Plane Hop Spanners for Unit Disk Graphs
Simpler and Better

Ahmad Biniaz[*]

Cheriton School of Computer Science
University of Waterloo
ahmad.biniaz@gmail.com

Abstract

The unit disk graph (UDG) is a widely employed model for the study of wireless networks. In this model, wireless nodes are represented by points in the plane and there is an edge between two points if and only if their Euclidean distance is at most one. A *hop spanner* for the UDG is a spanning subgraph H such that for every edge (p, q) in the UDG the topological shortest path between p and q in H has a constant number of edges. The *hop stretch factor* of H is the maximum number of edges of these paths. A hop spanner is *plane* (i.e. embedded planar) if its edges do not cross each other.

The problem of constructing hop spanners for the UDG has received considerable attention in both computational geometry and wireless ad hoc networks. Despite this attention, there has not been significant progress on getting hop spanners that (i) are plane, and (ii) have low hop stretch factor. Previous constructions either do not ensure the planarity or have high hop stretch factor. The only construction that satisfies both conditions is due to Catusse, Chepoi, and Vaxès (2010); their plane hop spanner has hop stretch factor at most 449.

Our main result is a simple algorithm that constructs a plane hop spanner for the UDG. In addition to the simplicity, the hop stretch factor of the constructed spanner is at most 341. Even though the algorithm itself is simple, its analysis is rather involved. Several results on the plane geometry are established in the course of the proof. These results are of independent interest.

1   Introduction

Computational geometry techniques are widely used to solve problems, such as topology construction, routing, and broadcasting, in wireless ad hoc networks. A wireless ad hoc network is usually modeled as a unit disk graph (UDG). In this model wireless devices are represented by points in the plane and assumed to have identical unit transmission radii. There exists an edge between two points if their Euclidean distance is at most one unit; this edge indicates that the corresponding devices are in each other's transmission range and can communicate.

A *geometric graph* is a graph whose vertices are points in the plane and whose edges are straight-line segments between the points. A geometric graph is *plane* if its edges do not cross each other. Let G be a geometric graph. A *topological shortest path* between any two vertices u and v in G is a path that connects u and v and has the minimum number of edges. The *hop distance* $h_G(u, v)$ between u and v is the number of edges of a topological shortest path between them.

---

[*]Supported by NSERC Postdoctoral Fellowship

For a point set P in the plane, the *unit disk graph* $UDG(P)$ is a geometric graph with vertex set P that has an edge between two points p and q if and only if their Euclidean distance $|pq|$ is at most 1. A *hop spanner* for $UDG(P)$ is a spanning subgraph H such that for any edge $(p, q) \in UDG(P)$ it holds that $h_H(p, q) \leqslant t$, where t is some positive constant. The constant t is called the *hop stretch factor* of H. This definition of hop spanner H implies that for any two points p and q in P (not necessarily connected by an edge of $UDG(P)$) it holds that $h_H(p, q) \leqslant t \cdot h_{UDG(P)}(p, q)$. In this paper we study the problem of constructing UDG hop spanners that are plane and have low hop stretch factor.

We note that for general graphs we cannot hope to always get a planar hop spanner. For example take the complete graph with five vertices, which is not planar, and replace each of its edges by an arbitrary long path. The resulting graph—which is still not planar—does not have any planar hop spanner (with constant hop stretch factor).

The *Euclidean spanner* and *Euclidean stretch factor* are defined in a similar way, but for the distance measure they use the total Euclidean length of path edges. Both hop spanners and Euclidean spanners have received considerable attention in computational geometry and wireless ad hoc networks; see e.g. the surveys by Eppstein [8], Bose and Smid [4], Li [14], and the book by Narasimhan and Smid [17]. Unit disk graph spanners have been used to reduce the size of a network and the amount of routing information. They are also used in topology control for maintaining network connectivity, improving throughput, and optimizing network lifetime; see the surveys by Li [14] and Rajaraman [18]. Constructions of UDG spanners, both centralized and distributed, also with additional properties like planarity and power saving have been widely studied [11, 15, 16, 13]. Researchers also studied the construction of spanners for general disk graphs [9] and for quasi unit disk graphs [6].

1.1 Related Work

In this section we review some previous attempts towards getting plane hop spanners for unit disk graphs. Gao et al. [11] proposed a randomized algorithm that constructs a spanner with constant Euclidean and hop stretch factors. They use a hierarchical clustering algorithm of [10] to create several clusters of points each containing a point as the clusterhead. Then they connect the clusters by a restricted Delaunay graph, and then connect the remaining points to clusterheads. The restricted Delaunay graph can be maintained in a distributed manner when points move around. Although the underlying restricted Delaunay graph is plane, the entire spanner is not. This spanner has constant Euclidean stretch factor in expectation, and constant hop stretch factor for some unspecified constant.

Alzoubi et al. [2] proposed a distributed algorithm for the construction of a hop spanner for the UDG. Their algorithm integrates the connected dominating set and the local Delaunay graph of [15] to form a backbone for the spanner. Although the backbone is plane, the entire spanner is not. The hop stretch factor of this spanner is at most 15716 (around 15000 as estimated in [5]).

Yan, Xiang, and Dragan [19] showed a balanced separator result for unit disk graphs. By applying this result recursively on any n-vertex unit disk graph G, they obtain a system of $O(\log n)$ spanning trees such that for any two vertices p and q there exists a spanning tree T for which $h_T(p, q) \leqslant 3 \cdot h_G(p, q) + 12$. This immediately gives a hop spanner with $O(n \log n)$ edges for G. Although this spanner is not planar, it has some interesting properties, e.g., it has small hop stretch factor (which is 15) and it gives a compact routing scheme with guaranteed delivery for G.

To the best of our knowledge, the only construction that guarantees the planarity of the entire hop spanner is due to Catusse, Chepoi, and Vaxès [5]. First they use a regular square-grid to partition input points into clusters. Then they add edges between points in different clusters, and

also between points in the same cluster to obtain a hop spanner, which is not necessarily plane. Then they go through several steps and in each step they remove some edges to ensure planarity, and add some new edges to maintain constant hop stretch factor. At the end they obtain a plane hop spanner with hop stretch factor at most 449. This spanner can be obtained by a localized distributed algorithm.

1.2 Our Contribution

Our main contribution in this paper is a polynomial-time simple algorithm that constructs a plane hop spanner, with hop stretch factor at most 341, for unit disk graphs. Our algorithm works as follows: Given a set P of points in the plane, we first select a subset S of P (in a clever way), then compute a plane graph $DT_1(S)$ (which is the Delaunay triangulation of S minus edges of length more than 1), and then connect every remaining point of P to its closest visible vertex of $DT_1(S)$. In addition to improving the hop stretch factor, this algorithm is straightforward and the planarity proof is simple, in contrast to that of Catusse et al. [5]. Our analysis of hop stretch factor is still rather involved. Towards the correctness proof of our algorithm, we prove several results on the plane geometry, which are of independent interest. Our construction uses only local information and can be implemented as a localized distributed algorithm.

Catusse et al. [5] also showed a simple construction of a hop spanner, with hop stretch factor 5 and with at most $10n$ edges, for any n-vertex unit disk graph. With a simple modification to their construction we obtain such a spanner with at most $9n$ edges.

2 Preliminaries and Some Geometric Results

We say that a set of points in the plane is in *general position* if no three points lie on a straight line and no four points lie on a circle. Throughout this paper, every given point set is assumed to be in general position. For a set P of points in the plane, we denote by $DT(P)$ the Delaunay triangulation of P. Let p and q be any two points in the plane. We denote by pq the straight-line segment between p and q, and by $\overrightarrow{pq}$ the ray that emanates from p and passes through q. The *diametral disk* $D(p,q)$ between p and q is the disk with diameter $|pq|$ that has p and q on its boundary. Every disk considered in this paper is closed, i.e., the disk contains its boundary circle.

Consider the Delaunay triangulation of a point set P. In their seminal work, Dobkin, Friedman, and Supowit [7] proved that for any two points $p, q \in P$ there exists a path, between p and q in $DT(P)$, that lies in the diametral disk between p and q. Their proof makes use of Voronoi cells (of the Voronoi diagram of P) that intersect the line segment pq. In the following theorem we give a simple inductive proof for a more general claim that shows the existence of such a path in any disk (not only the diametral disk) between p and q.

