W_G}) \\ &\leq (102\xi c^4 + \frac{40\log n}{c^3\alpha})3c^3\alpha D(x,y) \\ &\leq (306\xi c^7\alpha + 120\log n)D(x,y) \end{aligned}$$

Given a graph of maximum edge weight W_G, the procedure RecursiveTree might perform recursive calls on graphs with maximum edge weight $c^3\delta = W_G/\alpha$. Since the minimum distance in M is 1, and the spread of M is Δ, it follows that the maximum number of recursive calls can be at most $\log\Delta/\log\alpha$. Thus,

$$D_S(x,y) \leq (c^{O(1)}\log n)^{\log_\alpha \Delta} D(x,y)$$

■

Theorem 3.6.24 *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 3.6.22, and 3.6.23, 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 $(c\log n)^{O(\sqrt{\log\Delta})}$. ■

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

3.6.3 The Relation Between Embedding Into Trees and Embedding Into Subtrees

In this section we study the relation between embedding into trees, and embedding into spanning subtrees. More specifically, let $G = (V, E)$ be an unweighted graph.

Assume that G embeds into a tree with distortion c, and also that G embeds into a spanning subtree with distortion c^*.

Clearly, since every spanning subtree is also a tree, we have $c \leq c^*$. We are interested in determining how large the ratio c^*/c can be. We show that for every n_0, there exists $n \geq n_0$, and an n-vertex unweighted subgraph G, for which the ratio is $\Omega(\log n / \log\log n)$. We complement this lower bound by showing that for every unweighted graph G, the ratio is at most $O(\log n)$.

The Lower Bound

In this section we prove a gap between the distortion of embedding graph metrics into trees, and into spanning subtrees. We do this by giving an explicit infinite family of graphs.

Let $n > 0$ be an integer. We define inductively an unweighted graph $G = (V, E)$ with $\Theta(n)$ vertices, and prove that G $O(\log n)$-embeds into a tree, while any embedding of G into a subtree has distortion $\Omega(\log^2 n / \log\log n)$.

Let G_1 be a cycle on $\log n$ vertices. We say that the cycle of G_1 is *at level* 1. Given G_i, we obtain G_{i+1} as follows. For any edge $\{u, v\}$ that belongs to a cycle at level i, but not to a cycle at level $i-1$, we add a path $p_{u,v}$ of length $\log n - 1$ between u and v. We say that the resulting cycle induced by path $p_{u,v}$ and edge $\{u, v\}$ is at level $i+1$.

Let $G = G_{\log n / \log\log n}$. It is easy to see that $|V(G)| = \Theta(n)$. Moreover, every edge of G belongs to either only one cycle of size $\log n$ at level $\log n / \log\log n$, or exactly two cycles of size $\log n$; one at level i, and one at level $i+1$, for some i, with $1 \leq i < \log n / \log\log n$.

We associate with G a tree $T_C = (V(T_C), E(T_C))$, such that $V(T_C)$ is the set of cycles of length $\log n$ of G, and $\{C, C'\} \in E(T_C)$ iff C and C' share an edge. We consider T_C to be rooted at the unique cycle of G at level 1.

Lemma 3.6.25 *Any embedding of G into a subtree has distortion $\Omega(\log^2 n / \log\log n)$.*

Proof: Let T be a spanning subtree of G. Let $k = \log n / \log\log n$. We will compute inductively a set of cycles C, while maintaining a set of edges $E' \subseteq E(G)$. Initially, we set $C = C_1$, where C_1 is the cycle of G at level 1, and $E' = \emptyset$. At each iteration, we consider the subgraph

$$G' = \left(\bigcup_{C \in \mathcal{C}} C \right) \setminus E'.$$

We pick a cycle $C \notin \mathcal{C}$, such that C shares an edge e with G', and we add C in $\mathcal{C}$, and e in E'. Observe that at every iteration G' is a cycle. Thus, we can pick e and C such that $e \notin T$. The process ends when we cannot pick any more such e and C, with $e \notin T$.

Consider the resulting graph $G' = \left(\bigcup_{C \in \mathcal{C}} C \right) \setminus E'$. Since G' is a cycle, it follows that there exists an edge $e' = \{u, v\} \in G'$, such that $e' \notin T$. Since there is no cycle $C' \notin \mathcal{C}$, with $e' \in C'$, it follows that e' belongs to a cycle at level k. Thus, there exists a sequence of length $\log n$ cycles, $K_1, \ldots, K_k$, with $K_1 = C_1$, and $K_k = C'$, and such that $K_i \in \mathcal{C}$, for each i, with $1 \leq i \leq k$, and the there exists a common edge $e_i \in E'$ in K_i and K_{i+1}, for each i, with $1 \leq i < k$.

Consider the sequence of graphs obtained from G after removing the edges $e', e_{k-1}, e_k, \ldots, e_1$, in this order. It is easy to see that after removing each edge, the distance between u and v in the resulting graph increases by at least $\Omega(\log n)$. Since none of there edges is in T, it follows that the distance between u and v in T is at least $k \log n = \log^2 n / \log \log n$. ■

Lemma 3.6.26 *There exists an embedding of G into a tree, with distortion $O(\log n)$.*

Proof: We will construct a tree $T = (V(T), E(T))$ as follows: Initially, we set $V(T) = V(G)$, and $E(T) = \emptyset$. For the cycle C_1 at level 1, we pick an arbitrary vertex $v_{C_1} \in C_1$. Next, for each $u \in C_1$, with $u \neq v_{C_1}$, we add an edge between u and v_{C_1} in T of length $D_G(u, v_{C_1})$.

For every other cycle C' at some level $i > 1$, let $e' = \{u', v'\}$ be the unique edge that C' shares with a cycle C'' at level $i-1$. We pick a vertex $v_{C'}$ arbitrarily between one of the two endpoints of e'. For every vertex $x \in C'$, with $x \neq v_{C'}$, we add an edge between x and $v_{C'}$ in T, of length $D_G(x, v_{C'})$.

Clearly, the resulting graph T is a tree. It is straightforward to verify that for every $\{u, v\} \in E(T)$, $D_T(u, v) = D_G(u, v)$, and thus the resulting embedding is non-contracting. It remains to bound the expansion for any pair of vertices $x, y \in V(G)$. We will consider the following cases.

Case 1. There exists a cycle $C \in V(T_C)$, such that $x, y \in C$: We have

$$\begin{aligned}
D_T(x, y) &= D_T(x, v_C) + D_T(v_C, y) \\
&= D_G(x, v_C) + D_G(v_C, y) \\
&< \log n \\
&\leq D_G(x, y) \log n
\end{aligned}$$

Case 2. There exist $C_x, C_y \in V(T_C)$, with $x \in C_x$, and $y \in C_y$, such that C_y lies on the path in T_C from C_x to the root of T_C: Consider the path $K_1, \ldots, K_l$ in T_C, with $C_x = K_1$, and $C_y = K_l$. For each i, with $1 \leq i < l$, let $e_i = \{x_i, y_i\} \in E(G)$ be the common edge of K_i and K_{i+1}. Note that the shortest path p from x to y in G visits at least one of the endpoints of each edge e_i. Assume w.l.o.g. that p visits $x_1, x_2, \ldots, x_{l-1}$ (in this order). Observe that each i, with $1 \leq i < l$, for each $v \in K_i$ we have either $D_T(x_i, v) = D_G(x_i, v)$, or $D_T(x_i, v) = D_G(x_i, y_i) + D_G(y_i, v) \leq D_G(x_i, v) + 2$. Thus, we obtain

$$\begin{aligned}
D_T(x, y) &\leq D_T(x, x_1) + D_T(x_1, x_2) + \ldots + D_T(x_{l-2}, x_{l-1}) + D_T(x_{l-1}, y) \\
&< D_G(x, x_1) + D_G(x_1, x_2) + \ldots + D_G(x_{l-2}, x_{l-1}) + 2(l-2) + \\
&\quad D_G(x_{l-1}, y) + \log n/2 \\
&< D_G(x, y) + 2 \log n / \log \log n + (\log n)/2 \\
&< D_G(x, y) 3 \log n
\end{aligned}$$

Case 3. There exist $C_x, C_y, C_z \in V(T_C)$, with $x \in C_x$, and $y \in C_y$, such that C_z is the nca of C_x and C_y in T_C: This Case is similar to Case 2.

■

Theorem 3.6.27 *For every $n_0 > 0$, there exists $n \geq n_0$, and an n-vertex unweighted graph G, such that the minimum distortion for embedding G into a tree is $O(\log n)$, while the minimum distortion for embedding G into any of its subtrees is $\Omega(\log^2 n / \log \log n)$.*

Proof: It follows by Lemmata 3.6.25 and 3.6.26. ■

The Upper Bound

We now complement the lower bound given above with an almost matching upper bound for unweighted graphs. The idea is to first use the $O(1)$-approximation algorithm from Section 3.6.2 for embedding unweighted graphs into trees to obtain the clustering $\mathcal{K}_G$. Then, by slightly modifying this clustering, we can guarantee that each cluster induces a connected subgraph of the original graph, and thus it can be easily embedded into a spanning subtree. Next, for each cluster we define a new randomly chosen clustering. This new clustering will be used in the final step to merge the computed subtrees of the clusters, into a spanning subtree of the graph, while losing only a $O(\log n)$ factor in the distortion.

Let $G = (V, E)$ be an unweighted graph, that embeds into an unweighted tree with distortion c. For a subset $V' \subseteq V(G)$, and for every $u, v \in V'$, we denote by $D_{V'}(u, v)$ the shortest path distance between u and v in $G[V']$. If $G[V']$ is disconnected, we can assume that $D_{V'}(u, v) = \infty$.

Consider the set tree-like partition $(T_{\mathcal{K}}^G, \mathcal{K}_G)$ constructed by the algorithm of Section 3.6.2. Let $\mathcal{K}_G = \{K_{r_1}, K_{r_2}, \ldots\}$, and assume that $T_{\mathcal{K}}^G$ is rooted at K_r.

Let $F_{\mathcal{K}}$ be the forest obtained by removing from $T_{\mathcal{K}}^G$ all the edges between vertices at levels $21j$ and $21j + 1$, for all j, with $1 \leq j < \lfloor \text{depth}(T_{\mathcal{K}}^G)/21 \rfloor - 1$. Let $C(F_{\mathcal{K}})$ be the set of connected components of $F_{\mathcal{K}}$. Let

$$\mathcal{J} = \bigcup_{A \in C(F_{\mathcal{K}})} \left\{ \bigcup_{K_i \in A} K_i \right\}.$$

Clearly, $\mathcal{J}$ is a partition of $V(G)$. Let $T_{\mathcal{J}}$ be the tree on vertex set $\mathcal{J}$, where the edge

$\{J_i, J_j\}$ is in $T_{\mathcal{J}}$ if there exist $\{K_{i'}, K_{j'}\} \in E(T_{\mathcal{K}}^G)$, such that $K_{i'} \in J_i$, and $K_{j'} \in J_j$. We consider $T_{\mathcal{J}}$ as being rooted at a vertex $J_r \in \mathcal{J}$, where $K_r \in J_r$.

Lemma 3.6.28 *For each $J_i \in \mathcal{J}$, $G[J_i]$ is connected.*

Proof: Assume w.l.o.g., that J_i is the union of sets of vertices K_j, for all $K_j \in A$, where $A \in C(F_{\mathcal{K}})$ is a subtree of $T_{\mathcal{J}}$. Assume that $K_{r'}$ is the vertex of A that is closest to K_r in $T_{\mathcal{K}}^G$. Let p_l be the unique path in A from $K_{r'}$ to a leaf K_l of A. Let also $J_i^l = \bigcup_{K_k \in p_l} K_k$. It suffices to show that for each leaf l, the induced subgraph $G[J_i^l]$ is connected.

Let $p_l = K_1, K_2, \ldots K_t$, where $K_{r'} = K_1$, and $K_l = K_t$. Note that $t \geq 21$. Assume now that $G[J_i^l]$ is disconnected, and let $C(G[J_i^l])$ be the set of connected components of $G[J_i^l]$.

Claim 3.6.29 *There exists t', with $1 \leq t' \leq t$, and $C_1 \neq C_2 \in C(G[J_i^l])$, such that $K_{t'} \cap C_1 \neq \emptyset$, and $K_{t'} \cap C_2 \neq \emptyset$.*

Proof: Assume that the assertion in not true. That is, for each t', with $1 \leq t' \leq t$, $K_{t'}$ is contained in a connected component $C'_{t'} \in C(G[J_i^l])$. Observe that for each t'', with $1 \leq t'' < t$, there exists at least one edge between $K_{t''}$ and $K_{t''+1}$. This means that all the $C'_{t'}$s are in fact the same connected component, and thus $C(G[J_i^l])$ contains a single connected component. It follows that J_i^l is connected, a contradiction. ■

Claim 3.6.30 *There exist $C_1, C_2 \in C(G[J_i^l])$, such that $K_{11} \cap C_1 \neq \emptyset$, and $K_{11} \cap C_2 \neq \emptyset$.*

Proof: Let t', with $1 \leq t' \leq t$, and $C_1, C_2 \in C(G[J_i^l])$ be given by Claim 3.6.29. If $t' = 11$, then there is nothing to prove.

Otherwise, pick $v_1 \in K_{t'} \cap C_1$, and $v_2 \in K_{t'} \cap C_2$. By the construction of $\mathcal{K}$, we have that there exists a path p from v_1 to v_2, such that p is the concatenation of the paths $q_{t'}, \ldots, q_1, q, q'_1, \ldots, q'_{t'}$, where for each $i \in [1, t']$, q_i and q'_i are paths of length at most c in K_i. Moreover, there exists a path $\bar{p}$ from v_1 to v_2, such that $\bar{p}$ is the concatenation of the paths $w_{t'}, \ldots, w_t, w, w'_t, \ldots, w'_{t'}$, where for each $i \in [t', t]$, w_i and w'_i are paths of length at most c in K_i.

If $t' > 11$, then pick $v'_1 \in q_{11}$, and $v'_2 \in q'_{11}$. Otherwise, if $t' < 11$, pick $w'_1 \in q_{11}$, and $v'_2 \in w'_{11}$. Clearly, in both cases we have $v'_1 \in C_1$, and $v'_2 \in C_2$. ■

Let now $C_1, C_2 \in C(G[J_i^l])$ be the connected components given by Claim 3.6.30. Pick $v_1 \in K_{t'} \cap C_1$, and $v_2 \in K_{t'} \cap C_2$. Let p be the shortest path between v_1 and v_2 in G. We observe that there are two possible cases for p:

Case 1: p is the concatenation of the paths $q_{11}, \ldots, q_1, q, q'_1, \ldots, q'_{11}$, where for each $i \in [1, 11]$, q_i and q'_i are contained in K_i.

Case 2: p is the concatenation of the paths $q_{11}, \ldots, q_t, q, q'_t, \ldots, q'_{11}$, where for each $i \in [11, t]$, q_i and q'_i are contained in K_i.

Since the above two Cases can be analyzed identically, we assume w.l.o.g. that p satisfies Case 1. Observe that for each $i \in [1, 11)$, each q_i and each q'_i visits c vertices of K_i. It follows that the length of p is greater than $20c$, contradicting Lemma 3.6.5.

■

For each $J_i \in \mathcal{J}$, we define a set $\mathcal{J}_i$ of subsets of J_i as follows. First, we pick a vertex $r_i \in J_i$, and we construct a BFS tree T_{J_i} of $G[J_i]$, rooted at r_i. Note that by Lemma 3.6.28, $G[J_i]$ is connected, and thus there exists such a BFS tree. We also pick an integer $\alpha_{J_i} \in [0, 100c)$, uniformly at random. Let F_{J_i} be the forest obtained from T_{J_i} by removing the edges between vertices at levels $100cj + \alpha_{J_i}$ and $100cj + \alpha_{J_i} + 1$, for all j, with $1 \leq j < \left\lfloor \frac{\text{depth}(T_{J_i})}{100c} \right\rfloor - 2$. The set $\mathcal{J}_i$ can now be defined as the set of sets of vertices of the connected components of F_{J_i}. Clearly, $\mathcal{J}_i$ is a partition of J_i.

Lemma 3.6.31 *For each $J_i, J_j \in \mathcal{J}$, such that J_i is the parent of J_j in $T_{\mathcal{J}}$, and for each $J_{j,k} \in \mathcal{J}_j$, there exist $u \in J_i$, and $v \in J_{j,k}$, such that $\{u, v\} \in E(G)$.*

Proof: It is easy to verify by the construction of $\mathcal{K}_G$ that J_j is a subset of the vertices of at least $21c$, and at most $42c$ consecutive levels of a BFS tree of G. Let $l_1, \ldots, l_t$ be these levels, where l_1 is the level closest to the root of the BFS tree of G. For every vertex $x \in J_j$, there exists a vertex $y \in J_i$, such that $\{x, y\} \in E(G)$, iff $x \in l_1$. Thus, it suffices to show that for every $J_{j,k} \in \mathcal{J}_j$, $J_{j,k} \cap l_1 = \emptyset$.

