A Constant-Factor Approximation Algorithm for Embedding Unweighted Graphs into Trees

Mihai Badoiu, Piotr Indyk and Anastasios Sidiropoulos

AI Memo 2004-015 July 2004
Abstract

We present a constant-factor approximation algorithm for computing an embedding of the shortest path metric of an unweighted graph into a tree, that minimizes the multiplicative distortion.
1 Introduction

Embedding distance matrices into geometric spaces is a fundamental problem occurring in many areas of Mathematics, and Computer Science. The applications of embeddings include data visualization, computational chemistry, and approximation algorithms (see [Wor] for discussion). The work of Shepard [She62a, She62b], Kruskal [Kru64a, Kru64b], and others, in the area of Multi-dimensional Scaling (MDS) gives the first approaches for computing such embeddings [Wor].

In this paper we present an approximation algorithm for the following embedding problem: given an unweighted graph $G = (V(G), E(G))$, compute a tree $T = (V(T), E(T))$, and a non-contracting (i.e. $D_T(f(u), f(v)) \geq D_G(u, v)$) for all $u, v \in V$) mapping $f$ of $V(G)$ into $V(T)$, such that the distortion of $f$, defined as

$$\max_{u,v \in V(G)} \frac{D_T(f(u), f(v))}{D_G(u,v)},$$

is minimized. We give a constant-factor approximation algorithm for this problem.

To our knowledge, our results provide the first non-trivial approximation guarantees for the standard (multiplicative) notion of distortion for embeddings into trees. Other results are known for the additive distortion, as described in the following section.

1.1 Related work

Combination vs Algorithmic Problem. The problem of computing low-distortion embeddings of metrics into geometric spaces has been long a subject of extensive mathematical studies. [Ind01] surveys many applications of embeddings in computer science, that have been discovered in the recent years.

The problem studied in this paper however, is inherently different from most of the embedding-related problems considered so far. More specifically, our problem is algorithmic, as opposed to combinatorial. That is, we are interested in computing efficiently the best possible distortion embedding of a given metric. This problem is algorithmic in nature, as opposed to the problem of determining the worst case embedding of a class of metrics into some host space. In fact, it is a well-known fact (see e.g. [Gup01]), that the worst case embedding of an $n$-point metric (even if it is the shortest path metric induced by an unweighted graph) into a tree, is $\Omega(n)$. Thus, the (combinatorial) problem of computing an embedding which is optimal in the worst case, is not interesting. However, the (algorithmic) problem of approximating the best possible distortion gives rise to exciting new algorithmic challenges.

Previous Work on the Algorithmic Problem. To our knowledge there have been few algorithmic embedding results. Hastad et al. gave a 2-approximation algorithm for embedding an arbitrary metric into a line $\mathbb{R}$, when the maximum additive two-sided error was considered; that is, the goal was to optimize the quantity $\max_{u,v} |f(u) - f(v)| - D(u,v)$. They also showed that the same problem cannot be approximated within 4/3 unless $P = NP$ [HIL98, Iva00]. Bádoiu extended the algorithm to the 2-dimensional plane with maximum two-sided additive error when the distances in the target plane are computed using the $l_1$ norm [B03]. Bádoiu, Indyk and Rabinovich [BIR03] gave a weakly-quasi-polynomial time algorithm for the same problem in the $l_2$ norm.

Very recently, Kenyon, Rabani and Sinclair [KRS04] gave exact algorithms for minimum (multiplicative) distortion embeddings of metrics onto simpler metrics (e.g., line metrics). Their algo-
rithms work as long as the minimum distortion is small, e.g., constant. We note that constraining
the embeddings to be onto (not into, as in our case) is crucial for the correctness of their algorithms.

In general, one can choose non-geometric metric spaces to serve as the host space. For example, in
computational biology, approximating a matrix of distances between different genetic sequences
by an ultrametric or a tree metric allows one to retrace the evolution path that led to formation of
the genetic sequences. Motivated by these applications M. Farach-Colton and S. Kannan show how
to find an ultrametric $T$ with minimum possible maximum additive distortion [FCKW93]. There
is also an approximation algorithm for the case of embedding into tree metrics, with minimum
additive distortion [ABFC+96].

