Approximating a Finite Metric by a Small Number of Tree Metrics

M. CHARIKAR * C. CHEKURI † A. GOEL ‡ S. GUHA § S. PLOTKIN ¶

Computer Science Department
Stanford University

Abstract

Bartal [4, 5] gave a randomized polynomial time algorithm that given any n point metric G, constructs a tree T such that the expected stretch (distortion) of any edge is at most $O(\log n \log\log n)$. His result has found several applications and in particular has resulted in approximation algorithms for many graph optimization problems. However approximation algorithms based on his result are inherently randomized. In this paper we derandomize the use of Bartal's algorithm in the design of approximation algorithms.

We give an efficient polynomial time algorithm that given a finite n point metric G, constructs $O(n \log n)$ trees and a probability distribution μ on them such that the expected stretch of any edge of G in a tree chosen according to μ is at most $O(\log n \log\log n)$. Our result establishes that finite metrics can be probabilistically approximated by a small *number of tree metrics. We obtain the first deterministic approximation algorithms for buy-at-bulk network design [2] and vehicle routing [7]; in addition we subsume results from our earlier work [8] on derandomization. Our main result is obtained by a novel view of probabilistic approximation of metric spaces as a deterministic optimization problem via linear programming. This view also provides a new proof of the result in [5] that might be easier to generalize.*

We also show that graphs induced by points in $\Re_p^d$ (d-dimensional real normed space equipped with the l_p norm) can be $O(f(d,p) \cdot \log n)$-probabilistically approximated by tree metrics where $f(d,p) = d^{1/p}$ for $1 \leq p \leq 2$ and $f(d,p) = d^{1-1/p}$ for $2 \leq p$. We use an improved graph partitioning algorithm for normed spaces that obliviously partitions the space into clusters of diameter at most D such that the probability of two points u and v falling in different clusters is at most $O(f(d,p) \cdot \|u-v\|_p / D)$. We also show that our clustering is optimal for all p by giving matching lower bounds.

1 Introduction

Several algorithmic problems (both offline and online) are concerned with distances in a finite metric space induced by an undirected graph (possibly weighted). In some cases the metric property is an integral part of the problem itself while in many other problems the metric property of the graphs is implicit or is used as an aid to solve the problem. An arbitrary metric space (in particular a finite metric defined by a general graph) might not have enough structure to exploit algorithmically. A powerful technique that has been successfully used recently in this context is to embed the given metric space in a *simpler* metric space such that the distances are approximately preserved in the embedding. New and improved algorithms have resulted from this idea for several important problems [20, 10, 4, 6].

Based on the work of Karp [16] and Alon *et al.* [1], Bartal defined *probabilistic* approximation of metric spaces by a set of simpler metric spaces. Formally, let M be a finite metric space defined on a vertex set V. For two metric spaces M_1 and M_2 defined on the same vertex set V, M_1 dominates M_2 if $d_{M_1}(u,v) \geq d_{M_2}(u,v)$ for all $u, v \in V$. Let $\mathcal{S}$ be a set of metric spaces on V such that each metric space in $\mathcal{S}$ *dominates* M. $\mathcal{S}$ is said to α*-probabilistically* approximate M if there is a probability distribution μ over $\mathcal{S}$ such that, for every pair of vertices in M, the expected distance between them in a space chosen according to μ, is

*Supported by the Pierre and Christine Lamond Fellowship, an ARO MURI Grant DAAH04-96-1-0007 and NSF Award CCR-9357849, with matching funds from IBM, Schlumberger Foundation, Shell Foundation, and Xerox Corporation. Email: moses@cs.stanford.edu.

†Supported by an IBM Cooperative Fellowship, an ARO MURI Grant DAAH04-96-1-0007 and NSF Award CCR-9357849, with matching funds from IBM, Schlumberger Foundation, Shell Foundation, and Xerox Corporation. Email: chekuri@cs.stanford.edu.

‡Supported by Army contracts DAAG55-98-1-0170/DAAG55-97-1-0221 and ONR N00014-98-1-0189. Email: agoel@cs.stanford.edu.

§Supported by an ARO MURI Grant DAAH04-96-1-0007 and NSF Award CCR-9357849, with matching funds from IBM, Schlumberger Foundation, Shell Foundation, and Xerox Corporation. Email: sudipto@cs.stanford.edu.

¶Supported by Army contracts DAAG55-98-1-0170/DAAG55-97-1-0221 and ONR N00014-98-1-0189. Email: plotkin@cs.stanford.edu

at most α times the distance of the pair in M. Probabilistic approximation is a weaker notion of metric approximation than that of deterministic embedding into a single space, but is sufficient for many applications. Bartal [4] showed that any n point metric space can be $O(\log^2 n)$-probabilistically approximated by a class of *tree metrics*. He also gave a polynomial time algorithm to pick a tree from the distribution. He subsequently improved his result to obtain an $O(\log n \log \log n)$ approximation [5]. Tree metrics are a very natural class of simple metric spaces since many algorithmic problems become tractable on them. It can be shown that there are finite metric spaces that cannot be deterministically approximated by one tree metric with a distortion of $o(n)$ (see [26] for a lower bound for the n cycle). In fact probabilistic approximations were motivated by the preceding observation. Bartal's result has found several applications in online algorithms, approximation algorithms, distributed computing and others (see [4, 5] for more details).

Our focus in this paper is on the application of Bartal's results to approximation algorithms for NP-hard graph optimization problems. Consider a graph optimization problem where the goal is to find a set of edges satisfying some constraints, and the objective function is to minimize a linear combination of the lengths of the chosen edges. Many problems in network design and routing are examples of such optimization problems where the constraints typically express connectivity requirements. Bartal's result can be used to approximate the graph by a tree such that the expected distance between a pair of vertices in the tree is at most an $O(\log n \log \log n)$ factor of their distance in the graph. The problem can now be solved on the tree and the solution interpreted on the original graph. Since the objective function is a linear combination of distances, linearity of expectations implies that the optimal solution on the tree is, in expectation, within an $O(\log n \log \log n)$ factor of the optimal solution on the graph. This approach has been used to obtain improved approximations for several problems including group Steiner trees [12], k-median [5, 8], minimum communication cost spanning trees [28], buy-at-bulk network design [2], and vehicle routing [7]. Since the first step in the above approach consists of constructing a tree probabilistically, all the approximation algorithms which result are *randomized*. Our goal in this paper is to *derandomize* these algorithms and future applications.

