A Tight Bound on Approximating Arbitrary Metrics by Tree Metrics

Jittat Fakcharoenphol*
Computer Science Division,
University of California,
Berkeley, CA 94720.
jittat@cs.berkeley.edu

Satish Rao†
Computer Science Division,
University of California,
Berkeley, CA 94720.
satishr@cs.berkeley.edu

Kunal Talwar‡
Computer Science Division,
University of California,
Berkeley, CA 94720.
kunal@cs.berkeley.edu

ABSTRACT

In this paper, we show that any n point metric space can be embedded into a distribution over dominating tree metrics such that the expected stretch of any edge is $O(\log n)$. This improves upon the result of Bartal who gave a bound of $O(\log n \log\log n)$. Moreover, our result is existentially tight; there exist metric spaces where any tree embedding must have distortion $\Omega(\log n)$-distortion. This problem lies at the heart of numerous approximation and online algorithms including ones for group Steiner tree, metric labeling, buy-at-bulk network design and metrical task system. Our result improves the performance guarantees for all of these problems.

Categories and Subject Descriptors

G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph Algorithms*

General Terms

Algorithms, Theory

Keywords

Metrics, Embeddings, Tree metrics

1. INTRODUCTION

1.1 Metric approximations

The problem of approximating a given graph metric by a "simpler" metric has been a subject of extensive research, motivated from several different perspectives. A particularly simple metric of choice, also favored from the algorithmic point of view, is a tree metric, i.e. a metric arising from shortest path distance on a tree containing the given points. Ideally we would like that distances in the tree metric are no smaller than those in the original metric and we would like to bound the *distortion* or the maximum increase. However, there are simple graphs (e.g. the n-cycle) for which the distortion must be $\Omega(n)$ [41, 7, 25].

To circumvent this, Karp [30] considered approximating the cycle by a probability distribution over paths, and showed a simple distribution such that the expected length of each edge is no more than twice its original length. This gave a competitive ratio of 2 for the k-server problem (on a cycle) that had motivated this approach. Alon, Karp, Peleg and West [1] looked at approximating arbitrary graph metrics by (a distribution over) spanning trees, and showed an upper bound of $2^{O(\sqrt{\log n \log\log n})}$ on the distortion.

Bartal [7] formally defined probabilistic embeddings and improved on the previous result by showing how to probabilistically approximate metrics by tree metrics with distortion $O(\log^2 n)$. Unlike the result of Alon et.al. [1], Bartal's trees were not spanning trees of the original graph, and had additional Steiner points. He however showed that this probabilistic approximation leads to approximation algorithms for several problems, as well as the first polylogarithmic competitive ratios for a number of on-line problems. We should note that the trees that Bartal used have a special structure which he termed hierarchically well separated. This meant that weights on successive levels of the tree differed by a constant factor. This was important for several of his applications.

Konjevod, Ravi and Salman [34] showed how Bartal's result improves to $O(\log n)$ for planar graphs, and Charikar et.al. [17] showed similar bounds for low dimensional normed spaces. Inspired by ideas from Seymour's work on feedback arc set [45], Bartal [8] improved his earlier result to $O(\log n \log\log n)$. This of course led to improved bounds on the performance ratios of several applications. Bartal also observed that any probabilistic embedding of an expander graph into a tree has distortion at least $\Omega(\log n)$.

In this paper, we show that an arbitrary metic space can be approximated by a distribution over dominating tree metrics with distortion $O(\log n)$, thus closing the gap between the

‡Supported by NSF grants CCR-0105533 and CCR-9820897.
*Supported in part by a DPST scholarship and NSF grant CCR-0105533.
†Supported by NSF grant CCR-0105533.

lower and the upper bounds. Our result is constructive and we give a simple algorithm to sample a tree from this distribution. Our trees are also heirarchically well separated, like Bartal's. This gives improved approximation algorithms for various problems including group Steiner tree [24], metric labeling [19, 32], buy-at-bulk network design [4], and vehicle routing [16]. We give a more comprehensive list in section 3.

1.2 Related Work

Divide and conquer methods have been used to provide polylogarithmic-factor approximation algorithms for numerous graph problems since the discovery of an $O(\log n)$ approximation algorithm for finding a graph separator [36]. The algorithms proceeded by recursively dividing a problem using the above approximation algorithm, and then using the decomposition to find a solution. Typically, the approximation factor was $O(\log^2 n)$: a logarithmic factor came from the $O(\log n)$ separator approximation, another $O(\log n)$ factor came from the recursion. Using this framework, polynomial-time approximation algorithms for many problems are given in [36], for example: crossing number, VLSI layout, minimum feedback arc set, and search number.

