

Fast C-K-R Partitions of Sparse Graphs*

Manor Mendel

Chaya Schwob

Received: September 10, 2008; revised: June 22, 2009; published: July 16, 2009.

Abstract: We present fast algorithms for constructing probabilistic embeddings and approximate distance oracles in sparse graphs. The main ingredient is a fast algorithm for sampling the probabilistic partitions of Calinescu, Karloff, and Rabani in sparse graphs.

1 Introduction

Metric decompositions aim to partition the points of a metric space into blocks such that close-by points tend to be placed in the same block while distant pairs of points in different blocks. For most metric spaces, no straightforward interpretation of these goals exists.

One successful compromise is the notion of *probabilistic partition*. A Δ -bounded probabilistic partition is a probability distribution over partitions of the metric space, such that in every partition in the distribution, the diameters of the blocks are at most Δ , while “close-by” pairs of points are in the same block with “high” (or at least “non-negligible”) probability.

Probabilistic partitions first appeared, to the best of our knowledge,¹ in a paper of Linial and Saks [64], and publicized in the work of Bartal [10] on probabilistic embeddings. Calinescu, Karloff and Rabani [27] introduced the following probabilistic partition of metric spaces which we describe as an algorithm that samples a partition from the probability distribution.

We call the probabilistic partition P sampled by Algorithm 1, *Δ -bounded CKR partition*. Naïve implementations of Algorithm 1 take $\Omega(n^2)$ time for n -point metric spaces. It seems hard to break the $\Omega(n^2)$ barrier on the running time in general finite metric spaces. However, in many situations, the metric spaces we deal with come from the shortest-path metric on relatively sparse graphs. In those cases we can do better, as the following theorem shows.

*M. Mendel was partially supported by an ISF grant no. 221/07, a BSF grant no. 2006009, and a gift from Cisco research center. This work is part of the M.Sc. thesis of C. Schwob prepared in the Computer Science Division of the Open University of Israel.

¹Closely related notions of partitions appeared before, e.g. in [59].

Algorithm 1 CKR-Partition

Input: A finite metric space (X, ρ) , scale $\Delta > 0$

Output: Partition P of X

$\pi :=$ random permutation of X

$R :=$ random number in $[\frac{\Delta}{4}, \frac{\Delta}{2}]$

for $i = 1$ to $|X|$ **do**

$C_i := \{y \in X : \rho(y, x_{\pi(i)}) \leq R\} \setminus \bigcup_{j=1}^{i-1} C_j$

return $P := \{C_1, \dots, C_{|X|}\} \setminus \{\emptyset\}$

Theorem 1.1. *Suppose we are given a positive number $\Delta > 0$ and an undirected graph with positive edge weights $G = (X, E, \omega)$. Suppose G has n vertices and m edges, and let ρ denote the shortest-path metric in G . One can sample a Δ -bounded CKR partition of (X, ρ) in expected $O(m \log n + n \log^2 n)$ time.*

The sampling will be accomplished by Algorithm 2 (Section 3).

CKR partitions have found many algorithmic (as well as mathematical) applications, and we mention only few of them here. They were introduced as part of an approximation algorithm to the 0-extension problem [27, 40]. Fakcharoenphol, Rao and Talwar [41] used them to obtain an asymptotically tight probabilistic embedding into trees, which we call FRT-embedding. Probabilistic embeddings are used in many of the best known approximation and online algorithms as a reduction step from general metrics into tree metrics. Mendel and Naor [70] showed that FRT-embedding possesses a stronger embedding property, which they called “maximum gradient embedding”. Recently, Räcke [75] used FRT-embeddings to obtain hierarchical decompositions for congestion minimization in networks, and used them to give an $O(\log n)$ approximation algorithm for the minimum bisection problem and an $O(\log n)$ competitive online algorithm for the oblivious routing problem. Krauthgamer et. al [56] used CKR-partitions to give a new proof of Bourgain’s embedding theorem. Mendel and Naor [71] used them to obtain an asymptotically tight metric Ramsey theorem and approximate distance oracles.

The improved running time of the sampling of CKR partitions may improve the running time of many of their applications. In order to keep the paper short we work out the details of only two (related) applications of CKR partitions: FRT-embeddings, and approximate distance oracles based on CKR-partitions.

Probabilistic embedding into ultrametrics [5, 10].

An ultrametric ν on X is a metric which satisfies $\nu(x, z) \leq \max\{\nu(x, y), \nu(y, z)\}$, for every $x, y, z \in X$. A probabilistic embedding of a metric space (X, ρ) into ultrametrics with distortion D is a probability distribution Π over ultrametrics ν on X such that

1. For every $x, y \in X$, $\Pr_{\Pi}[\nu(x, y) \geq \rho(x, y)] = 1$.
2. For every $x, y \in X$, $\mathbb{E}_{\Pi}[\nu(x, y)] \leq D \cdot \rho(x, y)$.

FRT-embedding is a probabilistic embedding into ultrametrics with distortion $O(\log n)$ for every n -point metric space [41]. This bound is asymptotically tight for certain classes of finite metric spaces, such as graphs of high girth[10], grids [5], and expanders [63].

Approximate distance oracles.

An approximate distance oracle is a data structure with “compact” ($o(n^2)$) storage that answers (approximate) distance queries in a given n -point metric space in constant time. A simple counting argument over all bi-partite graphs shows that exact, and even 2.99 approximation is impossible when the storage is $o(n^2)$. The history of this problem is nicely summarized in [85]. In particular, Thorup and Zwick [85] gave an asymptotically tight trade-off between the approximation and the storage²: For every $k \in \mathbb{N}$, they constructed $(2k - 1)$ -approximate distance oracle requiring $O(kn^{1+1/k})$ storage, and answering queries in $O(k)$ time. Recently Mendel and Naor [71] presented different approximate distance oracles, based on CKR partitions. While those oracles do not give optimal approximation/storage trade-off,³ they answer distance queries in an absolute constant time, regardless of the approximation parameter.

Theorem 1.2. *Let $G = (X, E, \omega)$ be an n -vertex weighted graph with m edges, and let ρ be the shortest-path metric on X . Then*

1. *It is possible to sample from FRT-embedding of (X, ρ) in $O(m \log^3 n)$ expected time.*
2. *It is possible to construct in $O(mn^{1/k} \log^3 n)$ expected time an $O(k)$ -approximate distance oracle for (X, ρ) based on CKR partitions whose storage is $O(n^{1+1/k})$.*

For approximate distance oracles, it is also possible to improve the naïve construction time even when the metric is given as distance matrix, by first constructing a spanner of the metric with $o(n^2)$ edges, and then use the fast CKR partitions for sparse graphs on that spanner.

Theorem 1.3. *For n -point metric spaces given as distance matrix, it is possible to construct $O(k)$ -approximate distance oracle based on CKR partitions whose storage is $O(n^{1+1/k})$, in $O(n^2)$ expected time.*

We remark that a different probabilistic partition, developed by Bartal [11, 12] and Abraham et. al. [1], have properties similar to (and even stronger than) CKR partitions. However, we do not see an easy way to quickly obtain a sample from this distribution when the graph is sparse.

