

Linear Size Planar Manhattan Network for Convex Point Sets

Satyabrata Jana^{a,*}, Anil Maheshwari^b, Sasanka Roy^a

^a*Indian Statistical Institute, Kolkata, India*

^b*School of Computer Science, Carleton University, Ottawa, Canada*

Abstract

Let $G = (V, E)$ be an edge weighted *geometric graph* such that every edge is horizontal or vertical. The weight of an edge $uv \in E$ is its length. Let $W_G(u, v)$ denote the length of a shortest path between a pair of vertices u and v in G . The graph G is said to be a **Manhattan network** for a given point set P in the plane if $P \subseteq V$ and $\forall p, q \in P, W_G(p, q) = \|pq\|_1$. In addition to P , graph G may also include a set T of *Steiner points* in its vertex set V . In the **Manhattan network problem**, the objective is to construct a **Manhattan network** of small size for a set of n points. This problem was first considered by Gudmundsson et al.[1]. They give a construction of a **Manhattan network** of size $\Theta(n \log n)$ for general point set in the plane. We say a **Manhattan network** is planar if it can be embedded in the plane without any edge crossings. In this paper, we construct a linear size **planar Manhattan network** for convex point set in linear time using $\mathcal{O}(n)$ Steiner points. We also show that, even for convex point set, the construction in Gudmundsson et al. [1] needs $\Omega(n \log n)$ Steiner points and the network may not be planar.

Keywords: Convex point set, L_1 norm, Manhattan Network, Histogram, Planar Graph, Steiner points, Plane Graph

*Corresponding author

Email addresses: satyamtma@gmail.com (Satyabrata Jana), anil@scs.carleton.ca (Anil Maheshwari), sasanka.ro@gmail.com (Sasanka Roy)

1. Introduction

In computational geometry, constructing a minimum length Manhattan network is a well-studied area [2]. A graph $G = (V, E)$ is said to be a Manhattan network for a given point set P in the plane if $P \subseteq V$ and $\forall p, q \in P, W_G(p, q) = \|pq\|_1$, where $W_G(u, v)$ denotes the length of a shortest path between a pair of vertices u and v in G . The graph G may also include a set T of Steiner points in its vertex set V . A Minimum Manhattan network (MMN) problem on P is to construct a Manhattan network of minimum possible length. Below in Figure 1(a) and Figure 1(b), we show examples of a Manhattan network and a Minimum Manhattan network on the same set of points.

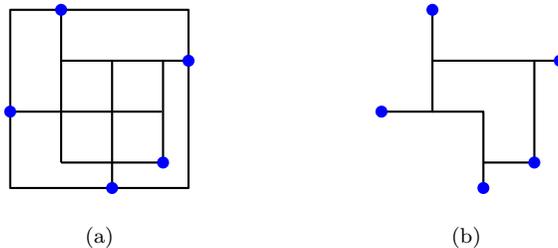


Figure 1: (a) A Manhattan network, and (b) A minimum Manhattan network.

MMN problem has a wide number of applications in city planning, network layouts, distributed algorithms [3], VLSI circuit design [2], and computational biology [4]. The MMN problem was first introduced in 1999 by Gudmundsson et al. [2]. Several approximation algorithms (with factors 4 [1], 2 [5], and 1.5 [6]) with time complexity $\mathcal{O}(n^3)$ have been proposed in the last few years. Also, there are $\mathcal{O}(n \log n)$ time approximation algorithms with factors 8 [1], 3 [7], and 2 [8]. Recently Chin et al. [9] proved that the decision version of the MMN problem is strongly NP-complete. Recently, Knauer et al. [10] showed that this problem is fixed parameter tractable.

In 2007, Gudmundsson et al. [1] considered a variant of the MMN problem where the goal is to minimize the number of vertices (Steiner) and edges. In $\mathcal{O}(n \log n)$ time, they construct a Manhattan network with $\mathcal{O}(n \log n)$ ver-

tices and edges using divide and conquer strategy. They also proved that there are point sets in \mathbb{R}^2 where every Manhattan network on these points will need $\Omega(n \log n)$ vertices and edges.

A set of points is said to be a convex point set if all of the points are vertices of their convex hull. A *plane Manhattan network* is a Manhattan network without non-crossing edges. Gudmundsson et al. [1] showed that there exists a convex point set for which a *plane Manhattan network* requires $\Omega(n^2)$ vertices and edges. Now we explain the construction of the plane Manhattan network given by Gudmundsson et al. [1]. To keep it simple, we would use the same notations as they use. Let P be a set of points in the plane as follows:

$$P = \bigcup_{i=1}^{n-1} \left\{ \left(\frac{i}{n}, 0 \right), \left(\frac{i}{n}, 1 \right), \left(0, \frac{i}{n} \right), \left(1, \frac{i}{n} \right) \right\}$$

If G is a plane Manhattan network of P then there must be a shortest L_1 path between every pair of points $(\frac{i}{n}, 0), (\frac{i}{n}, 1)$ and $(0, \frac{i}{n}), (1, \frac{i}{n})$. These paths need to be orthogonal straight line segments because in the first case the x -coordinates are the same and in the second case the y -coordinates are the same. This would force us to add Steiner points at all the $\Theta(n^2)$ intersection points. For an illustration, see Figure 2(a).

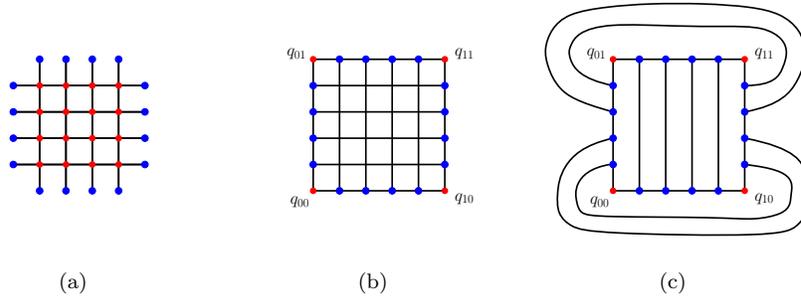


Figure 2: (a) Lower bound construction of plane Manhattan network of P (b) Planar Manhattan network G^* of P and (c) Planar embedding of G^* . Blue circles represent the points in P and red circles represent Steiner points.

A natural question that arises is what if we want the network to be planar

(and not necessarily plane). We say a Manhattan network is planar if it can be embedded in the plane without any edge crossings. For the above example, we can construct a **planar Manhattan network** $G = (V = P \cup T, E)$ of $\mathcal{O}(n)$ size as follows: Note that, P lies on the boundary of a square $Q = [(0, 0), (0, 1)] \times [(1, 0), (1, 1)]$ (see Figure 2(b)). We add four Steiner points $q_{00} = (0, 0)$, $q_{01} = (0, 1)$, $q_{10} = (1, 0)$, $q_{11} = (1, 1)$, and we define $T = \{q_{00}, q_{01}, q_{10}, q_{11}\}$. For $i = 1, 2, \dots, n - 1$, we add the edges between every pair of consecutive points (including these four Steiner points) on the boundary of Q . We also add the edges between every pair of points $(\frac{i}{n}, 0), (\frac{i}{n}, 1)$ and $(0, \frac{i}{n}), (1, \frac{i}{n})$. To show that G is a Manhattan network, we prove that $\forall p, q \in P$, $W_G(p, q) = \|pq\|_1$. Following is the description of all these paths in G . The paths between every pair of points $(\frac{i}{n}, 0), (\frac{j}{n}, 1)$ and $(0, \frac{i}{n}), (1, \frac{j}{n})$ is a straight line segment (horizontal and vertical). The paths between every pair of points $(\frac{i}{n}, 0), (0, \frac{j}{n})$ go through q_{00} . Likewise, the paths between every pair of points $(0, \frac{i}{n}), (\frac{j}{n}, 1)$ go through q_{01} , the paths between every pair of points $(0, \frac{i}{n}), (\frac{j}{n}, 1)$ go through q_{10} , the paths between every pair of points $(\frac{i}{n}, 1), (1, \frac{j}{n})$ go through q_{11} . Between every pair of points $(\frac{i}{n}, 0), (\frac{j}{n}, 1)$ there exists a path through $(\frac{i}{n}, 1)$. Similarly, between every pair of points $(0, \frac{i}{n}), (1, \frac{j}{n})$ there exists a path through $(1, \frac{i}{n})$. To show that G is planar, we provide its planar embedding. For the planar embedding of G , we keep the edges between every pair of points $(\frac{i}{n}, 0)$ and $(\frac{j}{n}, 1)$ inside the interior face of Q and draw the edges between $(0, \frac{i}{n})$ and $(1, \frac{j}{n})$ in the exterior face of Q . For an illustration, see Figure 2(c).

A closely related problem is to construct geometric spanner from a given point set. For a real number $t \geq 1$, a geometric graph $G = (S, E)$ is a t -spanner of S if for any two points p and q in S , $W_G(p, q) \leq t|pq|$. The *stretch factor* of G is the smallest real number t such that G is a t -spanner of S . A large number of algorithms have been proposed for constructing t -spanners for any given point set [3]. Keil et al. [11] showed that the Delaunay triangulation of S is a 2.42-spanner of S . For convex point sets, Cui et al. [12] proved that the Delaunay triangulation has a stretch factor of at most 2.33. Xia [13] provides a 1.998-spanner for general point sets. Steiner points have also been

used for constructing spanners. For example, Arikati et al. [14] use Steiner points to answer exact shortest path queries between any two vertices of a geometric graph. Authors [14] consider the problem of finding an obstacle-avoiding L_1 path between a pair of query points in the plane. They find a $(1 + \epsilon)$ spanner with space complexity $\mathcal{O}(n^2/\sqrt{r})$, preprocessing time $\mathcal{O}(n^2/\sqrt{r})$ and $\mathcal{O}(\log n + \sqrt{r})$ query time, where ϵ is an arbitrarily small positive constant and r is an arbitrary integer, such that $1 < r < n$. Recently, Amani et al. [15] show how to compute a plane 1.88-spanner in L_2 norm for convex point sets in $\mathcal{O}(n)$ time without using Steiner points. For a general point set of size n , Gudmundsson et al. [1] construct a $\sqrt{2}$ -spanner (may not be planar) in L_2 norm and its size is $\mathcal{O}(n \log n)$. But as a corollary of our construction in this paper, for a convex point set, we obtain a planar $\sqrt{2}$ spanner in L_2 norm using $\mathcal{O}(n)$ Steiner points. The MMN problem for a point set is same as the problem of finding a 1-spanner in L_1 -metric [9]. Given a rectilinear polygon with n vertices, in linear time, Schuierer [16] constructs a data structure that can report the shortest path (in L_1 -metric) for any pair of query points in that polygon in $\mathcal{O}(\log + k)$ time where k is the number of segments in the shortest path. De Berg [17] shows that given two arbitrary points inside a polygon, the L_1 -distance between them can be reported in $\mathcal{O}(\log n)$ time. In this paper, we consider the following problem.

