Probabilistic Approximation of Metric Spaces and its Algorithmic Applications

Yair Bartal *

Abstract

The goal of approximating metric spaces by more simple metric spaces has led to the notion of graph spanners [PU89, PS89] and to low-distortion embeddings in low-dimensional spaces [LLR94], having many algorithmic applications.

This paper provides a *novel technique* for the analysis of randomized algorithms for optimization problems on metric spaces, by relating the randomized performance ratio for any metric space to the randomized performance ratio for a set of "simple" metric spaces.

We define a notion of a set of metric spaces that *probabilistically-approximates* another metric space.

We prove that any metric space can be probabilistically-approximated by *hierarchically well-separated trees* (HST) with a polylogarithmic distortion. These metric spaces are "simple" as being: (1) tree metrics. (2) natural for applying a divide-and-conquer algorithmic approach.

The technique presented is of particular interest in the context of *on-line computation*. A large number of on-line algorithmic problems, including metrical task systems [BLS87], server problems [MMS88] , distributed paging [BFR92], and dynamic storage rearrangement [FMRW95], are defined in terms of some metric space. Typically for these problems, there are linear lower bounds on the competitive ratio of deterministic algorithms. Although *randomization* against an *oblivious* adversary has the potential of overcoming these high ratios, very little progress has been made in the analysis.

We demonstrate the use of our technique by obtaining substantially improved results for two different on-line problems. For *metrical task systems* [BLS87] we give *first* sub-linear randomized competitive ratio for a large set of metric spaces. For *constrained file migration* [BFR92] we give *first* randomized algorithms for general networks with polylogarithmic competitive ratio.

*International Computer Science institute, 1947 Center Street, Berkeley CA 94704. *E-mail:* yairb@icsi.berkeley.edu. Research supported in part by the Rothschild Postdoctoral fellowship, NSF operating grants CCR-9304722 and NCR-9416101, ARPA/Army contract DABT63-93-C-0038, and ARPA/Air Force contract F19628-95-C-0137.

1 Introduction

1.1 Probabilistic Approximations of Metric Spaces

Many optimization problems can be defined in terms of some metric space (e.g., transportation/communication problems). Having fixed such a problem, and a set of desirable algorithms (e.g. polynomial-time, on-line), we are interested in the worst case performance ratio of an algorithm in the set compared to an optimal algorithm.

For such problems, typically, it is easier to get good bounds on the performance ratio for some classes of "simple" metric spaces. Suppose that for some class of metric spaces $\mathcal{S}$, we can obtain performance ratio β. Then, one way to get bounds for the performance ratio for metric spaces not in the class is to approximate such metric spaces by a metric space in $\mathcal{S}$.

Approximations of metric spaces by more simple metric spaces has been intensively studied in various areas of mathematics and computer science. We mention but a few. Johnson and Lindenstrauss [JL84] and Bourgain [Bour85] study embeddings in Hilbert spaces (from a perspective of functional analysis). Graham and Winkler [GW85] study embeddings in $\mathbb{Z}^d$ (from a graph theoretic motivation). Algorithmic applications in distributed computation and graph algorithms, have led to the notion of graph spanners, introduced by Peleg and Ullman [PU89] and later studied in many papers including [PS89, ADDJS90, CDNS92], and to low-distortion embeddings in low-dimensional real normed spaces by Linial, London and Rabinovich [LLR94].

We extend this idea to deal with randomized algorithms by defining a *probabilistic approximation* of metric spaces by a set of "simpler" metric spaces $\mathcal{S}$.

Given a metric space M over a finite set V, we consider sets $\mathcal{S}$ of "simple" metric spaces over V such that distances in M are dominated by the corresponding distances in metric spaces in $\mathcal{S}$. M is said to be α-*probabilistically-approximated* by $\mathcal{S}$, if there exists a probability distribution over $\mathcal{S}$ such that for *every* pair of nodes in V, the expected ratio between the distance in a metric space in $\mathcal{S}$ chosen from the probability distribution, and the distance in M is at most α. Moreover we are interested in efficiently computing such a probability distribution.

We deal with sets of metric spaces that cannot provide a good (*deterministic*) approximation for every metric space, but *can* provide a good probabilistic approximation.

Regarding a randomized algorithm as a distribution over deterministic algorithms, its performance ratio is the expected performance ratio of these algorithms over its own coin tosses. Our approach provides a *novel technique* for bounding the performance ratio of *randomized* algorithms for optimization problems on metric spaces with a cost function depending linearly on the distances in the metric space. If every metric space is α-probabilistically-approximated by $\mathcal{S}$, and the performance ratio of randomized algorithms for $\mathcal{S}$ is at most β, then the performance ratio of randomized algorithms for any metric space is at most $\alpha\beta$.

This technique is of particular interest in *on-line computation*. The performance ratio in this case is known as the *competitive ratio*, proposed by Sleator and Tarjan [ST85].

A large number of on-line algorithmic problems, including *metrical task systems* [BLS87], *server problems* [MMS88] , *distributed paging* [BFR92], and *dynamic storage rearrangement* [FMRW95], are defined in terms of metric spaces. Such problems address issues concerning with various fields such as resource management, communication networks, and dynamic data structures.

While the study of deterministic algorithms for these problems has been quite successful, the competitive ratio is typically linear. Although *randomization* against an *oblivious* adversary has the potential of overcoming these high ratios (cf. [BBKTW90, FKLMSY88]) very little progress has been made in the analysis other than for specific cases.

We focus on a particular class of metric spaces that probabilistically approximates every metric spaces with a polylogarithmic distortion. The main reason these metric spaces are "nice" is that they invite a divide-and-conquer approach to the solution of optimization problems.

We then demonstrate how this approach is useful by obtaining substantial improvements for some of the above mentioned problems.

1.2 Hierarchically Well-Separated Trees

The metric space approximation problem, of course, depends on what we take as our class of "nice" metric spaces.

One natural set to consider is: *tree metrics* (also called additive metrics). Indeed, intensive recent research has been concerned with approximating metric spaces by tree metrics. The *numerical taxonomy* problem [BG92, ABFNPT96] deals with such approximations. The main difference is that in numerical taxonomy it is assumed that the original metric is close to being additive, and so one expects approximations to within a very small error. Recently, the study of *tree spanners* of a graph has been initiated by Cai and Corneil [Cai92, CC95] considering approximating specific types of graphs by spanning trees.

However, there exist simple metric spaces for which every specific tree dominating the distances of the metric space has distortion $\Omega(n)$ (trivially matched by the upper bound given by MST in a weighted graph defined by the metric space). In particular this is true for the n-cycle as follows from the recent work of Rabinovich and Raz [RR95].

This leads to the notion of *probabilistic* metric approximations we propose.

Seminal work along these lines begun with Karp's observation [Karp89] that the n-cycle can be 2-probabilistically-approximated by an n-line, motivated by its application to the K-server problem. Alon, Karp, Peleg and West [AKPW91] have taken this idea forward in considering probabilistic-approximations of graphs by spanning trees. Their result implies that every metric space can be $2^{O(\sqrt{\log n \log\log n})}$-probabilistically-approximated by a tree metric.

We improve this result by presenting a set of tree metrics that provides a polylogarithmic probabilistic-approximation for any metric space. We also prove that even for the set of all tree metrics this bound cannot be lower than $\log n$.

Moreover we consider another desirable property of metric space that seems even more important for the design of algorithms.

The "*uniform*" metric space where all distances are equal is usually the most simple, and indeed, solutions for most problems we have mentioned are known for this case.