Independently, Seymour [45] gave an $O(\log n \log\log n)$ bound on the integrality gap for a linear programming relaxation of the feedback arc set problem (for which the above techniques had given an $O(\log^2 n)$ bound). In doing so, he developed a technique that balanced the approximation factor of his separator based procedure against the cost of the recursion to significantly improve the bounds.

Even et al.[20] introduced linear programming relaxations for a number of problems and combined them with Seymour's techniques to give $O(\log n \log\log n)$-approximation algorithms for many of the problems that previously had $O(\log^2 n)$ approximation algorithms, e.g., linear arrangement, embedding a graph in d-dimensional mesh, interval graph completion, minimizing storage-time product, and (subset) feedback sets in directed graphs.

Bartal's results [8] implied $O(\log n \log\log n)$-approximations for still more problems. Moreover, he used probabilistic techniques so as to bound the expected stretch of each edge, not just the average. This led to polylogarithmic competitive ratio algorithms for a number of online problems (against oblivious adversaries) such as metrical task system [10]. Charikar et.al. [16, 17] showed how to derandomize the approximation algorithms that follow from Bartal's embeddings.

This work also follows the line of research on embeddings, with low distortion, graphs into other "nice" metric spaces, which have good structural properties, such as Euclidean and ℓ_1 spaces [37, 26, 18, 43, 23].

The work of Bourgain [14] showed that any finite metric on n nodes can be embedded into ℓ_2 with logarithmic distortion with the number of dimensions exponential in n. Linial, London, and Rabinovich [37] modified Bourgain's result to apply for ℓ_1 metrics and to use $O(\log^2 n)$ dimensions. Aumann and Rabani [3] and Linial, London and Rabinovich [37] gave several applications, including a proof of a logarithmic bound on max-flow min-cut gap for multicommodity flow problems. They also gave a lower bound on the distortion of any embeddings of general graphs into ℓ_1. For more details, we point the reader to Chapter 15 in Matousek [38].

Embeddings of special graphs have also been considered by many researchers. Gupta et al. [26] considered embeddings or series-parallel graphs and outerplanar graphs into ℓ_1 with constant distortion; Chekuri et al. [18] show a constant-distortion embedding for k-outerplanar graphs. For planar graph, Rao [43] gave an $O(\sqrt{\log n})$-distortion embedding into ℓ_2, which matched the lower bound given by Newman and Rabinovich [39].

Graph decomposition techniques for many interesting classes of graphs have also been extensively studied. For example, Klein, Plotkin, and Rao's [31] result provided a constant factor approximation for graphs that exclude fixed sized minors (which includes planar graphs). Similar results were given by Charikar et al. [17] for geometric graphs.

1.3 Our techniques

The algorithm relies on techniques from the algorithm for 0-extension given by Calinescu, Karloff and Rabani [15], and improved by Fakcharoenphol, Harrelson, Rao and Talwar [21]. The CKR procedure implies a randomized algorithm that outputs clusters of diameter about Δ such that the probability of an edge e being cut is $(d_e/\Delta)\log n$, where d_e is the length of the edge e. The analysis can in fact be improved to replace the $\log n$ by the logarithm of the ratio of number of vertices within distance Δ of e to the no. of vertices within distance $\Delta/2$; i.e. the number of times the size of a neighbourhood of e doubles between $\Delta/2$ and Δ. Our algorithm runs a CKR like procedure for diameters 2^i, $i = 0, 1, 2, \ldots$ to get a decomposition of the graph (which can then be converted to a tree). Since the total number of doublings over all these levels is bounded by $\log n$.

2. THE ALGORITHM

In this section, we outline the algorithm for probabilistically embedding an n point metric into a tree, and show that the expected distortion of any distance is $O(\log n)$. Like previous algorithms, we first decompose the graph hierarchically and then convert the resulting laminar family to a tree.

2.1 Preliminaries

We define some notation first. Let the input metric be (V, d). We shall refer to the elements of V as vertices or points. We shall refer to a pair of vertices (u, v) as an edge. Without loss of generality, the smallest distance is strictly more than 1. Let Δ denote the diameter of the metric (V, d). Without loss of generality, $\Delta = 2^\delta$.

A metric (V', d') is said to *dominate* (V, d) if for all $u, v \in V$, it is the case that $d'(u, v) \geq d(u, v)$. We shall be looking for tree metrics that dominate the given metric.

Let $\mathcal{S}$ be a family of metrics over V, and let $\mathcal{D}$ be a distribution over $\mathcal{S}$. We say that $(\mathcal{S}, \mathcal{D})$ *α-probabilistically approximates* a metric (V, d) if every metric in $\mathcal{S}$ dominates d and for every pair of vertices $(u, v) \in V$, $E_{d' \in (\mathcal{S},\mathcal{D})}[d'(u, v)] \leq \alpha \cdot d(u, v)$.