Theorem 1. *Let P be a set of points in the plane in general position and let $DT(P)$ be the Delaunay triangulation of P. Let p and q be any two points of P and let D be any disk in the plane that has p and q on its boundary. There exists a path, between p and q in $DT(P)$, that lies in D.*

Proof. We prove this theorem by induction on the number of points in D. If D does not contain any point of $P \setminus \{p, q\}$ in its interior then (p, q) is an edge of $DT(P)$, and thus (p, q) is a desired path. Assume that D contains a point $r \in P \setminus \{p, q\}$ in its interior. Let c be the center of D. Consider the ray $\overrightarrow{pc}$. Fix D at p and shrink it along $\overrightarrow{pc}$ until r becomes on its boundary circle; see Figure 1. Denote the resulting disk by D_{pr}; this disk lies fully in D. Compute the disk D_{qr} in a similar fashion by shrinking D along $\overrightarrow{qc}$. Since r is in the interior of D, the disk D_{pr} does not contain q and the disk D_{qr} does not contain p. Thus, the number of points in each of D_{pr} and D_{qr}
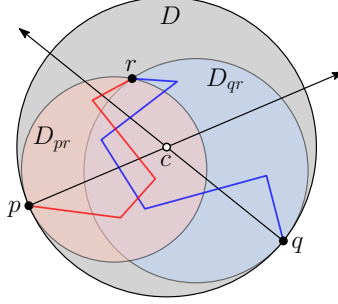
3

Figure 1: Illustration of the proof of Theorem 1.

is smaller than that of D. Therefore, by induction hypothesis there exists a path, between p and r in $DT(P)$, that lies in D_{pr}, and similarly there exists a path, between q and r in $DT(P)$, that lies in D_{qr}. The union of these two paths contains a path, between p and q in $DT(P)$, that lies in D. □

Let G be a plane geometric graph and let $p \notin G$ be any point in the plane. We say that a vertex $q \in G$ is *visible* from p if the straight-line segment pq does not cross any edge of G. One can simply verify that for every p such a vertex q exists. Among all vertices of G that are visible from p, we refer to the one that is closest to p by the *closest visible vertex* of G from p.

The following theorem (though simple) turns out to be crucial in the planarity proof of our hop spanner; this theorem is of independent interest. Although it answers a basic question, we were unable to find such a result in the literature; there exist however related results, see e.g. [1, 3, 12].
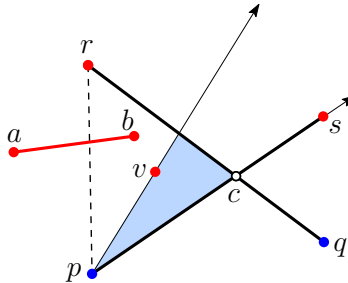

Figure 2: Illustration of the proof of Theorem 2. The red vertices belong to G.

Theorem 2. *Let G be a plane geometric graph, and let Q be a set of points in the plane that is disjoint from G. The graph, that is obtained by connecting every point of Q to its closest visible vertex of G, is plane.*

Proof. Let E be the set of edges that connect every point of Q to its closest visible vertex of G. To prove the theorem, it suffices to show that the edges of $G \cup E$ do not cross each other. The edges of G do not cross each other because G is plane. It is implied from the definition of visibility that the edges of E do not cross the edges of G.

It remains to prove that the edges of E do not cross each other. We prove this by contradiction. To that end consider two crossing edges (p, s) and (q, r) of E where p, q are two points of Q and s, r are two vertices of G. Let c be their intersection point of (p, s) and (q, r). By the triangle inequality we have $|pr| < |ps|$ or $|qs| < |qr|$. After a suitable relabeling assume that $|pr| < |ps|$, and thus p is closer to r than to s. The reason that p was not connected to r, is that r is not

4

visible from p. Therefore there are edges of G that block the visibility of r from p. Take any such edge (a, b). The edge (a, b) does not intersect any of (p, s) and (q, r) because otherwise (a, b) blocks the visibility of s from p or the visibility of r from q, and as such we wouldn't have these edges in E; see Figure 2. Therefore, exactly one endpoint of (a, b), say b, lies in the triangle $\triangle pcr$. Rotate the ray $\overrightarrow{ps}$ towards b and stop as soon as hitting a vertex of G in $\triangle pcr$. This vertex is visible from p. Denote this vertex by v (it might be that $v = b$). Since v lies in $\triangle pcr$, it turns out that $|pv| \leqslant \max\{|pr|, |pc|\} < |ps|$. Thus, v is a closer visible vertex of G from p. This contradicts the fact that s is a closest visible vertex from p. $\qquad\square$

Lemma 1. *Let C be a convex shape of diameter d in the plane, and let pq be a straight-line segment that intersects C. Then for any points $r \in C$ we have $\min\{|rp|, |rq|\} \leqslant \sqrt{d^2 + |pq|^2/4}$.*

Proof. Let s be a point in the intersection of C and pq. Let l_{pq} be the line through pq, and let c be the point of l_{pq} that is closest to r. Observe that $\triangle rsc$ is a right triangle with hypotenuse rs, and thus $|rc| \leqslant |rs| \leqslant d$. If an endpoint of pq lies on segment sc (as depicted in Figure 3) then the distance from r to that endpoint is at most d. Assume that no endpoint of pq lies on sc, and thus c lies on pq. After a suitable relabeling assume that c is closer to p than to q, and thus $|cp| \leqslant |pq|/2$. In this setting, by the Pythagorean equation we get $|rp| = \sqrt{|rc|^2 + |cp|^2} \leqslant \sqrt{d^2 + |pq|^2/4}$. $\qquad\square$
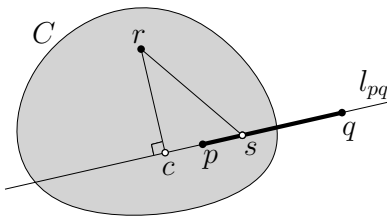

Figure 3: Illustration of the proof of Lemma 1

We refer to a hop spanner with hop stretch factor t as a *t-hop spanner*. Catusse et al. [5] showed a simple construction of a sparse 5-hop spanner with at most $10n$ edges, for any n-vertex unit disk graph. With a simple modification to their construction we obtain a 5-hop spanner with at most $9n$ edges.
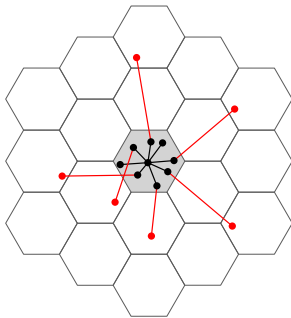

Figure 4: Every cell can have edges to at most 18 other cells.

Theorem 3. *Every n-vertex unit disk graph, has a 5-hop spanner with at most $9n$ edges.*

Proof. Consider the unit disk graph $UDG(P)$ on any set P of n points in the plane. Consider a regular hex-grid on the plane with hexagons (cells) of diameter 1. In every nonempty cell π pick

a point as the *center* and connect it to all other points in this cell; these edges are in $UDG(P)$ because the diameter of π is 1. Then take exactly one edge of $UDG(P)$ between any two cells if such an edge exists; from each cell we can have edges to at most 18 other cells as depicted in the Figure 4 (Catusse et al. [5] use a square-grid in which every cell can have edges to at most 20 other cells). We claim that the resulting graph, which we call it H, is a desired spanner. By a counting argument one can verify that H has at most $9n$ edges. To verify the hop stretch factor consider two points $p, q \in P$. If p and q are in the same cell, then there is a path, of length at most 2 between p and q in H, that goes through the center of the cell. Assume that p and q lie in different cells, say $\pi(p)$ and $\pi(q)$. By our construction there is an edge, say (p', q'), in H between $\pi(p)$ and $\pi(q)$. Therefore, there is a path of length at most 5 between p and q, that goes through p', q', and through the centers of $\pi(p)$ and $\pi(q)$. □

3 Plane Hop Spanner Algorithm

This section presents our main contribution which is a polynomial-time algorithm for construction of plane hop spanners for unit disk graphs.