The main result of this paper is a polynomial time algorithm that given a weighted graph on n vertices constructs a probability distribution on $O(n \log n)$ trees such that the expected stretch of any edge of the graph in a tree chosen from the distribution is $O(\log n \log \log n)$. This is in contrast to Bartal's algorithms [4, 5] that generate trees from exponentially large distributions. Using our result we can derandomize applications to approximation algorithms by solving the problem at hand on each of the trees from the distribution, and picking the best of the solutions. This result builds upon our earlier work [8], where applications that fit a certain restricted framework that included group Steiner trees, k-median, and minimum communication cost spanning tree, were derandomized. The framework of [8] was not applicable to some applications and the construction of a small distribution of trees was identified as a way to derandomize additional applications. Using our main result we obtain the first *deterministic* approximation algorithms for buy-at-bulk network design [2] and some variants of vehicle routing [7] that match the randomized approximation ratios. Further, our result establishes that future applications can assume a small distribution and hence be derandomized. Our derandomization is practical for two reasons: the small size of the distribution, and an efficient algorithm to construct the distribution. The small size of the distribution also makes it feasible to preprocess a graph and store the trees for repeated use.

Our main result is a consequence of viewing probabilistic approximations in a novel way. Let M be a finite metric space and $\mathcal{S}$ a collection of finite metric spaces that dominate M. We phrase the problem of finding the best (in the sense of least distortion) probabilistic approximation of M by $\mathcal{S}$ as a *fractional packing* problem (with $|\mathcal{S}|$ variables.) We then use the algorithm of Plotkin, Shmoys, and Tardos (PST) [25] for approximately solving this fractional packing problem. The PST framework requires a certain dual subroutine; when $\mathcal{S}$ is the space of tree metrics the dual subroutine turns out to be the minimum communication cost spanning tree problem which is NP-hard [11, 28]. However an approximate dual subroutine can be obtained using graph partitioning techniques from [13, 27, 9], as shown in [5, 8] which give an $O(\log n \log \log n)$ guarantee on the approximation ratio. Since the best upper bound we have on the optimal value of the fractional packing problem is $O(\log n \log \log n)$, a naive application of the approximate dual in PST gives only an $O((\log n \log \log n)^2)$-probabilistic approximation. We modify the PST framework to exploit the fact that the dual subroutine gives a bound on the *integrality gap* as well as an approximation ratio of $O(\log n \log \log n)$; this results in an $O(\log n \log \log n)$ probabilistic approximation of a finite metric by a distribution on $O(n \log n)$ trees. We believe that our modification to PST (Theorem 2.3) is of independent interest. Using improved graph partitioning results, we show that planar graphs can be $O(\log n)$-probabilistically approximated (matching the result of [18] but using a small distribution), and points in d-dimensional Euclidean spaces can be $O(\sqrt{d} \log n)$-probabilistically approximated using a distribution on $O(n \log n)$ and $O(dn \log n)$ trees, respectively. For Euclidean spaces our result improves on the earlier known result of $O(d \log n)$ [22]. The improvement is obtained by a new graph partitioning procedure for Euclidean spaces that we describe below. One of our contributions, apart from obtaining a polynomial size distribution, are our new proofs for probabilistic approximation by tree metric spaces. Further, probabilistic approximation by any class of metric spaces $\mathcal{S}$ is reduced to an optimization problem, thus providing a concrete way to approach classes of metric spaces other than trees.

Low Diameter Partitioning of Real Normed Spaces

Clustering of metric spaces has applications in data analysis, classification, learning, facility location, and several other areas. Clustering objectives vary considerably depending on the application. One particular objective that has found several applications is that of low diameter clustering. The objective is to partition a metric space (induced usually by a weighted undirected graph) into subgraphs such that the diameter of each of the subgraphs is low. Peleg *et al.* and others [23, 3] studied such partitions motivated by applications in network routing and distributed computing. Leighton and Rao [19] pioneered the use of low diameter clustering for the design of approximation algorithms based on a divide and conquer approach. See [13, 17, 9] for related results. Bartal [4] used probabilistic partitioning of graphs into low diameter clusters to obtain tree approximations.

In this paper we consider low diameter clustering when the graphs are induced by points in real normed spaces. Given a metric space G with a distance function c, let μ be a probability distribution over the partitions of G into clusters of diameter at most D. Let x_{uv} be the probability of vertices u and v belonging to different clusters in a partition picked according to μ. Define β to be the quantity $\max_{u \neq v}\{x_{uv} \cdot D/c_{uv}\}$. The probability of vertices u and v being "cut" is at most $\beta c_{uv}/D$. If β is small then vertices which are close together in G belong to the same cluster with high probability. Our objective is to give a probabilistic partitioning algorithm which guarantees a small value of β.

We obtain an improved graph partitioning algorithm when the graph is induced by points in $\Re_2^d$. For any $D > 0$ we show that a graph embedded in $\Re_2^d$ can be clustered into regions of diameter at most D, such that β is at most $2\sqrt{d}$. Notice that β is independent of n. Earlier partitioning algorithms achieved $\beta = O(d)$ for Euclidean graphs [22], and $O(\log n)$ for general metrics [4]. Since the dimension d can be reduced to $O(\log n)$ with only a small increase in distances [15], our upper bound gives at least an $\Omega(\min(\sqrt{d}, \sqrt{\log n}))$ improvement over the previous best results for graphs induced by $\Re_2^d$. We generalize our results to graphs induced by points in $\Re_p^d$ ($\Re^d$ equipped with the l_p norm). We prove a lower bound of $\Omega(d)$ on β for graphs induced by $\Re_1^d$; an identical lower bound has recently been proved by Indyk for graphs induced by $\Re_\infty^d$ [14]. These two lower bounds and our upper bound for $\Re_2^d$ can be combined to obtain matching upper and lower bounds of $\Theta(f(d,p))$ on β, where $f(d,p) = d^{1/p}$ for $1 \leq p \leq 2$ and $f(d,p) = d^{1-1/p}$ for $p \geq 2$. We use our partitioning algorithm to obtain a $O(f(d,p) \cdot \log n)$-probabilistic approximation for n point metric spaces induced by points in $\Re_p^d$.

