

Relaxed spanners for directed disk graphs

D. Peleg^{*}

L. Roditty[†]

Abstract

Let (V, δ) be a finite metric space, where V is a set of n points and δ is a distance function defined for these points. Assume that (V, δ) has a doubling dimension d and assume that each point $p \in V$ has a disk of radius $r(p)$ around it. The disk graph that corresponds to V and $r(\cdot)$ is a *directed* graph $I(V, E, r)$, whose vertices are the points of V and whose edge set includes a directed edge from p to q if $\delta(p, q) \leq r(p)$. In [10] we presented an algorithm for constructing a $(1 + \epsilon)$ -spanner of size $O(n\epsilon^{-d} \log M)$, where M is the maximal radius $r(p)$. The current paper presents two results. The first shows that the spanner of [10] is essentially optimal, i.e., for metrics of constant doubling dimension it is not possible to guarantee a spanner whose size is independent of M . The second result shows that by slightly relaxing the requirements and allowing a small augmentation of the radius assignment, considerably better spanners can be constructed. In particular, we show that if it is allowed to use edges of the disk graph $I(V, E, r_{1+\epsilon})$, where $r_{1+\epsilon}(p) = (1 + \epsilon) \cdot r(p)$ for every $p \in V$, then it is possible to get a $(1 + \epsilon)$ -spanner of size $O(n/\epsilon^d)$ for $I(V, E, r)$. Our algorithm is simple and can be implemented efficiently.

1 Introduction

This paper concerns efficient constructions of spanners for disk graphs, an important family of directed graphs. A *spanner* is essentially a skeleton of the graph, namely, a sparse spanning subgraph that faithfully represents distances. Formally, a subgraph H of a graph G is a t -spanner of G if $\delta_H(u, v) \leq t \cdot \delta_G(u, v)$ for every two nodes u and v , where $\delta_{G'}(u, v)$ denotes the distance between u and v in G' . We refer to t as the *stretch factor* of the spanner. Graph spanners have received considerable attention over the last two decades, and were used implicitly or explicitly as key ingredients of various distributed applications. It is known how to efficiently construct a $(2k - 1)$ -spanner with $O(n^{1+1/k})$ edges for every weighted undirected graph, and this size-stretch tradeoff is conjectured to be tight. Baswana and Sen [1] presented a linear time randomized algorithm for computing such a spanner. In directed graphs, however, the situation is different. No such general size-stretch tradeoff can exist, as indicated by considering the example of a directed bipartite graph G in which all the edges are directed from one side to the other; clearly, the only spanner of G is G itself, as any spanner for G must contain every edge.

The main difference between undirected and directed graphs is that in undirected graphs the distances are symmetric, that is, a path of a certain length from u to v can be used also from v to u . In directed graphs, however, the existence of a path from u to v does not imply anything on the distance in the opposite direction from v to u . Hence, in order to obtain a spanner for a directed graph one must impose some restriction either on the graph or on its distances. In order to bypass the problem of asymmetric distances of directed graphs, Cowen and Wagner [7] introduced the notion of *roundtrip distances* in which

^{*}Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel, email: david.peleg@weizmann.ac.il. Supported in part by grants from the Minerva Foundation and the Israel Ministry of Science.

[†]Department of Computer Science, Bar-Ilan University, Ramat-Gan 52900, Israel, email: liamr@macs.biu.ac.il

the distance between u and v is composed of the shortest path from u to v plus the shortest path from v to u . It is easy to see that under this definition distances are symmetric also in directed graphs. It is shown by Cowen and Wagner [7] and later by Roditty, Thorup and Zwick [8] that methods of path approximations from undirected graphs can work using more ideas also in directed graphs when roundtrip distances are considered. Bollobás, Coppersmith and Elkin [2] introduced the notion of *distance preservers* (i.e., a subgraph is a D -preserver if for every two vertices of distance at least D it preserves their exact distance) and showed that they exist also in directed graphs.

In [10] we presented a spanner construction for directed graphs without symmetric distances. The restriction that we imposed on the graph was that it must be a disk graph. More formally, let (V, δ) be a finite metric space of doubling dimension d , where V is a set of n points and δ is a distance function defined for these points. A metric is said to be of *doubling dimension* d if a ball with radius r can be covered by at most a 2^d of balls of radius $r/2$. Every point $p \in V$ is assigned with a radius $r(p)$. The disk graph that corresponds to V and $r(\cdot)$ is a directed graph $I(V, E, r)$, whose vertices are the points of V and whose edge set includes a directed edge from p to q if q is inside the disk of p , that is, $\delta(p, q) \leq r(p)$. In [10] we presented an algorithm for constructing a $(1 + \epsilon)$ -spanner with size $O(n\epsilon^{-d} \log M)$, where M is the maximal radius. In the case that we remove the radius restriction the resulted graph is the complete undirected graph where the weight of every edge is the distance between its endpoint. In such a case it is possible to create $(1 + \epsilon)$ -spanners of size $O(n/\epsilon^d)$, see [6], [4] and [11] for more details. Moreover, when the radii are all the same and the graph is the unit disk graph then it is also possible to create $(1 + \epsilon)$ -spanners of size $O(n/\epsilon^d)$, see [5], [10].

As a result of that, a natural question to ask is whether a spanner size of $O(n\epsilon^{-d} \log M)$ in the case of directed disk graph is indeed the best possible or maybe it is possible to get a spanner of size $O(n/\epsilon^d)$ as in the cases of the complete graph and the unit disk graph. For the case of the Euclidean metric space, the answer turns out to be positive; a simple modification of the Yao graph construction [13]¹ yields a directed spanner of size $O(n/\epsilon^d)$. However, the question remains for more general metric spaces, and in particular for the important family of metric spaces of bounded doubling dimension.

