Prioritized Metric Structures and Embedding

[Extended Abstract]*

Michael Elkin†
Ben-Gurion University of the Negev
Beer-Sheva, Israel
elkinm@cs.bgu.ac.il.

Arnold Filtser‡
Ben-Gurion University of the Negev
Beer-Sheva, Israel
arnoldf@cs.bgu.ac.il.

Ofer Neiman§
Ben-Gurion University of the Negev
Beer-Sheva, Israel
neimano@cs.bgu.ac.il.

ABSTRACT

Metric data structures (distance oracles, distance labeling schemes, routing schemes) and low-distortion embeddings provide a powerful algorithmic methodology, which has been successfully applied for approximation algorithms [21], online algorithms [7], distributed algorithms [19] and for computing sparsifiers [28]. However, this methodology appears to have a limitation: the worst-case performance inherently depends on the cardinality of the metric, and one could not specify in advance which vertices/points should enjoy a better service (i.e., stretch/distortion, label size/dimension) than that given by the worst-case guarantee.

In this paper we alleviate this limitation by devising a suit of *prioritized* metric data structures and embeddings. We show that given a priority ranking $(x_1, x_2, \ldots, x_n)$ of the graph vertices (respectively, metric points) one can devise a metric data structure (respectively, embedding) in which the stretch (resp., distortion) incurred by any pair containing a vertex x_j will depend on the rank j of the vertex. We also show that other important parameters, such as the label size and (in some sense) the dimension, may depend only on j. In some of our metric data structures (resp., embeddings) we achieve both prioritized stretch (resp., distortion) and label size (resp., dimension) *simultaneously*. The worst-case performance of our metric data structures and embeddings is typically asymptotically no worse than of their non-prioritized counterparts.

*A full version of this paper is available at http://arxiv.org/abs/1502.05543.

†Supported by the Israeli Academy of Science, grant 593/11.

‡Partially supported by the Lynn and William Frankel Center for Computer Sciences.

§Supported in part by ISF grant No. (523/12) and by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement $n^\circ 303809$.

STOC'15, June 14–17, 2015, Portland, Oregon, USA.

ACM 978-1-4503-3536-2/15/06 ...$15.00.
http://dx.doi.org/10.1145/2746539.2746623.

Categories and Subject Descriptors
G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms*

General Terms
Algorithms, Theory

Keywords
Metric embedding; priorities; distance oracles; routing

1. INTRODUCTION

The celebrated distance oracle of Thorup and Zwick [31] enables one to preprocess an undirected weighted n-vertex graph $G = (V, E)$ so that to produce a data structure (aka *distance oracle*) of size $O(t \cdot n^{1+1/t})$ (for a parameter $t = 1, 2, \ldots$) that supports distance queries between pairs $u, v \in V$ in time $O(t)$ per query. (The query time was recently improved to $O(1)$ by [14, 32].) The distance estimates provided by the oracle are within a factor of $2t - 1$ from the actual distance $d_G(u, v)$ between u and v in G. The approximation fctor ($2t - 1$ in this case) is called the *stretch*. Distance oracles can serve as an example of a *metric data structure*; other very well-studied examples include *distance labeling* [24, 17] and *routing* [30, 6]. Thorup-Zwick's oracle can also be converted into a distance-labeling scheme: each vertex is assigned a label of size $O(n^{1/t} \cdot \log^{1-1/t} n)$ so that given labels of u and v the query algorithm can provide a $(2t - 1)$-approximation of $d_G(u, v)$. Moreover, the oracle also gives rise to a routing scheme [30] that exhibits a similar tradeoff.

A different but closely related thread of research concerns *low-distortion embeddings*. A celebrated theorem of Bourgain [11] asserts that any n-point metric (X, d) can be embedded into an $O(\log n)$-dimensional Euclidean space with *distortion* $O(\log n)$. (Roughly speaking, distortion and stretch are the same thing. See Section 2 for formal definitions.) Fakcharoenphol et al. [16] (following Bartal [8, 9]) showed that any mertic (X, d) embeds into a distribution over trees (in fact, ultrametrics) with expected distortion $O(\log n)$.

These (and many other) important results are not only appealing from a mathematical perspective, but they also were found extremely useful for numerous applications in Theoretical Computer Science and beyond [21, 7, 19, 28]. A natural disadvantage is the dependence of all the relevant parameters on n, the cardinality of the input graph/metric. However, all these results are either completely tight, or very close to being completely tight. In order to address this issue, metric data structures and embeddings in which some pairs of vertices/points enjoy better stretch/distortion or with improved label size/dimension were developed. Specifically, [20, 1, 3, 13] studied embeddings and distance oracles in which the distortion/stretch of at least $1-\epsilon$ fraction of the pairs is improved as a function of ϵ, either for a fixed ϵ or for all $\epsilon \in [0,1]$ simultaneously (e.g. for a fixed ϵ, embeddings into Euclidean space of dimension $O(\log 1/\epsilon)$ with distortion $O(\log(1/\epsilon))$, or a distance oracle with stretch $2\lceil t \cdot \frac{\log(2/\epsilon)}{\log n}\rceil + 1$ for $1-\epsilon$ fraction of the pairs). Also, [2, 27, 4] devised embeddings and distance oracles that provide distortion/stretch $O(\log k)$ for all pairs (x,y) of points such that y is among the k closest points to x, and distance labeling schemes that support queries only between k-nearest neighbors, in which the label size depends only on k rather than n.

An inherent shortcoming of these results is, however, that the pairs that enjoy better than worst-case distortion cannot be specified in advance. In this paper we alleviate this shortcoming and devise a suit of prioritized metric data structures and low-distortion embeddings. Specifically, we show that one can order the graph vertices $V = (x_1, \ldots, x_n)$ *arbitrarily in advance*, and devise metric data structures (i.e., oracles/labelings/routing schemes) that, for a parameter $t = 1, 2, \ldots$, provide stretch $2\lceil t \cdot \frac{\log j}{\log n}\rceil - 1$ (instead of $2t-1$) for *all pairs involving* x_j, while using the same space as corresponding non-prioritized data structures! In some cases the label size can be simultaneously improved for the high priority points, as described in the sequel.

The same phenomenon occurs for low-distortion embeddings. We devise an embedding of general metrics into an $O(\log n)$-dimensional Euclidean space that provides *prioritized distortion* $O(\log j \cdot (\log\log j)^{1/2+\epsilon})$, for any constant $\epsilon > 0$ (i.e., the distortion for all pairs containing x_j is $O(\log j \cdot (\log\log j)^{1/2+\epsilon})$). Similarly, our embedding into a distribution of trees provides prioritized expected distortion $O(\log j)$.

We introduce a novel notion of *improved dimension* for high priority points. In general we cannot expect that the dimension of an Euclidean embedding with low distortion (even prioritized) will be small (as Euclidean embedding into dimension D has worst-case distortion of $\Omega(n^{1/D} \cdot \log n)$ for some metrics [3]). What we can offer is an embedding in which the high ranked points have only a few "active" coordinates. That is, only the first $O(\text{poly}(\log j))$ coordinates in the image of x_j will be nonzero, while the distortion is also bounded by $O(\text{poly}(\log j))$. This could be useful in a setting where the high ranked points participate in numerous computations, then since representing these points requires very few coordinates, we can store many of them in the cache or other high speed memory. We remark that our framework is *the first* which allows simultaneously improved distortion and dimension (or improved stretch and label size) for the high priority points, while providing some guarantee for all pairs.

We have a construction of prioritized distance oracles that exhibits a qualitatively different behavior than of our aforementioned oracles. Specifically, we devise a distance oracle with space $O(n \log\log n)$ (respectively, $O(n \log^* n)$) and prioritized stretch $O(\frac{\log n}{\log(n/j)})$ (respectively, $2^{O(\frac{\log n}{\log(n/j)})}$). Observe that as long as $j < n^{1-\epsilon}$ for any fixed $\epsilon > 0$, the prioritized stretch of both these oracles is $O(1)$. The query time is $O(1)$. These oracles are, however, not path-reporting (a path reporting oracle can return an actual approximate shortest path in the graph, in time proportional to its length). We also devise a path-reporting prioritized oracle, which was mentioned above: it has space $O(t \cdot n^{1+1/t})$, stretch $2\lceil t \cdot \frac{\log j}{\log n}\rceil - 1$, and the query time[1] is $O(t \cdot \frac{\log j}{\log n})$. In the full version of this paper we also devise a path-reporting prioritized distance oracle (extending [15]) with space $O(n \log\log n)$, stretch $O((\frac{\log n}{\log(n/j)})^{\log_{4/3} 7})$, and query time $O(\log(\frac{\log n}{\log(n/j)}))$. (Observe that this stretch and query time are $O(1)$ for all $j \le n^{1-\epsilon}$.)