The rest of the paper is organized as follows. We present the linear programming formulation in Section 2 and show how to apply the fractional packing framework of [25] to solve it approximately in polynomial time. We describe our partitioning results in Section 3.

2 Probabilistic Approximation via Linear Programming

In this section we show how the problem of probabilistic approximation by a class of metric spaces $\mathcal{S}$ can be formulated as a linear programming problem. We consider only the case when $\mathcal{S}$ is the class of tree metrics but the formulation and solution strategy generalize in a straight forward manner to other spaces as well. Let G be a weighted undirected graph on n vertices that induces a finite metric. Let $\mathcal{S} = \{T_1, T_2, \ldots, T_N\}$ be a set of tree metrics on the vertices of G. An important point to remember is that the trees T_i could have virtual vertices in addition to the vertices of G. We assume without loss of generality that $\mathcal{S}$ is finite although N could be an arbitrarily large function of n. We assume that each of the metrics in $\mathcal{S}$ *dominates* G, that is $d_{T_i}(u,v) \geq d_G(u,v)$ for every pair of vertices (u,v). Let $c(e)$ denote the length of an edge e in G. We assume that $c(u,v)$ is less than or equal to the distance between u and v in G. A probability distribution on $\mathcal{S}$ can be represented by assigning to each $T_i \in \mathcal{S}$ a real value $0 \leq x_i \leq 1$, with the constraint that the x_i sum to one. With this simple observation we obtain the following linear program to find the optimal probabilistic approximation of G by $\mathcal{S}$.

$$
\begin{array}{rcll}
\min & \lambda & & \\
\sum_i d_{T_i}(e) \cdot x_i & \leq & \lambda \cdot c(e) & \text{for every edge } e \\
\sum_i x_i & = & 1 & \\
x_i & \geq & 0 &
\end{array}
$$

In other words , we have to solve the problem $\min\{\lambda : Dx \leq \lambda c, \sum_i x_i = 1\}$ where $D_{e,i} = d_{T_i}(e)$. Let D_e refer to the e-th row of D. The LP above has a large number of variables, but only $(m+1)$ constraints where m is the number of edges. It can be shown that there is an optimal solution to the above LP that has at most $(m+1)$

non-zero x_i. Thus, the existence of a small distribution is guaranteed from the formulation itself (this was first observed in [8]). In addition, the formulation can be looked upon as a fractional packing problem, and we will exploit this structure to solve it. Let $\rho_e = \max_i d_{T_i}(e)/c(e)$ be the maximum stretch of edge e in any of the trees, and let $\rho = \max_e \rho_e$ be the maximum stretch of any edge. It follows that $D_e x \leq \rho c_e$ for all x satisfying $\sum_i x_i = 1$. The parameter ρ is important in solving the LP and we will show that we can restrict ourselves to a space of trees with ρ polynomial in n.

The key to solving the above LP is to view it as a packing problem. A packing linear program is formulated as $\min\{\lambda : Ax \leq \lambda b, x \in P\}$, where P is a convex polytope, and $Ax \geq 0$ for $x \in P$. Plotkin, Shmoys, and Tardos [25] showed that a packing problem can be solved to ϵ optimality (that is find a λ such that $\lambda \leq (1+\epsilon)\lambda^*$) given the following subroutine.

- A DUAL procedure that given a vector $y \geq 0$ (y is an $m \times 1$ column vector if A is an $m \times n$ matrix) finds an $\tilde{x} \in P$, such that $c\tilde{x} = \min_{x \in P} cx$ where $c = y^t A$.

A crucial parameter for their algorithm is ρ, the *width* of the polytope P with respect to the constraint $Ax \leq \lambda b$. The width is defined as $\max_i \max_{x \in P} \frac{A_i x}{b_i}$. The following theorem states the main result of [25].

Theorem 2.1 (PST94) *There is an algorithm that solves a packing problem to ϵ optimality with $O(\epsilon^{-2}\rho\log(\rho/\epsilon))$ calls to the* DUAL *procedure.*

The running time of the above algorithm depends only on the DUAL procedure and the width. In particular there is no explicit dependence on the number of variables. Our goal in the rest of the section is to show how we can adapt the algorithm of [25] to solve our LP formulation. It turns out that the DUAL procedure as required by the PST algorithm is NP-hard for our program. We will show however that we can approximately solve the dual problem and also modify the proof of [25] to show that an approximate solution is sufficient.

2.1 Solving the Fractional Packing Problem

As we saw, our LP formulation is easily viewed as a packing problem. In particular the polytope P is defined by the constraint $\sum_i x_i = 1$ and $x_i \geq 0$. There are two important consequences of the simplicity of P.

1. The DUAL procedure is required to return a $\tilde{x} \in P$ that minimizes $y^t Ax$ over P. Since any point in our polytope is a convex combination of trees, it follows that there exists a *single* tree that achieves the minimum. This implies that a bound on the number of calls to the DUAL procedure gives a bound on the number of trees generated in the approximate solution.

2. The space of trees $\mathcal{S}$ and the width ρ are implicitly defined by the DUAL procedure. This follows from the fact that the only trees considered by the PST algorithm are those returned by the DUAL procedure.

We now interpret the DUAL procedure for our formulation. The vector y of dual variables corresponds to assigning weights $y(e) \geq 0$ for each edge e. Let $C^*(y) = \min_{T \in \mathcal{S}} \sum_e y(e) \cdot d_T(e)$. Thus the DUAL procedure is to find a tree that minimizes the average y-weighted edge lengths of G. Unfortunately, as the following remark shows, we cannot hope to solve the DUAL problem optimally in polynomial time.