It is easy to verify that for every $v \in J_j$, there exists $u \in l_1$, such that $D_{J_j}(v,u) < 42c$. In the construction of $\mathcal{J}_j$, we pick a vertex $r_j \in J_j$, and we compute a BFS tree T' of G_{J_j}. Every $J_{j,k} \in \mathcal{J}_j$ is a subtree $T_{j,k}$ of T' rooted at a vertex $r_{j,k}$. $T_{j,k}$ contains all the predecessors of $r_{j,k}$ that are at distance at most $\delta_{j,k}$, for some $100c \leq \delta_{j,k} \leq 200c$. Assume now that there is no vertex of l_1 in the $42c$ first levels of $T_{j,k}$. Pick a vertex of $T_{j,k}$ at level $42c+1$. By the above argument, there exists a vertex $u \in l_1$ that is at distance at most $42c$ from v. This implies that u is contained within the $84c+1$ first levels of $T_{j,k}$. Thus, $T_{j,k} \cap l_1 \neq \emptyset$, and $J_{j,k} \cap l_1 \neq \emptyset$. ∎

Lemma 3.6.32 *For each $J_i, J_j \in \mathcal{J}$, such that J_i is the parent of J_j in $T_{\mathcal{J}}$, and for each $u, v \in J_i$, and $u', v' \in J_j$, such that $\{u,u'\} \in E(G)$, and $\{v,v'\} \in E(G)$, $D_{J_i}(u,v) \leq 90c$.*

Proof: Note that the partition $\mathcal{K}_G$ is obtained on a BFS tree of G with root some $r \in V(G)$. If $r \in J_i$, then $D_{J_i}(u,v) \leq D_{J_i}(u,r) + D_{J_i}(r,v) \leq 84c$.

It remains to consider the case $r \notin V(G)$. This implies that there exists $J_k \in \mathcal{J}$, such that J_k is the parent of J_i in $T_{\mathcal{J}}$. Assume that the assertion is not true. That is, there exist $u, v \in J_i$, and $u', v' \in J_j$, with $\{u,u'\} \in E(G)$, $\{v,v'\} \in E(G)$, and $D_{J_i}(u,v) > 90c$. By the construction of $\mathcal{K}_G$, and since $r \notin J_i$ it follows that there exist $w, z \in J_i$, and $w', z' \in J_k$, with $\{w,w'\} \in E(G)$, and $\{z,z'\} \in E(G)$, and moreover there exists a shortest path p_1 in G from w to u, and a shortest path p_2 from v to z in G, such that p_1 and p_2 are contained in J_i. It is easy to verify that the length of each of the paths p_1 and p_2 is at least $22c$.

Furthermore, there exists a path p_3 from w' to z', and a path p_4 from u' to v', such that both p_3 and p_4 do not visit J_i. Let p_3' be the path obtained from p_3 by adding the edges $\{w,w'\}$, and $\{z',z\}$. Similarly, let p_4' be the path obtained from p_4 by adding the edges $\{u,u'\}$, and $\{v',v\}$.

Let x_1 be a vertex of p_1 such that $D_G(x_1,u) > 5c$, and $D_G(x_1,w) > 5c$. Similarly, let x_2 be a vertex of p_2 such that $D_G(x_2,v) > 5c$, and $D_G(x_2,z) > 5c$. We need to define the following set of paths:

- Let q_1 be the subpath of p_1 from u to x_1.

- Let q_2 be the path obtained by concatenating the subpath of p_1 from x_1 to w, with p_3.
- Let q_3 be the subpath of p_2 from z to x_2.
- Let q_4 be the path obtained by concatenating the subpath of p_2 from x_2 to v, with p_4.

It is straight-forward to verify that $D_G(q_1, q_3) > 5c$, and $D(q_2, q_4) > 5c$. By applying Lemma 3.6.1, we obtain that the optimal distortion for embedding G into an unweighted tree is more than $5c$, a contradiction. ■

Theorem 3.6.33 *If an unweighted graph G can be embedded into a tree with distortion c, then G can be embedded into a subtree with distortion $O(c \log n)$.*

Proof: We can compute an embedding of G into a subtree T as follows. Initially, we set T equal to the empty subgraph. We pick a vertex $r \in V(G)$, and we compute a (r, c)-partition of G. We compute the partition $\mathcal{J}$, and for each $J_i \in \mathcal{J}$, we compute the partition $\mathcal{J}_i$, as described above. For each $J_i \in \mathcal{J}$, and for each $J_{i,j} \in \mathcal{J}_i$, we add to T a spanning tree of $J_{i,j}$ of radius $O(c)$.

It remains to connect the subtrees by adding edges between the sets $J_{i,j}$. Observe that if $r \in J_i$, then $\mathcal{J}_i$ contains a single set $J_{i,j}$.

Assume now that $r \notin J_j$, and let J_i be the parent of J_j in $T_{\mathcal{J}}$. By Lemma 3.6.31, for each $J_{j,k} \in \mathcal{J}_j$, there an edge between $J_{j,k}$ and J_i in G. For each such $J_{j,k}$, we pick one such edge, uniformly at random, and we add it to T.

Consider now two subsets $J_{j,k}, J_{j,l} \in \mathcal{J}_j$. It is easy to see that $J_{j,k}$, and $J_{j,l}$ get connected to the same subset $J_{i,t} \in \mathcal{J}_i$, with probability at least $1 - \frac{90c}{100c} = \Omega(1)$. Thus, the probability that two such subsets have not converged to the same subset in an ancestor after $O(\log n)$ levels is at most 1/poly(n). Since there are at most n^2 pairs of such subsets $J_{i,j}$, it follows that the above procedure results in a tree with distortion $O(c \log n)$ with high probability. ■

Chapter 4

Ordinal embeddings

Credits: The results in this section is work done with Noga Alon, Erik Demaine, Martin Farach-Colton, MohammadTaghi Hajiaghayi, and Anastasios Sidiropoulos, and has appeared in SODA'05.

In this chapter, we introduce a new notion of embedding, called *minimum-relaxation ordinal embedding*, parallel to the standard notion of minimum-distortion (metric) embedding. In an ordinal embedding, it is the relative order between pairs of distances, and not the distances themselves, that must be preserved as much as possible. The (multiplicative) relaxation of an ordinal embedding is the maximum ratio between two distances whose relative order is inverted by the embedding. We develop several worst-case bounds and approximation algorithms on ordinal embedding. In particular, we establish that ordinal embedding has many qualitative differences from metric embedding, and capture the ordinal behavior of ultrametrics and shortest-path metrics of unweighted trees.

4.1 Introduction

The classical field of *multidimensional scaling (MDS)* has witnessed a surge of interest in recent years with a slew of papers on *metric embeddings*; see e.g. [IM04]. The problem of multidimensional scaling is that of mapping points with some measured pairwise distances into some target metric space. Originally, the MDS community considered embeddings into an ℓ_p space, with the goal of aiding in visualization, compression, clustering, or nearest-neighbor searching; thus, low-dimensional embeddings were sought. An *isometric embedding* preserves all distances, while more generally, *metric embeddings* tradeoff the dimension with the fidelity of the embeddings.

Note, however, that the distances themselves are not essential in nearest-neighbor searching and many contexts of visualization, compression, and clustering. Rather, the order of the distances captures sufficient information, that is, we might only need an embedding into a metric space with any monotone mapping of the distances. Such embeddings were heavily studied in the early MDS literature [CS74, Kru64a, Kru64b, She62a, She62b, Tor52] and have been referred to as *ordinal embeddings*, *nonmetric MDS*, or *monotone maps*. Here, we use the first term.

While the early work on ordinal embeddings was largely heuristic, there has been some work with mathematical guarantees since then. Define a *distance matrix* to be any matrix of pairwise distances, not necessarily describing a metric. In [SFC04], it was shown that it is NP-hard to decide whether a distance matrix can be ordinally embedded into an additive metric, i.e., the shortest-path metric in a tree. Define the *ordinal dimension* of a distance matrix to be the smallest dimension of a Euclidean space into which the matrix can be ordinally embedded. Bilu and Linial [BL04] have shown that every matrix has ordinal dimension at most $n-1$. They also applied the methods of [AFR85] to show that (in a certain well-defined sense) almost every n-point metric space has ordinal dimension $\Omega(n)$. Because ultrametrics can be characterized by the order of distances on all triangles, they are closed under monotone mappings. Holman [Hol72] showed that every n-point ultrametric can be isometrically embedded into $(n-1)$-dimensional Euclidean space and that $n-1$ dimensions are necessary. Combined with the closure property just noted, this shows that the ordinal dimension of every ultrametric is exactly the maximal $n-1$.[1]

Relaxations of ordinal embeddings have involved problems of deciding the realiza-

[1]This observation settles an open problem posed in [BL04] asking for the worst-case ordinal dimension of any metric on n points, which they showed was between $n/2$ and $n-1$. Ultrametrics show that the answer is $n-1$.

tion of partial orders. For example, Opatrny [Opa79] showed that it is NP-hard to decide whether there is an embedding into one dimension satisfying a partial order that specifies the maximum edge for some triangles. Such partial orders on triangles are called *betweenness constraints*. Chor and Sudan [CS98] gave a 1/2-approximation for maximizing the number of satisfied constraints. It is also NP-hard to decide whether there is an embedding into an additive metric that satisfies a partial order defined by the total order of each triangle [SFC04].

4.1.1 Our Results

We take a different approach. We define a metric M' to be an *ordinal embedding with relaxation* $\alpha \geq 1$ of a distance matrix M if $\alpha M[i,j] < M[k,l]$ implies $M'[i,j] < M'[k,l]$. In other words, significantly different distances have their relative order preserved. Note that in an ordinary ordinal embedding, we must respect distance equality, while in an ordinal embedding with relaxation 1, we may break ties. It is now natural to minimize the relaxation needed to embed a distance matrix M into a target family of metric spaces. Here we optimize the confidence with which we make an ordinal assertion, rather than the number of ordinal constraints satisfied.

In this chapter, we prove a variety of results about the *Ordinal Relaxation Problem*. We show that the best relaxation achievable is always at most the best *distortion* of a metric embedding. Furthermore, while the optimal relaxation is bounded by the ratio between the largest and smallest distances in M, the optimal distortion can grow arbitrarily. Indeed, the ratio between the optimal relaxation and distortion can be arbitrarily large even when embedding into the line, and can be infinite when embedding into cut metrics. (We also give a polynomial-time algorithm to compute the best ordinal embedding into a cut metric.) We show that, if the target class of the embedding is ultrametrics, the relaxation and distortion are equal, and the optimal embedding can be computed in polynomial time. More surprisingly, we show that ultrametrics are the only target metrics for which all distance matrices have a bounded ratio between the best distortion and the best relaxation.

We demonstrate many more differences between ordinal and metric embeddings.

While any metric can be isometrically embedded into ℓ_∞, there are four-point metrics that cannot be so embedded into any ℓ_p, $p < \infty$. In contrast, we show that it is possible to ordinally embed any distance matrix into ℓ_p for any fixed $1 \le p \le \infty$. We show that the shortest-path metric of an unweighted tree can be ordinally embedded into d-dimensional Euclidean space with relaxation $\tilde{O}(n^{1/d})$. We also show that relaxation $\Omega(n^{1/(d+1)})$ is sometimes necessary. In contrast, the best bounds on the worst-case distortion required are $O(n^{1/(d-1)})$ and $\Omega(n^{1/d})$ [Gup00b]. The proof techniques required for the ordinal case are also substantially different (in particular because the usual "packing" arguments fail) and lead to approximation algorithms described below. We show that ultrametrics can be ordinally embedded into $O(\lg n)$-dimensional ℓ_p space with relaxation 1. In contrast, the best known metric embedding of ultrametrics into $c \lg n$-dimensional space has distortion $1+\Omega(1/\sqrt{c})$ [BM04a], and ordinary (no-relaxation) ordinal embeddings require $n-1$ dimensions. For general metrics, we show a lower bound of $\Omega(\lg n/(\lg d + \lg\lg n))$ on the relaxation of any ordinal embedding into d-dimensional ℓ_p space for fixed integers p or $p = \infty$. In particular, for $d = \Theta(\lg n)$, this lower bound is $\Omega(\lg n/\lg\lg n)$, leaving a gap between the upper bound of $O(\lg n)$ which follows from Bourgain embedding. In contrast, for metric embeddings, there is an $\Omega(\lg n)$ lower bound on distortion for $d = \Theta(\lg n)$ [LLR95, Mat97].

We also develop approximation algorithms for finding the minimum possible relaxation for an ordinal embedding of a specified metric. Specifically, we give a 3-approximation for ordinal embedding of the shortest-path metric of a specified unweighted tree into the line. In contrast, only $O(n^{1/3})$-approximation algorithms are known for the same problem with distortion [BDG+05]. In general, approximation algorithms for embedding are a central challenge in the field, and few are known [HIL98, Iva00, B$\breve{0}$3, ABFC+96, FCK96, BDHI04]. We also expect that our techniques will extend to obtain approximation algorithms for more general ordinal embedding problems.

4.2 Definitions

In this section, we define ordinal embeddings and relaxation, as well as the standard notions of metric embeddings and distortion.

Consider a finite metric $D : P \times P \to [0, \infty)$ on a finite point set P—the *source metric*—and a class $\mathcal{T}$ of metric spaces $(T, d) \in \mathcal{T}$ where d is the distance function for space T—the *target metrics.* An *ordinal embedding (with no relaxation)* of D into $\mathcal{T}$ is a choice $(T, d) \in \mathcal{T}$ of a target metric and a mapping $\varphi : P \to T$ of the points into the target metric such that every comparison between pairs of distances has the same outcome: for all $p, q, r, s \in P$, $D(p, q) \leq D(r, s)$ if and only if $d(\varphi(p), \varphi(q)) \leq d(\varphi(r), \varphi(s))$. Equivalently, φ induces a monotone function $D(p, q) \mapsto d(\varphi(p), \varphi(q))$, and for this reason ordinal embeddings are also called *monotone embeddings.* An *ordinal embedding with relaxation* α of D into $\mathcal{T}$ is a choice $(T, d) \in \mathcal{T}$ and a mapping $\varphi : P \to T$ such that every comparison between pairs of distances not within a factor of α has the same outcome: for all $p, q, r, s \in P$ with $D(p, q)/D(r, s) > \alpha$, $d(\varphi(p), \varphi(q)) > d(\varphi(r), \varphi(s))$. Equivalently, we can view a relaxation α as defining a partial order on distances $D(p, q)$, where two distances $D(p, q)$ and $D(r, s)$ are comparable if and only if they are not within a factor of α of each other, and the ordinal embedding must preserve this partial order on distances.

An ordinal embedding with relaxation 1 is a different notion from ordinal embedding with no relaxation, because the former allows violation of equalities between pairs of distances. Indeed, we will show in Section 4.6.1 that the two notions have major qualitative differences. We define ordinal embedding with relaxation in this way, instead of making the $> \alpha$ inequality non-strict, because otherwise our notion of relaxation 1 would have to be phrased as "relaxation $1 + \epsilon$ for any $\epsilon > 0$". Another consequence is that we can define the minimum possible relaxation $\alpha^* = \alpha^*(D, \mathcal{T})$ of an ordinal embedding of D into $\mathcal{T}$, instead of having to take an infimum. (The infimum will be realized provided the space $\mathcal{T}$ is closed.)

We pay particular attention to contrasts between ordinal embedding and "standard" embedding, which we call "metric embedding" for distinction. A *contractive* *metric embedding with distortion* c of a source metric D into a class $\mathcal{T}$ of target metrics is a choice $(T, d) \in \mathcal{T}$ and a mapping $\varphi : P \to T$ such that no distance increases and every distance is preserved up to a factor of c: for all $p, q \in P$, $1 \leq D(p,q)/d(\varphi(p), \varphi(q)) \leq c$. Similarly, we can define an *expansive metric embedding with distortion* c with the inequality $1 \leq d(\varphi(p), \varphi(q))/D(p,q) \leq c$. When $c = 1$, these two notions coincide to require exact preservation of all distances; such an embedding is called a *metric embedding with no distortion* or an *isometric embedding*. In general, $c^* = c^*(D, \mathcal{T})$ denotes the minimum possible distortion of a metric embedding of D into $\mathcal{T}$. (This definition is equivalent for both contractive and expansive metric embeddings, by scaling.)

4.3 Comparison between Distortion and Relaxation

The following propositions relate α^* and c^*.

Proposition 4.3.1 *For any source and target metrics, $\alpha^* \leq c^*$.*

Proof: Consider a contractive metric embedding φ into (T, d) with distortion c. We show that φ is also an ordinal embedding into (T, d) with relaxation $\alpha \leq c$. Consider a pair of distances $D(p,q)$ and $D(p',q')$ with ratio $D(p,q)/D(p',q')$ larger than c. (Thus, in particular, we label p, q, p', q' so that $D(p,q) > D(p',q')$.) Then $d(\varphi(p), \varphi(q))/d(\varphi(p'), \varphi(q')) \geq D(p,q)/(cD(p',q'))$ by expansiveness of $D(p,q)$ and distortion of $D(p',q')$. Thus $d(\varphi(p), \varphi(q))/d(\varphi(p'), \varphi(q')) > 1$, so $d(\varphi(p), \varphi(q)) > d(\varphi(p'), \varphi(q'))$ as desired. ∎

Next we show that c^* and α^* can have an arbitrarily large ratio, even when the target metric is the real line.

Proposition 4.3.2 *Embedding a uniform metric (where $D(p,q) = 1$ for all $p \neq q$) into the real line has $c^* = n - 1$ and $\alpha^* = 1$.*

Proof: The mapping $\varphi(p) = 0$, for all $p \in P$, is an ordinal embedding with no relaxation, because every distance remains equal (albeit 0). Any expansive metric embedding into the real line must have distance at least 1 between consecutively embedded points, so the entire embedding must occupy an interval of length at least $n-1$. The two points embedded the farthest away from each other therefore have distance at least $n-1$, for a distortion of at least $n-1$. On the other hand, any embedding in which consecutively embedded points have distance exactly 1 has distortion $n-1$. ■

Next we give a general bound on α^* that is essentially always finite. Define the *diameter* $\operatorname{diam}(D)$ of a metric D to be the ratio of the maximum distance to the minimum distance. (If the minimum distance is zero and the maximum distance is positive, then $\operatorname{diam}(D) = \infty$; if both are zero, then $\operatorname{diam}(D) = 1$.)

