Theory Comput. Systems **39**, 93–111 (2006)
DOI: 10.1007/s00224-005-1259-6

Theory of Computing Systems

Approximation Algorithms for Minimizing Average Distortion*

Kedar Dhamdhere,[1] Anupam Gupta,[1] and R. Ravi[2]

[1]School of Computer Science, Carnegie Mellon University,
Pittsburgh, PA 15213, USA
{kedar,anupamg}@cs.cmu.edu

[2]Tepper School of Business, Carnegie Mellon University,
Pittsburgh, PA 15213, USA
ravi@cmu.edu

Abstract. This paper considers embeddings f of arbitrary finite metrics into the line metric $\Re$ so that none of the distances is shrunk by the embedding f; the quantity of interest is the factor by which the average distance in the metric is stretched. We call this quantity the *average distortion* of the non-contracting map f.

We prove that finding the best embedding of even a tree metric into a line metric to minimize the average distortion is NP-hard, and hence focus on *approximating* the average distortion of the best possible embedding for the given input metric. We give a constant-factor approximation for the problem of embedding general metrics into the line metric. For the case of n-point tree metrics, we provide a quasi-polynomial time approximation scheme which outputs an embedding with distortion at most $(1+\varepsilon)$ times the optimum in time $n^{O(\log n/\varepsilon^2)}$. Finally, when the average distortion is measured only over the endpoints of the edges of an input tree metric, we show how to exploit the structure of tree metrics to give an exact solution in polynomial time.

1. Introduction

Metric embeddings have recently attracted much attention in theoretical computer science because of their many algorithmic applications. These range from simplifying the structure of the input data for problems in approximation algorithms and online computation [4], [7], [8], [16], [19], [25], serving as well-roundable relaxations of important

* The first author was supported by NSF ITR Grants CCR-0085982 and CCR-0122581. The third author was supported in part by NSF Grant CCR-0105548 and ITR Grant CCR-0122581.

NP-hard problems [6], [10], [11], [13], [18], [28] or simply by being the object of study arising from applications such as computational biology [1], [17]. Embedding techniques have become an indispensable addition to the algorithms toolbox, providing powerful and elegant solutions to many algorithmic problems (see, e.g., [30, Chapter 15] and [22] for surveys).

An embedding of a metric (V, d) into a "simpler" host metric (H, δ) is a map $f\colon V \to H$. Informally, the embedding is "good" if the distance $d(u, v)$ between any two points u, v in (V, d) is close to the distance $\delta(f(u), f(v))$ between their images in (H, δ). To make this formal, we call an embedding *non-contracting* if the map does not decrease any distances, i.e., $d(x, y) \le \delta(f(x), f(y))$ for all $x, y \in V$. We always deal with maps that are one–one, and hence abbreviate $\delta(f(x), f(y))$ to $\delta(x, y)$ if there is no danger of confusion. Unless explicitly specified, this paper only deals with non-contracting embeddings.

A popular measure of goodness of a non-contracting embedding f is the *distortion* $\alpha = \alpha(f)$, which is defined as

$$\text{distortion } \alpha = \max_{x,y\in V} \frac{\delta(x, y)}{d(x, y)}. \tag{1}$$

In this paper we instead consider a related measure: that of *average distortion*, which is defined as

$$\text{average distortion } \rho(f) = \frac{\sum_{x,y\in V} \delta(x, y)}{\sum_{x,y\in V} d(x, y)}. \tag{2}$$

Note that this is the factor by which the average distortion of the metric is changed by the map f; this is the object of attention in this paper.

Apart from considering the average distortion as a measure of goodness of the embedding, this paper differs from many previous papers in another important aspect: we do not want to give worst-case uniform bounds to embed classes of metrics into host spaces, but instead want to approximate the best embedding for the particular input metric. This is best shown by a concrete example: in [29], Matoušek showed that *any* metric space (V, d) can be embedded into the real line with distortion $O(|V|)$; furthermore, the result is existentially tight, as the n-cycle cannot be embedded into the line with distortion $o(|V|)$ (see, e.g., [32] and [21]). However, no algorithm is known which offers *per-instance* guarantees; hence, while it may be possible to embed (V, d) into $\Re$ with distortion $\alpha = O(1)$, there is no algorithm known which gives us embeddings with distortion, say, that is even within $O(|V|^{1-\varepsilon})$ times ρ.

Hence, in contrast to analyses of most embedding techniques and algorithms known which only offer uniform bounds on the distortion of the embeddings, we try to *approximate* the (average) distortion of the embedding for the *given input metric* to better than these uniform bounds. Very few such results are known; one notable exception is the remark of [28] that the optimal embedding of any finite metric into (unbounded dimensional) Euclidean spaces to minimize the distortion can be computed as a solution to a semidefinite program. Kenyon et al. [23] recently gave an approximation algorithm for minimizing distortion of bijections between point sets in the line.

1.1. *Our Results*

In this paper we consider non-contracting embeddings f into the line $\Re$, and give results on approximating the average distortion incurred by such embeddings. Note that for an optimal embedding into $\Re$ to be non-contracting, it is necessary and sufficient for the distance $|f(u) - f(v)|$ between an adjacent pair of vertices u, v on the line to be the same as their distance $d(x, y)$ in the input metric (V, d). Hence we restrict ourselves to only such embeddings. We can think of such an embedding into the line as defining a tour on the nodes of the original metric, where we start from the leftmost vertex on the line and visit the vertices in order from left to right.

Our results build on this simple observation, and demonstrate a close relationship between minimizing average distortion and the related problems of finding short TSP tours [26], minimum latency tours [9], [20], [3], and optimal k-repairmen solutions [15]. In particular, we prove the following results:

1. **Hardness for average distortion:** In Section 2.1 we prove that the problem of finding the minimum average distortion non-contracting embedding of finite metrics into the line is NP-hard, even when the input metric is a tree metric. The proof proceeds via a reduction from the Minimum Latency Problem on trees [35].
2. **Constant-factor approximations:** In Section 2.2 we give an algorithm that embeds any metric (V, d) into the line with average distortion that is within a constant of the minimum possible over all non-contracting embeddings. In fact, we prove a slightly more general bound on non-contracting embeddings into k-spiders (i.e., homeomorphs of stars with k leaves). This result uses a lower bound on the minimum average distortion of a non-contracting embedding into a k-spider in terms of the minimum k-repairmen tour [15] on the metric.
3. **QPTAS on trees:** For tree metrics on n nodes, we give an algorithm for finding a $(1+\varepsilon)$-approximation to the minimum average distortion non-contracting embedding into a line in $n^{O(\log n/\varepsilon^2)}$ time. Our algorithm, which appears in Section 3, uses a lower bound on the minimum average distortion related to the TSP tour length and latencies of appropriately chosen segments of an optimal tour. In this way it extends the ideas of Arora and Karakostas [5] for minimizing latency on trees to the more general time-dependent TSPs [9], and provides a quasi-polynomial time approximation scheme (QPTAS) for the latter problem as well.
4. **Poly-time algorithm for tree-edge distortion:** For a tree metric as input, if the minimum average distortion is measured only over the endpoints of the edges of the tree (we call this objective the average tree-edge distortion), then we show that an embedding following a certain Euler tour of the tree is optimal. In Section 4 we show how to find this tour in polynomial time by dynamic programming.