The property we define allows us to view a metric space M as a uniform metric space where each node is replaced by a sub-metric-space of diameter smaller by a factor of at least k (for some parameter $k > 1$) from the diameter of the complete metric space, and these properties hold recursively.

It is only natural to apply a divide-and-conquer approach to the solution of an optimization problem defined on M. I.e., give an upper bound for the problem on M under the assumption that we can obtain upper bounds for the problem in metric spaces of smaller diameter.

The key idea is that the large distance between sub-metric-spaces may provide a lower bound on an (adversarial) algorithm which is not restricted to a particular sub-metric-space.

The "simplicity" arising from the property above, is the reason that special cases of metric spaces falling into this category are those for which nearly-optimal randomized algorithms for *metrical task systems* are known (see more about that in the sequel).

A similar approach of using *spreading metrics* was used by Even, Naor, Rao and Schieber [ENRS95] to obtain approximation algorithms for a large set of optimization problems. However their method is deterministic and therefore does not apply for randomization against an *oblivious* adversary.

The *two* properties above are combined in a class of tree metric spaces we call *k-hierarchically well-separated trees* (k-HST) formally defined in Section 2.

Our main result states that for any k, every metric space M over a set V of size n can be α-probabilistically-approximated by k-HST's, where $\alpha = O(k \log n \log_k n)$. For particular metric spaces we get smaller α.

1.3 Probabilistic Partitions

The main construction proving the probabilistic metric proximity result is based on a tool that may be of independent interest. This tool is a *probabilistic partition* defined with respect to some distance parameter r. It is a distribution over partitions of the graph into clusters of diameter $O(r \log n)$ such that the probability that two close vertices in the graph are in different clusters is inversely proportional to r. This result can be viewed as an oblivious probabilistic version of low-diameter graph decompositions [AP90, LS91, ENRS95].

1.4 Applications to On-line Computation

As already pointed out, the probabilistic approximation of metric spaces is of particular importance in the case of *on-line* problems, where the randomization against *oblivious* adversaries (I.e., adversaries that cannot see the algorithm's coin tosses [BBKTW90]) is very powerful.

The most obvious example is the *paging* problem with a cache of size K, where the deterministic competitive ratio is K [ST85] while the randomized competitive ratio (against oblivious adversaries) is $\Theta(\log K)$ [FKLMSY88].

Many problems that are defined in terms of metric spaces (e.g., [BLS87, MMS88, BFR92, FMRW95]) are still almost entirely open in terms of their randomized competitive ratio (against an oblivious adversary).

Our approach implies that it suffices to give algorithms for k-HST's in order to obtain upper bounds for any metric space, with only polylogarithmic overhead.

We demonstrate our paradigm in several different examples. The first two give *first* sub-linear bounds for extensively studied on-line problems.

Metrical Task Systems. *Metrical task systems*, due to Borodin, Linial and Saks [BLS87], consist of a set of states forming a metric space and a set of tasks associated with costs in the different states. The goal is to move between states in order to minimize the total move and task costs. The problem has deterministic competitive ratio of $2n-1$ and the best randomized competitive ratio for general networks is $\frac{en-1}{e-1}$, due to Irani and Seiden [IS95].

The metrical task system problem is closely related to the the $(n-1)$-*server problem* [MMS88] on n points also called the *pursuit-evasion* game. Both upper and lower bounds for one problem translate to the other.

$\Theta(\log n)$ bounds for the *uniform* metric space were given in [FKLMSY88, BLS87]. A lower bound of $\Omega(\sqrt{\frac{\log n}{\log\log n}})$ was given in [BKRS93]. They also give $O(\log n)$ upper bounds in graphs that can be viewed as binary k-HST's for very large k. Recently, Blum, Furst and Tomkins [BFT96] have given an $O(\log n)$ upper bound for a special case of a 2-HST which is a line graph. Similar results have been independently given by Seiden [Seiden96]. Finally, Blum, Furst and Tomkins [BFT96] get $O(\log^2 n)$ competitive algorithm for the weighted cache by approximating it with a metric space that can also be viewed as a 2-HST.

Blum, Raghavan and Schieber [BRS91] give a competitive ratio of $2^{O(\sqrt{\log n \log\log n})}$ for the special case of equally spaced points on the line.

We give a randomized algorithm with competitive ratio of $2^{O(\sqrt{\log \Delta \log\log n})}$ (where the $\Delta \geq \log n$ is the diameter) against oblivious adversaries thus extending the [BRS91] result to a large set of metric spaces including those defined by any unweighted graph.

These results are described in Section 5.1.

Distributed Paging. The *constrained file migration* problem, due to Bartal, Fiat and Rabani [BFR92], is the problem of migrating files in a network with limited memory capacity at the processors in order to minimize file access and migration costs. This is a natural generalization of the uni-processor *paging* problem and a special case of the *distributed paging* problem [BFR92, ABF93b, ABF96] where files may also be replicated, but is of special interest since many systems allow only file migrations, to avoid issues of copy consistency (see [GS90] for a survey).

Let a network be represented by a weighted graph $G = (V, E, w)$. A set of files $\mathcal{F}$ resides in different nodes in the network. Processor v can accommodate in its local memory up to K_v files. The cost of an access to file F initiated by processor v is the distance from v to the processor holding the file F. A file may be migrated from one processor to another at a cost of D times the distance between the two processors. The goal is to minimize the total cost.

Let $m = \sum_{v \in V} K_v$ be the total memory in the network. Let $K = \max_{v \in V} K_v$. The constrained file migration problem has been first suggested in [BFR92]. We show that for some network the problem has an $\Omega(m)$ lower bound on the deterministic competitive ratio and an $\Omega(\log m)$ lower bound on the randomized competitive ratio, where m is the total aggregate memory in the network.

Albers and Koga [AK95] solve the problem in the uniform metric case giving a randomized upper bound of $O(\log K)$.

We give a randomized algorithm for k-HST's with competitive ratio $O(\log m)$, and thus get a randomized algorithm for general networks with competitive ratio of $O(\log m \log^2 n)$.

These results are described in Section 5.2.

The K-Server Problem. The *K-server problem* [MMS88] consists of K-servers in a metric space. Points are requested over time and a server must be moved to the request location. The K-server problem has been extensively studied. Koutsoupias and Papadimitriou [KP94] prove an upper bound of $2K-1$ on the competitive ratio of WFA (the work function algorithm). However WFA requires computing of optimal costs to end in every configuration and thus requires both time and space complexity of the order of $\binom{n}{K}$.

Therefore, it may be preferable to have a more efficient algorithm with a higher competitive ratio. By applying our method on the *memoryless* K-server algorithm for trees of Chrobak and Larmore [CL91] (an extension of the Double-Coverage algorithm for the line [CKPV90]) we obtain $O(K \log^2 n)$-competitive randomized algorithm for any metric space with time $O(K)$ per request and space $\tilde{O}(n+K)$.

Distributed Problems. Our method is also useful for the competitive analysis of *distributed* problems [BFR92] in networks. Again, the solution for the problem can be restricted to the solution for tree networks that represent k-HST's. This follows from the fact that the structure of the k-HST tree can be efficiently kept in the original network, for routing purposes, by using ITR (interval routing scheme), as described in [ABLP89, AP90], with only polylogarithmic memory per processor.

The *distributed K-server* problem was studied in [BR92], modeling distributed resource management. Bartal and Rosén give a distributed translator for (global-control) K-server algorithms. The competitive ratio of the distributed algorithm is a function of the number of memory states of the original K-server algorithm. From the discussion in the previous paragraph follows a distributed randomized K-server algorithm with only $O(\log^2 n)$ overhead.

