Greedy Construction of 2-Approximation Minimum Manhattan Network *

Zeyu Guo[1], He Sun[1], Hong Zhu[2]

[1]Fudan University, China
[2]East China Normal University, China

Abstract. Given a set T of n points in $\mathbb{R}^2$, a Manhattan Network G is a network with all its edges horizontal or vertical segments, such that for all $p, q \in T$, in G there exists a path (named a Manhattan path) of the length exactly the Manhattan distance between p and q. The Minimum Manhattan Network problem is to find a Manhattan network of the minimum length, *i.e.*, the total length of the segments of the network is to be minimized. In this paper we present a 2-approximation algorithm with time complexity $O(n \log n)$, which improves the 2-approximation algorithm with time complexity $O(n^2)$. Moreover, compared with other 2-approximation algorithms employing linear programming or dynamic programming technique, it was first discovered that only greedy strategy suffices to get 2-approximation network.

Key words: Minimum Manhattan Network, approximation algorithm, greedy strategy

1 Introduction

A *rectilinear path* between two points $p, q \in \mathbb{R}^2$ is a path connecting p and q with all its edges horizontal or vertical segments. Furthermore, a *Manhattan path* between p and q is a rectilinear path with its length exactly $\text{dist}(p, q) := |p.x - q.x| + |p.y - q.y|$, *i.e.*, the Manhattan distance between p and q.

Given a set T of n points in $\mathbb{R}^2$, a network G is said to be a *Manhattan network* on T, if for all $p, q \in T$ there exists a Manhattan path between p and q with all its segments in G. For the given network G, let the length of G, denoted by $L(G)$, be the total length of all segments of G. For the given point set T, the *Minimum Manhattan Network* (MMN) Problem is to find a Manhattan network G on T with minimum $L(G)$.

From the problem description, it is easy to show that there is a close relationship between the MMN problem and planar t-spanners. For $t \geq 1$, if there exists a planar graph G such that for all $p, q \in T$, there exists a path in G connecting p and q of length at most t times the distance between p and q, G is said to be

* This work is supported by Shanghai Leading Academic Discipline Project(Project Number:B412), National Natural Science Fund (grant #60496321), and the ChunTsung Undergraduate Research Endowment. Correspondence author: He Sun

a *t-spanner* of T. The MMN Problem for T is exactly the problem to compute the 1-spanner of T under the L_1-norm.

Related works: Due to the numerous applications in city planning, network layout, distributed algorithms, and VLSI circuit design, the MMN problem was first introduced by Gudmundsson, Levcopoulos *et al.* [5], and until now, it is open whether this problem belongs to the complexity class P. Gudmundsson *et al.* [5] proposed an $O(n^3)$-time 4-approximation algorithm, and an $O(n \log n)$-time 8-approximation algorithm. Kato, Imai *et al.* [7] presented an $O(n^3)$-time 2-approximation algorithm. However, the proof of their algorithm correctness is incomplete [3]. In spite of that, their paper still provided a valuable idea, that it suffices for G to be a Manhattan network if for each of $O(n)$ certain pairs there exists a Manhattan path connecting its two points. Thus it is not necessary to enumerate all the pairs in $T \times T$. Following this idea, Benkert, Wolff *et al.* [1, 2] proposed an $O(n \log n)$-time 3-approximation algorithm. They also described a mixed-integer programming (MIP) formulation of the MMN problem. After that, Chepoi, Nouioua *et al.* [3] proposed a 2-approximation rounding algorithm by solving the linear programming relaxation of the MIP. In this paper, the notions Pareto Envelope and a nice strip-staircase decomposition has been proposed first of all. In K. Nouioua's Ph.D thesis [8], the primal-dual based algorithm with 2-approximation and running time $O(n \log n)$ has been presented. After these works, it was Z. Guo *et al.* [6] who observed that the same approximation ratio can also be achieved using combinatorial construction. In their paper, the dynamic programming speed-up technique of quadrangle inequality was first used in this problem and, therefore the time complexity $O(n^2)$ has been achieved. In [9], S. Seibert and W. Unger proposed a 1.5-approximation algorithm. However, their proof is incorrect and 2-approximation is, to our best knowledge, the lowest ratio for this problem.

Our contributions: In this paper, we present a very simple 2-approximation algorithm for constructing Manhattan network with running time $O(n \log n)$. Compared with the simple 3-approximation algorithm with running time $O(n \log n)$ proposed recently [4] and the previous 2-approximation result [6] relying on dynamic programming speed-up technique, a highlight in our paper is that, except Pareto Envelope which is widely used in the previous literatures, it is proven simply greedy strategy is enough for constructing 2-approximation Minimum Manhattan Network.