Let P be a set of points in the plane in general position, and let $UDG(P)$ be the unit disk graph of P. Our algorithm first partitions P into some clusters by using a regular square-grid; this is a standard initial step in many UDG algorithms, see e.g. [5, 6]. We use this partition to select a subset S of P that satisfies some properties, which we will describe later. Then we compute the Delaunay triangulation of S and remove every edge that has length more than 1. We denote the resulting graph by $DT_1(S)$. Then we connect every point of $P \setminus S$ to its closest visible vertex of $DT_1(S)$. Let H denote the final resulting graph. We claim that H is a plane hop spanner, with hop stretch factor at most 341, for $UDG(P)$. In Section 3.1 we show how to compute S. The points of S are distributed with constant density, i.e., there are $O(1)$ points of S in any unit disk in the plane. Based on this and the fact that $DT_1(S)$ has only edges of length at most 1, $DT_1(S)$ can be computed by a localized distributed algorithm (see [2] for sufficient conditions to construct a graph locally by a distributed algorithm, and see [11] for more details on the construction). In Section 3.2 we prove the correctness of the algorithm that H is plane and H is a subgraph of $UDG(P)$. In Section 3.3 we analyze the stretch factor of H. The following theorem summarizes our result in this section.

Theorem 4. *There exists a plane* 341*-hop spanner for the unit disk graph of any set of points in the plane in general position. Such a spanner can be computed in polynomial time.*

3.1 Computation of S

In this section, we compute the subset S; we will see properties of S at the end of this section. Let Γ be a regular square-grid on the plane with squares of diameter 1. The side-length of these squares is $1/\sqrt{2}$. Without loss of generality we assume that no point of P lies on a grid line (this can be achieved by moving the grid by a small amount horizontally and vertically). Let E be the edge set containing the shortest edge of $UDG(P)$ that runs between any two nonempty cells of Γ if such an edge exists. Since every edge of $UDG(P)$ has length at most 1, for every cell π there are at most 20 edges in E going from π to other cells $\pi_1, \ldots, \pi_{20}$ as depicted in Figure 5. Let $V(E)$ be the set of endpoints of E, i.e., endpoints of the edges of E. The set $V(E)$ has the following two properties:

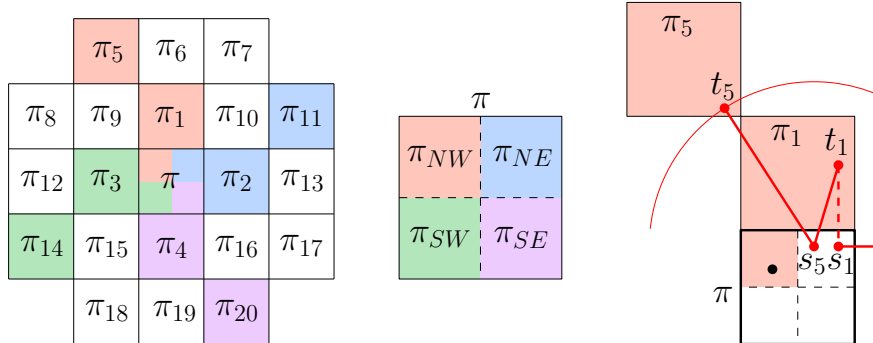- Every cell of Γ contains at most 20 points of $V(E)$.

Figure 5: Illustration of the computation of S.

- For every cell $\pi \in \Gamma$ and every $i \in \{1, \ldots, 20\}$ if there is an edge in $UDG(P)$ between π and π_i, then there are two points $s_i, t_i \in V(E)$ such that $s_i \in \pi$, $t_i \in \pi_i$, and (s_i, t_i) is the shortest edge of $UDG(P)$ that runs between π and π_i.

We want to modify the edge set E and also compute a point set T such that $V(E) \cup T$ satisfies some more properties that we will see later. To that end we partition every cell π of Γ into four sub-cells of diameter $1/2$, namely $\pi_{NW}, \pi_{NE}, \pi_{SW}, \pi_{SE}$ as in Figure 5. For each cell π, consider four triplets (π_{NW}, π_1, π_5), $(\pi_{NE}, \pi_2, \pi_{11})$, $(\pi_{SW}, \pi_3, \pi_{14})$, and $(\pi_{SE}, \pi_4, \pi_{20})$; these triplets are colored in Figure 5. Let T be the empty set. We perform the following three-step process on each of the four triplets of every cell π. We describe the process only for (π_{NW}, π_1, π_5); the processes of other triplets are analogous. In our description "a point of π_{NW}" refers to a point of P that lies in π_{NW}.

1. If π_{NW} is empty (contains no point of P) then we do nothing and stop the process. Assume that π_{NW} contains some points of P. If π_{NW} contains an endpoint of E, i.e. an endpoint of some edge of E, then we do nothing and stop the process.

2. Assume now that π_{NW} contains some points of P but does not contain any endpoint of E. If there is no edge in E that runs between π and π_1 or between π and π_5 then we take a point of π_{NW} arbitrary and add it to T, and then stop the process.

3. Assume that E contains an edge between π and π_1, and an edge between π and π_5. We are now in the case where π_{NW} contains some points of P but not any endpoint of E, and both s_1 and s_5 exist. If $s_1 = s_5$, then we add a point of π_{NW} to T, and then stop the process. Assume that $s_1 \neq s_5$. Since π_{NW} does not contain any endpoint of E, the points s_1 and s_5 do not lie in π_{NW}. In particular, s_5 lies in sub-cell π_{NE} because the distance between π_5 and each of π_{SE} and π_{SW} is more than 1; however s_1 might lie in other sub-cells. In this setting the disk with center s_5 and radius $|s_5 t_5|$ contains the entire π_1 (see Figure 5), and thus the distance between s_5 and any point in π_1 is at most 1. We replace the edge (s_1, t_1) of E by the edge (s_5, t_1), which has length at most one; see Figure 5. Then we add a point of π_{NW} to T, and then stop the process. (We note that in the future steps when we process triplets of the cell π_1, the edge (s_5, t_1) might be replaced by another edge that runs between π_1 and π. This is okay for the purpose of Lemma 2 because we replace an edge between two cells by another edge between the same two cells.)

This is the end of process for triplet (π_{NW}, π_1, π_5). After performing this process on all triplets of all cells, we obtain an edge set E and a point set T. We define the subset S to be union of T

and the endpoints of edges of E, i.e., $S = V(E) \cup T$. We will use the properties in the following lemma in correctness proof and analysis of hop stretch factor.

Lemma 2. *The set S satisfies the following four properties:*

(P1) *Every cell of Γ contains at most 20 points of S.*

(P2) *For every cell π and every $i \in \{1, 2, 3, 4\}$, if there is an edge in $UDG(P)$ between π and π_i, then there are two points $s_i, t_i \in S$ such that $s_i \in \pi$, $t_i \in \pi_i$, and $|s_i t_i| \leqslant 1$.*

(P3) *For every cell π and every $i \in \{5, \ldots, 20\}$, if there is an edge in $UDG(P)$ between π and π_i, then there are two points $s_i, t_i \in S$ such that $s_i \in \pi$, $t_i \in \pi_i$, $|s_i t_i| \leqslant 1$, and (s_i, t_i) is the shortest edge of $UDG(P)$ that runs between π and π_i.*

(P4) *The set S contains at least one point from every nonempty sub-cell π_{NW}, π_{NE}, π_{SW}, π_{SE} of each cell π.*

Proof. Recall that E initially contains shortest edges between different cells. In step 3 we replace only the edges of E, that run between each cell π and the cells π_1, π_2, π_3, π_4, with new edges of length at most 1 between the same cells. Therefore properties (P2) and (P3) hold. Every nonempty sub-cell contains either a point in $V(E)$ (step 1) or a point in T (steps 2 and 3), and thus property (P4) holds.

To verify property (P1), we use the discharging method as follows. Consider one cell π. Before the process, we give charge 1 to each π_i for $i \in \{1, \ldots, 20\}$. Thus, the total available charge for π is 20. Then, for every edge $(s_i, t_i) \in E$ we move the charge of π_i to s_i. Since each s_i can be an endpoint of more than one edge of E, it may get charges from more than one cell. During the process we move charges as follows. In step 2, if there is no edge in E that runs between π and π_1 or between π and π_5, then we move the charge of π_1 or π_5 to the point of π_{NW} that we add to T, respectively. Now consider step 3. If $s_1 = s_5$ then s_1 has charge at least 2 that come from π_1 and π_5. In this case we move charge 1 from s_1 to the point of π_{NW} that we add to T. If $s_1 \neq s_5$, then after replacing (s_1, t_1) with (s_5, t_1), we move the charge of π_1 from s_1 to the point of π_{NW} that we add to T. After this replacement if s_1 is not an endpoint of any edge of E other than (s_1, t_1) then s_1 gets removed from $V(E)$, otherwise it still holds charges of some cells other than π_1. Thus, after processing π, the final charge of every point of S, that lies in π, is at least 1. Observe that π_1 and π_5 belong to only one of the four triplets that are associated to π, and thus we do not double count their charges. Since the total available charge for π was 20, it turns out that the number of points of S, that lie in π, is at most 20.