Remark 1.1. It is important to note that while any non-contracting embedding can be converted to a non-expanding embedding with the same average distortion by scaling down all the distances, the converse is not true. Indeed, a non-expanding embedding f might not be one-one, and may map two points in the guest metric to the same point in the host metric. This is a crucial difference between the two problems, and hence our result does not give a constant-factor approximation for the average distortion of non-expanding embeddings into the line $\Re$.

1.2. *Related Work*

The definition of average distortion is not new; e.g., Alon et al. [2] study the question of embedding a metric into a tree with low average distortion. In recent work on average distortion that is closer to ours, Rabinovich [31] proves bounds on average distortion of *non-expanding* embeddings into a line and shows the close connection between this and the max-flow min-cut ratio for concurrent multicommodity flow with applications to finding quotient cuts in graphs [27].

While our problem appears similar to that of finding the *Minimum Linear Arrangement* (*MLA*), for which Rao and Richa [33] gave an $O(\log n)$ approximation using the notion of spreading metrics, it is subtly different: the MLA problem involves minimizing the average stretch of the edges $\sum_{\{u,v\}\in E} |\pi(u) - \pi(v)|$ under all maps $\pi\colon V \to [n]$, whereas the mappings in our problem are $f\colon V \to \Re$, and must ensure that $|f(u) - f(v)| \geq d(u, v)\ \forall\{u, v\} \in V \times V$.

The problem of finding *Minimum Latency tours* (a.k.a. the Traveling Repairman problem) is relevant to our discussion in terms of techniques used. In this problem, one is given a metric space (V, d) and a root depot $r \in V$; a repairman starting at r has to visit all $|V| = n$ customers, one at each node of the metric. The goal is to minimize the *average waiting time* of the customers, where the waiting time (or *latency*) of a customer is the sum of the distances of all edges traversed by the repairman before visiting this customer. There are extensions of this problem to the k-repairman case, where k repairmen start off at r, and the latency of a customer is now the time at which any one of the repairman visits this customer. The version with only one repairman is known to be NP-hard even on a tree [35], and is MAX-SNP hard in general [9]. The first constant-factor approximation for this problem was given by Blum et al. [9]; the approximation factor was improved by Goemans and Kleinberg [20] to 7.18, and most recently by Chaudhuri et al. [14] to 3.59. For the special cases of the latency problem on trees, Arora and Karakostas [5] gave a QPTAS; similar results were given for the case when the points lie in $\Re^d$ for fixed dimension d. The k-repairmen version of the problem was studied by Fakcharoenphol et al. [15] who show a 16.994-approximation for arbitrary k; this was improved to 8.49 in [14].

Finally, a problem that combines both the cost of a tour as well as its latency into one objective function is that of finding *time dependent TSP tours*; the paper by Blum et al. [9] gives a constant factor approximation algorithm for this problem.

2. Embedding Arbitrary Metrics into the Line

In this section we show that we can approximate the average distortion into a line for a given metric to within a constant; to this end, we show that the problem is closely related to that of finding the minimum latency tours and its generalizations in a finite metric space.

2.1. *Hardness of Embeddings*

Theorem 2.1. *It is NP-hard to find a non-contracting embedding of a given metric induced by a tree into a line that minimizes the average distortion.*

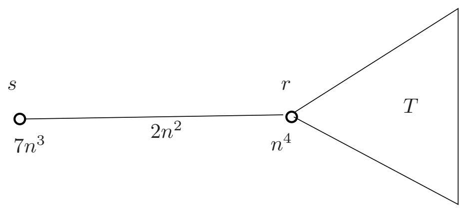

Fig. 1. Hardness construction.

Proof. We show how to reduce the problem of finding minimum latency tour on trees to our problem. The minimum latency problem on trees (tree-MLP) was shown to be NP-hard by Sitters [35] even when the edge lengths are in {0, 1}.

Given an instance of tree-MLP, our reduction will define an instance of the average distortion problem on a tree where the vertices have integer weights and the edges have lengths, and we generalize the definition of average distortion to be

$$\rho_w(f) = \frac{\sum_{x,y \in V} w_x w_y \delta(x, y)}{\sum_{x,y \in V} w_x w_y d(x, y)}. \tag{3}$$

As long as the weights are only polynomially bounded, we can convert such an instance to one with unit vertex-weights by the simple expedient of replacing any vertex with weight w by a set of w vertices at distance zero from one another. We also note that minimizing the average distortion is equivalent to minimizing the total distance in the embedding, and hence we show the hardness of minimizing the total distance.

Given a tree T rooted at r as an instance of a tree-MLP problem with edge lengths in {0, 1}, we construct an instance of the average distortion problem (see Figure 1). We introduce a new vertex s and connect it to the root r. We assign weight $7n^3$ to s and n^4 to r. Let the distance between r and s be $d_{r,s} = 2n^2$. The rest of the vertices have weight 1.

Claim 2.2. *In the optimal embedding*, r *and* s *are adjacent to each other.*

Proof. Consider any embedding in which r and s are not adjacent to each other. Therefore, the distance between r and s is at least $d(r, s) + 1$ in such an embedding. The total distance in this embedding is at least $w_r w_s \cdot (d(r, s) + 1) = 14n^9 + 7n^7$.

On the other hand, consider any embedding in which r and s are adjacent to each other and the pairwise distance between adjacent pairs is same as that in the guest tree metric. We now compute an upper bound on the total distance in such an embedding. The contribution due to the pair (r, s) is $w_r w_s \cdot d(r, s) = 14n^9$. The contribution due to the pairs of the form (r, v_i) or (s, v_i) is at most $(w_r + w_s) \cdot (2d(r, s) + n^2) \cdot n \le 6n^7$, since the distance between any two points in the embedding is at most $2d(r, s) + n^2$. Finally, the contribution from the pairs (v_i, v_j) is at most $n^2 \cdot (2d(r, s) + n^2) \le 5n^4$. Thus the total contribution is at most $12n^8 + 6n^7 + 5n^4$.

Therefore, any embedding in which r and s are adjacent is better than any embedding in which they are not. Therefore, in any optimal embedding, r and s have to be adjacent to each other. □

Claim 2.3. *In any optimal embedding, no vertex v_i and the vertex s are on the same side of r.*

Proof. Suppose that the vertex v_i and s are on the same side of r. From the previous claim it follows that s must be between v_i and r. Therefore the pair (v_i, r) contribute at least $w_r d(r, s)$ to the total distance. Now we construct an alternative embedding the current one. We keep the order of all the vertices except v_i the same. We embed v_i on the opposite side of r at the end. In this process only the contributions from the pairs (v_i, v_j) for all j and (v_j, s) go up, while the contribution from the pair (v_i, r) goes down. Note that, in the new embedding, the contribution of the pairs (v_i, v_j) can be at most $(2d(r, s) + n^2) \cdot n$ and the contribution of the pair (v_i, s) is at most $w_s(2d(r, s) + n^2)$. The contribution due to the pair (v_i, v_r) goes down by at least $w_r \cdot d(r, s) - w_r \cdot n^2$. Adding up the changes in contributions, we get that the new embedding has a smaller total distance.