This second oracle can be distributed as a labeling scheme, in which not only the stretch $2\lceil t \cdot \frac{\log j}{\log n}\rceil - 1$ is prioritized, but also the label size is smaller for high priority points: it is $O(n^{1/t} \cdot \log j)$ rather than the non-prioritized $O(n^{1/t} \cdot \log n)$. In our routing scheme, if j is the priority rank of the destination x_j, it has prioritized stretch $4\lceil t \cdot \frac{\log j}{\log n}\rceil - 3$ (instead of $4t-5$), the routing tables have size $O(n^{1/t} \cdot \log j)$ (instead of $O(n^{1/t} \cdot \log n)$), and labels have size $O(\log j \cdot \lceil t \frac{\log j}{\log n}\rceil)$ (instead of $O(t \cdot \log n)$).

We also consider the dual setting in which the stretch is fixed, and label size $\lambda(j)$ of x_j is smaller when $j \ll n$. The function $\lambda(j)$ will be called *prioritized label size*. Specifically, with prioritized label size $O(j^{1/t} \cdot \log j)$ we can have stretch $2t-1$. For certain points on the tradeoff curve we can even have both stretch and label size prioritized simultaneously! In particular, a variant of our distance labeling scheme provides a prioritized stretch $2\lceil \log j\rceil - 1$ and prioritized label size $O(\log j)$. For routing we have similar gaurantees independent of n. We also devise a distance labeling scheme for graphs that exclude a fixed minor with stretch $1+\epsilon$ and prioritized label size $O(1/\epsilon \cdot \log j)$ (extending [5, 29]).

Another notable result in this context is our prioritized embedding into a *single tree*. It is well-known that any metric can be embedded into a single dominating tree with linear distortion, and that it is tight [25]. We show that any n-point metric (X,d) enjoys an embedding into a single dominating tree with prioritized distortion $\alpha(j)$ *if and only if* the sum of reciprocals $\sum_{j=1}^{\infty} 1/\alpha(j)$ converges. In particular, prioritized distortion $\alpha(j) = j \cdot \log j \cdot (\log\log j)^{1.01}$ is admissible, while $\alpha(j) = j \cdot \log j \cdot \log\log j$ is not, i.e., both our upper and lower bounds are tight. This lower bounds stands out as it shows that it is not always possible to replace non-prioritized distortion of $\alpha(n)$ by a prioritized distortion $\alpha(j)$. For single-tree embedding the non-prioritized distortion is linear, while the prioritized one is provably superlinear.

1.1 Overview of Techniques

We elaborate briefly on the methods used to obtain our results.

Distance Oracles, Distance Labeling and Routing.

We have two basic techniques for obtaining distance oracles with prioritized stretch. The first one is manifested in Corollary 2, and the idea is as follows: Partition the vertices into sets according to their priority, and for each set $K \subseteq V$,

[1] We believe the query time can be improved to $O(1)$: [14] combines the oracles of [31] and of [23] to obtain query time $O(1)$. In the full version of our paper we show that the oracle of [23] can be altered to give prioritized stretch, similar to that of [31] we show here. Using the techniques of [14] should thus yield prioritized stretch with $O(1)$ query time.

apply as a black-box a known distance oracle on K, while for the other vertices store the distance to their nearest neighbor in K. We show that the stretch of pairs in $K \times V$ is only a factor of 2 worse than the one guaranteed for $K \times K$. Furthermore, we exploit the fact that for sets K of small size, we can afford very small stretch and still maintain small space. The exact choice of the black-box oracle and of the partitions enables a range of tradeoffs between space and prioritized stretch.

Our second technique for an oracle with prioritized stretch, used in Theorem 5, is based on a non-black-box variation of the [31] oracle. In their construction for stretch $2t - 1$, a (non-increasing) sequence of $t - 1$ sets is generated by repeated random sampling. We show that if a vertex is chosen i times, then the query algorithm can be changed to improve the stretch from $2t - 1$ to $2(t - i) - 1$, for *any pair* containing such a vertex. This observation only shows that there exists a priority ranking for which the oracle has the required prioritized stretch. In order to handle *any* given ranking, we alter the construction by forcing high ranked elements to be chosen numerous times, and show that this increases the space usage by at most a factor of 2.

In order to build a distance labeling scheme out of their distance oracle, [31] pay an additional factor of $O(\log^{1-1/t} n)$ in the label size (which essentially comes from applying concentration bounds). Attempting to circumvent this logarithmic dependence on n, in Theorem 6 we give a different bound on the deviation probability that depends on the priority ranking of the point. Thus the increase in the label size for the j-th point in the ranking is only $O(\log j)$. To obtain arbitrary fixed stretch $2t - 1$ for all pairs, we combine this scheme with an iterative application of a *source restricted* distance labeling of [26].

Most results on distance labeling for bounded treewidth graphs, planar graphs, and graphs excluding a fixed minor, are based on recursively partitioning the graph into small pieces using small separators (as in [22]). The label of a vertex essentially consists of the distances to (some of) the vertices in the separator. In order to obtain prioritized label size, high ranked vertices should participate in few iterations. To this end, we define multiple phases of applying separators, where each phase tries to separate only certain subset of the vertices (starting with the highest ranked, and finishing in the lowest). This way high ranked vertices will belong to a separator after a few levels, thus their label will be short.

Tree-routing of [29] is based on categorizing tree vertices as either heavy or light, depending on the size of their subtree. Our prioritized tree-routing assigns each vertex a weight which depends on its priority, and a vertex is heavy if the sum of weights of its descendents is sufficiently large. This idea paves way to our prioritized routing scheme for general graphs as well.

Embeddings.

It is folklore that a metric minimum spanning tree (henceforth, MST) achieves distortion $n - 1$. For our prioritized embedding of general metrics (X, d) into a single tree we consider a complete graph $G = (X, \binom{X}{2})$ with weight function that depends on the priority ranking. Specifically, edges incident on high-priority points get higher weights. We then compute an MST in this (generally non-metric) graph, and show that, given a certain convergence condition on the priority ranking, this MST provides a desired prioritized single-tree embedding. Remarkably, we also show that when this condition is not met, no such an embedding is possible even for the metric induced by C_n. Hence this embedding is tight.

Our probabilistic embedding to trees with prioritized expected distortion in Theorem 4 is based on the construction of [16]. The method of [16] involves sampling a random permutation and a random radius, then using these to create a hierarchical partitioning of the metric from which a tree is built. We make the observation that, in some sense, the expected distortion of a point depends on its position in the permutation. Rather than choosing a permutation uniformly at random, we choose one which is strongly correlated with the given priority ranking. One must be careful to allow sufficient randomness in the permutation choice so that the analysis can still go through, while guaranteeing that high ranked points will appear in the first positions of the permutation.

The embedding of Theorem 8 for arbitrary metrics (X, d) into Euclidean space (or any ℓ_p space) with prioritized distortion uses similar ideas. We partition the points to sets according to the priorities, for every set $K \subseteq X$ apply as a black-box the embedding of [12]. We show that since the embedding has certain properties, it can be extended in a Lipschitz manner to all of the metric, while having distortion guarantee for any pair in $K \times X$.

The result of Theorem 9, which gives prioritized distortion and dimension, is more technically involved. In order to ensure that high priority points are mapped to the zero vector in the embeddings tailored for the lower priority points, we change Bourgain's embedding, which is defined as distances to randomly chosen sets. Roughly speaking, when creating the embedding for a set K, we add all the higher ranked points to the random sets. This means the original analysis does not work directly, and we turn to a subtle case analysis to bound the distortion; see the full version for more details.

1.2 Organization

After a few preliminary definitions, we show the single tree prioritized embedding in Section 3, and the probabilistic version in Section 4. In Section 5 we discuss some of our prioritized distance oracles and labeling schemes. Finally, in Section 6 we present some of our prioritized embedding results into normed spaces. Our constructions of prioritized distance labeling for minor-free graphs and prioritized routing are deferred to the full version.