We also obtain *first* randomized *distributed* on-line algorithms for the *generalized Steiner* problem and the *network-leasing* problem [AAB96], and improved algorithms for the *file allocation* problem [BFR92, ABF93a], all with competitive ratio $O(\log n \log(\min\{n, \Delta\}))$, by exploiting the simplicity of these problems for trees, and in particular for k-HST's. These results will be further discussed in the full version of this paper.

2 Definitions and Results

2.1 Probabilistic Approximations of Metric Spaces

Let V be a set of n points. We study the set of metric spaces defined over V. Given such a metric space M, the distance between u and v in V is denoted $d_M(u,v)$.

Definition 1 *A metric spaces N over V, dominates a metric space M over V, if for every $u, v \in V$, $d_N(u,v) \geq d_M(u,v)$.*

Definition 2 *A metric spaces N over V, α-approximates a metric space M over V, if it dominates M and for every $u, v \in V$, $d_N(u,v) \leq \alpha \cdot d_M(u,v)$.*

This paper is concerned with the following notion:

Definition 3 *A set of metric spaces $\mathcal{S}$ over V, α-probabilistically-approximates a metric space M over V, if every metric space in $\mathcal{S}$ dominates M and there exists a probability distribution over metric spaces $N \in \mathcal{S}$ such that for every $u, v \in V$, $\mathrm{E}(d_N(u,v)) \leq \alpha \cdot d_M(u,v)$.*

Consider an optimization problem $\mathcal{P}$ defined on metric spaces, where the cost of an algorithm for $\mathcal{P}$ is a linear combination of distances between vertices in the metric space.

The following gives the motivation for Definition 3.

Theorem 4 *Let $\mathcal{S}$ be a set of metric spaces over V that α-probabilistically-approximates M. If there exists a (randomized) algorithm for metric spaces in $\mathcal{S}$ with performance ratio β for $\mathcal{P}$ then there exists a randomized algorithm for M with performance ratio $\alpha\beta$ for $\mathcal{P}$. Moreover, if the distribution on $\mathcal{S}$ is efficiently computable then so is the algorithm for M.*

Proof. We define an algorithm A for M. The algorithm chooses at random a metric space $N \in \mathcal{S}$ according to the (computable) probability distribution, and then runs the algorithm A_N for N with performance ratio β. Consider an algorithm B for M, and define the algorithm B_N for N that behaves exactly the same as B.

Since $\mathcal{S}$ probabilistically-approximates M we have that for any input sequence σ, $\mathrm{E}(B_N(\sigma)) \leq \alpha \cdot B(\sigma)$, and that $\mathrm{E}(A(\sigma)) \leq \mathrm{E}(A_N(\sigma))$.

From the performance guarantee for A_N we obtain $\mathrm{E}(A(\sigma)) \leq \mathrm{E}(A_N(\sigma)) \leq \beta \cdot \mathrm{E}(B_N(\sigma)) \leq \alpha\beta \cdot B(\sigma)$. ■

2.2 Hierarchically Well-Separated Trees

Any finite metric space can be represented by a weighted connected graph (and vice-versa). For most of the technical part of the paper it will be convenient to use graph terminology. However we will use the definitions of the previous subsection for graphs with the obvious meaning.

For a weighted connected graph $G = (V, E, w)$, where $w(e)$ is the weight of edge e, $d_G(u, v)$ denotes the sum of edge weights over a shortest path between u and v. We assume w.l.o.g that the minimum edge weight in G is equal to 1. Let $\mathrm{diam}(G)$ denote the weighted diameter of the graph G; i.e., the maximum distance between a pair of vertices. When there is no confusion we will use $\Delta = \mathrm{diam}(G)$.

Definition 5 *A tree metric (or additive metric) over V is a metric space corresponding to a weighted tree spanning V.*

The following lower bound gives the justification for considering probabilistic metric approximations.

Theorem 6 ([RR95]) *If a tree metric α-(deterministically) approximates the n-cycle then $\alpha = \Omega(n)$.*

.

Definition 7 *A k-hierarchically well-separated tree (k-HST) is defined as a rooted weighted tree with the following properties:*

- *The edge weight from any node to each of its children is the same.*
- *The edge weights along any path from the root to a leaf are decreasing by a factor of at least k.*

The height of a tree T is the maximum number of edges on a path from the root to a leaf, denoted $h(T)$. Note that a k-HST, T, has height $h(T) \leq \log_k(\mathrm{diam}(T))$.

Our main result is

Theorem 8 *Every weighted connected graph G can be α-probabilistically-approximated by the set of k-HST's of diameter $\mathrm{diam}(T) = O(\mathrm{diam}(G))$, where $\alpha = O(k \log n \log_k(\min\{n, \Delta\}))$.*

If G is a tree then $\alpha = O(k \log_k(\min\{n, \Delta\}))$. If G represents the d-dimensional mesh with the l_∞ metric then $\alpha = O(k \log_k n/d)$.

In the appendix we give a somewhat stronger result for meshes, adding information about the structure of the k-HST's, based on a simpler technique than that of our main construction.

These results are complemented with the following lower bound (proof in appendix).

Theorem 9 *There exists a connected graph G such that if G is α-probabilistically-approximated by tree metrics then $\alpha = \Omega(\log n)$.*

2.3 Probabilistic Partitions and HST's

We start by describing a partitioning problem. Let $G = (V, E, w)$ be a weighted connected graph.

Definition 10 *An ℓ-partition of G is a collection of subsets of vertices $P = \{V_1, V_2, \ldots V_s\}$ such that:*

- *For all $1 \leq i \leq s$, $V_i \subseteq V$ and $\bigcup_{1 \leq i \leq s} V_i = V$.*
- *For all $1 \leq i, j \leq s$ such that $i \neq j$, $V_i \cap V_j = \emptyset$.*
- *Let $C(V_i)$ denote the subgraph of G induced by V_i. Such a subgraph is also called a cluster of the partition. For every $1 \leq i \leq s$, the strong diameter of V_i is bounded as: $\mathrm{diam}(C(V_i)) \leq \ell$.*

Similarly, one can define ℓ-weak partition by replacing the third requirement with a bound on the weak diameter of V_i (i.e., the maximal distance in G between vertices of V_i).

Let $\mathcal{D}$ be a probability distribution over the set of ℓ-partitions P of G.

Given such a probability distribution and consider some edge $e = (u, v) \in E$; let $x(e)$ denote the probability under $\mathcal{D}$ that u and v belong to different clusters. i.e., the probability that for any cluster C of P, $e \notin E(C)$.

For a parameter $0 \leq r \leq \Delta$, we define:

Definition 11 *An (r, ρ, λ)-probabilistic partition of G is a probability distribution $\mathcal{D}$, over the set of $(r\rho)$-partitions P of G, such that*

- $\max_{e \in E}\{x(e) \cdot \frac{r}{w(e)}\} \leq \lambda$.

The probabilistic partition is ϵ-forcing if the following holds:

- *For every $e \in E$, if $\frac{w(e)}{r} \leq \epsilon$ then $x(e) = 0$.*

We note that the notion of ϵ-forcing probabilistic partition is useful only in the case that the graph has non-equal edge weights (Note that any probabilistic partition is $\frac{1}{\Delta}$-forcing.).

We can also speak about *weak* probabilistic partitions similarly defined with respect to *weak* partitions.

Let $x(u,v)$ denote the probability that for two (not necessarily adjacent) vertices u and v, the shortest path between them is not contained in a single cluster. Notice that the probability that u and v belong to different clusters is bounded above by $x(u,v)$. Our definition of a probabilistic partition appears more natural by noting the following:

Fact 12 *For a weighted graph G and an (r,ρ,λ)-probabilistic partition P of G:*

- $\max_{u \neq v \in V} \{x(u,v) \cdot \frac{r}{d(u,v)}\} \leq \lambda$.