Remark 1 *The* DUAL *procedure for our problem in NP-hard. In fact it is exactly the minimum communication cost spanning tree problem on metric spaces [11, 28].*

Therefore we settle for approximation algorithms. The first deterministic approximation algorithms for the minimum communication cost spanning tree problem on metric spaces achieving a $O(\log n \log\log n)$ approximation were provided independently in [5] and [8]. We need a stronger version of the result in [5, 8] that in addition to giving the above approximation ratio also provides a guarantee on the worst case stretch of any edge. In this paper we also give improved results for metric spaces induced by real normed spaces that we discuss in more detail in Section 3. The theorem below encapsulates the required conditions. We provide more details in Subsection 2.2.

Theorem 2.2 *Let G be a weighted graph on n vertices that induces a metric space. Given positive weights $y(e)$ on the edges, there is a polynomial time algorithm that finds a tree T that dominates G such that $\sum_e y(e) \cdot d_T(e) \leq O(\log n \log\log n) \sum_e y(e) \cdot c(e)$. Further, for every edge e, $d_T(e)/c(e) = O(n)$. For planar graphs and for graphs induced by points in d dimensional Euclidean space the $O(\log n \log\log n)$ term improves to $O(\log n)$ and $O(\sqrt{d}\log n)$ respectively.*

As mentioned before, we need a modified version of the PST algorithm [25] that gives a guarantee when the DUAL procedure is approximate. Since every tree $T \in \mathcal{S}$ dominates G, $\sum_e y(e) \cdot c(e)$ is a lower bound on $C^*(y)$ and thus trivially $C(y) \leq O(\log n \log\log n) C^*(y)$. The upper bound we have on λ^*, the optimal of our formulation, is $O(\log n \log\log n)$. Theorem 2.2 viewed as providing an approximation algorithm for $C^*(y)$ with a ratio of $O(\log n \log\log n)$ guarantees only a $O((\log n \log\log n)^2)$-probabilistic approximation. We will exploit the stronger guarantee of Theorem 2.2 to obtain a $O(\log n \log\log n)$-probabilistic approximation. We restrict ourselves to packing problems in the formalism of Plotkin *et al.* [25] that satisfy the additional condition that $Ax \geq b$ for $x \in P$. This guarantees that $C^*(y) \geq y^t b$ for all $y \geq 0$. We obtain the following theorem by a careful restatement of the algorithm and proof in [25]; the details of these modifications are omitted from this version. We believe that the theorem is of independent interest and might find future applications.

Theorem 2.3 *Consider a packing problem with $Ax \geq b$ for $x \in P$. If the* DUAL *procedure guarantees $C(y) \leq \beta y^t b$, the algorithm in [25] finds a $\tilde{x} \in P$ such that $A\tilde{x} \leq (1+\epsilon)\beta b$ in $O(\epsilon^{-2}\rho\log(\rho/\epsilon))$ calls to the* DUAL *procedure. If the* DUAL *procedure has the property that $C(y) \leq \beta C^*(y)$, $\tilde{x}$ satisfies $A\tilde{x} \leq (1+\epsilon)\beta\lambda^* b$.*

With Theorems 2.2 and 2.3 in place we put the pieces together to state and prove our main result.

Theorem 2.4 *Every finite metric on n points can be $O(\log n \log\log n)$-probabilistically approximated by a probability distribution on $O(n \log n)$ trees. The trees and the distribution can be computed in polynomial time.*

Proof: We apply Theorem 2.3 to our packing formulation of the problem. Our packing problem satisfies the condition of Theorem 2.3 since every tree in $\mathcal{S}$ dominates G. From Theorem 2.2, we have a dual procedure that guarantees that $C(y) = O(\log n \log\log n) y^t c$. Therefore we obtain an $\tilde{x}$ that satisfies $D\tilde{x} \leq O(\log n \log\log n) \cdot c$. Since the width guaranteed by the DUAL procedure of Theorem 2.2 is $O(n)$, it follows that PST algorithm will terminate in $O(\epsilon^{-2} n \log(n/\epsilon))$ calls to DUAL . Each call to the DUAL procedure results in one tree, therefore $\tilde{x}$ has at most $O(\epsilon^{-2} n \log(n/\epsilon))$ non-zero entries. Choosing some small but fixed ϵ we obtain the desired result. $\Box$

We obtain the following corollary using stronger results for special cases.

Corollary 2.5 *Planar graphs can be $O(\log n)$-probabilistically approximated and points in $\Re_2^d$ can be $O(\sqrt{d}\log n)$-probabilistically approximated by a distribution on $O(n\log n)$ and $O(dn\log n)$ trees respectively.*

We also obtain results for points in $\Re_p^d$ where the distance is measured using the l_p norm instead of the standard Euclidean norm. We describe the generalization in Section 3.

We end this subsection with some comments.

- Our proof of probabilistic approximation is based on the deterministic tree construction guaranteed by Theorem 2.2 that produces a single tree to minimize a weighted linear combination of edge lengths. Earlier algorithms in [5] and [18] for probablistic approximation are based on randomized versions of the deterministic tree construction. Our approach differs in that we use linear programming to view the problem as a deterministic optimization problem. Our view results in a distribution that is small and has applications to derandomization. By establishing a direct connection to the dual procedure in our proof, we are able to obtain stronger results for special cases like planar graphs and real normed spaces in a simple way.
- There is a lower bound of $O(\log n)$ on the expected distortion provided by tree metrics [18] even for planar graphs. However a specific metric space might have a better approximation. This is captured by the optimal solution to our linear programming formulation. The second part of Theorem 2.3 shows that an improved approximation to the minimum communication cost spanning tree can result in relating the distortion we guarantee to the best possible for the given metric space.
- The tree metrics constructed by Bartal [4, 5] called HSTs satisfy a hierarchical separation property. An HST is a rooted tree such that the edge lengths along each root to leaf path decrease by a constant factor. This additional property is useful in certain applications (see [4] for details). We note that the trees guaranteed by Theorem 2.2 are also HSTs.
- It is possible to use the parallel algorithm for positive linear programming of Luby and Nisan [21] in place of [25] at the cost increasing the number of trees generated by a poly-logarithmic factor. However we do not have a parallel tree construction procedure to take advantage of the algorithm of [21].