In this paper we provide an answer for this question. We show that our construction from [10] is essentially optimal by providing a metric space with a constant doubling dimension and a radius assignment whose corresponding disk graph has $\Omega(n^2)$ edges and none of its edges can be removed. (This does not contradict our spanner construction from [10] as the maximal radius in that case is $\Theta(2^n)$ and hence $\log M = n$.)

This (essentially negative) optimality result motivates our main interest in the current paper, which focuses on attempts to slightly relax the assumptions of the model, in order to obtain sparse spanner constructions. Indeed, it turns out that such sparser spanner constructions are feasible under a suitably relaxed model. Specifically, we demonstrate the fact that if a small augmentation of the radius assignment is allowed, then a $(1 + \epsilon)$ -spanner of size $O(n/\epsilon^d)$ is attainable. More formally, we show that if we are allowed to use edges of the disk graph $I(V, E, r_{1+\epsilon})$, where $r_{1+\epsilon}(p) = (1 + \epsilon) \cdot r(p)$ for every $p \in V$, then it is possible to get a $(1 + \epsilon)$ -spanner of size $O(n/\epsilon^d)$ for the original disk graph $I(V, E, r)$. This approach is similar in its nature to the notation of *emulators* introduced by Dor, Halperin and Zwick [3]. An emulator of a graph may use any edge that does not exist in the graph in order to approximate its distances. It was used in the context of spanners with an additive stretch.

Disk graphs and unit disk graphs. The main application of disk graph spanners is for topology control in the wireless ad hoc network model. In this model the power required for transmitting from p to q is commonly taken to be $\delta(p, q)^\alpha$, where $\delta(p, q)$ denotes the distance between p and q and α is

¹Each point p inserts to the spanner an incoming edge from each of the cones (for some small value of Θ that sets the number of cones). From each cone p picks the incoming edge whose other endpoint is the closest point to p from that cone.

a constant typically assumed to be between 2 and 4. Most of the ad hoc network literature makes the assumption that the transmission range of all nodes is identical, and consequently represents the network by a *unit disk graph* (UDG), namely, a graph in which two nodes p, q are adjacent if their distance satisfies $\delta(p, q) \leq 1$. A unit disk graph can have as many as $O(n^2)$ edges.

There is an extensive body of literature on spanners of unit disk graphs. Gao et al. [5], Wang and Yang-Li [12] and Yang-Li et al. [9] considered the restricted Delaunay graph, whose worst-case stretch is constant (larger than $1 + \epsilon$). In [10] we showed that any $(1 + \epsilon)$ -geometric spanner can be turned into a $(1 + \epsilon)$ -UDG spanner.

Disk graphs are a natural generalization of unit disk graphs, that provide an intermediate model between the complete graph and the unit disk graph. Our size efficient spanner construction for disk graphs whose radii are allowed to be slightly larger shows that considerably improved topology can be built for a network when the problem is slightly relaxed.

Our result has both practical and theoretical implications. From a practical point of view it shows that, in certain scenarios, extending the transmission radii even by a small factor can significantly improve the overall quality of the network topology. The result is also very intriguing from a theoretical standpoint, as to the best of our knowledge, our relaxed spanner is the first example of a spanner construction for directed graphs that enjoys the same properties as the best constructions for undirected graphs. (As mentioned above, it is easy to see that for general directed graphs, it is not possible to have an algorithm that given any directed graph produces a sparse spanner for it.) In that sense, our result can be viewed as a significant step towards gaining a better understanding for some of the fundamental differences between directed and undirected graphs. Our result also opens several new research directions in the relaxed model of disk graphs. The most obvious research questions that arise are whether it is possible to obtain other objects that are known to exist in undirected graphs, such as compact routing schemes and distance oracles, for disk graphs as well.

The rest of this paper is organized as follows. In the next section we present a metric space of constant doubling dimension in which no edge can be removed from its corresponding disk graph. Section 3 first describes a simple variant of our construction from [10], and then uses it together with new ideas in order to obtain our new relaxed construction. Finally, in Section 4 we present some concluding remarks and open problems.

2 Optimality of the spanner construction

In this section we build a disk graph G with $2n$ vertices and $\Omega(n^2)$ edges that is non-sparsifiable, namely, whose only spanner is G itself. In this graph $M = \Omega(2^n)$ hence our spanner construction from [10] has a size of $\Omega(n^2)$ and is essentially optimal.

Given a set of points S , we present a distance function δ such that for a given assignment of radii for the points of S any spanner of the resulting disk graph must have $\Omega(n^2)$ edges. We then prove that the metric space (S, δ) has a constant doubling dimension.

The set S is composed of two subsets X and Y . Let $X = \{x_1, \dots, x_n\}$ and let $Y = \{y_1, \dots, y_n\}$. The main idea is to create a bipartite graph $G(X, Y, E)$ in which every point of Y is connected by a directed edge to all the points of X .

For $X = \{x_1, \dots, x_n\}$, let $\delta(x_i, x_{i+1}) = 1 + \epsilon$, for some small $\epsilon > 0$. For $j > i + 1$, let $\delta(x_i, x_j) = \sum_{k=i}^{j-1} \delta(x_k, x_{k+1})$. For every $x \in X$, let $r(x) = 1$. As the distance between any two points of X is at least $1 + \epsilon$ and the radius assignment of every point is exactly 1, there are no edges between the points of X .