While processing each cell π, we add to T only points that lie in π. Moreover, the edge-replacement of step 3, does not add any new point to $V(E)$. After processing π, there is an edge in E running between π and π_i if and only if there was such an edge before processing π. Therefore, after processing all cells, every cell contains at most 20 points of S, and thus (P1) holds. $\square$

3.2   Correctness Proof

In this section we prove the correctness of our algorithm. Recall the grid Γ, and the subset S of P that is computed in Section 3.1. Recall that our algorithm computes the Delaunay triangulation $DT(S)$ and removes every edge of length more than 1 to obtain $DT_1(S)$, and then connects every point of $P \setminus S$ to its closest visible vertex of $DT_1(S)$. Let H denotes the resulting graph. One can simply verify that this algorithm takes polynomial time. Since $DT(S)$ is plane, its subgraph $DT_1(S)$ is also plane. It is implied from Theorem 2 (where $DT_1(S)$ and $P \setminus S$ play the roles of G

and Q) that H is plane. As we stated at the outset, except for the computation of S which is a little more involved, the algorithm and the planarity proof are straightforward.

To finish the correctness proof it remains to show that every edge of H has length at most 1. Consider any edge e of H. By our construction, either the two endpoints of e belong to S, or one endpoint of e belongs to S and its other endpoint belongs to $P \setminus S$. If both endpoints of e are in S, then e belongs to $DT_1(S)$ and hence has length at most 1. If one endpoint of e is in S and its other endpoint is in $P \setminus S$, then by following lemma the length of e is at most $1/\sqrt{2}$.
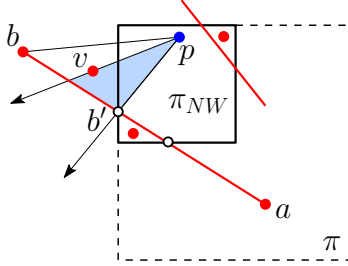

Figure 6: Illustration of the proof of Lemma 3. The red points belong to S.

Lemma 3. *The length of every edge of H, that has an endpoint in S and an endpoint in $P \setminus S$, is at most $1/\sqrt{2}$.*

Proof. Consider any edge $(p, s) \in H$ with $p \in P \setminus S$ and $s \in S$. By our construction, s is the closest visible vertex of $DT_1(S)$ from p. Thus, to prove the lemma, it suffices to show the existence of a vertex $v \in DT_1(S)$ that is visible from p and for which $|pv| \leqslant 1/\sqrt{2}$; this would imply that the distance between p and s, which is the closest visible vertex from p, is at most $1/\sqrt{2}$. In the rest of the proof we show the existence of such vertex v.

Let π be the cell that contains p (the dashed cell in Figure 6). After a suitable rotation we assume that p lies in sub-cell π_{NW}. Since π_{NW} is nonempty, by property (P4) in Lemma 2 the set S contains at least one point from π_{NW}. Let S' be the set of points of π_{NW} that are in S. Notice that $S' \subseteq S$ and $S' \neq \emptyset$. If any point of S' is visible from p, then this point is a desired vertex v with $|pv| \leqslant 1/2$ because the diameter of π_{NW} is $1/2$.

Assume that no point of S' is visible from p. The visibility of (points of) S' from p is blocked by some edges of $DT_1(S)$; these edges properly cross π_{NW} and separate p from points of S' (the red edges in Figure 6). Among these edges take one whose intersection points with the boundary of π_{NW} are visible from p (observe that such an edge always exists). Denote this edge by (a, b). Since the diameter of π_{NW} is $1/2$ and $|ab| \leqslant 1$, it is implied from Lemma 1 that the distance from p to a or to b is at most $1/\sqrt{2}$; after a suitable relabeling assume that $|pb| \leqslant 1/\sqrt{2}$. Of the two intersection points of (a, b) with the boundary of π_{NW}, denote by b' the one that is closer to b. By our choice of (a, b), b' is visible from p. We rotate the ray $\overrightarrow{pb'}$ towards b and stop as soon as hitting a vertex $v \in S$ in triangle $\triangle pbb'$ (it might be that $v = b$). The vertex v is visible from p. Since v is in triangle $\triangle pbb'$ it holds that $|pv| \leqslant \max\{|pb|, |pb'|\}$. Since $|pb| \leqslant 1/\sqrt{2}$ and $|pb'| \leqslant 1/2$ it turns out that $|pv| \leqslant 1/\sqrt{2}$. $\qquad\square$

3.3 Hop Stretch Factor

In this section we prove that the hop stretch factor of H is at most 341. We show that for any edge $(u, v) \in UDG(P)$ there exists a path of length at most 341 between u and v in H.

In this section a "cell" refers to the interior of a square of Γ, a "grid point" refers to the intersection point of a vertical and a horizontal grid line, and a "corner of π" refers to a grid point on the boundary of a cell π. We define *neighbors* of a cell π to be the set of eight cells that share sides or corners with π. We partition the neighbors of π into +-*neighbors* and ×-*neighbors*, where +-neighbors are the four cells that share sides with π, and ×-neighbors are the four cells each sharing exactly one grid point with π. In Figure 5 the cells $\pi_1, \pi_2, \pi_3, \pi_4$ are the +-neighbors of π, and the cells $\pi_9, \pi_{10}, \pi_{15}, \pi_{16}$ are the ×-neighbors of π.

Consider any two points $p, q \in S$. If $|pq| \leqslant 1$ then every edge of $DT(S)$, that lies in $D(p, q)$, has length at most 1, and thus all these edges are present in $DT_1(S)$. Combining this with Theorem 1 we get the following corollary.

Corollary 1. *For any two points $p, q \in S$, with $|pq| \leqslant 1$, there exists a path, between p and q in $DT_1(S)$, that lies in $D(p, q)$.*

(a) Configuration A

(b) Configuration B

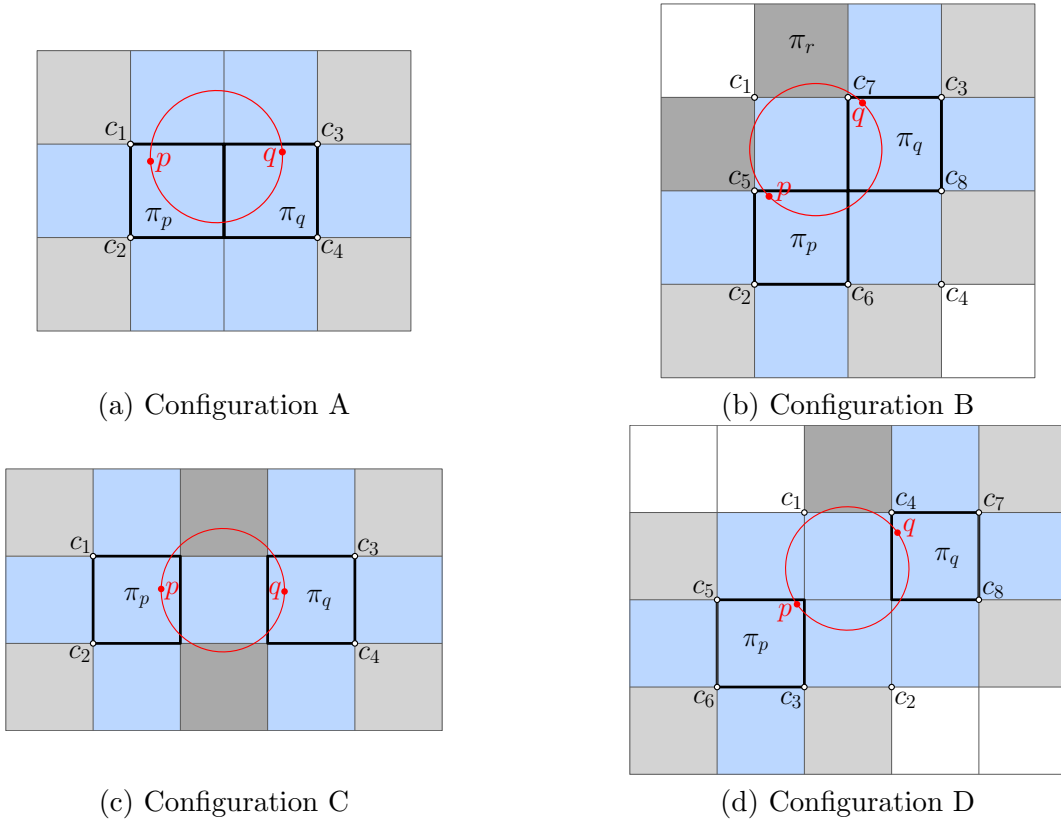(c) Configuration C

(d) Configuration D

Figure 7: Relative positions of the cells π_p and π_q where $|pq| \leqslant 1$.