If P is ϵ-forcing then the following holds:

- *For every $u, v \in V$, if $\frac{d(u,v)}{r} \leq \epsilon$ then $x(u,v) = 0$.*

Proof. For the first part assume $r > 0$. Let the shortest path between u and v be composed of edges $e_0, e_1, \ldots e_l$ where $e_i = (a_i, a_{i+1})$ and $a_0 = u, a_{l+1} = v$. The probability $x(u,v)$ is the probability that for at least one edge e_i the endpoints a_i and a_{i+1} belong to different clusters. Therefore: $x(u,v) \leq \sum_{0 \leq i \leq l} x(e_i) \leq \sum_{0 \leq i \leq l} \lambda \frac{w(e_i)}{r} \leq \lambda \frac{d(u,v)}{r}$. The second part is obvious from Definition 14. ■

The following theorem explains how probabilistic partitions are related to our original problem:

Theorem 13 *Given weighted graph G. If there exist (r,ρ,λ)-probabilistic partitions of any subgraph of G, then G is α-probabilistically-approximated by the set of k-HST's of diameter $\mathrm{diam}(T) = O(\mathrm{diam}(G))$, where $\alpha = O(\rho\lambda k \cdot \log_k \Delta)$. Moreover if the probabilistic partitions can be efficiently computed so can be the distribution over HST's.*

If the probabilistic partitions are ϵ-forcing then $\alpha = O(\rho\lambda k \cdot \log_k(\min\{\Delta, \frac{\rho}{\epsilon}\}))$.

It remains to show how to construct good probabilistic partitions:

Theorem 14 *Given a weighted graph G. For any $1 \leq r \leq \Delta$ there exists a $\frac{1}{n}$-forcing $(r, 2\ln n + 1, 2)$-probabilistic partition of G. Moreover the distributions can be computed in $O(|E|)$ time.*

If either G is a weighted tree or that G represents the d-dimensional mesh with the l_∞ norm then there exists a $\frac{1}{n}$-forcing $(r,1,1)$-probabilistic partition of G.

The proofs of the claims about trees and meshes are omitted from this version of the paper.

Combining Theorem 14 with Theorem 13 gives our main result Theorem 8.

We complement Theorem 14 with a matching lower bound (proof in appendix).

Theorem 15 *Given a weighted graph G. If for any $1 \leq r \leq \Delta$ there exists an (r,ρ,λ)-weak probabilistic partition of G then $\rho\lambda = \Omega(\log n)$.*

The rest of the paper contains the proof for the main results described above and details of the applications.

3 Constructing Probabilistic Partitions

In this section we give the proof of Theorem 14 for general graphs. We first show how one can make the probabilistic partition to be $\epsilon = \frac{1}{n}$-forcing. We start by greedily contracting edges of weight $w(e) \leq \frac{r}{n}$ by joining their endpoints into a single vertex. The resulting graph has edge weights higher than $\frac{r}{n}$. We will show how to construct probabilistic partitions for such graphs with diameter at most $2r \ln n$ for every cluster. After constructing such a partition we will insert the low weight edges back thereby increasing the diameter of the clusters by at most $n \cdot \frac{r}{n} = r$. This would give the claimed bound.

For a graph H, let $B_z^H(v) = \{u \in V(H) | d(v,u) \leq z\}$ denote the ball of radius z around v.

In what follows we describe the construction of the probabilistic partition. The construction proceeds in stages: At each stage we construct one cluster of the partition. With the t'th stage there is associated a subgraph H_t of G induced by a vertex set $U_t \subseteq V$. At the beginning $U_0 = V$ and $H_0 = G$.

The Algorithm.

Repeat the following process until $U_t = \emptyset$.

Let v_t be an arbitrary node in U_t. If the connected component of H_t including v_t has diameter at most $2r \ln n$ then define it to be the next cluster, C_t, of the partition.

Otherwise proceed as follows. Consider the real interval $I = [0, r \ln n)$. Now choose a radius z at random from I according to the probability distribution $p(z) = (\frac{n}{n-1})\frac{1}{r}e^{-\frac{z}{r}}$ (This is indeed a probability distribution for I. We note that this continuous distribution is only defined for the sake of esthetics of the proof. However it is enough to consider a discrete distribution with only $O(\log n)$ random bits.)

We define the subgraph induced by $B_z^{H_t}(v_t)$ to be the next cluster, C_t, of the partition.

Finally, set $U_{t+1} = U_t \setminus V(C_t)$. Let H_{t+1} be the subgraph of G induced by U_{t+1} and proceed with the next stage.

Analysis.

The bound on the radius (ρ) of the partition is obvious from the description.

Consider some edge $e = (u, w)$. Our goal is to bound the probability that e is included in none of the clusters.

Fix the stage t. Let $v = v_t$ be the vertex chosen and let z be the random radius chosen at the t'th stage. We define events in the construction with respect to the edge $e = (u, w)$. Assume w.l.o.g that $d(v, u) \leq d(v, w)$. Let $\tilde{d}(x, y) = \min\{d(x, y), r \ln n\}$ (We note that triangle inequality is preserved.)

- A_t: The event that $u, w \in U_t$.
- M_t^{I}: The event $\tilde{d}(v, w) \leq z$, conditional on A_t.
- M_t^{X}: The event $\tilde{d}(v, u) \leq z < \tilde{d}(v, w)$, conditional on A_t.
- M_t^{N}: be the event $z < \tilde{d}(v, u)$, conditional on A_t.
- X_t: The event that (u, w) is in none of the clusters C_j for $j \geq t$, conditional on A_t.

Note that $\Pr(A_0) = 1$. Our goal is to give an upper bound on $x(e) = \Pr(X_0)$.

The event M_t^{I} corresponds to the case that the edge (u, w) is *included* in C_t. The event M_t^{X} corresponds to the case that u is included C_t but w is not and therefore the edge (u, w) will *excluded* from any cluster in the partition. The event M_t^{N} corresponds to the case that both u and w are not included in C_t and are therefore passed onto the *next* stage.

We have the following recursion relation:

$$\Pr(X_t) \quad = \quad \Pr(M_t^X) + \Pr(M_t^{\mathrm{N}}) \Pr(X_{t+1}). \tag{1}$$

We can bound the probabilities above as

$$\begin{aligned}
\Pr(M_t^X) &= \int_{\tilde{d}(v,u)}^{\tilde{d}(v,w)} p(z)dz \\
&= \left(\frac{n}{n-1}\right)(1 - e^{-\frac{\tilde{d}(v,w)-\tilde{d}(v,u)}{r}})e^{-\frac{\tilde{d}(v,u)}{r}} \\
&\le \left(\frac{n}{n-1}\right)\frac{\tilde{d}(u,w)}{r}e^{-\frac{\tilde{d}(v,u)}{r}},
\end{aligned} \tag{2}$$

by using $1 - e^{-x} \le x$ and $\tilde{d}(v,w) - \tilde{d}(v,u) \le \tilde{d}(u,w)$. We also have

$$\begin{aligned}
\Pr(M_t^{\mathrm{N}}) &= \int_0^{\tilde{d}(v,u)} p(z)dz \\
&= \left(\frac{n}{n-1}\right)(1 - e^{-\frac{\tilde{d}(v,u)}{r}}).
\end{aligned} \tag{3}$$

We prove that

$$\Pr(X_t) \quad \le \quad \left(2 - \frac{t}{n-1}\right)\frac{\tilde{d}(u,w)}{r}. \tag{4}$$

