Efficient Spanner Construction for Directed Transmission Graphs*

Haim Kaplan†️ Wolfgang Mulzer‡ Liam Roditty§ Paul Seiferth¶

Abstract

Let $P \subset \mathbb{R}^2$ be a set of $n$ points, each with an associated radius $r_p > 0$. The transmission graph $G$ for $P$ has vertex set $P$ and a directed edge from $p$ to $q$ if and only if $q$ lies in the ball with radius $r_p$ around $p$. Let $t > 1$. A $t$-spanner $H$ for $G$ is a sparse subgraph such that for any two vertices $p$ and $q$ connected by a path of length $\ell$ in $G$, there is a path of length at most $t \cdot \ell$ from $p$ to $q$ in $H$. Given $G$ implicitly as points with radii, we show how to compute a $t$-spanner for $G$ in time $O(n \log n + \log \Psi)$, where $\Psi$ is the ratio of the largest and smallest radius in $P$.

1 Introduction

A common model for wireless sensor networks is the unit-disk graph: each sensor is modeled by a unit disk, and there is an edge between two sensors iff their disks intersect. Intersection graphs of disks with arbitrary radii have been used to model different transmission radii. These graphs are undirected, while for some networks a directed model would be more appropriate. This motivated Peleg and Roditty [5] to define transmission graphs. The vertex set of a transmission graph $G$ is a point set $P \subset \mathbb{R}^2$, where each $p \in P$ has a radius $r_p > 0$. There is a directed edge $pq$ from $p$ to $q$ iff $q$ lies in the disk $D(p)$ of radius $r_p$ around $p$.

Although transmission graphs are represented succinctly, they may have $\Theta(n^2)$ edges. Thus we would like to approximate them by sparse spanners. For $t > 1$, a subgraph $H \subseteq G$ is a $t$-spanner for $G$ if the distance between any two vertices $p$ and $q$ in $H$ is at most $t$ times the distance between $p$ and $q$ in $G$ (cf., e.g., [4]). Fürer and Kasiviswathan showed how to compute spanners for unit and general disk graphs by adapting the Yao graph [3, 6]. Peleg and Roditty [5] gave a spanner-construction for transmission graphs in any metric of bounded doubling dimension. Except for the unit-disk case, the running times depend on the number of edges. We avoid this dependency and give an efficient algorithm to construct $t$-spanners for transmission graphs for the planar Euclidean case.

Preliminaries and Results. Let $P \subset \mathbb{R}^2$ be a point set with radii, and let $G$ be its transmission graph. Let $\Phi = \max_{p,q \in P} |pq|/\min_{p,q \in P} |pq|$ be the spread of $P$. In §2, we give a construction depending on $\Phi$:

**Theorem 1** Let $G$ be the transmission graph for an $n$-point set $P \subset \mathbb{R}^2$ with spread $\Phi$. For any $t > 1$, we can find a $t$-spanner for $G$ in time $O(n (\log n + \log \Phi))$.

The radius ratio $\Psi = \max_{p,q \in P} r_p/r_q$ of $P$ is the ratio of the largest and the smallest radius in $P$. In §3 we extend our construction to depend on $\Psi$ instead of $\Phi$.

**Theorem 2** Let $G$ be the transmission graph for an $n$-point set $P \subset \mathbb{R}^2$ with radius ratio $\Psi$. For $t > 1$, we can find a $t$-spanner for $G$ in time $O(n (\log n + \log \Psi))$.

We may assume that $\Psi \leq \Phi$: a radius less than the smallest distance $c$ in $P$ can be set to $c/2$, and a radius larger than the diameter $d$ of $P$ can be set to $d$.

Our construction uses planar grids. For $i = 0, 1, \ldots$, the grid at level $i$, $Q_i$, consists of axis-parallel squares of diameter $2^i$ that partition the plane in grid-like fashion (the cells). $Q_i$ is aligned so that the origin is a grid vertex. The distance between two cells is the smallest distance of any two points contained in them. We assume that our computational model can find the grid cell containing a given point in $O(1)$ time.