Proposition 4.3.3 *For any source metric D and any target metrics, $\alpha^* \leq \operatorname{diam}(D)$.*

Proof: The mapping $\varphi(p) = 0$, for all $p \in P$, has ordinal relaxation $\operatorname{diam}(G)$, because all non-equal comparisons between distances are violated, and the largest ratio between any two distances is precisely $\operatorname{diam}(D)$. ■

No such general finite upper bound exists for c^*, as evidenced by "cut metrics". A *cut metric* is defined by a partition $P = A \cup B$ of the point set P into two disjoint sets A and B. The metric assigns a distance of 0 between pairs of points in A and pairs of points in B, and assigns a distance of 1 between other pairs of points. If the source metric D has no zero distances and the target metrics are the cut metrics, then $c^* = \infty$, because some distance must become 0 which requires infinite distortion.

In contrast, α^* remains at most $\operatorname{diam}(D)$, and in some sense measures the quality of a clustering of the points into two clusters. Furthermore, the optimal α^* and clustering can be computed efficiently:

Proposition 4.3.4 *The minimum-relaxation ordinal embedding of a specified metric into a cut metric can be computed in polynomial time.*

Proof: First we guess the optimal relaxation α^* among $O(n^4)$ possibilities (the ratio of the distance between any two pairs of points). Second we guess a pair (p, q) of points on different sides of the cut and with minimum distance $D(p, q)$. Thus all pairs (r, s) of points with smaller distance $D(r, s) < D(p, q)$ must have r and s on the same side of the cut. Also, if there is any ordinal embedding of relaxation α^*, there cannot be pairs (r, s) of points with distance larger by a factor of α^*, i.e., with $D(r, s) > \alpha D(p, q)$, because such distances will be mapped to a distance smaller or equal to 1, the mapped distance of (p, q). Similarly, there cannot be pairs (r, s) and (r', s') of points with distance less than $D(p, q)$ and with $D(r, s) > \alpha D(r', s')$, because those pairs are forced to map to equal distances of 0. Finally, all pairs (r, s) of points with $D(p, q) \leq D(r, s) \leq \alpha D(p, q)$ must have r and s on different sides of the cut if there is another distance $D(r', s') < D(r, s)/\alpha$, and otherwise are unconstrained.

All constraints of the form "r and s must be on the same side of the cut" and "r and s must be on different sides of the cut" can be phrased as a 2-SAT instance. Each point r has a variable x_r which is 0 if it placed in set A and 1 if it placed in set B. Each constraint thus has the form $x_r = x_s$ or $x_r \neq x_s$, which can be phrased in 2-CNF. Thus we can find an ordinal embedding into a cut metric with relaxation at most the guessed value of α^*, if one exists. ■

Next we consider the related problem of ordinal embedding into the real line, which is a generalization of cut metrics. First we show that we can decide whether $\alpha^* = 1$ in this case. The algorithm requires more sophistication (namely, guessing) than the trivial algorithm for metric embedding with distortion 1, where one can incrementally build an embedding in any Euclidean space in linear time.

Proposition 4.3.5 *In polynomial time, we can decide whether a given metric can be ordinally embedded into the line with relaxation* 1*.*

Proof: The algorithm guesses the leftmost point p and greedily places every point q at position $D(p, q)$ on the line. (In particular, the algorithm places p at position 0.) It is easy to show that this embedding has ordinal relaxation 1 whenever such an embedding exists. ■

Next we consider the worst case for ordinal embedding into the line. We show in particular that the cycle requires large relaxation. The cycle also requires large distortion into the line, but the proof technique for ordinal relaxation is very different from the usual "packing argument" that suffices for metric distortion.

Proposition 4.3.6 *Ordinal embedding of the shortest-path metric of an unweighted cycle of even length n into the line requires relaxation at least $n/2$.*

Proof: Suppose to the contrary that there is an ordinal embedding φ of the cycle into the line with relaxation less than $n/2$. Label the vertices of the cycle 1 through n in cyclic order. Assume without loss of generality that $\varphi(1) < \varphi(n/2+1)$. We must also have $\varphi(2) < \varphi(n/2+1)$, because otherwise $|\varphi(2) - \varphi(1)| \geq |\varphi(n/2+1) - \varphi(1)|$, contradicting that $\alpha < n/2$. Similarly, $\varphi(2) < \varphi(n/2+2)$, because otherwise $|\varphi(n/2+2) - \varphi(n/2+1)| \geq |\varphi(n/2+2) - \varphi(2)|$, again contradicting that $\alpha < n/2$. Repeating this argument shows that $\varphi(3) < \varphi(n/2+3)$, etc., and finally that $\varphi(n/2+1) < \varphi(1)$, a contradiction. ∎

Section 4.5 shows that some trees also require $\Omega(n)$ ordinal relaxation into the line.

4.4 ℓ_p Metrics are Universal

In this section we show that every distance matrix can be ordinally embedded without relaxation into ℓ_p space of a polynomial number of dimensions, for any fixed $1 \leq p \leq \infty$. This result is surprising in comparison to metric embeddings. Every metric can be embedded into ℓ_p using $O(\lg n)$ distortion [Bou85, LLR95], and in the worst case $\Omega(\lg n)$ distortion is necessary for any $p < \infty$, as proved in [LLR95] for $p = 2$ and in [Mat97] for all other values of p. In particular, the shortest-path metric of a constant-degree expander graph requires $\Omega(\lg n)$ distortion.

Theorem 4.4.1 *Every distance matrix can be ordinally embedded without relaxation into $O(n^5)$-dimensional ℓ_p space, for any fixed $1 \leq p \leq \infty$.*

The same result was established independently in [BL04] using an algebraic proof. Specifically, they show that every distance matrix can be ordinally embedded into $(n-1)$-dimensional Euclidean space, and then use the property that any Euclidean metric can be isometrically embedded into any ℓ_p space with at most $\binom{n}{2}$ dimensions. In constrast, our proof is purely combinatorial.

We can also reduce the number of dimensions for some values of p. For example, for $p = 2$, a simple rotation reduces the number of dimensions to $n - 1$.

Our proof proceeds in two steps. First we show that 0/1 Hamming metrics are universal in the same sense as Theorem 4.4.1. To conclude the proof, we note that there is an ordinal embedding without relaxation from 0/1 Hamming metrics into any ℓ_p metric. In fact, the pth root of the distances in a 0/1 Hamming metric can be metrically embedded without distortion into ℓ_p with the same number of dimensions. This second part is merely an observation, so the main work is in showing that 0/1 Hamming metrics are universal:[2]

Lemma 4.4.2 *Every distance matrix can be ordinally embedded without relaxation into a 0/1 Hamming metric with $O(n^5)$ dimensions. In other words, any desired ordering on the distances between pairs of n points can be realized by a 0/1 Hamming metric on those n points.*

Proof: Given a partial order $\mathcal{P}$ on a set of distances, we construct a 0/1 Hamming metric H such that $P_{i,j} < P_{k,l}$ implies $H_{i,j} < H_{k,l}$. If $\mathcal{P}$ is non-total, then we can take any topological sort of $\mathcal{P}$ and realize it as a Hamming metric. This ordinal embedding will satisfy the original partial order, so from now on, we assume that $\mathcal{P}$ is a total order. Because $\mathcal{P}$ is an order on distances, defined by pairs of points, we can define it as a sequence of pairs $\mathcal{P} = [(a_0, b_0), (a_1, b_1), \ldots, (a_{\binom{n}{2}}, b_{\binom{n}{2}})]$, where in each pair, we arbitrarily select which node is a and which is b.

We now must produce a 0/1 vector for each point of the space so that the Hamming metric induced preserves the order $\mathcal{P}$. We assume that n is a power of 2; otherwise we can simply round n up to the next power of two.

[2]Note that finite 0/1 Hamming metrics and finite Hamming metrics are essentially the same, because one can be converted into the other with a dimension blowup that is multiplicative in the number of points. Thus our result could have been established with general Hamming metrics. However, our construction directly yields a 0/1 Hamming metric, so we do not need this extra conversion detail.

Our main tool will be Hadamard matrices, defined as follows. Let $H_0 = [0]$, and

$$H_{i-1} = \begin{pmatrix} H_i & H_i \\ H_i & \overline{H_i} \end{pmatrix}$$

where $\overline{H_i}$ is the bitwise negation of H_i. Notice that the first row is the all-0 vector, denoted $\vec{0}$. Also, each row other than the first row consists of half 0s and half 1s. More strikingly, any two rows of H_i have Hamming distance 2^{i-1}, that is, they differ in half their positions. Finally, $\vec{1}$ has Hamming distance 2^{i-1} from any row except the first row, with which it has Hamming distance 2^i.

We generate a set of dimensions that code for each distance $\mathcal{P}_i$ and concatenate all the dimensions at the end. To code for distance $\mathcal{P}_i = (a_i, b_i)$, we set a_i's bits to be 0^{in} and b_i's bits to be 1^{in}. Every other point in the space besides these two gets a distinct row from the Hadamard matrix, repeated i times. Now the induced distances are $in/2$ for any pair of points except for a_i and b_i, which are at distance in.

Let the total number of dimensions be $d = n\binom{n}{2}(\binom{n}{2} + 1)/2$.

Consider now the distances between any pair a and b resulting from the concatenation of all d dimensions, and assume that $a = a_i$ and $b = b_i$, that is, their pairwise distance is the ith in the list. Then their pairwise distance is $(d + in)/2$. Thus, this embedding assigns to the ith smallest distance in $\mathcal{P}$ the ith smallest distance in the Hamming metric. ∎

4.5 Approximation Algorithms for Unweighted Trees into the Line

In this section, we give a 3-approximation algorithm for ordinally embedding the shortest-path metric induced by an unweighted tree into the line with approximately minimum relaxation. In contrast, the best approximation algorithm known for metrically embedding trees into the line with approximately minimum distortion is a recently discovered $O(n^{1/3})$-approximation [BDG+05].

First we find a structure for proving lower bounds on the optimal relaxation:

Lemma 4.5.1 *Given n such that 3 divides $n-1$, ordinal embedding of the shortest-path metric of an unweighted 3-spider with $(n-1)/3$ vertices on each leg of the spider (i.e., a 3-star with each edge subdivided into a path of $(n-1)/3$ edges) requires relaxation at least $(n-1)/3$.*

Proof: Suppose to the contrary that there is an ordinal embedding φ of the 3-spider into the line with relaxation $\alpha < (n-1)/3$. Label the vertices as follows: 0 denotes the root, and $a_1, \ldots, a_{(n-1)/3}$, $b_1, \ldots, b_{(n-1)/3}$, and $c_1, \ldots, c_{(n-1)/3}$ denote the nodes on the legs of the spider in order of their distance from the root 0. Because $\alpha < (n-1)/3$, $|\varphi(a_{(n-1)/3}) - \varphi(0)| > 0$, and the same holds for $b_{(n-1)/3}$ and $c_{(n-1)/3}$. Because the spider has three legs, two of $a_{(n-1)/3}, b_{(n-1)/3}, c_{(n-1)/3}$ are on the same side of the root 0 on the line. Without loss of generality, assume that the a and b legs are both to the right of 0, and that $\varphi(a_{(n-1)/3}) \geq \varphi(b_{(n-1)/3}) > \varphi(0)$. Let k be such that $\varphi(a_k) < \varphi(b_{(n-1)/3}) < \varphi(a_{k+1})$ (where the label a_0 refers to the root 0). Such a k exists because $\alpha < (n-1)/3$, so $\varphi(a_k) \neq \varphi(b_{(n-1)/3})$ for all k, and because $\varphi(0) < \varphi(b_{(n-1)/3}) < \varphi(a_{(n-1)/3})$. It follows that $|\varphi(b_{(n-1)/3}) - \varphi(a_{k+1})| < |\varphi(a_{k+1}) - \varphi(a_k)|$. In contrast, in the 3-spider graph, $b_{(n-1)/3}$ and a_{k+1} have distance at least $(n-1)/3$, and a_{k+1} and a_k have distance 1. Therefore $\alpha > (n-1)/3$. ■

Definition 15 *Given a tree T, a* tripod (a, b, c) *is the union of shortest paths in T connecting every pair of vertices among $\{a, b, c\}$. The* root r *of the tripod is the common vertex among all three shortest paths. The* length *of the tripod is $k = \min\{D(r, a), D(r, b), D(r, c)\}$.*

Any tripod of length k induces a 3-spider with k vertices on each leg, by truncating all longer arms of the tripod to length k. Thus by Lemma 4.5.1, any tree with a tripod of length k must have ordinal relaxation at least k. Using this lower bound, we obtain a constant-factor approximation algorithm.

Theorem 4.5.2 *Given a tree T, there is an ordinal embedding $\varphi : T \to \mathbb{R}$ of T into the line with relaxation $2k + 1$, where k is the length of the largest tripod of T. The* *embedding can be computed in polynomial time.*

Proof: If there are at most two leaves in the tree T, then T can be trivially embedded into the line without distortion or relaxation. Otherwise, T has a tripod. Let (A, B, C) be a longest tripod, let r be its root, and let k be its length. We view T as rooted at r. Let (a, b, c) be a tripod rooted at r that maximizes $D(r, a) + D(r, b) + D(r, c)$. This tripod corresponds to taking the longest three paths starting from different neighbors of r. In particular all three paths have length at least k, so the tripod (a, b, c) has length k. Relabel $\{a, b, c\}$ so that $D(r, a) = k$.

Claim 4.5.3 *For any $d \in \{a, b, c\}$, for any $d' \neq r$ on the path from r to d, and for any descendant x of d', $D(d', x) \leq D(d', d)$.*

Proof: Assume, to the contrary, that $D(d', x) > D(d', d)$. If $d = a$, then there would be a larger tripod (x, b, c) rooted at r. Otherwise, assume without loss of generality that $d = b$. Then there would be a tripod (a, x, c), of the same length, and such that $D(r, a) + D(r, x) + D(r, c) > D(r, a) + D(r, b) + D(r, c)$, a contradiction. ■

Claim 4.5.4 *For any $d \in \{b, c\}$, for any $d' \neq r$ on the path from r to d, and for any descendant x of d', such that the path from x to d' intersects the path from r to d only at vertex d', $D(d', x) \leq k$.*

Proof: Suppose to the contrary that $D(d', x) > k$. By the definition of d', $D(d', a) > D(r, a) = k$. By Claim 4.5.3, $D(d', d) \geq D(d', x)$. If $D(d', d) \leq k$, then $D(d', x) \leq D(d', d) \leq k$, a contradiction. If $D(d', d) > k$, then the tripod (x, d, a) (rooted at d') has length at least $k + 1$, which is again a contradiction. ■

Now we construct the embedding φ as follows. For every vertex x on the shortest path between b and c, we contract every subtree that intersects the path only at x into the single vertex x. The resulting graph is the same path from b to c, but where each vertex represents several vertices of the original graph. We embed this path into the line, placing the ith vertex along the path at coordinate i. This embedding places several vertices of the original graph at the same point in the line.

We claim that the depth of each contracted tree is at most k. For each subtree rooted at r (e.g., the one containing a), no vertex x in the subtree can have $D(r,x) > k$ because then we could have chosen that vertex as a and increase the objective function $D(r,a) + D(r,b) + D(r,c)$, a contradiction. For each subtree rooted at another node $b' \neq r$ on the path from b to c, we can apply Claim 4.5.4 and obtain that $D(b',x) \leq k$ for any vertex x in the subtree rooted at b'. Therefore the depth of each contracted tree is at most k.

Finally we claim that the ordinal relaxation of this mapping is at most $2k+1$. Consider two vertices x and y belonging to contracted subtrees rooted at s and t, respectively. Their original distance is at most $2k + D(s,t)$, and their new distance is $D(s,t)$. Therefore the distance changes order with respect to distances at least $D(s,t)$, for a worst-case ratio of $(2k + D(s,t))/D(s,t)$. This ratio is maximized when $D(s,t) = 1$ in which case it is $2k+1$. ■

Corollary 4.5.5 *There is a polynomial-time algorithm to find φ of Theorem 4.5.2. The algorithm is a 3-approximation algorithm for ordinally embedding trees into a line.*

Proof: The proof of Theorem 4.5.2 is constructive, thus it gives an algorithm. Since the length of the largest tripod is a lower bound of embedding ordinally the tripod into a line, we obtain that the algorithm is a $(2+1/k)$-approximation algorithm. ■

4.6 Ultrametrics

In this section we establish several results about ordinal embedding when the source metric or the target metrics are ultrametrics.

4.6.1 Ultrametrics into ℓ_p with Logarithmic Dimensions

First we demonstrate that ultrametrics can be ordinally embedded into $O(\lg n)$-dimensional ℓ_p space, for any fixed $1 \leq p \leq \infty$, with relaxation 1. Here we exploit the minor difference between "relaxation 1" and "no relaxation"—that equality constraints can be violated—because, as described in the introduction, any ordinal embedding without relaxation of any ultrametric into Euclidean space requires $n-1$ dimensions. Thus the ordinal dimension of an ultrametric is "just barely" $n-1$; the slightest relaxation allows us to obtain a much better embedding. Our result also contrasts metric embeddings where ultrametrics can be embedded into Euclidean space with $1+\epsilon$ distortion, but such an embedding requires $\epsilon^{-2}\lg n$ dimensions [BM04a]. The number of dimensions in our ordinal embeddings is independent of any such ϵ.