2 Definitions and Preliminaries

For a graph $G = (V(G), E(G))$, let $c_w(G)$, and $c_u(G)$, be the minimum distortion of an embedding
of $G$ into a weighted, and unweighted tree, respectively. For a node $v \in V(G)$, and an integer $t \geq 0$,
we denote by $B_G(v, t)$ the set of nodes in $G$, which are at distance at most $t$ from $v$.

Lemma 1. For any unweighted graph $G$, we have $c_u(G) \leq 16c_w(G)$.

Proof. Consider an optimal embedding $f$ of $G$, into a weighted tree $T$, with distortion $c = c_w(G)$. Using
Gupta’s algorithm [Gup01], we can compute an embedding $f'$, into a weighted tree $T'$, without steiner nodes, and such that the distortion of $f'$ is at most $8c$.

By scaling the weights of $T'$, we can assume that $f'$ is non-contracting. Since $G$ is unweighted,
it follows that the weight of each edge of $T'$ is at least 1. We can construct an unweighted tree $T''$,
by replacing each edge of $T'$ of weight $k$, by a path of length $[k]$. Since $k \geq 1$, the distortion of $T''$ is at most $16c$.

3 The Algorithm

Let $G = (V(G), E(G))$ be an unweighted graph, such that $G$ can be embedded into an unweighted
tree with distortion $c$. Consider the following algorithm for embedding $G$ into an unweighted tree.

Step 1. Set $G' := G$. Pick a node $v \in V(G')$, add a node $r$ in $V(G')$, and add the edge $\{r, v\}$ in $E(G')$, of weight $c$. Set $R := \{r\}$, $K := \emptyset$, and $U := \emptyset$.

Step 2. While $R \neq \emptyset$, repeat Steps 2.1–2.2.

Step 2.1. Pick $r \in R$, and set $R := R \setminus \{r\}$. Let $K_r := B_{G'}(r, 2c - 1) \setminus U$. Set $U := U \cup K_r$, and $K := K \cup \{K_r\}$.

Step 2.2. Let $V_1, V_2, \ldots, V_t$ be the connected components of $G[V(G') \setminus U]$. For each component $V_i$, we add a node $r_i$ in $V(G')$, and we set $R := R \cup \{r_i\}$. Also, for each $v \in V_i$, with $D_{G'}(r, v) = 2c$, we add the edge $\{r_i, v\}$ to $E(G')$, of weight $c$. Finally, we set parent$(r_i) = r$.

Step 3. We construct a tree $T$ as follows. For each $K_r \in K$, we construct a star with center $r$, and leaves the nodes in $K_r \setminus \{r\}$. Next, for each $K_{r_1}, K_{r_2} \in K$, with parent$(r_1) = r_2$, we connect the stars of $K_{r_1}$ and $K_{r_2}$, by adding an edge $\{r_1, r_2\}$ in $T$. 

3
Lemma 2. Let $G$ be an unweighted 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 $v_{i+1} \mod 4$, and
- for each $i$, with $0 \leq i < 4$, $D_G(p_i, p_{i+2} \mod 4) > \lambda$,

then, $c_u(G) > \lambda$.

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

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

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

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

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

Case 1: $T_i \cap T_{i+2} \mod 4$ contains a node from $V(p_i) \cup V(p_{i+2} \mod 4)$. W.l.o.g., we assume that there exists $w \in V(p_{i+2} \mod 4)$, such that $w \in T_i \cap T_{i+2} \mod 4$. By Claim 1, it follows that there exists $\{u, v\} \in E(p_i)$, such that $f(w)$ lies on $P_{u,v}$. This implies

$$\text{D}_T(f(u), f(v)) = \text{D}_T(f(u), f(w)) + \text{D}_T(f(w), f(v)).$$

On the other hand, we have $D_G(p_i, p_{i+2} \mod 4) > \lambda$, and since $f$ is non-contracting, we obtain

$$D_T(f(u), f(v)) > 2\lambda.$$

Thus, $c \geq D_T(f(u), f(v))/D_G(u, v) > 2\lambda$.