Manhattan network problem

Input: A set S of n points in convex position.

Goal: To construct a linear size planar Manhattan network.

1.1. Our Contributions

- In linear time, we construct a planar Manhattan network G for a convex point set S of size n . G uses $\mathcal{O}(n)$ Steiner points as vertices.
- We show that the construction in Gudmundsson et al. [1] needs $\Omega(n \log n)$ points even for a convex point set and may not result in a planar graph.

1.2. Organization

In Section 2, we sketch the $\mathcal{O}(n \log n)$ construction of Gudmundsson et al. [1]. We prove that, even for convex point set, their construction needs $\Omega(n \log n)$ points. We also show that their construction is not planar by considering a convex point set of 16 points for which their Manhattan network has a minor homeomorphic to $K_{3,3}$. In Section 3, we provide our construction of $\mathcal{O}(n)$ size planar Manhattan network G for a convex point set S .

2. Manhattan Network for General Point Sets

For general point sets, Gudmundsson et al. [1] proved the following theorem.

Theorem 1. [1] *Let P be a set of n points. A Manhattan network of P consisting of $\Theta(n \log n)$ vertices and edges can be computed in $\mathcal{O}(n \log n)$ time.*

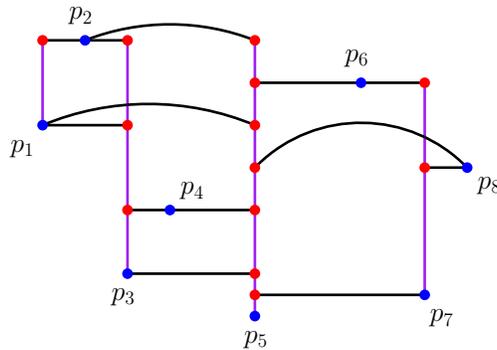


Figure 3: Construction of the *Manhattan network* for S . Points in S are in blue color and Steiner points are in red color.

Their construction is as follows: Sort the points in P according to their x -coordinate. Let m be the median x -coordinate in P . Then draw a vertical line L_m through $(m, 0)$. For each point p of S , take an orthogonal projection on the line L_m . Add Steiner points at each projection and join p with its corresponding projection point. Then recursively do the same, on the $\frac{n}{2}$ points that have less x -coordinate than p and $\frac{n}{2}$ points that have greater x -coordinate than p . Add a Steiner point at each projection. Figure 3 illustrates the algorithm of Gudmundsson et al. [1].

Now we show that even for convex point set, this construction will need $\Omega(n \log n)$ Steiner points. In Figure 4, for a set of sixteen points in convex position, we show that their network is not planar as it has a minor homeomorphic to $K_{3,3}$ and the network uses 38 Steiner points.

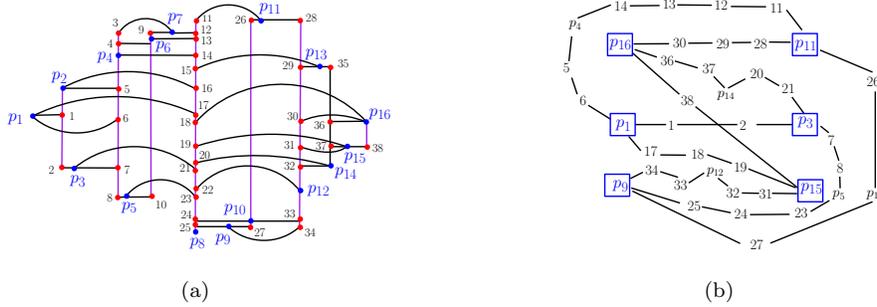


Figure 4: (a) Manhattan network G_A of a convex point set $A = \{p_1, p_2, \dots, p_{16}\}$ (blue color). Points colored in red are Steiner points, and (b) G'_A , subgraph of G_A , that is homeomorphic to $K_{3,3}$.

3. Planar Manhattan Network for a Convex Point Set

In this section, we construct a linear size planar Manhattan network G for a convex point set S . G uses $\mathcal{O}(n)$ Steiner points and can be constructed in linear time. We organize this section as follows: After introducing some definitions and notations in Section 3.1, we construct a histogram partition $\mathcal{H}(\mathcal{OCP}(S))$ of an ortho-convex polygon $\mathcal{OCP}(S)$ of the convex point set S in Section 3.2.

In Section 3.3 we construct our desired graph $G = (V, E)$ where $S \subseteq V$. In Section 3.4 we prove that G is a **Manhattan network** for S . In Section 3.5 we show that G is planar. In Section 4 we draw conclusions and state some interesting open problems.

3.1. Preliminaries

A *polygonal chain*, with n vertices in the plane, is defined as an ordered set of vertices (v_1, v_2, \dots, v_n) , such that any two consecutive vertices v_i, v_{i+1} are connected by the line segment $\overline{v_i v_{i+1}}$, for $1 \leq i < n$. It is said to be *closed* when it divides the plane into two disjoint regions. A *polygon* is a bounded region which is enclosed by a closed polygonal chain in \mathbb{R}^2 . A line segment is *orthogonal* if it is parallel either to the x -axis or y -axis.

Definition 1. (*Orthogonal polygon*) A *polygon* is said to be an *orthogonal polygon* if all of its sides are orthogonal.

Definition 2. (*Ortho-convex polygon*)[18] An *orthogonal polygon* \mathcal{P} is said to be *ortho-convex* if every horizontal or vertical line segment connecting a pair of points in \mathcal{P} lies totally within \mathcal{P} .

Definition 3. (*Shortest L_1 path*) A path between two points p and q is said to be a *shortest L_1 path* between them if the path consists of orthogonal line segments with total length $\|pq\|_1$.

Lemma 1. [19] For all pair of points in an *ortho-convex polygon* \mathcal{P} , there exist a *shortest L_1 path* between them in \mathcal{P} .

3.2. $\mathcal{OCP}(S)$ and $\mathcal{H}(\mathcal{OCP}(S))$

Let $S = \{p_1, p_2, \dots, p_n\}$ be a convex point set of size n in \mathbb{R}^2 . For any point $p \in S$, let $x(p)$ and $y(p)$ be its x and y -coordinate, respectively. We assume that the points in S are ordered with respect to an anticlockwise orientation along their convex hull. Without loss of generality let this ordering be p_1, p_2, \dots, p_n and also we assume that p_1 is the top most point in S , i.e., point having the largest y -coordinate in S (for multiple points having largest y -coordinate, we

take the one that has smallest x -coordinate). We denote the right most point of S as r . Analogously, let l , t , and b denote the left most, the top most and the bottom most point of S , respectively. So $t = p_1$. We will consider the point set for the case that $x(p_1) < x(b)$. For the case of $x(p_1) \geq x(b)$, both the construction and the proof are symmetric (by taking the mirror image of the point set with respect to the line $y = y(p_1) + 1$).

A polygonal chain is said to be a xy -monotone if any orthogonal line segment intersects the chain in a connected set. Now we will construct an orthoconvex polygon $\mathcal{OCP}(S)$, where points in S lie on the boundary of $\mathcal{OCP}(S)$. $\mathcal{OCP}(S)$ consists of four xy -monotone chains. Let us denote these chains as C_{rt} , C_{tl} , C_{lb} , and C_{br} . C_{rt} defines a xy -monotone chain with the endpoints at r and t . Analogously, C_{tl} , C_{lb} , and C_{br} are defined. While constructing the chain C_{rt} , we do the following: For any pair of consecutive points p, q , if $x(p) > x(q)$ then we draw two line segments $\overline{pp'}, \overline{qp'}$, where $p' = (x(p), y(q))$, else we extend the chain upto the next point. In Algorithm 1 and Algorithm 2, we describe the construction of C_{rt} and C_{br} respectively. Construction for the all other monotone chains follows the same set of rules. See Figure 5 for an illustration.

Algorithm 1 Construction of the chain C_{rt}

Input: A set of k points $p_i (= r), p_{i+1}, \dots, p_{i+k-1} (= t)$ such that

$$x(p_{j+1}) \leq x(p_j), y(p_{j+1}) \geq y(p_j) \text{ for } i \leq j < (i + k - 1)$$

Output: The chain C_{rt}

- 1: **for** $j = i$ to $(i + k - 2)$ **do**
 - 2: **if** $x(p_j) = x(p_{j+1})$ or $y(p_j) = y(p_{j+1})$ **then**
 - 3: Join the line segments $\overline{p_j p_{j+1}}$
 - 4: **else**
 - 5: Create a Steiner point $p_{j,j+1} = (x(p_j), y(p_{j+1}))$
 - 6: Join the line segments $\overline{p_j p_{j,j+1}}$ and $\overline{p_{j,j+1} p_{j+1}}$
-

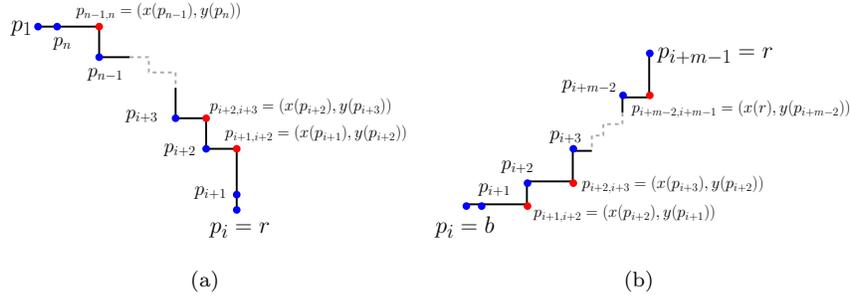


Figure 5: Construction of chains (a) C_{rt} and (b) C_{br} from a given convex point set (blue color)

Algorithm 2 Construction of the chain C_{br}

Input: A set of m points $p_i (= b), p_{i+1}, \dots, p_{i+m-1} (= r)$ such that

$$x(p_{j+1}) \geq x(p_j), y(p_{j+1}) \geq y(p_j) \text{ for } i \leq j < (i + m - 1)$$

Output: The chain C_{br}

- 1: **for** $j = i$ to $(i + m - 2)$ **do**
 - 2: **if** $x(p_j) = x(p_{j+1})$ or $y(p_j) = y(p_{j+1})$ **then**
 - 3: Join the line segments $\overline{p_j p_{j+1}}$
 - 4: **else**
 - 5: Create a Steiner point $p_{j,j+1} = (x(p_{j+1}), y(p_j))$
 - 6: Join the line segments $\overline{p_j p_{j,j+1}}$ and $\overline{p_{j,j+1} p_{j+1}}$
-

In Figure 6, we illustrate an example of a convex point set S of size 15 and the ortho-convex polygon $\mathcal{OCP}(S)$ is shown in Figure 6(b).