Our construction is based on monotone stretching of the discrepancy between different distances:

Lemma 4.6.1 *For any $k > 1$, and for any ultrametric $M = (P, D)$, there is an ultrametric $M' = (P, D')$ such that, for any $p, q, r, s \in P$, if $D(p, q) = D(r, s)$, then $D'(p, q) = D'(r, s)$, and if $D(p, q) > D(r, s)$, then $D'(p, q) \geq kD'(r, s)$.*

Proof: Because M is an ultrametric, we can construct a weighted tree T, with P forming the set of leaves, such that the weights are nondecreasing along any path of T starting from the root. Moreover, for any $u, v \in P$, the ultrametric distance $D(u, v)$ is equal to the maximum weight of an edge along the path from u to v in T.

For $u, v \in P$, define $r(D(u, v)) = i$ where $D(u, v)$ is equal to the ith smallest distance in M. Consider now the weighted tree T' obtained from T by replacing an edge of weight w by an edge of weight $k^{r(w)}$. Let M' be the resulting ultrametric induced by T'. If $D(p, q) = D(r, s)$, then $r(D(p, q)) = r(D(r, s))$, so $D'(p, q) = D'(r, s)$. Finally, if $D(p, q) > D(r, s)$, then $r(D(p, q)) \geq r(D(r, s)) + 1$, so $D'(p, q) \geq kD'(r, s)$. ■

We combine this lemma with a result for the metric case:

Lemma 4.6.2 (Bartal and Mendel [BM04a]) *For any $1 \leq p \leq \infty$, any n-point ultrametric can be metrically embedded into $O(\epsilon^{-2}\lg n)$-dimensional ℓ_p space with distortion at most $1 + \epsilon$.*

Now we are ready to prove the main result of this subsection:

Theorem 4.6.3 *For any $1 \le p \le \infty$, any n-point ultrametric can be ordinally embedded into $O(\lg n)$-dimensional ℓ_p space with relaxation 1.*

Proof: Given an ultrametric $M = (P, D)$, by Lemma 4.6.1, we can obtain an ultrametric $M' = (P, D')$ such that, for any $p, q, r, s \in P$, if $D(p, q) = D(r, s)$, then $D'(p, q) = D'(r, s)$, and if $D(p, q) > D(r, s)$, then $D'(p, q) \ge 2D'(r, s)$. Applying Lemma 4.6.2 with $\epsilon = 1/2$, we obtain a contractive metric embedding φ of P into $O(\lg n)$-dimensional ℓ_p space such that, for any $p, q, r, s \in P$, if $D(p, q) > D(r, s)$, then $\|\varphi(p) - \varphi(q)\| \ge \frac{2}{3} D'(p, q) \ge \frac{4}{3} D'(r, s) \ge \frac{4}{3} \|\varphi(r) - \varphi(s)\|$. Therefore φ is an ordinal embedding with relaxation 1. ■

4.6.2 Arbitrary Distance Matrices into Ultrametrics

In this subsection, we give a polynomial-time algorithm for computing an ordinal embedding of an arbitrary metric into an ultrametric with minimum possible relaxation.

We will show that the optimal ordinal embedding of a distance matrix M into an ultrametric is the *subdominant* of M [FKW95]. One recursive construction of the subdominant is as follows. First, we compute a partition $P = P_1 \cup P_2 \cup \cdots \cup P_k$, for some $k \ge 2$, such that the minimum distance between any P_i and P_j is maximized. Such a partition can be found by computing a minimum spanning tree T of M, and partitioning the points by removing all the edges of T of maximum length. Let Δ be the maximum distance between any two points in P. We create a hierarchical tree representation for an ultrametric by starting with a root v_P and k children $v_{P_1}, \ldots, v_{P_k}$. The length of the edge $\{v_P, v_{P_i}\}$ is equal to Δ for each $i \in \{1, 2, \ldots, k\}$. We recursively compute hierarchical tree representations for the metrics induced by the point sets $P_1, P_2, \ldots, P_k$, and then we merge these trees by identifying, for each $i \in \{1, 2, \ldots, k\}$, the root of the tree for P_i with the node v_{P_i}. In fact this entire construction can be carried out with a single computation of the minimum spanning tree, and thus takes linear time.

Lemma 4.6.4 *Let $\Delta = \max_{p,q \in P} D(p, q)$ and let δ be the minimum distance between two points in different sets P_i and P_j. Then any ordinal embedding has relaxation at* *least Δ/δ.*

Proof: Suppose that the maximum distance Δ is attained by points u, v with $u \in P_i$ and $v \in P_j$, where $i \neq j$. Consider an optimal ordinal embedding φ of M into a hierarchical tree representation T of an ultrametric. Thus the distance between two leaves p and q is equal to the maximum length of an edge along the unique path between p and q. No matter how φ splits P into subsets at the root of T, there exist $r, s \in P$ such that $D(r, s) = \delta$ and the path from r to s in T visits the root of T. Thus the path from r to s passes through the maximum edge in T. Hence, the maximum distance along the path between u and v in T cannot be larger than the maximum distance along the path between r and s in T. Therefore $d(\varphi(u), \varphi(v)) \leq d(\varphi(r), \varphi(s))$, while $D(u, v) = \Delta > \delta = D(r, s)$, so the relaxation is at least Δ/δ. ■

Theorem 4.6.5 *Given any distance matrix M, we can compute in polynomial time an optimal ordinal embedding of M into an ultrametric.*

Proof: Let φ be the ordinal embedding of $M = (P, D)$ computed by the algorithm, with a hierarchical tree representation T. The maximum relaxation α of φ is attained for some $p, q, r, s \in P$ such that $D(p, q) \geq \alpha D(r, s)$ and $d(\varphi(p), \varphi(q)) < d(\varphi(r), \varphi(s))$. It follows that there exists an internal node v of T, with children v_1 and v_2, such that leaves p and q are descendants of v_1, while only one of the leaves r or s is a descendant of v_1. Assume without loss of generality that r is a descendant of v_1 and s is a descendant of v_2.

Consider the recursive call of the algorithm on a subset of points $P' \subseteq P$ in which the node v was created. Because r and s are in different subtrees of v, it follows that, in the partition of the set P' of points computed by the algorithm, the minimum distance between distinct sets is at most $D(r, s)$. On the other hand, the maximum distance between pairs of points in P' is at least $D(p, q)$. Thus, by Lemma 4.6.4, the optimal relaxation for ordinal embedding of M into an ultrametric is at least $D(p, q)/D(r, s) \geq \alpha$. ■

By a similar argument it can be shown that the same algorithm also computes a metric embedding of M into an ultrametric with minimum possible distortion. Furthermore, the distortion is equal to the relaxation in this embedding. In the next section we show that ultrametrics are essentially the only case where this can happen universally.

4.6.3 When Distortion Equals Relaxation

Finally we show that, in a certain sense, ultrametrics are the only target metrics that have equal values of α^* and c^*, or even a universally bounded ratio between α^* and c^*.

Theorem 4.6.6 *If a set $\mathcal{T}$ of target metrics is closed under inclusion (i.e., closed under taking the submetric induced on a subset of points), and there is a constant C such that every distance matrix D has $c^*/\alpha^* \leq C$ (when embedding D into $\mathcal{T}$), then every metric in $\mathcal{T}$ is an ultrametric.*

Proof: Consider any metric M in $\mathcal{T}$. We claim that M has more than one diameter pair. Suppose to the contrary that only p and q attain the maximum distance in M. Let M_{+d} be the distance matrix identical to M except for $M_{+d}(p,q) = M(p,q) + d$. Let d be any positive real greater than the sum of the second- and third-largest distances. Then M_{+d} is not in $\mathcal{T}$ because it violates the triangle inequality and $\mathcal{T}$ is a family of metrics. Because no other distance in M is equal to $M(p,q)$, M_{+d} can be ordinally embedded with no relaxation into $\mathcal{T}$ simply by taking M. However, M_{+d} cannot be metrically embedded into $\mathcal{T}$ without distortion, because M_{+d} is not in $\mathcal{T}$. Furthermore M_{+cd} cannot be metrically embedded into $\mathcal{T}$ with distortion less than c, because any contractive metric embedding must reduce the distance between p and q by a factor of c. Therefore the ratio between the minimum metric distortion c^* and the minimum ordinal relaxation α^* cannot be bounded.

Now by inclusion, any submetric of M induced by three points is also in $\mathcal{T}$, and therefore has a non-unique maximum edge. Thus all triangles in M are tall isosceles, which is one characterization of M being an ultrametric. ■

In fact, this theorem needs only that the set $\mathcal{T}$ of target metrics is closed under taking the induced metric on any triple of points.

4.7 Worst Case of Unweighted Trees into Euclidean Space

In this section, we consider the worst-case relaxation required for ordinal embedding of the shortest-path metric of an unweighted tree T into d-dimensional ℓ_2 space. Our work is motivated by that of Gupta [Gup00b] and Babilon, Matoušek, Maxová, and Valtr [BMMV02]. We show that, for any $d \geq 2$, and for any unweighted tree T on n nodes, $\alpha^* = \tilde{O}(n^{1/d})$. We complement this result by exhibiting a family of trees with optimal ordinal relaxation $\Omega(n^{1/(d+1)})$. In contrast, the best bounds on the worst-case distortion required are $\tilde{O}(n^{1/(d-1)})$ and $\Omega(n^{1/d})$ [Gup00b]. These ranges overlap at the endpoint of $\tilde{\Theta}(n^{1/d})$, but it seems that ordinal embedding and metric embedding behave fundamentally differently, in particular because different proof techniques are required for both the upper and lower bounds.

First we prove the upper bound. At a high level, the algorithm finds nodes that can be contracted to a single point, which can be an effective ordinal embedding, unlike metric embedding where it causes infinite distortion.

Theorem 4.7.1 *Any weighted tree can be ordinally embedded into d-dimensional ℓ_2 space with relaxation $\tilde{O}(n^{1/d})$.*

Proof: Let $T = (V(T), E(T))$ be an unweighted tree with $|V(T)| = n$. We show how to obtain an ordinal embedding of T into d-dimensional ℓ_2 space with relaxation $\tilde{O}(n^{1/d})$.

We construct a new tree T' as follows. Initially, we set $T'_0 := T$. For $i = 1, \ldots, n^{1/d}$, we repeat the following process: Set $T'_i := T'_{i-1}$. For any leaf v of T'_{i-1}, we remove v from T'_i. Let $T' := T'_{n^{1/d}}$.

Define the function $p : V(T) \to V(T')$, such that for any $v \in V(T) \setminus V(T')$, $p(v)$ is the node in $V(T')$, which is closest to v, and for any $v \in V(T')$, $p(v) = v$. It is easy to see that for every leaf v of T', there are at least $n^{1/d}$ nodes $u \in V(T) \setminus V(T')$, with $p(u) = v$. Thus, the number of leaves of T' is at most $n^{\frac{d-1}{d}}$.

It follows that using Gupta's algorithm [Gup00b], we can compute an expansive metric embedding φ' of T' into d-dimensional ℓ_2 space with distortion at most $kn^{1/d}$, for some $k = \text{polylog}(n)$. To obtain an embedding φ of T, we simply set $\varphi(v) = \varphi'(p(v))$ for each $v \in V(T)$.

It remains to show that φ' has ordinal relaxation $\tilde{O}(n^{1/d})$. Let $v_1, v_2, v_3, v_4 \in V(T)$, with $v_3 \neq v_4$ and

$$d_T(v_1, v_2) > (2 + k)n^{1/d} d_T(v_3, v_4).$$

We have

$$\begin{aligned}
\|\varphi(v_1) - \varphi(v_2)\| &= \|\varphi'(p(v_1)) - \varphi'(p(v_2))\| \\
&\geq d_{T'}(p(v_1), p(v_2)) \\
&\geq d_T(v_1, v_2) - 2n^{1/d} \\
&> (2 + k)n^{1/d} d_T(v_3, v_4) - 2n^{1/d} \\
&\geq kn^{1/d} d_T(v_3, v_4) \\
&\geq kn^{1/d} d_{T'}(p(v_3), p(v_4)) \\
&\geq \|\varphi'(p(v_3)) - \varphi'(p(v_4))\| \\
&= \|\varphi(v_3) - \varphi(v_4)\|.
\end{aligned}$$

Thus, we obtain that φ has ordinal relaxation at most $(2 + k)n^{1/d} = \tilde{O}(n^{1/d})$. ■

Next we prove the worst-case lower bound. The main novelty here is a new packing argument for bounding relaxation. Let $F(m, L)$ denote the m-spider with arms of length L, that is, an m-star with each edge refined into a path of length L.

Lemma 4.7.2 *Any ordinal embedding of $F(m, L)$ into d-dimensional ℓ_2 space requires relaxation $\Omega(\min\{L, m^{1/d}\})$.*

Proof: Let $T = F(m, L)$, and let $r \in V(T)$ be the only vertex of T with degree greater than 2. For any i, with $0 \leq i \leq L$, let $U_i = \{v \in V(T) \mid d_T(r, v) = i\}$.

Consider an optimal embedding $\varphi : V(T) \to \mathbb{R}^d$ with relaxation α. We define

$$\begin{aligned}
\mu_i &= \min_{u,v \in V(T)} \{\|\varphi(u) - \varphi(v)\| \mid d_T(u,v) = i\}, \\
\lambda_i &= \max_{u,v \in V(T)} \{\|\varphi(u) - \varphi(v)\| \mid d_T(u,v) = i\}.
\end{aligned}$$

Observe that, if $\mu_{2L} = 0$, then there exist $u, v \in U_L$ such that $\varphi(u) = \varphi(v)$. It follows that, if $\alpha < 2L$, then for any $\{x, y\} \in E(T)$, $\varphi(x) = \varphi(y)$, which implies that all the vertices are mapped to the same point, and thus $\alpha = \Omega(L)$.

It remains to show that the assertion is true in the case $\mu_{2L} > 0$. Consider the nodes of U_L. For any $u, v \in U_L$, we have $d_T(u,v) = 2L$, and thus $\|\varphi(u) - \varphi(v)\| \geq \mu_{2L}$. For any $v \in U_L$, let B_v be the ball of radius $\mu_{2L}/2$ centered at $\varphi(v)$. It follows that, for any $u, v \in U_L$, the balls B_u, B_v can intersect only on their boundary. Thus,

$$\begin{aligned}
\left| \bigcup_{v \in U_L} B_v \right| &= \sum_{v \in U_L} |B_v| \\
&= \Omega(m \mu_{2L}^d)
\end{aligned}$$

By a packing argument, we obtain that there exist $u, v \in U_L$ such that $\|\varphi(u) - \varphi(v)\| = \Omega(m^{1/d} \mu_{2L})$, which implies

$$\lambda_{2L} = \Omega(m^{1/d} \mu_{2L}). \tag{4.1}$$

Now consider two nodes $u, v \in U_L$ such that $\|\varphi(u) - \varphi(v)\| = \lambda_{2L}$, and let p be the path from u to v in T. It follows that there exist nodes $x, y \in p$ with $d_T(x, y) = 2L/\alpha$ and $\|\varphi(x) - \varphi(y)\| \geq \lambda_{2L}/\alpha$. Thus

$$\lambda_{2L/\alpha} \geq \frac{\lambda_{2L}}{\alpha}. \tag{4.2}$$

Also, by the definition of the ordinal relaxation, we have

$$\mu_{2L} > \lambda_{2L/\alpha}. \tag{4.3}$$

Combining (4.1), (4.2), and (4.3), we obtain $\alpha\lambda_{2L/\alpha} = \Omega(m^{1/d}\mu_{2L}) = \Omega(m^{1/d}\lambda_{2L/\alpha})$. Thus we have shown that, if $\mu_{2L} > 0$, then $\alpha = \Omega(m^{1/d})$. The lemma follows. ∎

Theorem 4.7.3 *For any $n > 0$ and any $d \geq 2$, there is a tree T on n nodes for which every ordinal embedding has relaxation $\Omega(n^{1/(d+1)})$.*

Proof: The theorem follows from Lemma 4.7.2, for $T = F(n^{d/(d+1)}, n^{1/(d+1)})$. ∎

4.8 Arbitrary Metrics into Low Dimensions

By Lemma 4.3.1, a general $O(\lg n)$ upper bound on relaxation carries over from metric embeddings of any n-point metric space into $O(\lg n)$-dimensional Euclidean space, using theorems of Bourgain and of Johnson and Lindenstrauss. For metric distortion, this bound is tight [LLR95], but one might suspect that the ordinal relaxation can be smaller. Here we show that it cannot be much smaller: some n-point metric spaces require relaxation $\Omega(\log n/\log\log n)$. This claim is a special case of the following result.

Theorem 4.8.1 *There is an absolute constant $c > 0$ such that, for every d and n, there is a metric space T on n points such that the relaxation of any ordinal embedding of T into d-dimensional Euclidean space is at least $\frac{\log n}{\log d + \log\log n + c} - 1$.*

The proof is based on two known results. The first is a bound of Warren on the number of sign patterns of a system of real polynomials. The second is the existence of dense graphs with no short cycles. We first state these two results.