If some $t < n$ is the last step then $\Pr(X_t) = 0$ and the bound is trivial. Assume it is true for $t+1$. The proof follows by induction from bounds 1,2, and 3:

$$\begin{aligned}
\Pr(X_t) &= \Pr(M_t^X) + \Pr(M_t^{\mathrm{N}}) \Pr(X_{t+1}) \\
&\le \left(\frac{n}{n-1}\right)\frac{\tilde{d}(u,w)}{r}e^{-\frac{\tilde{d}(v,u)}{r}} + \left(\frac{n}{n-1}\right)(1 - e^{-\frac{\tilde{d}(v,u)}{r}})\left(2 - \frac{t+1}{n}\right)\frac{\tilde{d}(u,w)}{r} \\
&= \left(\frac{n}{n-1}\right)\left[1 + \frac{n-t-2}{n-1}(1 - e^{-\frac{\tilde{d}(v,u)}{r}})\right]\frac{\tilde{d}(u,w)}{r}.
\end{aligned}$$

Since $e^{-\frac{\tilde{d}(v,u)}{r}} \ge \frac{1}{n}$ we get

$$\begin{aligned}
\Pr(X_t) &\le \left(\frac{n}{n-1}\right)\left[\left(2 - \frac{t+1}{n-1}\right) - \frac{1}{n}\left(1 - \frac{t+1}{n-1}\right)\right]\frac{\tilde{d}(u,w)}{r} \\
&\le \left(2 - \frac{t}{n-1}\right)\frac{\tilde{d}(u,w)}{r}.
\end{aligned}$$

We thus get $\Pr(X_0) \le 2\frac{\tilde{d}(u,w)}{r}$ which concludes the proof. ■

4 Probabilistic Approximations by HST's

In this section we prove Theorem 13. The k-HST construction is done recursively. Let $G_1 = G$. With every stage $i \ge 1$ of the recursion there is associated a graph G_i and a parameter r_i. We set $r_i = \lfloor \mathrm{diam}(G_i)/\rho k \rfloor$ for $1 \le i \le t$, where t is the depth of the recursion.

At the i'th stage we compute an (r_i, ρ, λ)-probabilistic partition of the graph G_i. This results with a partition P of G_i (chosen from some distribution). Let the clusters of this partition be $C_1, C_2, \ldots C_s$. We now recursively compute the k-HST's for each of these clusters (setting $G_{i+1} = C_j$ for $j = 1, 2, \ldots s$). Let the computed trees be $T^1_{i+1}, T^2_{i+1}, \ldots, T^s_{i+1}$ with corresponding roots $q^1_{i+1}, q^2_{i+1}, \ldots, q^s_{i+1}$.

We now construct the k-HST, T_i, for G_i by adding one more node q_i as the root of the tree and attaching all trees $T^1_{i+1}, T^2_{i+1}, \ldots, T^s_{i+1}$ to it by letting $q^1_{i+1}, q^2_{i+1}, \ldots, q^s_{i+1}$ be the children of q_i. The weight on the edges from q_i to each of its children is set to $\frac{1}{2}\mathrm{diam}(G_i)$.

The last stage in the recursive process stops at depth t at which G_t consists of a single vertex.

Observe that

$$\mathrm{diam}(G_{i+1}) \quad \leq \quad \rho r_i \quad \leq \quad \rho \cdot \frac{\mathrm{diam}(G_i)}{\rho k} \quad = \quad \frac{1}{k}\mathrm{diam}(G_i).$$

It follows that the tree constructed in the process described is indeed a k-HST.

In addition we get $\mathrm{diam}(G_i) \leq \Delta / k^{i-1}$. Hence we obtain a bound on the depth of the recursion: $t \leq \log_k \Delta$.

We first check that in the process we have not created a tree with much bigger diameter than that of the original graph. We do that by giving a bound on the maximum length path from a root to a leaf in the k-HST constructed at the i'th stage, denoted $\gamma(T_i)$. We prove by induction that $\gamma(T_i) \leq \frac{1}{2}(1 + \frac{1}{k-1})\mathrm{diam}(G_i)$. At the last stage, the clusters of the partition consist of only a single vertex and so do the corresponding computed trees and therefore the claim trivially holds. We assume the claim holds for $i+1$ and prove it for i:

$$\begin{aligned}
\gamma(T_i) \quad &= \quad \max_{1 \leq j \leq s} \gamma(T^j_{i+1}) + \frac{1}{2}\mathrm{diam}(G_i) \leq \max_{1 \leq j \leq s} \frac{1}{2}(1 - \frac{1}{k-1})\mathrm{diam}(C^j_{i+1})\} + \frac{1}{2}\mathrm{diam}(G_i) \\
&\leq \quad \frac{1}{2}(1 + \frac{1}{k-1}) \cdot \frac{1}{k}\mathrm{diam}(G_i) + \frac{1}{2}\mathrm{diam}(G_i) = \frac{1}{2}(1 + \frac{1}{k-1})\mathrm{diam}(G_i).
\end{aligned}$$

This implies that $\mathrm{diam}(T_i) \leq 2 \cdot \gamma(T_i) \leq (1 + \frac{1}{k-1})\mathrm{diam}(G_i)$.

We now turn to bound the ratio between the distances in the tree and in the original graph. First, it immediately follows from the construction that for any pair of vertices u, v in G_i, $d_{T_i}(u,v) \geq d_{G_i}(u,v)$. The proof is by induction. It is obvious when T_i and G_i both consist of a single vertex. Assume the claim holds for $i+1$. If there exists $1 \leq j \leq s$ such that u and v belong to the same cluster C^j_{i+1} then by induction $d_{T_i}(u,v) = d_{T^j_{i+1}}(u,v) \geq d_{C^j_{i+1}}(u,v) \geq d_{G_i}(u,v)$. Otherwise u and v belong to different clusters. Hence,

$$\begin{aligned}
d_{T_i}(u,v) \quad &\geq \quad 2 \cdot \frac{1}{2}\mathrm{diam}(G_i) \\
&= \quad \mathrm{diam}(G_i) \qquad \geq \quad d_{G_i}(u,v).
\end{aligned}$$

Finally we give an upper bounds on $\frac{\mathrm{E}(d_{T_i}(u,v))}{d_{G_i}(u,v)}$.

We prove by induction that for some value $h \leq t$, $\mathrm{E}(d_{T_i}(u,v)) \leq \lambda \rho k (1 + \frac{1}{k-1})(h - i) \cdot d_{G_i}(u,v)$. Assume the claim holds for $i+1$. If the shortest path between u and v is contained in some single cluster C^j_{i+1} we get by induction

$$\begin{aligned}
\mathrm{E}(d_{T^j_{i+1}}(u,v)) \quad &\leq \quad \lambda \rho k (1 + \frac{1}{k-1})(h - i - 1) \cdot d_{C^j_{i+1}}(u,v) \\
&= \quad \lambda \rho k (1 + \frac{1}{k-1})(h - i - 1) \cdot d_{G_i}(u,v).
\end{aligned}$$

Let $x_i(u,v)$ denote the probability that the shortest path between u and v is not contained in any single cluster. Using Fact 12 we obtain

$$\mathrm{E}(d_{T_i}(u,v)) \quad \leq \quad (1 - x_i(u,v)) \cdot \lambda \rho k (1 + \frac{1}{k-1})(h - i - 1) \cdot d_{G_i}(u,v) + x_i(u,v) \cdot \mathrm{diam}(T_i)$$

$$
\begin{aligned}
&\le \lambda\rho k(1+\tfrac{1}{k-1})(h-i-1)\cdot d_{G_i}(u,v) + \lambda\tfrac{d_{G_i}(u,v)}{r_i}\cdot(1+\tfrac{1}{k-1})\mathrm{diam}(G_i)\\
&\le \lambda\rho k(1+\tfrac{1}{k-1})(h-i-1)\cdot d_{G_i}(u,v) + \lambda\rho k(1+\tfrac{1}{k-1})d_{G_i}(u,v)\\
&\le \lambda\rho k(1+\tfrac{1}{k-1})(h-i)d_{G_i}(u,v).
\end{aligned}
$$

Since $h \le t \le \log_k \Delta$ we obtain the $O(\lambda\rho k \log_k \Delta)$ upper bound.