Consider any two points p and q in the plane that lie in different cells, say π_p and π_q. If $|pq| \leqslant 1$ then the relative positions of π_p and π_q is among four configurations A, B, C, and D that are shown in Figure 7. In the rest of this section we consider different configurations of a disk intersecting some cells of Γ. Although mentioned before, we emphasis that a "cell" refers to the interior of a square of grid (and hence a cell is open and does not contain its boundary) while a "disk" is closed (and hence contains its boundary).

3.3.1 Disk-Cell Intersections

To cope with the number of cases that appear in the analysis of hop stretch factor we use lemmas 4, 5, and 6 about disk-cell intersections. These lemmas enable us to reduce the number of cases in our analysis. We say that an element x is "outside" a set X if $x \notin X$.

Lemma 4. *Let p and q be any two points in the plane with $|pq| \leqslant 1$.*

1. *If p and q are in different cells, then $D(p, q)$ intersects at most 7 cells.*

2. *If p and q are in the same cell π, then $D(p, q)$ can intersect only π and its four $+$-neighbors.*

Proof. Statement 1 is implied by the fact that $D(p, q)$ contains at most two grid points in its interior. To verify statement 2, it suffices to show that $D(p, q)$ does not contain any corner of π. Consider a corner c of π. Since p and q lie in π, the convex angle $\angle pcq$ is acute. Combing this with Thales's theorem implies that c is outside $D(p, q)$. $\square$

Lemma 5. *Let p and q be any two points in the plane that are in different cells π_p and π_q. Let X be the set containing the cells π_p and π_q and their $+$-neighbors.*

1. *If $|pq| \leqslant 1$, then $D(p, q)$ does not intersect any cell outside the neighborhoods of π_p and π_q.*

2. *If $|pq| \leqslant 1$, then $D(p, q)$ intersects at most two cells outside X.*

3. *If $|pq| \leqslant 1/\sqrt{2}$, then $D(p, q)$ does not intersect any cell outside X.*

Proof. We prove each statement separately.

Statement 1. Any cell π, that is outside the neighborhoods of π_p and π_q, has distance more $1/\sqrt{2}$ from each of π_p and π_q. Thus, for any point $r \in \pi$ we have $|rp| > 1/\sqrt{2}$ and $|rq| > 1/\sqrt{2}$. Since $|pq| \leqslant 1$, for any point x in $D(p, q)$ it holds that either $|xp| \leqslant 1/\sqrt{2}$ or $|xq| \leqslant 1/\sqrt{2}$. Therefore, r cannot be in $D(p, q)$. This implies that $D(p, q)$ does not intersect π.

Statement 2. The relative positions of π_p and π_q is among the four configurations in Figure 7. In this figure, the cells of X are colored blue, the $\times$-neighbors of π_p and π_q that are not in X and not intersected by $D(p, q)$ are colored light gray, and the $\times$-neighbors of π_p and π_q that are not in X but intersected by $D(p, q)$ are colored dark gray. We prove this statement for each configuration.

- Configuration A. By an application of Thales's theorem as in the proof of Lemma 4, one can verify that $D(p, q)$ does not contain any of grid points c_1, c_2, c_3, c_4, and hence does not intersect any cell outside X; see Figure 7(a).

- Configuration B. By Thales's theorem, $D(p, q)$ does not contain any of grid points c_1, c_2, c_3, c_4; see Figure 7(b). The mutual distances between grid points c_5, c_6, c_7, c_8 is at least 1, and thus $D(p, q)$ contains at most one of them. With these constraints, it turns out that $D(p, q)$ intersects at most two cells outside X.

- Configuration C. By Thales's theorem, $D(p, q)$ does not contain any of grid points c_1, c_2, c_3, c_4; see Figure 7(c). In this setting, $D(p, q)$ intersects at most two cells outside X.

- Configuration D. See Figure 7(d). Since $|c_1 q| > 1/\sqrt{2}$ and $|c_1 p| > 1/\sqrt{2}$, by an argument similar to the proof of statement 1, one can verify that $D(p, q)$ does not contain c_1; by symmetry it also does not contain c_2. Since the distance between q and each of c_3, c_5, c_6 is more than 1, $D(p, q)$ does not contain any of c_3, c_5, and c_6; by symmetry it also does not contain any of c_4, c_7, and c_8. With these constraints, it turns out that $D(p, q)$ intersects at most one cell outside X.

11

Statement 3. Since $|pq| \leqslant 1/\sqrt{2}$, the cells π_p and π_q are neighbors, and thus their relative positions is among configurations A and B in Figure 7. We have seen in the proof of statement 2 that in configuration A the disk $D(p,q)$ does not intersect any disk outside X. We prove our claim for configuration B. Every cell π_r outside X is at a distance more than $1/\sqrt{2}$ from π_p or π_q; see Figure 7(b). Since the diameter of $D(p,q)$ is at most $1/\sqrt{2}$, this implies that π_r lies outside $D(p,q)$. □

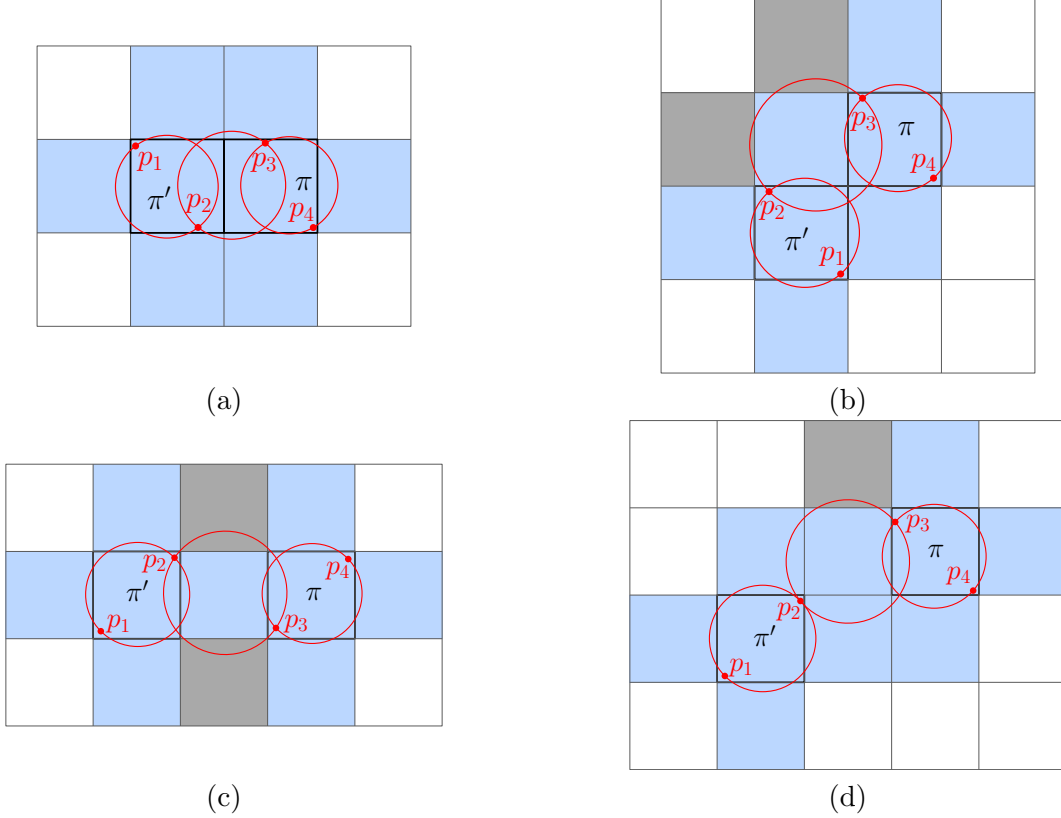(a)　　　　　　　　　　　(b)

(c)　　　　　　　　　　　(d)

Figure 8: Illustration of the cells that are intersected by $D(p_1, p_2) \cup D(p_2, p_3) \cup D(p_3, p_4)$.