Therefore in any optimal embedding, the vertices v_i and s cannot be on the same side of r. □

In order to finish the proof of the theorem, we now show that the ordering of the vertices in an optimal tree-MLP tour is $r, v_1, v_2, \ldots, v_n$ if and only if $s, r, v_1, v_2, \ldots, v_n$ is the ordering of the vertices in the embedding that minimizes the average distortion. Let $s, r, \pi(1), \pi(2), \ldots, \pi(n)$ be the ordering of the vertices in an embedding. Let $L(\pi)$ denote the total latency of the ordering given by $r, \pi(1), \ldots, \pi(n)$. Let $Av(\pi)$ denote the sum of the distances in the embedding consisting of $\pi(1), \ldots, \pi(n)$ in that order.

Then the total distance in the embedding is

$$w_s \cdot w_r + w_s \cdot n + (w_s + w_r) \cdot L(\pi) + Av(\pi).$$

Note that $Av(\pi)$ is bounded above by n^4 since we sum the distances over $\binom{n}{2}$ pairs and the maximum distance between any pair $\{v_i, v_j\}$ in the embedding is at most n^2. Thus, $Av(\pi)$ is smaller than $(w_r + w_s)$.

Note that the difference between optimal value of $L(\pi)$ and that in any other solution is at least 1, while it's multiplying factor $(w_r + w_s)$ dominates $Av(\pi)$. Hence, in order to minimize the total distance, we have to minimize $L(\pi)$. This is exactly the tree-MLP problem. Hence, the problem of minimizing the average distortion is NP-hard. □

2.2. *A Constant-Factor Approximation Algorithm*

In order to make the exposition of our approximation algorithm simple, we first show a simple 2-approximation for embedding a given metric into trees. Then we consider embeddings into k-spiders and show how a similar technique works for them (a k-spider is a tree with all vertices except the *center* having degrees 1 or 2, and hence is a homeomorph of the star with k leaves). In particular, we show how to take a ρ-approximation algorithm for the k-repairmen problem [15], and use it to produce a 2ρ-approximation for the average distortion of embedding a given metric into a k-spider. Finally, since a line metric is equivalent to a 2-spider, we get the embedding into a line metric as a corollary.

Embeddings into Trees. Consider the problem of embedding the given metric d into a tree metric δ to minimize average distortion. Let $\Delta = \sum_{x,y \in V} d(x, y)$ denote the sum of all the distances in the metric d, and hence $\mathrm{av}(d) = \Delta/n^2$ is the average distance in d. The *median* of the metric d is the point $v \in V$ that minimizes $\Delta_v = \sum_{w \in V} d(v, w)$, and will be denoted by med. Note that we can decompose Δ as follows:

$$\Delta = \sum_{u,v \in V} d(u, v) = \sum_{u \in V} \left(\sum_{v \in V} d(u, v) \right) = \sum_{u \in V} \Delta_u \geq n\, \Delta_{\mathrm{med}} \tag{4}$$

since $\Delta_{\mathrm{med}} \leq \Delta_v$ for all $v \in V$. Consider a shortest-path tree T (which is a star in a general metric d) rooted at med, and let d_T denote the metric induced by this shortest path tree. Then the total distance in this tree T is

$$\begin{aligned} \Delta_T = n^2 \cdot \mathrm{av}(d_T) &= \sum_{u,v \in V} d_T(u, v) \leq \sum_{u,v \in V} d_T(\mathrm{med}, u) + d_T(\mathrm{med}, v), \\ &= \sum_{u,v \in V} d(\mathrm{med}, u) + d(\mathrm{med}, v), = 2n\Delta_{\mathrm{med}}, \end{aligned}$$

where the inequality in the second step is just the triangle inequality. This implies that $n\Delta_{\mathrm{med}} \leq \Delta \leq \Delta_T \leq 2n\Delta_{\mathrm{med}}$, and thus:

Lemma 2.4 (see also [36]). *Given any graph, the total distance* Δ_T *for the shortest path tree rooted at the median is at most* $2\,\Delta$, *and is a* 2*-approximation for the problem of embedding the graph into trees.*

The bound of 2 is tight. For example, in a complete graph the total distance is $n(n-1)$ and it is $n(2n-3)$ for the shortest path tree. Also note here that the bound of 2 above is an *absolute* bound on the worst-case ratio between the average distance in the output tree and the graph, and is in the same flavor as the more traditional results on bounding the maximum distortion of embeddings. We next move toward an approximation approach by restricting the class of trees into which we embed.

Embeddings into Spiders. We now generalize the previous result to the case of embeddings into k-spiders. The vertex of degree k is called the *center* of the spider, and the components obtained by removing the center are called its *legs* [24].

Let d_k^* denote the optimal k-spider embedding. We decompose the sum of distances in d_k^* as the sum of k-repairman path rooted at each vertex. Recall that, in the k-traveling repairman problem, we are given k repairmen starting at a common depot s. The k repairmen are to visit n customers sitting one per node of the input metric space. The goal is to find tours on which to send the repairmen to minimize the total time customers have to wait for a repairman to arrive [15].

Let c be the center of the spider in the optimal k-spider embedding. To construct a k-repairman paths starting from a vertex r, we do the following. We send one repairman away from the center along the leg of the spider which contains r. The other $k-1$ repairmen travel toward the center c of the spider. From the center, they go off, one per remaining leg of the spider. The cost of this k-repairman tour is $\Delta_r^* = \sum_j d_k^*(r, j)$.

Summing over all choices of the root we see that this is same as the sum of distances in the embedding d_k^*:

$$n^2 \cdot \mathrm{av}(d_k^*) = \sum_{u,v \in V} d_k^*(u, v) = \sum_{v \in V} \Delta_v^*.$$

Hence, n times the cost of the cheapest k-repairman tour over all choices of the depots (denoted by Δ^{opt}), is a lower bound on the sum of all the distances, i.e.,

$$\sum_{u,v \in V} d_k^*(u, v) \geq n \cdot \min_r \{\Delta_r^{\mathrm{opt}}\}.$$

Consider the cheapest k-repairman tour over all choices of centers. Let it be centered at a vertex c. This tour defines a non-contracting embedding into a k-spider with c at the center of the spider. Let $d^c(u)$ denote the distance of vertex u from the center c in the tour. We can bound the sum of distances in this embedding as follows:

$$\sum_{u,v \in V} d_k^c(u, v) \leq \sum_{u,v \in V} d^c(u) + d^c(v) \leq 2n \sum_{u \in V} d^c(u) \leq 2 \sum_{u,v \in V} d_k^*(u, v).$$

Thus, if we could compute the optimal k-repairman tour centered at c exactly, we would obtain a 2-approximation to the problem of embedding the metric into k-spiders. Although the problem of finding an optimal k-repairman tour is NP-hard, the argument above proves the following.