Let $P_j = P_j(x_1, \ldots, x_\ell)$, $j = 1, \ldots, m$, be m real polynomials. For a point $u = (u_1, \ldots, u_\ell) \in \mathbb{R}^\ell$, the *sign pattern* of the P_j's at u is the m-tuple $(\epsilon_1, \ldots, \epsilon_m) \in (-1, 0, 1)^m$, where $\epsilon_j = \text{sign } P_j(u)$. Let $s(P_1, P_2, \ldots, P_m)$ denote the total number of sign patterns of the polynomials $P_1, P_2, \ldots, P_m$, as u ranges over all points of $\mathbb{R}^\ell$.

The following result is derived in [Alo95] as a slight modification of a theorem of Warren [War68].

Theorem 4.8.2 *Let $P_1 \dots P_m$ be m real polynomials in ℓ real variables, and suppose the degree of each P_j does not exceed k. If $2m \geq \ell$, then $s(P_1 \dots P_m) \leq (8ekm/\ell)^\ell$.*

The following statement follows from a result of Erdős and Sachs [ES63], and can be also proved directly by a simple probabilistic argument.

Lemma 4.8.3 *For every $g \geq 3$ and every $n \geq 3$, there are (connected) graphs on n vertices with at least $\frac{1}{4}n^{1+1/g}$ edges, and with no cycle of length at most g.*

We note that there are slightly better known results based on certain algebraic constructions, but for our purpose here the above estimate suffices.

We can now prove Theorem 4.8.1. Throughout the proof and the rest of the section, we assume that n is large, whenever this is needed, and omit all floor and ceiling signs whenever these are not crucial.

Proof:[of Theorem 4.8.1] Without trying to optimize the constants, define $g = \frac{\log n}{\log d + \log\log n + 5}$. We will show that some n-point metric spaces require relaxation at least $g-1$. Without loss of generality, assume $g-1$ is bigger than 1, as otherwise there is nothing to prove. By Lemma 4.8.3, there is a graph $G = (V, E)$ on a set $V = \{1, 2, \dots, n\}$ of n labeled vertices, with $m \geq \frac{1}{4}n^{1+1/g} > 7nd\log n$ edges, and with no cycles of length at most g. For every subset $E' \subset E$ of precisely $m/2$ edges, the subgraph (V, E') of G defines a metric space $T(E')$ on the set V (if the subgraph is disconnected, some distances can be defined to be infinite; alternatively, we can fix a spanning tree in G and include it in all subgraphs to make sure they are all connected). This gives us a collection of $2^{(1+o(1))m}$ metric spaces on V, with the following property.

(*) For every two distinct spaces (T, d) and (T', d') in the collection, there are two pairs of points x, y and z, w so that $d(x, y) = 1$ and $d'(x, y) > g-1$, whereas $d'(z, w) = 1$ and $d(z, w) > g-1$.

Indeed, this follows from the fact that, for every two distinct subgraphs in our collection, there is an edge $\{x, y\}$ belonging to the first one and not to the second, and vice versa. As the shortest cycle in G is of length exceeding g, the claim in (*)

follows.

Fix a space T is our collection, and let φ_T be a minimum relaxation embedding of it into d-dimensional Euclidean space. Let $\varphi_T(i) = (x^T_{i,1}, \ldots, x^T_{i,d})$. Then the square of the Euclidean distance between each two points in the embedding can be expressed as a polynomial of degree 2 in the dn variables $x^T_{i,j}$. The difference between two such squares of distances is thus also a polynomial of degree 2 in these variables. It follows that the order of all $\binom{n}{2}$ distances is determined by the signs of $\binom{n}{2}^2 < n^4/4$ polynomials of degree 2 each, in dn variables. By Theorem 4.8.2, the total number of such orders is at most

$$\left(\frac{16en^4}{4dn}\right)^{dn} \leq n^{(3+o(1))dn} = 2^{(3+o(1))nd\log n}.$$

This is smaller than the number of spaces in our collection, and hence, by the pigeonhole principle, there are two distinct spaces T and T' in our collection, so that the orders of the distances in their embeddings are the same. This, together with (*), implies that the relaxation in at least one of these embeddings is at least $g - 1$, completing the proof. ■

The last proof easily extends to embeddings into d-dimensional ℓ_p space for any even integer p. The only difference is that, in this case, the pth power of the distance between a pair of given points in the embedding is a polynomial of degree p in the dn variables describing the embedding. Working out the computation in the proof above, this yields the following result.

Theorem 4.8.4 *There is an absolute constant $c > 0$ such that, for every d and n, and for every even integer p, there is a metric space T on n points such that the relaxation in any ordinal embedding of T into d-dimensional ℓ_p space is at least* $\frac{\log n}{\log d + \log(\log n + \log p) + c} - 1$.

The above argument, combined with an additional trick, can in fact be extended to handle ordinal embeddings into d-dimensional ℓ_p space for odd integers p, as well as embeddings into d-dimensional ℓ_∞ space.

Theorem 4.8.5 *(i) For every $n \geq d$, there is a metric space T on n points such that the relaxation in any ordinal embedding of T in d-dimensional ℓ_∞ space is at least* $\frac{\log n}{\log d + \log\log n + O(1)} - 1$.

(ii) For every $n \geq d$, and for every odd positive integer p, there is a metric space T on n points such that the relaxation of any ordinal embedding of T into d-dimensional ℓ_p space is at least $\frac{\log n}{\log(2d^2 + 3d\log n + d\log p + O(d))} - 1$.

Proof: As before, the result is proved by a counting argument: we prove that the number of possible orders between all distances in a set of n points in the relevant spaces is not too large, and use the fact that there are many significantly different metric spaces on n points, concluding that for two such metric spaces the embedding orders the distances identically, and hence deriving the required lower bound on relaxation.

(i) We start by bounding the number of possible orders of all distances in a set X of n points in d-dimensional ℓ_∞ space. Given such a set, define, for each ordered set (x, y, z, w) of (not necessarily distinct) four points of X, and for each two indices i, j in $\{1, 2, \ldots, d\}$, the following linear polynomial in the dn variables representing the coordinates of the points: $(x_i - y_i) - (w_j - z_j)$. By Theorem 4.8.2 these d^2n^4 polynomials have at most $(O(1)dn^3)^{dn} \leq 2^{(4+o(1))dn\log n}$ sign patterns. (In fact, because the polynomials here are linear, there is a slightly better, and simpler, estimate than the one provided by Warren's Theorem here—see [Har67]—but the asymptotic of the logarithm in this estimate is the same.) We claim that the signs of all these polynomials determine completely the order on all the $\binom{n}{2}$ distances between pairs of the points. Indeed, the signs of the polynomials $(x_i - y_i) - (x_j - y_j)$, $(x_i - y_i) - (y_j - x_j)$ (and their inverses) determines a coordinate i such that $||x - y||_\infty$ is $x_i - y_i$ or $y_i - x_i$ (as this is simply the maximum of all $2d$ differences of the form $(x_i - y_i), (y_i - x_i)$). Suppose, now, that $||x - y||_\infty = x_i - y_i$ and $||w - z||_\infty = w_j - z_j$. Then the sign of $(x_i - y_i) - (w_j - z_j)$ determines which of the two distances is bigger. It follows that the total number of orders of the distances of n points in d-dimensional ℓ_∞ space is at most $2^{(4+o(1))dn\log n}$.

Define $g = \frac{\log n}{\log d + \log\log n + 5}$, take a graph $G = (V, E)$ as in the proof of Theorem 4.8.1, and construct a collection of $2^{(1+o(1))7nd \log n}$ metric spaces on a set of n points satisfying (*). The desired result follows, just as in the proof of Theorem 4.8.1.

(ii) As in the proof of part (i), we first bound the number of possible orders of all distances in a set X of n points in d-dimensional ℓ_p space. Given such a set, define, for each two (not necessarily distinct) pairs $\{x, y\}$ and $\{z, w\}$ of points, and each two sign vectors

$$(\epsilon_1, \epsilon_2, \ldots, \epsilon_d), (\delta_1, \delta_2, \ldots, \delta_d) \in \{-1, 1\}^d,$$

the following polynomial in the dn coordinates of the points:

$$\sum_{i=1}^{d} \epsilon_i (x_i - y_i)^p - \sum_{j=1}^{d} \delta_j (z_j - w_j)^p.$$

This is a set of $2^{2d} n^4$ polynomials, each of degree p, and thus, by Theorem 4.8.2, the number of their sign patterns is bounded by

$$2^{2d^2 n + 3dn \log n + dn \log p + O(dn)}. \tag{4.4}$$

As before, it is not difficult to see that the signs of all these polynomials determine completely the order of all distances between pairs of points. Therefore, the number of such orders does not exceed (4.4). The desired result now follows as before, by considering metrics induced by subgraphs with half the edges of a graph on n vertices with at least $\frac{1}{4} n^{1+1/g}$ edges, and no cycles of length at most g, where $g = \frac{\log n}{\log(2d^2 + 3d \log n + d \log p + O(d))}$. ■

4.9 Conclusion and Open Problems

We have introduced minimum-relaxation ordinal embeddings and shown that they have distinct and sometimes surprising behavior. Yet many problems remain to be explored in this context; our hope is that this paper forms the foundation of a fruitful body of research. Here we highlight some of the more important directions for future exploration.

An important line of study is to continue comparing ordinal embeddings with metric embeddings. One interesting question is whether the dimensionality-reduction results of Bourgain [Bou85] and Johnson and Lindenstrauss [JL84] can be improved for ordinal relaxation. From Theorem 4.8.1 and Proposition 4.3.1, we know that the optimal worst-case relaxation for an ordinal embedding of a general metric into $O(\lg n)$-dimensional Euclidean space is between $\Omega(\lg n / \lg \lg n)$ and $O(\lg n)$. Closing this $\Theta(\lg \lg n)$ gap is an intiguing open problem; a better upper bound would improve on Bourgain-based metric embeddings into $O(\lg n)$ dimensions. Another problem is how much relaxation is required for dimensionality *reduction* of a metric already embedded in arbitrary dimensional ℓ_p space. For $p = 2$, we obtain an ideal relaxation of $1 + \epsilon$ using Johnson-Lindenstrauss combined with Proposition 4.3.1. For $p = \infty$, dimensionality reduction is impossible, by Theorem 4.8.5(i), because ℓ_∞ is universal in the metric sense. For $p \neq 2, \infty$, the problem is open; in contrast, it is known for metric embeddings that dimensionality reduction is impossible for ℓ_1 [BC03, LN04b].

Another important direction is to develop more approximation algorithms for minimum-relaxation ordinal embedding. Unlike general upper bounds on distortion, existing approximation algorithms for minimum-distortion metric embedding do not carry over to ordinal embedding because the optimum solution is generally smaller. Our $O(1)$-approximation result in Section 4.5, and the lack of a matching result for metric embedding despite much effort, shows that in some contexts ordinal embedding problems may prove more easily approximable than metric embedding. We expect that our approximation result can be generalized using similar techniques to unweighted graphs, weighted trees, and/or higher dimensions, and that it can be strengthened to a PTAS. A related open problem is to consider trees as target metrics, and find the tree metric into which a given metric can be ordinally embedded with approximately minimum relaxation. Another family of approximation problems arise with the related notion of *additive relaxation*, in contrast to (multiplicative) relaxation, where pairs of distances within an additive α must have their relative order preserved. In some cases, approximation results may be harder for ordinal embedding than metric embedding. For example, in the problem of approximating the minimum additive distortion/relaxation for an ordinal embedding of an arbitrary metric into the line, the simple greedy algorithm of Proposition 4.3.5 is a 3-approximation for metric embedding but can be arbitrarily bad for ordinal embedding.[3]

A final direction to consider is finding other applications of ordinal embedding. In particular, in the context of approximation algorithms for other problems, when are low-relaxation ordinal embeddings as useful as (and more powerful than) low-distortion metric embeddings? Nearest neighbor is a simple example where only the order of distances is relevant, but we expect there are several other such problems.

Acknowledgments

We thank Amir Ben-Dor, Piotr Indyk, Howard Karloff, and Yuval Rabani for helpful discussions. In particular, Amir Ben-Dor contributed key ideas to the proof of universality (Theorem 4.4.1).

[3]The example is as follows. The graph has four points a, b, p, q and $D(p, q) = 50$, $D(p, a) = 100$, $D(p, b) = 100 + \epsilon$, $D[q, a] = 70$, $D(q, b) = 60$, and $D(a, b) = 10$. The optimum ordinal embedding can place the points in the order p, q, b, a, and has additive relaxation ϵ, while greedy places the points in the order p, q, a, b and has additive relaxation 10, resulting in an arbitrarily large ratio.

Chapter 5

Embeddings with extra information

Credits: The results in this section is work done with Erik Demaine, Mohammad-Taghi Hajiaghayi, and Piotr Indyk, and has appeared in SoCG'04.

A frequently arising problem in computational geometry is when a physical structure, such as an ad-hoc wireless sensor network or a protein backbone, can measure local information about its geometry (e.g., distances, angles, and/or orientations), and the goal is to reconstruct the global geometry from this partial information. More precisely, we are given a graph, the approximate lengths of the edges, and possibly extra information, and our goal is to assign 2D coordinates to the vertices such that the (multiplicative or additive) error on the resulting distances and other information is within a constant factor of the best possible. We obtain the first pseudo-quasipolynomial-time algorithm for this problem given a complete graph of Euclidean distances with additive error and no extra information. For general graphs, the analogous problem is NP-hard even with exact distances. Thus, for general graphs, we consider natural types of extra information that make the problem more tractable, including approximate angles between edges, the order type of vertices, a model of coordinate noise, or knowledge about the range of distance measurements. Our pseudo-quasipolynomial-time algorithm for no extra information can also be viewed as a polynomial-time algorithm given an "extremum oracle" as extra information. We give several approximation algorithms and contrasting hardness results for these scenarios.

5.1 Introduction

Suppose we have a geometric structure (a graph realized in Euclidean space), but we can only measure local properties in this structure, such as distances between pairs of vertices, and the measurements are just approximate. In many applications, we would like to use this approximate local information to reconstruct the entire geometric structure, that is, the realization of the graph. Two interesting questions arise in this context: when is such a reconstruction unique, and can it be computed efficiently?

These problems have been studied extensively in the fields of computational geometry [CL92, EHKN99, Yem79, Sax79], rigidity theory [Hen92, Con91, JJ05], sensor networks [ČHH01, SRB01], and structural analysis of molecules [BKL99, ABC$^+$05, CH88, Hen95]. The reconstruction problem arises frequently in several distributed physical structures such as the atoms in a protein [BKL99, CH88, Hen95] or the nodes in an ad-hoc wireless network [ČHH01, SRB01, PCB00].

A reconstruction is always unique (up to isometry) and easy to compute for a complete graph of exact distances, or any graph that can be "shelled" by incrementally locating nodes according to the distances to three noncollinear located neighbors. More interesting is that such graphs include visibility graphs [CL92] and segment visibility graphs [EHKN99]. In general, however, the reconstruction problem is NP-hard [Yem79], even in the strong sense [Sax79]. It is also NP-hard to test whether a graph has a unique reconstruction [Sax80b, Sect. 6]. The uniqueness of a reconstruction in the *generic* case[1] was recently shown to be testable in polynomial time in 2D by a

[1] In the generic case [JJ05], we are given the promise that the goal embedding is "generic". An embedding of a graph into d-dimensional Euclidean space is *generic* if the coordinates of the vertices are algebraically independent over the rationals, i.e., no polynomial over the vertex coordinates with rational coefficients evaluates to zero, except for the zero polynomial.

simple characterization related to generic infinitesimal rigidity [Hen92, JJ05], but this result has not yet led to efficient algorithms for actual reconstruction in the generic case.

The goal of this chapter is to overcome these difficulties by obtaining efficient algorithms for approximate embedding of metrics into the plane. Our approach is to explore possible additional types of local information and study their influence on the complexity of the problem. In many practical scenarios, such information is readily available. In other cases, the amount of extra information needed is so small that it can be "guessed" via exhaustive enumeration, which leads to a pseudo-quasipolynomial-time algorithm that uses no extra information.[2] This algorithm is in fact the first such algorithm for embedding into low-dimensional Euclidean space with approximately optimal additive distortion.

We consider the following types of extra information:

Angle information: Between every pair of incident edges, we are given the approximate counterclockwise angle.

Extremum oracle: Suppose that the x coordinates of the embedding are known (fixed). Let f be an optimal (minimum-distortion) embedding subject to these and all other constraints. The extremum oracle reports, in any specified vertical slab of the optimal embedding, the minimum y coordinate of a point and a point achieving that coordinate, and symmetrically for the maximum y coordinate. More precisely, given a range $[x_l, x_r]$, the oracle reports the data point $p = \mathrm{argmin}_{p' : f_x(p') \in [x_l, x_r]} f_y(p')$ and $f(p)$, and symmetrically with argmax. In addition, we require that the answers returned by the oracle to different queries are consistent, that is, based on the same embedding f.

Guessing this extra information is exactly what causes one of our algorithms to use pseudo-quasipolynomial time when given no extra information. This result is presented in the chapter on additive distortion.

[2] An algorithm's running time is *quasipolynomial* if it is $2^{\log^{O(1)} n}$, *pseudopolynomial* if it is $N^{O(1)}$ where N is the maximum value of any number in the problem instance, and *pseudo-quasipolynomial* if it is $2^{\log^{O(1)} n \,\cdot\, \log^{O(1)} N}$.

Order type: For some point p and all pairs of points q, r, we are given the clockwise/counterclockwise orientation of $\triangle pqr$.

Distribution information: We know that the metric is induced by random points in a square (as in, e.g., [GRK04]) plus adversarial noise added to their pairwise distances.

Range constraints: Each point p has a range r_p such that we know the (approximate) distance between p and a point q precisely when this distance is at most r_p.