Outline of our approach: From a high-level overview, our algorithm is as follows: partition the input into several blocks (ortho-convex regions) that can be solved independently of each other. For the blocks, some can be trivially solved optimally, whereas only one type of blocks is difficult to solve. For such a non-trivial block there are some horizontal and vertical strips which can be solved by horizontal and vertical nice covers plus switch segments to connect neighboring points in the same strip. In such manner, we divide each block into several staircases. In order to connect the points in each staircase, simple greedy strategy has been used.

2 Preliminaries

Basic notations: For $p = (p.x, p.y) \in \mathbb{R}^2$, let $\mathcal{Q}_k(p)$ denote the k-th closed quadrant with respect to the origin p, e.g., $\mathcal{Q}_1(p) := \{q \in \mathbb{R}^2 \mid p.x \le q.x, p.y \le q.y\}$.

Define $R(p,q)$ as a closed rectangle (possibly degenerate) where $p, q \in \mathbb{R}^2$ are its two opposite corners. $B_V(p,q)$ is defined as the vertical closed band bounded by p, q, whereas $B_H(p,q)$ denotes the horizontal closed band bounded by p, q.

For the given point set T, let Γ be the union of vertical and horizontal lines which pass through some point in T. In addition, we use $[c,d]$ to represent the vertical or horizontal segment with endpoints c and d, as Fig. 1 shows.

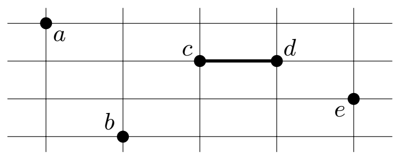

Fig. 1. $T = \{a, b, c, d, e\}$. The vertical and horizontal lines compose Γ.

Pareto envelope: The Pareto envelope, originally proposed by Chepoi *et al.* [3], plays an important role in our algorithm and we give a brief introduction.

Given the set of points T, a point p is said to be *dominated* by q if $\big(\forall t \in T : \mathrm{dist}(q,t) \le \mathrm{dist}(p,t)\big) \wedge \big(\exists t \in T : \mathrm{dist}(q,t) < \mathrm{dist}(p,t)\big)$. A point is said to be an *efficient point* if it is not dominated by any point in the plane. The *Pareto envelope* of T is the set of all efficient points, denoted by $\mathcal{P}(T)$. Fig. 2 shows an example of $\mathcal{P}(T)$. It is not hard to prove that $\mathcal{P}(T) = \bigcap_{u \in T} \bigcup_{v \in T} R(u,v)$. For $|T| = n$, $\mathcal{P}(T)$ can be built in $O(n \log n)$ time. [3] also presented some other properties of $\mathcal{P}(T)$. In particular, $\mathcal{P}(T)$ is *ortho-convex*, *i.e.*, the intersection of $\mathcal{P}(T)$ with any vertical or horizontal line is continuous, which is equivalent to the fact that for any two points $p, q \in \mathcal{P}(T)$, there exists a Manhattan path in $\mathcal{P}(T)$ between p and q.

In [3] Chepoi *et al.* also showed that the Pareto envelope is the union of some ortho-convex (possibly degenerate) rectilinear polygons (called *blocks*). Two blocks can overlap at only one point which is called a *cut vertex*. We denote by C the set of cut vertices, and let $T^+ := T \cup C$. For a block B, denote by H_B and W_B its height and width respectively. Let $T_B := T^+ \cap B$. We say B is *trivial* if B is a rectangle (or degenerate to a segment) such that $|T_B| = 2$. It is known that the two points in T_B must be two opposite corners of B when it is trivial. In Fig. 2, $C = \{a, b, c, d\}$ and only the block between c and d is non-trivial.

Chepoi *et al.* [3] proved that an MMN on T^+ is also an MMN on T, and to obtain an MMN on T^+, it suffices to build an MMN on T_B for each $B \subseteq \mathcal{P}(T)$. The MMN in any trivial block B can be built by simply connecting the two points in T_B using a Manhattan path. So we have reduced the MMN problem on T to MMN on non-trivial blocks.

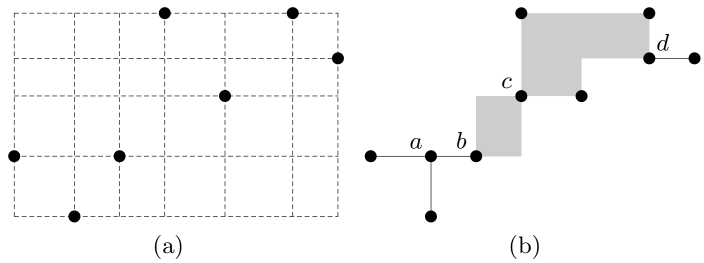

Fig. 2. An example of a Pareto envelope. The black points in (a) are the set T. The two separate grey regions in (b) are non-degenerate blocks, whereas the black lines are degenerate blocks. All these blocks form the Pareto envelope $\mathcal{P}(T)$.