Now, assume the probabilistic partitions are ϵ-forcing. Let l be the smallest such that $\frac{d_{G_l}(u,v)}{r_l} > \epsilon$. Recall that for every $i < l$ we have $x_i(u,v) = 0$. Thus we have for all $i < l$, the distance between u and v in T_i is equal to their distance in the $i+1$ level tree, and thus by induction we can obtain $\mathrm{E}(d_{T_i}(u,v)) \le \lambda\rho k(1+\frac{1}{k-1})(h-l)\cdot d_{G_i}(u,v)$.

Next consider the smallest value for $i > l$ such that both u and v belong to G_i. Obviously $\mathrm{diam}(G_i) \ge d_{G_i}(u,v) \ge d_{G_l}(u,v)$. We therefore obtain

$$
\frac{\mathrm{diam}(G_i)}{\mathrm{diam}(G_l)} \ge \frac{d_{G_l}(u,v)}{\rho k \cdot r_l} > \frac{\epsilon}{\rho k}.
$$

On the other hand $\mathrm{diam}(G_i) \le \mathrm{diam}(G_l)/k^{i-l}$ and thus we obtain $i - l - 1 \le \log_k(\min\{\Delta, \frac{\rho}{\epsilon}\})$. Hence, by setting an appropriate value for h we obtain the claimed upper bound of $O(\rho\lambda k \cdot \log_k(\min\{\Delta, \frac{\rho}{\epsilon}\}))$. ■

5 Applications

This sections presents the randomized algorithms for *metrical task systems* and *constrained file migration* based on the probabilistic metric approximations by HST's.

Since HST's are "almost uniform" between subtrees it is just natural to consider applying algorithms for a uniform case recursively.

Both of our algorithms run recursively copies of the randomized *marking* algorithm of [FKLMSY88] for the K-server problem on a uniform metric space (or equivalently for the paging problem). The marking algorithm works in phases. At the beginning of a phase all servers are unmarked. When a request arrives, if there is a server at the requested point it is just marked. Otherwise an unmarked server is chosen uniformly at random and then marked. If all servers are marked a new phase begins. The marking algorithm is $O(\log K)$-competitive.

5.1 Metrical Task Systems

In this section we give sub-linear upper bounds for *metrical task systems*. We give the upper bound for the $(n-1)$-server problem on n points. This can be then translated into an upper bound for metrical task systems.

Theorem 16 *Given a k-HST, T, there exists a $2^{O(h(T)\log\log n)}$ competitive algorithm for the $(n-1)$-server problem on T.*

Proof Sketch. Apply marking algorithms recursively in each level, considering each subtree as a vertex. The location of the "hole" is defined by the subtree not completely covered by servers. Phases of the subtrees marking algorithms are repeated until the optimal cost at a certain subtree reaches the distance to the parent vertex, and then the higher level marking algorithm is invoked with a request at the vertex corresponding to the appropriate subtree. Since the marking algorithm is $O(\log n)$-competitive we get that the competitive ratio increases by a factor of $O(\log n)$ at each level of the recursion. Hence the bound follows. ■

Assume $\Delta \geq \log n$. By using Theorem 8 with $k = 2^{\sqrt{\log \Delta \log \log n}}$ we get that $h(T) = O(\log_k \Delta) = O(\sqrt{\frac{\log \Delta}{\log \log n}})$. From Theorem 4 we obtain the following:

Corollary 17 *Given a weighted graph G of diameter $\Delta \geq \log n$, there exists a $2^{O(\sqrt{\log \Delta \log \log n})}$-competitive algorithm for the metrical task systems problem on G.*

5.2 Distributed Paging

We first state lower bounds for the problem.

Theorem 18 *There exist networks on which the constrained file migration problem has deterministic competitive ratio $\Omega(m)$ and randomized competitive ratio $\Omega(\log m)$.*

Theorem 19 *Given a network represented by a k-HST, T, for some $k \geq 2$ then there exists a randomized $O(\log m)$ competitive algorithm for the constrained file migration problem on G.*

Proof Sketch. We can concentrate on the case $D = 1$ (as follows from the general reductions in [AAB96, BCI96]). Apply a copy of the marking algorithm for each subtree as follows. Given a request for a file invoke the algorithm at the topmost level containing the request, where the file is missing, and then repeat until the file arrives at the requesting processor. Now consider a marking algorithm for a particular subtree. Its cache size is at most m. The virtual configuration of the algorithm contains some files that may actually be outside its subtree. Since a file in the virtual configuration has been requested inside the subtree in either the current or the previous marking phases, this may happen only if the file has been later requested outside the subtree. Thus these two requests have an inherent cost proportional to the cost of migrating the file into the subtree. The claim follows since the expected cost on requests for files not in the virtual configuration of the algorithm is bounded by $O(\log m)$ times the optimal cost. ■

From Theorems 8 and 4 we get

Corollary 20 *Given a network G, there exists a randomized $O(\log m \log n \log(\min\{n, \Delta\}))$ competitive algorithm for the constrained file migration problem on G.*

6 Conclusions and Open Problems

This paper presents a method for exploiting the *power of randomization* against an oblivious adversary in the context of metric spaces related problems.

We introduce the notion of probabilistically approximating metric spaces by some restricted set of metric spaces. It is interesting to investigate the power of *probabilistic metric-space approximation* with respect to different sets of "nice" metric spaces other than the ones discussed here.

The concept of *hierarchically well-separated trees* (HST's) seems quite fundamental in the sense that the solution to many problems becomes considerably simpler. Our main result shows that with the use of randomization every metric space can be approximated by HST's up to a polylogarithmic factor, thus restricting the attention to the solution of such problems on HST's.

An obvious open problem is to close the gap between the $O(\log^2 n)$ upper bound and $\Omega(\log n)$ lower bound for the HST's approximation factor.

Our result is of special interest when dealing with incomplete information, where randomization is often very powerful, such as in *on-line computation*. Perhaps the most exciting avenue of research is obtaining polylogarithmic competitive ratios for *metrical task systems*, a long-standing open problem [BLS87]. We hope that this work will eventually lead to the resolution of this problem.

Viewing a distributed network as a HST makes another step in simplifying the exploitation of locality of reference in a *distributed environment*. While constructions such as [AP90, LS91] allow to use graph decompositions for similar purposes, the properties of HST's make our (randomized) construction more powerful and more simple to use. The tool of a probabilistic partition which is in the heart of the HST's construction may be of independent interest.

Other problems mentioned in this paper serve as natural candidates for applications of our technique. It remains to see what other applications and application areas the results presented here may have.

Acknowledgments

The author would like to thank Avrim Blum, Eli Gafni, Sandy Irani, Stefano Leonardi, Christos Papadimitriou, Satish Rao and Orli Waarts for helpful discussions. Special thanks to Mike Saks and Danny Raz for many helpful comments.