One of our motivations for studying these problems is "autoconfiguration" in the Cricket Compass [PMBT01, MIT] location system. In this system, several beacons are placed in a physical environment, and the goal is to find the global geometry of these beacons in order to enable private localization of mobile devices such as PDAs (personal digital assistants). In general, the beacons live in 3D space, but a common scenario is that they all live in a common plane (the ceiling). Beacons can measure approximate pairwise distances, with sub-centimeter accuracy and a range of up to several meters, using a combination of ultrasonic and radio pulses (measuring the difference in travel time between the sound-speed pulse and the light-speed pulse). Using two or more ultrasonic transceivers to measure distances from two or more points on a beacon, beacons can also measure approximate counterclockwise angles of other beacons within range, relative to a local coordinate system. In this practical scenario, distribution information, range constraints, order type, and especially angle information are all reasonable assumptions to consider.

We show that any of the types of extra information described above, in addition to the approximate distance information given by D, often allow us to design efficient algorithms to construct embeddings into 2D with approximately optimal distortion. Specifically, we develop polynomial-time algorithms for the following versions of this embedding-with-extra-information problem:

1. Embedding a general graph with approximate angle information into two-dimensional ℓ_s space, $s \in \{1, 2, \infty\}$, with approximately optimal multiplicative distortion.

If we are given the counterclockwise angle of each edge with respect to a fixed axis, or we are given counterclockwise angles between incident pairs of edges in the complete graph, our approximation factor is $O(1)$. If we are given counterclockwise angles between pairs of incident edges in a general graph, our approximation factor is $O(\text{diam})$ where diam is the diameter of the graph. The approximation factors depend on the additive error on the angles; see Section 5.2 for details.

These algorithms are the first subexponential-time algorithms for embedding an arbitrary metric into a low-dimensional space (even in the one-dimensional case) to approximately minimize multiplicative distortion. Without angles, even embedding tree metrics into the line with approximately minimum multiplicative distortion is hard to approximate better than a factor of $\Theta(n^{1/12})$, by a recent result of Bădoiu et al. [BCIS05].

2. Embedding a complete graph into the Euclidean plane with $O(1)$-approximate additive distortion in pseudo-quasipolynomial time of $2^{O(\log n \cdot \log^2 \Delta)}$ where Δ is the "spread" of the input point set. We obtain this result in Section 2.4 using a polynomial number of calls to an extremum oracle, which can be simulated in pseudo-quasipolynomial time.

   This algorithm is the first algorithm for minimizing the additive distortion of an embedding into a low-dimensional Euclidean space, other than trivial exponential-time algorithms.

3. Embedding a complete graph into the Euclidean plane with $O(1)$-approximate additive distortion given the orientation of all triples of points involving a common point (Section 5.3).

4. Embedding a complete graph into the Euclidean plane with $O(1)$-approximate additive distortion given the prior that the distances D are approximately induced by a random set of points in a unit square. In this case, our algorithm returns an embedding with additive distortion that is within a constant factor

of the maximum noise added to any distance. See Section 5.4 for the detailed formulation.

5. Embedding a general graph that satisfies the range constraints into the line with $O(1)$-approximate additive distortion (Section 5.5).

6. In contrast, we show that embedding a general graph that satisfies the range constraints into two-dimensional ℓ_p space, for $p \in \{1, 2, \infty\}$, is NP-hard (Section 5.6). This problem was known to be NP-hard without range constraints in d-dimensional Euclidean space for all d [Sax79].

Several of these algorithms are practical; often they are based on simple linear programs.

5.2 Embedding with Angle Information

This section considers embedding a graph with given edge lengths up to multiplicative error and given angles with additive error, in ℓ_1, ℓ_2, and ℓ_∞. We consider several possible angle specifications in the next section, and reduce to the case that we know the counterclockwise angle between every edge and one fixed edge.

5.2.1 Different Types of Angle Information

Lemma 5.2.1 *Given a complete graph, and given counterclockwise angles between pairs of incident edges each with (one-sided or two-sided) additive error at most γ, we can compute the counterclockwise angle of every edge with respect to a particular edge with additive error at most 2γ.*

Proof: Fix one edge (p, q) and call it the x axis. To estimate the counterclockwise angle of an edge (v, w) with respect to the x axis, we use the known counterclockwise angles $\theta_1 = \angle pqv$ and $\theta_2 = \angle qvw$. If the angles were exact, the counterclockwise angle of (v, w) with respect to (p, q) would be $\theta_1 + \theta_2 - 180°$ (modulo $360°$). With additive error, the errors in $\theta_1 + \theta_2$ accumulate to at most double in the worst case. ■

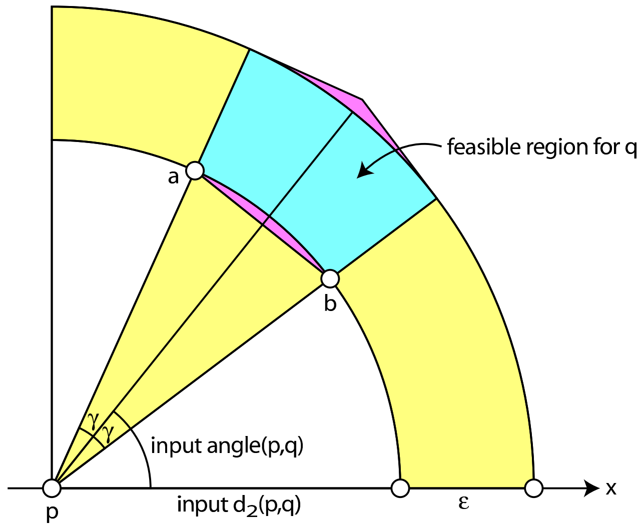


Figure 5-1: Feasible region of a point q with respect to p given the ℓ_2 distance within a multiplicative ϵ and given the counterclockwise angle to the x axis within an additive γ. (Measuring counterclockwise angle, instead of just angle, distinguishes between q being "above" or "below" the x axis.)

Lemma 5.2.2 *Given a general graph, and given counterclockwise angles between pairs of incident edges each with (one-sided or two-sided) additive error at most γ, we can compute the counterclockwise angle of every edge with respect to a particular edge with additive error at most $(diam+1)\gamma$ where diam is the diameter of the graph.*

Proof: Similar to Lemma 5.2.1, except now we must combine counterclockwise angles along a path $p, q, \ldots, v, w$, which might have length up to diam + 2, and therefore involves at most diam + 1 angles. ■

This lemma is the best we can obtain in the worst case. We can of course improve the angles estimates by, e.g., choosing (p, q) to be maximally central, computing shortest paths, etc. If the errors are known to be independent and randomly distributed with mean 0, the error is $O(\sqrt{\text{diam}})$ with high probability, where diam is the diameter of the graph.

5.2.2 ℓ_2 Algorithm

Our algorithm for embedding into the Euclidean plane assumes, possibly using the reductions of the previous section, that we are given the approximate counterclockwise angle of every edge with respect to one edge (which we view as the x axis). The algorithm sets up a constraint program for finding the coordinates of each vertex. The straightforward setup has nonconvex constraints and is difficult to solve. We relax the program to a convex program at the cost of some error. We further relax the program to a linear program at the cost of additional error.

The basic optimization problem has the following constraints. For every edge (p, q), the distance and angle information of that edge specifies a (nonconvex) *constraint region*, relative to the location of p, that must contain q. (See Figure 5-1.) Conditioning on that there is an embedding of the graph achieving multiplicative error ϵ on the distances and additive error γ on the angles, we can find such an embedding by finding a feasible solution satisifying all constraints. If only one of these error parameters (e.g., γ) is known, we can still find such an embedding by setting the objective function to minimize the other error parameter (e.g., ϵ). If neither error parameter is known, we obtain a family of solutions by minimizing one error parameter subject to various choices for the other parameter; alternatively, we can minimize any desired linear combination of the error parameters by setting the objective function accordingly.

We can relax each constraint region to be convex by taking its convex hull. More precisely, we add one edge (a, b) to cut off the inner arc of the region; see Figure 5-1. This relaxation, applied to every constraint region defined by an edge (p, q), produces a convex program. The maximum possible error is obtained when q is placed at the midpoint between a and b. Then the distance between p and q is $\cos \gamma$ times the input distance between p and q. We can transform this contraction into an expansion by multiplying all distances by $1/\cos \gamma$. Thus, the maximum expansion is at most $(1 + \epsilon)/\cos \gamma$, proving the following theorem:

Theorem 5.2.3 *Given a graph, given the length of each edge with multiplicative error ϵ, and given the counterclockwise angle of every edge with respect to a particular edge with additive error γ, we can compute in polynomial time an ℓ_2 embedding with angles of maximum additive error γ and distances of maximum multiplicative error*

$(1+\epsilon)/\cos\gamma - 1$.

We can further relax the constraint region to be piecewise-linear by approximating the unique arc of the region with a polygonal chain. Then we obtain a linear program from combining the relaxed constraint for each edge (p, q). If we use $k+1 \geq 2$ segments in a regular chain, the maximum expansion factor of a distance is $(1+\epsilon)/\cos(\gamma/k)$. By incorporating the expansion factor from the previous theorem as well, we obtain the following theorem:

Theorem 5.2.4 *Given a graph, given the length of each edge with multiplicative error ϵ, and given the counterclockwise angle of every edge with respect to a particular edge with additive error γ, we can compute in polynomial time an ℓ_2 embedding with angles of maximum additive error γ and distances of maximum multiplicative error*

$$\frac{1+\epsilon}{\cos\gamma\cos(\gamma/k)} - 1 = \frac{1+\epsilon}{\cos\gamma} - 1 + O\left(\frac{\gamma^2}{k^2}\right).$$

5.2.3 More Types of Angle Information

For embedding into ℓ_1, we need additional information about the global rotation of the graph. More precisely, we need to know, for each edge (p, q), the quadrant of q with respect to p. In other words, we need to know the two high-order bits of the counterclockwise angle of each edge (p, q) with respect to the x axis, i.e., whether this angle is in $[0, 90°]$, $[90°, 180°]$, $[180°, 270°]$, or $[270°, 360°]$. Because of our additive angle errors, we may not know to which quadrant an edge belongs; in this case, we would like to know that the edge is borderline between two particular quadrants.

If our input specifies counterclockwise angles of edges with respect to the x axis, we are done. For other types of input, we can apply the following reductions:

Lemma 5.2.5 *Given a graph, given counterclockwise angles between pairs of incident edges each with (one-sided or two-sided) additive error at most γ, and given the counterclockwise angle of one edge relative to the x axis with the same additive error, we can compute the counterclockwise angle of every edge with respect to the x axis* *with additive error at most* $(diam + 2)\gamma$*.*

Proof: Apply Lemma 5.2.2 relative to the edge for which we know the counterclockwise angle with respect to the x axis, and translate using this angle. ∎

Lemma 5.2.6 *Given a graph, and given counterclockwise angles between pairs of incident edges each with (one-sided or two-sided) additive error at most* γ*, we can compute a family of* $O(1/\gamma')$ *possible assignments of counterclockwise angles relative to the* x *axis with additive error at most* $\gamma + \gamma'$*.*

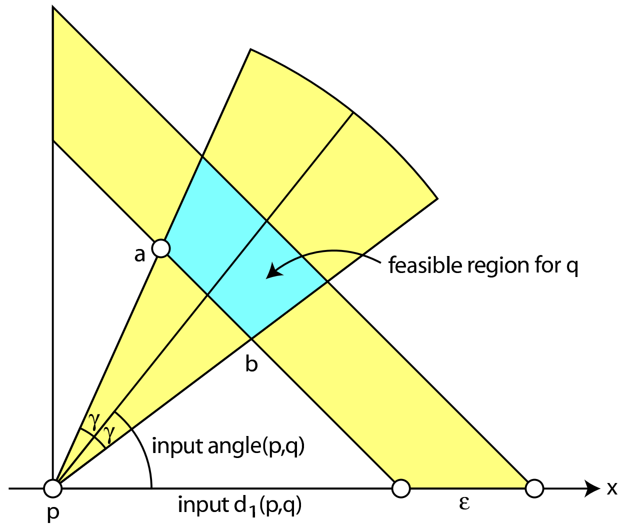
Proof: Apply Lemma 5.2.2 to obtain counterclockwise angles relative to an edge e, and then "guess" the counterclockwise angle of the x axis with respect to e among the $\lceil 360°/\gamma' \rceil$ angles of the form $0, \gamma', 2\gamma', \ldots$. ∎

5.2.4 ℓ_1 Algorithm

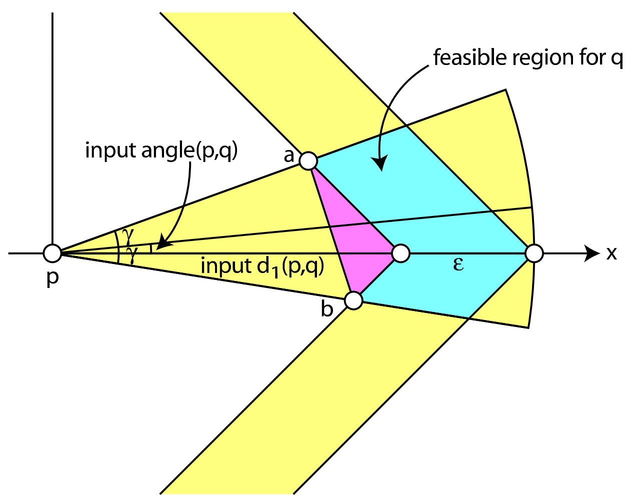
We can adapt the ℓ_2 algorithm to an ℓ_1 algorithm as follows. The convex program and linear program are the same as before; the only difference is the shape of the constraint region of q with respect to p. For an edge (p, q) that is known to be in a particular quadrant, the region is a trapezoid as shown in Figure 5-2(a). In this case, the region is already convex and polygonal, and we find an embedding with no error beyond the optimal distortion.

The difficult case is when the edge (p, q) straddles two quadrants, i.e., is almost parallel to a coordinate axis. See Figure 5-2(b). In this case, the angular wedge intersects two sides of the ℓ_1 circle around p, and the region becomes a nonconvex 'V'. As before, we convexify this region by closing the mouth of the 'V'. The resulting region is also polygonal, so we can apply linear programming.

The worst-case error arises when (p, q) is exactly parallel to a coordinate axis. Then the smallest distance between p and a relaxed position for q is $1 - (\tan\gamma)/(1 + \tan\gamma)$ times the input distance between p and q. Again we can transform this contraction into an expansion by multiplying all distances by $1/[1 - (\tan\gamma)/(1 + \tan\gamma)]$, and the maximum expansion is at most $(1 + \epsilon)/[1 - (\tan\gamma)/(1 + \tan\gamma)]$:



(a) Convex case



(b) Nonconvex case

Figure 5-2: Feasible region of a point q with respect to p given the ℓ_1 distance within a multiplicative ϵ and given the counterclockwise angle to the x axis within an additive γ.

Theorem 5.2.7 *Given a complete graph, given the length of each edge with multiplicative error ϵ, and given the counerclockwise angle of every edge with respect to the x axis with additive error γ, we can compute in polynomial time an ℓ_1 embedding with angles of maximum additive error γ and distances of maximum multiplicative error*

$$\frac{1+\epsilon}{1-(\tan\gamma)/(1+\tan\gamma)} - 1 = (1+\epsilon)(\gamma + O(\gamma^3)) + \epsilon.$$

If we are given the approximate counterclockwise angles between incident pairs of edges, and the approximate counterclockwise angle between one edge and the x axis, then we can apply this theorem in combination with Lemma 5.2.5. If we are just given the approximate counterclockwise angles between incident pairs of edges, we can consider all "combinatorial rotations" with respect to the x axis, and extract whether each edge is roughly horizontal, roughly vertical, or substantially within one of the four quadrants. This partial information increases the region error for near-horizontal and near-vertical edges, and does not preserve the angle for all other edges, but will approximately preserve distances in the resulting embedding.

5.2.5 Extension to ℓ_∞

We can directly adapt the ℓ_1 algorithm to an ℓ_∞ algorithm. If we rotate an ℓ_∞ input by $45°$, and scale by a factor of $1/\sqrt{2}$ in each dimension, then we obtain an "identical" ℓ_1 input. The two inputs are identical in the sense that the ℓ_∞ distance between any pair of points in the ℓ_∞ input is equal to the ℓ_1 distance between that pair in the ℓ_1 input. Thus, we can apply the ℓ_1 embedding algorithm to the ℓ_1 input, and then undo the transformation, and we obtain an ℓ_∞ embedding of an ℓ_∞ input.

5.3 Embedding with Order Type

In this section, we consider the situation in which we are given all pairwise Euclidean distances between points in the plane as well as the "order type" of the points. The *order type* of a set of (labeled) points in the plane specifies, for each triple (p, q, r)

of points, the *orientation* of that triple, i.e., whether visiting those points in order (forming a triangle) proceeds clockwise or counterclockwise (or, in degenerate cases, collinear).

While we present the initial algorithm assuming that we know the entire order type, we later relax the assumption to knowing only the orientations of all triples including a fixed point p. This relaxation reduces the amount of required extra information from $\binom{n}{3}$ orientations to $\binom{n-1}{2}$ orientations. In fact, this information is equivalent to knowing the counterclockwise order of points around point p.

