Planar Hop Spanners for Unit Disk Graphs*

Nicolas Catusse, Victor Chepoi, and Yann Vaxès

Laboratoire d'Informatique Fondamentale
Université d'Aix-Marseille
Faculté des Sciences de Luminy
F-13288 Marseille Cedex 9, France
{nicolas.catusse,victor.chepoi,yann.vaxes}@lif.univ-mrs.fr

Abstract. The simplest model of a wireless network graph is the Unit Disk Graph (UDG): an edge exists in UDG if the Euclidean distance between its endpoints is ≤ 1. The problem of constructing planar spanners of Unit Disk Graphs with respect to the Euclidean distance has received considerable attention from researchers in computational geometry and ad-hoc wireless networks. In this paper, we present an algorithm that, given a set X of terminals in the plane, constructs a planar hop spanner with constant stretch factor for the Unit Disk Graph defined by X. Our algorithm improves on previous constructions in the sense that (i) it ensures the planarity of the whole spanner while previous algorithms ensure only the planarity of a backbone subgraph; (ii) the hop stretch factor of our spanner is significantly smaller.

Key words: Planar spanner, Unit Disk Graph, Geometric Networks, Wireless Networks

1 Introduction

The problem of constructing sparse spanners (i.e., subgraphs approximating the distances between the vertices of the original graph up to a certain stretch factor) of geometric graphs has received considerable attention in computational geometry and ad-hoc wireless networks; we refer the reader to the book by Narasimhan and Smid [10]. The simplest model of a wireless network graph is the *Unit Disk Graph* (UDG): an edge between two terminals u, v exists in this graph if the Euclidean distance between u and v is at most one. Some routing algorithms such as *Greedy Perimeter Stateless Routing* require a *planar* subgraph to route the messages through the network. Therefore, additionally to the small stretch factor, it is plausible to require that the obtained spanner is also planar.

In this paper, we design an algorithm that, given a set X of n points on the plane, constructs a planar spanner with constant hop stretch factor for the Unit Disk Graph defined by X. Contrary to the problem of constructing planar *Euclidean length* spanners, for which several algorithms provide small stretch

* This research was partly supported by the ANR grant BLAN06-1-13889 (projet OPTICOMB).

factors (see [9], for instance), the problem of constructing planar *hop* spanners with constant stretch factor remained open. Some partial solutions ensuring the planarity of a certain backbone subgraph were proposed in [1, 8]. Our algorithm improves on the results of [1, 8] in the sense that (i) our construction ensures the planarity of the whole spanner; (ii) the hop stretch factor provided by our algorithm is significantly better. Additionally, our spanner can be constructed via a localized distributed algorithm. Planarity, low stretch factor, and localized construction constitute key ingredients to obtain efficient routing schemes for ad-hoc and wireless geometric networks.

The rest of the paper is organized as follows. In Section 2, we briefly review the literature related to geometric spanners and Unit Disk Graphs. Section 3 presents a very simple construction that provides a sparse spanner for UDG with low hop stretch factor. In general, this construction does not ensure the planarity of the spanner. Section 4 describes an algorithm that updates the spanner defined in Section 3 in order to obtain a planar spanner still preserving a small hop stretch factor. In Section 5, we prove some results necessary to prove the planarity and the hop stretch of the spanner computed by our algorithm.

2 Previous Work

We start with basic definitions. Given a connected graph $G = (V, E)$ with n vertices embedded in the Euclidean plane, the hop length of a path $\gamma(u, v)$ between two vertices u, v of G is the number of edges of $\gamma(u, v)$. The *hop distance* $d_G(u, v)$ between u and v in G is the length of a shortest path connecting u and v in G. A subgraph H of G is a *spanner* of G if there is a positive real constant t such that for any two vertices, $d_H(u, v) \leq t d_G(u, v)$. The constant t is called the *hop stretch factor* of H. If instead of number of edges in a path $\gamma(u, v)$ we consider the total Euclidean length of the edges of $\gamma(u, v)$, then we can define another distance measure on G and a corresponding notion of a spanner, which is called *Euclidean spanner* and its stretch factor is called *Euclidean stretch factor*. All results of our paper concern hop spanner of Unit Disk Graphs, therefore we often write "spanner" and "distance" instead of "hop spanner" and "hop distance".