References

[AAB96] B. Awerbuch, Y. Azar, and Y. Bartal. On-line Generalized Steiner Problem. In *Proc. of the 7th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 68–74, January 1996.

[ABF93a] B. Awerbuch, Y. Bartal, and A. Fiat. Competitive Distributed File Allocation. In *Proc. 25th ACM Symp. on Theory of Computing*, pages 164–173, May 1993.

[ABF93b] B. Awerbuch, Y. Bartal, and A. Fiat. Heat & Dump: Randomized competitive distributed paging. In *Proc. 34rd IEEE Symp. on Foundations of Computer Science*. IEEE, pages 22–31,November 1993.

[ABF96] B. Awerbuch, Y. Bartal, and A. Fiat. Distributed Paging for General Networks. In *Proc. of the 7th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 574–583, January 1996.

[ABFNPT96] R. Agarwala, V. Bafna, M. Farach, B. Narayanan, M. Paterson, and M. Thotup. On the Approximability of Numerical Taxonomy. In *Proc. of the 7th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 365–372, January 1996.

[ABLP89] B. Awerbuch, A. Bar-Noy, N. Linial, and D. Peleg. Compact Distributed Data Structures for Adaptive Network Routing. In *Proc. 21st ACM Symp. on Theory of Computing*, pages 230–240, May 1989.

[ADDJS90] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares. On Sparse Spanners of Weighted Graphs. In *Discrete and Computational Geometry*.

[AK95] S. Albers and H. Koga. Page Migration with Limited Local Memory Capacity. In *Proc. of the 4th Workshop on Algorithms and Data Structures*, , August 1995.

[AKPW91] N. Alon, R.M. Karp, D. Peleg, and D.B. West. Graph-Theoretic Game, and its Applications to the k-Server Problem. To appear in *SIAM J. on Computing*.

[AP90] Baruch Awerbuch and David Peleg. Sparse partitions. In *Proc. 31st IEEE Symp. on Foundations of Computer Science*, pages 503–513, 1990.

[Boll78] B. Bollobás. *Extremal Graph Theory*. Academic Press, London, England, 1978.

[Bour85] J. Bourgain. On Lipshitz Embedding of Finite Metric Spaces in Hilbert Space. In *Israel J. Math.* 52, pages 46–52, 1985.

[BBKTW90] S. Ben-David, A. Borodin, R.M. Karp, G. Tardos, and A. Wigderson. On the Power of Randomization in Online Algorithms. In *Proc. of the 22nd Ann. ACM Symp. on Theory of Computing*, pages 379–386, May 1990.

[BCI96] Y. Bartal, M. Charikar and P. Indyk. On Page Migration and Other Relaxed Task Systems. Manuscript, June 1996.

[BFR92] Y. Bartal, A. Fiat, and Y. Rabani. Competitive algorithms for distributed data management. In *Proc. 24th ACM Symp. on Theory of Computing*, pages 39–50, 1992. Also accepted for publication in the STOC 92 special issue *Journal of Computer and Systems Sciences*, 1993.

[BFT96] A. Blum, M. Furst, and A. Tomkins. What To Do With You Free Time: On-line Algorithms for Infrequent Requests. Manuscript, 1996.

[BG92] J-P. Barthélemy and A. Guénoche. *Trees and Proximity Representations*. Wiley, New York, 1991.

[BKRS93] A. Blum, H. Karloff, Y. Rabani, and M. Saks. A Decomposition Theorem and Bounds for Randomized Server Problems. In *Proc. of the 33rd Ann. IEEE Symp. on Foundations of Computer Science*, pages 197–207, October 1992.

[BLS87] A. Borodin, N. Linial, and M. Saks. An Optimal On-Line Algorithm for Metrical Task Systems. In *Proc. of the 19th Ann. ACM Symp on Theory of Computing*, pages 373–382, May 1987.

[BR92] Y. Bartal and A. Rosén. The Distributed k-Server Problem — A Competitive Distributed Translator for k-Server Algorithms. In *Proc. of the 33rd Ann. IEEE Symp. on Foundations of Computer Science*, pages 344-353, October 1992.

[BRS91] A. Blum, P. Raghavan, and B. Schieber. Navigating in Unfamiliar Geometric Terrain. In *Proc. of the 23rd Ann. ACM Symp. on Theory of Computing*, May 1991.

[BS89] D.L. Black and D.D. Sleator. Competitive Algorithms for Replication and Migration Problems. Technical Report CMU-CS-89-201, Department of Computer Science, Carnegie-Mellon University, 1989.

[Cai92] L. Cai. *Tree Spanners: Spanning trees that Approximate Distances.* Ph.D Thesis. Tech. Report 260/92, Univ. of Toronto, May 1992.

[CC95] L. Cai and D.G. Corneil. Tree Spanners. In *SIAM J. on Discrete Math.*, 8(3), pages 359–387, 1995.

[CDNS92] B. Chandra, G. Das, G. Narasimhan, and J. Soares. New Sparseness Results on Graph Spanners. Manuscript, 1996.

[CKPV90] M. Chrobak, H. Karloff, T. Payne, and S. Vishwanathan. New Results on Server Problems. In *Proc. 1st ACM-SIAM Symp. on Discrete Algorithms*, pages 291–300, January 1990.

[CL91] M. Chrobak and L. Larmore. An Optimal On-Line k-Server Algorithm for Trees. *SIAM J. of Computing*, 20:144-148, 1991.

[ENRS95] G. Even, J. Naor, S. Rao, and B. Scheiber. Divide-and-Conquer Approximation Algorithms via Spreading Metrics. In *Proc. of the 36th Ann. IEEE Symp. on Foundations of Computer Science*, pages 62–71, October 1995.

[FKLMSY88] A. Fiat, R.M. Karp, M. Luby, L.A. McGeoch, D.d. Sleator , and N.E. Young. Competitive Paging Algorithms. Technical Report, Carnegie Mellon University, 1988.

[FMRW95] A. Fiat, Y. Mansour, A. Rosén, and O. Waarts. Competitive Access Time via Dynamic Storage Rearrangement. In *Proc. of the 36th Ann. IEEE Symp. on Foundations of Computer Science*, pages 392–403, October 1995.

[GS90] B. Gavish and O.R.L. Sheng. Dynamic File Migration in Distributed Computer Systems. In *Comm. of the ACM*, 33(2):177-189, 1990.

[GW85] R.L. Graham and P.M. Winkler. On Isometric Embeddings of Graphs. In *Trans. of the Amer. Math. Soc.* 288, pages 527–536, 1985.

[IS95] S. Irani and S. Seiden. Randomized Algorithms for Metrical Task Systems. In *Proc. of the 4th Int. Workshop on Algorithms and Data Structures*, LNCS 955, pages 159–170, 1995.

[JL84] W.B. Johnson and J. Lindenstrauss. Extensions of of Lipschitz Mappings into a Hilbert Space. In *Contemporary Methematics* 26, pages 189–206, 1984.

[Karp89] R.M. Karp. A $2k$-competitive Algorithm for the Circle. Manuscript, August 1989.

[KP94] E. Koutsoupias, and C. Papadimitriou. On the k-Server Conjecture. In *Proc. 26th Annual ACM Symp. on Theory of Computing*, pages 507–511, 1994.

[LLR94] N. Linial, E. London, and Y. Rabinovich. The Geometry of Graphs and Some of its Algorithmic Applications. In *Proc. of the 35th Ann. IEEE Symp. on Foundations of Computer Science*, pages 577–591, October 1994.

[LS91] N. Linial and M. Saks. Decomposing Graphs into Regions of Small Diameter. In *Proc. of the 2th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 320–330, January 1991.