For $Y = \{y_1, \dots, y_n\}$, let $\delta(y_i, y_j) = 2^{i-1}n + \epsilon'$, for every $j \in [0, i - 1]$, where $\epsilon' > 0$. For every $y_i \in Y$, let $r(y_i) = 2^{i-1}n$. We now define the distances between the points of Y and the points of X . For every

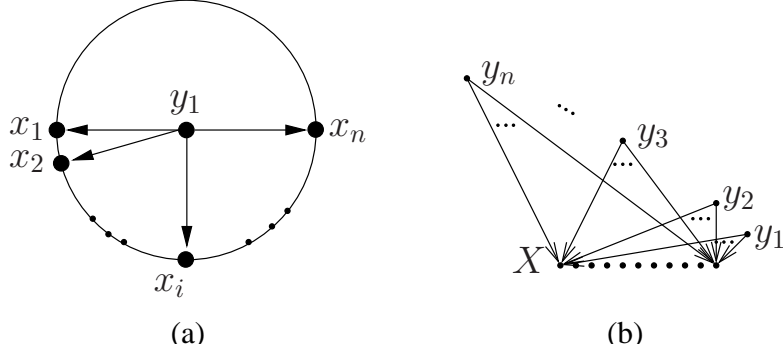


Figure 1: (a) First step in constructing the non-sparsifiable disk graph G . (b) The non-sparsifiable disk graph G .

$y_i \in Y$ and every $x \in X$, let $\delta(y_i, x) = 2^{i-1}n$. The situation for y_1 is illustrated in Figure 1(a). The points of X are placed on the boundary of a ball of radius n centered at y_1 such that the distance between any two consecutive points x_i and x_{i+1} is exactly $1 + \epsilon$. It is easy to verify that y_i has outgoing edges to the points of X (and to them only) and it does not have any incoming edges. See Figure 1(b).

The resulting disk graph G has $2n$ vertices and $\Omega(n^2)$ edges. Clearly, removing any edge from G will increase the distance between the head and tail of that edge to infinity, and thus the only spanner of G is G itself.

It is left to show that the metric space $(X \cup Y, \delta)$ has a constant doubling dimension. Given a metric space, its *doubling dimension* is defined to be the minimal value d such that every ball B of radius r in the metric space can be covered by 2^d balls of radius $r/2$. In the next Theorem we prove that for the metric space described above, d is constant.

Theorem 2.1 *The metric space $(X \cup Y, \delta)$ has a constant doubling dimension.*

Proof: Let B be a ball with an arbitrary radius r . We show that it is possible to cover all the points of $X \cup Y$ within B using a constant number of balls whose radius is $r/2$. Denote by $B(x, r)$ a ball of radius r centered at point x . The proof is divided into two cases.

Case 1: $X \cap B \neq \emptyset$ and $Y \cap B = \emptyset$. Let $x_j \in X \cap B$ be an arbitrary point. Consider the ball $B' = B(x_j, 2r)$. Clearly $B \subseteq B'$. Let x_i be the point with the largest index inside B' . The distance between x_j and x_i is $(1 + \epsilon)(i - j)$ which is at most $4r$. If $(1 + \epsilon)(i - j) \leq r/2$ then all points with index $i' > j$ are covered by $B(x_j, r/2)$. If not let $x_{j^1} \in B'$ be the point with the smallest index that is larger than j such that $x_{j^1} \notin B(x_j, r/2)$. We take now the ball $B(x_{j^1}, r/2)$ and add it to our ball cover. We continue in the same way until we cover all points with index $i' > j$. As each ball is of radius $r/2$, it follows that each time that we add a ball, the center of the next ball is closer to x_i by at least $r/2$. The distance between x_j and x_i is at most $4r$, hence a constant number of $r/2$ -balls is needed. The same argument holds for covering all points whose index is smaller than j .

Case 2: $Y \cap B \neq \emptyset$. Consider the ball $B' = B(y_j, 2r)$, where $y_j \in Y$ is the point with the highest index of $Y \cap B$. Clearly $B \subseteq B'$. In what follows we show that all the points of $(X \cup Y) \cap B'$ are covered by a constant number of balls of radius $r/2$. Let y_i be the point within $Y \cap B'$ whose index is maximal. We have to consider two possible subcases.

Subcase 2(a): $y_j = y_i$. This implies that $y_{j+1} \notin B'$, hence $2r < \delta(y_{j+1}, y_j) = 2^j n + \epsilon$. We now show that it is possible to cover B' by a constant number of balls of radius $r/2$. If $2r < 2^{j-1}n$, then only y_j is within B' and it is covered by a ball of radius $r/2$ centered at itself.

If $2^{j-1}n \leq 2r < 2^{j-1}n + \epsilon'$, then B' contains all the points of X and y_j . An $r/2$ -ball centered at y_j is used to cover y_j and the points of X can be covered with a constant number of $r/2$ -balls using the same arguments as in the first case.

Finally, if $2^{j-1}n + \epsilon' \leq 2r$, then $r/2$ is at least $2^{j-3}n + \epsilon'/4$. The ball $B(y_{j-3}, r/2)$ covers every y_k within B' , where $1 \leq k \leq j-3$, as $\delta(y_{j-3}, y_k) = 2^{j-4}n + \epsilon'$.

Hence, we cover $Y \cap B'$ by the balls $B(y_j, r/2)$, $B(y_{j-1}, r/2)$, $B(y_{j-2}, r/2)$ and $B(y_{j-3}, r/2)$. We cover $X \cap B'$ as before.

Subcase 2(b): $y_j \neq y_i$. This implies that $2^{i-1}n + \epsilon' = \delta(y_i, y_j) \leq 2r$ and $r/2 \geq 2^{i-3}n + \epsilon'/4$. The ball $B(y_{i-3}, r/2)$ covers every y_k within B' , where $k \leq i-3$, as $\delta(y_{i-3}, y_k) = 2^{i-4}n + \epsilon'$.

Hence, we cover $Y \cap B'$ by the balls $B(y_i, r/2)$, $B(y_{i-1}, r/2)$, $B(y_{i-2}, r/2)$ and $B(y_{i-3}, r/2)$. We cover $X \cap B'$ as before. □

3 Improved spanner in the relaxed disk graph model

The (negative) optimality result from the previous section motivates us to look for a slightly relaxed definition of disk graphs in which it will still be possible to create a spanner of size $O(n/\epsilon^d)$.