Spanner properties of geometric graphs have been surveyed by Eppstein [5] and more recently by Bose and Smid [3]. Bose et al. [2] proved that the *Gabriel Graph* is an $\Omega(\sqrt{n})$ hop spanner and a $\Theta(\sqrt{n})$ Euclidean spanner for UDG, and that the *Relative Neighborhood Graph* is a $\Theta(n)$ and a $\Theta(n)$ Euclidean spanner for UDG. Gao et al. [8] proposed a randomized algorithm to construct an Euclidean and a hop spanner for UDG. This algorithm creates several clusters (using a method described in [7]) connected by a *Restricted Delaunay graph.* The subgraph consisting of edges between distinct clusters is planar. This construction provides a constant Euclidean stretch factor in expectation but its hop stretch factor is not given. Alzoubi et al. [1] proposed for the same problem a distributed algorithm that uses the *Local Delaunay Triangulation* defined by Li, Călinescu, and Wan in [9]. However, the hop stretch factor obtained by [1] is huge (around 15000) and the intra-cluster edges may cross the edges of the triangulation (and therefore does not provide a full planar hop spanner). Chen et al. [4] presented the construction of Euclidean spanners for *Quasi-UDG* which can be used for routing. Their construction method is similar to our approach in the sense that it also uses a regular squaregrid to partition the set of terminals into clusters (for a similar partition of the plane used for routing, see [11]). Recently Yan, Xiang, and Dragan [12] established a *balanced separator* result for Unit Disk Graph which mimics the celebrated Lipton-Tarjan planar balanced separator theorem. Based on this result, they derive a compact and low delay routing labeling scheme for UDG. Finally, for the construction of spanner of general disk graphs see also [6].

3 Sparse almost planar spanners

Let X be a set of n points (terminals) in the plane. In this section, we describe a very simple algorithm, namely Algorithm 1, that constructs a sparse 5-spanner $H' = (X, E')$ of the Unit Disk Graph $G = (X, E)$ defined by X. It uses a regular grid Γ on the plane with squares of side $\frac{\sqrt{2}}{2}$. A square of Γ is said to be *nonempty* if it contains at least one terminal from X. For any point $x \in X$, let $\pi(x)$ denote the square of Γ containing x. The graph $H' = (X, E')$ has two types of edges : a subset $E'_0 \subseteq E'$ of edges connecting terminals lying in the same square and a subset E'_1 of edges running between terminals lying in distinct squares; let $E' = E'_0 \cup E'_1$. To define E'_0, in each nonempty square π we pick a terminal (the *center* of π) and add to E'_0 an edge between this terminal and every other terminal located in π. Clearly, all of them are edges of G because the distance between two points lying in the same square of side $\frac{\sqrt{2}}{2}$ is at most 1. In E'_1 we put exactly one edge of G running between two nonempty squares if such an edge exists. In the sequel, with some abuse of notation, we will denote by $\pi\pi'$ the shortest edge of UDG running between two squares π and π'.

Algorithm 1 Construction of sparse spanner H'

1: For each square π, pick a terminal c_π (the *center* of π) and add to E'_0 an edge between c_π and every other terminal located in π.
2: For all squares π, π' connected by an edge of G, add to E'_1 the shortest edge between π and π'.

Proposition 1. *The graph $H' = (X, E')$ is a 5-hop spanner for G with at most $10n$ edges.*

Proof. For the first assertion, it suffices to prove that $d_{H'}(u, v) \leq 5d_G(u, v)$ for any two adjacent in G vertices u, v. If u and v belong to the same square π, then they are neighbors of the center of π, hence they are connected in H' by a path of length 2. Now, suppose that u and v belong to different squares. Since

uv is an edge of G, the graph H' must contain an edge $u'v'$ of G with $u' \in \pi(u)$ and $v' \in \pi(v)$. Therefore the terminals u and v are connected in H' by a path of length ≤ 5 consisting of two paths of length 2 passing via the centers of the clusters $\pi(u)$ and $\pi(v)$ and connecting u and v to u' and v', respectively, and the edge $u'v'$ joining these clusters. To prove the second assertion, let n_0 denote the number of non-empty squares. Then obviously $|E'_0| \leq n - n_0$. Since from each nonempty square π we can have edges of E'_1 to at most 20 other such squares(see Fig. 2), and since each such edge is counted twice, we conclude that $|E'_1| \leq 10n_0$. Therefore $|E'| \leq (n - n_0) + 10n_0 \leq 10n$. □

4 Planar spanners

In this section, we describe the Algorithm 2 that builds a planar hop spanner H for a UDG graph G. We first compute a planar set of *inter-cluster edges* E_3 whose end-vertices belong to distinct squares of Γ and then, a second set of *intra-cluster edges* E_0 connecting the vertices that belong to the same square.