2. PRELIMINARIES

All the graphs $G = (V, E)$ we consider are undirected and weighted. Let $x_1, \ldots, x_n \in V$ be a priority ranking of the vertices. Let d_G be the shortest path metric on G, and let $\alpha, \beta : [n] \to \mathbb{R}_+$ be monotone non-decreasing functions.

A distance oracle for a graph G is a succinct data structure, that can approximately report distances between vertices of G. The parameters of this data structure we will care about are its space, query time, and stretch factor. We always measure the space of the oracle as the number of words needed to store it (where each word is $O(\log n)$ bits). The oracle has *prioritized stretch* $\alpha(j)$, if for any $1 \le j < i \le n$, when queried for x_j, x_i the oracle reports a distance $\tilde{d}(x_j, x_i)$ such that

$$d_G(x_j, x_i) \le \tilde{d}(x_j, x_i) \le \alpha(j) \cdot d_G(x_j, x_i) \ .$$

Some oracles can be distributed as a labeling scheme, where each vertex is given a short label, and the approximate distance between two vertices should be computed by inspecting their labels alone. We say that the a labeling scheme has *prioritized label size* $\beta(j)$, if for every $j \in [n]$, the label of x_j consists of at most $\beta(j)$ words.

Let (X, d_X) be a finite metric space, and let $x_1, \ldots, x_n$ be a priority ranking of the points in X. Given a target metric (Y, d_Y), and a non-contractive map $f : X \to Y$,[2] we say that f has *priority distortion* $\alpha(j)$ if for all $1 \le j < i \le n$,

$$d_Y(f(x_j), f(x_i)) \le \alpha(j) \cdot d_X(x_j, x_i) \ .$$

Similarly, if $f : X \to Y$ is non-expansive, then it has priority distortion $\alpha(j)$ if for all $1 \le j < i \le n$, $d_Y(f(x_j), f(x_i)) \ge d_X(x_j, x_i)/\alpha(j)$. For probabilistic embedding, we require that each map in the support of the distribution is non-contractive, and the prioritized bound on the distortion holds in expectation.

In the special case that the target metric is a normed space ℓ_p, we say that the embedding has *prioritized dimension* $\beta(j)$, if for every $j \in [n]$, only the first $\beta(j)$ coordinates in $f(x_j)$ may be nonzero.

3. SINGLE TREE EMBEDDING WITH PRIORITIZED DISTORTION

In this section we show tight bounds on the priority distortion for an embedding into a single tree. The bounds are somewhat non-standard, as they are not attained for a single specific function, but rather for the following family of functions. Define Φ to be the family of functions $\alpha : \mathbb{N} \to \mathbb{R}_+$ that satisfy the following properties:

- α is non-decreasing.
- $\sum_{i=1}^{\infty} 1/\alpha(i) \le 1$.

THEOREM 1. *For any finite metric space (X, d) and any $\alpha \in \Phi$, there is a (non-contractive) embedding of X into a single tree with priority distortion $2\alpha(j)$.*

PROOF. Let $x_1, \ldots, x_n$ be the priority ranking of X, and let $G = (X, E)$ be the complete graph on X. For $e = \{u, v\} \in E$, let $\ell(e) = d(u, v)$. We also define the following (prioritized) weights $w : E \to \mathbb{R}$, for any $1 \le j < i \le n$ the edge $e = \{x_j, x_i\}$ will be given the weight $w(e) = \alpha(j) \cdot \ell(e)$. Observe that the w weights on G do not necessarily satisfy the triangle inequality. Let T be the minimum spanning tree of (X, E, w) (this tree is formed by iteratively removing the heaviest edge from a cycle). Finally, return the tree T with the edges weighted by ℓ. We claim that this tree has priority distortion $\alpha(j)$.

Consider some $x_j, x_i \in X$, if the edge $e = \{x_j, x_i\} \in E(T)$ then clearly this pair has distortion 1. Otherwise, let P be the unique path between x_j and x_i in T. Since e is not in T, it is the heaviest edge on the cycle $P \cup \{e\}$, and for any edge $e' \in P$ we have that $w(e') \le w(e) = \alpha(j) \cdot d(x_j, x_i)$. Consider some $x_k \in X$, and note that there can be at most 2 edges touching x_k in P. If $e' \in P$ is such an edge, and its weight by w was changed by a factor of $\alpha(k)$, then $\alpha(k) \cdot \ell(e') \le \alpha(j) \cdot d(x_j, x_i)$. Summing this over all the possible values of k we obtain that the length of P is at most

$$\sum_{e' \in P} \ell(e') \le 2 \sum_{k=1}^{n} \frac{\alpha(j)}{\alpha(k)} \cdot d(x_j, x_i) \le 2\alpha(j) \cdot d(x_j, x_i) \ . \quad (1)$$

□

Corollary 1. For any fixed $0 < \epsilon < 1/2$, one can take the function $\alpha : \mathbb{N} \to \mathbb{R}$ defined by $\alpha(1) = 1 + \epsilon$, and for $j \ge 2$, $\alpha(j) = \frac{j(\log j)^{1+\epsilon}}{c}$, which lies in Φ for $c \approx \epsilon^2$, and obtain priority distortion $O\left(j(\log j)^{1+\epsilon}\right)$. Furthermore, the distortion of the pairs containing x_1 is only $1 + 3\epsilon$.

The proof of Corollary 1 appears in the full version. Next, we show a matching lower bound (up to a constant, which is only 2 for trees without Steiner nodes) on the possible functions admitting an embedding into a tree with priority distortion. We begin by showing that a (non-decreasing) function which is not in Φ cannot bound the priority distortion in a *spanning tree* embedding.

THEOREM 2. *For any non-decreasing function $\alpha : \mathbb{N} \to \mathbb{R}$ with $\alpha \notin \Phi$, there exists an integer n, a graph $G = (V, E)$ with $|V| = n$ vertices, and a priority ranking of V, such that no spanning tree of G has priority distortion less than α.*

PROOF. Since $\alpha \notin \Phi$, there exists an integer n' such that $\sum_{i=1}^{n'} 1/\alpha(i) > 1$. Take some integer $n > n'$ such that $\frac{n}{\alpha(i)+1}$ is an integer for all $1 \le i \le n'$ (assume w.l.o.g that the $\alpha(i)$ are rational numbers). Then let $G = C_n$, a cycle on n points with unit weight on the edges. Clearly, a spanning tree of C_n is obtained by removing a single edge, thus we will choose the priorities $x_1, \ldots, x_n \in V$ in such a way that no edge can be spared.

Seeking contradiction, assume that there exists a spanning tree with priority distortion less than α. Let x_1 be an arbitrary vertex, and note that if u is a vertex within distance $a_1 = n/(\alpha(1) + 1)$ from x_1, then all the edges on the shortest path from x_1 to u must remain in the tree. Otherwise, the distortion of the pair $\{x_1, u\}$ will be at least $\frac{n-a_1}{a_1} = \alpha(1)$. There are $\frac{2n}{\alpha(1)+1}$ such edges that must belong to the tree (since we consider vertices from both sides of x_1). Now take x_2 to be a vertex at distance $\frac{n}{\alpha(1)+1} + \frac{n}{\alpha(2)+1}$ from x_1. By a similar argument, the $\frac{2n}{\alpha(2)+1}$ edges closest to x_2 must be in the tree as well. Observe that these edges form a continuous sequence on the cycle with those edges near x_1. Continue in this manner to define $x_3, \ldots, x_{n'}$, and conclude that there are at least

$$\sum_{i=1}^{n'} \frac{2n}{\alpha(i) + 1} \ge \sum_{i=1}^{n'} \frac{n}{\alpha(i)} > n \quad (2)$$

edges that are not allowed to be removed, but this is a contradiction, as there are only n edges in C_n. □

In the full version, using a technique similar to that of [18], we extend the lower bound and show the following

THEOREM 3. *For any non-decreasing function $\alpha : \mathbb{N} \to \mathbb{R}$ with $\alpha \notin \Phi$, there exists an integer n, a metric (X, d) on n points and a priority ranking $x_1, \ldots, x_n \in X$, such that there is no embedding of X into a dominating tree metric[3] with priority distortion less than $\alpha/8$.*

[2]The map f is non-contractive if for any $u, v \in X$, $d_X(u, v) \le d_Y(f(u), f(v))$.

[3]A tree T dominates a metric d on the same set of points if $d_T \ge d$.