2.2 The Dual Procedure

In this subsection we sketch some details of the DUAL procedure and the proof of Theorem 2.2. We recall the problem below. Given a weighted graph G and a positive weight $y(e)$ associated with each edge e, the objective is to find a tree such that $\sum_e y(e) \cdot d_T(e)$ is minimized. The constraint on the tree is that $d_T(e) \geq c(e)$, that is the metric on T dominates G. This problem is closely related to the minimum communication cost spanning tree (MCST), the only difference being that in MCST, the tree is required to be a spanning tree of G. The metric version of MCST is when G is a complete graph and forms a metric space. Metric MCST is NP-hard even when $y(e) = 1$ for all e [28]. The metric version corresponds exactly to our dual problem. The dual procedure can be thought of as finding a tree that approximates a weighted combination of distances in G, the weights being provided by y.

The first *deterministic* approximation algorithm for the metric MCST problem were provided independently by Bartal [5] and Charikar *et al.* [8]. They show that a tree that satisfies $\sum_e y(e)d_T(e) \leq O(\log n \log\log n)\sum_e y(e)c(e)$ can be obtained in polynomial time. For sake of completeness we give some details of the construction. The main idea is to use a divide and conquer strategy based on decomposing the graph into small diameter subgraphs. The algorithm is outlined below.

1. Partition G into $G_1, G_2, \ldots, G_k$ such that $\Delta(G_i) \leq \Delta(G)/2$. Δ denotes the diameter.
2. Recursively construct trees $T_1, \ldots, T_k$ for $G_1, \ldots, G_k$.
3. Create a tree T by joining the roots of each T_i to a new root r by edges of length $\Delta(G)/2$.

The approximation guarantee relies on the graph partitioning in Step 1. Low diameter partitionings have been extensively used in the design of approximation algorithms pioneered by the work of Leighton and Rao [19]. By using the partitioning algorithm of Garg, Vazirani, and Yannakakis [13] in the above scheme, we can obtain an $O(\log^2 n)$ approximation. This can be improved to $O(\log n \log\log n)$ approximation by a more sophisticated analysis of the partitioning procedure first used by Seymour [27], and subsequently developed into a more broadly applicable divide and conquer paradigm called spreading metrics by Even *et al.* [9]. It is worth noting that the above mentioned partitioning methods have been applied to problems where the underlying metric is obtained by solving a linear program, while the weights are capacities given as input to the problem. In contrast, in our case, the metric is defined by the edge lengths of the graph and the weights $y(e)$ are provided by the packing algorithm of [25].

2.2.1 Reducing the Width

The algorithm for the tree construction described above provides a good approximation to preserve the weighted sum of distances. However, as mentioned before, we need an additional condition on the width of the polytope. In Theorem 2.2 we guarantee that for all edges $d_T(e)/c(e) = O(n)$. We employ a simple trick to make sure that only sufficiently long edges are cut in each partitioning step (a similar idea was used by Bartal [4] in a slightly different context). We alter Step 1 of the algorithm described earlier. In the description below, H is a graph at some level of the recursion.

- Contract all edges of H of length less than $\Delta(H)/4n$ to obtain H'.
- Partition H' into $H'_1, \ldots, H'_k$ such that $\Delta(H'_i) \leq \Delta(H)/2$.
- Expand all the contracted edges in each of the H'_is to obtain $H_1, \ldots, H_k$.

The contraction ensures that only edges of length at least $\Delta(H)/4n$ are cut in the partitioning procedure. This guarantees that the stretch of any edge is $O(n)$. Another easy observation is that $\Delta(H_i) \leq \Delta(H'_i) + \Delta(H)/4 \leq 3\Delta(H)/4$. It can be verified that the framework of [27, 9] can be used in conjunction with this contraction – the details are tedious and are omitted. The same contraction idea works for planar graphs as well. The partitioning procedure in Section 3 for a graph induced by points in a geometric space uses the continuous properties of the space. The idea of contracting edges cannot be applied. We will use other properties of the partitioning step to reduce the width.

3 Low Diameter Partitioning of Real Normed Spaces

Let G be a graph induced by n points in $\Re_p^d$ i.e, the vertices correspond to points in d-dimensional space, and the distances are induced by the l_p norm (the l_p distance between two d dimensional vectors u and v, denoted as $||u - v||_p$, is defined to be $(\sum_{i=1}^d |u_i - v_i|^p)^{1/p}$). Equivalently, the graph G is said to be embedded in $\Re_p^d$. Let c_{uv} represent the distance between the vertices u and v.

We solve the following problem: given a graph $G(V)$ embedded in $\Re_p^d$, construct a probability distribution over partitions of G into disjoint clusters of diameter $\leq D$, such that the quantity $\beta(G) = \max_{u \neq v}\{(D/c_{uv}) \cdot x_{uv}\}$ is small, where x_{uv} is the probability of vertices u and v belonging to different clusters. Define $\beta_p(d)$ to be the maximum value of $\beta(G)$ over all graphs embedded in $\Re_p^d$. We will omit the subscript p and the argument d from $\beta_p(d)$ where these values are clear from the context. An algorithm that solves the above problem can be used to construct an $O(\beta \log n)$-probabilistic approximation of G using tree metrics [4].

For any graph G there is a trivial lower bound of $\Omega(1)$ on β. For arbitrary graphs Bartal [4] gave an algorithm which guarantees $\beta = O(\log n)$, and also showed the existence of graphs for which $\Omega(\log n)$ is a lower bound. For planar graphs, an algorithm for obtaining $\beta = O(1)$ is implicit in [17] and was used by [18] to obtain an $O(\log n)$-probabilistic approximation of planar graphs by trees.