Further Results

The second named author presents in [81] an efficient PRAM algorithm for sampling CKR partitions and constructing approximate distance oracles in weighted graphs. The running time of the algorithm is $\text{polylog}(n)$ and the total work is $O(m \text{polylog}(n))$.

Outline of the paper.

After setting up in Section 2 the notation and reviewing the properties of CKR partitions, we prove Theorem 1.1 in Section 3.

²The lower bound on the approximation assumes a conjecture of Erdős about the number of edges possible in graph with a given number of vertices and a given girth, see [85].

³As reported in [71], oracles of size $O(n^{1+1/k})$ support approximation factor of $128k$ in the queries. While the constant 128 can be reduced by optimizing the parameters in the construction, it is unlikely to get below 8.

In many applications (and in particular, probabilistic embeddings and approximate distance oracles), probabilistic partitions are applied hierarchically, using an exponentially decreasing series of scales. This naïvely implies an added $O(\log \Phi)$ factor in the running time, where Φ is the spread⁴ of the metric. There is a standard technique that converts this $\log \Phi$ factor into a $\log n$ factor. However, we are not aware of a concrete implementation that satisfies the efficiency requirements needed in this paper. We therefore sketch the details of this technical step in Section 4. The specific applications examined in this paper, Theorem 1.2 and Theorem 1.3, are discussed in Section 5.

2 Preliminaries

For simplicity of the presentation, the model of computation we assume is a unit-cost, real-word RAM machine. In this model words can hold real numbers and arithmetic, comparison, and truncation operations take unit time. Our algorithms, however, do not take advantage of the unrealistic power of this model, and can also be presented in a more realistic computational models such as the unit cost floating-point word RAM model (cf. [50, Sec. 2.2]).

The diameter of a finite subset $Y \subseteq (X, \rho)$ is defined as $\text{diam}(Y) = \max\{\rho(x, y) : x, y \in Y\}$. For simplicity of the presentation, we assume that the given finite metric (X, ρ) has minimum non-zero distance 1, and diameter $\text{diam}(X) = \Phi$.

The (closed) ball around x at radius r is defined as $B_\rho(x, r) = \{y \in X : \rho(x, y) \leq r\}$. When ρ is clear from the context we may omit it from the notation. Given a partition P of X , and $x \in X$, we denote by $P(x)$ the block of P which contains x .

Δ -bounded CKR partitions have an obvious upper bound of Δ on the diameter of the blocks in the partition. The following is the padding property they enjoy.

Lemma 2.1 ([41, 71]). *Let P be a Δ -bounded CKR partition of the metric (X, ρ) . Then, for every $x \in X$, and $t \leq \Delta/8$,*

$$\Pr_{P \sim \mathcal{P}} [B(x, t) \subseteq P(x)] \geq \left(\frac{|B_X(x, \Delta/8)|}{|B_X(x, \Delta)|} \right)^{\frac{16t}{\Delta}}. \quad (2.1)$$

In this paper we define hierarchical partition of a metric space (X, ρ) as a sequence of $\lceil \log_8 \Phi \rceil + 2$ partitions $P_{-1}, P_0, \dots, P_{\lceil \log_8 \Phi \rceil}$ such that P_i is a partition of X at scale 8^i , and P_i is a refinement of P_j when $i \leq j$, i.e., for every $x \in X$, $P_i(x) \subseteq P_j(x)$. Given a sequence $(Q_j)_{j \geq -1}$ of partitions, where Q_j is 8^j -bounded partition of X , the common refinement of $(Q_j)_{j \geq -1}$ is a hierarchical partition $(P_j)_{j \geq -1}$ in which $P_j = \{\bigcap_{\ell \geq j} C_\ell : C_\ell \in Q_\ell\}$

By sampling stochastically independent CKR partitions at the different scales and then taking their common refinement, we obtain the following result.

Lemma 2.2 ([71]). *Fix a finite metric space (X, ρ) . Then there exists (efficiently sampleable) probability distribution \mathcal{H} over hierarchical partitions such that for every $x \in X$, and every $0 < \beta < 1/8$,*

$$\Pr_{(P_{-1}, \dots, P_{\lceil \log_8 \Phi \rceil}) \sim \mathcal{H}} \left[\forall k \geq -1, B(x, \beta 8^k) \subseteq P_k(x) \right] \geq |X|^{-16\beta}.$$

⁴The spread is the ratio between the diameter and the smallest non-zero distance in the metric

A finite ultrametric (X, ν) can be represented by a tree as follows.

Definition 2.3. An ultrametric tree (T, Γ) is a metric space whose elements are the leaves of a rooted finite tree T . Associated with every vertex $u \in T$ is a label $\Gamma(u) \geq 0$ such that $\Gamma(u) = 0$ iff u is a leaf of T . If a vertex u is a child of a vertex v then $\Gamma(u) \leq \Gamma(v)$. The distance between two leaves $x, y \in T$ is defined as $\Gamma(\mathbf{lca}(x, y))$, where $\mathbf{lca}(x, y)$ is the least common ancestor of x and y in T .

Every finite ultrametric can be represented by an ultrametric tree, and vice versa: the metric on ultrametric tree is a finite ultrametric. Hierarchical partition $\{P_k\}_{k=-1}^{\lceil \lg \phi \rceil}$ of (X, ρ) naturally corresponds to an ultrametric ν on X where $\nu(x, y) = 8^{\min\{j: P_j(x)=P_j(y)\}}$.

Let $G = (X, E, \omega)$ be an undirected positively weighted graph. Let $\rho : X \times X \rightarrow [0, \infty)$ be the shortest-path metric on G . We denote by $n = |X|$ the number of vertices, and by $m = |E|$ the number of edges. We assume an adjacency list representation of graphs.

The single source shortest paths in weighted undirected graphs problem [USSSP] is used as a subroutine in our algorithm. Given a weighted graph with n vertices and m edges, Dijkstra’s classical USSSP algorithm [35] with source w maintains for each vertex v an upper bound on the distance between w and v , $\delta(v)$. If $\delta(v)$ has not been assigned yet, it is interpreted as infinite. Initially, we just set $\delta(w) = 0$, and we have no visited vertices. At each iteration, we select an unvisited vertex u with the smallest finite $\delta(u)$, visit it, and relax all its edges. That is, for each incident edge $(u, v) \in E$, we set $\delta(v) \leftarrow \min\{\delta(v), \delta(u) + \omega(u, v)\}$. We continue until no vertex is left unvisited. Using Fibonacci heaps [44] or Bordal’s priority queues [25], Dijkstra’s algorithm is implemented in $O(m + n \lg n)$ time.

3 Fast CKR partitions

Given an undirected positively weighted graph $G = (X, E, \omega)$ with n vertices and m edges whose shortest path metric is denoted by ρ , and $\Delta > 0$, we show how to implement Algorithm 1 in $O(m \lg n + n \log^2 n)$ expected time.

First, we sample a random permutation π , which can be generated in linear time using several methods, e.g., Knuth Shuffle (see [22]). Next, we sample R uniformly⁵ in the range $[\frac{\Delta}{4}, \frac{\Delta}{2}]$.