4. PROBABILISTIC EMBEDDING INTO ULTRAMETRICS WITH PRIORITIZED DISTORTION

An ultrametric (U, d) is a metric space satisfying a strong form of the triangle inequality, that is, for all $x, y, z \in U$, $d(x, z) \leq \max\{d(x, y), d(y, z)\}$. The following definition is known to be an equivalent one (see [10]).

Definition 1. An ultrametric U is a metric space (U, d) whose elements are the leaves of a rooted labeled tree T. Each $z \in T$ is associated with a label $\Phi(z) \geq 0$ such that if $q \in T$ is a descendant of z then $\Phi(q) \leq \Phi(z)$ and $\Phi(q) = 0$ iff q is a leaf. The distance between leaves $z, q \in U$ is defined as $d_T(z, q) = \Phi(\mathrm{lca}(z, q))$ where $\mathrm{lca}(z, q)$ is the least common ancestor of z and q in T.

THEOREM 4. *For any metric space (X, d), there exists a distribution over embeddings of X into ultrametrics with expected prioritized distortion $O(\log j)$.*

PROOF. Let $x_1, \ldots, x_n$ be the priority ranking of X, and let Δ be the diameter of X. We assume w.l.o.g that the minimal distance in X is 1, and let δ be the minimal integer so that $\Delta \leq 2^\delta$. We shall create a hierarchical laminar partition, where for each $i \in \{0, 1, \ldots, \delta\}$, the clusters of level i have diameter at most 2^i, and each of them is contained in some level $i+1$ cluster. The ultrametric is built in the natural manner, the root corresponds to the level δ cluster which is X, and each cluster in level i corresponds to an inner node of the ultrametric with label 2^i, whose children correspond to the level $i-1$ clusters contained in it. The leaves correspond to singletons, that is, to the elements of X. Clearly, the ultrametric will dominate (X, d).

In order to define the partition, we choose a random permutation $\pi : X \to [n]$ which is strongly correlated with the priority ranking, and in addition we choose some number $\beta \in [1, 2]$. Let $K_0 = \{x_1, x_2\}$, and for any integer $1 \leq j \leq \lceil \log \log n \rceil$ let $K_j = \{x_h \ : \ 2^{2^{j-1}} < h \leq 2^{2^j}\}$. The permutation π is created by choosing a uniformly random permutation on each K_i, and concatenating these. Note that $\pi^{-1}\left(\left\{h \in \mathbb{N} \ : \ h \in \left(2^{2^{j-1}}, 2^{2^j}\right]\right\}\right) = K_j$, and $\pi^{-1}(\{1, 2\}) = K_0$.

In each step i, we partition a cluster S of level $i+1$ as follows. Each point $x \in S$ chooses the point $u \in X$ with minimal value according to π among the points of distance at most $\beta_i := \beta \cdot 2^{i-2}$ from x, and joins to the cluster of u. Note that a point may not belong to the cluster associated with it, and some clusters may be empty (which we can discard). The description of the hierarchical partition appears in Algorithm 1.

Let T denote the ultrametric created by the hierarchical partition of Algorithm 1, and $d_T(u, v)$ the distance between u to v in T. Consider the clustering step at some level i, where clusters in D_{i+1} are picked for partitioning. In each iteration l, all unassigned points z such that $d(z, \pi(l)) \leq \beta_i$, assign themselves to the cluster of $\pi(l)$. Fix an arbitrary pair $\{v, u\}$. We say that center w *settles* the pair $\{v, u\}$ at level i, if it is the first center so that at least one of u and v gets assigned to its cluster. Note that exactly one center w settles any pair $\{v, u\}$ at any particular level. Further, we say that a center w *cuts* the pair $\{v, u\}$ at level i, if it settles them at this level, and exactly one of u and v is assigned to the cluster of w at level i. Whenever w cuts a pair $\{v, u\}$ at level i, $d_T(v, u)$ is set to be $2^{i+1} \leq 8\beta_i$. We blame this length to the point w and define $d_T^w(v, u)$ to be $\sum_i \mathbf{1}(w \text{ cuts } \{v, u\} \text{ at level } i) \cdot 8\beta_i$ (where $\mathbf{1}(\cdot)$ denotes an indicator function). We also define $d_T^{K_j}(v, u) = \sum_{w \in K_j} d_T^w(v, u)$. Clearly, $d_T(v, u) \leq \sum_j d_T^{K_j}(v, u)$.

Fix some $0 \leq j \leq \lceil \log \log n \rceil$, our next goal is to bound the expected value of $d_T^{K_j}(v, u)$ by $O(\log(|K_j|))$. We arrange the points of K_j in non-decreasing order of their distance from the pair $\{v, u\}$ (breaking ties arbitrarily). Consider the sth point w_s in this sequence. W.l.o.g assume that $d(w_s, v) \leq d(w_s, u)$. For a center w_s to cut $\{v, u\}$, it must be the case that 1) $d(w_s, v) \leq \beta_i < d(w_s, u)$ for some i, and 2) w_s settles $\{v, u\}$ at level i.

Note that for each $x \in [d(w_s, v), d(w_s, u))$, the probability that $\beta_i \in [x, x + dx)$ is at most $\frac{dx}{x \cdot \ln 2}$. Conditioning on β_i taking such a value x, any one of $w_1, \ldots, w_s$ can settle $\{v, u\}$. The probability that w_s is first in the permutation π among $w_1, \ldots w_s$ is $\frac{1}{s}$. (In fact, there may be points from $\bigcup_{0 \leq r < j} K_r$ that settle $\{v, u\}$ before w_s. It is safe to ignore that, as it can only decrease the probability that w_s cuts $\{v, u\}$.) Thus we obtain,

$$\begin{aligned} \mathbb{E}[d_T^{w_s}(v, u)] &\leq \int_{d(w_s, v)}^{d(w_s, u)} 8x \cdot \frac{dx}{x \ln 2} \cdot \frac{1}{s} \qquad (3) \\ &= \frac{8}{s \cdot \ln 2}(d(w_s, u) - d(w_s, v)) \leq \frac{16}{s} \cdot d(v, u) \ . \end{aligned}$$

Hence we conclude,

$$\begin{aligned} \mathbb{E}[d_T^{K_j}(v, u)] &\leq \sum_{w_s \in K_j} \mathbb{E}[d_T^{w_s}(v, u)] \qquad (4) \\ &\overset{(3)}{\leq} 16 d(v, u) \sum_{s=1}^{|K_j|} \frac{1}{s} = \log|K_j| \cdot O(d(v, u)) \ . \end{aligned}$$

Assume $v = x_h$ is the h-th vertex in the priority ranking for some $h > 2$. Let a be the integer such that $v \in K_a$, and recall that $2^{2^{a-1}} < h \leq 2^{2^a}$, i.e., $2^a \leq 2 \log h$. The crucial observation is that if $y \in K_b$ such that $b > a$, then y cannot settle $\{v, u\}$. The reason is that v always appears before y in π, so v will surely be assigned to a cluster when it is the turn of y to create a cluster. This leads to the conclusion that for all $b > a$, $\mathbb{E}[d_T^{K_b}(v, u)] = 0$. We conclude:

$$\begin{aligned} \mathbb{E}[d_T(v, u)] &\leq \sum_{j=0}^{a} \mathbb{E}[d_T^{k_j}(v, u)] \\ &\overset{(4)}{\leq} O(d(v, u)) \sum_{j=0}^{a} \log|K_j| \\ &= O(d(v, u)) \sum_{j=0}^{a} \log\left(2^{2^j}\right) \\ &= O(d(v, u)) \sum_{j=0}^{a} 2^j \\ &= O(d(v, u)) \cdot 2^a \\ &= O(d(v, u)) \cdot \log h \ . \end{aligned}$$

When $h \in \{1, 2\}$ we can take $a = 0$, and thus obtain a bound of $O(d(v, u))$. $\square$

Algorithm 1 `Modified FRT`(X, π)

Choose a random permutation $\pi : X \to [n]$ as above.
Choose $\beta \in [1, 2]$ randomly by the distribution with the following probability density function $p(x) = \frac{1}{x \ln 2}$.
Let $D_\delta = X$; $i \leftarrow \delta - 1$.
while D_{i+1} has non-singleton clusters **do**
  Set $\beta_i \leftarrow \beta \cdot 2^{i-2}$.
  for $l = 1, \ldots, n$ **do**
    for every cluster S in D_{i+1} **do**
      Create a new cluster in D_i, consisting of all unassigned points in S closer than β_i to $\pi(l)$.
    end for
  end for
  $i \leftarrow i - 1$.
end while

5. METRIC DATA STRUCTURES

5.1 Distance Oracles with Prioritized Stretch

Our first result provides a range of distance oracles with prioritized stretch and extremely low space. They also exhibit a somewhat non-intuitive (although very good) dependence of the stretch on the priority of the vertices. The drawbacks of these oracles are that they cannot report the approximate paths in the graph between the queried vertices, and it is not clear that they can be distributed as a labeling scheme.