Let (V, δ) be a metric space of constant doubling dimension d with a radius assignment $r(\cdot)$ for its points and let $I = (V, E, r)$ be its corresponding disk graph. Assume that we multiply the radius assignment of every point by a factor of $1+\epsilon$, for some $\epsilon > 0$, and let $I' = (V, E', r_{1+\epsilon})$ be the corresponding disk graph. It is easy to see that $E \subseteq E'$. In this section we show that it is possible to create a $(1+\epsilon)$ -spanner of size $O(n/\epsilon^d)$ if we are allowed to use edges of I' . As a first step we present a simple variant of our $(1+\epsilon)$ -spanner construction of size $O(n\epsilon^{-d} \log M)$ from [10]. This variation is needed in order to obtain the efficient construction in the relaxed model which is presented right afterwards.

3.1 Spanners for general disk graphs

Let (V, δ) be a metric space of constant doubling dimension and assume that any point $p \in V$ is the center of a ball of radius $r(p)$, where $r(p)$ is taken from the range $[1, M]$. In this section we describe a simple variant of our construction from [10], which computes a $(1+\epsilon)$ -spanner with $O(n\epsilon^{-d} \log M)$ edges for a given disk graph. We then use this variant, together with new ideas, in order to obtain (in the next section) our main result, namely, a spanner with only $O(n/\epsilon^d)$ edges.

The spanner construction algorithm receives as input a directed graph $I(V, E, r)$ and an arbitrarily small constant $\epsilon > 0$, and constructs a set of spanner edges $E_{\text{SP}}^{\text{DIR}}$, returning the spanner subgraph $H^{\text{DIR}}(V, E_{\text{SP}}^{\text{DIR}})$. The construction of the spanner is based on a hierarchical partition of the points of V that takes into account the different radius of each point. The construction operates as follows. Let α and β be two small positive constants depending on ϵ , to be fixed later on. Assume that the ball radii are scaled so that the smallest edge in the disk graph is of weight 1. Fix $\rho = \lfloor \log_{1+\alpha} M \rfloor$. Let i be an integer from the range $[0, \rho]$ and let $M_i = M/(1+\alpha)^i$. The edges of $I(V, E, r)$ are partitioned into classes by length, letting $E(M_{i+1}, M_i) = \{(x, y) \mid M_{i+1} < \delta(x, y) \leq M_i\}$. Notice that $E(M_1, M_0)$ contains the longest edges.

Let $\ell(x, y)$ be the level of the edge (x, y) , that is, $\ell(x, y) = i$ such that $(x, y) \in E(M_{i+1}, M_i)$. Let p be a point whose ball is of radius $r(p) \in (M_{i+1}, M_i]$. Thus, p may have outgoing edges in $E(M_{i+1}, M_i)$ but p does not have outgoing edges in $E(M_{i'+1}, M_{i'})$ for any $i' \in [i+1, \rho]$. Denote this level by $\ell(p)$.

For every $i \in [0, \rho]$, starting from $i = 0$, the edges of the class $E(M_{i+1}, M_i)$ are considered by the algorithm in a non-decreasing order (assume that in each class the edges are sorted by their weight), that

```

Algorithm disk-spanner ( $I(V, E, R), \epsilon$ )
 $E_{\text{SP}}^{\text{DIR}} \leftarrow \phi$ 
 $P_0 \leftarrow \phi$ 
for  $i \leftarrow 0$  to  $\rho = \lfloor \log_{1+\alpha} M \rfloor$ 
  for each  $(x, y) \in E(M_{i+1}, M_i)$  do
    if  $\delta(\text{NN}(x, P_i), x) > \beta M_{i+1}$  then  $P_i \leftarrow P_i \cup \{x\}$ 
    if  $\delta(\text{NN}(y, P_i), y) > \beta M_{i+1} \wedge r(y) \geq M_{i+1}$  then  $P_i \leftarrow P_i \cup \{y\}$ 
    if  $r(y) \geq M_{i+1}$ 
      if  $\nexists (x', y') \in E_{\text{SP}}^{\text{DIR}}$  s.t.  $x' \in \Gamma_i(\text{NN}(x, P_i)) \wedge y' \in \Gamma_i(\text{NN}(y, P_i))$ 
        then  $E_{\text{SP}}^{\text{DIR}} \leftarrow E_{\text{SP}}^{\text{DIR}} \cup \{(x, y)\}$ 
    if  $r(y) < M_{i+1}$ 
      if  $\nexists (x', y) \in E_{\text{SP}}^{\text{DIR}}$  s.t.  $x' \in \Gamma_i(\text{NN}(x, P_i))$ 
        then  $E_{\text{SP}}^{\text{DIR}} \leftarrow E_{\text{SP}}^{\text{DIR}} \cup \{(x, y)\}$ 
   $P_{i+1} \leftarrow P_i$ 
return  $H^{\text{DIR}}(V, E_{\text{SP}}^{\text{DIR}})$ 

```

Figure 2: A high level implementation of the spanner construction algorithm for *general* disk graphs. The value of α and β is smaller than $\epsilon/6$.

is, we start from the class of longest edges but within each class of edges we consider the shortest edges first.

In each stage of the construction we maintain a set of pivots $P_i \subseteq V$. Let $x \in V$ and let $\text{NN}(x, P_i)$ be the closest pivot from P_i to x . Notice that closeness is measured using the metric space and hence is bidirectional.

For a pivot $p \in P_i$, define the *close neighborhood* of p as $\Gamma_i(p) = \{x \mid x \in V, \text{NN}(x, P_i) = p, r(x) \geq \delta(x, p)\}$, namely, all the points that have a directed edge to p and p is their closest pivot.

The algorithm is given formally in Figure 2. Let $(x, y) \in E(M_{i+1}, M_i)$ be an edge considered by the algorithm in the i th iteration. The algorithm has two types of selections to make, namely, *pivot* selection and *edge* selection.