We then use a variant of Dijkstra’s algorithm for computing the blocks. The algorithm performs $|X|$ iterations. In the i -th iteration, all vertices in $B_\rho(x_{\pi(i)}, R)$ not yet assigned to some block are put in C_i . In order to gain the improved running time of Theorem 1.1, we change Dijkstra’s algorithm to return the distance of a point v from $\pi(i)$ *only if* this distance is smaller than the distance of v from $\pi(j)$ for all $j < i$.

Technically, this is done as follows. Consider the i -th iteration and let $\delta(v)$ be the variable that holds the Dijkstra’s algorithm’s current estimate on the distance between $\pi(i)$ and v . In Dijkstra’s algorithm $\delta(v)$ is usually initialized to ∞ and then gradually decreases until u is extracted from the priority queue, at which point $\delta(v) = \rho(\pi(i), v)$. In the variant of Dijkstra’s algorithm used in Algorithm 2, $\delta(\cdot)$ are not reinitialized when the value of i is changed. This means that now at the end of the $(i - 1)$ -th iteration,

⁵A closer look on the analysis of the CKR partitions (see [71]) reveals that it is sufficient to sample R from discrete distribution having resolution of $\Delta/c \log n$, and therefore this step can be carried out in a “realistic” computational model such as the unit cost floating-point word RAM model.

$\delta(v) = \min_{j < i} \rho(\pi(j), v)$, which in turn means that an edge (u, w) is relaxed in the i -th iteration only when $\pi(i)$ is the closest center to both u , and w among $\pi(j)$, $j \leq i$. This dramatically reduces the number of relaxations being done, and does not hurt the correctness of the algorithm. The full details are given in Algorithm 2.

Algorithm 2 Graph-CKR-Partition

Input: Graph $G = (X, E, \omega)$, scale $\Delta > 0$

Output: Partition P of X

```

1: Generate random permutation  $\pi$  of  $X$ 
2: Sample a random  $R \in [\frac{\Delta}{4}, \frac{\Delta}{2}]$ 
3: for all  $v \in X$  do
4:      $\delta(v) := \infty$ 
5:      $P(v) := 0$ 
6: for  $i := 1$  to  $|X|$  do // Perform modified Dijkstra's alg starting from  $\pi(i)$ 
7:      $\delta(\pi(i)) := 0$ 
8:      $Q := \emptyset$  //  $Q$  is a priority queue with  $\delta$  being the key
9:      $w := \pi(i)$ 
10:    while  $\delta(w) \leq R$  do //  $w$  is visited now
11:        if  $P(w) = 0$  then
12:             $P(w) := i$ 
13:        for all  $u : (u, w) \in E$  do
14:            if  $\delta(u) > \delta(w) + \omega(u, w)$  then // Relax edges adjacent to  $w$ 
15:                 $\delta(u) := \delta(w) + \omega(u, w)$ 
16:                if  $u \notin Q$  then
17:                    Insert  $u$  into  $Q$ 
18:            Extract  $w \in Q$  with minimal  $\delta(w)$ 
19: return  $P$ 

```

Lemma 3.1. *After the i -th iteration of the loop on lines 6–18 of Algorithm 2,*

$$\delta(v) = \begin{cases} \min_{j \leq i} \rho(\pi(j), v) & \text{if } \min_{j \leq i} \rho(\pi(j), v) \leq R, \\ \infty & \text{otherwise.} \end{cases} \quad (3.1)$$

Sketch of a proof. Proof by induction on i . When $i = 1$, and as long $\delta(w) \leq R$ in the *While* loop of line 10, the algorithm behaves *exactly* as Dijkstra's algorithm and hence (3.1) is true for $i = 1$.

Assume inductively that (3.1) is correct for $i - 1$. If $\min_{j \leq i-1} \rho(\pi(j), v) \leq \rho(\pi(i), v)$, then clearly the i -th iteration will not change $\delta(v)$, and by the induction hypothesis we are done.

Assume now that $\min_{j \leq i-1} \rho(\pi(j), v) > \rho(\pi(i), v)$, and $\rho(\pi(i), v) \leq R$. Let $\pi(i) = v_0, v_1, \dots, v_\ell = v$ be a shortest-path between $\pi(i)$ and v . We claim that for every $t \in \{1, \dots, \ell\}$, $\min_{j \leq i-1} \rho(\pi(j), v_t) > \rho(\pi(i), v_t)$, since otherwise we had

$$\min_{j \leq i-1} \rho(\pi(j), v) \leq \min_{j \leq i-1} \rho(\pi(j), v_t) + \rho(v_t, v) \leq \rho(\pi(i), v_t) + \rho(v_t, v) = \rho(\pi(i), v).$$

Hence all the edges along the path $\pi(i) = v_0, \dots, v_\ell = v$ will be relaxed in the i -th iteration, and so in the end of the i -th iteration, $\delta(v) = \rho(\pi(i), v)$. \square

Proof of Theorem 1.1. We first prove the correctness of Algorithm 2, i.e., that $P(v) = \min \{i : \rho(\pi(i), v) \leq R\}$ for every $v \in V$. Let $i_0 = \min \{i : \rho(\pi(i), v) \leq R\}$. This means that $\min_{j < i_0} \rho(\pi(j), v) > R \geq \rho(\pi(i_0), v)$. By Lemma 3.1 at the beginning of the i_0 -th iteration, $\delta(v) = \infty$, and hence $P(v) = 0$ and by the end of the (i_0) -th iteration, $\delta(v) = \rho(\pi(i_0), v)$, and necessarily $P(v) = i_0$. Note that once $P(v)$ is set to a non-zero value, its value will not change.

We next bound the running time. we will show that every vertex is inserted into the queue $O(\log n)$ times in expectation, and every edge (u, v) of G undergoes $O(\log n)$ relaxations in expectation. Consider the non-increasing sequence $a_i = \min_{j \leq i} \rho(\pi(j), v)$. In the i -th iteration, $\delta(v)$ decreases if and only if $a_{i-1} > a_i$. Note that $a_{i-1} > a_i$ means that $\rho(\pi(i), v)$ is the minimum among $\{\rho(\pi(j), v) \mid j \leq i\}$, and the probability (over π) for this to happen is at most $1/i$. By linearity of the expectation, the expected number of rounds of the i -loop where $\delta(v)$ decreases (and hence v is inserted into the queue) is at most

$$\sum_{i=1}^n \frac{1}{i} \leq 1 + \ln n.$$

Furthermore, by another application of the linearity of expectation, the expected number of edge relaxations is at most

$$O\left(\sum_{v \in V} \ln n \cdot \deg(v)\right) = O(m \log n).$$

Using Fibonacci heaps [44] or Brodal's priority queues [25], the total running time of Algorithm 2 is $O(r + s \log n)$, where r is the number of relaxations, and s is the number of "insert" and "extract minimum" operations. In our cases $\mathbb{E}[r] = O(m \log n)$, and $\mathbb{E}[s] = O(n \log n)$. Therefore the total expected running time of Algorithm 2 is $O(m \log n + n \log^2 n)$. \square

4 Hierarchical Partitions

In this section we explain how to dispense with the $O(\log \Phi)$ factor in the naïve implementation of the hierarchical partitions, and replace it with $O(\log n)$. The method being used is standard. Similar arguments appeared previously, e.g., in [10, 50, 71, 70]. However, the context here is slightly different, and the designated time bound is $O(m \log^3 n)$, which is faster than the implementations we are aware of. While the argument is relatively straightforward, a full description of it is tedious to write and read. Instead we only sketch the implementation here. A complete description, including algorithmic implementation, appears in [81].