We shall be interested in α-probabilistically approximating an arbitrary metric (V, d) by a distribution over tree metrics.

For a parameter r, an *r-cut decomposition* of (V, d) is a partitioning of V into clusters, each centered around a vertex and having radius at most r. Thus each cluster will have diameter at most $2r$.

A *hierarchical cut decomposition* of (V, d) is a sequence of $\delta + 1$ nested cut decompositions $D_0, D_1, \ldots, D_\delta$ such that

- $D_\delta = \{V\}$, i.e. the trivial partition (that puts all vertices in a single cluster).
- D_i is a 2^i-cut decomposition, and a refinement of D_{i+1}.

Note that each cluster in D_0 has radius at most 1 and hence must be a singleton vertex.

2.2 Decompositions to trees

A hierarchical cut decomposition defines a laminar family[1], and naturally corresponds to a rooted tree as follows. Each set in the laminar family is a node in the tree and the children of a node corresponding to a set S are the nodes corresponding to maximal subsets of S in the family. Thus the node corresponding to V is the root and the singletons are the leaves. Also note that the children of a set in D_{i+1} are sets in D_i. (See figure 1).

We define a distance function on this tree as follows. The links from a node S in D_i to each of its children in the tree have length equal to the 2^i (which is an upper bound on the radius of S). This induces a distance function $d^T(\cdot, \cdot)$ on V where $d^T(u, v)$ is equal to the length of the shortest path distance in T from node $\{u\}$ to node $\{v\}$. Given the length function, it is easy to see that $d^T(u, v) \geq d(u, v)$ for all u and v.

We shall also like to place upper bounds on $d^T(u, v)$. We say an edge (u, v) is at level i if u and v are first separated in the decomposition D_i. Note that if (u, v) is at level i, then $d^T(u, v) = 2\sum_{j=0}^{i} 2^j \leq 2^{i+2}$.

2.3 Decomposing the graph

We shall describe a random process to define a hierarchical cut decomposition of (V, d), such that the probability that an edge (u, v) is at level i decreases geometrically with i.

We first pick a random permutation π of $\{v_1, v_2, \ldots, v_n\}$, which will be used throughout the process. We also pick a β uniformly at random in the interval $[1, 2]$. For each i, we compute D_i from D_{i+1} as follows. First set β_i to be $2^{i-1}\beta$. Let S be a cluster in D_{i+1}. We assign a vertex $u \in S$ to the first (according to π) vertex $v \in V$ closer than β_i to u. Each child cluster of S in D_i then consists of the set of vertices in S assigned to a single center v. Note that the center v itself need not be in S. Thus one center v may correspond to more than one cluster, each inside a different level $(i + 1)$ cluster (see for example, the center $\pi(8)$ in figure 2). Note that since $\beta_i \leq 2^i$, the radius of each cluster is at most 2^i and thus we indeed get a 2^i-cut decomposition. More formally,

[1] Recall that a *laminar family* $\mathcal{F} \subseteq 2^V$ is a family of subsets of V such that for any $A, B \in \mathcal{F}$, it is the case that $A \subseteq B$ or $B \subseteq A$ or $A \cap B = \phi$.

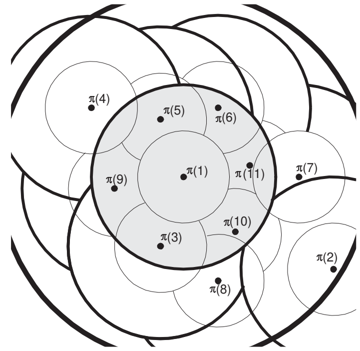

Figure 2: A possible hierarchical cut decomposition output by the algorithm. The varying thicknesses indicate cuts at different levels.

Algorithm *Partition* (V, d)
1. Choose a random permutation π of $v_1, v_2, \ldots, v_n$.
2. Choose β uniformly at random in $[1, 2]$.
3. $D_\delta \leftarrow \{V\}; i \leftarrow \delta - 1$.
4. while D_{i+1} has non-singleton clusters do
4.1 $\quad \beta_i \leftarrow 2^{i-1}\beta$.
4.2 $\quad$ For $l = 1, 2, \ldots, n$ do
4.2.1 $\quad\quad$ For every cluster S in D_{i+1}.
4.2.1.1 $\quad\quad\quad$ Create a new cluster consisting of all unassigned vertices in S closer than β_i to $\pi(l)$.
4.3 $\quad i \leftarrow i - 1$.

It is easy to see that the algorithm can be implemented in time $O(n^3)$. A more careful implementation can actually be made to run in time $O(n^2)$ (i.e. linear in the size of the input).