4.1 Computing E_3

First we define the l_1*-distance* $l_1(\pi, \pi')$ between two squares π and π' in Γ as the graph distance between π and π' in the dual grid (squares become vertices and two vertices are adjacent if their squares have a common side). The l_1*-length* $l_1(ab)$ of an edge ab of G is the l_1-distance between the squares that contain their end-vertices and the *interval* $I(x, y)$ between two terminals $x, y \in X$ consists of all squares lying on a shortest l_1-path of the dual grid between $\pi(x)$ and $\pi(y)$.

Now, we give an brief and informal description of Algorithm 2 for computing E_3. The edge set E_3 is a planar subset of the edge set E'_1 defined in the previous section. We remove some edges to obtain a planar graph while preserving a bounded hop stretch factor of the resulting spanner. The principle of our algorithm is to minimize the l_1-length of preserved edges: if the end-vertices of an edge uv of E'_1 are joined by a path in E'_1 having the same total l_1-length as uv, then uv is removed. An edge with large l_1-length potentially crosses many squares and, as a consequence, many edges of UDG. Hence, taking such an edge in our planar spanner would exclude many other (potentially, good) edges from the spanner. For each removed edge, there is a path having a constant number of edges between its end-vertices. Therefore, after the removal of these edges, the stretch factor is still bounded by a constant. The minimization of the l_1-length is not sufficient to obtain a planar graph but we show that this operation considerably decreases the number of crossing configurations. The next step of the algorithm consists in repairing the remaining crossings. During this step, some edges are removed to ensure the planarity and some other edges are added back to preserve a small distance between the end-vertices of removed edges. Notice that our construction uses only local information and can be easily implemented as a localized distributed algorithm. Using the proof outlined below, we establish that the edge set obtained at the end of this process is indeed planar.

Algorithm 2 Construction of the spanner H

1: Let $H' = (X, E'_1)$ be the inter-cluster graph returned by Construction of Sparse Spanner.
2: Let $G_1 = (X, E_1)$ be the graph obtained from H' by removing every edge $ab \in E'_1$ whose end-vertices are joined by a *replacement path*, i.e., a path $P \neq ab$ between $\pi(a)$ and $\pi(b)$ such that $l_1(P) = l_1(ab)$.
3: For each pair of crossing edges $xy, x'y'$ of E_1, identify the crossing configuration (according to Fig. 4) and remove the edge $x'y'$. Let $G_2 = (X, E_2)$ be the graph obtained from G_1 by removing these edges.
4: For each edge $x'y'$ removed in Step 3, unless $xy, x'y'$ form a Configuration 0 or 4′, if there is no replacement path in G_2 for $\pi(y)\pi(y')$, then add the edge $\pi(y)\pi(y')$ (according to Fig. 4). Let $G_3 = (X, E_3)$ be the resulting graph.
5: Compute the set of intra-cluster edges E_0 as described in subsection 4.2.
6: Output the graph $H = (X, E_0 \cup E_3)$.