Less is known about partitioning graphs that are embeddable in geometric spaces. We give tight upper and lower bounds for the value of β as a function of d, for all $p \geq 1$. The bound is $\Theta(d^{1/p})$ if $1 \leq p \leq 2$ and $\Theta(d^{1-1/p})$ if $p \geq 2$. The upper bound follows from the partitioning algorithm for graphs embedded in $\Re_2^d$, presented in Section 3.1. The lower bound follows by interpolating our lower bound for $\Re_1^d$ (Section 3.2) and a recent lower bound for $\Re_\infty^d$ by Indyk [14]. Our upper bound is a significant improvement over the bound of $O(\log n)$ for general graphs. Specifically, if $p = 2$ then the dimension of the space can be reduced to $O(\log n)$ without distorting edge lengths by more than a constant factor[15]. Thus for Euclidean graphs, it is possible to beat the lower bound for general graphs by at least a factor of $\Omega(\sqrt{\log n})$. Peleg *et al.* [22] have recently given a partitioning procedure that results in $\beta = O(d)$ for Euclidean graphs.

3.1 Upper Bound: Partitioning Using Spheres

We restrict ourselves to graphs embedded in $\Re_2^d$. The partitioning algorithm is the following: draw spheres of radius $R = D/2$ around each vertex. Repeatedly pick points uniformly at random from the region defined by the union of all these spheres. Each time a point is picked, look at all the vertices within a distance R from this point. Put all these vertices into a cluster of their own, delete these vertices from the graph, and proceed. The algorithm terminates after at most n iterations.

Theorem 3.1 *The above algorithm guarantees $\beta \leq 2\sqrt{d}$ for graphs embedded in $\Re_2^d$.*

Proof: Consider any two vertices u and v in G. Let $S(x)$ denote the Euclidean sphere of radius R centered at x. Let $V_k(r)$ be the volume of a sphere of radius r in k dimensions. Further, let $V_k(r) = C_k \cdot r^k$.

The two points u and v go into different clusters if a point in $(S(u)-S(v)) \cup (S(v)-S(u))$ gets picked before a point in $S(u) \cap S(v)$. But

$$\text{vol}\,(S(u) - S(v)) = \text{vol}\,(S(u) \cup S(v)) - \text{vol}\,(S(v))\,.$$

Draw two hyperplanes orthogonal to the line (u, v) and passing through the points u and v respectively (Figure 1). Let S_l denote the hemisphere to the left of u and S_r the hemisphere to the right of v. Further, let L denote the cylindrical region bounded by the base of the hemispheres. The area of the base of L is $V_{d-1}(R)$ and the height of L is c_{uv}; therefore the volume of L is $c_{uv}.V_{d-1}(R)$. The region $S_l \cup S_r \cup L$ contains the region $S(u) \cup S(v)$. Therefore

$$\begin{aligned}
&\text{vol}\,(S(u) \cup S(v)) - \text{vol}\,(S(v)) \\
&\leq \text{vol}(L) + \text{vol}(S_l) + \text{vol}(S_r) - \text{vol}\,(S(v)) \\
&\leq c_{uv} \cdot V_{d-1}(R).
\end{aligned}$$

i.e. $x_{uv} \leq 2c_{uv}V_{d-1}(R)/V_d(R) = (2c_{uv}/R)\cdot(C_{d-1}/C_d)$.

The diameter of each cluster is at most $2R$. By definition of β, we now have $\beta \leq 4C_{d-1}/C_d$. But $C_{d-1}/C_d = \dfrac{\Gamma(1 + d/2)}{2\Gamma(1/2 + d/2)\Gamma(3/2)}$ [24]. It is easy to verify that this expression is at most $\sqrt{d}/2$, which implies that $\beta \leq 2\sqrt{d}$. □

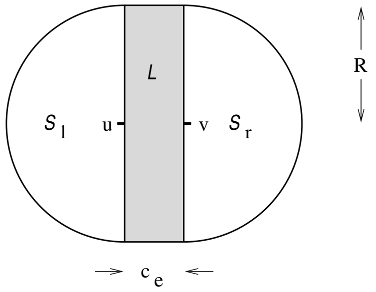

Figure 1. The shaded region, L, is an upper bound on the volume of $S(u) - S(v)$.

The same algorithm can also be used to cluster spaces with any l_p norm where $p \geq 1$. Therefore the partitioning algorithm does not need to know the underlying norm associated with the space – it just needs to know the diameter D of the spheres that are used for partitioning.

We will use the following elementary fact repeatedly.

Fact 3.1 *Let x be a vector in $\Re^d$, and let $p > q$. Then $||x||_p \leq ||x||_q \leq d^{1/q-1/p}||x||_p$.*

Theorem 3.2 *For $1 \leq p \leq 2$, $\beta \leq 2d^{1/p}$, and for $p \geq 2$, $\beta \leq 2d^{1-1/p}$.*

Proof: Consider the points u and v. Let $c_p = ||u - v||_p$. Further, let D_p be the l_p diameter of the convex region defined by a Euclidean ball of l_2 diameter D. Let x be the probability of u and v falling into different clusters. Assume that the vertices u and v are the ones that maximize the ratio x/c_p. Then $\beta = xD_p/c_p = (xD_2/c_2)\cdot(D_p/D_2)\cdot(c_2/c_p)$. From Theorem 3.1, $xD_2/c_2 \leq 2\sqrt{d}$. Using Fact 3.1, if $1 \leq p < 2$ then $c_2/c_p \leq 1$ and $D_p/D_2 \leq d^{1/p-1/2}$. Therefore $\beta \leq 2d^{1/p}$. And if $p > 2$ then $c_2/c_p \leq d^{1/2-1/p}$ while $D_p/D_2 \leq 1$, which implies that $\beta \leq 2d^{1-1/p}$. □

3.2 Lower Bound on β

We show that the upper bound on β in Theorem 3.2 is tight up to constant factors for all $p \geq 1$ if the underlying graph is the d-dimensional mesh of side n.