For the sake of brevity, denote by $\tau(j) = \left\lfloor \frac{\log n}{\log(n/j)} \right\rfloor$ (where n is always the number of vertices). The following corollary is a result of a general theorem stated and proved in the full version.

Corollary 2. Any weighted graph $G = (V, E)$ on n vertices admits distance oracles with query time $O(1)$ and the following possible tradeoffs between space and prioritized stretch:
1) Space $O(n \log^2 n)$, prioritized stretch $\min\{4\tau(j) - 1, \log n\}$;
2) Space $O(n \log n)$, prioritized stretch $\min\{8\tau(j) - 5, \log n\}$;
3) Space $O(n \log \log n)$, prioritized stretch $\min\{O(\tau(j)), \log n\}$;
4) Space $O(n \log \log \log n)$, prioritized stretch $\min\{O(\tau(j)^2), \log n\}$;
5) Space $O(n \log^* n)$, prioritized stretch $\min\{O(2^{\tau(j)}), \log n\}$.

Observe that the first two oracles have stretch 3 for all points of priority less than $\sqrt{n}$, and that in all of these oracles, for any fixed $\epsilon > 0$, all vertices of priority at most $n^{1-\epsilon}$ have *constant stretch*.

The second result of this section extends the distance oracle of [31]. Unlike the oracles presented in Corollary 2, this oracle can also support path queries, that is, return a path in the graph that achieves the required stretch, in time proportional to its length (plus the distance query time). Additionally, it can be distributed as a labeling scheme, which we exploit in Section 5.2. Furthermore, this oracle matches the best known bounds for the worse-case stretch of [31], which are conjectured to be optimal.

THEOREM 5. *Let $G = (V, E)$ be a graph with n vertices. Given a parameter $t \geq 1$, there exists a distance oracle of space $O(tn^{1+1/t})$ with prioritized stretch $2\lceil \frac{t \log j}{\log n} \rceil - 1$ and query time $O(\lceil \frac{t \log j}{\log n} \rceil)$.*[4]

[4]As mentioned in the introduction, we believe the query time can be reduced to $O(1)$.

Overview.
Recall that in the distance oracle construction of [31], a sequence of sets $V = A_0 \supseteq A_1 \supseteq \cdots \supseteq A_t = \emptyset$ is sampled randomly, by choosing each element of A_{i-1} to be in A_i with probability $n^{-1/t}$. We make the crucial observation that the distance oracle provides improved stretch of $2(t-i)-1$, rather than $2t-1$, to points in A_i. However, as these sets are chosen randomly, they have no correlation with our given priority list over the vertices. We therefore alter the construction to ensure that points with high priority will surely be chosen to A_i for sufficiently large i.

PROOF OF THEOREM 5. Let $x_1, \ldots, x_n \in V$ be the priority ranking of V. For each $i \in \{0, 1, \ldots, t-1\}$ let $S_i = \{x_j \ : \ 1 \leq j \leq n^{1-i/t}\}$. Let $A_0 = V$, $A_t = \emptyset$, and for each $1 \leq i \leq t-1$ define A'_i by including every element of A_{i-1} with probability $n^{-1/t}/2$, and let $A_i = A'_i \cup S_i$. For each $v \in V$ and $0 \leq i \leq t-1$, define the i-th pivot $p_i(v)$ as the nearest point to v in A_i, and $B_i(v) = \{w \in A_i \ : \ d(v, w) < d(v, A_{i+1})\}$.[5] Also the *bunch* of v is defined as $B(v) = \bigcup_{0 \leq i \leq t-1} B_i(v)$. The distance oracle will store in a hash table, for each $v \in V$, all the distances to points in $B(v)$, and also the $p_i(v)$ vertices.

The query algorithm for the distance between u, v is essentially the same as in [31], with the main difference is that we start the process at level i rather than level 0, for a specified value of i.

Algorithm 2 `Dist`(v, u, i)

$w \leftarrow v$;
while $w \notin B(u)$ **do**
  $i \leftarrow i + 1$;
  $(u, v) \leftarrow (v, u)$;
  $w \leftarrow p_i(v)$;
end while
return $d(w, u) + d(w, v)$;

Stretch.
Let $v = x_j$ be the j-th point in the ordering for some $j > 1$, and fix any $u \in V$. (Observe that every vertex of A_{t-1} lies in all the bunches, so when considering $x_1 \in A_{t-1}$, we have that $x_1 \in B(u)$ and so Algorithm 2 will return the exact distance.) Let $0 \leq i \leq t-1$ be the integer satisfying that $n^{1-(i+1)/t} < j \leq n^{1-i/t}$, that is, the maximal i such that $v \in S_i$. By definition we have that $v \in A_i$ as well, so we may run `Dist`(v, u, i). Assuming that all operations in the hash table cost $O(1)$, the query time is $O(t - i)$. The stretch analysis is similar to [31]: let u_k, v_k and w_k be the values of u, v and w at the k-th iteration, it suffices to show that at every iteration in which the algorithm did not stop, $d(v_k, w_k)$ increases by at most $d(u, v)$. It suffices because there are at most $t - 1 - i$ iterations (since $w_{t-1} \in A_{t-1}$, it lies in all bunches), so if ℓ is the final iteration, it must be that $d(v_\ell, w_\ell) \leq (\ell - i) \cdot d(u, v)$ (initially $d(w_i, v_i) = 0$), and by the triangle inequality $d(w_\ell, u_\ell) \leq d(u, v) + d(v_\ell, w_\ell) \leq (\ell - i + 1) \cdot d(u, v)$, and as $\ell \leq t - 1$ we conclude that

$$d(w, u) + d(w, v) \leq (2(t - i) - 1) \cdot d(u, v) \ .$$

To see the increase by at most $d(u, v)$ at every iteration, we first note that $w_i = v_i \in A_i$ (this fact enables us to

[5]We assume that $d(v, \emptyset) = \infty$ (this is needed as $A_t = \emptyset$).

start at level i rather than in level 0). In the k-th iteration, observe that as $w_k \notin B(u_k)$ but $w_k \in A_k$, it must be that $d(u_k, p_{k+1}(u_k)) \leq d(u_k, w_k)$. The algorithm sets $w_{k+1} = p_{k+1}(u_k)$, $v_{k+1} = u_k$ and $u_{k+1} = v_k$, so we get that

$$\begin{aligned} d(v_{k+1}, w_{k+1}) &= d(u_k, p_{k+1}(u_k)) \leq d(u_k, w_k) \\ &\leq d(u_k, v_k) + d(v_k, w_k) = d(u, v) + d(v_k, w_k) . \end{aligned}$$

Note that as $n^{1-(i+1)/t} < j \leq n^{1-i/t}$, it follows that $t - i - 1 < \frac{t \log j}{\log n} \leq t - i$, so that $t - i = \lceil \frac{t \log j}{\log n} \rceil$. The guaranteed stretch for pairs containing x_j is thus bounded by $2\lceil \frac{t \log j}{\log n} \rceil - 1$ (or stretch 1 for x_1).

Space.