In a naïve implementation, the number of scales in which we sample CKR partitions is $\Theta(\lg \Phi)$. This leads to $O((n \log^2 n + m \log n) \log \Phi)$ bound on the expected running time. Here we develop an implementation having $O(m \log^3 n)$ expected running time. We define for each scale an appropriate quotient of the input graph. We then show that CKR partitions of those substitutive graph metrics retain the properties of CKR partitions on original metric. Using those quotients, not all scales need to be processed, and the total size of the quotient graphs in all processed scales is $O(m \lg n)$.

For $y, y' \subseteq X$, let $\rho(y, y') = \min \{ \rho(x, x') \mid x \in y, x' \in y' \}$. Given a partition Y of the space (X, ρ) we define the quotient metric ν on Y as

$$\nu(y, y') = \min \left\{ \sum_{j=1}^l \rho(y_{j-1}, y_j) : y_0, \dots, y_l \in Y, y_0 = y, y_l = y' \right\}.$$

Definition 4.1. A space (Y, ν) is called Δ -bounded quotient of an n -point metric space (X, ρ) if Y is a Δ -bounded partition of X , ν is a quotient metric on Y , and for every $x \in X$, $B_\rho(x, \Delta/n) \subseteq Y(x)$.

Note that a Δ -bounded quotient of n -point metric space exists: define a relation $x \sim x'$ if $\rho(x, x') \leq \Delta/n$, and take the transitive closure. The quotient subsets are the equivalence classes, and by the triangle inequality, the diameter of those equivalence classes is at most Δ .

The following lemma follows easily from Lemma 2.1, see the proof of [69, Lemma 5].

Lemma 4.2. Fix $\Delta > 0$, and let (Y, ν) be a $\frac{\Delta}{2}$ -bounded quotient of (X, ρ) . Let $\sigma : X \rightarrow Y$ be the natural projection, assigning each vertex $x \in X$ to its cluster $Y(x)$. Let L be a $(\Delta/2)$ -bounded CKR partition of Y .

Let P be the pullback of L under σ , i.e., $P = \{ \sigma^{-1}(A) \mid A \in L \}$. Then P is a Δ -bounded partition of X such that for every $0 < t \leq \Delta/16$ and every $x \in X$,

$$\Pr [B_\rho(x, t) \subseteq P(x)] \geq \left(\frac{|B_\rho(x, \Delta/16)|}{|B_\rho(x, \Delta)|} \right)^{\frac{32t}{\Delta}}. \quad (4.1)$$

and furthermore, if $t \leq \Delta/2n$, then

$$\Pr [B_\rho(x, t) \subseteq P(x)] = 1. \quad (4.2)$$

We define $G|_\Delta$ as the subgraph of G with edges of weight at most Δ and no isolated vertices.

Definition 4.3. Given a weighted graph $G = (X, E, \omega)$ and $\Delta > 0$. Define the graph $G|_\Delta = (X|_\Delta, E|_\Delta, \omega|_\Delta)$ as follows.

$$\begin{aligned} E|_\Delta &= \{ (u, v) \in E : u \neq v, \text{ and } \omega(u, v) \leq \Delta \}, \\ X|_\Delta &= \{ u \in X : \exists v \in X, (u, v) \in E|_\Delta \}, \text{ and} \\ \omega|_\Delta &= \omega|_{E|_\Delta}. \end{aligned}$$

Lemma 4.4. Given a weighted graph $G = (X, E, \omega)$, and $\Delta > 0$. Let L be a Δ -bounded CKR partition of $X|_\Delta$, using the metric induced by $G|_\Delta$. Then $P = L \cup \{ \{v\} : v \in X \setminus (X|_\Delta) \}$ is a Δ -bounded CKR partition of X , using the metric induced by G .

Proof. Let ρ be the shortest-path metric on G . Observe that when computing a Δ -bounded CKR partition of (X, ρ) no edge of weight larger than Δ is “used” by the Dijkstra’s algorithm for computing the balls, and therefore discarding them does not change the behavior of the algorithm. Also, for each $v \in X \setminus X|_\Delta$, $B_\rho(v, \Delta) = \{v\}$, i.e., in any Δ -bounded CKR partition of X , v will appear in a singleton subset. \square

Lemma 4.4 and Lemma 4.2 form the basis for dispensing with the dependence on the spread in the construction time. We next sketch the scheme we use. Denote the input graph $G = (X, E, \omega)$, $|X| = n$, $|E| = m$, and let ρ be the graph metric on G .

We first construct an ultrametric ν on V , represented by an ultrametric tree $H = (T, \Gamma)$ such that for every $u, v \in V$, $\rho(u, v) \leq \nu(u, v) \leq n\rho(u, v)$. H can be constructed in $O(m + n \log n)$ time using minimum spanning tree procedure, see [50, Section 3.2]. For a given $\Delta \geq 0$, and a leaf $v \in T$, denote by $\sigma_\Delta(v)$ the highest ancestor u of v for which $\Gamma(u) \leq \frac{\Delta}{2n}$. Using the level-ancestor data structure (cf. [19]) the tree T can be preprocessed in $O(n)$ time such that queries for $\sigma_\Delta(v)$ (given Δ , and v) are answered in $O(\log n)$ time. See [50, Section 3.5] for a similar supporting data structure.

Given $\Delta > 0$, define the weighted graph $G_{(\Delta)}$ as follows. $G_{(\Delta)} = (X_{(\Delta)}, E_{(\Delta)}, \omega_{(\Delta)})$ where,

$$\begin{aligned} X_{(\Delta)} &= \{\sigma_\Delta(v) : v \in X\}, \\ E_{(\Delta)} &= \{(\sigma_\Delta(u), \sigma_\Delta(v)) : (u, v) \in E, \sigma_\Delta(u) \neq \sigma_\Delta(v)\}, \\ \omega_{(\Delta)}(u, v) &= \min\{\omega(w, z) : \sigma_\Delta(w) = u, \sigma_\Delta(z) = v\}. \end{aligned}$$

Let $\rho_{(\Delta)}$ be the shortest-path metric on $G_{(\Delta)}$. Then, directly from the definitions, $(X_{(\Delta)}, \rho_{(\Delta)})$ is a $\frac{\Delta}{2}$ -bounded quotient of (X, ρ) .

For an integer $j \geq -1$ denote $G_j = (V_j, E_j, \omega_j)$ where $G_j = (G_{(8^j)}) \lfloor_{8^{j/2}}$. The following lemma gives an upper bound on the total size of the graphs G_j .