We now fix an arbitrary edge (u, v), and show that the expected value of $d^T(u, v)$ is bounded by $O(\log n) \cdot d(u, v)$. We shall make no attempts to optimize the constants in this analysis. From the discussion above, it follows that

$$E[d^T(u, v)] \quad \leq \quad \sum_{i=0}^{\delta} \Pr[(u, v) \text{ is at level } i] \cdot 2^{i+2} \qquad (1)$$

We shall show that the right hand side of this equation is bounded by $O(\log n) \cdot d(u, v)$.

If vertices u and v are in separate clusters in D_i, we say that D_i separates (u, v). Now note that (u, v) is at level i if

(a) D_i separates (u, v).

(b) D_j does not separate (u, v) for any $j > i$.

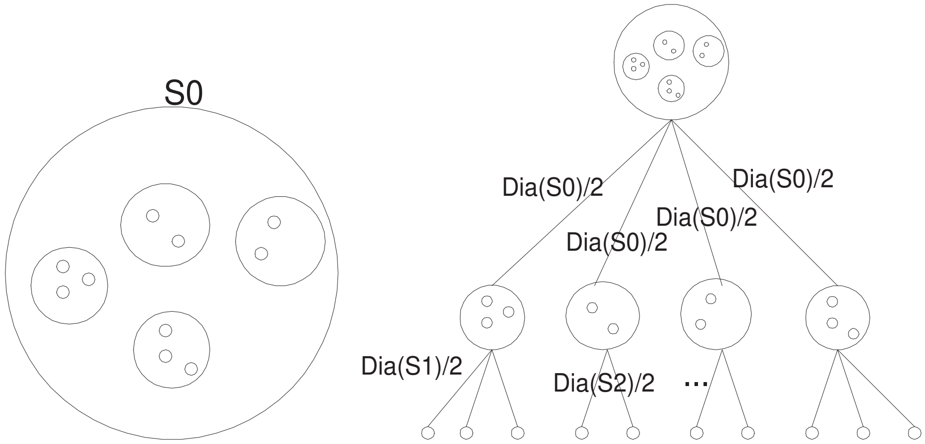

Figure 1: Converting a laminar family into a tree. Note that the values we put on the links ensure that the embedding is an expansion.

Clearly if $d(u,v) > 2^{i+2}$, then u and v cannot be in the same cluster in D_{i+1}, i.e., D_{i+1} separates (u,v). From (b) above, (u,v) cannot be at level i. Let j^* be the smallest i such that $d(u,v) \leq 2^{i+2}$. Thus $\Pr[(u,v)$ is at level $i] = 0$ for any $i < j*$. For $i \geq j^*$, we shall bound the probability that (u,v) is at level i.

From (a) and (b) above, for any $i \geq j^*$,

$$\begin{aligned}
&\Pr[(u,v) \text{ is at level } i] \\
&\quad = \Pr[D_i \text{ separates } (u,v)] \cdot \\
&\qquad \Pr[\nexists j > i : D_j \text{ separates } (u,v) \mid D_i \text{ separates } (u,v)] \\
&\quad \leq \Pr[D_i \text{ separates } (u,v)]
\end{aligned}$$

For any $j^* \leq j \leq \delta$, let K_j^u be the set of vertices in V closer than 2^j to vertex u, and let $k_j^u = |K_j^u|$. We define K_j^v and k_j^v similarly. For $j < j^*$, we let $k_j^u = 0$.[2]

Now consider the clustering step at level $i \geq j*$. In each iteration, all unassigned vertices v such that $d(v, \pi(l)) \leq \beta_i$ assign themselves to $\pi(l)$. For some initial iterations of this procedure, both u and v remain unassigned. Then at some step l, at least one of u and v gets assigned to the center $\pi(l)$. We say that center $\pi(l)$ *settles* the edge (u,v) at level i if it is the first center to which at least one of u and v get assigned. Note that exactly one center settles any edge (u,v) at any particular level. Further, we say that center $\pi(l)$ *cuts* the edge $e = (u,v)$ at level i if it settles e at this level, but exactly one of u and v is assigned to $\pi(l)$ at level i. Clearly, D_i separates (u,v) iff some center w cuts it at this level. Hence $\Pr[D_i$ separates $(u,v)] = \sum_w \Pr[w$ cuts (u,v) at level $i]$.

We say that center w *cuts* u *out* of (u,v) at level i if w cuts (u,v) at this level and u is assigned to w (and v is not assigned to w) at this level. For each center w, we shall bound the probability that w cuts u out of (u,v) at level i. Let us arrange the centers in K_i^u in increasing order of distance from u, say $w_1, w_2, \ldots, w_{k_i^u}$. For a center w_s to cut (u,v) such that only u is assigned to w_s, it must be the case that

(a) $d(u, w_s) \leq \beta_i$.