Lemma 6. *Consider two cells π and π'. Let p_1 and p_2 be any two points in π', and let p_3 and p_4 be any two points in π. Let $\mathcal{D}$ be the union of three disks $D(p_1, p_2)$, $D(p_2, p_3)$, and $D(p_3, p_4)$. Then the following statements hold:*

1. *If $|p_2 p_3| \leqslant 1/\sqrt{2}$ then $\mathcal{D}$ intersects at most 8 cells.*

2. *If $|p_2 p_3| \leqslant 1$, and π and π' are +-neighbors, then $\mathcal{D}$ intersects at most 8 cells.*

3. *If $|p_2 p_3| \leqslant 1$, and π and π' are $\times$-neighbors, then $\mathcal{D}$ intersects at most 10 cells.*

4. *If $|p_2 p_3| \leqslant 1$, and π and π' are not neighbors, then $\mathcal{D}$ intersects at most 11 cells.*

Proof. We define X, as in Lemma 5, to be the set containing the cells π and π' and their +-neighbors (where π and π' play the roles of π_p and π_q). By Lemma 4 the disk $D(p_1, p_2)$ can intersect only π' and its four +-neighbors, and the disk $D(p_3, p_4)$ can intersect only π and its four +-neighbors. Thus $D(p_1, p_2)$ and $D(p_3, p_4)$ can intersect only cells in X. Now we verify each statement.

- Statement 1. Since $|p_2p_3| \leqslant 1/\sqrt{2}$, the cells π and π' are neighbors, and thus their relative positions is among configurations A and B in Figures 7(a) and 7(b). In each of these configurations the set X contains 8 cells. By Lemma 5, $D(p_2, p_3)$ does not intersect any cell outside X. Therefore, the union of the three disks, i.e. $\mathcal{D}$, intersects at most 8 cells.

- Statement 2. Since π and π' are +-neighbors, their relative position is configuration A. In this configuration, X contains 8 cells, and — as we have seen in the proof of statement 2 of Lemma 5 — the disk $D(p_2, p_3)$ does not intersect any cell outside X. Therefore, $\mathcal{D}$ intersects at most 8 cells; see Figure 8(a).

- Statement 3. The relative position of π and π' is configuration B. In this configuration, X contains 8 cells, and — as we have seen in the proof of statement 2 of Lemma 5 — the disk $D(p_2, p_3)$ intersects at most two cells outside X. Therefore, $\mathcal{D}$ intersects at most 10 cells (8 cells in X and 2 cells outside X); see Figure 8(b).

- Statement 4. The relative positions of π and π' is among configurations C and D. In configuration C, X contains 9 cells, and by the proof of Lemma 5 the disk $D(p_2, p_3)$ intersects at most two cells outside X. Therefore, $\mathcal{D}$ intersects at most 11 cells; see Figure 8(c). In configuration D, X contains 10 cells, and by the proof of Lemma 5 (statement 2, configuration D) the disk $D(p_2, p_3)$ intersects at most one cell outside X. Therefore, $\mathcal{D}$ intersects at most 11 cells; see Figure 8(d). $\square$

3.3.2 Analysis of Hop Stretch Factor

With lemmas in the previous section, we have all tools for proving the hop stretch factor of H. Recall that no point of P lies on a grid line of Γ, and thus every point of P is in the interior of some square of Γ. Consider any edge $(u, v) \in UDG(P)$, and notice that $|uv| \leqslant 1$. In this section we prove the existence of a path, of length at most 341, between u and v in H. Depending on whether u or v belong to S, we have three cases: (1) $u \notin S$ and $v \notin S$, (2) $u \in S$ and $v \in S$, and (3) $u \notin S$ and $v \in S$, or vice versa. These cases are treated using similar arguments. We give a detailed description of case (1) which gives rise to the worst stretch factor for our algorithm. We give a brief description of other cases at the end of this section. We denote by "(p, q)-path" a simple path between two points p and q.

Case (1): In this case $u, v \in P \setminus S$. Recall that, in H, u and v are connected to their closest visible vertices of $DT_1(S)$; let u_1 and v_1 denote these vertices respectively. Therefore, in H, there is a (u, v)-path that consists of the edge (u, u_1), a (u_1, v_1)-path in $DT_1(S)$, and the edge (v_1, v); see Figure 9-top. In the following description we prove the existence of a (u_1, v_1)-path in $DT_1(S)$ of desired length. By Lemma 3 we have $|uu_1| \leqslant 1/\sqrt{2}$ and $|vv_1| \leqslant 1/\sqrt{2}$; we will use these inequalities in our description.

Let π_u, π_v, π'_u and π'_v denote the cells containing u, v, u_1 and v_1 respectively. Depending on the identicality of these cells we can have — up to symmetry — the following five sub-cases: (i) $\pi_u = \pi_v$ or (ii) $\pi'_u = \pi_u$ or (iii) $\pi'_u = \pi_v$ or (iv) $\pi'_u = \pi'_v$ or (v) all four cells are pairwise distinct. These sub-cases are treated using similar arguments. We give a detailed description of sub-case (v) which gives rise to the worst stretch factor for our algorithm. We give a brief description of other sub-cases at the end of this section..

Assume that π_u, π_v, π'_u and π'_v are pairwise distinct. Since $|uu_1| \leqslant 1/\sqrt{2}$, π_u and π'_u are neighbors; similarly π_v and π'_v are neighbors. Since $(u, u_1) \in UDG(P)$, by properties (P2) and (P3) in Lemma 2 there exist two points $u_2, u_3 \in S$ such that $u_2 \in \pi'_u$, $u_3 \in \pi_u$, and $|u_2u_3| \leqslant 1$. Similarly, there exist two points $v_2, v_3 \in S$ such that $v_2 \in \pi'_v$, $v_3 \in \pi_v$, and $|v_2v_3| \leqslant 1$. Moreover,

since $(u, v) \in UDG(P)$, there exist two points $u_4, v_4 \in S$ such that $u_4 \in \pi_u$, $v_4 \in \pi_v$, and $|u_4 v_4| \leqslant 1$. See Figure 9. It might be the case that $u_1 = u_2$, $u_3 = u_4$, $v_3 = v_4$, or $v_1 = v_2$. Since u_1 and u_2 are in the same cell, $|u_1 u_2| \leqslant 1$; similarly $|u_3 u_4| \leqslant 1$, $|v_1 v_2| \leqslant 1$, and $|v_3 v_4| \leqslant 1$. Having these distance constraints, Corollary 1 implies that in $DT_1(S)$ there exists a walk between u_1 and v_1 that consists of a (u_1, u_2)-path in $D(u_1, u_2)$, a (u_2, u_3)-path in $D(u_2, u_3)$, a (u_3, u_4)-path in $D(u_3, u_4)$, a (u_4, v_4)-path in $D(u_4, v_4)$, a (v_4, v_3)-path in $D(v_4, v_3)$, a (v_3, v_2)-path in $D(v_3, v_2)$, and a (v_2, v_1)-path in $D(v_2, v_1)$. Thus, there is a (u_1, v_1)-path in $DT_1(S)$ that lies in the union of these seven disks; see Figure 9.

Let $\mathcal{D}$ denote the union of the seven disks. We want to obtain an upper bound on the number of cells intersected by $\mathcal{D}$. To that end, set $\mathcal{D}_u = D(u_1, u_2) \cup D(u_2, u_3) \cup D(u_3, u_4)$, and $\mathcal{D}_v = D(v_4, v_3) \cup D(v_3, v_2) \cup D(v_2, v_1)$. Define X_u as the set containing the cells π_u and π'_u and their +-neighbors. Since π_u and π'_u are neighbors, their relative positions is among configurations A and B (Figures 7(a) and 7(b)); in these configurations X_u contains 8 cells. Analogously, define X_v with respect to π_v and π'_v, and notice that X_v also contains 8 cells.

Claim. Each of $\mathcal{D}_u$ and $\mathcal{D}_v$ intersects at most 8 cells. Moreover, the cells that are intersected by $\mathcal{D}_u$ and $\mathcal{D}_v$ belong to X_u and X_v, respectively.