Pivot selection. The algorithm checks whether x or y or both should be added to the pivots set P_i according to the following rule: If the distance from x to its closest pivot in P_i is greater than βM_{i+1} then x is added to P_i . If the distance from y to its closest pivot in P_i is greater than βM_{i+1} and the radius of y is at least M_{i+1} then y is added to P_i .

Edge selection. To decide whether the edge (x, y) is added to the spanner, the following two cases are considered. The first case is when $r(y) \geq M_{i+1}$. In this case, if there is no edge from the close neighborhood of x to the close neighborhood of y then (x, y) is added to the spanner. The second case is when $r(y) < M_{i+1}$. In this case, if there is no edge from the close neighborhood of x to y then (x, y) is added to the spanner.

When i reaches ρ , the algorithm handles all the edges that belong to $E(M_{\rho+1}, M_\rho)$. This includes also edges whose weight is 1, the minimal possible weight. The algorithm returns the directed graph $H^{\text{DIR}}(V, E_{\text{SP}}^{\text{DIR}})$.

In what follows we prove that for suitably chosen α and β , $H^{\text{DIR}}(V, E_{\text{SP}}^{\text{DIR}})$ is a $(1 + \epsilon)$ -spanner with $O(n/\epsilon^d \log M)$ edges of the directed graph $I(V, E, r)$.

Lemma 3.1 (Stretch) *Let $\epsilon > 0$, set $\alpha = \beta < \epsilon/6$ and let $H = H^{\text{DIR}}(V, E_{\text{SP}}^{\text{DIR}})$ be the graph returned by*

Algorithm disk-spanner($I(V, E, r), \epsilon$). If $(x, y) \in E$, then $\delta_H(x, y) \leq (1 + \epsilon)\delta(x, y)$.

Proof: Recall that the radii are scaled so that the shortest edge is of weight 1. We prove by induction on i that $\delta_H(x, y) \leq (1 + \epsilon)\delta(x, y)$ for every edge $(x, y) \in E(M_{i+1}, M_i)$.

For the induction base, $i = \rho$, we prove that the edges of $E(M_{\rho+1}, M_\rho)$ have a $1 + \epsilon$ approximation. For every such an edge (x, y) it holds that x is a pivot in P_i , as any other point is at distance at least 1. Thus every edge (x, y) is added to $E_{\text{SP}}^{\text{DIR}}$.

We now assume that every edge $(x, y) \in E(M_{i'+1}, M_i')$ has a $1 + \epsilon$ approximation for every $i' \in [i+1, \rho]$, and prove that every edge $(x, y) \in E(M_{i+1}, M_i)$ has a $1 + \epsilon$ approximation. Let $p = \text{NN}(x, P_i)$. Assume that $r(y) \geq M_{i+1}$ and let $q = \text{NN}(y, P_i)$. It follows from the definition that $\delta(x, p) \leq \beta M_{i+1}$ and $\delta(y, q) \leq \beta M_{i+1}$. If the edge (x, y) is not in the spanner, then there must be an edge $(\hat{x}, \hat{y}) \in E_{\text{SP}}^{\text{DIR}}$, where $\hat{x} \in \Gamma_i(p)$ and $\hat{y} \in \Gamma_i(q)$. The crucial observation is that $r(x)$ and $r(\hat{y})$ are at least M_{i+1} . By the choice of β , it follows that $2\beta M_{i+1} < M_{i+1}$ and $(x, \hat{x}), (\hat{y}, y) \in E$. Thus, there is a (directed) path from x to y of the form $\langle x, \hat{x}, \hat{y}, y \rangle$ whose length is $4\beta M_{i+1} + M_i$. However, only its middle edge, (\hat{x}, \hat{y}) , is in $E_{\text{SP}}^{\text{DIR}}$. The length of this edge is bounded by the length of the edge (x, y) since the algorithm picked the minimal edge that connects between $\Gamma_i(p)$ and $\Gamma_i(q)$. This implies that the length of (\hat{x}, \hat{y}) is at most M_i .

From the inductive hypothesis it follows that the edges (x, \hat{x}) and (\hat{y}, y) whose weight is at most $2\beta M_{i+1}$ are approximated with $1 + \epsilon$ stretch. Thus, there is a path in the spanner from x to y whose length is at most $(1 + \epsilon)\delta(x, \hat{x}) + M_i + (1 + \epsilon)\delta(\hat{y}, y)$, and this can be bounded by

$$(1 + \epsilon)4\beta M_{i+1} + M_i = ((1 + \epsilon)4\beta + (1 + \alpha))M_{i+1}.$$

As the edge $(x, y) \in E(M_{i+1}, M_i)$ it follows that $\delta(x, y) \geq M_{i+1}$. It remains to prove that $1 + 4\epsilon\beta + 4\beta + \alpha \leq 1 + \epsilon$, which follows directly from the choice of α and β .

If $r(y) < M_{i+1}$ then there must be an edge $(\hat{x}, y) \in E_{\text{SP}}^{\text{DIR}}$, where $\hat{x} \in \Gamma_i(p)$. Following similar arguments to those used above it can be shown that there is a path in the spanner from x to y of length at most $(1 + \epsilon)2\beta M_{i+1} + M_i$ and hence bounded by $(1 + \epsilon)M_{i+1}$. \square

The size of the spanner. We now prove that the size of the spanner $H^{\text{DIR}}(V, E_{\text{SP}}^{\text{DIR}})$ is $O(n\epsilon^{-d} \log M)$.

The next lemma establishes a bound on the number of incoming spanner edges that a point may be assigned on stage $i \in [0, \rho]$ of the algorithm.