Definition 4. (Histogram) A histogram H is an orthogonal polygon consisting of a boundary edge e , called as its base, such that for any point $p \in H$, there exists a point $q \in e$ such that the line segment \overline{pq} is orthogonal and it lies completely in H .

If the base is horizontal (respectively, vertical) we say it is a *horizontal* (respectively, *vertical*) histogram. If its interior is above the base it is called an *upper* histogram. Similarly, we can define the *lower*, *left*, and *right* histograms. Now we construct a histogram partition $\mathcal{H}(\mathcal{OCP}(S))$ of $\mathcal{OCP}(S)$.

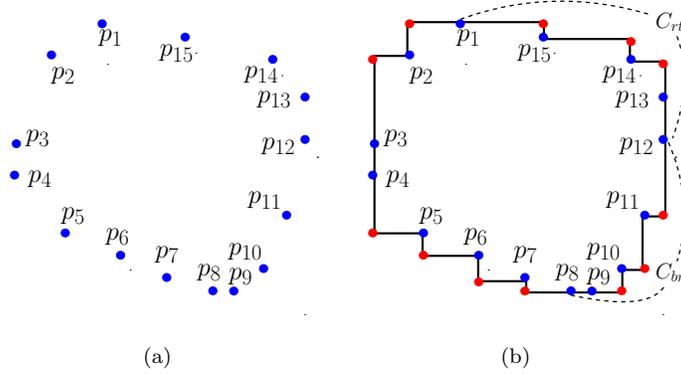


Figure 6: (a) Example of a set S of 12 points in convex position, (b) $\mathcal{OCP}(S)$ of S .

Let $L = \overline{pq}$ be a vertical line segment such that both the points p and q are on the boundary of $\mathcal{OCP}(S)$. We define H_L^r and H_L^l to denote a right-vertical and left-vertical histogram, respectively, with base $L = \overline{pq}$. Similarly, for a horizontal line segment $L' = \overline{p'q'}$, where both the points p' and q' are on the boundary of $\mathcal{OCP}(S)$, we define $H_{L'}^u$ and $H_{L'}^b$ to denote an upper-horizontal and lower-horizontal histograms, respectively, with base $L' = \overline{p'q'}$. Let $\text{proj}_L(p)$ be the orthogonal projection of the point p on the line containing the segment L . For a set A of orthogonal line segments and a point set S , we say A can see S if $\forall p \in S$ there is at least one line segment $L \in A$ such that $\text{proj}_L(p) \in L$. For a vertical (respectively, horizontal) line segment L , we define $x(L)$ (respectively, $y(L)$) to be the x -coordinate (respectively, y -coordinate) of L .

We obtain a histogram partition $\mathcal{H}(\mathcal{OCP}(S))$ of $\mathcal{OCP}(S)$ by recursively drawing vertical and horizontal lines as follows (see Figure 8):

Step 1 Let $q_1 (\in C_{lb})$ be the intersection point of the boundary of $\mathcal{OCP}(S)$ with the vertical line containing p_1 . First, we draw a vertical line segment $L_1 = \overline{p_1q_1}$. We define two sets $S(H_{L_1}^l)$ and $S(H_{L_1}^r)$ such that $S(H_{L_1}^l) = \{q \in S : y(q) \geq y(q_1) \text{ and } x(q) \leq x(q_1)\}$, $S(H_{L_1}^r) = \{q \in S : y(q) \geq y(q_1) \text{ and } x(q) \geq x(q_1)\}$. In this step, we construct two vertical histograms $H_{L_1}^l$ and $H_{L_1}^r$. If $S(H_{L_1}^l) \cup S(H_{L_1}^r) = S$, i.e., L_1 can see S we

stop, else we proceed to Step 2.

Step 2: Let q_2 ($\notin C_{lb}$) be the intersection point of the boundary of $\mathcal{OCP}(S)$ with the horizontal line containing q_1 . Then we draw a horizontal line segment $L_2 = \overline{q_1q_2}$. Here we define the set $S(H_{L_2}^b) = \{z \in S: x(q_1) \leq x(z) \leq x(q_2) \text{ and } y(z) \leq y(q_2)\}$. In this step, we construct the lower histogram $H_{L_2}^b$ with base L_2 . If $S(H_{L_1}^l) \cup S(H_{L_1}^r) \cup S(H_{L_2}^b) = S$, i.e., $\{L_1, L_2\}$ can see S we stop, else we proceed to the next step.

Step 3: Let q_3 ($\notin C_{rt}$) be the intersection point of the boundary of $\mathcal{OCP}(S)$ with the vertical line containing q_2 . Then we draw a vertical line segment $L_3 = \overline{q_2q_3}$. Here we define the set $S(H_{L_3}^r) = \{w \in S: y(q_2) \geq y(w) \geq y(q_3) \text{ and } x(w) \geq x(q_3)\}$. In this step, we construct the right histogram $H_{L_3}^r$ with base L_3 . If $S(H_{L_1}^l) \cup S(H_{L_1}^r) \cup S(H_{L_2}^b) \cup S(H_{L_3}^r) = S$, i.e., $\{L_1, L_2, L_3\}$ can see S we stop, else we proceed in the similar manner.

We assume that this process terminates after k steps, and we obtain a set \mathcal{L} of orthogonal line segments $\{L_1, L_2, \dots, L_k\}$ for some $k \in \mathbb{N}$ such that $\{L_1, L_2, \dots, L_k\}$ can see S . In this process, we add k Steiner points $\{q_i: 1 \leq i \leq k\}$. Each q_i belongs to the boundary of $\mathcal{OCP}(S)$.

The process terminates in one of the four following configurations which are based on the position of the points b and r (see Figure 7).

Type-1 L_k is vertical and $\text{proj}_{L_{k-1}}(b) \in L_{k-1}$, i.e., L_{k-1} sees b .

Type-2 L_k is vertical and $\text{proj}_{L_{k-1}}(b) \notin L_{k-1}$.

Type-3 L_k is horizontal and $\text{proj}_{L_{k-1}}(r) \in L_{k-1}$, i.e., L_{k-1} sees r .

Type-4 L_k is horizontal and $\text{proj}_{L_{k-1}}(r) \notin L_{k-1}$.

From now onwards, we assume that L_1, L_2, \dots, L_k are the segments inserted in $\mathcal{OCP}(S)$ while constructing $\mathcal{H}(\mathcal{OCP}(S))$. Let $\mathcal{L} = \cup_{i=1}^k L_i$. So for any point $p \in S$, there is at least one line segment $L \in \mathcal{L}$ such that $\text{proj}_L(p) \in L$ and the segment $\overline{p \text{proj}_L(p)}$ completely lies in $\mathcal{OCP}(S)$.

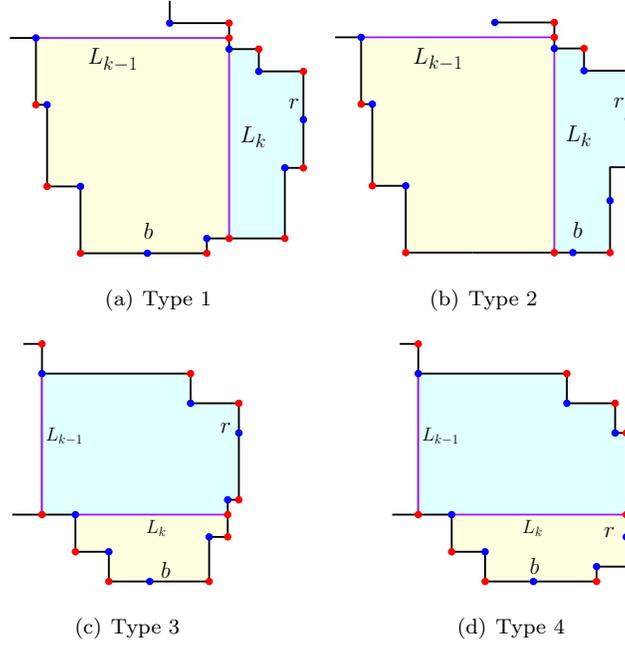


Figure 7: Types of the histogram containing b and r in $\mathcal{OCP}(S)$.

Lemma 2. $\mathcal{H}(\mathcal{OCP}(S))$ can be constructed in linear time.