Proof. Because of symmetry, we prove this claim only for $\mathcal{D}_u$. Recall that π_u and π'_u are neighbors. If π_u and π'_u are +-neighbors, then $\mathcal{D}_u$ intersects at most 8 cells by statement 2 in Lemma 6. The proof of statement 2 also implies that these (at most 8) cells belong to X_u. If π_u and π'_u are ×-neighbors, then by property (P3) in Lemma 2, (u_2, u_3) is the shortest edge of $UDG(P)$ that runs between π'_u and π_u. Since (u_1, u) is also an edge between π'_u and π_u, we have $|u_2 u_3| \leqslant |u_1 u| \leqslant 1/\sqrt{2}$. In this case $\mathcal{D}_u$ intersects at most 8 cells by statement 1 in Lemma 6. The proof of statement 1 implies that these cells belong to X_u. $\square$

Notice that $\mathcal{D} = \mathcal{D}_u \cup \mathcal{D}_v \cup D(u_4, v_4)$. Based on this and the above claim, in order to obtain an upper bound on the number of cells that are intersected by $\mathcal{D}$ it suffices to obtain an upper bound on the number of cells, outside $X_u \cup X_v$, that are intersected by $D(u_4, v_4)$. To that end, define X as the set containing the cells π_u and π_v and their +-neighbors, and notice that $X \subseteq X_u \cup X_v$. By statement 2 of Lemma 5, the disk $D(u_4, v_4)$ intersects at most 2 cells outside X, and hence at most 2 cells outside $X_u \cup X_v$. Therefore, the number of cells intersected by $\mathcal{D}$ is at most $|X_u \cup X_v| + 2 \leqslant 8 + 8 + 2 = 18$. Since by property (P1) in Lemma 2 each cell contains at most 20 points of S, the set $\mathcal{D}$ contains at most 360 points of S. Therefore, the (u_1, v_1)-path in $DT_1(S)$ has at most 360 vertices, and hence at most 359 edges. Thus, the (u, v)-path in H has at most 361 edges (including (u, u_1) and (v_1, v)).

With a closer look at relative positions of π_u and π_v we show that $\mathcal{D}$ in fact intersects at most 17 cells. This would imply that the (u, v)-path has at most 341 edges as claimed. To that end we consider four configurations A, B, C, and D for π_u and π_v, which we refer to them as sub-cases (v)-A, (v)-B, (v)-C, and (v)-D, respectively.

- (v)-A. In this case X_u and X_v share π_u and π_v and thus $|X_u \cup X_v| \leqslant 14$. Moreover, by the proof of Lemma 5 the disk $D(u_4, v_4)$ does not intersect any cell outside $X_u \cup X_v$; see Figure 7(a). Thus $\mathcal{D}$ intersects at most 14 cells.

- (v)-B. In this case X_u and X_v share at least two cells (two +-neighbors of π_u and π_v) and thus $|X_u \cup X_v| \leqslant 14$. Moreover, by the proof of Lemma 5 the disk $D(u_4, v_4)$ intersects at most two cells outside $X_u \cup X_v$; see Figure 7(b). Thus $\mathcal{D}$ intersects at most 16 cells (the shaded cells in Figure 9-top).

14

- (v)-C. In this case X_u and X_v share at least one cell (one +-neighbor of π_u and π_v), and by the proof of Lemma 5 the disk $D(u_4, v_4)$ intersects at most two cells outside $X_u \cup X_v$; see Figure 7(c). Thus $\mathcal{D}$ intersects at most 17 cells (the shaded cells in Figure 9-middle).

- (v)-D. In this case X_u and X_v may not share any cell, but by the proof of Lemma 5 the disk $D(u_4, v_4)$ intersects at most one cell outside $X_u \cup X_v$; see Figure 7(d). Thus $\mathcal{D}$ intersects at most 17 cells (the shaded cells in Figure 9-bottom).

Even though $\mathcal{D}$ may intersect exactly 17 cells, at the end of this section we show a possibility of decreasing the upper bound on the length of the (u, v)-path even further. This will require more case analysis which we avoid.

Other Cases and Sub-Cases: We gave a detailed analysis for sub-case (v) (of case (1)) where u, v, u_1, v_1 lie in distinct cells π_u, π_v, π'_u, π'_v. Our analysis shows the existence of a (u_1, u_4)-path in $\mathcal{D}_u$ which intersects at most 8 cells (which belong to X_u), and the existence of a (v_1, v_4)-path in $\mathcal{D}_v$ which intersects at most 8 cells (which belong to X_v). In the sequel we give short descriptions of case (2), case (3), and remaining sub-cases of case (1).

Recall case (1) where $u, v \in P \setminus S$. In sub-case (i) where $\pi_u = \pi_v$, the sets X_u and X_v share at least 5 cells (π_u and its four +-neighbors). Therefore, $\mathcal{D}$ intersects at most $|X_u \cup X_v| + 2 \leqslant 8 + 8 - 5 + 2 = 13$ cells. Similarly, in each of sub-cases (iii) where $\pi'_u = \pi_v$ and (iv) where $\pi'_u = \pi'_v$, the sets X_u and X_v share at least 5 cells, and thus $\mathcal{D}$ intersects at most 13 cells. In sub-case (ii) where $\pi'_u = \pi_u$, the set X_u contains 5 cells (π_u and its four +-neighbors), and thus $\mathcal{D}$ intersects at most $5 + 8 + 2 = 15$ cells. Thus, in all these remaining sub-cases, the (u, v)-path has at most 301 edges (including (u, u_1) and (v_1, v)).

Now consider case (3) where $v \in S$ or $u \in S$ but not both. By symmetry we assume that $v \in S$, and thus $u \in P \setminus S$. In this case we do not have the point v_1 nor the cell π'_v; one may assume that $v_4 = v_3 = v_2$. Thus, X_v contains at most 5 cells (π_v and its +-neighbors). By an argument similar to that of case (1), there exists a (u_1, v)-path in $DT_1(S)$ that lies in $\mathcal{D}$ which intersects at most $|X_u \cup X_v| + 2 \leqslant 8 + 5 + 2 = 15$ cells. Therefore, there is a (u, v)-path in H that has at most 300 edges (including the edge (u, u_1)).

Consider case (2) where $u, v \in S$. Since $|uv| \leqslant 1$, by Corollary 1 there is a (u, v)-path in $DT_1(S)$ that lies in $D(u, v)$. By Lemma 4, $D(u, v)$ intersects at most seven cells, and thus contains at most 140 points of S. Therefore, the (u, v)-path has at most 139 edges.

Further Improvement of Hop Stretch Factor: Recall the set $\mathcal{D}$, from Section 3.3.2, which intersects at most 17 cells, and hence contains at most 340 points of S; this implied that the length of the (u, v)-path in H is at most 341. In this section we show a possibility of how one could improve the upper bound on the number of points in $\mathcal{D}$ to 319; this would decrease the upper bound on the length of the (u, v)-path to 320. However, to show this, one requires to go through some case analysis, which we avoid in this paper.

Recall that in sub-cases (v)-B, (v)-C, and (v)-D the set $\mathcal{D}$ intersects at most 16, 17, and 17 cells respectively; see Figure 9 for an illustration of these sub-cases. In all other cases and sub-cases, $\mathcal{D}$ intersects at most 15 cells, and hence contains at most 300 points of S. Thus, it suffices to show, only for sub-cases (v)-B, (v)-C, (v)-D, that $\mathcal{D}$ contains at most 319 points of S. This involves some case analysis which we provide an overview of that.