[MMS88] M.S. Manasse, L.A. McGeoch, and D.D. Sleator. Competitive Algorithms for On-Line Problems. In *Proc. of the 20th Ann. ACM Symp. on Theory of Computing*, pages 322–333, May 1988.

[PS89] D. Peleg and A.A. Schäffer. Graph Spanners. In *Journal of Graph Theory*, 13(1):99–106, 1989.

[PU89] D. Peleg and J. Ullman. An Optimal Synchronizer for the Hypercube. In *SIAM J. Computing*, 18, pages 740-747, 1989.

[Seiden96] Steve Seiden. Unfair Problems and Randomized Algorithms for Metrical Task Systems. Manuscript, 1996.

[RR95] Y. Rabinovich and R. Raz. Lower Bounds on the Distortion of Embedding Finite Metric Spaces in Graphs. To appear in *Discrete and Computational Geometry.*

[ST85] D. Sleator and R.E. Tarjan. Amortized Efficiency of List Update and Paging Rules. *Comm. of ACM* , 28(2):202–208, 1985.

A Probabilistic Approximations of Meshes by HST's

Theorem 21 *Consider the metric space G defined by the d-dimensional torus over $n = s^d$ vertices, where s is a power of k, with the l_∞ norm. For k a power of 2, M can be α-probabilistically-approximated by k-HST's in the form of a complete tree of degree k^d and diameter $\mathrm{diam}(T) = O(\mathrm{diam}(G))$, where $\alpha = O(k \log_k n/d)$.*

(Note that for the l_p norm we would get the same with just a factor of $d^{1/p}$, that is $\alpha = O(k \log_k n/d^{1-1/p})$. Also the results can be easily adapted when s is not a power of k and to non-cyclic meshes).

Proof Sketch of Theorem 21. We first consider $k = 2$. Consider a d-dimensional cube (of size 2). It contains 2^d vertices (at distance 1 from each other). The d-dimensional torus over $n = (2^i)^d$ vertices can be partitioned into disjoint cubes. This can be done in 2^d different partitions such that every pair of adjacent vertices is included in exactly 1/2 of the partitions.

Now choose one of the partitions at random. We add new nodes and edges to the graph that will be part of the final tree construction For each cube in the partition, construct a star from the 2^d vertices of the cube by adding another center vertex at distance 1/2 of all of them. Then remove the original edges of the cube. We call this the level 0 construction. The original vertices are considered to belong to level 0 while the new vertices belong to level 1. It follows that the distances between the new center vertices match distances on a torus of size $n/2^d = (2^{i-1})^d$ where the length between two adjacent nodes is $2(= 1/2 + 1 + 1/2)$.

Now consider this new torus and construct the appropriate tree for it in the same manner, by repeating this process in $t = \log_{2^d} n = \log n/d$ levels.

It follows that the distances between vertices in the j and $j+1$ levels in the construction is $2^j/2$ and therefore the tree is 2-HST. The tree is a complete tree of degree 2^d.

It remains to bound the expected distance in the tree between any pair of adjacent nodes. Let the expected distance between a pair of adjacent nodes in the j level be X_j. Let $X_t = 0$. The distance between a pair of adjacent nodes in the j level in the tree is the sum of their distances (that is $2^j/2$) to their respective parents plus the expected distance between these parents, which is with probability 1/2 zero and with probability 1/2 equal to X_{j+1}. Thus $X_j = 2^j/2 + 2^j/2 + X_{j+1}/2 = 2^j + X_{j+1}/2$. It follows that $X_j = (t-j) \cdot 2^j$, and thus $\alpha = X_0 = \log n/d$.

Now consider $k = 2^h$ for $h > 1$. We perform the following procedure on the tree constructed above: Connect directly all vertices at level $t-h$ to the root with edges of distance $2^t/2$ and remove the former edges in the levels in between. The repeat the procedure for the subtrees of these vertices. Thus the resulting tree is a complete k-HST with degree k^d.

Let $j = hl$ for $l = 0, \ldots, t/h - 1$ be the level in the original construction, and let Y_l denote the expected distance between to vertices in level j, in the new tree. This distance is sum of their distances to their respective parents, which is $2^{j+h}/2$, plus the expected distance between these parents. We want to compute the probability that this distance is not 0. This is the probability that the respective ancestors at the $j+r$ (for $r = 1, \ldots, h$) are different at all levels which is $1/2^h$. Thus $Y_l = 2^{(l+1)h}/2 + 2^{(l+1)h}/2 + Y_{l+1}/2^h = 2^{(l+1)h} + Y_{l+1}/2^h$. It follows that $Y_l = (t/h - l) \cdot 2^{(l+1)h}$, and thus $\alpha = Y_0 = 2^h \cdot \log n/(dh) = k \cdot \log_k n/d$. ■

B Lower Bounds

Proof of Theorem 15. We show that the constrcution of probabilistic partitions in Theorem 14 cannot be improved even for weak probabilistic partitions.

The following is a known result in extremal graph theory (cf. [Boll78], pp. 107–109): there exists a constant a such that, for all n, there exists an n-vertex (unweighted) graph G_n with $2n$ edges and girth (I.e., the length of the shortest cycle in G_n) $a \log n$. Let $r = 2\lambda$. Assume that $\lambda\rho < \frac{1}{4} a \log n$, then every set U of the $(r\rho)$-weak partitions has weak diameter of at most $r\rho < \frac{1}{2} a \log n$. Now, observe the subgraph of G_n induced by such a set U. If this subgraph contains a cycle then its length is at least $a \log n$. Thus there must be two nodes on this cycle at distance at least $\frac{1}{2} a \log n$ from each other, a contradiction. Therefore every cluster is a forest, and thus the number of edges not contained in clusters of the partition is greater than n. Hence, under the probability distribution where every edge of G_n is chosen unifromly at random we have that the expected chance for an edge not to be contained in a cluster of the partition is at greater than $1/2 = \frac{\lambda}{r}$ a contradiction. Therefore $\lambda\rho = \Omega(\log n)$. ■

Proof Sketch of Theorem 9. Let G be a connected graph. Consider a set of tree metrics $\mathcal{S}$ that α-probabilistically-approximates G. We will show that this implies $(r, \alpha, 2)$-weak probabilistic partitions for G, and thus the lower bound follows from Theorem 15.

We begin by transforming the distribution over $\mathcal{S}$ into a new distribution over tree metrics as follows. For any tree $T \in \mathcal{S}$ first construct an $(r\alpha, 1, 1)$-probabilistic partition. Then define a new tree by assigning every edge not included in a subtree of the partition with an enlarged edge weight of $r\alpha$. It follows that for an edge e of T the expected edge weight in the new tree is at most $w(e) + \frac{w(e)}{r\alpha} \cdot r\alpha = 2w(e)$. Therefore the new set of trees 2α-probabilistically-approximates G. We now construct the probabilistic partitions for G by letting the sets of the partition be equal to the sets of the tree partition we previously computed. It follows that for any two vertices inside such a set $d_G(u, v) \leq d_T(u, v) \leq r\alpha$, and thus the partition constructed is an $(r\alpha)$-weak partition.

Consider an edge $e = (u, v)$ in G. Under the new probability distribution we have $\mathrm{E}(d_T(u, v)) \leq 2\alpha \cdot d_G(u, v)$. Assume that u and v are in different clusters of the partition, then the distance between them in the new tree T is at least $r\alpha$. Applying Markov inequality we have that the probability that this occurs is at most $\frac{2\alpha \cdot d_G(u,v)}{r\alpha} = 2\frac{d_G(u,v)}{r}$.
■