Case 2: $T_i \cap T_{i+2} \mod 4$ does not contain nodes from $V(p_i) \cup V(p_{i+2} \mod 4)$. Let $w \in T_i \cap T_{i+2} \mod 4$. By Claim 1, there exist $\{u_1, v_1\} \in E(p_i)$, and $\{u_2, v_2\} \in E(p_{i+2} \mod 4)$, such that $w$ lies in both $P_{u_1, v_1}$, and $P_{u_2, v_2}$. We have

$$\text{D}_T(f(u_1), f(v_1)) + \text{D}_T(f(u_2), f(v_2)) = \text{D}_T(f(u_1), f(w)) + \text{D}_T(f(w), f(v_1)) + \text{D}_T(f(u_2), f(w)) + \text{D}_T(f(w), f(v_2))$$

$$= \text{D}_T(f(u_1), f(u_2)) + \text{D}_T(f(v_1), f(v_2))$$

$$\geq D_G(u_1, u_2) + D_G(v_1, v_2)$$

$$\geq 2D_G(p_i, p_{i+2} \mod 4)$$

$$> 2\lambda$$
Thus, we can assume w.l.o.g., that
\[ D_T(f(u_1), f(v_1)) > \lambda. \]

It follows that \( c \geq D_T(f(u_1), f(v_1))/D_G(u_1, v_1) > \lambda. \)

Moreover, since \( p_i \) and \( p_{i+1} \mod 4 \), share an end-point, we have
\[ T_i \cap T_{i+1} \mod 4 \neq \emptyset \]

By Claim 2, it follows, that \( \bigcup_{i=0}^{3} T_i \subseteq T \), contains a cycle, a contradiction.

**Lemma 3.** For every \( K_r \in K \), and for every \( x, y \in K_r \), we have \( D_G(x, y) \leq 8c. \)

**Proof.** Assume that the assertion is not true, and pick \( K_r \in K \), and \( x, y \in K_r \), such that \( D_G(x, y) > 8c. \) Let \( r_1 = r \), and for each \( i > 1 \), with parent\((r_i) \neq \emptyset \), let \( r_{i+1} = \text{parent}(r_i) \).

Pick a node \( x_1 \in K_r \), with \( \{r, x_1\} \in E(G') \), such that \( D_G(x_1, x) \) is minimized. Similarly, pick a node \( y_1 \in K_r \), with \( \{r, y_1\} \in E(G') \), such that \( D_G(y_1, y) \) is minimized. Inductively, pick \( x_i, y_i \), for \( i > 1 \) as follows: Pick a node \( x_i \in K_{r_i} \), with \( \{r_i, x_i\} \in E(G') \), such that \( D_G(x_i, x_{i-1}) \) is minimized. Similarly, pick a node \( y_i \in K_{r_i} \), with \( \{r_i, y_i\} \in E(G') \), such that \( D_G(y_i, y_{i-1}) \) is minimized.

Let \( p^x_i \), and \( p^y_i \), be shortest paths from \( x_i \) to \( x_{i+1} \), and from \( y_i \) to \( y_{i+1} \), respectively. Let also \( p^x \), and \( p^y \), be the paths resulting from the concatenation of the paths \( p^x_1, p^x_2, \) and \( p^y_1, p^y_2 \), respectively.

**Claim 3.** \( D_G(p^x, p^y) > 2c. \)

**Proof.** We have \( D_G(x, y) > 8c, \) \( D_G(x, x_1) < c, \) and \( D_G(y, y_1) < c, \) thus \( D_G(x_1, y_1) > 6c. \) Observe that \( D_G(x_{i+1}, x_i) = c, \) and \( D_G(y_{i+1}, y_i) = c. \) Thus
\[
D_G(p^x_i, p^y_i) \geq D_G(x_1, y_1) - 2c,
\]
and
\[
D_G(p^x, p^y) \geq D_G(x_1, y_1) - 4c > 2c. \]

Consider now the nodes \( x_3 \), and \( y_3 \), and let \( z \) be the node \( r \), picked at Step 1 of the algorithm. Let \( t_x \), be the shortest path from \( x_3 \) to \( r \), and let also \( t_y \), be the shortest path from \( y_3 \) to \( r \). It follows by the construction, that \( V(t_x) \cap K_{r_3} = \{x_3\} \), and \( V(t_y) \cap K_{r_3} = \{y_3\} \). By the choice of \( x_3 \), and \( y_3 \), and since \( t_x \), and \( t_y \), share an end-point, it follows that there exists a path \( p^{xy} \) on \( G \), with endpoints \( x_3 \), and \( y_3 \), such that \( p^{xy} \) does not visit any of the nodes of the sets \( K_{r_i} \), for \( i \leq 2 \).