Fix any $u \in V$, and let us analyze the expected size of $B(u)$. Fix any $0 \leq i \leq t-2$, and consider $B_i(u)$. Assume we have already chosen the set A_i, and arrange the vertices of $A_i = \{a_1, \ldots a_m\}$ in order of increasing distance to u. Note that if a_r is the first vertex in the ordering to be in A_{i+1}, then $|B_i(u)| = r - 1$. Every vertex of A_i is either in S_{i+1} and thus will surely be included in A_{i+1}, otherwise it has probability $n^{-1/t}/2$ to be in A'_{i+1} and so in A_{i+1} as well. The number of vertices that we see until the first success (being in A_{i+1}) is stochastically dominated by a geometric distribution with parameter $p = n^{-1/t}/2$, which has expectation of $2n^{1/t}$. For the last level $t-1$, note that each vertex in $S_i \setminus S_{i+1}$ has probability exactly $(n^{-1/t}/2)^{t-1-i} = n^{-1+(i+1)/t}/2^{t-1-i}$ to be included in A_{t-1}, independently of all other vertices. As $|S_i \setminus S_{i+1}| \leq |S_i| = n^{1-i/t}$, the expected number of vertices in A_{t-1} is

$$\sum_{i=0}^{t-1} n^{1-i/t} \cdot n^{-1+(i+1)/t}/2^{t-1-i} < 2n^{1/t} . \tag{5}$$

This implies that $\mathbb{E}[|B_{t-1}(u)|] \leq 2n^{1/t}$ as well, so $\mathbb{E}[|B(u)|] \leq 2t \cdot n^{1/t}$. The total expected size of all bunches is therefore at most $2t \cdot n^{1+1/t}$. $\square$

5.2 Prioritized Distance Labeling

In this section we discuss distance labeling schemes, in which every vertex receives a short label, and it should be possible to approximately compute the distance between any two vertices from their labels alone. The novelty here is that we would like "important" vertices, those that have high priority, to have both improved stretch and also short labels.

We begin by showing that the stretch-prioritized oracle of Theorem 5 can be made into a labeling scheme, with the same stretch guarantees, and small label for high ranking points. The result has some dependence on n in the label size, and it seems to be interesting particularly for large values of t. Indeed, we shall use this result with parameter $t = \log n$ in the sequel, to obtain fully prioritized label size which will be independent of n, and can support any desired maximum stretch (this result is appears in the full version). Furthermore, this result is the basis for our routing schemes with prioritized label size and stretch, which are deferred to the full version.

THEOREM 6. *For any graph $G = (V, E)$ with n vertices and any $t \geq 1$, there exists a distance labeling scheme with prioritized stretch $2\lceil \frac{t \log j}{\log n} \rceil - 1$ and prioritized label size $O(n^{1/t} \cdot \log j)$.*

PROOF. Using the same notation as Section 5.1, the label of vertex $v \in V$ consists of its hash table (which contains distances to all points in the bunch $B(v)$, and the identity of the pivots $p_i(v)$ for $0 \leq i \leq t-1$). Note that Algorithm 2 uses only this information to compute the approximate distance. The stretch guarantee is prioritized as above, and it remains to give an appropriate bound on the label sizes.

Let $x_1, \ldots, x_n \in V$ be the priority ranking of V. Fix a point $v = x_j$ for some $j > 1$, and let i be the maximal such that $v \in S_i$. Note that this implies that $t - i - 1 < \frac{t \log j}{\log n}$. Observe that $B_0(v) \cup \cdots \cup B_{i-1}(v) = \emptyset$, so it remains to bound the size of $B_i(v), \ldots, B_{t-1}(v)$. For the last set $B_{t-1}(v) = A_{t-1}$, let $\mathcal{E}$ be the event that $|A_{t-1}| \leq 8n^{1/t}$. We already noted in (5) that the expected size of A_{t-1} is at most $2n^{1/t}$, thus using Markov, with probability at least $3/4$ event $\mathcal{E}$ holds.

For $i \leq k \leq t-2$, let X_k be a random variable distributed geometrically with parameter $p = n^{-1/t}/2$, thus $\mathbb{E}[X_k] = 2n^{1/t}$ for all k. We noted above that the distribution of X_k is stochastically dominating the cardinality of $B_k(v)$, thus it suffices to bound $\sum_{k=i}^{t-2} X_k$. Observe that for any integer s, if $\sum_{k=i}^{t-2} X_k > s$ then it means that in a sequence of s independent coin tosses with probability p for heads, we have seen less than $t - 1 - i$ heads. That is, if $Z \sim \mathrm{Bin}(s, p)$ is a Binomial random variable then

$$\begin{aligned} \Pr\left[\sum_{k=i}^{t-2} X_k > s\right] &= \Pr[Z < t - 1 - i] \\ &\leq \Pr\left[Z < \frac{t \log j}{\log n}\right] \leq \Pr[Z < \log j] . \end{aligned}$$

Take $s = 16n^{1/t} \cdot \log j$ (assume this is an integer), so that $\mu := \mathbb{E}[Z] = 8 \log j$, and by a standard Chernoff bound

$$\Pr[Z < \log j] = \Pr[Z < \mu/8] \leq e^{-3\mu/8} < 1/j^3 .$$

Let $\mathcal{F} = \left\{\exists\, 2 \leq j \leq n \ : \ \left|\bigcup_{k=0}^{t-2} B_k(x_j)\right| > 16n^{1/t} \cdot \log j\right\}$, then by a union bound over all $2 \leq j \leq n$ (note that the bound is non-uniform, and depends on j), we obtain that

$$\begin{aligned} \Pr[\mathcal{F}] &\leq \sum_{j=2}^{n} \Pr\left[\left|\sum_{k=0}^{t-2} B_k(x_j)\right| > 16n^{1/t} \cdot \log j\right] \\ &\leq \sum_{j=2}^{n} 1/j^3 < 1/4 . \end{aligned}$$

We conclude that with probability at least $1/2$ both events $\mathcal{E}$ and $\bar{\mathcal{F}}$ hold, which means that the size of the bunch of each x_j is bounded by $O(n^{1/t} \cdot \log j)$, as required. (Recall that $x_1 \in A_{t-1}$, so its label size is $|A_{t-1}| \leq 8n^{1/t}$ when event $\mathcal{E}$ holds.) $\square$

Corollary 3. Any graph $G = (V, E)$ has a distance labeling scheme with prioritized stretch $2\lceil \log j \rceil - 1$ and prioritized label size $O(\log j)$.

In the full version we also prove the following result.

THEOREM 7. *For any graph $G = (V, E)$ and an integer $t \geq 1$, there exists a distance labeling scheme with stretch $2t - 1$ and prioritized label size $O(j^{1/t} \cdot \log j)$.*

6. PRIORITIZED EMBEDDING INTO NORMED SPACES

In this section we study embedding arbitrary metrics into normed spaces. In our first result the distortion is prioritized according to the given ranking of the points in the metric.

THEOREM 8. *For any $p \in [1,\infty]$, $\epsilon > 0$, and any finite metric space (X,d) with priority ranking $X = (x_1,\dots,x_n)$, there exists an embedding of X into $\ell_p^{O(\log^2 n)}$ with priority distortion $O(\log j \cdot (\log\log j)^{(1+\epsilon)/2})$.*

We defer the proof to the full version. Our main result in the context of embedding to normed spaces is the following one, which has both prioritized distortion *and dimension*.

THEOREM 9. *For any $p \in [1,\infty]$, any fixed $\epsilon > 0$ and any metric space (X,d) on n points, there exists an embedding of X into $\ell_p^{O(\log^2 n)}$ with priority distortion $O\left(\log^{4+\epsilon} j\right)$, and prioritized dimension $O(\log^4 j)$.*

Proof overview.

The basic framework of this embedding appears at a first glance to be similar to that of Theorem 8, which is applying a variation of Bourgain's embedding, while sampling only from certain subsets S_i of the points. However, the crux here is that we need to ensure that high priority points will be mapped to the zero vector in the embeddings that "handle" the lower ranked points.

Recall that the coordinates of the embedding are given by distances to sets. The idea is the following: while creating the embedding for the points in S_i, we insert all the points with higher ranking (those in $S_0 \cup \cdots \cup S_{i-1}$) into every one of the randomly sampled sets. This will certify that the high ranked points are mapped to zero in every one of these coordinates. However, the analysis of the distortion no longer holds, as the sets are not randomly chosen. Fix some point $u \in S_i$ and $v \in X$. The crucial observation is that if none of the higher ranked points lie in certain neighborhoods around u and v (the size of these neighborhoods depends on $d(u,v)$), then we can still use the randomness of the selected sets to obtain some bound (albeit not as good as the standard embedding achieves). While if there exists a high ranked point nearby, say $z \in S_{i'}$ for some $i' < i$, then we argue that u, v should already have sufficient contribution from the embedding designed for $S_{i'}$. The formal derivation of this idea is captured in Lemma 1.

The calculation shows that the distortion guarantee for u, v deteriorates by a logarithmic factor for each i, that is, it is the product of the distortion bound for points in S_{i-1} multiplied by $O(\log |S_i|)$. This implies that the optimal size of S_i is *triple exponential* in i, which yields the best balance between the price paid due to the size of S_i and the product of the logarithms of $|S_0|,\dots,|S_{i-1}|$.