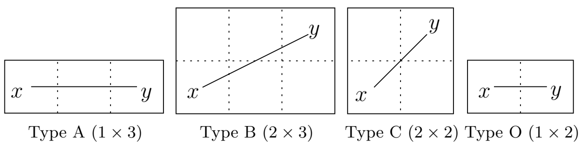

Fig. 1. Classification of edges

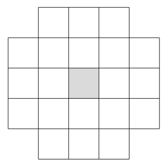

Fig. 2. Twenty neighboring squares

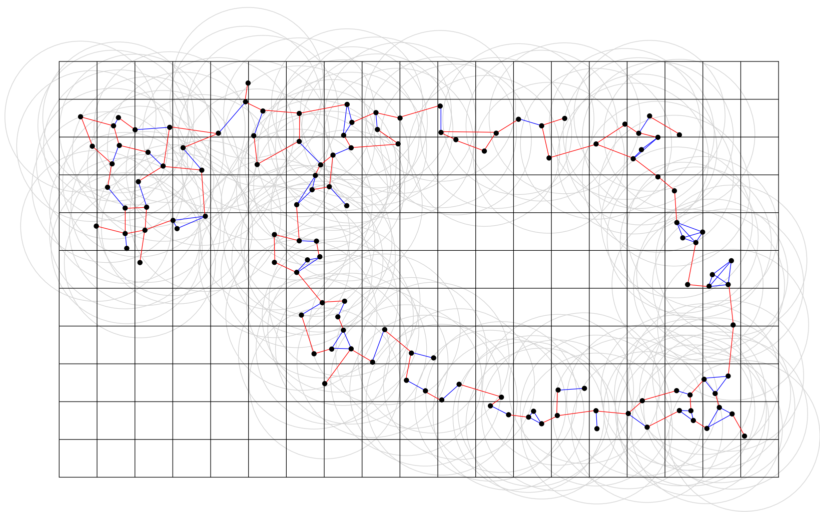

Fig. 3. Planar spanner H

Theorem 1. *The inter-cluster graph* $G_3 = (X, E_3)$ *is planar.*

To prove this theorem, we proceed in several steps (the proofs of Propositions 2, 4, 6, and 7 are postponed to the next section while the proofs of Propositions 5 and 8 are omitted due to space limitation). First, we classify the edges of G according to the relative positions of the squares containing their end-vertices (see Fig. 1). Then, using this classification, we consider all possible crossing configurations between two edges of E_1. Proposition 3 analyzes these configurations and shows that in most cases one of the two crossing edges has a replacement path.

Proposition 2. *If the crossing configuration between two edges* $xy, x'y' \in E_1'$ *does not belong to the list of Fig. 4, then one of these edges, say* xy*, has a replacement path passing via* $\pi(x')$ *or* $\pi(y')$*.*

Since the edges of E_1 do not admit replacement paths, we deduce that only a few crossing configurations may occur between two edges of E_1.

Proposition 3. *For two edges of* E_1*, there exist only seven possible crossing configurations listed in Fig. 4.*

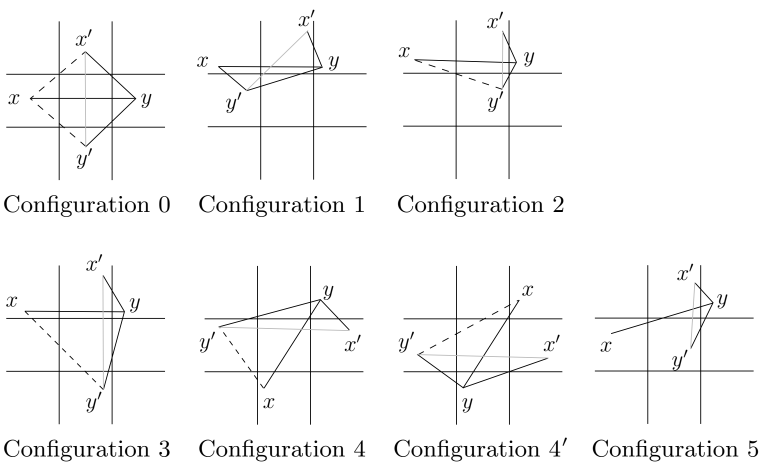

Fig. 4. The seven remaining configurations after Step 2. The solid lines indicate the edges in E_1' and the dashed lines indicate the edges that may be or not in E_1'.

Now we consider the edges added at Step 4. To prove that the graph returned by our algorithm is planar, we will show that these new edges do not intersect each other and do not intersect the edges that survive Step 3. Let $e' \in E_3 - E_2$ be an edge added during Step 4, let $e \in E_3 \cap E_2$ be an edge crossing e' that survived Step 3. Let also $xy, x'y' \in E_1$ be the crossing edges due to which the edge $x'y'$ has been removed and the edge $e' = \pi(y)\pi(y')$ has been added. By a case analysis of all configurations listed in Fig. 4, we verify that in all cases except Configuration 0 and 4′ (easily treated separately) there is a replacement path for e' in G_1 passing via $\pi(x')$. Since the edge e' is added at Step 4 only if it does not admit a replacement path in G_2, the following result excludes the existence of a replacement path for e' distinct from the path going through $\pi(x')$.

Proposition 4. *If the edge $\pi(y)\pi(y')$ from Fig. 4 has a replacement path in G_1 distinct from the path that goes through $\pi(x')$, then there is a replacement path between $\pi(y)$ and $\pi(y')$ in G_2.*

To deal with the crossing configurations formed by the edges e and e', we distinguish two cases. Then either such a configuration belongs to the list from Fig. 4 and a case analysis lead to a contradiction, or, by Proposition 2, one of these two edges admits a replacement path that passes through a square containing an end-vertex of the other edge. This edge must be e' because e was not removed during Step 2. However, as noticed above, Proposition 4 implies that this path must be the one passing through $\pi(x')$, otherwise this replacement path would survive Step 3 and e' would not be added. Since the replacement path arising in Proposition 2 passes via a square containing an end-vertex of e, the edge e must be incident to $\pi(x')$. Therefore, in the proof of Proposition 5 we analyze all possible crossings between e' and an edge incident to $\pi(x')$.