Lemma 4.5.

$$\sum_{j \geq -1} (|V_j| + |E_j|) = O(m \lg n).$$

Proof. Fix $(u, v) \in E$ and $j \geq -1$ such that $(\sigma_{8^j}(u), \sigma_{8^j}(v)) \in E_{(8^j)}$. By the definition of $E_{(8^j)}$, $\sigma_{8^j}(u) \neq \sigma_{8^j}(v)$. By the definition of $\omega_{(8^j)}$, $\omega(u, v) \geq \omega_{(8^j)}(u, v) \geq \frac{8^j}{2n}$. Also, $(\sigma_{8^j}(u), \sigma_{8^j}(v)) \in E_j$ if and only if $\omega_{(8^j)}(\sigma_{8^j}(u), \sigma_{8^j}(v)) \leq \frac{8^j}{2}$. So by the triangle inequality $\omega(u, v) \leq \omega_{(8^j)}(\sigma_{8^j}(u), \sigma_{8^j}(v)) + 8^j$, and hence $\omega(u, v) \leq 1.5 \cdot 8^j$. That is, each edge of G is represented in G_j only when $\omega(u, v) \in \left[\frac{8^j}{2n}, 1.5 \cdot 8^j\right]$. A total of $O(\log n)$ scales. By definition, G_j contains only non-isolated vertices, so $\forall j, |V_j| \leq 2|E_j|$. \square

Let **Processed** = $\{j \geq -1 : V_j \neq \emptyset\}$.

Lemma 4.6. *The set of graphs $(G_j)_{j \in \text{Processed}}$ can be constructed in $O(m \log^2 n)$ expected time.⁶*

Sketch of a proof. First we sort the edges in $E = \{e_1, \dots, e_m\}$ in non increasing order. Keep a “sliding window” $[i_L(t), i_R(t)]$, $i_L(t), i_R(t) \in \{1, \dots, m\}$, $t \in \{1, \dots, |\text{Processed}|\}$, as follows: Let $j_1 = \lceil \log_8 \Phi \rceil$. $i_L(1) = 1$, $i_R(1) = \max\{i : \omega(e_i) \geq 8^{j_1}/2n\}$. Assuming j_{t-1} is already defined, define $j_t = \max\{j < j_{t-1} : \exists i, 8^j \geq \omega(e_i) \geq 8^j/2n\}$, $i_L(t) = \min\{i : \omega(e_i) \leq 8^{j_t}\}$, and $i_R(t) = \max\{i : \omega(e_i) \geq 8^{j_t}/2n\}$. Note that $\{j_t\}_t = \text{Processed}$, and the definition gives $O(m)$ time algorithm for computing the sequences $(j_t)_t$,

⁶With a bit more care the running time can be improved to $O(m \log n)$. This improvement, however, will not improve the total construction time of the hierarchical partition.

$(i_L(t))_t$, and $(i_R(t))_t$. Constructing G_{j_t} can now be done in $O((i_R(t) - i_L(t) + 1) \log n)$ time, by observing that the set of vertices is

$$V_{j_t} = \{\sigma_{8^{j_t}}(u_i), \sigma_{8^{j_t}}(v_i) : i \in \{i_L(t), \dots, i_R(t)\}, (u_i, v_i) = e_i\},$$

and similarly the set of edges is

$$E_{j_t} = \{(\sigma_{8^{j_t}}(u_i), \sigma_{8^{j_t}}(v_i)) : i \in \{i_L(t), \dots, i_R(t)\}, (u_i, v_i) = e_i\}.$$

Another $\log n$ factor in the construction time comes from the $O(\log n)$ time needed for each query of the form “ $\sigma_\Delta(u)$ ”. Since $i_R(t) - i_L(t) + 1 = |E_{j_t}|$, by Lemma 4.5, $\sum_t (i_R(t) - i_L(t) + 1) = O(m \log n)$. \square

Next, we sample $(8^{j_t}/2)$ -bounded CKR partition L_{j_t} for each G_{j_t} . By Lemma 4.2, $(L_{j_t})_t$ (implicitly) represents CKR partitions of G in *all* scales. Using Theorem 1.1 and Lemma 4.5 computing $(L_{j_t})_t$ is done in $O(m \log^3 n)$ expected time.

Hierarchical partitions have an $O(n)$ storage representation. It is similar to an efficient ultrametric tree representation, such as the *nettree* in [50]. Using a rooted tree \mathcal{P} whose leaves correspond to the points of X , each internal vertex u has at least two children, and is labeled with a (logarithm of) scale, $s(u)$. The 8^j -bounded partition P_j is now defined as follows: For $x \in X$, $P_j(x)$ is the highest ancestor u of x in \mathcal{P} such that $s(u) \leq j$. Since the tree \mathcal{P} does not have vertices of degree 2, except maybe the root, its size is $O(n)$.

We are left to describe how to compute the the common refinement of the pullbacks of $(L_{j_t})_t$ as a hierarchical partition represented in the tree structure \mathcal{P} of the previous paragraph. This is done by top-down fashion as follows:

In the initialization step, \mathcal{P} is created as a rooted star whose root, r is labeled by 8^{j_1} , and its leaves correspond to $\{\sigma_{8^{j_1}}(u) : u \in X\}$.

Next, inductively assume that \mathcal{P} is a hierarchical partition of $\{\sigma_{8^{j_{t-1}}}(u) : u \in X\}$ corresponding to $\{L_{j_s} : s \leq t-1\}$. We refine \mathcal{P} to include L_{j_t} as follows:

- Replace the leaves of \mathcal{P} : Each $\sigma_{8^{j_{t-1}}}(u)$ is replaced by

$$\{\sigma_{8^{j_t}}(v) : v \in X, \sigma_{8^{j_t}}(v) \text{ is a descendant of } \sigma_{8^{j_{t-1}}}(u)\}.$$

This step is done in $O(|V_{j_t}|)$ time by simply starting from an “old leaf” $\sigma_{8^{j_{t-1}}}(u)$ as a vertex in T and descending in T to level $8^{j_t}/2n$.

- Next, incorporate L_{j_t} into the hierarchical partition in a straightforward way: Scan the leaves of \mathcal{P} , which are in V_{j_t} grouped by their parents. Fixing such a parent u whose children v_1, \dots, v_ℓ are all leaves, we partition v_1, \dots, v_ℓ to subsets $\{v_1, \dots, v_\ell\} \cap C : C \in L_{j_t}\}$. For every such subset of size 2 or more we define a new parent w (which will be a child of u) with the label 8^{j_t} .

Hence, the t -th iteration in the algorithm above is executed in $O(|V_{j_t}|)$ time, so the total time for constructing the common refinement is $O(m \log n)$.

5 Applications

Proof of the first part of Theorem 1.2. As observed in Section 2, hierarchical partitions correspond to ultrametrics. As shown in [41], when the partition in every scale is a CKR partition, the resulting distribution over ultrametrics is a probabilistic embedding with $O(\log n)$ distortion.⁷ The algorithm described in Section 4 samples a hierarchical partition (and hence an ultrametric) in $O(m \log^3 n)$ expected time. \square

Proof of the second part of Theorem 1.2. A point $x \in X$ is called β -padded in hierarchical partition $\mathcal{H} = (P_{-1}, \dots, P_{\log \Phi})$, if for every j , $B(x, \beta 8^j) \subset P_j(x)$.

The main part of constructing $O(\beta^{-1})$ -approximate distance oracle based on CKR partitions works as follows [71]: Set $X_0 = X$, and iteratively on $i = 0, 1, \dots$ do: Compute a hierarchical CKR partition \mathcal{H}_i of X_i , and obtain an ultrametric H_i from \mathcal{H}_i . Let Y_i be a set of β -padded points in \mathcal{H}_i that is found in Lemma 2.2. Set $X_{i+1} := X_i \setminus Y_i$, $i := i + 1$ and repeat until $X_i = \emptyset$. The set of ultrametrics $(H_i)_i$, together with some supporting data-structures constitute the approximate distance oracle.