Lemma 3.2 *Let $i \in [0, \rho]$ and let $y \in V$. The total number of incoming edges of y that were added to the spanner on stage i is $O(\epsilon^{-d})$ for $\epsilon \in (0, 1)$.*

Proof: Let (x, y) be a spanner edge and let $\text{NN}(x, P_i) = p$. We associate (x, y) with p . From the spanner construction algorithm it follows that this is the only incoming edge of y whose source is in $\Gamma_i(p)$. Thus, this is the only incoming edge of y which is associated with p . Now consider all the incoming edges of y on stage i . The source of each of these edges is associated with a unique pivot within distance of at most $M_i + 2\beta M_{i+1}$ away from y and any two pivots are βM_{i+1} apart from each other. As the metric space is of doubling dimension d , we get that the number of edges entering y is $(\frac{M_i + 2\beta M_{i+1}}{\beta M_{i+1}} + 1)^d = ((1 + \alpha)/\beta + 3)^d = O(\epsilon^{-d})$. \square

It follows from the above lemma that the total number of edges that were added to $E_{\text{SP}}^{\text{DIR}}$ in the main loop is $O(n\epsilon^{-d} \log M)$. The total cost of the construction algorithm is $O(m \log n)$. We scan each edge once. For each edge we need to perform nearest neighbor queries. This can be done in $O(\log n)$ time using nearest neighbor data structures. For more details on the construction time see [10].

3.2 Spanner for relaxed disk graphs

Let (V, δ) be a metric space of doubling dimension d with a radius assignment $r(\cdot)$ for its points and let $I = (V, E, r)$ be its corresponding disk graph. Assume that we multiply the radius assignment of every point by a factor of $1 + \epsilon$, for some $\epsilon > 0$, and let $I' = (V, E', r_{1+\epsilon})$ be the corresponding disk graph. In this section we show that it is possible to create a $(1 + \epsilon)$ -spanner of I of size $O(n/\epsilon^d)$ if we are allowed to use edges of I' .

Our algorithm is composed of two stages: a *building* stage and a *pruning* stage.

Building stage. In this stage that algorithm computes two spanners, H and H' , using the algorithm of Section 3.1, where H is the spanner of I and H' is the spanner of I' .

Pruning stage. In this stage the algorithm prunes the union of these two spanners. Throughout this stage we use the radius assignment of each point before the increase. Let $q \in V$ and let $\ell(q)$ be the smallest level in which q can have outgoing edges, that is, $r(q) \in (M_{\ell(q)+1}, M_{\ell(q)}]$ (recall that as the levels get larger the edges get shorter). In the pruning stage we only prune incoming edges of q whose level is below $\ell(q)$. In other words, we do not touch the incoming edges of q that are shorter than the radius of q . The pruning is done as follow. Let $\gamma = \log_{1+\alpha} 1/\beta + 1$. We keep in the spanner the incoming edges of q that come from levels $\ell(q), \ell(q) - 1, \dots, \ell(q) - 4\gamma$. We remove any incoming edges from levels $\ell(q) - 4\gamma - 1, \dots, 0$.

The algorithm runs in $O(m \log n)$ time. In the first stage we build spanners in $O(m \log n)$ time. In the second stage we scan for every point all its edges in the spanner, thus the cost of this stage is bounded by $O(m \log n)$.

Let \hat{H} be the resulting spanner and let \hat{E} be the remaining set of edges after the pruning step. In the remainder of this section we show that the size of \hat{H} is $O(n/\epsilon^d)$ and its stretch with respect to the distances in $I(V, E, r)$ is $1 + \epsilon$. We start by showing that the size of \hat{H} is $O(n/\epsilon^d)$. Notice that the first part of the proof below is possible only due to the change we have done in the previous section to our spanner construction from [10]. Roughly speaking, given an edge $(p, q) \in E$ that is shorter than $r(q)$ we use pivot selection also on q 's side (and not only on p 's) to sparsify the graph. This allows us to deal separately with edges of q of length larger than $r(q)$ and those of length smaller than $r(q)$.

Lemma 3.3 $|\hat{E}| = O(n/\epsilon^d)$.

Proof: Let (p, q) be a spanner edge that survived the pruning step. There are two possible cases to consider.

The first case is that $\ell(p, q) > \ell(q)$. Let $i = \ell(p, q)$ and let $x = \text{NN}(p, P_i)$ and $y = \text{NN}(q, P_i)$. As (V, δ) has a doubling dimension d it follows that the total number of edges at level i that connects between two pivots as the edge (p, q) that are associated with x (and with y) is $O(1/\epsilon^d)$. The distance between x and y is at most $2\beta M_{i+1} + M_i$, therefore at level $i - 2\gamma$ either x or y are no longer pivots.

Let j be such that $x \in P_j \setminus P_{j-1}$, that is, P_j is the first pivot set that contains x . Then we charge x with every (incoming and outgoing) edge such as (p, q) from levels $[j, j + 2\gamma]$ that is incident to x . Now given such an edge (p, q) whose level is i , either x or y are not pivots in level $i - 2\gamma$, which means that either x or y has been charged for this edge, since one of them first becomes a pivot between levels $i - 2\gamma$ and i .

The second case is that $\ell(p, q) \leq \ell(q)$. In this case, it must be that level $\ell(p, q)$ is among the 4γ first different levels below $\ell(q)$ from which an incoming edge is allowed to enter q . Subsequently, we associate the edge (p, q) with q , as the total number of such edges that q can have is $O(\gamma/\epsilon^d)$. \square

We now turn to prove that the stretch of the spanner \hat{H} with respect to the disk graph I is $1 + \epsilon$.