Proof. Let $L_i(S) = \{p \in S : L_i \text{ can see } p\}$. First we show that $\mathcal{H}(\mathcal{OCP}(S))$ is a histogram partition in $\mathcal{OCP}(S)$, i.e., $\cup_{i=1}^k L_i(S) = S$. L_1 sees all points $q \in S$ having the property that $y(q_1) \leq y(q) \leq y(t)$ as $\mathcal{OCP}(S)$ is an ortho-convex polygon and these points are part of xy -monotone chains $\{C_{rt}, C_{tl}, C_{lb}, C_{br}\}$. So $L_1(S)$ consists of all the points in S that lie above L_2 . Moreover, all the points above L_2 are part of the histogram defined by the base L_1 . Now our concern is only about the points of S that are below L_2 . Now L_2 can see the points $q \in (S \setminus L_1(S))$ having the property that $x(q_1) \leq x(q) \leq x(q_2)$. These points are part of the histogram with the base L_2 . Now we can apply the same argument inductively. This leads to the claim that $\cup_{i=1}^k L_i(S) = S$, i.e., $\mathcal{L} = \{L_1, \dots, L_k\}$ can see S . Observe that the segments in \mathcal{L} can be computed by walking around the boundary of $\mathcal{OCP}(S)$ in linear time. Hence, $\mathcal{H}(\mathcal{OCP}(S))$ can be constructed in linear time. \square

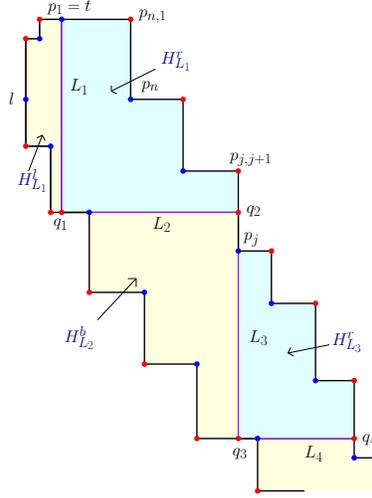


Figure 8: $\mathcal{H}(\mathcal{OCP}(S))$ of a convex point set S

3.3. Construction of Planar Manhattan Network

Now we describe our construction of planar Manhattan network $G = (V, E)$ for a convex point set S . For an illustration of the steps of Algorithm 3, see Figure 9. Recall that $\text{proj}_L(p)$ denotes the orthogonal projection of the point p on the line containing the segment L and $q(H)$ denotes the histogram containing $q \in S$ in $\mathcal{H}(\mathcal{OCP}(S))$. Let e_1, e_2, e_3 be the bases of $l(H), b(H)$, and $r(H)$, respectively, where $l(H)$ (respectively $b(H)$ and $r(H)$) denotes the histogram containing l (respectively b and r) of S . First, we draw the segments $e'_1 = \overline{l \text{ proj}_{e_1}(l)}$, $e'_2 = \overline{b \text{ proj}_{e_2}(b)}$, and $e'_3 = \overline{r \text{ proj}_{e_3}(r)}$ in $\mathcal{OCP}(S)$. Let $\mathcal{L}' = \mathcal{L} \cup \{e'_1, e'_2, e'_3\}$. Next, for each $q \in S$, if $\text{proj}_L(q) \in L$ where $L \in \mathcal{L}' \cap q(H)$, we draw the line segment $\overline{q \text{ proj}_L(q)}$ in $\mathcal{OCP}(S)$. Then if both $H^r_{L_k}$ and e_2 exist, we draw the segments $\text{proj}_{e_2}(z)$, for each point $z \in S \cap H^r_{L_k}$. Also if both $H^b_{L_k}$ and e_3 exist, we draw the segments $\text{proj}_{e_3}(w)$, for each point $w \in S \cap H^b_{L_k}$. In this process, all the line segments we join, we add them into edges of T . Also all the extra points we created to make an orthogonal projection, we add them into the set T of Steiner vertices. Our algorithm ends with removing some specific line segments, that is stated in the Steps 28-32 in Algorithm 3. We illustrate this algorithm in Figure 11(a).

Algorithm 3 Construction of $G = (V = S \cup T, E)$

Input: $\mathcal{H}(\mathcal{OCP}(S))$ of a convex point set $S = \{p_1(=t), p_2, \dots, p_n\}$.

Let $\{L_1, L_2, \dots, L_k\}$ be the segments and $\{q_i: 1 \leq i \leq k\}$ be the set of points inserted in $\mathcal{OCP}(S)$ during the construction of $\mathcal{H}(\mathcal{OCP}(S))$.

Output: A planar Manhattan network $G = (V = S \cup T, E)$ of S .

- 1: $S \leftarrow \{p_i: 1 \leq i \leq n\}$;
- 2: $T \leftarrow \{p_{i,i+1}: 1 \leq i \leq n\} \cup \{q_i: 1 \leq i \leq k\}$;
- 3: $E \leftarrow \{\overline{p_i p_{i,i+1}}: 1 \leq i \leq (n-1)\} \cup \{\overline{p_{i+1} p_{i,i+1}}: 1 \leq i \leq (n-1)\} \cup \overline{p_n p_{n,1}} \cup \overline{p_1 p_{n,1}}$;
▷ see Figure 9(a)
- 4: Draw the line segments (if they do not exist) $e'_1 = \overline{l \text{ proj}_{e_1}(l)}$, $e'_2 = \overline{b \text{ proj}_{e_2}(b)}$,
and $e'_3 = \overline{r \text{ proj}_{e_3}(r)}$ ▷ e_1, e_2 , and e_3 are the bases of the histograms $l(H)$, $b(H)$, and $r(H)$, respectively.
- 5: $T = T \cup \{\text{proj}_{e_1}(l), \text{proj}_{e_2}(b), \text{proj}_{e_3}(r)\}$
- 6: $\mathcal{L}' = \{L_1, L_2, \dots, L_k, e'_1, e'_2, e'_3\}$
- 7: **for each point** $q \in S$ **do**
- 8: **for each line** $L \in \mathcal{L}' \cap q(H)$ **do**
- 9: **if** $\text{proj}_L(q) \in L$ **then**
- 10: $T = T \cup \text{proj}_L(q)$;
- 11: $E = E \cup \overline{q \text{ proj}_L(q)}$ ▷ see Figure 9(b)
- 12: **if** Both $H_{L_k}^r$ and e_2 exist **then**
- 13: **for each point** $z \in S \cap H_{L_k}^r$ **do**
- 14: $T = T \cup \text{proj}_{e_2}(z)$;
- 15: $E = E \cup \overline{z \text{ proj}_{e_2}(z)}$ ▷ see Figure 9(c)
- 16: **if** Both $H_{L_k}^b$ and e_3 exist **then**
- 17: **for each point** $w \in S \cap H_{L_k}^b$ **do**
- 18: $T = T \cup \text{proj}_{e_3}(w)$;
- 19: $E = E \cup \overline{w \text{ proj}_{e_3}(w)}$ ▷ see Figure 9(d)
- 20: **for each horizontal line segment** $L \in \mathcal{L}'$ **do**
- 21: Let L contains k_1 vertices a_1, a_2, \dots, a_{k_1} , where $x(a_i) < x(a_{i+1})$ for

$1 \leq i < k_1$
22: **for** $1 \leq i \leq (k_1 - 1)$ **do**
23: $E = E \cup \overline{a_i a_{i+1}}$ ▷ see Figure 9(e)
24: **for** each vertical line segment $L \in \mathcal{L}'$ **do**
25: Let L contains k_2 vertices b_1, b_2, \dots, b_{k_2} , where $y(b_i) < y(b_{i+1})$ for $1 \leq i < k_2$
26: **for** $1 \leq i \leq (k_2 - 1)$ **do**
27: $E = E \cup \overline{b_i b_{i+1}}$ ▷ see Figure 9(g)
28: Delete the following three edges if they exist.
29: (i) The edge $(u_1, \text{proj}_{e_1}(l))$ on the line e'_1 provided that $u_1 \neq l$. ▷ see Figure 9(f)
30: (ii) For the **Types 1 or 4**, the edge $(u_2, \text{proj}_{e_2}(b))$ on the line e'_2 provided that $u_2 \neq b$. ▷ see Figure 9(h)
31: (iii) For the **Types 2 or 3**, the edge $(u_3, \text{proj}_{e_3}(r))$ on the line e'_3 provided that $u_3 \neq r$. ▷ see Figure 9(i)
32: For the **Types 1 or 3**, delete all the vertices v on the line L_k where $v \notin \{\text{proj}_{e_3}(r), \text{proj}_{e_2}(b)\}$ and v is not a point on the boundary of $\mathcal{OCP}(S)$.
33: **return** $G = (S \cup T, E)$

Notice that for each point in S , Algorithm 3 adds at most three Steiner vertices in G . Specifically, $|V(G)| \leq 4n$ and $|E(G)| \leq 5n$. So both the number of vertices and edges in G are $\mathcal{O}(n)$. Now we prove the following lemma.

Lemma 3. *For the point set S , G can be constructed in $\mathcal{O}(n)$ time.*

Proof. The construction of G from S consists of three Steps. In Step 1, we construct $\mathcal{OCP}(S)$ from S . As for each point $p \in S$, we add exactly one Steiner point and draw two edges, $\mathcal{OCP}(S)$ consists of $2n$ points including S . So, Step 1 takes $\mathcal{O}(n)$ time. In Step 2, we construct a histogram partition $\mathcal{H}(\mathcal{OCP}(S))$ of $\mathcal{OCP}(S)$. By Lemma 2, it needs $\mathcal{O}(n)$ time. In the final Step, we apply Algorithm 3 in $\mathcal{H}(\mathcal{OCP}(S))$ to construct our desired graph $G = (V, E) = (S \cup T, E)$. Now we show Algorithm 3 runs in $\mathcal{O}(n)$ time. In this algorithm, Steps 1-4 take linear time. In Steps 7-11, for each point $q \in S$, we perform orthogonal