By Lemma 2.2, $\mathbb{E}|Y_i| \geq |X_i|^{1-32\beta}$, the number of iterations until $X_i = \emptyset$ is in expectation at most $O(\beta^{-1} n^{O(\beta)})$, and hence the total storage is as claimed.

There are two issues in the construction of $(H_i)_i$ that we have not covered yet: First, the task is to sample a hierarchical partition of X_i which is only a subset of the vertices in the graph $G = (X, E, \omega)$. This is rather easy to handle by adapting the algorithms in Section 3 and Section 4 to work with subsets of the vertices.

The second issue is the computation of β -padded points. The β -padded points of a (single) Δ -bounded partition P of a weighted graph $G = (V, E, \omega)$ can be computed as follows: Add a new vertex s_0 . For every edge $(u, v) \in E$ such that $P(u) \neq P(v)$, add an edge (s_0, u) whose weight is $\omega(u, v)$. Execute Dijkstra's shortest path algorithm from s_0 , and delete all vertices at distance at most $\beta\Delta$ from s_0 . This can be implemented in $O(m + n \log n)$ time. Note that in the hierarchical partition if a point is not in V_j then it is padded at scale 8^j . Hence in order to compute a β padded point set in hierarchical partition, for every t , we cross out the points which are not 2β -padded in L_{j_t} . The remained points are β -padded in the pullbacks of $(L_{j_t})_t$ (as follows from Lemma 4.2) and hence also in the hierarchical partition. When implemented carefully, this can be done on every graph G_j in $O(|E_j| + |V_j| \log n)$, and by Lemma 4.5, in a total $O(m \log^2 n)$ time. \square

A t -spanner of a weighted graph $G = (V, E, \omega)$, is a subset of the edges $E' \subset E$ such that the shortest-path metric on $(V, E', \omega|_{E'})$ is at most t times the shortest-path metric on G . We need the following result.

Theorem 5.1 ([15]). *Let $G = (V, E, w)$ be a weighted graph with n vertices and m edges, and let $k \geq 1$ be an integer. A $(2k - 1)$ -spanner of with $O(kn^{1+1/k})$ edges can be computed in $O(km)$ expected time.*

⁷Technically, in [41] the hierarchical partition was built differently: instead of taking a CKR partition of the whole space in every scale, and then the common refinement, at each scale they took many CKR partitions, one for each block of the partition of the previous scale. This technicality is inconsequential for probabilistic embeddings. However, for the construction of approximate distance oracles, the approach of [41] does not seem to work since it does not have stochastically independent partitions in the different scales. See also [71].

Proof of Theorem 1.3. By Theorem 5.1, given an n -point metric space (X, ρ) , a 5-spanner H of (X, ρ) with $O(n^{4/3})$ edges can be constructed in $O(n^2)$ time. We next apply the second part of Theorem 1.2 whose running time is $o(n^2)$. \square

References

- [1] ITTAI ABRAHAM, YAIR BARTAL, AND OFER NEIMAN: Advances in metric embedding theory. In Kleinberg [55], pp. 271–286. 3
- [2] MILTON ABRAMOWITZ AND IRENE A. STEGUN: *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, NY, USA, 9th edition, 1964.
- [3] SUSANNE ALBERS AND TOMASZ RADZIK, editors. *Algorithms - ESA 2004, 12th Annual European Symposium, Bergen, Norway, September 14-17, 2004, Proceedings*, volume 3221 of *Lecture Notes in Computer Science*. Springer, 2004. 12
- [4] NOGA ALON, SHLOMO HOORY, AND NATHAN LINIAL: The moore bound for irregular graphs. 18(1):53–57, 2002.
- [5] NOGA ALON, RICHARD M. KARP, DAVID PELEG, AND DOUGLAS WEST: A graph-theoretic game and its application to the k -server problem. *SIAM J. Comput.*, 24(1):78–100, 1995. 2
- [6] INGO ALTHÖFER, GAUTAM DAS, DAVID P. DOBKIN, DEBORAH JOSEPH, AND JOSÉ SOARES: On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993.
- [7] MATTIAS ANDERSSON, JOACHIM GUDMUNDSSON, AND CHRISTOS LEVCOPOULOS: Approximate distance oracles for graphs with dense clusters. *Comput. Geom. Theory Appl.*, 37(3):142–154, 2007.
- [8] SANJEEV ARORA, JAMES R. LEE, AND ASSAF NAOR: Euclidean distortion and the sparsest cut. *J. Amer. Math. Soc.*, 21(1):1–21, 2008.
- [9] MIKHAIL J. ATALLAH AND SUSAN FOX: *Algorithms and Theory of Computation Handbook*. CRC Press, Inc., Boca Raton, FL, USA, 1998.
- [10] YAIR BARTAL: Probabilistic approximations of metric space and its algorithmic application. In *Proc. 37th Ann. IEEE Symp. Foundations of Computer Science (FOCS'96)*, pp. 183–193. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 1996. 1, 2, 7
- [11] YAIR BARTAL: Graph decomposition lemmas and their role in metric embedding methods. In Albers and Radzik [3], pp. 89–97. 3
- [12] YAIR BARTAL: Embedding finite metric spaces in low dimension. Technical report, Hebrew University of Jerusalem, 2005. 3

- [13] YAIR BARTAL, NATHAN LINIAL, MANOR MENDEL, AND ASSAF NAOR: On metric Ramsey type phenomena. *Ann. of Math.*, 162(2):643–709, 2005.
- [14] SURENDER BASWANA AND TELIKEPALLI KAVITHA: Faster algorithms for approximate distance oracles and all-pairs small stretch paths. In *Proc. 47th Ann. IEEE Symp. Foundations of Computer Science (FOCS'06)*, pp. 591–602. IEEE Computer Society, Washington, DC, USA, 2006.
- [15] SURENDER BASWANA AND SANDEEP SEN: A simple and linear time randomized algorithm for computing sparse spanners in weighted graphs. *Random Struct. Algorithms*, 30(4):532–563, 2007. [11](#)
- [16] RUDOLF BAYER: Symmetric binary b-trees: data structure and maintenance algorithms. *Acta Inf.*, 1:290–306, 1972.
- [17] MICHAEL BEN-OR: Lower bounds for algebraic computation trees. In *Proc. 15th Ann. ACM Symp. Theory of computing (STOC'83)*, pp. 80–86. ACM, New York, NY, USA, 1983.
- [18] MICHAEL A. BENDER AND MARTIN FARACH-COLTON: The LCA problem revisited. In *Proc. 4th Latin American Conf. in Theoretical Informatics (LATIN'00)*, pp. 88–94. Springer, Berlin, Germany, 2000.
- [19] MICHAEL A. BENDER AND MARTÍN FARACH-COLTON: The level ancestor problem simplified. *Theor. Comput. Sci.*, 321(1):5–12, 2004. [9](#)
- [20] OMER BERKMAN AND UZI VISHKIN: Recursive $*$ -tree parallel data structure. *SIAM J. Comput.*, 22(2):221–242, 1993.
- [21] OMER BERKMAN AND UZI VISHKIN: Finding level-ancestors in trees. *J. Comput. Syst. Sci.*, 48(2):214–230, 1994.
- [22] DAVID BERMAN AND M. S. KLAMKIN: A reverse card shuffle. *SIAM Review*, 18(3):491–492, 1976. [5](#)
- [23] JEAN BOURGAIN: On lipschitz embedding of finite metric spaces in hilbert spaces. *Israel J. Math.*, 54:46–52, 1985.
- [24] RICHARD P. BRENT: The parallel evaluation of general arithmetic expressions. *J. ACM*, 21(2):201–206, 1974.
- [25] GERH STØLTING BRODAL: Worst-case efficient priority queues. In *SODA '96: Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*, pp. 52–58, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics. [5](#), [7](#)
- [26] SERGIO CABELLO: Many distances in planar graphs. In *Proc. 17th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA'06)*, pp. 1213–1220. ACM, New York, NY, USA, 2006.