Proposition 5. *An edge from $E_3 - E_2$ cannot cross an edge from $E_3 \cap E_2$.*

Finally, we prove that the edges added during Step 4 do not cross each other. First, notice that if two edges of $E_3 - E_2$ cross, then one of them cross an edge from E_2. By Proposition 5, this edge of E_2 was removed during Step 3. Hence, we get a crossing between an edge of $E_3 - E_2$ and an edge of $E_2 - E_3$. The list of these configurations can be extracted from the proof of Proposition 5. They are analyzed case by case to get the following proposition (and to conclude the outline of the proof of Theorem 1):

Proposition 6. *The edges of $E_3 - E_2$ cannot cross.*

4.2 Computing E_0

We will consider several choices for the set of edges E_0 that interconnect the vertices lying in the same square. Let $\alpha(E_0)$ be the maximum distance in the graph $G_0 = (X, E_0)$ between two vertices belonging to the same square. In the next subsection, the hop stretch factor of our spanner is expressed as a function of $\alpha(E_0)$. A possible choice for E_0 is to set a clique or a star on terminals on each non-empty square, yielding $\alpha(E_0) = 1$ and $\alpha(E_0) = 2$, respectively. In this case, the diameter of the clusters is small but we do not get a planar spanner. The following proposition shows that an appropriate choice of E_0 ensures both planarity and constant diameter of clusters.

Proposition 7. *There exists a set of intra-cluster edges E_0 such that $H = (X, E_3 \cup E_0)$ is planar and $\alpha(E_0) \leq 44$.*

4.3 Hop stretch factor

Now, we analyze the hop stretch factor of H, namely we show that H is a $10\alpha(E_0)+9$ hop spanner for G. As noticed above, it suffices to prove the spanner property for two adjacent vertices of G. After Step 1, we get a spanner H' whose hop stretch factor is at most $2\alpha(E_0)+1$. Since the maximal l_1-length of an edge of a Unit Disk Graph G is 3, if an edge from E_1 is removed during Step 2, then it is replaced by a path containing at most 3 edges from E_2 for an edge of type B. Taking into account the edges from E_0, we obtain a stretch factor $\leq 4\alpha(E_0)+3$. In Step 3, again some edges are removed and replaced by paths. The following proposition asserts that the resulting length of these paths is bounded. Finally, in Step 4 edges are only added, thus the stretch factor cannot be worsened.

Proposition 8. *If an edge e of E_1 is removed during Step 3, then it is replaced in E_3 by a path containing at most $3l_1(e) \leq 9$ edges.*

Proposition 8 shows that after Step 4 an inter-cluster edge of G is replaced by at most 9 inter-cluster edges of E_3. Taking into account the edges of E_0, we get a final stretch factor equal to $10\alpha(E_0) + 9$. Summarizing, here is the main result of this paper:

Theorem 2. *Given a set X of n terminals in the plane, the graph $H = (X, E_0 \cup E_3)$ computed by our algorithm is a planar spanner of the Unit Disk Graph G of X with hop stretch factor at most $10\alpha(E_0) + 9$.*

5 Proof of Theorem 2

In this section, we sketch the proofs of most intermediate results necessary to establish Theorem 2. Most of these proofs consist in a case analysis of several configurations of points, which are treated using similar arguments. Due to space limitation and to the repetitive nature of most proofs, in each proof we provide a complete analysis only of some (most) representative configurations.

5.1 Preliminary results

We start with few simple useful observations.

Lemma 1. *If $[x, y]$ and $[x', y']$ are two crossing line segments, then either $|xx'| < |xy|$ or $|yy'| < |x'y'|$.*

Proof. Let $p = [x, y] \cap [x', y']$. By applying the triangle inequality to the triplets (x, p, x') and (y, p, y'), we get $|xx'|+|yy'| < |xp|+|px'|+|yp|+|py'| = |xy|+|x'y'|$, whence at least one of the inequalities $|xx'| < |xy|$ or $|yy'| < |x'y'|$ holds. □

Lemma 2. *If xy and $x'y'$ are two crossing edges of G, then at least one edge from xx', yy' and one edge from $xy', x'y$ belongs to G.*

Lemma 3. *If xy and $x'y'$ are two crossing edges from E'_1 (i.e., shortest edges between two squares), then the vertices x, y, x', y' belong to distinct squares of Γ.*

Proof. Suppose that x and x' belong to a square π. By Lemma 1, either $|xy'| < |x'y'|$ and $x'y'$ is not the shortest edge between $\pi(x')$ and $\pi(y')$, or $|x'y| < |xy|$ and xy is not the shortest edge between $\pi(x)$ and $\pi(y)$, a contradiction with the assumption that xy and $x'y'$ belong to E'_1. □