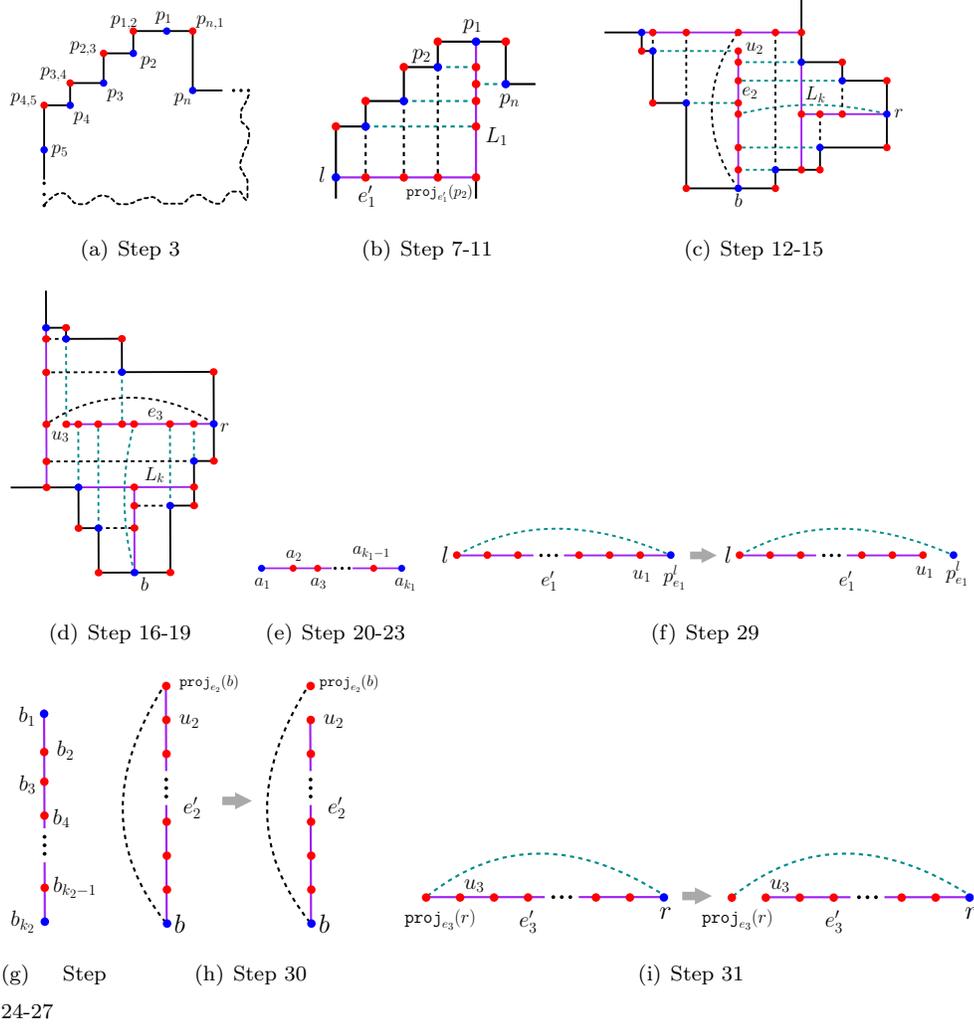


Figure 9: Illustration of the Steps in Algorithm 3. We maintain following convention of colors. We use purple color while drawing the line segment of the set \mathcal{L}' . We use dashed black and dashed cyan line to denote vertical and horizontal projections, respectively of the points S to lines of \mathcal{L}' . Blue and red color points identify points from S and Steiner points, respectively.

projections at most two times, i.e., we add at most two Steiner vertices and two edges. The points of S are given in sorted order along their convex hull. Also, we have an ordered set of k line segments L_1, L_2, \dots, L_k with the ordering based on the construction of $\mathcal{H}(\mathcal{OCP}(S))$. Now, for any pair of points p_i and p_{i+1} ,

where $1 \leq i \leq n$ if the point p_i has an orthogonal projection on L_m for some m then p_{i+1} can not have an orthogonal projection onto any line segment in $\mathcal{L} \setminus \{L_{m-1}, L_m, L_{m+1}\}$. So it takes $\mathcal{O}(n+k)$ time to perform all the projections in Steps 7-11 by walking around the boundary of $\mathcal{OCP}(S)$ once. The Steps 12-15 occur only when both $H_{L_k}^t$ and e_2 exist. Now we have to do one more projection for each point of $S \cap H_{L_k}^t$ to e_2 . So Steps 12-15 take linear time. Similarly, Steps 16-19 take linear time. In Steps 20-25, we add edges to E by looking at each line segment of $\{L_1, L_2, \dots, L_k, e_1, e_2, e_3\}$. As the number of projections is linear so the number of edges we add in Steps 20-25 is also linear. In Step 26, we delete some edges from $\{e_1, e_2, e_3, L_k\}$. So the total time complexity is $\mathcal{O}(n+k)$. As $k \leq n$, Algorithm 3 produces G in $\mathcal{O}(n)$ time. Hence the proof. \square

3.4. G is a Manhattan Network

To show that G is a Manhattan network, we have to prove that G contains a shortest L_1 path between every pair of points in S . Recall that $p(H)$ denotes the histogram containing $p \in S$ in $\mathcal{H}(\mathcal{OCP}(S))$ and $\mathcal{L} = \{L_1, L_2, \dots, L_k\}$ denotes the set of k segments inserted in $\mathcal{OCP}(S)$ while constructing $\mathcal{H}(\mathcal{OCP}(S))$. First we prove the following lemma.

Lemma 4. *For any two points w and z in S , if $w(H) \neq z(H)$ then there always exist lines L and L' such that (i) $\text{proj}_L(w) \in L$, $\text{proj}_{L'}(z) \in L'$ and (ii) if we draw a line L^* that contains line L (respectively, L') then w and z belong to opposite sides of L^* .*

Proof. Let w and z be two points in S such that $w(H) \neq z(H)$. Without loss of generality we assume that $x(w) < x(z)$. By our construction of $\mathcal{H}(\mathcal{OCP}(S))$, $x(L_1) < x(L_3) < \dots$ and $y(L_2) > y(L_4) > \dots$. If $x(w) \leq x(L_1)$ then $L = L_1$. Let $x(w) \geq x(L_1)$ and i be the largest integer such that $x(L_i) \leq x(w)$. If L_{i+2} exists and $\text{proj}_{L_{i+2}}(w) \in L_{i+2}$ then $L = L_{i+2}$, else $L = L_{i+1}$. Similarly let j be the largest integer such that $x(L_j) \leq x(z)$. If $\text{proj}_{L_j}(z) \in L_j$ then $L' = L_j$, else $L' = L_{j+1}$. \square

Now we prove the following lemma.

Lemma 5. *For each pair of points p_i and p_j of S where $1 \leq i, j \leq n$, there exists a shortest L_1 path in G between them.*

Proof. $\mathcal{OCP}(S)$ consists of four xy -monotone chains C_{rt}, C_{tl}, C_{lb} , and C_{br} . Let p_i and p_j be two arbitrary points of S where $1 \leq i, j \leq n$. Let $\pi_G(a, b) = \langle a, \dots, v_i, \dots, b \rangle$ denotes a shortest L_1 path between a pair of vertices a and b in G . Let P_1 and P_2 be two paths from a to b and b to c , respectively. By $P_1 \rightsquigarrow P_2$ we mean the path from a to c that is obtained by concatenating the paths P_1 and P_2 . The proof of this theorem can be divided into Case A and Case B.

Case A: Both p_i and p_j belong to the same xy -monotone chain: Each xy -monotone chain of the ortho-convex polygon $\mathcal{OCP}(S)$ is a Manhattan network for the points it contains.

Case B: p_i and p_j belong to different chains: We divide this case into two subcases B.1. and B.2.

Case B.1. $p_i(H) = p_j(H)$, i.e., p_i, p_j belong to the same histogram

- (1) **$p_i, p_j \in l(H)$:** If $p_i \in C_{tl}, p_j \in C_{lb}$ then $\pi_G(p_i, p_j) = \langle p_i, \mathbf{proj}_{e'_1}(p_i) \rangle \rightsquigarrow \pi_G(\mathbf{proj}_{e'_1}(p_i), \mathbf{proj}_{e'_1}(p_j)) \rightsquigarrow \langle \mathbf{proj}_{e'_1}(p_j), p_j \rangle$.
- (2) **$p_i, p_j \in r(H)$:** For Types 1, 2, or 3, if $p_i \in C_{rt}$ and $p_j \in C_{br} \cup C_{lb}$, then $\pi_G(p_i, p_j) = \langle p_i, \mathbf{proj}_{e'_3}(p_i) \rangle \rightsquigarrow \pi_G(\mathbf{proj}_{e'_3}(p_i), \mathbf{proj}_{e'_3}(p_j)) \rightsquigarrow \langle \mathbf{proj}_{e'_3}(p_j), p_j \rangle$. For Type 3, if $p_i \in C_{lb}$ and $p_j \in C_{br}$ then $\pi_G(p_i, p_j) = \langle p_i, \mathbf{proj}_{L_k}(b) \rangle \rightsquigarrow \pi_G(\mathbf{proj}_{L_k}(b), p_j)$. For Type 4, if $p_i \in C_{lb}$ and $p_j \in C_{br}$ then $\pi_G(p_i, p_j) = \langle p_i, \mathbf{proj}_{e'_2}(p_i) \rangle \rightsquigarrow \pi_G(\mathbf{proj}_{e'_2}(p_i), \mathbf{proj}_{e'_2}(p_j)) \rightsquigarrow \langle \mathbf{proj}_{e'_2}(p_j), p_j \rangle$.
- (3) **$p_i, p_j \in b(H)$:** For Types 1 or 3, if $p_i \in C_{lb}$ and $p_j \in C_{br} \cup C_{rt}$, then $\pi_G(p_i, p_j) = \langle p_i, \mathbf{proj}_{e'_2}(p_i) \rangle \rightsquigarrow \pi_G(\mathbf{proj}_{e'_2}(p_i), \mathbf{proj}_{e'_2}(p_j)) \rightsquigarrow \langle \mathbf{proj}_{e'_2}(p_j), p_j \rangle$. For Type 1, (i) if $p_i \in C_{rt}$ and $p_j \in C_{br}$ then $\pi_G(p_i, p_j) = \langle p_i, \mathbf{proj}_{L_k}(r) \rangle \rightsquigarrow \pi_G(\mathbf{proj}_{L_k}(r), p_j)$ or $\pi_G(p_j, p_i) = \langle p_j, \mathbf{proj}_{L_{k-1}}(p_j) \rangle \rightsquigarrow \pi_G(\mathbf{proj}_{L_{k-1}}(p_j), p_i)$. (ii) if $p_i \in C_{lb}$ and $p_j \in$

C_{rt} then the shortest L_1 path between \mathbf{p}_i and \mathbf{p}_j in G is $\pi_G(p_i, p_j) = \langle p_i, \mathbf{proj}_{e_2}(p_i) \rangle \rightsquigarrow \pi_G(\mathbf{proj}_{e_2}(p_i), \mathbf{proj}_{e_2}(p_j)) \rightsquigarrow \langle \mathbf{proj}_{e_2}(p_j), p_j \rangle$ or $\pi_G(p_j, p_i) = \langle p_i, \mathbf{proj}_{L_{k-1}}(p_i) \rangle \rightsquigarrow \pi_G(\mathbf{proj}_{L_{k-1}}(p_i), p_j)$. For Types 2 or 4 as $b(H) = r(H)$, it is similar as subcase (2) of B.1.