- [27] GRUIA CALINESCU, HOWARD KARLOFF, AND YUVAL RABANI: Approximation algorithms for the 0-extension problem. In *Proc. 12th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA'01)*, pp. 8–16. SIAM, Philadelphia, PA, USA, 2001. [1](#), [2](#)
- [28] SHUCHI CHAWLA, ANUPAM GUPTA, AND HARALD RÄCKE: Embeddings of negative-type metrics and an improved approximation to generalized sparsest cut. *ACM Trans. Algorithms*, 4(2):Art. 22, 18, 2008.
- [29] EDITH COHEN: Using selective path-doubling for parallel shortest-path computations. *J. Alg.*, 22:30–56, 1997.
- [30] EDITH COHEN: Fast algorithms for constructing t-spanners and paths with stretch t. *SIAM J. Comput.*, 28(1):210–236, 1999.
- [31] EDITH COHEN: Polylog-time and near-linear work approximation scheme for undirected shortest paths. *J. ACM*, 47(1):132–166, 2000.
- [32] EDITH COHEN AND URI ZWICK: All-pairs small-stretch paths. In *Proc. 8th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA'97)*, pp. 93–102. SIAM, Philadelphia, PA, USA, 1997.
- [33] RICHARD COLE: Parallel merge sort. *SIAM J. Comput.*, 17(4):770–785, 1988.
- [34] RICHARD COLE, PHILIP N. KLEIN, AND ROBERT ENDRE TARJAN: Finding minimum spanning forests in logarithmic time and linear work using random sampling. In *Proc. 8th Ann. ACM Symp. on Parallel Algorithms and Architectures (SPAA'96)*, pp. 243–250. ACM, New York, NY, USA, 1996.
- [35] E. W. DIJKSTRA: A note on two problems in connexion with graphs. *Numer. Math.*, 1:269–271, 1959. [5](#)
- [36] EDSGER W. DIJKSTRA: A note on two problems in connexion with graphs. *Numer. Math*, 1:269–271, 1959.
- [37] JAMES R. DRISCOLL, HAROLD N. GABOW, RUTH SHRAIRMAN, AND ROBERT E. TARJAN: Relaxed heaps: an alternative to fibonacci heaps with applications to parallel computation. *Commun. ACM*, 31(11):1343–1354, 1988.
- [38] MICHAEL ELKIN: Computing almost shortest paths. In *Proc. 20th Ann. ACM Symp. Principles of Distributed Computing (PODC'01)*, pp. 53–62. ACM, New York, NY, USA, 2001.
- [39] PAUL ERDŐS: Extremal problems in graph theory. In *Proc. Symp. Theory of Graphs and its Applications (smolenice, 1963)*, pp. 29–36. Publ. House Czechoslovak Acad. Sci., Prague, Czechoslovak, 1964.
- [40] JITTAT FAKCHAROENPHOL, CHRIS HARRELSON, SATISH RAO, AND KUNAL TALWAR: An improved approximation algorithm for the 0-extension problem. In *Proc. 14th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA'03)*, pp. 257–265. ACM, New York, NY, USA, 2003. [2](#)

- [41] JITTAT FAKCHAROENPHOL, SATISH RAO, AND KUNAL TALWAR: A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. System Sci.*, 69(3):485–497, 2004. [2](#), [4](#), [11](#)
- [42] WILLIAM FELLER: *An Introduction to Probability Theory and its Applications*. Volume 1. Wiley, New York, NY, USA, 3th edition, 1968.
- [43] STEVEN FORTUNE: Robustness issues in geometric algorithms. In *Proc. 1st Workshop Applied Computational Geometry: Towards Geometric Engineering (WACG'96)*. Springer, Berlin, Germany, 1996.
- [44] MICHAEL L. FREDMAN AND ROBERT ENDRE TARJAN: Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, 1987. [5](#), [7](#)
- [45] NAVEEN GARG, GORAN KONJEVOD, AND R. RAVI: A polylogarithmic approximation algorithm for the group Steiner tree problem. *J. Algorithms*, 37(1):66–84, 2000. Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (San Francisco, CA, 1998).
- [46] JOACHIM GUDMUNDSSON, CHRISTOS LEVCOPOULOS, GIRI NARASIMHAN, AND MICHEL SMID: Approximate distance oracles for geometric graphs. In *Proc. 13th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA'02)*, pp. 828–837. SIAM, Philadelphia, PA, USA, 2002.
- [47] ANUPAM GUPTA, MOHAMMAD T. HAJIAGHAYI, AND HARALD RÄCKE: Oblivious network design. In *Proc. 17th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA'06)*, pp. 970–979. ACM, New York, NY, USA, 2006.
- [48] TORBEN HAGERUP: Fast parallel generation of random permutations. In *Proc. 18th Int. Coll. Automata, Languages, and Programming (ICALP'91)*, pp. 405–416. Springer, Berlin, Germany, 1991.
- [49] JEFFERSON Y. HAN, VICTOR PAN, AND JOHN REIF: Efficient parallel algorithms for computing all pair shortest paths in directed graphs. In *Proc. 4th Ann. ACM Symp. on Parallel Algorithms and Architectures (SPAA'92)*, pp. 353–362. ACM, New York, NY, USA, 1992.
- [50] SARIEL HAR-PELED AND MANOR MENDEL: Fast construction of nets in low dimensional metrics, and their applications. *SIAM J. Comput.*, 35:1148–1184, 2006. [4](#), [7](#), [9](#), [10](#)
- [51] DOV HAREL AND ROBERT ENDRE TARJAN: Fast algorithms for finding nearest common ancestors. *SIAM J. Comput.*, 13(2), 1984.
- [52] JOSEPH JÁJÁ: *An Introduction to Parallel Algorithms*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1992.
- [53] DAVID R. KARGER, DAPHNE KOLLER, AND STEVEN J. PHILLIPS: Finding the hidden path: time bounds for all-pairs shortest paths. *SIAM J. Comput.*, 22(6):1199–1217, 1993.
- [54] DAVID R. KARGER, NOAM NISAN, AND MICHAL PARNAS: Fast connected components algorithms for the erew pram. *SIAM J. Comput.*, 28(3):1021–1034, 1998.