Orientations can be very sensitive to small perturbations, and we are told only approximate information about the pairwise distances between points, so for orientations to be useful we need to know a range in which they apply. For a set of points in the plane, we call a set of triples of points *totally δ-robust* if perturbing the x and y coordinates of every point by at most $\pm\delta$ does not change the orientations of any of the triples in the set. A set of orientations is *δ-robust* if perturbing the x and y coordinates of any single point by at most $\pm\delta$ does not change the orientation of any of the triples in the set. Obviously, total δ-robustness implies δ-robustness, but in fact, the two notions are equivalent up to constant factors:

Lemma 5.3.1 *If a set of triples is 3δ-robust, then it is totally δ-robust.*

Proof: If the set of triples is not totally δ-robust, there must be a perturbation of the points such that some triple (p, q, r) in the set changes orientation, i.e., p crosses the line segment between q and r. Because the total movement of p, q, and r in such a situation is at most 3δ, we can instead change the orientation of (p, q, r) by fixing q and r (and all other points except p) and just perturbing p by 3δ. But this contradicts the assumption that the set of triples is 3δ-robust. ∎

Our embedding algorithm assumes that the given orientations are totally δ-robust, for a particular choice of δ related to the distortion of the desired embedding. By Lemma 5.3.1, it suffices to assume that they are 3δ-robust. More precisely, the main theorem of this section is as follows:

Theorem 5.3.2 *Suppose that we are given a complete graph with specified edge lengths, and we are given an orientation for each triple of points involving one common point. Suppose we are promised that there is an embedding into the Euclidean plane with additive distortion ϵ in which these triples involving one common point have the specified orientations and are totally $c\epsilon$-robust (or $3c\epsilon$-robust). Then in polynomial time, multiplied by a factor of $O(\lg \Delta)$ if ϵ is not approximately known, we can compute an embedding f of the graph into the Euclidean plane with additive distortion at most $c\epsilon$, for a global constant c.*

Proof: First we guess ϵ up to a constant factor as in Section 2.4 by trying values of the form $\text{diam}(D)/2^i$ for $i = 0, 1, 2, \ldots$, where $\text{diam}(D)$ is the maximum distance in the given metric D. Then we apply Lemma 2.4.2 to guess the x coordinates of the vertices up to an additive $\pm 5\epsilon$. By setting $c \geq 5$, robustness tells us that the orientations remain valid within this fixing of x coordinates. Also, changing the x coordinates of the promised embedding f according to this assignment increases the additive distortion by at most 10ϵ.

Next we show how to assign the y coordinates of f such that, for every pair (v, w) of vertices, $|D[v, w] - \|f(v) - f(w)\|_2| \leq 3\epsilon$ (not counting the distortion introduced by fixing the x coordinates). Because the x coordinates are fixed, this constraint forces $f_y(v) - f_y(w)$ to lie within the union of (at most) two intervals, one interval for when $f_y(v) \geq f_y(w)$ and the other for when $f_y(v) \leq f_y(w)$. We show how to obtain the y coordinates by setting up a linear program, using the orientations to disambiguate between the two intervals.

We define a graph G whose vertex set is the same as the input graph. The edges of G are of two types: strong and weak. We connect vertices v and w by a *strong edge* in G if $D[v, w] \geq \sqrt{(f_x(v) - f_x(w))^2 + 3\epsilon^2}$. We connect two points v and w by a *weak edge* in G if there are two points u_1 and u_2, connected via paths of strong edges to w but not to v, such that $D[v, w] > \sqrt{(f_x(v) - f_x(w))^2 + \epsilon^2}$ and $f_x(u_1) \leq f_x(v) \leq f_x(u_2)$. The proofs of the following lemmas are very similar to Claims 4.1 and 4.2 of Bădoiu [B̆03] and hence omitted.

Lemma 5.3.3 *No two connected components of G overlap in x extent; that is, there is a vertical line (not passing through any vertices) that separates the vertices of the first component from the vertices of the second component.*

Call an edge $\{v, w\}$ *oriented up* if $f_x(v) \le f_x(w)$ and $f_y(v) \le f_y(w)$, and call an edge $\{v, w\}$ *oriented down* if $f_x(v) \le f_x(w)$ and $f_y(v) \ge f_y(w)$.

Lemma 5.3.4 *If we fix the orientation of an edge of G, we can uniquely determine the orientation of all other edges in the same connected component.*

By the definition of a strong edge, if there is no strong edge between two points v and w, the horizontal distance already fixed as $f_x(v) - f_x(w)$ is "good enough" for a 3ϵ-approximation. To ensure that the distortion remains sufficiently small, we form the constraint $D[v, w] + \epsilon \ge \|f(v) - f(w)\|$, which is equivalent to the pair of linear constraints

$$-\sqrt{(D[p,q]+\epsilon)^2 - (f_x(p) - f_x(q))^2} \le f_y(p) - f_y(q) \le \sqrt{(D[p,q]+\epsilon)^2 - (f_x(p) - f_x(q))^2}.$$

For any edge $\{v, w\} \in E(G)$ that is oriented up and for which $f_x(v) \le f_x(w)$, we form this linear constraint on f_y:

$$\sqrt{(f_x(w) - f_x(v))^2 + (f_y(w) - f_y(v))^2} - \epsilon \le D[v,w] \le \sqrt{(f_x(w) - f_x(v))^2 + (f_y(w) - f_y(v))^2} + \epsilon,$$

or equivalently,

$$\begin{aligned} \sqrt{D[v,w]^2 - 2\epsilon D[v,w] + \epsilon^2 - (f_x(w) - f_x(v))^2} &\le f_y(w) - f_y(v) \\ &\le \sqrt{D[v,w]^2 + 2\epsilon D[v,w] + \epsilon^2 - (f_x(w) - f_x(v))^2}. \end{aligned}$$

We have a similar relation for edges $\{v, w\} \in E(G)$ that are oriented down.

Now, using Lemmas 5.3.3 and 5.3.4 and the description above, we can obtain a $c\epsilon$-approximation solution for the problem provided that G has only one connected component. However, if there are several connected components, each connected component can be oriented up or down, and the total number of cases can be exponential.

Instead, we use the given orientations of triples to disambiguate the orientations of components. Because the orientations are totally $c\epsilon$-robust, and we guess the x and y coordinates within $\epsilon + 10\epsilon + 3\epsilon = 14\epsilon$ total additive distortion (counting the ϵ distortion in f), the orientations remain correct if we set $c \geq 14$. Without loss of generality, we assume that the leftmost component is oriented up. Now consider a point v in this component. We show that, for each other component C, we can use orientations of triples involving v to determine whether C is oriented up or down. Consider a strong edge $(u, w) \in C$. (Such a strong edge should exist, because otherwise C has only one point and its orientation is trivial.) Because there is no strong edge between u and v, the segment connecting u to v is almost horizontal (see the definition of strong edge). Using this property, using that (u, w) is a strong edge, and using the orientation of the triple (v, u, w), we can determine the orientation of edge (u, w) and thus by Lemma 5.3.4 the orientation of the whole component C. Thus, fixing the orientation of the leftmost component, we can determine the orientation of all edges of other components. Finally, by setting up the following linear program, we obtain the desired approximation embedding:

$$\begin{aligned}
&\sqrt{D[v,w]^2 - 2\epsilon D[v,w] + \epsilon^2 - (f_x(w) - f_x(v))^2} \\
\le\ & f_y(w) - f_y(v) \\
\le\ & \sqrt{D[v,w]^2 + 2\epsilon D[v,w] + \epsilon^2 - (f_x(w) - f_x(v))^2} \\
& \qquad \text{if } \{v,w\} \in E \text{ is oriented up and } f_x(w) \ge f_x(v), \\
&\sqrt{D[v,w]^2 - 2\epsilon D[v,w] + \epsilon^2 - (f_x(w) - f_x(v))^2} \\
\le\ & f_y(v) - f_y(w) \\
\le\ & \sqrt{D[v,w]^2 + 2\epsilon D[v,w] + \epsilon^2 - (f_x(w) - f_x(v))^2} \\
& \qquad \text{if } \{v,w\} \in E \text{ is oriented down and } f_x(w) \ge f_x(v), \\
&-\sqrt{(D[v,w] + \epsilon)^2 - (f_x(v) - f_x(w))^2} \\
\le\ & f_y(v) - f_y(w) \\
\le\ & \sqrt{(D[v,w] + \epsilon)^2 - (f_x(v) - f_x(w))^2} \\
& \qquad \text{if } \{v,w\} \notin E.
\end{aligned}$$

■

5.4 Embedding with Distribution Information

In this section we consider embedding the complete graph on n vertices into the Euclidean plane while approximately minimizing additive distortion of specified edge lengths that come from a kind of adversarial distribution. Roughly speaking, we are given the promise that the distances satisfy that, after perturbing each distance within $\pm\epsilon$, the resulting distances are exactly the pairwise distances between n points sampled uniformly from the unit square $[0,1]^2$. More precisely, the specified distances come from first randomly sampling n points uniformly from the unit square, then exactly measuring their Euclidean distances, and then *adversarially* perturbing each distance within $\pm\epsilon$. Our goal is to construct an embedding with additive distortion $O(\epsilon)$.

Theorem 5.4.1 *There is a polynomial-time algorithm that, given a complete graph with edge lengths arising from the adversarial distribution described above, finds an embedding that has additive distortion $O(\epsilon)$ with probability $1 - o(1)$, as long as $\epsilon = \omega(1/\sqrt{n})$. The algorithm is deterministic; the probability is taken over the uniform sample of points in the unit square.*

Proof: Let r be any value such that $r = \omega(1/\sqrt{n})$ and $r = O(\epsilon)$. The algorithm first guesses a "frame" for the square, and then uses a "triangulation" approach relative to this frame:

1. For every quadruple (v_1, v_2, v_3, v_4) of vertices (the *frame*), construct the following embedding f:

   (a) Embed v_i, $i \in \{1, 2, 3, 4\}$, as follows: $f(v_1) = (0, 0)$, $f(v_2) = (0, 1)$, $f(v_3) = (1, 1)$, and $f(v_4) = (1, 0)$.

   (b) Embed every other vertex w to an arbitrary point $f(w)$ in the region

   $$R_w = [0,1]^2 \cap \bigcap_{i=1,2,3,4} R\left(f(v_i), D[v_i, w], \epsilon + 2\sqrt{2}r\right),$$

   where $R(p, r, \delta)$ is the annulus centered at point p with inner radius $r - \delta$ and outer radius $r + \delta$. If R_w is empty, we ignore this (incomplete) embedding and skip this iteration of the loop.

2. Report the embedding f with the smallest additive distortion.

This algorithm has the feature that every constructed embedding maps the vertices into the unit square. It remains to analyze the quality of the best embedding f found. Let f^* denote the uniformly random embedding into the unit square that we assume exists, and which has additive distortion ϵ.

We start by showing that there is a good choice of the frame. The following claim follows from basic calculations:

Claim 5.4.2 *With probability $1 - o(1)$, each of the four $r \times r$ subsquares of the unit square that each share a corner with the unit square contain $f^*(v)$ for some $v \in V$.*

We condition on the event that there is at least one vertex v_1, v_2, v_3, and v_4 mapped via f^* to the lower-left, lower-right, upper-right, and upper-left corner sub-squares, respectively. (By Claim 5.4.2, this event happens with probability $1-o(1)$.) Consider the iteration of Step 1 of the algorithm that chooses this quadruple of points for the frame. If we modify f^* by performing the assignment as in Step 1(a) of the algorithm, then the resulting embedding has additive distortion at most $\epsilon + 2\sqrt{2}r$. Therefore, in this iteration, every region R_w includes $f^*(w)$ and is thus nonempty.

It suffices to show that the diameter of each set R_w is $O(\epsilon + r)$. Consider any vertex w other than v_1, v_2, v_3, and v_4. We need the following claim, which can be proved using the same type of argument as in the proof of Theorem 5.3.2:

Claim 5.4.3 *Consider any two points $p, q \in [0,1]^2$ and any $r_1, r_2, \delta > 0$ such that $r_1, r_2 = O(\|p - q\|)$. The set $R(p, r_1, \delta) \cap R(q, r_2, \delta)$ is contained in a strip of width $O(\delta)$ whose direction (i.e., an infinite line contained in the strip) is orthogonal to the line passing through p and q.*

Recall that R_p is an intersection of four annuli (and the unit square). Applying Claim 5.4.3 to the annuli around points $(0,0)$ and $(1,0)$, we conclude that R_p is contained in a vertical strip of width $O(\epsilon + r)$. Applying Claim 5.4.3 to the annuli around points $(0,0)$ and $(0,1)$, we conclude that R_p is contained in a horizontal strip of the same width. It follows that that the diameter of R_p is $O(\epsilon + r)$ as claimed, and therefore that the additive distortion of the embedding f computed by the algorithm is $O(\epsilon + r)$. ■

5.5 Embedding with Range Graphs

In this section we are interested in embedding a graph with specified edge lengths into the line subject to the following condition. An embedding $f : V \to \mathbb{R}$ of a graph $G = (V, E)$ with edge lengths specified by D satisfies the *range condition* if, for every three points $p, q, r \in V$, (a) if $\{p, q\} \in E$ and $\{p, r\} \notin E$, $|f(p) - f(q)| \leq |f(p) - f(r)|$, and (b) if $\{p, q\}, \{p, r\} \in E$, $|f(p) - f(q)| \leq |f(p) - f(r)|$ precisely if $D[p, q] \leq D[p, r]$. Among all such embeddings, we will find one that minimizes the additive distortion with respect to the specified edge lengths on G. Part (b) of this definition will be satisfied provided the difference between adjacent distances in a near-optimal embedding is at least the additive distortion.

5.5.1 The Exact Case

In this subsection we consider embedding with zero distortion:

Lemma 5.5.1 *Given a graph G with edge lengths specified by D, we can check in polynomial time whether there is an embedding f that satisfies the range condition and matches D exactly on the edges of f, and construct such an embedding if it exists.*

Proof: Without loss of generality we assume that the graph G is connected. Let p be the leftmost point in an embedding f into the line that satisfies the conditions of the lemma. We guess p by enumerating all $|V|$ possibilities. Without loss of generality, p has coordinate 0. All neighbors of p in G lie to the right of p. Let q be such a neighbor. Let r be a neighbor of q but not a neighbor of p. By the range condition, we have $|f(p) - f(r)| > |f(p) - f(q)|$. Therefore, $f(r) > f(q)$ and thus $f(r) = f(q) + D[q, r]$. By traversing G in a breadth-first manner, we can reconstruct f. The running time of our algorithm is $O(|V| \cdot |E|)$. ■

5.5.2 The Additive Error Case

In this subsection we consider the case when the optimal embedding has minimum additive distortion ϵ. We say an edge $(p, q) \in G$ is a *forward edge* if $f(p) \leq f(q)$ and a *backward edge* if $f(p) > f(q)$. We call this distinction the *orientation* of an edge. Note that if (p, q) is a forward edge then (q, p) is a backward edge.

Lemma 5.5.2 *Given a graph G with edge lengths specified by D for which there is an embedding f that satisfies the range condition, and for any two incident edges $\{p, q\}$ and $\{q, r\}$ in G, we can determine the orientation of (q, r) in f given the orientation of (p, q) in f using just D.*

Proof: Without loss of generality (p, q) is a forward edge and $D[p, q] > D[q, r]$. By part (b) of the range condition, if $D[p, r] < D[p, q]$, then (q, r) must be a backward edge. By both parts of the range condition, if $D[p, r] > D[p, q]$ or $D[p, r]$ is unknown, then (q, r) must be a forward edge. ∎

Theorem 5.5.3 *Given a graph G with edge lengths specified by D, we can construct in polynomial time an embedding f that satisfies the range condition and matches D up to the minimum possible additive distortion subject to the range condition.*

Proof: Let (p, q) be an edge in G. Without loss of generality we can assume (p, q) is a forward edge. Lemma 5.5.2 implies that we know the orientation of all the incident edges. By applying this argument multiple times, we can determine the orientation of all the edges within the connected component of G containing p. We cannot determine the relative orientation between different connected components, but this is not necessary. By placing the locally embedded connected components far away from each other, the resulting embedding satisfies the range condition. Knowing the orientations, we can construct the following linear program which minimizes additive distortion:

$$\begin{array}{lll}
\text{minimize} & \epsilon & \\
\text{subject to} & f(p) + D[p, q] - \epsilon < f(q) < f(p) + D[p, q] + \epsilon & \text{if } (p, q) \text{ is a forward edge,} \\
 & f(p) - D[p, q] - \epsilon < f(q) < f(p) - D[p, q] + \epsilon & \text{if } (p, q) \text{ is a backward edge.}
\end{array}$$

∎

In Section 5.6, we show that embedding a graph with given edge lengths in two-dimensional ℓ_1 and ℓ_2 space, even using exact distances and a more restricted form of range-condition, is NP-hard.

5.6 Hardness Results

Saxe [Sax79] proved that deciding embeddability of a given graph with exact ℓ_2 edge lengths is strongly NP-hard in d dimensions, for any $d \geq 1$. Independently, Yemini [Yem79] proved weak NP-hardness of the same problem for $d = 2$ with a simple reduction from Partition. Here we prove weak NP-hardness for both ℓ_1 and ℓ_2 in 2D, even when the graph satisfies the *constant-range condition*: two vertices v, w are connected by an edge precisely when their distance is at most a fixed range r. This condition is a special case of the *(variable) range condition* defined in Section 5.5, and hence our hardness results apply under that restriction as well. One interesting feature of our restricted form of the problem is that the problem is not hard in 1D, and thus our proofs require us to use the structure of 2D. In contrast, previous hardness proofs start with 1D, and then trivially extend to higher dimensions.