Lemma 4. *Let xy an edge of G. If a vertex z belongs to rectangle $R(xy)$ having xy as diagonal, then $xz, zy \in G$ and thus the edge xy does not belong to E_1.*

Proof. As z belongs to a right triangle having xy as hypotenuse, we get $|zx| < |xy|$ and $|zy| < |xy|$. Since $xy \in G$, we deduce that xz and zy also belong to G, yielding that xy does not belong to E_1. □

Lemma 5. *If $\pi(x)\pi(x'), \pi(x')\pi(y) \in G$ and $x' \in I(x, y)$, then $xy \notin E_1$.*

Lemma 6. *If xy is an edge of type A and z a vertex in the square between $\pi(x)$ and $\pi(y)$, then z is adjacent to at least one of the end-vertices of xy.*

5.2 Proof of Proposition 2

For each type of edges, we analyze the possible crossings between an edge xy of this type and another edge $x'y'$. We specify two subsets of squares $\{\pi_i, i = 1, ..., k\}$ and $\{\pi'_j, j = 1, ..., l\}$ with $x' \in \bigcup_{i=1}^{k} \pi_i$ and $y' \in \bigcup_{j=1}^{l} \pi'_j$ that cover, modulo rotations and symmetry, all possible crossing configurations (note that, by Lemma 3, the vertices x, y, x', y' must belong to distinct squares). We provide a complete analysis only for the case when the edge xy is of type A.

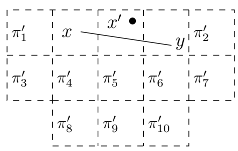

Fig. 5. Case $* \times A$: $x' \in I(x, y)$

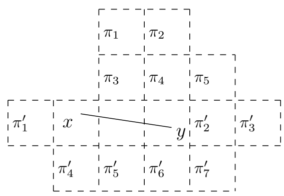

Fig. 6. Case $* \times A$: $x' \notin I(x, y)$

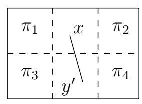

Fig. 7. $\pi(x)\pi(y')$

Case 1: $x' \in I(x, y)$ (Fig. 5). In this case, we assert that exactly one edge xx' or $x'y$ belongs to G. Indeed, by Lemma 6 at least one of these edges belongs to G. If both edges belong to G, then xy has a replacement path and does not belong to E_1. Suppose without loss of generality that xx' does not belong to G. By Lemma 3, we deduce that $yy' \in G$, thus $x'y, yy' \in G$. Furthermore, since $x'y'$ belongs to E_1, by Lemma 5 y does not belong to the interval between x' and y'. Then $y' \notin \pi'_2 \cup \pi'_6 \cup \pi'_7 \cup \pi'_{10}$. Since $xx' \notin G$, necessarily $y' \notin \pi'_1 \cup \pi'_3$ because, if y' belongs to one of these squares, then either $x \in R(x'y')$ or $x' \in R(xy)$ and

$xx' \in G$ by Lemma 4, a contradiction. If $y' \in \pi'_8$, then yy' does not belong to G, and by Lemma 3 $xx' \in G$, a contradiction. Finally, the cases $y' \in \pi'_4, \pi'_5$ or π'_9 correspond respectively to Configuration 1, 2 and 3 of Fig. 4.

Case 2: $x' \notin I(x, y)$ (Fig. 6). If $x' \in \pi_1 \cup \pi_2$, either $y' \in I(x, y)$ and this configuration was previously analyzed, or one end-vertex of xy belongs to $R(x'y')$ and by Lemma 4 the edge $x'y'$ does not belong to E_1. If $x' \in \pi_5$, then $y' \in \pi'_6$ and $y \in R(x'y')$, by Lemma 4 the edge $x'y'$ does not belong to E_1. For $x' \in \pi_4$, if $y' \in \pi'_6$ we get $y \in I(x', y')$ and this case was already analyzed. If $y' \in \pi'_2 \cup \pi'_3 \cup \pi'_7$, then $y \in R(x'y')$ and by Lemma 4 the edge $x'y'$ does not belong to E_1. Finally, for $x' \in \pi_3$, if $y' \in \pi'_4 \cup \pi'_6$, we obtain the same configuration as $x' \in \pi_4$ and $y' \in \pi'_5$. If $y' \in \pi'_1 \cup \pi'_2$, one end-vertex of the edge xy belongs to $R(x'y')$ and by Lemma 4 the edge $x'y'$ does not belong to E_1. The cases $(x' \in \pi_3, y' \in \pi'_5)$ and $(x' \in \pi_4, y' \in \pi'_5)$ correspond to Configuration 0, 4 and 4′ of Fig. 4.