2 Efficient Spanner Construction

Let $P \subset \mathbb{R}^2$ be a point set with radii, and let $\Phi$ be the spread of $P$. Let $G$ be the transmission graph of $P$. Our spanner construction is a modification of the Yao graph [6] that takes the disks into account. Ideally, our spanner $H$ should look as follows: we pick a suitable $k \in \mathbb{N}$, and we let $C$ be a set of $k$ cones with opening angle $2\pi/k$ and the origin as apex that partition the plane. For $q \in P$ and $C \in C$, let $C_q$ be the translated copy of $C$ with apex $q$. We pick the closest vertex $p \in P$ in $C_q$ with $q \in D(p)$, and we add the edge $pq$ to $H$. This gives $O(kn)$ edges, and one can show that $H$ is a $t$-spanner for $t = 1 + \Theta(1/k)$.

This is folklore in the spanners community [2, 5]. Since we do not know how to find these edges quickly, we present an approximate construction with similar properties. We partition each cone $C_q$ into “intervals” obtained by intersecting $C_q$ with annuli centered at $q$ whose inner and outer radii grow exponentially; see Fig. 1. Then we cover each interval with $O(1)$ grid cells whose diameter is “small” compared to the distance between the interval and $q$. 

---

*Supported by GIF project 11614/DFG project Mu/3501-1.
†Tel Aviv University, Israel. haimk@post.tau.ac.il
‡Institut für Informatik, Freie Universität Berlin, Germany {mulzer,seiferth}@inf.fu-berlin.de
§Bar Ilan University, Israel. lianr@macs.biu.ac.il

This is an extended abstract of a presentation given at EuroCG 2015. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.
This gives two properties that help us find an approximately shortest incoming edge for \( q \) in \( C_q \): once we have an incoming edge, we need not consider larger intervals, and if there are multiple edges from the same cell, it suffices to pick one of them. Below, we define a decomposition of \( P \) that represents the discretized intervals by a neighborhood relation between grid cells.

We first give the properties of this decomposition and use it to find the edges for \( H \). Then we prove that \( H \) is a \( t \)-spanner for an appropriate choice of parameters. Finally, we show how to use a quadtree to find this decomposition and how to implement the main steps in the desired running time.

Let \( c > 2 \) be a large constant. For a grid cell \( \sigma \), let \( m_\sigma \) be the point in \( P \cap \sigma \) with the largest radius.

**Definition 1** Let \( G \) be a transmission graph with vertex set \( P \subset \mathbb{R}^2 \). A \( c \)-separated annulus decomposition for \( G \) consists of a finite set \( \mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i \) of grid cells, a symmetric neighborhood relation \( \mathcal{N} \), and assigned sets \( R_\sigma \) for each \( \sigma \in \mathcal{Q} \) so that (i) for all \( \sigma, \sigma' \in \mathcal{N} \), \( \text{diam}(\sigma) = \text{diam}(\sigma') \) and \( d(\sigma, \sigma') \leq |\sigma - \sigma'| \); and (ii) for every edge \( \overline{pq} \) of \( G \), there is \( (\sigma, \sigma') \in \mathcal{N} \) with \( \sigma \subseteq \sigma' \), \( \text{diam}(\sigma') \geq c \text{diam}(\sigma) \), and either \( p \in R_\sigma \) or \( q \notin D(m_\sigma) \).

For \( \sigma \in \mathcal{Q} \), we define \( N(\sigma) = \{ \sigma' \mid (\sigma, \sigma') \in \mathcal{N} \} \). Property (i) in Def. 1 implies \( |N(\sigma)| = O(1) \).

**Getting a Spanner.** Let \( t > 1 \) be the desired stretch factor. Depending on \( t \), we choose constants \( c \) (separation) and \( k \) (number of cones) in a way to be described later. Let \( \mathcal{Q} \) be a \( c \)-separated annulus decomposition for \( G \). To get a \( t \)-spanner \( H \subseteq G \), we go through all cones \( C \in \mathcal{C} \) and pick all incoming edges for \( C \) as in Alg. 1. Instead of searching incoming edges for each point \( q \in P \) separately, we group points using the cells of \( \mathcal{Q} \). This gives the speed-up required for the desired running time, as shown later. We consider the cells \( \sigma \in \mathcal{Q} \) by increasing diameter, and we search incoming edges for points in \( \sigma \) without incoming edges so far. These are the active points. Initially all points are active. Fix a pair \( (\sigma, \sigma') \in N \) and let \( Q \) and \( R \) as in Alg. 1. We find for each point \( q \in Q \) one incoming edge whose other endpoint is in \( R \), if such an edge exists (edge selection). Having \( Q \) sorted (line 5) allows us to find these edges efficiently (see Algorithm 8).