- [55] JON M. KLEINBERG, editor. *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*. ACM, 2006. 12
- [56] ROBERT KRAUTHGAMER, JAMES R. LEE, MANOR MENDEL, AND ASSAF NAOR: Measured descent: A new embedding method for finite metrics. In *Proc. 45th Ann. IEEE Symp. Foundations of Computer Science (FOCS'04)*, pp. 434–443. IEEE Computer Society, Washington, DC, USA, 2004. 2
- [57] F. LAZEBNIK, V. A. USTIMENKO, AND A. J. WOLDAR: A characterization of the components of the graphs $d(k, q)$. In *Proc. 6th Conf. Formal Power Series and Algebraic Combinatorics (FPSAC'96)*, pp. 271–283. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 1996.
- [58] FELIX LAZEBNIK, VASILY A. USTIMENKO, AND ANDREW J. WOLDAR: A new series of dense graphs of high girth. *Bull Amer. Math. Soc.*, 32(1):73–79, 1995.
- [59] TOM LEIGHTON AND SATISH RAO: Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999. 1
- [60] CHARLES E. LEISERSON AND BRUCE M. MAGGS: Communication-efficient parallel algorithms for distributed random-access machines. *Algorithmica*, 3:53–77, 1988.
- [61] CHEN LI, SYLVAIN PION, AND CHEE K. YAP: Recent progress in exact geometric computation. *J. of Logic and Algebraic Programming*, 64:85–111, 2005.
- [62] CHIH-CHUNG LIN AND RUEI-CHUAN CHANG: On the dynamic shortest path problem. *J. Inf. Process.*, 13(4):470–476, 1990.
- [63] NATHAN LINIAL, ERAN LONDON, AND YURI RABINOVICH: The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. 2
- [64] NATHAN LINIAL AND MICHAEL SAKS: Low diameter graph decompositions. *Combinatorica*, 13(4):441–454, 1993. 1
- [65] RICHARD J. LIPTON AND ROBERT ENDRE TARJAN: A planar separator theorem. *SIAM J. Comput.*, 36(2):177–189, 1979.
- [66] ALEXANDER LUBOTZKY, ROB PHILLIPS, AND PETER SARNAK: Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.
- [67] MARK OVERMARS MARK DE BERG, MARK VAN KREVELD AND OTFRIED SCHWARZKOPF: *Computational Geometry - Algorithms and Applications*. Springer, Berlin, Germany, 1997.
- [68] JIRI MATOUŠEK: On the distortion required for embedding finite metric space into normed spaces. *Israel J. Math.*, 93:333–344, 1996.
- [69] MANOR MENDEL AND ASSAF NAOR: Maximum gradient embeddings and monotone clustering. 8

- [70] MANOR MENDEL AND ASSAF NAOR: Maximum gradient embeddings and monotone clustering. In *Proc. Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 10th Int. Workshop (APPROX'07) and 11th Int. Workshop (RANDOM'07)*, pp. 242–256. Springer, Berlin, Germany, 2007. [2](#), [7](#)
- [71] MANOR MENDEL AND ASSAF NAOR: Ramsey partitions and proximity data structures. *J. European Math. Soc.*, 9(2):253–275, 2007. [2](#), [3](#), [4](#), [5](#), [7](#), [11](#)
- [72] GARY L. MILLER AND JOHN H. REIF: Parallel tree contraction and its application. In *Proc. 26th Ann. IEEE Symp. Foundations of Computer Science (FOCS'85)*, pp. 478–489. IEEE Computer Society, Washington, DC, USA, 1985.
- [73] DAVID PELEG AND JEFFREY D. ULLMAN: An optimal synchronizer for the hypercube. *SIAM J. Comput.*, 18(4):740–747, 1989.
- [74] MICHAEL CLAUSEN PETER BUERGISSEER AND AMIN SHOKROLLAHI: *Algebraic Complexity Theory*. Springer, Berlin, Germany, 1996.
- [75] HARALD RÄCKE: Optimal hierarchical decompositions for congestion minimization in networks. In *Proc. 40th Ann. ACM Symp. Theory of computing (STOC'08)*, pp. 255–264. ACM, New York, NY, USA, 2008. [2](#)
- [76] JOHN H. REIF: *Synthesis of Parallel Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [77] LIAM RODITTY, MIKKEL THORUP, AND URI ZWICK: Deterministic constructions of approximate distance oracles and spanners. In *Proc. 32th Int. Coll. Automata, Languages, and Programming (ICALP'05)*, pp. 261–272. Springer, Berlin, Germany, 2005.
- [78] LIAM RODITTY AND URI ZWICK: Dynamic approximate all-pairs shortest paths in undirected graphs. In *Proc. 45th Ann. IEEE Symp. Foundations of Computer Science (FOCS'04)*, pp. 499–508. IEEE Computer Society, Washington, DC, USA, 2004.
- [79] BARUCH SCHIEBER AND UZI VISHKIN: On finding lowest common ancestors: simplification and parallelization. *SIAM J. Comput.*, 17(6):1253–1262, 1988.
- [80] ARNOLD SCHÖNHAGE: On the power of random access machines. In *Proc. 6th Int. Coll. Automata, Languages, and Programming (ICALP'79)*, pp. 520–529. Springer, Berlin, Germany, 1979.
- [81] CHAYA SCHWOB: Ramsey partitions based approximate distance oracles. Master's thesis, The Open University of Israel, Ra'anana, Israel, 2008. available at http://sites.google.com/site/mendelma/Home/students/RamseyPartitionsBasedADO_Final.pdf?attredirects=0. [3](#), [7](#)
- [82] DAVID STEVENSON: *ANSI/IEEE 754-1985, Standard for Binary Floating-Point Arithmetic*. IEEE Computer Society, Washington, DC, USA, 1985.

- [83] MIKKEL THORUP: Undirected single-source shortest paths with positive integer weights in linear time. *J. ACM*, 46(3):362–394, 1999.
- [84] MIKKEL THORUP: Floats, integers, and single source shortest paths. *J. Algorithms*, 35(2):189–201, 2000.
- [85] MIKKEL THORUP AND URI ZWICK: Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005. 3
- [86] JOSEPH F. TRAUB: A continuous model of computation. *Phys. Today*, May:39–43, 1999.
- [87] PETER VAN EMDE BOAS: Machine models and simulation. In *Handbook of Theoretical Computer Science*, volume A, pp. 1–66. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 1990.
- [88] VIRGINIA VASSILEVSKA, RYAN WILLIAMS, AND RAPHAEL YUSTER: All-pairs bottleneck paths for general graphs in truly sub-cubic time. In *Proc. 39th Ann. ACM Symp. Theory of computing (STOC’07)*, pp. 585–589. ACM, New York, NY, USA, 2007.
- [89] URI ZWICK: Exact and approximate distances in graphs—a survey. In *Proc. 9th Ann. European Symp. Algorithms (ESA’01)*, pp. 33–48. Springer, Berlin, Germany, 2001.
- [90] URI ZWICK: A slightly improved sub-cubic algorithm for the all pairs shortest paths problem with real edge lengths. *Algorithmica*, 46(2):181–192, 2006.

AUTHORS

Manor Mendel
Computer Science Division
The Open University of Israel
mendelma [at] gmail [dot] com

Chaya Schwob
Computer Science Division
The Open University of Israel
cschwob [at] nds [dot] com