5.3 Proof of Proposition 4

For each crossing remaining after Step 2 (configurations listed in Fig. 4) such that the edge $\pi(y)\pi(y')$ admits a replacement path in G_1 that does not pass via $\pi(x')$ we will prove that such a replacement path also exists in G_2. We consider each edge of the replacement path of G_1 and show that either this edge is not removed in Step 2, or that there exists another replacement path for the edge $\pi(y)\pi(y')$ in G_2. We provide a complete analysis only for Configuration 1 of Fig. 4, all other configurations are treated analogously.

First, notice that, if the edge xy' is not crossed by another edge of E_1, then together with the edge xy it forms a replacement path for the edge $\pi(y)\pi(y')$. We assert that the edge $\pi(x)\pi(y')$ always belongs to G. Indeed, since $xy \in G$, x belongs to the right half of $\pi(x)$. The vertex x' belongs to the right half of $\pi(x')$ because otherwise xx' would belong to G, excluding, by Lemma 5, the edge xy from being in E_1. Furthermore, the vertex x is above x' and to the left of y', otherwise, either $x' \in R(xy)$ or $x \in R(x'y')$ and, by Lemma 4, either $x'y'$ or xy does not belong to E_1. We conclude that $|x'y'| > |xy'|$ and since $x'y'$ belongs to G, $\pi(x)\pi(y')$ also belongs to G. If the edge $\pi(x)\pi(y')$ is not crossed by another edge of E_1, then we are done. Otherwise, let $ab \in E_1$ be an edge crossing $\pi(x)\pi(y')$. Since we should only consider the configurations listed in Fig. 4, we can suppose that $a \in \pi_2 \cup \pi_4$ and $b \in \pi_1 \cup \pi_3$ (see Fig. 7).

First, suppose that $a \in \pi_2 = \pi(x')$. From $ab \in G$ we deduce that a belongs to the left half of $\pi(x')$. As already noticed, x belongs to the right half of $\pi(x)$ and thus $xx' \in G$, by Lemma 5 a contradiction with $xy \in E_1$. Now if $a \in \pi_4$, from $ab, yy' \in G$, we can deduce that a belongs to left half of $\pi(a)$ and y' belongs to the right half of $\pi(y')$, then $y'a \in G$. If $b \in \pi_1$ by Proposition 3, ab and $x'y'$ cannot cross. Hence, either $y' \in R(ab)$ or $a \in R(x'y')$ and by Lemma 5 both contradict $ab, x'y' \in E_1$. Finally, if $b \in \pi_3$, there is no edge between b and a vertex of $\pi(y')$. Indeed, by Lemma 5, such an edge would exclude ab from being in E_1. Therefore, both y' and the end-vertex z of $\pi(x)\pi(y')$ are located outside the ball of radius ab centered in b. Since ab crosses the edge $\pi(x)\pi(y')$, z is located below ab. Since

$\pi(x)\pi(y')$ is a shortest edge between $\pi(x)$ and $\pi(y')$, y' cannot lie above ab. As previously noticed, ab cannot cross $x'y'$, the vertex a must belong to $R(x'y')$, by Lemma 4 a contradiction with $x'y' \in E_1$. This concludes the analysis of Configuration 1. The analysis of six remaining configurations is analogous.

5.4 Proof of Proposition 6

Assume by way of contradiction that two edges e and f of $E_3 - E_2$ cross. Since these edges have been added to restore a path between the end-vertices of an edge from E_1, one of them, say e, crosses the edge e' of E_1 because of which the edge f has been added. Therefore, to derive a contradiction and establish Proposition 6, it is sufficient to verify that the edge f added to replace the edge e' does not cross the edge e. This is actually an immediate consequence of Lemma 3 since for each possible crossing configuration between an edge $e' = x'y'$ and an edge $e = xy$ listed in Fig. 4, the edges $f = \pi(y)\pi(y')$ and $e = xy$ have an end-vertex in $\pi(y)$ and thus cannot cross.

5.5 Proof of Proposition 7

The set E_0 can be obtained as follows. First, we partition X into clusters consisting of all vertices which belong to the same non-empty square of the grid Γ. Then, we consider each non-empty square π crossed by an edge $e \in E_3$ (one can show that each square of Γ can be crossed by at most one edge) and so that the two regions R_1 and R_2 into which e partitions the square π are both non-empty. First notice that only edges of type A may define such partitions. Indeed, any edge of type O crosses only the squares containing its end-vertices while any edge of type B or C crossing a non-empty square defines a right triangle and, by Lemma 4, any terminal in this triangle yields the existence of a replacement path for this edge.