**Algorithm 1:** Finding edges for \( P \) in a cone \( C \in \mathcal{C} \).

1. Set all points in \( P \) to active.
2. For each \( \sigma \in \mathcal{Q} \) by increasing diameter do
3. For each \( \sigma' \in N(\sigma) \) do
4. \( Q \leftarrow \) all active \( q \in \sigma \cap P \) with \( \sigma \cap \sigma' \neq \emptyset \)
5. Sort \( Q \) in \( x/y \)-direction // preprocess
6. \( R \leftarrow R_{\sigma'} \cup \{ m_{\sigma'} \} \)
7. // edge selection
8. For all \( q \in Q \) find \( r \in R \) with \( q \in D(r) \), if it exists, and add \( pq \) to \( H \)
9. Make all \( q \in Q \) with incoming edges inactive.

For each \( C \in \mathcal{C} \) and \( q \in P \), at most one cell \( \sigma \in \mathcal{Q} \) with \( q \notin \sigma \) gives incoming edges for \( q \). Thus, \( q \) becomes inactive after processing \( \sigma \). Since \( |\mathcal{Q}| = k \), we have \( O(k) \) incoming edges, and \( H \) has \( O(n) \) edges. Next, we show that \( H \) is a \( t \)-spanner. For this, we need three technical lemmas: Lemma 3 deals with the imprecision due to the grid. Let \( \overline{pq} \) be an edge of \( G \) contained in the cone \( C_q \). We prove that if we slightly increase the opening angle of \( C_q \), Alg. 1 picks at least one edge \( \overline{pq} \) contained in the larger cone.

**Lemma 3** Let \( k \geq 8 \) and \( c > 3 + 2/(\sin \pi/k) \). Given \( i \in \mathbb{N}_0 \) and cells \( \sigma, \sigma' \in \mathcal{Q}_i \) with \( d(\sigma, \sigma') \geq (c - 2)2^i \), let \( C_q \) be a cone with opening angle \( 2\pi/k \) and apex \( q \) in \( \sigma \) that intersects \( \sigma' \). Then the cone obtained from \( C_q \) by doubling its opening angle contains \( \sigma' \).

**Lemma 4** Let \( C_q \) be a cone with apex \( q \) and opening angle \( 4\pi/k \). Suppose there are two points \( p \) and \( r \) in \( C_q \) with \( (c + 1)2^i \leq |pr| \leq |pq| \geq (c - 2)2^i \). Then \( |pr| \leq ((4\pi/k)(c + 1) + 3)2^i \).

**Lemma 5** Let \( k \geq 14 \) and let
\[
t = (1 + \sqrt{2 - 2 \cos(4\pi/k)})/(2 \cos(4\pi/k) - 1).
\]
For any distinct points \( p, q, r \in \mathbb{R}^2 \) with \( |rq| \geq |pq| \) and \( \alpha = \angle pqr \in [0, 4\pi/k] \), we have \( |pr| \leq |pq| - |rq|/t \).

We are now ready to prove that \( H \) is a \( t \)-spanner. This is done in a similar manner as for Yao graphs.

**Lemma 6** For any \( t > 1 \), there are constants \( c \) and \( k \) such that \( H \) is a \( t \)-spanner for \( G \).

**Proof.** We show by induction on the rank of the length of the edges in \( G \) that for each edge \( \overline{pq} \) in \( G \) there is a \( p \)-\( q \)-path of length at most \( t|pq| \) in \( H \).