5.6.1 ℓ_2 Case

Theorem 5.6.1 *It is NP-hard to decide whether a given graph with exact ℓ_2 edge lengths and satisfying the constant-range condition has an embedding with zero distortion.*

Our reduction from Partition is sketched in Figure 5-3. The range is $1.1L$, where L is a large number to be chosen later. In any embedding of our graph, all vertices lie roughly on a square grid with edge lengths $L/2$. We use strips of k vertices spaced every $L/2$ units to build rigid bars of length $kL/2$; the strips are rigid because each vertex can see the next two vertices in the strips. We use right isosceles triangles with edge lengths $L/2$, $L/2$, and $L/\sqrt{2}$ to force angles of 90°. All other pairs of vertices have distance at least $\sqrt{5}\big/2 > 1.1L$, so are not within range.

For a given instance $a_1, a_2, \ldots, a_n$ of Partition, we construct $2n$ edges, two with length $(L + a_i)/2$ for each i, and force them all to be parallel. We choose L large enough so that $\sum_{i=1}^{n} a_i < 0.1L$. For each pair of edges of length $(L + a_i)/2$, we also create a pair of edges of length $L/2$, so that the absolute horizontal shift caused by these four edges is $(L + a_i) - L = a_i$. Each such quadruple of edges can be

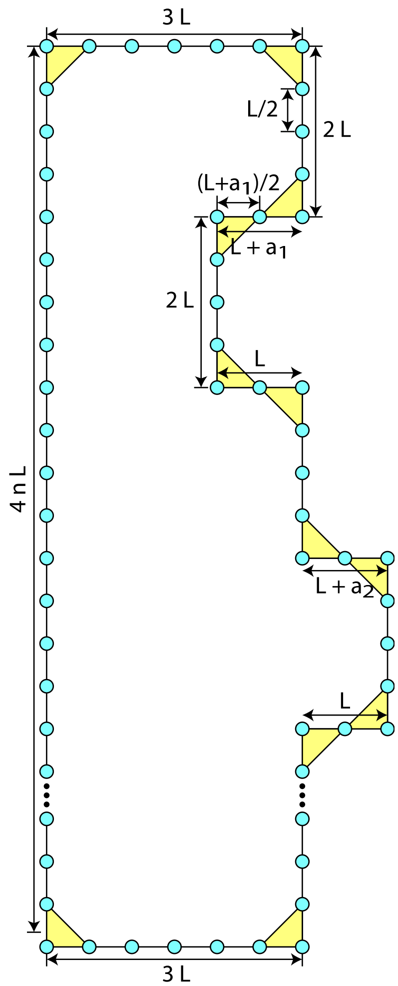


Figure 5-3: Our reduction from Partition to ℓ_2 embedding of a graph satisfying the constant-range condition. In the reduction, the a_i's are much smaller than L, and in this drawing, the a_i's are drawn as 0.

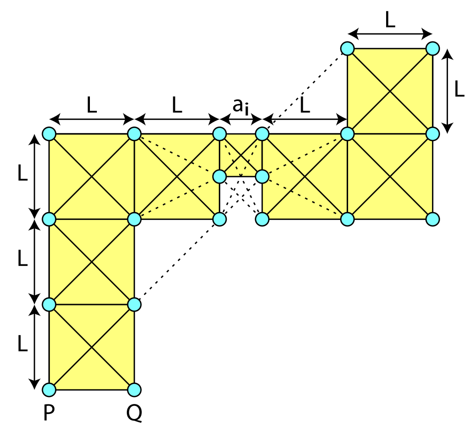


Figure 5-4: Analogous gadgets for use in Figure 5-3 for the ℓ_∞ case. Here a_i is drawn larger than reality. Dotted edges are present, but not necessary for rigidity.

independently flipped so that the shift is either a_i or $-a_i$. Finally, we add another connection between the two extreme edges which forces the total shift to be 0. Thus, a distortion-free embedding corresponds to a solution to Partition and vice versa.

5.6.2 ℓ_1 and ℓ_∞ Case

We prove the first hardness result about embedding with exact ℓ_1 or ℓ_∞ distances:

Theorem 5.6.2 *It is NP-hard to decide whether a given graph with exact ℓ_∞ edge lengths (or equivalently, exact ℓ_1 edge lengths) and satisfying the constant-range condition has an embedding with zero distortion.*

The proof is similar to the ℓ_2, except that the gadgets are slightly more complicated; see Figure 5-4. The radius r is now exactly L. We use a sequence of attached $L \times L$ boxes in place of a strip of vertices. As before, this construction acts as a rigid bar, except that it can be flipped. (In Figure 5-4, vertices p and q can be swapped.) To perturb a length by a_i from a multiple of L, we add a small $a_i \times a_i$ box and attach it in the middle of the strip. This box is in fact rigid and cannot be flipped with respect to its neighbors. Thus, the construction can be plugged into Figure 5-3 and we have the theorem.

5.7 Open Problems

An important open problem in this area is whether there is a polynomial-time algorithm for approximately minimizing additive distortion given all pairwise distance information and no extra information. Our pseudo-quasipolynomial-time algorithm is one step in this direction. The analogous problem for multiplicative distortion seems even harder.

A general theme of our work is to consider the case in which we do not know all distances. Another approach for making this case tractable is to constrain the connectivity to something less than $n-1$ (for the complete graph). For example, what can we say about c-connected graphs for sufficiently large c, or cn-connected graphs for $c < 1$? These special cases will still likely require extra information, because even for the case where we know all pairwise distances, we do not know approximation algorithms without extra information except for ℓ_1 and additive distortion [B\v{0}3].

It would seem natural to obtain angle estimates in a graph G "for free" using (approximate) distances in $G \cup G^2$, by analyzing triangles (p, q, r) in $G \cup G^2$. There are two problems with this approach. The first problem is that two vertices p, q in a triangle may be much closer to each other than to the third vertex r, and the multiplicative errors on distances allow p and q to spin around each other and allow p and q to have any angle. This problem can be surmounted by assuming that the ratio of lengths between any two incident edges is bounded. The second, more serious problem is that it is difficult to decode the orientations of triangles and hence the signs of the angles using purely distance information. We conjecture that this information can be decoded using distances in $G \cup G^2 \cup G^3 \cup G^4 \cup G^5 \cup G^6$, because 6-connected graphs have unique embeddings [JJ05].

Even with just distance information, the complexity of one interesting variation remains unresolved. Given a graph that is generically uniquely embeddable, in the sense that almost any assignment of edge lengths induces a unique embedding, can we construct the unique embedding for almost any assignment of edge lengths? Jackson and Jordán [JJ05] recently showed that, in polynomial time, we can test whether a graph has this property, but the proof is not entirely constructive. Another example of an NP-hard problem that can be solved in polynomial time almost always is Subset Sum. Our hardness reductions for embedding are based on Subset Sum, so there is hope that nongeneric examples are the only obstruction to polynomial-time algorithms.

In this chapter we have focused on embedding metrics into the plane, but it would be natural to try to extend our work to slightly higher dimensions, in particular 3D which is important in some appliactions. Some of our results extend relatively easily. For example, in fixed dimension, given the approximate angle of every edge with respect to every coordinate axis (with additive error), and given distances with multiplicative error, we can apply the constant-factor approximation algorithms described in Sections 5.2.2 and 5.2.4.

Acknowledgments

We thank the anonymous referees of SoCG for their helpful comments.

Bibliography

[ABC+05] Ittai Abraham, Yair Bartal, T-H. Hubert Chan, Kedar Dhamdhere, Anupam Gupta, Jon Kleinberg, Ofer Neiman, and Aleksandrs Slivkins. Metric embeddings with relaxed guarantees. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, 2005. To appear.

[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 *Proceedings of the Symposium on Foundations of Computer Science*, 2005.

[AFR85] Noga Alon, Peter Frankl, and Vojtech Rödl. Geometrical realization of set systems and probabilistic communication complexity. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science*, pages 277–280, Portland, Oregon, 1985.

[Alo95] Noga Alon. Tools from higher algebra. In R. L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of combinatorics*, volume 2, chapter 32, pages 1749–1783. MIT Press, 1995.

[Bar96] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. *Proceedings of the Symposium on Foundations of Computer Science*, 1996.

[BC03] Bo Brinkman and Moses Charikar. On the impossibility of dimension reduction in ℓ_1. In *Proceedings of the 44th Symposium on Foundations of Computer Science*, pages 514–523, 2003.

[BCIS05] M. Bădoiu, J. Chuzhoy, P. Indyk, and A. Sidiropoulos. Low-distortion embeddings of general metrics into the line. *Proceedings of the 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.

[BDHI04] M. Bădoiu, E. Demaine, M. Hajiaghai, and P. Indyk. Embeddings with extra information. *Proceedings of the ACM Symposium on Computational Geometry*, 2004.

[BJDG+03] Z. Bar-Joseph, E. D. Demaine, D. K. Gifford, A. M. Hamel, Tommi S. Jaakkola, and Nathan Srebro. K-ary clustering with optimal leaf ordering for gene expression data. *Bioinformatics*, 19(9):1070–8, 2003.

[BKL99] Bonnie Berger, Jon Kleinberg, and Tom Leighton. Reconstructing a three-dimensional model with arbitrary errors. *Journal of the ACM*, 46(2):212–235, 1999.

[BL04] Y. Bilu and N. Linial. Monotone maps, sphericity and bounded second eigenvalue. arXiv:math.CO/0401293, January 2004.

[BM04a] Y. Bartal and M. Mendel. Dimension reduction for ultrametrics. In *Proceedings of the 15th ACM-SIAM Symp. on Discrete Algorithms*, pages 664–665, 2004.

[BM04b] Yair Bartal and Manor Mendel. Dimension reduction for ultrametrics. In *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 664–665, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[BMMV02] R. Babilon, J. Matousek, J. Maxovă, and Pavel Valtr. Low-distortion embeddings of trees. In *GD '01: Revised Papers from the 9th International Symposium on Graph Drawing*, pages 343–351, London, UK, 2002. Springer-Verlag.

[Bor33] K. Borsuk. Drei Sätze über die n-dimensionale euklidische Sphäre. *Fund. Math.*, 20:177–190, 1933.

[Bou85] J. Bourgain. On lipschitz embedding of finite metric spaces into hilbert space. *Isreal Journal of Mathematics*, 52:46–52, 1985.

[B03̆] M Bădoiu. Approximation algorithm for embedding metrics into a two-dimensional space. *14th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2003.

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

[CH88] G. M. Crippen and T. F. Havel. *Distance Geometry and Molecular Conformation*. John Wiley & Sons, 1988.

[ČHH01] Srdan Čapkun, Maher Hamdi, and Jean-Pierre Hubaux. GPS-free positioning in mobile ad-hoc networks. In *Proceedings of the 34th Hawaii International Conference on System Sciences*, pages 3481–3490, January 2001.

[CL92] C. Coullard and A. Lubiw. Distance visibility graphs. *Internat. J. Comput. Geom. Appl.*, 2(4):349–362, 1992.

[Con91] R. Connelly. On generic global rigidity. In P. Gritzman and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, volume 4 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 147–155. AMS Press, 1991.

[CS74] J. P. Cunningham and R. N. Shepard. Monotone mapping of similarities into a general metric space. *Journal of Mathematical Psychology*, 11:335–364, 1974.

[CS98] B. Chor and M. Sudan. A geometric approach to betweennes. *SIAM Journal on Discrete Mathematics*, 11(4):511–523, 1998.

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

[EHKN99] H. Everett, C. T. Hoàng, K. Kilakos, and M. Noy. Distance segment visibility graphs. Manuscript, 1999. http://www.loria.fr/~everett/publications/distance.html.

[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.

[ES63] Paul Erdős and Horst Sachs. Reguläre Graphen gegebener Taillenweite mit minimaler Knotenzahl. *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe*, 12:251–257, 1963.

[FCK96] M. Farach-Colton and S. Kannan. Efficient algorithms for inverting evolution. *28th Symposium on Theory of Computing*, 1996.

[Fei00] U. Feige. Approximating the bandwidth via volume respecting embeddings. *Journal of Computer and System Sciences*, 60(3):510–539, 2000.

[FK99] M. Farach and S. Kannan. Efficient algorithms for inverting evolution. *Journal of the ACM*, 46, 1999.

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

[FKW95] M. Farach, S. Kannan, and T. Warnow. A robust model for finding optimal evolutionary trees. *Algorithmica*, 13(1-2):155–179, 1995.

[GRK04] Ashish Goel, Sanatan Rai, and Bhaskar Krishnamachari. Sharp thresholds for monotone properties in random geometric graphs. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 580–586, 2004.

[Gup00a] A. Gupta. Embedding trees into low dimensional euclidean spaces. *Discrete and Computational Geometry*, 24(1):105–116, 2000.

[Gup00b] Anupam Gupta. Improved bandwidth approximation for trees. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2000.

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

[Har67] E. F. Harding. The number of partitions of a set of N points in k dimensions induced by hyperplanes. *Proceedings of the Edinburgh Mathematical Society, Series II*, 15:285–289, 1966/1967.

[Hen92] B. Hendrickson. Conditions for unique graph realizations. *SIAM Journal on Computing*, 21(1):65–84, February 1992.

[Hen95] B. Hendrickson. The molecule problem: Exploiting structure in global optimization. *SIAM Journal on Optimization*, 5:835–857, 1995.

[HIL98] J. Hastad, L. Ivansson, and J. Lagergren. Fitting points on the real line and its application to rh mapping. *Lecture Notes in Computer Science*, 1461:465–467, 1998.

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

[Hol72] W. Holman. The relation between hierarchical and euclidean models for psychological distances. *Psychometrika*, 37(4):417–423, 1972.

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

[IM04] Piotr Indyk and Jiřı Matoušek. Low-distortion embeddings of finite metric spaces. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 8, pages 177–196. CRC Press, second edition, 2004.

[Iva00] L. Ivansson. Computational aspects of radiation hybrid. *Doctoral Dissertation, Department of Numerical Analysis and Computer Science, Royal Institute of Technology*, 2000.

[JJ05] Bill Jackson and Tibor Jordán. Connected rigidity matroids and unique realizations of graphs. *Journal of Combinatorial Theory, Series B*, 94(1):1–29, 2005.

[JL84] W. B. Johnson and J. Lindenstrauss. Extensions of lipshitz mapping into hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[Kir34] M. D. Kirszbraun. Über die zusammenziehenden und lipschitzschen Transformationen. *Fund. Math.*, 22:77–108, 1934.

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

[Kru64a] J.B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.

[Kru64b] J.B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:115–129, 1964.

[LLR95] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[LN04a] J. R. Lee and A. Naor. Absolute lipschitz extendability. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 2004.

[LN04b] James R. Lee and Assaf Naor. Embedding the diamond graph in L_p and dimension reduction in L_1. *Geometric and Functional Analysis*, 14(4):745–747, 2004.

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

[LTW+90] Joseph Y.-T. Leung, Tommy W. Tam, C. S. Wong, Gilbert H. Young, and Francis Y.L. Chin. Packing squares into a square. *J. Parallel Distrib. Comput.*, 10(3):271–275, 1990.

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

[Mat97] JiříMatoušek. On embedding expanders into l_p spaces. *Israel Journal of Mathematics*, 102:189–197, 1997.

[MDS] Algorithms for multidimensional scaling. *http://dimacs.rutgers.edu/SpecialYears/2001_Data/Algorithms/MDSdescription.html*.

[MIT] MIT CSAIL Networks and Mobile Systems group. The Cricket indoor location system. http://nms.csail.mit.edu/projects/cricket/.

[Opa79] J. Opatrny. Total ordering problem. *SIAM J. Computing*, 8:111–114, 1979.

[PCB00] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket location-support system. In *Proceedings of 6th Annual International Conference on Mobile Computing and Networking*, pages 32–43, Boston, MA, August 2000.

[PMBT01] Nissanka B. Priyantha, Allen K. L. Miu, Hari Balakrishnan, and Seth Teller. The Cricket compass for context-aware mobile applications. In *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, pages 1–14, Rome, Italy, July 2001.

[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.

[Sax79] J. B. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. In *Proceedings of the 17th Allerton Conference on Communication, Control, and Computing*, pages 480–489, 1979.

[Sax80a] J. B. Saxe. Dynamic-programming algorithms for recognizing small-bandwidth graphs in polynomial time. *SIAM J. Algebraic Discrete Methods*, 1:363–369, 1980.

[Sax80b] James B. Saxe. Two papers on graph embedding problems. Technical Report CMU-CS-80-102, Department of Computer Science, Carnegie-Mellon University, January 1980.

[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.

[SFC04] R. Shah and M. Farach-Colton. On the complexity of ordinal clustering. *Journal of Classification*, 2004. To appear.

[She62a] R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function 1. *Psychometrika*, 27:125–140, 1962.

[She62b] R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function 2. *Psychometrika*, 27:216–246, 1962.

[SRB01] C. Savarese, J. Rabaey, and J. Beutel. Locationing in distributed ad-hoc wireless sensor networks. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 2037–2040, Salt Lake City, UT, May 2001.

[Tor52] W. S. Torgerson. Multidimensional scaling I: Theory and method. *Psychometrika*, 17(4):401–414, 1952.

[Ung98] W. Unger. The complexity of the approximation of the bandwidth problem. *Proceedings of the Symposium on Foundations of Computer Science*, 1998.

[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.

[War68] Hugh E. Warren. Lower bounds for approximation by nonlinear manifolds. *Transactions of the American Mathematical Society*, 133:167–178, 1968.

[Yem79] Y. Yemini. Some theoretical aspects of position-location problems. In *Proceedings of the 20th Annual IEEE Symposium on Foundations of Computer Science*, pages 1–8, 1979.