(4) $\mathbf{p}_i, \mathbf{p}_j \notin \{\mathbf{l}(H), \mathbf{b}(H), \mathbf{r}(H)\}$: Let these histograms contain two elements say L and L' of \mathcal{L} . In this case $\pi_G(p_i, p_j) = \langle p_i, \mathbf{proj}_L(p_i) \rangle \rightsquigarrow \pi_G(\mathbf{proj}_L(p_i), \mathbf{proj}_L(p_j)) \rightsquigarrow \langle \mathbf{proj}_L(p_j), p_j \rangle$ or $\pi_G(p_i, p_j) = \langle p_i, \mathbf{proj}_{L'}(p_i) \rangle \rightsquigarrow \pi_G(\mathbf{proj}_{L'}(p_i), \mathbf{proj}_{L'}(p_j)) \rightsquigarrow \langle \mathbf{proj}_{L'}(p_j), p_j \rangle$.

Case B.2. $\mathbf{p}_i(H) \neq \mathbf{p}_j(H)$: First, we find line segments $L, L' \in \mathcal{L}$ such that (i) L can see p_i , L' can see p_j , and (ii) if we draw a line L^* that contains line L (respectively, L') then p_i and p_j belong to opposite sides of L^* . By Lemma 4 both L and L' exist in \mathcal{L} but it may happen that $L = L'$ e.g., for the points p_2 and p_n , $p_2(H) \neq p_n(H)$ with $L = L'$. By the construction of G , both $\mathbf{proj}_L(p_i)$ and $\mathbf{proj}_{L'}(p_j)$ belong to $T \subset V$. We complete this case by proving following lemma.

Lemma 6. *Let w and z be two points in S such that $w(H) \neq z(H)$. Also let L and L' be two segments such that (i) $\mathbf{proj}_L(w) \in L$, $\mathbf{proj}_{L'}(z) \in L'$ and (ii) if we draw a line L^* that contains line L (respectively, L') then w and z belong to opposite sides of L^* . Then there exist a shortest L_1 path between $\mathbf{proj}_L(w)$ and $\mathbf{proj}_{L'}(z)$ in G .*

Proof. Without loss of generality, we assume that $x(w) < x(z)$. If $L = L'$ then $\pi_G(\mathbf{proj}_L(w), \mathbf{proj}_{L'}(z))$ is along the line L . For example, if we take $w = p_2$ and $z = p_n$ then $L = L' = L_1$. So we are left with the case when $L \neq L'$. For example, in Figure 11(a), considering l as w and r as z we find $L = L_1$ and $L' = L_k$. Rest of the proof can be divided into two cases. Recall that $\{L_1, L_2, \dots, L_k\}$ are the segments inserted in $\mathcal{OCP}(S)$ while constructing $\mathcal{H}(\mathcal{OCP}(S))$. The point set $\{q_i : 1 \leq i \leq k\}$ comes from the construction of $\mathcal{H}(\mathcal{OCP}(S))$. Assuming $l = q_0$, L_i is the segment with end points q_{i-1} and q_i , where $1 \leq i \leq k$.

Case 1. L is vertical: Let $L = L_m$ for some $m, 1 \leq m \leq k$. So $L_m = \overline{q_{m-1}q_m}$. By the construction of $\mathcal{H}(\mathcal{OCP}(S))$, q_m is not only a point on the boundary of $\mathcal{OCP}(S)$ but also there exists a point say p_j in S such that $q_m \in \overline{p_{j,j-1}p_j}$. Now we divide this case into following two subcases.

Case 1.1 L' is vertical: By similar argument as L , there exists a point $p_{j'}$ such that $y(p_{j'}) \geq y(z)$ and $p_{j'} \in L'$. For this case, a shortest L_1 path between $\text{proj}_L(w)$ and $\text{proj}_{L'}(z)$ in G is $\pi_G(\text{proj}_L(w), q_m) \rightsquigarrow \pi_G(q_m, p_j) \rightsquigarrow \pi_G(p_j, p_{j'}) \rightsquigarrow \pi_G(p_{j'}, \text{proj}_{L'}(z))$. By repeatedly applying this argument we can find $\pi_G(p_j, p_{j'})$. For an illustration, see Figure 10(a).

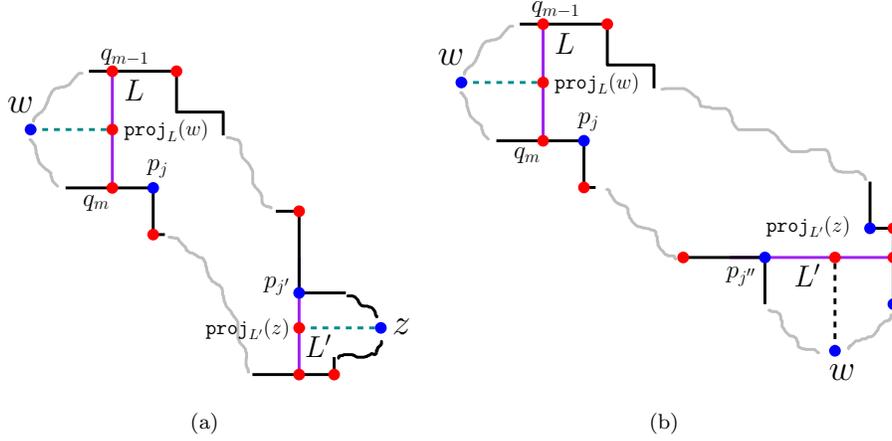


Figure 10: (a) Both L and L' are vertical. (b) L is vertical, L' is horizontal.

Case 1.2 L' is horizontal: By similar argument as L , there exists a point $p_{j''}$ such that $x(p_{j''}) \geq x(w)$ and $p_{j''} \in L'$. Rest of this case is similar as case 1.1. Here a shortest L_1 path between $\text{proj}_L(w)$ and $\text{proj}_{L'}(z)$ in G is $\pi_G(\text{proj}_L(w), q_m) \rightsquigarrow \pi_G(q_m, p_j) \rightsquigarrow \pi_G(p_j, p_{j''}) \rightsquigarrow \pi_G(p_{j''}, \text{proj}_{L'}(z))$. By repeatedly applying this argument we can find $\pi_G(p_j, p_{j''})$. For an illustration, see Figure 10(b).

Case 2. L is horizontal : Proof for this case is similar as case 1.

□

3.5. Planarity of G

In this section, we show that the graph $G = (V, E)$ is planar by providing a planar embedding. For an illustration, see Figure 11(b).

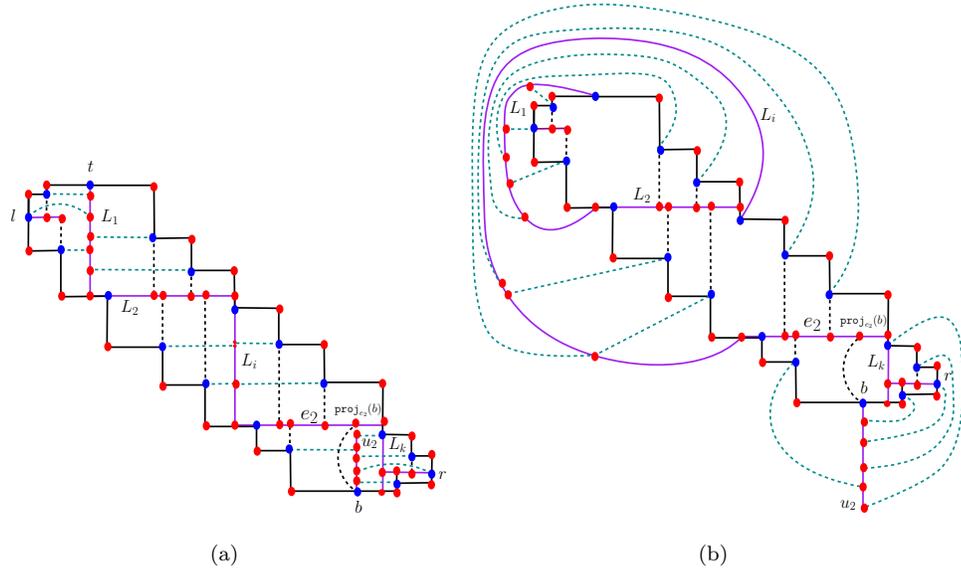


Figure 11: (a) Output G of Algorithm 3 for point set in blue color. (b) Planar embedding of G .

The *union* of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is defined as the graph $(V_1 \cup V_2, E_1 \cup E_2)$ [20]. We will make use of the following theorem regarding planar graphs.

Theorem 2. [21] *A planar embedding of a graph can be transformed into another planar embedding such that any specified face becomes the exterior face.*

Relation to k -plane graphs [22]. A geometric graph $G = (V, E)$ is said to be k -plane graph for some $k \in \mathbb{N}$ if E can be partitioned into k disjoint subsets, $E = E_1 \cup E_2 \cup \dots \cup E_k$, such that $G_1 = (V, E_1), G_2 = (V, E_2), \dots, G_k = (V, E_k)$ are all plane graphs, where \cup represents the disjoint union. For a finite general point

set P in the plane, $\mathcal{G}_k(P)$ denotes the family of k -plane graphs with vertex set P . As per as our construction, the graph we construct to form a *Manhattan network* for convex point set is basically a 2-plane graph.

Theorem 3. *Graph G computed in Algorithm 3 is planar.*

Proof. We decompose G into two subgraphs H and K such that $G = H \cup K$. This decomposition depends on the line L_{k-1} . In order to construct the histogram partition in $\mathcal{OCP}(S)$, L_{k-1} may be horizontal or vertical. For Types 1 and 2, L_{k-1} is horizontal. For Types 3 and 4, L_{k-1} is vertical. We analyse each of the following two cases.