Consider the shortest edge \( \overline{pq} \) of \( G \). Let \( C_q \) be the cone at \( q \) that contains \( p \). There is at least one pair in \( N \) that fulfills Def. 1(ii) for \( \overline{pq} \). Among those, we pick the pair \( (\sigma, \sigma') \in N \) with minimum diameter. Suppose that \( q \in \sigma \), \( p \in \sigma' \), and \( \text{diam}(\sigma) = \text{diam}(\sigma') = 2^i \).
Since \( \overrightarrow{pq} \) is the shortest edge, \( \sigma' \) contains only \( p \) (taking \( c > 3 \)) and \( m_{\sigma'} = p \), so \( R = \{ p \} \). Furthermore, since \( p \in \sigma' \), we have \( C_q \cap \sigma' \neq \emptyset \). Thus, if \( q \) is active in \( \sigma \), then \( q \in Q \), and we pick the edge \( \overrightarrow{pq} \) for \( H \) (Alg. 1, line 8). Suppose not. Then we have picked an incoming edge \( \overrightarrow{pq} \) for a smaller pair \( (\sigma, \sigma') \in N \) with \( \text{diam}(\sigma) \leq 2^{-1} \). By Def. 1(i), \( |rq| \leq (c+1)2^i \). Also by Def. 1(ii) we have \( |pq| \geq (c-2)2^i \) and since \( |rq| \geq |pq| \), we have \( \{c+1\}2^i \geq |rq| \geq (c-2)2^i \). By Lemma 3, \( \sigma' \) (and thus \( r \)) is contained in the cone \( C_q' \) obtained from \( C_q \) by doubling its angle to \( 4\pi/k \). Using Lemma 4 with \( C_q' \), we see that \( |pr| \leq ((4\pi/k)(c+1)+3)2^i \). Since for \( c, k \geq 14 \) we have \( (4\pi/k)(c+1)+3 < c-2 \), this would mean that \( |pr| > |pq| \leq rp \). Thus, \( \overrightarrow{pq} \) is an edge of \( G \) that is strictly shorter than \( \overrightarrow{pq} \), despite our choice of \( \overrightarrow{pq} \). Hence, when processing \( \sigma \), we will discover \( \overrightarrow{pq} \).

For the inductive step, consider an edge \( \overrightarrow{pq} \) and the cone \( C_q \) containing \( p \). Again, let \( (\sigma, \sigma') \in N \) be the smallest pair of cells with \( q \in \sigma \) and \( p \in \sigma' \) that fulfill Def. 1(ii) and suppose \( \text{diam}(\sigma) = 2^i \). We have \( C_q \cap \sigma' \neq \emptyset \), and we distinguish two cases.

**Case 1:** \( q \) is active. Then \( q \in Q \) and Def. 1(i) guarantees that Alg. 1 obtains an incoming edge \( \overrightarrow{pq} \) for \( r \) with \( r \in \sigma' \). If \( r = p \), we are done, so suppose \( r \neq p \). Since \( |pr| \leq 2^i \), by induction there is a path from \( p \) to \( r \) in \( H \) of length at most \( 2^i \). Using the triangle inequality, we estimate the distance \( d(p, q) \) in \( H \) by

\[
d(p, q) \leq t2^i + |rq| \leq 2t^2 + |pq| + 2^i = |pq| + (1 + t)2^i.
\]

For \( c \) large enough the bound \( |pq| > (c-2)2^i \) gives

\[
|pq| + (1 + t)2^i \leq 1 + (1 + t)/|c-2)| |pq| \leq t|pq|.
\]

**Case 2:** \( q \) is inactive. There is an edge \( \overrightarrow{pq} \) that was selected due to a pair \( (\sigma, \sigma') \in N \) with \( q \in \sigma \), \( r \in \sigma' \) and \( \text{diam}(\sigma) \leq 2^{-1} \). By Lemma 3, \( p \) and \( r \) are contained in the cone \( C_q' \) with opening angle \( 4\pi/k \). We distinguish two subcases.

First, suppose that \( |rq| \geq |pq| \). Then, since \( (c+1)2^i \geq |rq| \geq (c-2)2^i \), Lemma 4 implies that \( r \in D(p) \), so \( \overrightarrow{pq} \) is an edge of \( G \) of length at most \( ((4\pi/k)(c+1)+3)2^i \). Thus, we can bound \( d(p, q) \) by

\[
t|pr| + |rq| \leq t(4\pi(c+1)/k+3)2^i + (c+1)2^i = (t(4\pi(c+1)/k+3)+c+1)|pq|/(c-2) \leq |pq|t,
\]
for \( c, k = \Theta(t/(t-1)) \). Here we used the fact that \( |rq| \leq (c+1)2^i \) and that \( 2^i \leq |pq| \).