Theorem 2.5. *Given a γ-approximation for the minimum k-repairmen problem on a metric d, we can obtain a 2γ-approximation for embedding the metric d into a k-spider in a non-contracting fashion to minimize the average distortion.*

The current best known approximation factor for the k-repairman problem is 8.49 (due to Chaudhuri et al. [14]), leading to the following corollary.

Corollary 2.6. *There is a 16.98-approximation for minimizing the average distortion of a non-contracting embedding of a given finite metric into a k-spider.*

3. Approximation Schemes for Trees

In this section we restrict our attention to the special case of tree metrics. We give a QPTAS for minimizing the average distortion for embeddings into the line metric. Our algorithm is based on the QPTAS given by Arora and Karakostas [5] for the minimum latency problem. They proved that a near-optimal latency tour can be constructed by concatenating $O(\log|V|/\varepsilon)$ optimal traveling salesman paths, and the best such solution can be found by dynamic programming.

For an embedding $f\colon V \to \Re$ into the line, let the *span* of the embedding be defined as $\max_{x,y} |f(x) - f(y)|$, the maximum distance between two points on the line. We note that an embedding with the shortest span is just the optimal traveling salesman path. While embedding a given metric into the line metric, minimizing the span of the embedding could result in very high average distortion. However, we show that it suffices to minimize the span locally to find near optimal embedding. In particular, our solution within $(1+\varepsilon)$ of optimal minimum average distortion is to find an embedding that is the union of $O(\log|V|/\varepsilon^2)$ traveling salesman paths with a geometrically decreasing number of vertices.

In what follows, we use n to denote $|V|$, the number of vertices. For our algorithm, we assume that all the edge lengths are in the range $[1, n^2/\varepsilon]$. Indeed, if D is the diameter of the metric space and u and v are two vertices such that $d(u, v) = D$, then $\sum_{x,y\in V} d(x, y) \geq \sum_{x\in V} d(x, u) + d(x, v) \geq nD$. We can then merge all pairs of nodes with inter-node distance at most $\varepsilon D/n^2$, which affects the sum of distance by at most $\varepsilon n D$. Hence the ratio of maximum to minimum non-zero distance in the metric can be assumed to be n^2/ε.

Relation to TDTSPs. We first show that the Arora–Karakostas QPTAS works also for the case of the Time Dependent Traveling Salesman Problem (TDTSP) defined by Blum et al. [9]. In the TDTSP the objective is to minimize a positive linear combination of the TSP tour value and the total latency of the tour. The objective function is of the form $\alpha\,\mathsf{TSP} + \beta\,\mathsf{LAT}$, where TSP and LAT denote the span of the tour and total latency of the tour, respectively, and α and β are constants.

We now describe how to break up an optimal tour into locally optimal segments. Let $\mathcal{T}$ denote the optimal tour for the objective function $\alpha\,\mathsf{TSP} + \beta\,\mathsf{LAT}$. We break this tour into k segments (k is $O(\log n/\varepsilon)$). In segment i we visit n_i nodes, where

$$n_i = \lceil (1+\varepsilon)^{k-1-i} \rceil \qquad \text{for} \quad i = 1, \dots, k-1; \qquad n_k = \lceil 1/\varepsilon \rceil.$$

Note that these n_i's are chosen in such a way that $n_i \leq \varepsilon \sum_{j>i} n_j$. Denote $\sum_{j>i} n_j$ by r_i. Replace the optimal tour in each segment, except the last one, by the minimum-distance traveling salesman path on the vertices of that segment that starts and ends at the same pair of vertices.. The new tour now consists of the concatenation of $O(\log n/\varepsilon)$ locally optimal traveling salesman paths. This gives us the following lemma.

Lemma 3.1. *There is a tour that is a concatenation of* $O(\log n/\varepsilon)$ *minimum traveling salesman paths that has* $\alpha\,\mathsf{TSP} + \beta\,\mathsf{LAT}$ *objective value at most* $(1+\varepsilon)$ *times the optimal solution* (*OPT*).

Proof. We first give a lower bound on OPT. Let T_i denote the span of the segment i in OPT. Every node in the mth segment has latency bigger than $\sum_{j=1}^{m-1} T_j$. We sum over all vertices and get the lower bound on OPT: OPT $\geq \sum_{i=1}^{k-1} (\alpha + \beta r_i) T_i$.

Now we replace each segment of OPT with the minimum traveling salesman path on the same set of vertices with the same pair of vertices as start and endpoints. By replacing a segment with a minimum traveling salesman path, we reduce the span of that segment. However, latency of the vertices inside a segment can go up. The latency of each vertex in ith segment will increase by at most $n_i T_i$. Hence the cost of the concatenated tour increases by at most $\sum_{i=1}^{k-1} \beta n_i T_i$. From the property that $n_i \leq \varepsilon \cdot r_i$, it immediately follows that the cost of the concatenated tour is at most $(1+\varepsilon)$OPT. □

We now use Lemma 3.1 to show the following theorem for average distance.

Theorem 3.2. *Any finite metric has a non-contracting embedding into a line that is composed of $O(\log n/\varepsilon^2)$ minimum traveling salesman path segments with average distortion no more than $(1+\varepsilon)$ times the minimum possible over all such embeddings.*

Proof. Our strategy is same as in Lemma 3.1. Consider the optimal embedding of the input tree into a line. We break this embedding up into $O(\log n/\varepsilon)$ segments. Let n_i be the size of the ith segment defined as before. We now divide the objective function value according to the segments, so that only the share C_i of segment i changes, if we replace the embedding of segment i with a different embedding.

Let T_i be the span of the embedding of segment i. If i_0 is the leftmost node in the embedding of segment i, then let $L_i = \sum_{j\in n_i} \delta(i_0, j)$ be the sum of the distances of all nodes in segment i from node i_0. Note that L_i is the total latency of vertices in segment i with i_0 as the root. Let $D_i = \sum_{u,v\in n_i} \delta(u, v)$ be the sum of all the pairwise distances in segment i.

Let $q_i = \sum_{j<i} n_j$ and $r_i = \sum_{j>i} n_j$ be the number of total nodes to the left and right of segment i, respectively.

We now describe a lower bound on the total distance of the optimal solution. We define the contribution of segment i to the lower bound as the sum of the following distinct terms:

1. If a vertex u is to the left of segment i and a vertex v is to the right, then segment i adds T_i to the distance between them.
2. If a vertex u is to the left and w is in segment i, then the contribution is $\delta(i_0, w) =$ the distance from the leftmost vertex i_0 of segment i to w.
3. If a vertex v is to the right and w is in segment i, then the contribution is $T_i - \delta(i_0, w)$.
4. If both vertices w and w' are in segment i, then the contribution is $\delta(w, w')$.

These contributions, when summed up over all pairs of vertices, give

$$C_i = q_i r_i T_i + q_i L_i + r_i(n_i T_i - L_i) + D_i. \tag{5}$$

Note that $\sum_i C_i$ is a lower bound on the total distance. In the following argument we rearrange the embedding inside each component while making sure that the increase in the total distance is at most $\varepsilon \sum_i C_i$.