(b) $d(v, w_s) > \beta_i$.

(c) w_s settles e.

Thus β_i must lie in $[d(u,w_s), d(v,w_s)]$ (see figure 3). By triangle inequality, $d(v,w_s) \leq d(v,u) + d(u,w_s)$, and hence the interval $[d(u,w_s), d(v,w_s)]$ is of length at most $d(u,v)$. Since β_i is distributed uniformly in $[2^{i-1}, 2^i]$, the probability that β_i falls in the bad interval is at most $(d(u,v)/2^{i-1})$. Moreover for such a value of β_i, any of $w_1, w_2, \ldots, w_s$ can settle (u,v) at level i and hence the first amongst these in the permutation π will. Since π is a random permutation, the probability that w_s is the one to settle (u,v) at level i is at most $1/s$.

At this point, it is then easy to see that the probability that D_i separates (u,v) is at most

$$\begin{aligned}
&\Pr[D_i \text{ separates } (u,v)] \\
&\quad \leq \sum_{s=1}^{k_i^u} (d(u,v)/2^{i-1}) \cdot \frac{1}{s} + \sum_{s=1}^{k_i^v} (d(u,v)/2^{i-1}) \cdot \frac{1}{s} \\
&\quad \leq (d(u,v)/2^{i-1})(\ln k_i^u + \ln k_i^v).
\end{aligned}$$

Thus each i contributes at most $O(\log n)$ to the expected value of $d^T(u,v)$ (equation 1) and hence the expected length is bounded by $O(\log n \log \Delta)$.

We however promised to show an improved bound of $O(\log n)$. We shall do so by observing that the total number of centers over all δ levels is n. A more careful analysis of the above procedure will yield the result.

Let us first consider some $i \geq j^* + 4$. Since the radius of the cluster at level i is at least 2^{i-1}, centers very close to both u

[2] Though the notation does not explicitly suggest so, these k_j^u, etc. are then defined with respect to the edge (u,v).

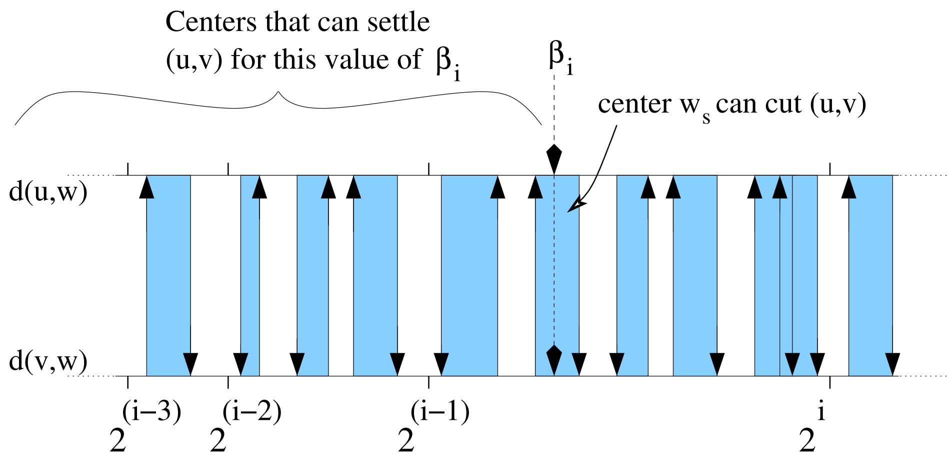

Figure 3: Bounding the probability of an edge being cut. Each shaded rectangle represents a center; arrow marks indicate distances from u and v. Width of each shaded rectangle is at most $d(u,v)$

and v can never cut the edge (u,v). More precisely, for any w in K^u_{i-2}, if u is assigned to w, it must be the case that v gets assigned to w also, because $d(v,w) \leq d(v,u)+d(u,w) \leq 2^{i-2}+2^{i-2} \leq 2^{i-1} \leq \beta_i$ (since $i \geq j^*+4$). Thus, no center in $w_1, w_2, \ldots, w_{k^u_{i-2}}$ can ever cut u out of (u,v). This implies that the probability that u gets cut out of edge e is in fact bounded by

$$\sum_{s=k^u_{(i-2)}+1}^{k^u_i} \frac{1}{s}\,(d(u,v)/2^{i-1})$$
$$= (d(u,v)/2^{i-1}) \cdot (H_{k^u_i} - H_{k^u_{(i-2)}})$$

Since (u,v) can be cut when either u or v is cut out by some vertex, the overall probability that D_i separates (u,v) is then at most $(d(u,v)/2^{i-1})\cdot[H_{k^u_i}+H_{k^v_i}-H_{k^u_{(i-2)}}-H_{k^v_{(i-2)}}]$.