Second, suppose \( |rq| < |pq| \). By Lemma 5, we get \( |pr| \leq |pq| - |rq| \). Thus, \( \overrightarrow{pq} \) is an edge of \( G \), and

\[
d(p, q) \leq t|pr| + |rq| \leq t(|pq| - |rq|)/t + |rq| = t|pq|,
\]
where the first inequality is by induction.

**Finding the Decomposition.** We show how to find the decomposition for \( G \) as in Def. 1. Let \( c > 3 \) and scale \( P \) so that the closest pair in \( P \) has distance \( c \). A **quadtree** for \( P \) is a rooted tree \( T \) where each internal node has degree four. Each node \( v \) of \( T \) has an associated cell \( \sigma_v \) from a grid \( Q_i, i \geq 0 \), and we say that \( v \) has **level** \( i \). If \( v \) is internal, the cells of its four children partition \( \sigma_v \) into four congruent squares with half the diameter of \( \sigma_v \). We compute a quadtree \( T \) for \( P \) and use it to find a c-separated annulus decomposition.

We construct \( T \) level-wise. To begin, we take the smallest integer \( L \) such that there is a cell \( \sigma \in Q_L \) that contains \( P \). Since \( c \) is constant and since \( P \) has spread \( \Phi \), the scaled point set has diameter \( \Phi \), and \( L = O(\log \Phi) \) (possibly after shifting \( P \)). We create the root \( v \) and set \( \sigma_v = \sigma \). This gives level \( L \). To construct level \( i-1 \) from level \( i \), we do the following for each level-i-node \( v \) whose cell \( \sigma_v \) is non-empty: we take the four cells of \( Q_{i-1} \) that partition \( \sigma_v \) and create four children \( v_1, \ldots, v_4 \) of \( v \). To each of \( v_1, \ldots, v_4 \) we assign one of the four cells. We stop at level 0. The scaling of \( P \) ensures that a cell of level 0 contains at most one point and has diameter 1.

We now set \( Q = \{ \sigma_v \mid v \in T \} \). We let \( (\sigma_v, \sigma_w) \in N \) if \( v \neq w \) and the same level and \( \text{diam}(\sigma_v, \sigma_w) \in [(c-2) \text{diam}(\sigma_v), 2\text{diam}(\sigma_v)) \). As \( R_{\sigma_v} \), we take all \( p \in \sigma_v \cap P \) with \( r_p \in [(c-2) \text{diam}(\sigma_v), 2(c+1) \text{diam}(\sigma_v)] \).

**Lemma 7** The set \( Q \) with \( N \) and \( R_{\sigma} \) as above is a c-separated annulus decomposition with \( |Q| = O(n) \).

**Proof.** Since \( T \) has \( O(n) \) nodes, we have \( |Q| = O(n) \). Property (i) of Def. 1 follows by construction. For Property (ii), let \( \overrightarrow{pq} \) be an edge of \( G \) and let \( i \in \mathbb{N}_0 \) such that \( |pq| \in [c^2, c^{2i+1}] \). Let \( \sigma, \sigma' \in Q_i \) with \( p \in \sigma \) and \( q \in \sigma' \). By construction, these cells are assigned to nodes of \( T \) and thus, \( \sigma, \sigma' \in Q \). Since \( \text{diam}(\sigma) = \text{diam}(\sigma') = 2^i \), we have \( (c-2)2^i \leq \text{diam}(\sigma, \sigma') \leq |pq| \leq c^{2i+1} \). Thus, \( \sigma, \sigma' \in N \). Since \( \overrightarrow{pq} \) is an edge of \( G \), we have \( r_p \geq |pq| \geq c^2 \). If \( r_p \leq (c+1)2^{i+1} \), then \( p \in R_{\sigma_v} \). Otherwise, \( r_{m_{\sigma'}}, \sigma' \geq c^{2i+1} \), and \( D(m_{\sigma'}) \) contains \( \sigma' \) and \( \sigma \).