Lemma 3.4 *Let (p, q) be an edge of the spanner H that was pruned. There is a path in \hat{H} whose length is at most $(1 + \epsilon)\delta(p, q)$.*

Proof: The proof is by induction on the lengths of the pruned edges. For the induction base let (p, q) be the shortest edge that was pruned. For every $x \in V$, let $s(x)$ be the head of an edge whose level is the γ -th level below $\ell(x)$ from which x has an incoming edge. Let q_1, \dots, q_i, \dots be a sequence of points, where $q_1 = q$ and $q_i = s(q_{i-1})$. As $q_{i+1} = s(q_i)$, it follows that $\ell(q_{i+1}, q_i) \leq \ell(q_i) - \gamma$. Combining this with the fact that $\ell(q_i) \leq \ell(q_i, q_{i-1})$ we get that $\ell(q_{i+1}, q_i) \leq \ell(q_i, q_{i-1}) - \gamma$. Therefore, $\delta(q_i, q_{i-1}) \leq \beta\delta(q_{i+1}, q_i)$.

The analysis distinguishes between two cases.

Case a: There is a point q_t such that $\delta(q_t, q) > \beta\delta(p, q)$. This situation is depicted in Figure 3. (If there is more than one point that satisfies this requirement, take the one whose index is minimal.)

Claim: $\delta(q_t, q_{t-1}) \geq \frac{\beta}{2}\delta(p, q)$.

Proof: For the sake of contradiction, assume that $\delta(q_t, q_{t-1}) < \frac{\beta}{2}\delta(p, q)$. This implies that

$$2\delta(q_t, q_{t-1}) < \beta\delta(p, q) < \delta(q_t, q) \leq \sum_{i=2}^t \delta(q_i, q_{i-1}), \quad (1)$$

where the last inequality follows from the triangle inequality as the distance between q and q_t is at most $\sum_{i=2}^t \delta(q_{i-1}, q_i)$. For every $2 \leq i \leq t-1$ we have $\delta(q_i, q_{i-1}) \leq \beta\delta(q_{i+1}, q_i)$, which implies that $\delta(q_i, q_{i-1}) \leq \beta^{t-i}\delta(q_t, q_{t-1})$. Combined with (1), we get

$$\delta(q_t, q_{t-1}) < \sum_{i=2}^{t-1} \delta(q_i, q_{i-1}) \leq \delta(q_t, q_{t-1}) \sum_{i=2}^{t-1} \beta^{t-i}.$$

If $\beta < 1/2$ we have $\sum_{i=2}^{t-1} \beta^{t-i} < 1$ and this yields a contradiction. \square

We now focus our attention on the point q_{t-1} . The minimality of q_t implies that $\delta(q, q_{t-1}) \leq \beta\delta(p, q)$. By combining it with the triangle inequality we get that $\delta(p, q_{t-1}) \leq \delta(p, q) + \beta\delta(p, q)$. Therefore, in the graph I' there must be an edge from p to q_{t-1} .

Let $i = \ell(p, q_{t-1})$. There are two possible scenarios for the spanner H' . The first scenario is when $r'(q_{t-1}) < M_{i+1}$. In this case, there is an edge in H' from some $x \in \Gamma_i(\text{NN}(p, i))$ to q_{t-1} , whose length is at most $\delta(p, q) + \beta\delta(p, q)$.

There are 4γ different levels below $\ell(q_{t-1})$ from which edges that belong to the spanners H and H' are not being pruned and survived to the spanner \hat{H} . We know that the edge (q_t, q_{t-1}) is such an edge from the γ -th non-empty level below $\ell(q_{t-1})$. We also know that $\delta(q_t, q_{t-1}) > \frac{\beta}{2}\delta(p, q)$. Therefore, as the length of the edge (x, q_{t-1}) is at most $\delta(p, q) + \beta\delta(p, q)$ it is within the 4γ non-empty levels below $\ell(q_{t-1})$ and it is not pruned. We can now build a path from p to q by concatenating three segments as follows: A path from p to x , the edge (x, q_{t-1}) and a path from q_{t-1} to q . The point x is at most $2\beta\delta(p, q) + 2\beta^2\delta(p, q)$ away from p and for the right choice of β it is less than $\delta(p, q)/(1 + \epsilon)$, hence the weight of every edge on the path that approximates the distance between x and p in $H \cup H'$ is less than $\delta(p, q)$, the shortest pruned edge, and the entire path survived the punning stage. Similarly, the point q_{t-1} is at most $\beta\delta(p, q)$ away from q and again for the right choice of β every edge on the path that approximates the distance between q_{t-1} and q survived the punning stage. Thus, we get that there is a path whose length is at most

$$(1 + \epsilon)(3\beta\delta(p, q) + 2\beta^2\delta(p, q)) + \delta(p, q) + \beta\delta(p, q),$$

which is less than $(1 + \epsilon)\delta(p, q)$ for $\beta < \epsilon/11$.