Lemma 1. Let $p \in [1,\infty]$ and $D \ge 1$. Given a metric space (X,d), two disjoint subsets $A, K \subseteq X$ where $|K| = k \ge 2$, and a non-expansive embedding $g : X \to \ell_p$ with contraction at most D for all pairs in $A \times X$, then there is a non-expansive embedding $f : X \to \ell_p^{O(\log^2 k)}$ such that the following properties hold:

1. For all $x \in A$, $f(x) = \vec{0}$.
2. For all $(x,y) \in K \times X$, $\|f(x) - f(y)\|_p \ge \frac{d(x,y)}{1000D\cdot\log k}$, or $\|g(x) - g(y)\|_p \ge \frac{d(x,y)}{2D}$.

We postpone the proof of Lemma 1 to Section 6.1, and prove Theorem 9 using the lemma.

PROOF OF THEOREM 9. Let $I = \lceil \log\log\log n \rceil$. Let $S_0 = \{x_1, x_2, x_3, x_4\}$, and for $1 \le i \le I$ let $S_i = \left\{ x_j \ : \ 2^{2^{2^{i-1}}} < j \le 2^{2^{2^i}} \right\}$. Also define $S_{<i} = \bigcup_{0\le k<i} S_k$.

The desired embedding $F : X \to \ell_p$ will be created by iteratively applying Lemma 1, each time using its output function f as part of the input for the next iteration. Formally, for each $0 \le i \le I$ apply Lemma 1 with parameters $A = S_{<i}$, $K = S_i$, $g = F^{(i-1)}$ and $D = 2^{2^i+5i^2}$, to obtain a map $f_i : X \to \ell_p$. The map $F^{(i)} : X \to \ell_p$ is defined as follows: $F^{(-1)} \equiv 0$, and $F^{(i)} = \bigoplus_{k=0}^{i} \alpha_k \cdot f_k$, where (α_k) is a sequence that ensures $F^{(i)}$ is non-expansive for all i. For concreteness, take $\alpha_k = \left(\frac{6}{\pi^2(k+1)^2}\right)^{1/p}$. The final embedding is defined by $F = F^{(I)}$.

Fix any pair $x, y \in X$. As f_i is non-expansive by Lemma 1, we obtain that F is non-expansive as well:

$$\begin{aligned} \|F(x) - F(y)\|_p^p &= \sum_{i=0}^{I} \alpha_i^p \cdot \|f_i(x) - f_i(y)\|_p^p \\ &\le \sum_{i=0}^{\infty} \frac{6}{\pi^2(i+1)^2} \cdot d(x,y)^p = d(x,y)^p \ . \end{aligned}$$

Next, we must show that for each $0 \le i \le I$, the embedding $F^{(i-1)}$ has contraction at most $2^{2^i+5i^2}$ for pairs in $S_{<i} \times X$, to comply with the requirement of Lemma 1. We prove this by induction on i, the base case for $i = 0$ holds trivially as $F^{(-1)}$ has no requirement on its contraction (since $S_{<0} = \emptyset$). Assume (for i) that $F^{(i-1)}$ has contraction at most $2^{2^i+5i^2}$ on pairs in $S_{<i} \times X$. For $i+1$, let $x \in S_{<i+1}$ and $y \in X$. Recall that $F^{(i)}$ is generated by applying Lemma 1 with $A = S_{<i}$, $K = S_i$, $g = F^{(i-1)}$, and $D = 2^{2^i+5i^2}$. Then the lemma returns f_i, and finally $F^{(i)} = g \oplus (\alpha_i \cdot f_i)$.

We may assume that $x \in S_i$, otherwise $g = F^{(i-1)}$ has the required contraction on x, y by the induction hypothesis. Applying condition (2) of the lemma: if it is the case that $\|g(x) - g(y)\|_p \ge d(x,y)/(2D)$, then clearly $2D < 2^{2^{i+1}+5(i+1)^2}$. The other case is that $\|f_i(x) - f_i(y)\|_p \ge \frac{d(x,y)}{1000D\cdot\log|S_i|}$. Since $\log|S_i| \le 2^{2^i}$ and $1/\alpha_i \le 2(i+1)^2$, the contraction of $F^{(i)}$ is at most the contraction of $\alpha_i \cdot f_i$, which is bounded by

$$\begin{aligned} \frac{1000D \cdot \log|S_i|}{\alpha_i} &\le 1000 \cdot 2^{2^i+5i^2} \cdot 2^{2^i} \cdot 2(i+1)^2 \\ &< 2^{2\cdot 2^i + 5i^2 + 2\log(i+1) + 11} \\ &< 2^{2^{i+1}+5(i+1)^2} \ . \end{aligned}$$

Observe that if $x = x_j \in S_i$ for some $j > 1$, then $2^{2^{i-1}} < \log j$, and thus the distortion of F for any pair containing x is at most $2^{2^{i+1}+5(i+1)^2} = O(\log^4 j) \cdot 2^{O((2+\log\log\log j)^2)} = O(\log^{4+\epsilon} j)$. Additionally, note that as the distortion of $F^{(I-1)}$ is at most $D = 2^{2^I+5I^2}$, the same argument suggests that the maximal distortion of $F = F^{(I)}$ for any pair is at most

$$\frac{1000D \cdot \log n}{\alpha_I} \le 1000 \cdot 2^{2^I+5I^2} \cdot \log n \cdot 2(I+1)^2 = O(\log^{3+\epsilon} n) \ .$$

Finally, let us bound the number of nonzero coordinates of the points. Recall that f_i maps X into $O(\log^2 |S_i|) \le O(2^{2^{i+1}})$ dimensions. Fix some $x = x_j$ for $j > 1$, and let i be such that $x_j \in S_i$. Note that $2^{2^{i-1}} < \log j$, so that $2^{2^{i+1}} < \log^4 j$. By Lemma 1, for every $i' > i$, $f_{i'}(x_j) = \vec{0}$, and the number of coordinates used by $F^{(i)}$ is at most

$$\sum_{k=0}^{i} O(2^{2^{k+1}}) = O(2^{2^{i+1}}) = O(\log^4 j) \ .$$

Since the dimension of f_I is at most $O(\log^2 n)$, we get that the total number of coordinates used by F is only

$$\begin{aligned} \sum_{k=0}^{I-1} O(2^{2^{k+1}}) + O(\log^2 n) &\le O(2^{2^{1+\log\log\log n}}) + O(\log^2 n) \\ &= O(\log^2 n) \ , \end{aligned}$$

which concludes the proof. □

6.1 Proof of Lemma 1

The basic approach is similar to Bourgain's: define the embedding by distances from subsets of K, sampled according to various densities. The main difference is that we insert all the points of A into each sampled set, to ensure $f(x) = \vec{0}$ for all $x \in A$. The standard analysis of Bourgain for a pair x, y, considers certain neighborhoods defined according to the density of points around x, y. We show that the analysis still works as long as no point of A is present in those neighborhoods. Thus we can obtain a contribution which is proportional to the distance of x, y to A (or to $d(x, y)$ if that distance is large). This motivates the following definition and lemma.

Definition 2. The γ-distance between x and y with respect to A is defined to be

$$\gamma_A(x, y) = \min\left\{\frac{d(x, y)}{2}, d(x, A), d(y, A)\right\} \ .$$

Lemma 2. Let $c = 24$. There exists a non-expansive embedding $\varphi : X \to \ell_p^{O(\log^2 k)}$, such that for all $z \in A$, $\varphi(z) = \vec{0}$, and for all $x, y \in K$,

$$\|\varphi(x) - \varphi(y)\|_p \ge \frac{\gamma_A(x, y)}{c \log k} \ .$$

We defer the proof of Lemma 2, and proceed first with the proof of Lemma 1. Define $h : X \to \mathbb{R}$ for $x \in X$ as $h(x) = d(x, A \cup K)$. Our embedding f is

$$f = \frac{\varphi \oplus h}{2^{1/p}} \ .$$

Since both φ and h are non-expansive and vanish on A, clearly f is non-expansive as well, and $f(z) = \vec{0}$ for any $z \in A$. It remains to show property (2) of the lemma. Fix any $x \in K$ and $y \in X$, and consider the following three cases:

Case 1: $d(\{x, y\}, A) \le \frac{d(x,y)}{4D}$. In this case we shall use the guarantees of the map g. Assume w.l.o.g that $z \in A$ is such that $d(y, z) \le \frac{d(x,y)}{4D}$. Then by the triangle inequality

$$d(x, z) \ge d(x, y) - d(y, z) \ge d(x, y) - \frac{d(x, y)}{4D} \ge \frac{3d(x, y)}{4} \ . \tag{6}$$

Now, using that g is non-expansive, and has contraction at most D for any pair in $A \times X$, we obtain that

$$\begin{aligned} \|g(x) - g(y)\|_p &\ge \|g(x) - g(z)\|_p - \|g(z) - g(y)\|_p \\ &\ge \frac{d(x, z)}{D} - d(z, y) \\ &\overset{(6)}{\ge} \frac{3d(x, y)}{4D} - \frac{d(x, y)}{4D} \\ &= \frac{d(x, y)}{2D} \ , \end{aligned}$$

which satisfies property (2).