**Running Time.** Considering the cells of \( Q \) in increasing order in Alg. 1 constitutes a level-order traversal of \( T \) starting from level 0. Fix a cell \( \sigma_v \) of a node \( v \) of \( T \). We can sort \( \sigma_v \cap P \) in the preprocess step (line 5) by merging the sorted lists of \( v \)'s children. This takes \( O(n) \) time per level and \( O(n \log \Phi) \) time in total. Now we bound the time for edge selection.

**Lemma 8** Let \( Q, R \) as in Alg. 1, line 8 with \( |Q| = n \) and \( |R| = m \). For each \( q \in Q \) we can find an \( r \in R \) with \( q \in D(r) \), if such \( r \) exists, in time \( O(m \log m + n) \).

**Proof.** Let \( Q, R \) are separated by one the supporting lines \( \ell \) of the cell \( \sigma \) that contains \( Q \). Since \( \sigma \) is axis-aligned, \( Q \) is sorted along \( \ell \) in the preprocess step. Consider a coordinate system with \( x \)-axis \( \ell \). The lower envelope \( E \) of the disks of \( R \) and \( \ell \) has \( O(m) \) arcs, can
Lemma 9 The construction of the spanner $H$ of $G$ takes $O(n(\log \Phi + \log n))$ time.

3 From Bounded Spread to Bounded Radii

To get Theorem 2, we extend Alg. 1 from §2. We show that the spread is irrelevant: points that are close together form cliques in $G$ and can be handled through classic spanners; points that are far away from each other form pairwise independent components.

Given $t$, we pick the separation parameter $c$ large enough. We scale $P$ such that the smallest radius is $c$. Let $M = O(\Psi)$ be the largest radius. We partition $P$ into independent components. For this, we put around each $p \in P$ an axis-parallel square of side length $2M$. The connected components of the intersection graph of the squares give the sets. We state this in the next lemma, whose proof we omit.

Lemma 10 In $O(n \log n)$ time, we can partition $P$ into sets $P_1, \ldots, P_t$ of diameter $O(n \Psi)$ so that for $i \neq j$, no point in $P_i$ can reach a point in $P_j$ in $G$.

By Lemma 10, we may assume that $P$ has diameter $O(n \Psi)$. As in §2, we compute a quadtree $T$ for $P$ with $L$ levels and $L = O(\log(n \Psi))$. Unlike in §2, $T$ does not directly yield a $c$-separated annulus decomposition for $G$. Def. 1(ii) does not hold, since there may be edges in $G$ that do not go between neighboring cells. These are the short edges.

First, we handle very short edges: let $v$ be a level 0 node of $T$ with associated cell $\sigma_v \in Q_0$. Let $Q \subseteq P$ be the points in cells of $Q_0$ with distance at most $c/2 - 3$ from $\sigma_v$. Since any two points in $Q$ have distance at most $c$, $Q$ is a clique in $G$. We compute a (classic) $t$-spanner for $Q$ in $O(|Q| \log |Q|)$ time [4]. Since any $p \in P$ is in $O(c^2)$ such spanners, we generate $O(n)$ edges in total and require $O(n \log n)$ running time.

Second, we handle not quite so short edges: for each $q \in P$, let $v$ be the level 0 node of $T$ whose cell $\sigma_v$ contains $q$. For any non-empty $\sigma' \in Q_0$ with $d(\sigma_v, \sigma') \in (c/2 - 3, c - 2)$, we take an arbitrary point $r \in \sigma' \cap P$ and add the edge $\overline{qr}$ to our spanner. All these edges have length at most $c$ and thus are edges in $G$. This takes $O(n)$ time and creates $O(n)$ edges.

Finally, we handle the remaining edges: we mark all points in $P$ as active, and we run Alg. 1 from §2 for the cells of $T$. Call the resulting graph $H$.

As in Lemma 6, we can show inductively that each edge of $G$ is approximated in $H$. The differences are in the base case: if the shortest edge in $G$ is very short, the classic spanner does the job. If it is not quite so short, a calculation as in Lemma 6 shows that we pick it. Otherwise, the base case is as in Lemma 6.

Lemma 11 For any $t > 1$, there are constants $c, k$ such that the graph $H$ as above is a $t$-spanner for $G$.

Thm. 2 follows as in §2. The running time analysis goes as in Lemma 9, but the quadtree has $O(\log n + \log \Psi)$ levels.

Acknowledgments. We like to thank Paz Carmi for valuable comments.

References