For each cell π, let π_L and π_R be left and right rectangles obtained by bisecting π with a vertical line, i.e., $\pi_L = \pi_{NW} \cup \pi_{SW}$ and $\pi_R = \pi_{NE} \cup \pi_{SE}$; see Figure 5. Recall points s_i in Lemma 2, i.e., the points of S that lie in π. All points s_8, s_{12}, s_{14} lie in π_L, and all points s_{11}, s_{13}, s_{17} lie in π_R. Consider a similar partitioning of π into top and bottom rectangles π_T and π_B, and observe that
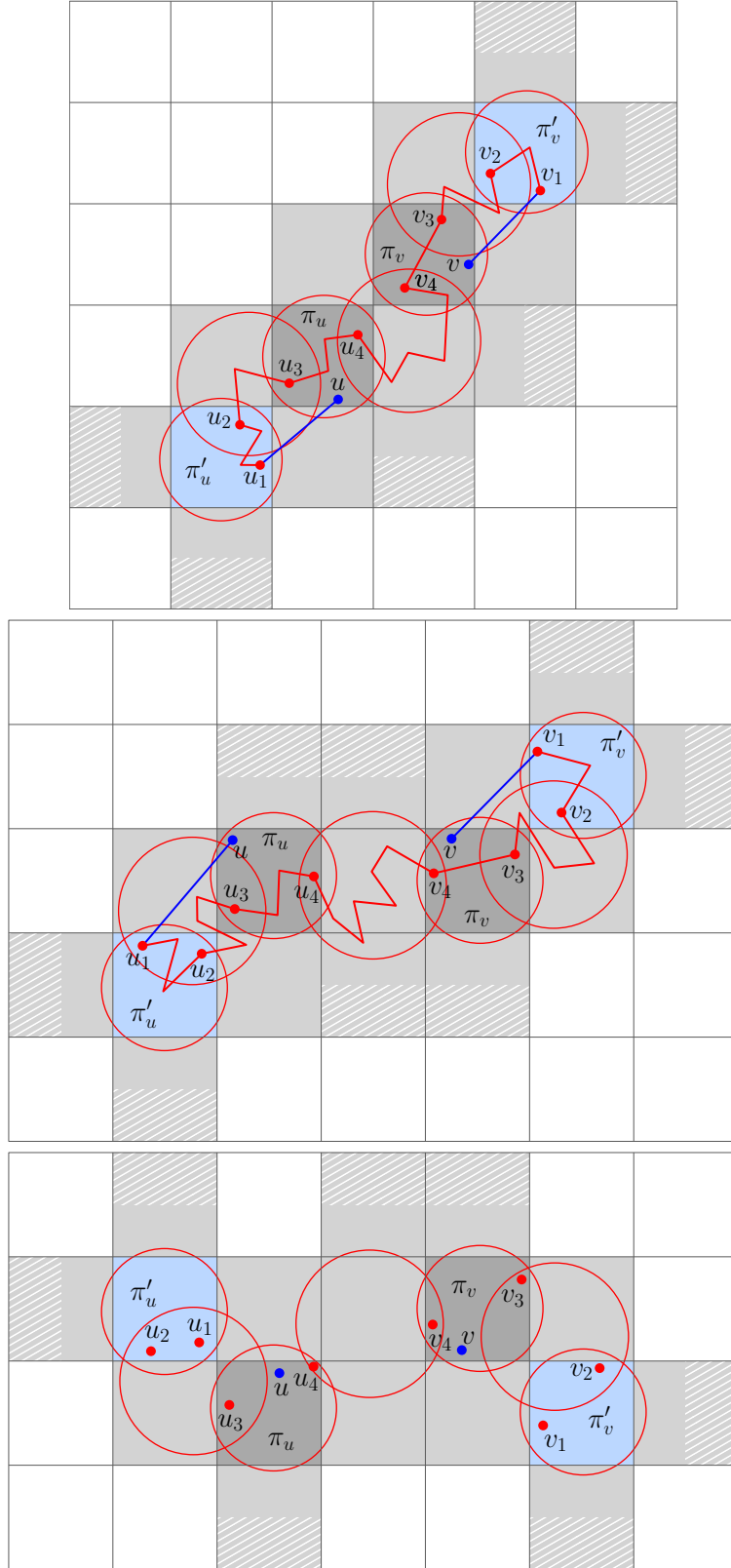
Figure 9: Points u, v, u_1, v_1 belong to distinct cells. The cells π_u, π_v are in configurations B (top), C (middle), and D (bottom). The red paths corresponds to (u_1, v_1)-paths in $DT_1(S)$.

all points s_5, s_6, s_7 lie in π_T and all points s_{18}, s_{19}, s_{20} lie in π_B. We refer to π_L, π_R, π_T, π_B by "rectangles". The tiled regions in Figure 9 correspond to these rectangles (from different cells). These rectangles (tiled regions in Figure 9) are not intersected by $\mathcal{D}$. This and the fact that each rectangle counts for three points, imply that for each rectangle we could subtract 3 from the number of points in $\mathcal{D}$. In Figures 9-top, 9-middle, 9-bottom, which correspond to sub-cases (v)-B, (v)-C, (v)-D, the number of these rectangles is 6, 8, and 7, respectively.

Consider any of sub-cases (v)-B, (v)-C and (v)-D. Fix the relative position of π_u and π_v. By checking all possible configurations of neighboring cells π'_u and π'_v around π_u and π_v, we always get either (i) at most 15 cells that are intersected by $\mathcal{D}$, or (ii) exactly 16 cells that are intersected by $\mathcal{D}$ and at least 1 rectangle that is not intersected by $\mathcal{D}$, or (iii) exactly 17 cells that are intersected by $\mathcal{D}$ and at least 7 rectangle that are not intersected by $\mathcal{D}$. In case (i), the set $\mathcal{D}$ contains at most 300 points of S. Since each rectangle counts for three points of S, in cases (ii) and (iii), the set $\mathcal{D}$ contains at most $320 - 3 = 317$ and $340 - 7 * 3 = 319$ points of S, respectively.

References

[1] H. A. Akitaya, J. Castello, Y. Lahoda, A. Rounds, and C. D. Tóth. Augmenting planar straight line graphs to 2-edge-connectivity. In *Proceedings of the 23rd Symposium on Graph Drawing and Network Visualization (GD)*, pages 563–564, 2015.

[2] K. M. Alzoubi, X. Li, Y. Wang, P. Wan, and O. Frieder. Geometric spanners for wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 14(4):408–421, 2003.

[3] P. Bose, R. Fagerberg, A. van Renssen, and S. Verdonschot. On plane constrained bounded-degree spanners. In *Proceedings of the 10th Latin American Symposium on Theoretical Informatics (LATIN)*, pages 85–96, 2012.

[4] P. Bose and M. Smid. On plane geometric spanners: A survey and open problems. *Comput. Geom.*, 46(7):818–830, 2013.

[5] N. Catusse, V. Chepoi, and Y. Vaxès. Planar hop spanners for unit disk graphs. In *Proceedings of the 6th Workshop on Algorithms for Sensor Systems, Wireless Ad Hoc Networks, and Autonomous Mobile Entities (ALGOSENSORS)*, pages 16–30, 2010.

[6] J. Chen, A. Jiang, I. A. Kanj, G. Xia, and F. Zhang. Separability and topology control of quasi unit disk graphs. *Wireless Networks*, 17(1):53–67, 2011. Also in *INFOCOM* 2007.

[7] D. P. Dobkin, S. J. Friedman, and K. J. Supowit. Delaunay graphs are almost as good as complete graphs. *Discrete & Computational Geometry*, 5:399–407, 1990. Also in *FOCS* 1987.

[8] D. Eppstein. Spanning trees and spanners. Technical report, Information and Computer Science, University of California, Irvine, 1996.

[9] M. Fürer and S. P. Kasiviswanathan. Spanners for geometric intersection graphs with applications. *Journal of Computational Geometry*, 3(1):31–64, 2012.

[10] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Discrete mobile centers. *Discrete & Computational Geometry*, 30(1):45–63, 2003. Also in *SoCG* 2001.

[11] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric spanners for routing in mobile networks. *IEEE Journal on Selected Areas in Communications*, 23(1):174–185, 2005. Also in *MobiHoc* 2001.

[12] F. Hurtado, M. Kano, D. Rappaport, and C. D. Tóth. Encompassing colored planar straight line graphs. *Computational Geometry: Theory and Applications*, 39(1):14–23, 2008. Also in *CCCG* 2004.

[13] I. A. Kanj and L. Perkovic. On geometric spanners of Euclidean and unit disk graphs. In *Proceedings of the 25th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 409–420, 2008.

[14] X. Li. Algorithmic, geometric and graphs issues in wireless networks. *Wireless Communications and Mobile Computing*, 3(2):119–140, 2003.

[15] X. Li, G. Călinescu, and P. Wan. Distributed construction of planar spanner and routing for ad hoc wireless networks. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1268–1277, 2002.

[16] X. Li and Y. Wang. Efficient construction of low weighted bounded degree planar spanner. *International Journal of Computational Geometry & Applications*, 14(1-2):69–84, 2004. Also in *COCOON* 2003.

[17] G. Narasimhan and M. Smid. *Geometric spanner networks*. Cambridge University Press, 2007.

[18] R. Rajaraman. Topology control and routing in ad hoc networks: a survey. *SIGACT News*, 33(2):60–73, 2002.

[19] C. Yan, Y. Xiang, and F. F. Dragan. Compact and low delay routing labeling scheme for Unit Disk Graphs. *Computational Geometry: Theory and Applications*, 45(7):305–325, 2012. Also in *WADS* 2009.