For $i = j^*, j^*+1, j^*+2, j^*+3$, we just bound this probability by $(d(u,v)/2^{i-1}) \cdot (H_{k^u_i} + H_{k^v_i}) \leq (d(u,v)/2^{i-1}) \cdot 2H_n$.

The expected value of $d^T(u,v)$ is therefore.

$$\begin{aligned}
&E[d^T(u,v)]\\
&\leq \sum_{i=0}^{\delta} \Pr[(u,v) \text{ is at level } i] \cdot 2^{i+2}\\
&\leq \sum_{i=j^*}^{\delta} \Pr[D_i \text{ separates } (u,v)] \cdot 2^{i+2}\\
&\leq \sum_{i=j^*}^{j^*+3} \cdot 2H_n \cdot \frac{d(u,v)}{2^{i-1}} \cdot 2^{i+2} +\\
&\qquad \sum_{i=j^*+4}^{\delta} (H_{k^u_i} + H_{k^v_i} - H_{k^u_{(i-2)}} - H_{k^v_{(i-2)}}) \cdot \frac{d(u,v)}{2^{i-1}} \cdot 2^{i+2}\\
&\leq 8d(u,v)(4 \cdot 2H_n + H_{k^u_\delta} + H_{k^u_\delta} + H_{k^u_{(\delta-1)}} + H_{k^v_{(\delta-1)}})\\
&\leq 8d(u,v)(12H_n)\\
&= 96 \ln n \cdot d(u,v)
\end{aligned}$$

The third to last inequality follows because alternate terms of the summation $\sum_i (H_{k^u_i} - H_{k^u_{(i-2)}})$ telescope. Thus, we have shown that for any edge (u,v), the expected value of $d^T(u,v)$ is $O(\log n) \cdot d(u,v)$. Hence,

THEOREM 1. *The distribution over tree metrics resulting from our algorithm $O(\log n)$-probabilistically approximates the metric d.*

We note that by choosing a slightly different distribution for β, we can ensure that for any x (not just in $[1,2]$), the probability that there is some β_i in $[x, x+dx)$ is $(\frac{1}{x \ln 2})dx$. This then makes the analysis simpler[3], and we do not have to deal with the corner cases above. We omit the details from this extended abstract.

2.4 HSTs

A tree T is said to be *k-hierarchically well separated* if on any root to leaf path the edge lengths decrease by a factor of k in each step. Bartal [7, 8] constructed distributions over trees which were hierarchically well separated, and such trees are more conducive to design of divide-and-conquer type algorithms. The fact that the trees are well separated has been used in applications such as metrical task system[10] and metric labeling [32]. We note that the trees we construct are 2-HSTs. Bartal [8] also observed that a 2-HST can be converted to a k-HST with distortion $O(k)$, later improved to $O(k/\log k)$ [11]. This combined with our result implies a probabilistic embedding into k-HSTs with distortion $O(k \log n/\log k)$. In fact, a slight modification of our technique (details omitted) can be used to directly get k-HSTs for any k, with distortion $O(\frac{k \log n}{\log k})$. This can be useful in some applications, e.g. min-sum k-clustering.

2.5 Derandomization

The problem of probabilistic approximation by tree metrics asks for a distribution over tree metrics such that the expected stretch of each edge is small. A dual problem is find

[3]But somewhat less intuitive.

a single tree such that the (weighted) average stretch of the edges is small. More precisely, given weights w_{uv} on edges, find a tree metric d^T such that for all u, v in V,

- $d^T(u,v) \geq d(u,v)$.
- $\sum_{u,v\in V} w_{uv} \cdot d^T(u,v) \leq \alpha \sum_{u,v\in V} w_{uv} \cdot d(u,v)$.

Charikar et.al.[17] showed that solving this problem is enough for most applications, and moreover can give deterministic algorithms. The algorithm of the previous section clearly gives a randomized algorithm that solves the dual problem for $\alpha = O(\log n)$. We were however looking for deterministic algorithms. The above algorithm can actually be derandomized by the method of conditional expectation as follows.

The algorithm described above tosses coins to choose $\beta \in [1, 2]$ and a permutation π. Since there are only n^2 distinct distances, there are only that many values of β that matter. For each of them, suppose we can compute exactly the expected cost of the tree when π is chosen randomly. Then we can find one for which this expectation is smaller than the average (which is $O(\log n)$). We then choose the permutation π one vertex at a time. To use the method of conditional expectation again, we need to be able to compute, having fixed β and a prefix of the permutation π, the expected cost of the tree, where the expectation is taken over random choices of the rest of π. Assuming we could do this in polynomial time, we simply try all possible choices for the next vertex in the permutation, and pick the one which maximizes the conditional expectation.