Case 2: $d(\{x, y\}, A) > \frac{d(x,y)}{4D}$ and $d(y, K) \ge \frac{d(x,y)}{20cD \cdot \log k}$ (where $c = 24$ is the constant of Lemma 2). Here we shall use the map h for the contribution. Since $d(y, A) \ge d(x, y)/(4D)$, we have that $h(y) = d(y, A \cup K) \ge \frac{d(x,y)}{20cD \cdot \log k}$ and of course $h(x) = 0$, so that

$$\|f(x) - f(y)\|_p \ge \frac{|h(x) - h(y)|}{2} \ge \frac{d(x, y)}{40cD \cdot \log k} \ ,$$

as required.

Case 3: $d(\{x, y\}, A) > \frac{d(x,y)}{4D}$ and $d(y, K) < \frac{d(x,y)}{20cD \cdot \log k}$. In this case the function φ will yield the required contribution. Let $k_y \in K$ be such that $d(y, k_y) = d(y, K)$. Note that $d(k_y, A) \ge d(y, A) - d(y, k_y) \ge \frac{d(x,y)}{4D} - \frac{d(x,y)}{20cD \cdot \log k} \ge \frac{d(x,y)}{5D}$, and it follows that

$$\gamma_A(x, k_y) \ge \frac{d(x, y)}{5D} \ . \tag{7}$$

By Lemma 2, since f is non-expansive, and using another application of the triangle inequality, we conclude that

$$\begin{aligned} \|f(x) - f(y)\|_p &\ge \|f(x) - f(k_y)\|_p - \|f(y) - f(k_y)\|_p \\ &\ge \frac{\|\varphi(x) - \varphi(k_y)\|_p}{2} - d(y, k_y) \\ &\ge \frac{\gamma_A(x, k_y)}{2c \log k} - \frac{d(x, y)}{20cD \cdot \log k} \\ &\overset{(7)}{\ge} \frac{d(x, y)}{10cD \cdot \log k} - \frac{d(x, y)}{20cD \cdot \log k} \\ &= \frac{d(x, y)}{20cD \cdot \log k} \ . \end{aligned}$$

This concludes the proof of Lemma 1. It remains to validate Lemma 2, which is similar in spirit to the methods of [12, 21], and is deferred to the full version.

7. REFERENCES

[1] Ittai Abraham, Yair Bartal, Hubert T.-H. Chan, Kedar Dhamdhere, Anupam Gupta, Jon M. Kleinberg, Ofer Neiman, and Aleksandrs Slivkins. Metric embeddings with relaxed guarantees. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 83–100. IEEE Computer Society, 2005.

[2] Ittai Abraham, Yair Bartal, and Ofer Neiman. Embedding metrics into ultrametrics and graphs into spanning trees with constant average distortion. In *Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 502–511, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[3] Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. *Advances in Mathematics*, 228(6):3026 – 3126, 2011.

[4] Ittai Abraham and Shiri Chechik. Distance labels with optimal local stretch. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 52–63, 2014.

[5] Ittai Abraham and Cyril Gavoille. Object location using path separators. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, Denver, CO, USA, July 23-26, 2006*, pages 188–197, 2006.

[6] Baruch Awerbuch and David Peleg. Routing with polynomial communication-space trade-off. *SIAM J. Discrete Math.*, 5(2):151–162, 1992.

[7] Nikhil Bansal, Niv Buchbinder, Aleksander Madry, and Joseph Naor. A polylogarithmic-competitive algorithm for the k-server problem. In *Proceedings of the 52th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '08, pages 267 –276, Washington, DC, USA, 2011. IEEE Computer Society.

[8] Yair Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, FOCS '96, pages 184–193, Washington, DC, USA, 1996. IEEE Computer Society.

[9] Yair Bartal. On approximating arbitrary metrices by tree metrics. In *Proceedings of the 30th annual ACM symposium on Theory of computing*, STOC '98, pages 161–168, New York, NY, USA, 1998. ACM.

[10] Yair Bartal, Nathan Linial, Manor Mendel, and Assaf Naor. On metric ramsey-type phenomena. *Annals of Mathematics*, 162(2):643–709, 2005.

[11] J. Bourgain. The metrical interpretation of superreflexivity in Banach spaces. *Israel Journal of Mathematics*, 56(2):222–230, 1986.

[12] Jean Bourgain. On Lipschitz embedding of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52(1-2):46–52, 1985.

[13] Hubert T.-H. Chan, Michael Dinitz, and Anupam Gupta. Spanners with slack. In *Algorithms - ESA 2006, 14th Annual European Symposium, Zurich, Switzerland, September 11-13, 2006, Proceedings*, pages 196–207, 2006.

[14] Shiri Chechik. Approximate distance oracles with constant query time. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 654–663, New York, NY, USA, 2014. ACM.

[15] Michael Elkin and Seth Pettie. A linear-size logarithmic stretch path-reporting distance oracle for general graphs. In *Proceedings of the 26th Annual Symp. on Discrete Algorithms, San Diego, CA, USA*, 2015. To appear.

[16] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485–497, 2004.

[17] Cyril Gavoille, David Peleg, Stéphane Pérennes, and Ran Raz. Distance labeling in graphs. *J. Algorithms*, 53(1):85–112, 2004.

[18] Anupam Gupta. Steiner points in tree metrics don't (really) help. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '01, pages 220–227, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.

[19] Maleq Khan, Fabian Kuhn, Dahlia Malkhi, Gopal Pandurangan, and Kunal Talwar. Efficient distributed approximation algorithms via probabilistic tree embeddings. *Distributed Computing*, 25(3):189–205, 2012.

[20] Jon Kleinberg, Aleksandrs Slivkins, and Tom Wexler. Triangulation and embedding using small sets of beacons. *Journal of the ACM*, 56(6):32:1–32:37, September 2009.

[21] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[22] Richard J. Lipton and Robert E. Tarjan. A SEPARATOR THEOREM FOR PLANAR GRAPHS. *Siam Journal on Applied Mathematics*, 1979.

[23] Manor Mendel and Assaf Naor. Ramsey partitions and proximity data structures. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '06, pages 109–118, Washington, DC, USA, 2006. IEEE Computer Society.

[24] David Peleg. Proximity-preserving labeling schemes and their applications. In *Graph-Theoretic Concepts in Computer Science, 25th International Workshop, WG '99, Ascona, Switzerland, June 17-19, 1999, Proceedings*, pages 30–41, 1999.

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

[26] Liam Roditty, Mikkel Thorup, and Uri Zwick. Deterministic constructions of approximate distance oracles and spanners. In *Proceedings of the 32Nd International Conference on Automata, Languages and Programming*, ICALP'05, pages 261–272, Berlin, Heidelberg, 2005. Springer-Verlag.

[27] Gideon Schechtman and Adi Shraibman. Lower bounds for local versions of dimension reductions. *Discrete Computational Geometry*, 41:273–283, January 2009.

[28] Yuval Shavitt and Tomer Tankel. Big-bang simulation for embedding network distances in euclidean space. *IEEE/ACM Trans. Netw.*, 12(6):993–1006, 2004.

[29] Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 242–251, 2001.

[30] Mikkel Thorup and Uri Zwick. Compact routing schemes. In *SPAA*, pages 1–10, 2001.

[31] Mikkel Thorup and Uri Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005.

[32] Christian Wulff-Nilsen. Approximate distance oracles with improved query time. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, Louisiana, USA, January 6-8*, pages 539–549, 2013.