Lemma 3.1 *Consider the d-dimensional mesh of side r and with distances defined by the l_1 norm. Then $\beta = \Omega(d)$ for any probabilistic partitioning of this mesh into clusters of l_1 diameter δ, $d < \delta < d \cdot r/24$.*

Proof: Two vertices in the mesh are said to be *adjacent* if they agree on $d-1$ coordinates and differ by one in the remaining coordinate; an edge is said to be *basic* if it is between two adjacent vertices. Consider any probabilistic partitioning of the mesh with clusters of l_1 diameter at most δ. Let S be the cluster with the largest number of points. Let $|S| = k^d$ (k need not be integral). Let A be the number of basic edges that are incident on exactly one vertex in S. Alon *et al.* [1] prove the following result:

$$A \geq dk^{d-1}(1 - k/r).$$

The number of basic edges within the cluster is at most $2dk^d$. Thus the probability of a basic edge being cut in this partitioning is $\Omega((1/k)(1 - k/r))$. If $\delta > d$ then the number of points in a cluster of l_1 diameter δ is at most $(12\delta/d)^d$ [24]. This implies $k \leq 12\delta/d$. Since $\delta < d \cdot r/24$, we have $1 - k/r > 1/2$. Thus the probability of a basic edge being cut is $\Omega(d/\delta)$. Since each basic edge has length 1, this gives a lower bound of $\Omega(d)$ on β for graphs embedded in $\Re_1^d$. □

Indyk [14] also uses the the result of Alon *et al.* [1] along with certain properties of cut-metric embeddability to prove an identical bound for the l_∞ norm. We combine his lower bound, Lemma 3.1, and Fact 3.1 to obtain the following theorem.

Theorem 3.3 *Consider the d-dimensional mesh of side n and with distances defined by the l_p norm. Then $\beta = \Omega(d^{1/p})$ if $1 \leq p \leq 2$ and $\beta = \Omega(d^{1-1/p})$ if $p \geq 2$ for any probabilistic partitioning of this mesh into clusters of l_p diameter $\delta > d$.*

Thus we have matching upper and lower bounds on β for all l_p norms, $p \geq 1$. Interestingly, for all norms, the upper bound is achieved by the same algorithm (independent of p) and the lower bound is achieved on the same graph.

3.3 Approximating Graphs in $\Re_p^d$ with Tree Metrics

In this section we show how to solve the minimum communication cost spanning tree problem where the underlying graph is embedded in $\Re_2^d$ – the construction for $\Re_p^d$ follows from the construction for $\Re_2^d$ using Fact 3.1. We also show that the width of the tree constructed is at most $d \cdot n$. Using this as the dual procedure in the algorithm of Section 2 gives us the result claimed in Corollary 2.5.

Consider a graph G consisting of n points embedded in $\Re_2^d$, and let c_{uv} denote the l_2 distance between vertices u and v. Let y_{uv} be the weight assigned to the pair (u, v). Clearly $L = \sum_{u,v \in G} c_{uv} \cdot y_{uv}$ is a lower bound on the cost of the minimum communication cost spanning tree. Let the diameter of the graph be Δ. We cluster the graph with spheres of radius $R = \Delta/4$ using the algorithm in Section 3.1. Let C be the set of all pairs of vertices such that the two vertices belong to different clusters. Call a partitioning *good* if it satisfies the following two criteria:

1. $R \sum_{(u,v) \in C} y_{uv} \leq 4\sqrt{d} \cdot L.$

2. No two vertices i and j with $c_{ij} \leq R/(8dn)$ belong to different clusters.

Condition 1 ensures that the total cost of the tree is low; condition 2 guarantees that the width of the polytope (see Section 2.2.1) is small.

Lemma 3.2 *A partitioning is good with probability at least* $1/2$.

Proof: Let E_1 be the expected value of $R\sum_{(u,v)\in C} y_{uv}$ for a partitioning. Also, let x_{uv} be the probability of vertices u and v belonging to C. From linearity of expectations,

$$E_1 = R \sum_{u \in G, v \in G} x_{uv} y_{uv}.$$

But $x_{uv} \leq 2\sqrt{d} c_{uv}/(2R)$ (Theorem 3.1). Therefore

$$E_1 \leq \sqrt{d} \sum_{u \in G, v \in G} c_{uv} y_{uv} = \sqrt{d} L.$$

Using Markov's inequality, the quantity $R\sum_{(u,v)\in C} y_{uv}$ is greater than $4\sqrt{d}L$ with probability at most $1/4$, and condition (1) above gets violated with probability at most $1/4$.

Draw a subgraph H with the same vertex set as G and with an edge between all vertices i and j such that $c_{ij} \leq R/(8dn)$. Let P be a maximal connected component of H with k vertices; embed P back into $\Re_2^d$ such that each vertex in P coincides with the corresponding vertex in G. There must exist some vertex v in P such that a ball around v of radius $kR/(16dn)$ will cover all the vertices in P. P is said to be cut if this ball gets partitioned during the partitioning. Draw two concentric balls of radius $R - kR/(16dn)$ and $R + kR/(16dn)$ with their center at v. P gets cut if and only if a sphere with a center in the shell defined by the two spheres gets chosen before a sphere with a center in the inner sphere. Let x_P be the probability of P being cut.

$$\begin{aligned} x_P &\leq \frac{(R + kR/(16dn))^d - (R - kR/(16dn))^d}{(R + kR/(16dn))^d} \\ &< (1 + \frac{k}{16dn})^d - (1 - \frac{k}{16dn})^d \end{aligned}$$

Since $(1+\alpha)^d - (1-\alpha)^d \leq (1+2\alpha)^d - 1$ for all $\alpha > 0$

$$\begin{aligned} x_P &\leq (1 + \frac{k}{8dn})^d - 1 \\ &\leq e^{k/(8n)} - 1 \\ &\leq k/(4n) \qquad [e^\alpha - 1 \leq 2\alpha \text{ for all } \alpha \leq 1] \end{aligned}$$

Summing over all maximal connected components of H, the probability of one or more connected components being cut is at most $1/4$. But if none of the connected components in H gets cut then no two vertices i and j with $c_{ij} \leq R/(8dn)$ belong to different clusters. Therefore, condition (2) gets violated with probability at most $1/4$.