It remains to show how to compute the conditional expectations. Given β and some (possibly empty) prefix of π, each edge, at each level is either settled or not. In the former case, the cost at that level is already determined. In the latter case, we know the set of vertices that can settle the edge and the set of vertices that can cut the edge at a particular level. Thus we can compute exactly the expected cost of a particular edge at a particular level. By linearity of expectation then, we can compute the total expected cost. Hence we can solve the dual problem in deterministic polynomial time. In fact, the computation above can be simplified, replacing the exact value of the expected cost above by the upper bounds used in the analysis (and thus using the method of pessimistic estimators [42]).

3. APPLICATIONS

Many problems are easy on trees. The partitioning algorithm we give produces a tree such that the expected stretch of each edge is at most $O(\log n)$. By using our result, the approximation ratios of various problems can be improved. The following is a list of some our favorite applications.

The metric labeling problem : The previous result of Kleinberg and Tardos [32] gives an $O(\log k \log\log k)$-approximation algorithm based on a constant factor approximation for the case that the terminal metric is an HST. Our result improves this to $O(\log n)$.

We also note that Archer, Talwar and Tardos [2] show that the earthmover linear program of Chekuri et.al.[19] is integral when the input graph is a tree. Using this result, the approximation ratio can be improved to $O(\min(\log k, \log n))$.

Buy-at-bulk network design : Awerbuch and Azar [4] give a $O(1)$-approximation algorithm on trees. Thus, we can get an $O(\log n)$-approximation algorithm.

Minimum cost communication network problem : This problem [28, 40, 46] is essentially the dual problem defined in section 2.5 and hence we get an $O(\log n)$ approximation.

The group Steiner tree problem : Garg, Konjevod, and Ravi [24] give an $O(\log k \log n)$-approximation algorithm for trees; thus, we obtain an $O(\log^2 n \log k)$-approximation algorithm, improving on the $O(\lambda \log n \log k)$ result by Bartal and Mendel [12], where $\lambda = O(\min\{\log n \log\log\log n, \log\Delta \log\log\Delta\})$.

Metrical Task system : Improving on the result of Bartal, Blum, Burch and Tomkins [10], Fiat and Mendel [22] gave an $O(\log n \log\log n)$-competitive algorithms on HSTs. Bartal and Mendel's [12] multiembedding result thus gives an $O(\lambda \log n \log\log n)$-competitive ratio, where λ is as defined above. Our result improves this to an $O(\log^2 n \log\log n)$-competitive ratio against oblivious adversaries.

The result also improves the performance guarantees of several other problems such as vehicle routing [16], min sum clustering [11, 9], covering steiner tree [33], hierarchical placement [35], topology aggregation [6, 44], mirror placement [29], distributed K-server [13], distributed queueing [27] and mobile user [5]. We refer the reader to to [8] and [17] for more detailed descriptions of these problems.

4. ACKNOWLEDGEMENTS

We would like to thank Yair Bartal for helpful comments on a previous draft of the paper, and for pointing us to several of the aforementioned applications.

5. REFERENCES

[1] N. Alon, R. M. Karp, D. Peleg, and D. West. A graph-theoretic game and its application to the k-server problem. *SIAM Journal on Computing*, 24(1):78–100, Feb. 1995.

[2] A. Archer, K. Talwar, and E. Tardos. Personal Communication, 2003.

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

[4] B. Awerbuch and Y. Azar. Buy-at-bulk network design. In *38th Annual Symposium on Foundations of Computer Science*, pages 542–547, Miami Beach, Florida, 20–22 Oct. 1997.

[5] B. Awerbuch and D. Peleg. Concurrent online tracking of mobile users. In *SIGCOMM*, pages 221–233, 1991.

[6] B. Awerbuch and Y. Shavitt. Topology aggregation for directed graphs. In *Proceedings of IEEE ISCC (Athens, Greece)*,pp. 47–52, 1998.

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

[8] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *STOC*, 1998.

[9] Y. Bartal. A constant factor approximation for min sum clustering on HSTs. *Personal Communication*, 2003.

[10] Y. Bartal, A. Blum, C. Burch, and A. Tomkins. A polylog(n)-competitive algorithm for metrical task systems. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 711–719, El Paso, Texas, 4–6 May 1997.

[11] Y. Bartal, M. Charikar and R. Raz. Approximating min-sum k-clustering in metric spaces. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, pages 11–20, Hersonissos, Crete, Greece 6–8 July 2001.

[12] Y. Bartal and M. Mendel. Multi-embedding and path approximation of metric spaces. In *Symposium on Discrete Algorithms*, 2003.

[13] Y. Bartal and A. Rosen. The distributed k-server problem-a competitive distributed translator for k-server algorithms. In *IEEE Symposium on Foundations of Computer Science*, pages 344–353, 1992.