Next, we partition the terminals located in π into two subsets corresponding to the regions R_1 and R_2. We consider each pair of adjacent partitions (i.e. the partitions whose regions share a common side) and we add to a set Y the shortest edge between them if one exists. Then, for each partition, we compute a minimum spanning tree on those vertices of the partition that are connected by edges to vertices outside π (i.e., the vertices of $V(Y) \cup V(E_3)$). Finally, we connect each vertex of π that does not belong to $V(Y) \cup V(E_3)$ to its closest neighbor in the spanning tree of its partition. We claim that the two subsets arising from a square π crossed by an edge of E_3 are connected by a path passing through at most one neighboring square of π. Indeed, consider a pair of vertices x and y that belong to distinct partitions of π. Since xy and e are crossing edges of G, from Lemma 2 and the fact that e has no replacement path, we deduce that one end-vertex of e is adjacent to both x and y. This shows that indeed x and y are joined by a path passing via a single neighboring square. As already noticed in Section 3, there are at most 20 edges of E_3 (see Fig. 2) having an end-vertex in a given square. We deduce that there is a path of length at most 44 between every pair of vertices of the same square of Γ.

It remains to show that the edges of E_0 do not cross each other and do not cross the edges of E_3. First, consider an edge xy connecting two neighboring partitions (i.e., an edge of E_3 or an edge of E_0 added due to splitting of a square) and suppose that xy crosses an edge $x'y'$ of a minimum spanning tree T. One end-vertex of xy, say x, must belong to the square containing x' and y', otherwise xy partitions this square into two non-empty regions. Since $x'y' \in E(T)$, among xx' and xy', one edge, say xx', is at least as long as $x'y'$. Hence, by Lemma 1, yy' is shorter than xy, a contradiction with the choice of xy. The proof is analogous if instead of an edge of T, we consider an edge $x'y'$ added to connect x' to its closest neighbor y' in $V(T)$ (by the choice of $x'y'$, again $|xx'| \geq |x'y'|$). Finally, if $x'y'$ connects two neighboring partitions, then the edges xy and $x'y'$ must have an end-vertex in the same square and we get a contradiction with Lemma 3. Now, it remains the case of two edges having their end-vertices in the same partition. Clearly, two edges of a minimum spanning tree cannot cross and the same holds for two edges connecting a terminal to its closest neighbor in $V(T)$. Now, suppose that x is the closest neighbor of y in $V(T)$ and that xy crosses $x'y' \in E(T)$. Since $x'y' \in E(T)$, either xx' or xy', say xx', is at least as long as $x'y'$. Hence, by Lemma 1, yy' is shorter than xy, a contradiction with the choice of xy. This concludes the proof of Proposition 7.

References

1. K. Alzoubi, X. Li, Y. Wang, P. Wan, and O. Frieder, Geometric spanners for wireless ad hoc networks, IEEE Trans. Par. Dist. Syst. 14 (2003), 408–421.
2. P. Bose, L. Devroye, W. Evans, and D. Kirkpatrick, On the spanning ratio of Gabriel graphs and β-skeletons, SIAM J. Discr. Math. 20 (2006) 412–427.
3. P. Bose and M. Smid, On plane geometric spanners : a survey and open problems, submitted, 2009.
4. J. Chen, A. Jiang, I. A. Kanj, G. Xia, and F. Zhang, Separability and topology control of quasi unit disk graphs, INFOCOM 2007, pp. 2225–2233.
5. D. Eppstein, Spanning trees and spanners, In J.-R. Sack and J. Urrutia, editors, Handbook of Computational Geometry, Elsevier Science Publishers B.V., North-Holland, Amsterdam (2000), pp. 425–461.
6. M. Fürer and S.P. Kasiviswanathan, Spanners for geometric intersection graphs, WADS 2007, pp. 312–324.
7. J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu, Discrete mobile centers, Discr. Comput. Geom. **30(1)** (2003), 45–63.
8. J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, Geometric spanner for routing in mobile networks, ACM MobiHoc 2001, pp. 45–55.
9. X. Li, G. Călinescu, and P. Wan, Distributed construction of planar spanner and routing for ad hoc wireless networks, INFOCOM 2002.
10. G. Narasimhan and M. Smid, Geometric Spanner Networks, Cambridge University Press, 2007.
11. S. Rührup, Position-based Routing Strategies, PhD Thesis, University of Paderborn, Germany, 2006.
12. C. Yan, Y. Xiang, and F.F. Dragan, Compact and low delay routing labeling scheme for unit disk graphs, WADS 2009, pp. 566–577.