Note that $D_i \le n_i^2 T_i$. For $i = 2, \ldots, k$, we know that $n_i \le q_i$ and $n_i \le \varepsilon \cdot r_i$. Hence, comparing D_i with the first term in (5), we get

$$(1+\varepsilon)(q_i r_i T_i + q_i L_i + r_i(n_i T_i - L_i)) \ge C_i \ge q_i r_i T_i + q_i L_i + r_i(n_i T_i - L_i). \tag{6}$$

To prove Theorem 3.2, it suffices to find an embedding of the ith segment such that the increase in the total distance is within ε times the lower bound in the right-hand side of the above inequality (6). The expression for the lower bound on the right-hand side of inequality (6) is a linear combination of TSP and latency values of the tour in segment i. We can apply Lemma 3.1 to obtain a tour composed of $O(\log n_i/\varepsilon)$ minimum traveling salesman paths. Note that replacing the original embedding with the tour obtained from

Lemma 3.1 can only increase the four distinct terms that make up the quantity C_i. From Lemma 3.1, the increase in the total distance is at most εC_i.

A technical detail in this argument is that the coefficient of L_i could be negative. Lemma 3.1 does not handle this case, but note that $n_i T_i - L_i$ is the total "reverse" latency in segment i with the rightmost endpoint being the root. Thus we can rewrite the lower bound as a linear combination of T_i and $n_i T_i - L_i$ with positive coefficients.

We can thus replace each segment i with a concatenation of $O(\log n_i/\varepsilon)$ traveling salesman paths, without increasing the cost by more than a factor of $(1+\varepsilon)$. Since there are $O(\log n/\varepsilon)$ segments in all, it follows that there is an embedding consisting of $O(\log^2 n/\varepsilon^2)$ shortest traveling salesman paths.

Finally, we show how to reduce this number to $O(\log n/\varepsilon^2)$. We rewrite the lower bound in (6) as $(q_i - r_i)L_i + (q_i + n_i)r_i T_i$. Note that $L_i \leq n_i T_i$. This gives us that the term $(q_i - r_i)L_i$ is at most $\varepsilon \cdot (q_i + n_i)r_i T_i$, whenever $q_i - r_i$ is positive. Hence, if we replace segment i with a shortest traveling salesman path on those vertices, the cost will be within $(1+\varepsilon)$ of the lower bound in (6). Note that, for $i \geq 1/\varepsilon$, we have $q_i \geq r_i$. Hence for $i = 1, \ldots, 1/\varepsilon$, using Lemma 3.1, we replace each segment by a concatenation of $O(\log n/\varepsilon)$ tours each. Then for segments $1/\varepsilon$ and above, we use only one minimum traveling salesman path per segment. Overall this results in a concatenation of $O(\log n/\varepsilon^2)$ traveling salesman paths with the average distortion within $(1+\varepsilon)$ times that of the optimal. □

Consider a $(\frac{1}{3}, \frac{2}{3})$-partition of the tree, i.e., a recursive partition of the tree into two subtrees with a common root, such that for each subtree

$$\frac{1}{3} \cdot n \leq (\text{size of subtree}) \leq \frac{2}{3} \cdot n.$$

It is a folklore result that a $(\frac{1}{3}, \frac{2}{3})$-partition exists for any tree. We use the term *separator node* for the common root of the subtrees. From the recursive partition, we get separator nodes for each level of recursion.

Note that an optimal traveling salesman path on a tree is obtained by depth-first search. Therefore, it needs to cross any separator node at most twice. In the previous theorem, we proved that a near-optimal non-contracting embedding is given by a concatenation of $O(\log n/\varepsilon^2)$ traveling salesman paths. Combining this with the recursive partition, we get the following theorem.

Theorem 3.3. *There exists a non-contracting embedding of a tree metric into a line with average distortion at most* $(1+\varepsilon)$ *times the minimum possible that, when viewed as a walk, crosses each separator node* $O(\log n/\varepsilon^2)$ *times in a recursive node-separator-based partition defined above.*

Using this theorem, we give a dynamic programming algorithm. This is very similar to the algorithm due to Arora and Karakostas [5].

Theorem 3.4. *For any given* $\varepsilon > 0$, *there is an algorithm that runs in time* $n^{O(\log n/\varepsilon^2)}$ *and computes a non-contracting embedding of a given input tree metric into a line with average distortion at most* $(1+\varepsilon)$*-times the minimum.*

Proof. We describe the dynamic program at the heart of our QPTAS.

ALGORITHM

"Guess" the leftmost vertex in the embedding. Find a recursive $(\frac{1}{3}, \frac{2}{3})$-partition of the tree. Do the following steps starting at the bottom level of the partition and working upwards:

1. Identify a separator node at the current level of the partition.
2. "Guess" the number of times the embedding crosses this node and for each crossing, the length of the embedding after the crossing and the number of nodes on that portion.
3. Search the dynamic programming table for subtours consistent with the "guesses."
4. Combine the subtours found to create a new bigger subtour and store it in the dynamic programming table and go to step 1.

"Guessing" in step 2 refers to exhaustive enumeration of all possible values for the triple (# of crossings, length, # of nodes). At the end of the enumeration, the algorithm will have created a collection of candidate solutions, one for each possible guess. Its output will be the embedding of minimum average distortion. One of the embeddings considered by this algorithm must be near-optimal. Hence the embedding produced by the algorithm is a $(1+\varepsilon)$ approximation for the optimal average distortion.

We now prove that the running time of the algorithm is bounded by $n^{O(\log n/\varepsilon^2)}$. The running time is dominated by the number of "guesses." The number of crossings through a node is at most $O(\log n/\varepsilon^2)$ and the number of nodes visited between two crossings cannot be greater than n. To bound the number of guesses for the length of the embedding between two crossings, we round the lengths as follows. Let L be the length of the longest path in the input tree. We merge all the pairs of vertices with pairwise distance smaller than $\varepsilon L/n^3$. We also round each edge length to its closest multiple of $\varepsilon L/n^3$ and divide all the lengths by $\varepsilon L/n^3$. In this rounded instance, the minimum length is 1, while the maximum internode distance is n^3/ε. After solving the rounded instance, we reinstate the merged edges to the output embedding. This does not change the pairwise distance between any pair by more than $O(\varepsilon L/n^2)$. Thus the total change due to rounding is bounded by $O(\varepsilon L) = O(\varepsilon \mathrm{OPT})$.

If we run the algorithm on a rounded instance, the total number of guesses for each crossing is $O(n^3/\varepsilon)\cdot n = O(n^4/\varepsilon)$. This gives a total of $O(\log n/\varepsilon^2 \cdot (n^4/\varepsilon)^{O(\log n/\varepsilon^2)}) = n^{O(\log n/\varepsilon^2)}$ guesses for a node. We do this for each node. Moreover, there are n choices for the leftmost vertex of the embedding. Therefore, the overall running time of the algorithm is bounded by $O(n \cdot n \cdot n^{O(\log n/\varepsilon^2)}) = n^{O(\log n/\varepsilon^2)}$. □

4. An Exact Algorithm for Minimizing Average Tree-Edge Distortion

For the tree metrics, we consider a slightly different objective function in this section. Let $M = (V, d)$ be the input metric to be embedded in a non-contracting mapping to a line. Assume that the input metric M arises from a tree $T = (V, E)$. Instead of considering distances between all pairs of nodes, we take the average of the distance over the edge set E of the tree. Let l denote the host metric (i.e., a line). Then we want to minimize $\sum_{(u,v)\in E} \delta(u, v)$. We call this the *average tree-edge distortion*. We give a polynomial time algorithm for minimizing the average tree-edge distortion.