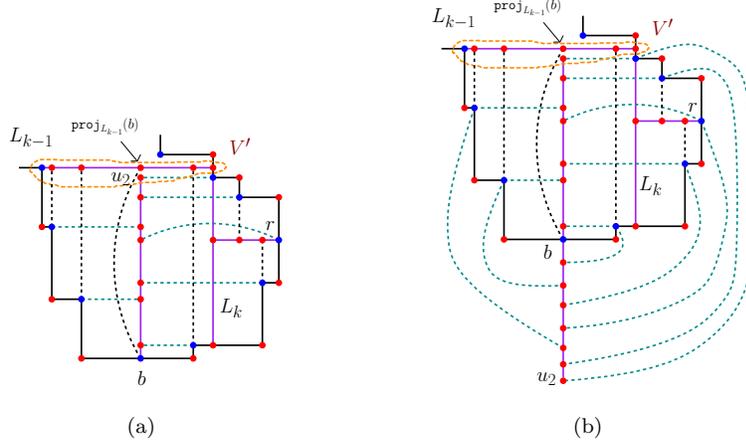


Figure 12: (a) The subgraph K of G for **Type 1** with the exterior face containing V' . (b) planar embedding of K . The edges of E_b are shown by dashed cyan segment.

Case 1. L_{k-1} is horizontal: H and K are the subgraphs of G induced by the vertices lying above and below, respectively of the line segment L_{k-1} , i.e., $V(H) = \{v: v \in V, y(v) \geq y(L_{k-1})\}$, where $y(L_{k-1})$ is the y -coordinate of any point on the segment L_{k-1} . Similarly, $V(K) = \{v: v \in V, y(v) \leq y(L_{k-1})\}$. Let $V' = V(H) \cap V(K)$. We want to show that G is planar, i.e., there exists a planar embedding G' of G . If we are able to show that there exist two planar embeddings, H' for H and K' for K , such that V' belongs to the exterior faces of both H' and K' , then we can obtain a planar embedding G' of G by attaching

the embeddings of H' and K' along the exterior face. Now our target is to show that H and K have planar embeddings H' and K' , respectively, such that V' is contained in the exterior face of both H' and K' . We define $V^f \subseteq V$ to denote the set of vertices of G along the boundary of $\mathcal{OCP}(S)$. To Get a planar embedding of G , we prove following two lemmas.

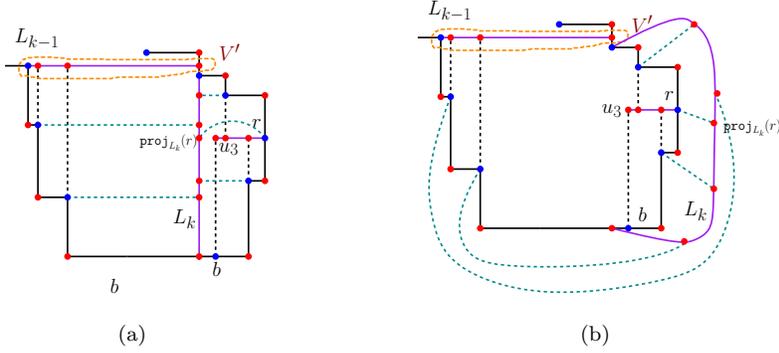


Figure 13: (a) The subgraph K of G for **Type 2** with the exterior face containing V' . (b) planar embedding of K . The edges of E_k are shown by dashed cyan segment.

Lemma 7. K has a planar embedding K' such that V' is contained in the exterior face of K' .

Proof. Let $V_k^f \subseteq V$ be the set of vertices in G along the exterior face of K . So $V_k^f = (V^f \cap V(K)) \cup V'$. For Type 1, let E_b be the set of horizontal edges that have at least one adjacent vertex on the segment $e_2' = \overline{b \text{proj}_{L_{k-1}}(b)}$. In this case, we draw the edges E_b in the exterior face of K in such a way that we obtain a planar embedding of K . In the planar embedding, all Steiner points on the line segment $\overline{bu_2}$ will go to the exterior of the polygon along with its adjacent edges. For Type 2, let E_k be the set of horizontal edges that have at least one adjacent vertex on the line L_k . In this case, we draw the edges E_k in the exterior faces of K in such a way that we obtain a planar embedding of K . In the embedding, all Steiner points on the line segment L_k will go to the exterior of the polygon along with its adjacent edges. In this planar embedding, V' still remains in the exterior face. Hence, we get a planar embedding K' of

K such that V' is contained in the exterior face of K' . For an illustration see Figure 12 and Figure 13. \square

Lemma 8. H has a planar embedding H' such that V' is contained in the exterior face of H' .

Proof. We prove this by weak induction. As L_{k-1} is horizontal, $(k-1)$ must be even. Let $(k-1) = 2m$ for some $m \in \mathbb{N}$. Let V_i consists of all the vertices in G on the line segment L_{2i} and G_i be the subgraph induced by the vertices lying on or above the line segment L_{2i} , where $2i \leq (k-1)$. So $G_m = H$. By induction, we prove that G_m is planar and it has a planar embedding H' such that V' is contained in the exterior face of H' . Let $P(i)$ be the following statement: G_i is planar and it has a planar embedding G'_i such that V_i is contained in the exterior face of G'_i . Now we need to show $P(m)$ is true. We first show that the base case is true. Next we show the inductive step.

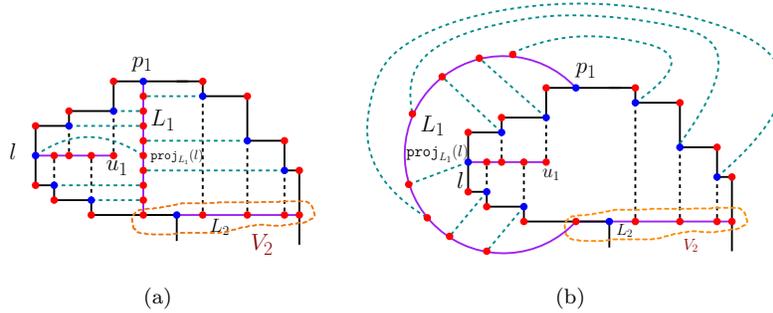


Figure 14: (a) The graph G_1 . (b) A planar embedding of G_1 with the exterior face containing V_2 .

Base Case: $P(1)$ is true: We divide the edges of G_1 into three sets E_{11} , E_{12} , and E_{13} . E_{11} is the set of edges in G_1 that are along the boundary of the exterior face of G_1 . E_{12} consists of all the edges in G_1 that have one endpoint on the segment L_1 . $E_{13} = E(G_1) \setminus (E_{11} \cup E_{12})$. Let G_{11} be the subgraph of G_1 consisting of the edges $E_{11} \cup E_{12}$, and G_{12} be the subgraph of G_1 consisting of the edges $E_{11} \cup E_{13}$. So $G_1 = G_{11} \cup G_{12}$, where both G_{11} and G_{12} are plane

graphs. In G_{11} there exists an interior face containing V_1 . Let V_{12}^f be the set of vertices in the exterior face of G_{12} . By Theorem 2, we can transform the planar embedding G_{12} into another planar embedding G'_{12} such that there exists an interior face, say f_1 , that contains V_{12}^f . As V_{12}^f is the set of vertices in the exterior face of G_{11} , so we can attach G_{11} in f_1 and obtain a planar embedding G''_1 of G_1 . In G''_1 there exists an interior face containing V_1 . Applying Theorem 2, we get a planar embedding G'_1 of G_1 such that V_1 is contained in the exterior face of G'_1 . We illustrate this step in Figure 14.

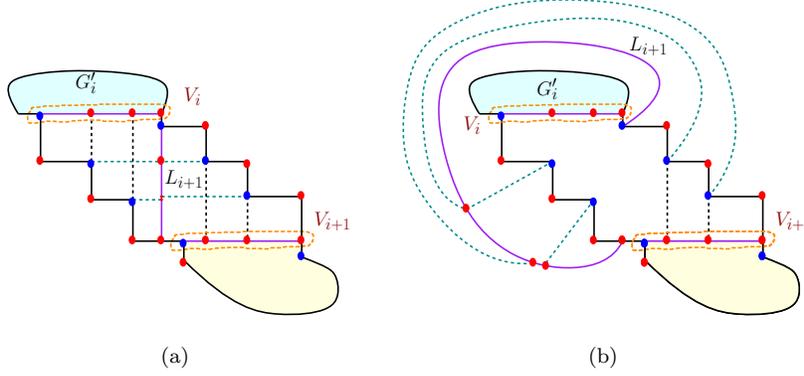


Figure 15: (a) The graph G with planar embedding G'_i having exterior face containing V_i . (b) The graph G with planar embedding G'_{i+1} having exterior face containing V_{i+1} .

Inductive Case: $P(i)$ is true $\Rightarrow P(i+1)$ is true: Assume that $P(i)$ is true, i.e., G_i has a planar embedding G'_i such that V_i is contained in the exterior face of G'_i (see Figure 15).

Let H_1 be the subgraph of G_{i+1} induced by the vertices lying on or below the line containing L_{2i} . Now $V_i = G'_i \cap H_1$, also V_i is contained in the exterior face of G'_i . As $G_{i+1} = G_i \cup H_1$ so in G_{i+1} , we can replace G_i by its planar embedding G'_i . Now $G_{i+1} = G'_i \cup H_1$. Now we divide the edges of G_{i+1} into three sets $E_{(i+1)1}, E_{(i+1)2}, E_{(i+1)3}$. $E_{(i+1)1}$ consists of edges in G_{i+1} that are along the boundary of the exterior face of H_1 . $E_{(i+1)2}$ consists of edges in H_1 that have one endpoint on the line containing L_{2i+1} . $E_{(i+1)3} = E(H_1) \setminus (E_{(i+1)1} \cup E_{(i+1)2})$. Let $G_{(i+1)1}$ be the subgraph of G_{i+1} consisting of the edges

$E_{(i+1)1} \cup E_{(i+1)2} \cup E(G'_i)$, and $G_{(i+1)2}$ be the subgraph of G_{i+1} consisting of the edges $E_{(i+1)1} \cup E_{(i+1)3} \cup E(G'_i)$. So $G_{i+1} = G_{(i+1)1} \cup G_{(i+1)2}$, where both $G_{(i+1)1}$ and $G_{(i+1)2}$ are plane graphs. In $G_{(i+1)1}$ there exists an interior face containing V_{i+1} . Let $V_{(i+1)2}^f$ be the set of vertices in the exterior face of $G_{(i+1)2}$. By Theorem 2, we can transform the planar embedding $G_{(i+1)2}$ into another planar embedding $G'_{(i+1)2}$ such that there exists an interior face, say f , that contains $V_{(i+1)2}^f$. As $V_{(i+1)2}^f$ is also the set of vertices in the exterior face of $G_{(i+1)1}$, so we can attach $G_{(i+1)1}$ in f and obtain a planar embedding G''_{i+1} of G_{i+1} . In G''_{i+1} there exists an interior face containing V_{i+1} . Applying Theorem 2, we get our desired planar embedding G'_{i+1} of G_{i+1} such that V_{i+1} is contained in the exterior face of G'_{i+1} .