Therefore both conditions are simultaneously satisfied with probability at least $1/2$. □

We obtain a deterministic version below.

Lemma 3.3 *A good partition can be found in deterministic polynomial time.*

We defer the proof of the above lemma to the full version and focus on its implications instead. We can use the partitioning guaranteed by the above lemma in the tree construction algorithm described in Subsection 2.2 to obtain the following theorem (the proof follows from the proofs of tree constructions in [5, 8, 18]). Corollary 3.5 then follows using the general framework described in Section 2.

Theorem 3.4 *There is a polynomial time deterministic algorithm that solves the minimum communication cost spanning tree problem on a graph embedded in $\Re_2^d$, and produces a tree whose cost is at most $O(\sqrt{d}\log n)$ times the lower bound L. In addition the distortion of any edge in the tree is $O(dn)$.*

Corollary 3.5 *A graph embedded in $\Re_p^d$ can be α-probabilistically approximated by a a distribution on $O(dn\log n)$ trees, where $\alpha = O(d^{1/p})$ if $1 \leq p \leq 2$ and $\alpha = O(d^{1-1/p})$ if $p \geq 2$. Such a distribution can be constructed in deterministic polynomial time.*

Acknowledgments

We would like to thank Piotr Indyk for several helpful suggestions, and for information about his recent lower bound [14] for the l_∞ norm.

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 J. Comput.*, 24:1, 78–100 (1995).

[2] B. Awerbuch, and Y. Azar. "Buy-at-bulk network design", *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pp. 542–547 (1997).

[3] B. Awerbuch, A. Bar-Noy, N. Linial, and D. Peleg. "Improved routing strategies with succinct tables", *Journal of Algorithms*, 11:3, 307–341 (1990).

[4] Y. Bartal. "Probabilistic approximation of metric spaces and its algorithmic applications", *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, pp. 184-93 (1996).

[5] Y. Bartal. "On approximating arbitrary metrics by tree metrics", *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, (1998).

[6] Y. Bartal, A. Blum, C. Burch, and A. Tomkins. "A polylog(n)-competitive algorithm for metrical task systems", *Proceedings of the 29th Annual Symposium on the Theory of Computing*, pages 711–719, (1997).

[7] M. Charikar, and B. Raghavachari. "The Finite Capacity Dial-a-Ride Problem", *these proceedings*.

[8] M. Charikar, C. Chekuri, A. Goel, and S. Guha. "Rounding via Trees: Deterministic Approximation Algorithms for Group Steiner Trees and k-median", *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, (1998).

[9] G. Even, J. Naor, S. Rao, and B. Schieber. "Divide-and-conquer approximation algorithms via spreading metrics", *Proceedings of IEEE 36th Symposium on Foundations of Computer Science*, 62–71 (1995).

[10] U. Feige. "Approximating the Bandwidth via Volume Respecting Embeddings", *Proceedings of the 30th Annual Symposium on the Theory of Computing*, (1998).

[11] M. R. Garey, and D. S. Johnson, "Computers and Intractability: A guide to the theory of NP-completeness", *Freeman, San Francisco* (1978).

[12] N. Garg, G. Konjevod, and R. Ravi. "A polylogarithmic approximation algorithm for the group Steiner tree problem", *Proceeding of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms* (1998).

[13] N. Garg, V. Vaziarni, and M. Yannakakis. "Approximate max-flow min-(multi)cut theorems and their applications", *Proceedings of 25th Annual ACM Symposium on the Theory of Computing*, 698–707 (1993). Also in *SIAM J. on Computing*, 25:2, 235–251 (1996).

[14] P. Indyk. *personal communication*, April 1998.

[15] W. Johnson and J. Lindenstrauss. "Extensions of Lipschitz Mappings into a Hilbert Space", *Contemporary Mathematics*, 26, 189-206 (1984).

[16] R. M. Karp. "A $2k$-competitive Algorithm for the Circle". *Manuscript*, August 1989.

[17] P. Klein, S. Plotkin, and S. Rao. "Excluded minors, network decomposition and multicommodity flows", *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing*, 682–690 (1993).

[18] G. Konjevod, R. Ravi, and F. S. Salman. "On approximating planar metrics by tree metrics", *manuscript* (1997).

[19] T. Leighton, and S. Rao. "An Approximate Max-Flow Min-Cut Theorem for Uniform Multicommodity Flow Problems with Application to Approximation Algorithms", *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science*, pages 422-431 (1988).

[20] N. Linial, E. London, and Y. Rabinovich. "The Geometry of Graphs and some of its Algorithmic Applications". *Proceedings of the 35th Annual IEEE Symposium of Foundations of Computer Science*, pages 577–591, (1994).

[21] M. Luby and N. Nisan. "A parallel approximation algorithm for positive linear programming", *Proceedings of 25th Annual Symposium on the Theory of Computing*, pages 448–457, (1993).

[22] D. Peleg, and E. Reshef. "Deterministic Polylog Approximation for Minimum Communication Spanning Tree", *25th International Colloquium on Automata, Languages, and Programming* (1998).

[23] D. Peleg, and E. Upfal. "A tradeoff between space and efficiency for routing tables", *JACM*, 36:3, 510–30 (1989).

[24] G. Pisier, "The Volume of Convex Bodies and Banach Space Geometry", *Cambridge University Press* (1989).

[25] S. Plotkin, D. Shmoys, and E. Tardos. "Fast approximation algorithms for fractional packing and covering problems", *Math of Oper. Research*, 20(2):257–301 (1995).

[26] Y. Rabinovich, and R. Raz. "Lower Bounds on the Distortion of Embedding Finite Metric Spaces in Graphs", *Discrete and Computational Geometry*, (1998).

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

[28] B. Y. Wu, G. Lancia, V. Bafna, K. Chao, R. Ravi, and C. Y. Tang. "A Polynomial Time Approximation Scheme for Minimum Routing Cost Spanning Trees", *Proceeding of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms* (1998).