This problem is quite similar to the MLA [34] problem on trees. Recall that, a linear arrangement of a graph (V, E) is a mapping $\pi\colon V \to [n]$. The objective is to minimize $\sum_{\{u,v\}\in E} |\pi(u)-\pi(v)|$. However, the crucial difference is that we require the embedding into a line to be non-contracting.

Our algorithm is based on the algorithm for MLA on trees given by Shiloach [34] with some crucial extensions. We first begin by finding a centroid of the tree. The following lemma is folklore (see, e.g., [12]). It is important to note that we allow subdivision of the edges here, i.e., we allow splitting an edge into two by adding a vertex anywhere along that edge.

Lemma 4.1. *Given a tree $T = (V, E)$ with edge weights, there exists a centroid vertex v^* in a subdivision of T, such that the subtrees of T rooted at v^* have edge weight at most half the total weight of the tree.*

We then show that the subtrees of the centroid are not interleaved in an optimal embedding. This lets us solve the problem recursively on the subtrees. The algorithm constructs an Eulerian tour of the tree as an optimal embedding.

4.1. *Cost Reducing Transformations*

We now show that, given a non-contracting embedding of a tree into the line, we can transform it *without increasing the average distortion*, so that the solutions for subtrees rooted at the centroid are disjoint contiguous segments of the line. We denote the embedding by a permutation π of the vertices. Note that for the embedding to be non-contracting, it suffices to have the distance between adjacent pair of vertices in the permutation the same as their distance in the tree metric (i.e. $\delta(i, i+1) = d(\pi^{-1}(i), \pi^{-1}(i+1))$).

We now explain the transformations. Let T be the input tree with v^* as the centroid. Let T_1 be a subtree of T rooted at v^*. We group all other subtrees as T_2 (see Figure 2). The transformations work toward uninterleaving the embeddings of T_1 and T_2. There are two different cases depending on whether end vertices are from the same or different subtrees.

1. Let the two endpoints be in different subtrees, i.e., we have $\pi^{-1}(1) \in T_1$ and $\pi^{-1}(n) \in T_2$. A transformation of type A (see Figure 2) converts the ordering π into π_a, such that π restricted to each of T_1 and T_2 is preserved, and T_1 is embedded entirely to the left of T_2; i.e., $\pi_a(u_i) < \pi_a(v_j)$ for all $u_i \in T_1$ and $v_j \in T_2$.

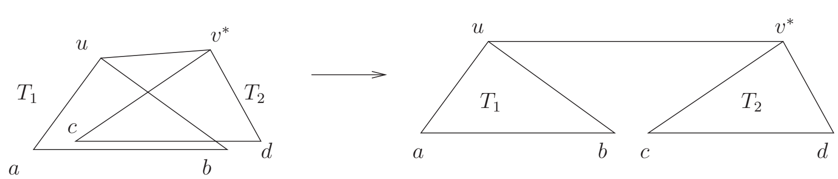

Fig. 2. Type A transformation.

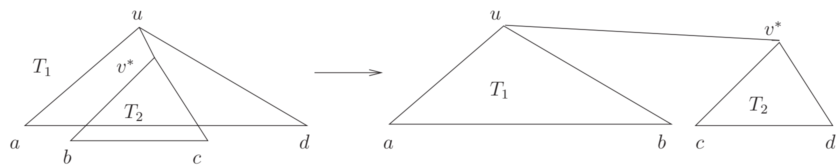

Fig. 3. Type B transformation.

2. Let the two endpoints of the embedding be in the same tree, i.e. $\pi^{-1}(1), \pi^{-1}(n) \in T_1$. A type B transformation (see Figure 3) produces an ordering π_b which is same as π when restricted to each of T_1 and T_2. We have two choices: T_1 or T_2 could be embedded to the left of the other subtree. We pick the one minimizing average tree-edge distortion.

We denote the embedding produced by π_a or π_b by $(T_1 : T_2)$. Note that there are two choices for the embedding of T_1 (resp. T_2): the same order as in π or completely opposite to π. We always pick the best of these choices.

Observation 4.2. *In the embeddings π_a and π_b, the length of the edges within the trees T_1 and T_2 is never more than their counterparts in the embedding π.*

Lemma 4.3. *The above two transformations do not increase the average tree-edge distortion of the embedding.*

Proof. We handle the two cases separately. We need the property that v^* is a centroid vertex only in the second case.

Type A. The only edge that possibly gets expanded in this transformation is (v^*, u). We show that the increase for this edge is offset by the savings in the edges of the trees T_1 and T_2. In particular, if $\pi(u_i) > \pi(v^*)$ for $i = 1, \dots, k$, then in π_a the vertices $u_1, \dots, u_k$ contribute to the cost of edge (v^*, u). However, in the initial ordering π, these vertices contribute at least this amount to the edges on the path $v^* \to \pi^{-1}(n)$. A symmetric argument holds for the change in the sum of edge lengths in the tree T_1.

Type B. Let $|T|$ denote the length of an Euler tour of tree T. We first compute the length of the edge (u, v^*) in π_b. Since we have picked the cheaper of the two available choices, the length is at most $(|T_1| + |T_2|)/2 + d(u, v^*)$. Thus the potential increase in the length of the edge (u, v^*) is $(|T_1| + |T_2|)/2$. The decrease in the sum of edge lengths of the subtree T_1 due to the transformation is at least $|T_2| + d(u, v^*)$. To see this, consider a path $\pi^{-1}(1) \to \pi^{-1}(n)$ in tree T_1. The embedding includes at least an Euler tour of tree T_2 along with the edge (u, v^*). Now if $|T_2| + 2d(u, v^*) \geq (|T_1| + |T_2|)/2$, then the decrease offsets the potential increase. In other words, if $|T_1| - |T_2| \leq 2d(u, v^*)$, then the type B transformation does not increase cost. This is certainly true since v^* is a centroid. □

4.2. *Optimal Embeddings Are Euler Tours*

Given any embedding π we can apply the transformations A or B to uninterleave the embeddings of the subtrees. Let v^* be the centroid. Let $T_0, T_1, \ldots, T_k$ be the subtrees rooted at v^*. Let $|T_i|$ denote the length of an Euler tour of tree T_i. Let the subtrees be arranged in decreasing order of the lengths of their Euler tours: $|T_0| \geq |T_1| \geq \cdots \geq |T_k|$. Let $\overline{T_0} = T - T_0$.

First we check if the embedding $(T_0 : \overline{T_0})$ has average tree-edge distortion at most that of π. If so, then we solve the problem recursively on T_0 and $\overline{T_0}$.