Now by the induction hypothesis, $P(m)$ is true, i.e., G_m is planar and it has a planar embedding G'_m such that V_m is contained in the exterior face of G'_m . Now V_m consists of all the vertices on the line L_{2m} . Now $2m = k$ implies that $V_m = V'$. Also $G_m = H$. So H has a planar embedding $H' (= G'_m)$ such that V' is contained in the exterior face of H' . \square

Case 2. L_{k-1} is vertical: Proof of the planarity of G for this case is similar to Case 1. When L_{k-1} is vertical, we partition G into H and K as follows: H and K are the subgraphs of G induced by the vertices lying to the left and right, respectively of the line L_{k-1} . Both H and K must include the vertices on L_{k-1} . Here, we only prove planarity for K . The rest of proof is similar to case 1.

Lemma 9. *K has a planar embedding K' such that V' is contained in the exterior face of K' .*

Proof. Let $V_k^f \subseteq V$ be the set of vertices in G along the exterior face of K . So $V_k^f = (V^f \cap V(K)) \cup V'$. For Type 3, let E_r be the set of vertical edges that have at least one adjacent vertex on the line $e'_3 = \overline{r \text{proj}_{L_{k-1}}(r)}$. In this case, we draw the edges E_r in the exterior faces of K in such a way that we obtain a planar embedding of K . In the embedding, all Steiner points on the line segment $\overline{ru_3}$ will go to the exterior of the polygon along with its adjacent

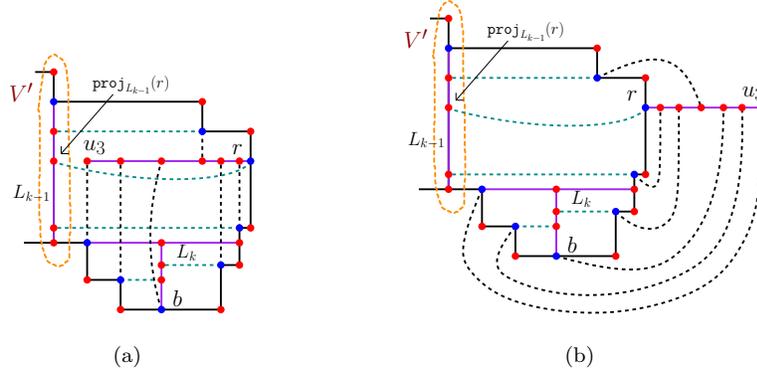


Figure 16: (a) The subgraph K of G for **Type 3** with the exterior face containing V' . (b) Planar embedding of K . The edges of E_r are shown by dashed black segment.

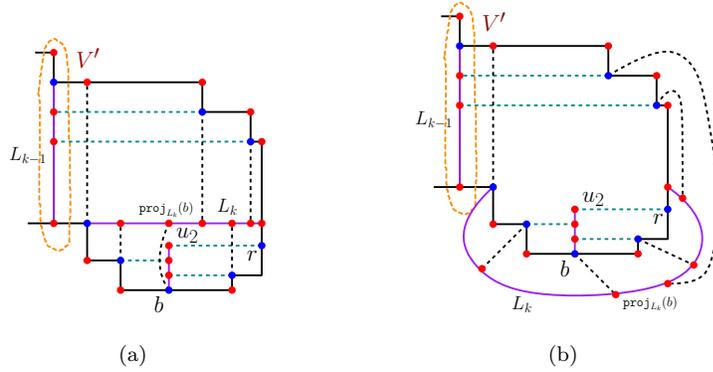


Figure 17: (a) The subgraph K of G for **Type 4** with the exterior face containing V' . (b) Planar embedding of K . The edges of E_k are shown by dashed black segment.

edges (see Figure 16). For Type 4, let E_k be the set of vertical edges that have at least one adjacent vertex on the line L_k . In this case, we draw the edges E_k in the exterior faces of K in such a way that we obtain a planar embedding of K . In the embedding, all Steiner points on the line segment L_k will go to the exterior of the polygon along with its adjacent edges (see Figure 17). In this planar embedding V' still remains in the exterior face. Hence, we get a planar embedding K' of K such that V' is contained in the exterior face of K' .

□

4. Conclusion

In this paper, we construct a **planar Manhattan network** G for a given convex point set S of size n in linear time, where G contains $\mathcal{O}(n)$ Steiner points. Our construction works for more general point set where it is possible to construct an ortho-convex polygon $\mathcal{OCP}(S)$ such that S lies on the boundary of $\mathcal{OCP}(S)$. For example, any convex point set satisfies the aforesaid property. It is also clear that there exists convex point set S for which **planar Manhattan network** G needs $\Omega(n)$ Steiner points. Let $S = \{(1, 1), (2, 2), \dots, (n, n)\}$ be a convex point set of size n . Then S would need $\Omega(n)$ Steiner points. In that sense, our construction is optimal for convex point sets. As a corollary of our construction, for a convex point set, we obtain a $\sqrt{2}$ (~ 1.41) planar spanner in L_2 norm using $\mathcal{O}(n)$ Steiner points. It remains an open question, if it is possible to construct a **planar Manhattan network** for general point sets using subquadratic number of Steiner points.

References

References

- [1] J. Gudmundsson, O. Klein, C. Knauer, M. Smid, Small Manhattan networks and algorithmic applications for the Earth movers distance, in: Proceedings of the 23rd European Workshop on Computational Geometry, 2007, pp. 174–177.
- [2] J. Gudmundsson, C. Levcopoulos, G. Narasimhan, Approximating minimum Manhattan networks, in: Randomization, Approximation, and Combinatorial Optimization. Algorithms and Techniques, Springer, 1999, pp. 28–38.
- [3] G. Narasimhan, M. Smid, Geometric spanner networks, Cambridge University Press, 2007.
- [4] F. Lam, M. Alexandersson, L. Pachter, Picking alignments from (steiner) trees, Journal of Computational Biology 10 (3-4) (2003) 509–520.

- [5] R. Kato, K. Imai, T. Asano, An improved algorithm for the minimum Manhattan network problem, in: International Symposium on Algorithms and Computation, Vol. 2518 of LNCS, Springer, 2002, pp. 344–356.
- [6] S. Seibert, W. Unger, A 1.5-approximation of the minimal Manhattan network problem, in: International Symposium on Algorithms and Computation, Vol. 3827 of LNCS, Springer, 2005, pp. 246–255.
- [7] M. Benkert, A. Wolff, F. Widmann, The minimum Manhattan network problem: a fast factor-3 approximation, in: Japanese Conference on Discrete and Computational Geometry, Springer, 2004, pp. 16–28.
- [8] Z. Guo, H. Sun, H. Zhu, Greedy construction of 2-approximation minimum Manhattan network, in: International Symposium on Algorithms and Computation, Vol. 5369 of LNCS, Springer, 2008, pp. 4–15.
- [9] F. Y. Chin, Z. Guo, H. Sun, Minimum Manhattan network is NP-complete, *Discrete & Computational Geometry* 45 (4) (2011) 701–722.
- [10] C. Knauer, A. Spillner, A fixed-parameter algorithm for the minimum Manhattan network problem., *Journal of Computational Geometry* 2 (1).
- [11] J. M. Keil, C. A. Gutwin, The delaunay triangulation closely approximates the complete euclidean graph, in: *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 1989, pp. 47–56.
- [12] S. Cui, I. A. Kanj, G. Xia, On the stretch factor of delaunay triangulations of points in convex position, *Computational Geometry* 44 (2) (2011) 104–109.
- [13] G. Xia, The stretch factor of the delaunay triangulation is less than 1.998, *SIAM Journal on Computing* 42 (4) (2013) 1620–1659.
- [14] S. Arikati, D. Z. Chen, L. P. Chew, G. Das, M. Smid, C. D. Zaroliagis, Planar spanners and approximate shortest path queries among obstacles in

- the plane, in: European Symposium on Algorithms, Vol. 1136 of LNCS, Springer, 1996, pp. 514–528.
- [15] M. Amani, A. Biniiaz, P. Bose, J.-L. De Carufel, A. Maheshwari, M. Smid, A plane 1.88-spanner for points in convex position, *Journal of Computational Geometry* 7 (1) (2016) 520–539.
- [16] S. Schuierer, An optimal data structure for shortest rectilinear path queries in a simple rectilinear polygon, *International Journal of Computational Geometry & Applications* 6 (02) (1996) 205–225.
- [17] M. De Berg, On rectilinear link distance, *Computational Geometry* 1 (1) (1991) 13–34.
- [18] A. Datta, G. Ramkumar, On some largest empty orthoconvex polygons in a point set, in: *International Conference on Foundations of Software Technology and Theoretical Computer Science*, Vol. 472 of LNCS, Springer, 1990, pp. 270–285.
- [19] V. Chepoi, K. Nouioua, Y. Vaxes, A rounding algorithm for approximating minimum Manhattan networks, *Theoretical Computer Science* 390 (1) (2008) 56–69.
- [20] D. B. West, et al., *Introduction to graph theory*, Vol. 2, Prentice Hall Upper Saddle River, NJ, 1996.
- [21] A. Gibbons, *Algorithmic graph theory*, Cambridge university press, 1985.
- [22] A. García, F. Hurtado, M. Korman, I. Matos, M. Saumell, R. I. Silveira, J. Tejel, C. D. Tóth, Geometric biplane graphs ii: Graph augmentation, *Graphs and Combinatorics* 31 (2) (2015) 427–452.