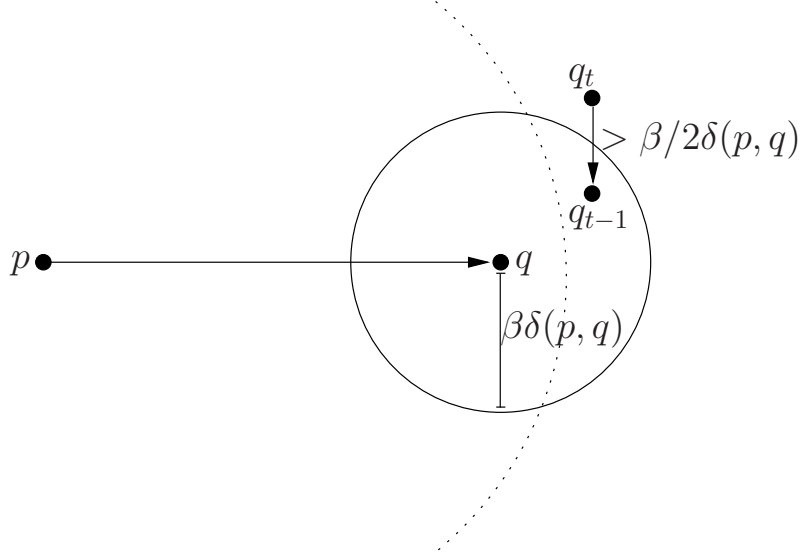


Figure 3: The case in which q_t exists

The second scenario is when $r'(q_{t-1}) \geq M_{i+1}$. In this case, there is an edge in H' from some $x \in \Gamma_i(\text{NN}(p, i))$ to some $y \in \Gamma_i(\text{NN}(q_{t-1}, i))$ whose length is at most $\delta(p, q) + \beta\delta(p, q)$, which is not being pruned. We can build a path from p to q by concatenating three segments as follows: A path from p to x , the edge (x, y) and a path from y to q . As before, for the right choice of β the paths from p to x and from y to q are composed from edges that are shorter from $\delta(p, q)$, the length of the shortest pruned edge, hence, from the minimality $\delta(p, q)$ every edge on these paths survived the punning stage. We get that there is a path whose length is at most

$$(1 + \epsilon)(4\beta\delta(p, q) + 5\beta^2\delta(p, q)) + \delta(p, q) + \beta\delta(p, q) ,$$

which is less than $(1 + \epsilon)\delta(p, q)$ for $\beta < \epsilon/19$. This completes the proof for case a.

Case b: There is no point q_t such that $\delta(q_t, q) > \beta\delta(p, q)$. In this case, let q_{t-1} be the last point in the sequence of points q_1, \dots, q_i, \dots , where $q_i = s(q_{i-1})$ and $q_1 = q$. Similarly to before, there are two possible scenarios for the spanner H' . Let $i = \ell(p, q_{t-1})$. The first scenario is when $r'(q_{t-1}) < M_{i+1}$. In this case, there is an edge in H' from some $x \in \Gamma_i(\text{NN}(p, i))$ to q_{t-1} whose length is at most $\delta(p, q) + \beta\delta(p, q)$. This edge could not be pruned, since if it was pruned then q_{t-1} could not have been the last point in the sequence. Hence we can construct a path from p to q exactly as we have done in the first scenario of case a, described above. The second scenario is when $r'(q_{t-1}) \geq M_{i+1}$. In this case, we can construct a path from p to q exactly as we have done in the second scenario of case a, described above.

This completes the proof of the induction base. The proof of the general inductive step is almost identical. The only difference is that when a path is constructed from p to q , its portions from p to x and from q_{t-1} to q in the first scenario and from p to x and from y to q in the second scenario exist in \hat{H} by the induction hypothesis and not by the minimality of $\delta(p, q)$. \square

We end this section by stating its main Theorem. The proof of this Theorem stems from Lemma 3.3 and Lemma 3.4.

Theorem 3.5 *Let (V, δ) be a metric space of constant doubling dimension with a radius assignment $r(\cdot)$ for its points and let $I = (V, E, r)$ be its corresponding disk graph. Let $I' = (V, E', r_{1+\epsilon})$ be the corresponding disk graph in the relaxed model. It is possible to create a $(1 + \epsilon)$ -spanner of size $O(n/\epsilon^d)$ for I using edges of I' .*

4 Concluding remarks and open problems

This paper presents a spanner construction for disk graphs in a slightly relaxed model that is as good as spanners for complete graphs and unit disk graphs. This result opens many other research directions for disk graphs. We list here two questions that we find particularly intriguing: Is it possible to design an efficient compact routing scheme for disk graphs? And is it possible to build an efficient distance oracle for disk graphs?

References

- [1] S. Baswana and S. Sen. A simple and linear time randomized algorithm for computing sparse spanners in weighted graphs. *Random Struct. Algorithms*, 30(4):532–563, 2007.
- [2] B. Bollobás, D. Coppersmith, and M. Elkin. Sparse distance preservers and additive spanners. *SIAM J. Discrete Math.*, 19(4):1029–1055, 2005.
- [3] U. Zwick D. Dor, S. Halperin. All-pairs almost shortest paths. *SIAM J. Comput.*, 29(5):1740–1759, 2000.
- [4] J. Gao, L. Guibas, and A. Nguyen. Deformable spanners and applications. In *Proc. 20th ACM Symp. on Computational Geometry*, pages 179–199, 2004.
- [5] Jie Gao, Leonidas J. Guibas, John Hershberger, Li Zhang, and An Zhu. Geometric spanners for routing in mobile networks. *IEEE J. on Selected Areas in Communications*, 23(1):174–185, 2005.
- [6] Sarel Har-Peled and Manor Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Computing*, 35:1148–1184, 2006.
- [7] C. Wagner L. Cowen. Compact roundtrip routing for digraphs. In *SODA '99: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 885–886, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics.
- [8] U. Zwick L. Roditty, M. Thorup. Roundtrip spanners and roundtrip routing in directed graphs. *ACM Trans. Algorithms*, 4(3):1–17, 2008.
- [9] Xiang-Yang Li, Gruia Calinescu, Peng-Jun Wan, and Yu Wang. Localized delaunay triangulation with application in ad hoc wireless networks. *IEEE Trans. on Parallel and Distributed Systems*, 14(10):1035–1047, 2003.
- [10] D. Peleg and L. Roditty. Localized spanner construction for ad hoc networks with variable transmission range. In *Proc. 7th Int. Conf. on Ad-Hoc Networks and Wireless (AdHoc-NOW)*, pages 622–633, 2008.
- [11] L. Roditty. Fully dynamic geometric spanners. In *23rd Annual Symposium on Computational Geometry (SoCG)*, pages 373–380, 2007.
- [12] Yu Wang and Xiang-Yang Li. Efficient delaunay-based localized routing for wireless sensor networks. *Int. J. of Communication Systems*, 20(7):767–789, 2006.
- [13] Andrew Chi-Chih Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM J. Comput.*, 11(4):721–736, 1982.