The other case is when $(T_0 : \overline{T_0})$ has average tree-edge distortion greater than π. From Lemma 4.3 we know that neither $\pi^{-1}(1)$ nor $\pi^{-1}(n)$ belongs to T_0. Let $\pi^{-1}(1) \in T_{i_1}$, then we can apply transformation A or B to π (depending on whether $\pi^{-1}(n) \in T_{i_1}$) and we get the embedding $(T_{i_1} : \overline{T_{i_1}})$. Let the leftmost endpoint of $\overline{T_{i_1}}$ belong to the subtree T_{i_2}. Once again we apply the appropriate transformation and get the embedding $(T_{i_1} : (T_{i_2} : T'))$, where $T' = T - T_{i_1} - T_{i_2}$. We continue this process until both endpoints of $T' = T - T_{i_1} - \cdots - T_{i_j}$ belong to T_0. At this step, the candidate transformation is B. However, it does not decrease cost at this point because v^* is no longer a centroid in T'. Hence we must adopt a different line of attack in this case. Let p be the greatest integer such that for all $i \leq p$, we have

$$2|T_i| \geq (|T_0|) + 2d(e_0) + (|T'|), \tag{7}$$

where $T' = T - T_0 - T_1 - \cdots - T_p$ and e_0 is the edge from v^* to the root of T_0. Then we can show that the embedding $(T_1 : T_2 : \cdots : T_p : \overline{T})$, where $\overline{T} = T - T_1 - \cdots - T_p$ has tree-edge distortion smaller than π. Moreover, since neither $\pi^{-1}(1)$ nor $\pi^{-1}(n)$ belongs to T_0, we have $p > 0$.

Thus we have shown that we can solve the problem recursively on these trees and combine their solutions. From this we get the following important observation.

Lemma 4.4. *An optimal non-contracting embedding of a weighted tree T into a line to minimize average tree-edge distortion corresponds to an Eulerian tour.*

4.3. *Algorithm*

We describe our recursive algorithm here. Let T be the tree from which the input metric (V, d) arises.

1. Find the centroid v^* of tree T. Let $T_0, \ldots, T_k$ be the subtrees of T rooted at v^*.
2. Find the greatest integer p such that for all $i \leq p$, we have $2|T_i| \geq (|T_0|) + 2d(e_0) + (|T'|)$, where $T' = T - T_0 - T_1 - \cdots - T_p$, and $|T_0| \geq |T_1| \geq |T_2| \geq \cdots$.
3. If $p = 0$, then recursively find the embeddings of T_0 and $\overline{T_0}$. Output the embedding $(T_0 : \overline{T_0})$.
4. If $p > 0$, then recursively find the embeddings of $T_1, \ldots, T_p, T'$ (where $T' = T - T_1 - \cdots - T_p$). Output the best embedding of these subtrees using the subroutine described below.

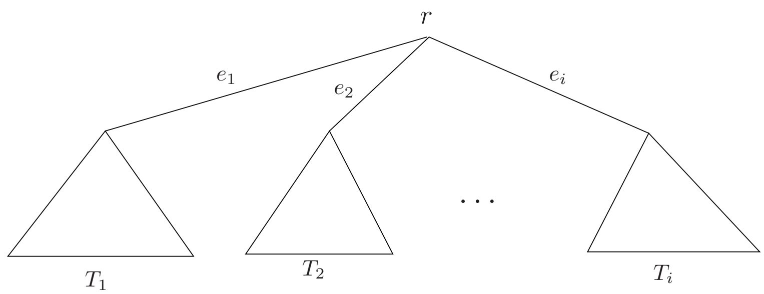

Fig. 4. Embedding the subtrees.

Subroutine. We now describe the subroutine to combine the embeddings of subtrees $T_1, \ldots, T_i$ rooted at r. We want to find the ordering of these subtrees which minimizes the tree-edge distortion of the embedding. The objective function for this subroutine is the sum of the lengths of edges $e_1, \ldots, e_i$ in the embedding. See Figures 4 and 5. Note that we only include the part of the edge from r to the closest point of its tree.

Let $d(e_j)$ be the length of edge e_j in the input metric. Since the embedding is an Eulerian tour, we know that if edge e_1 crosses trees T_2, T_3, and T_4, then it is expanded by $|T_2| + |T_3| + |T_4|$. Thus the total length of e_1 to account for is $d(e_1) + |T_2| + |T_3| + |T_4| + d'(e_1)$, where $d'(e_1)$ is the part of the length of e_1 inside tree T_1. The quantity $d'(e_1)$ can be taken as the distance of the root of T_1 to its closest endpoint. On the other hand, if there are j edges crossing over tree T_q, then the tree contributes the $|T_q|$ term in the length of each of those edges. Thus, if tree T_q is $(j+1)$st from left or right endpoint, then its contribution to the total cost is $j|T_q| + d(e_q)$.

This suggests that we can find the optimal ordering of the trees using the minimum cost matching algorithm. Consider a complete bipartite graph $K_{i,2i}$ where i is the number of subtrees hanging off the centroid. The i vertices on one side correspond to the trees $T_1, \ldots, T_i$. If tree T_q is the $(j+1)$st from the left in an embedding, this is represented by connecting vertex q on the left side of $K_{i,2i}$ to vertex $j+1$ on the other side, by an

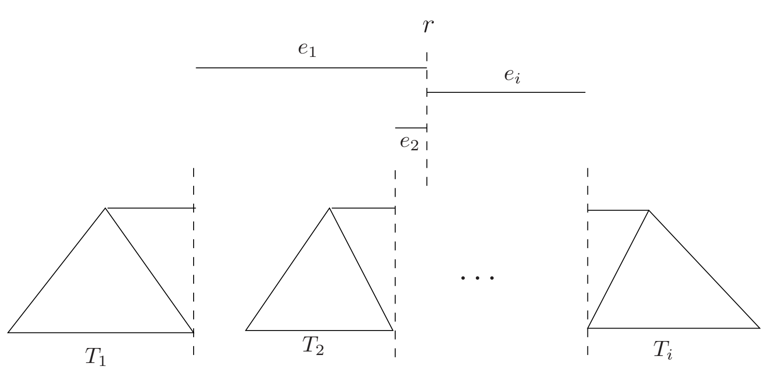

Fig. 5. Accounting for the lengths of edges.

edge of weight $2j|T_q| + d(e_q)$. If T_q is the $(j+1)$st from the right, then we connect the edge between vertex q to the vertex $i + j + 1$ on the other side of the same cost. Finding a minimum-weight matching in this bipartite graph will give us the ordering of trees on the left and right side of the root.

Theorem 4.5. *There is a polynomial-time algorithm for finding a non-contracting embedding of an input tree metric into a line to minimize average tree-edge distortion.*

We remark that it is not hard to construct instances where the optimal non-interleaving embedding in the same spirit as above provide very poor approximations to the minimum average distortion embeddings even for tree metrics. For example, consider a 3-spider where the vertices are placed at distances $l, l^2, l^3, \ldots$ on each leg. Any non-interleaving embedding has average distortion $\Omega(n)$, whereas the optimal (interleaving) embedding has average distortion $O(1)$.