Moreover, since \( x_1 \), and \( y_1 \), are both in \( K_{r_1} \), it follows that \( x \) and \( y \) are in the same connected component of \( G[V(G) \setminus \bigcup_{i \geq 2} K_{r_i}] \). In other words, there exists a path \( p^{yx} \), with endpoints \( x_1 \), and \( y_1 \), such that \( p^{yx} \) does not visit any of the nodes of the sets \( K_{r_i} \), for \( i > 1 \).

Observe that any shortest path in \( G \), from a node in \( K_{r_1} \), to a node in \( K_{r_3} \), must visit at least \( c \) nodes from \( K_{r_2} \). It follows that
\[
D_G(p^{xy}, p^{yx}) > c.
\]

We have shown that the nodes \( x_1, y_1, x_3, \) and \( y_3, \) together with the paths \( p^x, p^{xy}, p^y, \) and \( p^{yx}, \) satisfy the conditions of Lemma 2, for \( \lambda = c. \) Thus, \( c(G) > 2c, \) a contradiction.

\[ \square \]
Lemma 4. The contraction of the embedding, is at most \(4c\).

Proof. Let \(x, y \in V(G)\). We have to consider the following cases for \(x\), and \(y\):

Case 1: \(x, y \in K_r\).

We have \(D_T(x, y) = 2\), and by Lemma 3, \(D_G(x, y) < 8c\). Thus, in this case the contraction is at most \(4c\).

Case 2: There exist \(r_1, \ldots, r_k\), for some \(k > 1\), with \(x \in K_{r_1}\), and \(y \in K_{r_k}\), such that for any \(i\), with \(1 \leq i < k\), \(\text{parent}(r_i) = r_{i+1}\).

We have \(D_T(x, y) = k + 1\). By the construction, it follows that there exists a node \(y' \in K_{r_k}\), such that \(D_G(y', x) \leq kc\). Moreover, by Lemma 3, \(D_G(y', y) \leq 8c\), and thus \(D_G(x, y) \leq (k + 8)c\). Since \(k \geq 2\), the contraction is at most \((k + 8)c/(k + 1) \leq 10c/3\).

Case 3: There exist \(r_1, \ldots, r_k\), for some \(k > 1\), with \(x \in K_{r_1}\), and \(r'_1, \ldots, r'_l\), for some \(l > 1\), with \(y \in K_{r'_l}\), such that for any \(i\), with \(1 \leq i < k\), \(\text{parent}(r_i) = r_{i+1}\), and for any \(j\), with \(1 \leq j < l\), \(\text{parent}(r'_j) = r'_{j+1}\), and \(r_k = r'_l\).

We have \(D_T(x, y) = k + l\). By the construction, it follows that there exists a node \(x' \in K_{r_k}\), such that \(D_G(x', x) \leq kc\). Also, there exists a node \(y' \in K_{r_k}\), such that \(D_G(y', y) \leq lc\). By Lemma 3, \(D_G(x', y') \leq 8c\), and thus \(D_G(x, y) \leq (k + l + 8)c\). Since \(k, l \geq 2\), the contraction is at most \((k + l + 8)c/(k + l) \leq 3c\).

Lemma 5. The expansion of the embedding, is at most \(3\).

Proof. To bound the expansion of the embedding, it suffices to consider nodes \(x, y \in V(G)\), with \(\{x, y\} \in E(G)\). If \(x, y \in K_r\), for some \(K_r \in K\), then \(D_T(x, y) = 2\), in which case the expansion is at most \(2\).

Otherwise, let \(x \in K_r\), and \(y \in K_{r'}\), for some \(K_r, K_{r'} \in K\), with \(r \neq r'\). W.l.o.g., assume that \(K_r\) was created by the algorithm before \(K_{r'}\). It follows that before \(K_r\) was created, \(x\) and \(y\) where in the same connected component of \(G[V(G') \setminus U]\). Thus, after the creation of \(K_r\), the node \(r'\) is added in \(G'\), and the algorithm sets \(\text{parent}(r') = r\). Thus, \(D_T(x, y) = 3\), and the expansion is at most \(3\).

Theorem 1. There exists a polynomial time, constant-factor approximation algorithm, for the problem of embedding an unweighted graph, into a tree, with minimum multiplicative distortion.

Proof. It follows by Lemmata 1, 4, and 5.

References