[14] J. Bourgain. On lipschitz embeddings of finite metric spaces in hilbert space. *Israel Journal of Mathematics*, 52(1-2):46–52, 1985.

[15] G. Calinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 8–16, 2001.

[16] M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: deterministic approximation algorithms for group Steiner trees and k-median. In *STOC*, pages 114–123, 1998.

[17] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. A. Plotkin. Approximating a finite metric by a small number of tree metrics. In *IEEE Symposium on Foundations of Computer Science*, pages 379–388, 1998.

[18] C. Chekuri, A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair. Embedding k-outerplanar graphs into ℓ_1. In *SODA*, 2003.

[19] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *Symposium on Discrete Algorithms*, pages 109–118, 2001.

[20] G. Even, J. Naor, S. Rao, and B. Schieber. Divide-and-conquer approximation algorithms via spreading metrics (extended abstract). In *IEEE Symposium on Foundations of Computer Science*, pages 62–71, 1995.

[21] J. Fakcharoenphol, C. Harrelson, S. Rao, and K. Talwar. An improved approximation for the 0-extension problem. In *Symposium on Discrete Algorithms*, 2003.

[22] A. Fiat and M. Mendel. Better algorithms for unfair metrical task systems and applications. In *Proceedings of the 32nd STOC*, pages 725-734, 2000.

[23] U. Feige. Approximating the bandwidth via volume respecting embeddings (extended abstract). In ACM, editor, *Proceedings of the thirtieth annual ACM STOC*, pages 90–99, 1998.

[24] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. *Journal of Algorithms*, 37, 2000.

[25] A. Gupta. Steiner points in tree metrics don't (really) help. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 220–227. ACM Press, 2001.

[26] A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair. Cuts, trees and l_1-embeddings of graphs. In *40th FOCS*, pages 399–408, 1999.

[27] M. Herlihy, S. Tirthapura, and R. Wattenhofer. Competitive concurrent distributed queueing. In *Proceedings of the 20th ACM Symposium on Principles of Distributed Computing* 2001.

[28] T. C. Hu. Optimum communication spanning trees. *SIAM J. Computing*, 3:188–195, 1974.

[29] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. On the placement of internet instrumentation. In *Proceedings of IEEE INFOCOM (Tel-Aviv, Israel)*, 2000.

[30] R. Karp. A $2k$-competitive algorithm for the circle. Manuscript, August 1989.

[31] P. N. Klein, S. A. Plotkin, and S. Rao. Excluded minors, network decomposition, and multicommodity flow. In *ACM Symposium on Theory of Computing*, pages 682–690, 1993.

[32] J. M. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. In *IEEE Symposium on Foundations of Computer Science*, pages 14–23, 1999.

[33] G. Konjevod and R. Ravi. An approximation algorithm for the covering steiner problem. *Symposium on Discrete Algorithms*, pp. 338–344, 2000.

[34] G. Konjevod, R. Ravi, and F. Salman. On approximating planar metrics by tree metrics. *IPL: Information Processing Letters*, 80, 2001.

[35] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. Placement algorithms for hierarchical cooperative caching. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms* pp. 586–595, 1999.

[36] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with application to approximation algorithms. In *IEEE Symposium on Foundations of Computer Science*, pages 422–431, 1988.

[37] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *COMBINAT: Combinatorica*, 15, 1995.

[38] J. Matousek. Lectures on discrete geometry. Springer, in press.

[39] I. Newman and Y. Rabinovich. A lower bound on the distortion of embedding planar metrics into euclidean space. In *Proceedings of the eighteenth annual symposium on Computational geometry*, pages 94–96. ACM Press, 2002.

[40] D. Peleg and E. Reshef. Deterministic polylog approximation for minimum communication spanning trees. *Lecture Notes in Computer Science*, 1443:670–681, 1998.

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

[42] P. Raghavan. Probabilistic construction of deterministic algorithms: Approximating packing integer programs. *Journal of Computer and System Sciences*, 37(2):130–143, Oct. 1988.

[43] S. Rao. Small distortion and volume preserving embeddings for planar and euclidean metrics. In *Proceedings of the fifteenth annual symposium on Computational geometry*, pages 300–306. ACM Press, 1999.

[44] Y. Shavitt Topology aggregation for networks with hierarchical structure: A practical approach. In *Proceedings of 36th Annual Allerton Conference on Communication, Control, and Computing (Allerton Park, Illinois)*, 1998.

[45] P. D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.

[46] B. Y. Wu, G. Lancia, V. Bafna, K. Chao, R. Ravi, and C. Y. Tang. A polynomial time approximation scheme for mimimum routing cost spanning trees. In *ACM-SIAM Symposium on Discrete Algorithms*, 1998.