5. Open Problems

For the case of non-contracting embeddings considered in the paper, some open questions are:

1. Is there a simpler and better approximation algorithm for minimizing average distortion in trees?
2. Can the QPTAS be extended to planar graphs, or even the simpler case of outer-planar graphs?
3. A different objective function is to minimize the sum of the distortions of all pairs of vertices over non-contracting embeddings. Are there approximation algorithms for this objective?

References

[1] Richa Agarwala, Vineet Bafna, Martin Farach, Mike Paterson, and Mikkel Thorup. On the approximability of numerical taxonomy (fitting distances by tree metrics). *SIAM J. Comput.*, 28(3):1073–1085 (electronic), 1999.

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

[3] Aaron Archer and David P. Williamson. Faster approximation algorithms for the minimum latency problem. In *Proceedings of the* 14*th Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 88–96, 2003.

[4] Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998.

[5] Sanjeev Arora and George Karakostas. Approximation schemes for minimum latency problems. *SIAM J. Comput.*, 32(5):1317–1337 (electronic), 2003.

[6] Yonatan Aumann and Yuval Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM J. Comput.*, 27(1):291–301, 1998.

[7] Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *Proceedings of the* 37*th Annual IEEE Symposium on Foundations of Computer Science*, pages 184–193, 1996.

[8] Yair Bartal, Avrim Blum, Carl Burch, and Andrew Tomkins. A polylog(n)-competitive algorithm for metrical task systems. In *Proceedings of the* 29*th Annual ACM Symposium on Theory of Computing*, pages 711–719, 1997.

[9] Avrim Blum, Prasad Chalasani, Don Coppersmith, Bill Pulleyblank, Prabhakar Raghavan, and Madhu Sudan. The minimum latency problem. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pages 163–171. ACM Press, New York, 1994.

[10] Avrim Blum, Goran Konjevod, R. Ravi, and Santosh Vempala. Semi-definite relaxations for minimum bandwidth and other vertex-ordering problems. *Theoret. Comput. Sci.*, 235(1):25–42, 2000. (Preliminary version in *Proc.* 30*th STOC*, pages 100–105, 1998.)

[11] Jean Bourgain. On Lipschitz embeddings of finite metric spaces in Hilbert space. *Israel J. Math.*, 52(1–2):46–52, 1985.

[12] Fred Buckley and Frank Harary. *Distance in Graphs*. Addison-Wesley, Reading, MA, 1990.

[13] Gruia Calinescu, Howard J. Karloff, and Yuval Rabani. Approximation algorithms for the 0-extension problem. In *Proceedings of the* 12*th Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 8–16, 2001.

[14] Kamalika Chaudhuri, Brighten Godfrey, Satish Rao, and Kunal Talwar. Paths, trees and minimizing latency. In *Proceedings of the* 44*th Annual IEEE Symposium on Foundations of Computer Science*, pages 36–45, 2003.

[15] Jittat Fakcharoenphol, Chris Harrelson, and Satish Rao. The k-traveling repairman problem. In *Proceedings of the* 14*th Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 655–664, 2003.

[16] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the Thirty-Fifth ACM Symposium on Theory of Computing*, pages 448–455. ACM Press, New York, 2003.

[17] Martin Farach, Sampath Kannan, and Tandy Warnow. A robust model for finding optimal evolutionary trees. *Algorithmica*, 13(1–2):155–179, 1995.

[18] Uriel Feige. Approximating the bandwidth via volume respecting embeddings. *J. Comput. System Sci.*, 60(3):510–539, 2000. (30th Annual ACM Symposium on Theory of Computing, Dallas, TX, 1998).

[19] Naveen Garg, Goran Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *J. Algorithms*, 37(1):66–84, 2000. (Preliminary version in *Proc.* 9*th SODA*, pages 253–259, 1998.)

[20] Michel Goemans and Jon Kleinberg. An improved approximation ratio for the minimum latency problem. *Math. Program.*, 82(1–2, Ser. B):111–124, 1998.

[21] Anupam Gupta. Steiner points in tree metrics don't (really) help. In *Proceedings of the Twelfth Annual ACM–SIAM Symposium on Discrete Algorithms* (*Washington*, *DC*, 2001), pages 220–227. SIAM, Philadelphia, PA, 2001.

[22] Piotr Indyk. Algorithmic aspects of geometric embeddings. In *Proceedings of the* 42*nd Annual IEEE Symposium on Foundations of Computer Science*, pages 10–33, 2001.

[23] Claire Kenyon, Yuval Rabani, and Alistair Sinclair. Low distortion maps between point sets. In *Proceedings of the* 36*th Annual ACM Symposium on Theory of Computing*, pages 272–380, 2004.

[24] Philip N. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted steiner trees. *J. Algorithms*, 19(1):104–115, 1995.

[25] Jon Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *J. ACM*, 49(5):616–639, 2002. (Preliminary version in *Proc.* 40*th FOCS*, 1999.)

[26] Eugene L. Lawler, Jan Karel Lenstra, Alexander H. G. Rinnooy Kan, and David B. Shmoys (editors). *The Traveling Salesman Problem*. Wiley, New York, 1985.

[27] F. Thomson Leighton and Satish B. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999. (Preliminary version in *Proc.* 29*th FOCS*, pages 422–431, 1988.)

[28] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. (Preliminary version in *Proc.* 35*th FOCS*, 1994.)

[29] Jiří Matoušek. Bi-Lipschitz embeddings into low dimensional Euclidean spaces. *Comment. Math. Univ. Carolin.*, 31(3):589–600, 1990.

[30] Jiří Matoušek. *Lectures on Discrete Geometry*. Volume 212 of Graduate Texts in Mathematics. Springer-Verlag, New York, 2002.

[31] Yuri Rabinovich. On average distortion of embedding metrics into the line and into ℓ_1. In *Proceedings of the Thirty-Fifth ACM Symposium on Theory of Computing*, pages 456–462. ACM Press, New York, 2003.

[32] Yuri Rabinovich and Ran Raz. Lower bounds on the distortion of embedding finite metric spaces in graphs. *Discrete Comput. Geom.*, 19(1):79–94, 1998.

[33] Satish Rao and Andrea Richa. New approximation techniques for some ordering problems. In *Proceedings of the 9th Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 211–218, 1998.

[34] Yossi Shiloach. A minimum linear arrangement algorithm for undirected trees. *SIAM J. Comput.*, 8(3):422–430, 1979.

[35] René A. Sitters. The minimum latency problem is np-hard for weighted trees. In William J. Cook and Andreas S. Schulz (editors), *Integer Programming and Combinatorial Optimization*, pages 230–239. Volume 2337 of Lecture Notes in Computer Science 2337. Springer-Verlag, Berlin, 2002.

[36] Richard T. Wong. Worst-case analysis of network design problem heuristics. *SIAM J. Algebraic Discrete Math.*, 1(1):51–63, 1980.

Received September 1, 2004, *and in final form April* 19, 2005. *Online publication November* 29, 2005.