For a non-trivial block B denote its border by ∂B and let $\Gamma_B := \Gamma \cap B$. We call a corner p in ∂B a *convex corner* if the interior angle at p equals to $\pi/2$, otherwise p is called a *concave corner*.

Lemma 1. *[3] For any non-trivial block B and any convex corner p in ∂B, it holds that $p \in T_B$.*

Lemma 2. *[3] For any non-trivial block B, there exists an MMN G_B on T_B such that $G_B \subseteq \Gamma_B$. Furthermore, any MMN $G_B \subseteq \Gamma_B$ on T_B contains ∂B.*

Strips and staircase components: Informally, for $p, q \in T_B, p.y < q.y$, we call $R(p,q)$ a *vertical strip* if it does not contain any point of T_B in the region $B_V(p,q)$ except the vertical lines $\{(x,y)|x = p.x, y \leq p.y\}$ and $\{(x,y)|x = q.x, y \geq q.y\}$. Similarly, for the points $p, q \in T_B, p.x < q.x$, we call $R(p,q)$ a *horizontal strip* if it does not contain any point in the region $B_H(p,q)$ except the horizontal lines $\{(x,y)|x \leq p.x, y = p.y\}$ and $\{(x,y)|x \geq q.x, y = q.y\}$. Especially, we say a vertical or horizontal strip $R(p,q)$ is *degenerate* if $p.x = q.x$ or $p.y = q.y$. Fig. 3 gives an example of a horizontal strip.

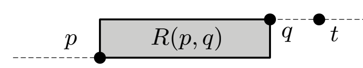

Fig. 3. The rectangle is a horizontal strip. Any point in T_B within $B_H(p,q)$ can only be placed on the dashed lines, *e.g.*, the point t.

The other notion which plays a critical role in our algorithm is the staircase component. There are four kinds of staircase components specified by four quadrants, and without loss of generality we only describe the one with respect to the third quadrant. Suppose $R(p,q)$ is a vertical strip and $R(p',q')$ is a horizontal strip, such that $q \in \mathcal{Q}_1(p)$, $q' \in \mathcal{Q}_1(p')$, $p, q \in B_V(p',q')$, $p', q' \in B_H(p,q)$, *i.e.*, they cross in the way as Fig. 4 shows. Denote by $T_{pp'|qq'}$ the set of any point $v \in T_B$ such that $v.x > q.x, v.y > q'.y$, where p is the leftmost point and p' is the topmost point in $\mathcal{Q}_3(v)$ besides v. A non-empty $T_{pp'|qq'}$ is said to be a *staircase component* (see Fig. 4). In this figure, no point in T_B is located in the dark grey area and the two light grey unbounded regions except those in $T_{pp'|qq'}$.

For a strip $R(p,q)$, (p,q) is called a *strip pair*. For each staircase component $T_{pp'|qq'}$ and each point v in $T_{pp'|qq'}$, (v,p) (also (v,p')) is called a *staircase pair*.

Theorem 1. *[3] A network G_B is a Manhattan network on T_B if and only if for any strip pair or staircase pair (p,q), $p,q \in T_B$, there exists a Manhattan path in G_B connecting p and q.*

3 Algorithm Description

Following the approach of [1], a union of vertical segments C_V is said to be *a vertical cover* if for any horizontal line ℓ and any vertical strip R that ℓ intersects, it holds that $\ell \cap R \cap C_V \neq \emptyset$. Similarly, a union of horizontal segments C_H is said to be *a horizontal cover* if for any vertical line ℓ and any horizontal strip R that ℓ intersects, it holds that $\ell \cap R \cap C_H \neq \emptyset$. Furthermore, a *nice vertical cover* (NVC) is a vertical cover such that any of its segments contains at least one point of T_B. A *nice horizontal cover* (NHC) is defined symmetrically. Fig. 5 shows an NVC.

For an NVC C_V, obviously $[p,q] \subseteq C_V$ for every degenerate vertical strip $R(p,q)$. Assume $R(p,q)$ is a non-degenerate vertical strip where $p.y < q.y$, then there exists vertical segments $[p,p']$ and $[q,q']$ in C_V where $p'.y \geq q'.y$ (it is possible that $p = p'$ or $q = q'$), as Fig. 6 shows. Obviously, a Manhattan path connecting p and q can be built by adding a horizontal segment $[u,v]$ where $u.x = p.x, v.x = q.x, q'.y \leq u.y = v.y \leq p'.y$. Such a segment $[u,v]$ is said to be a *switch segment* of $R(p,q)$. The same concept for NHC is defined symmetrically.

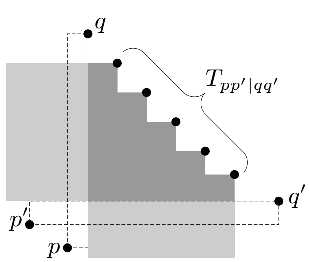

Fig. 4. A staircase component

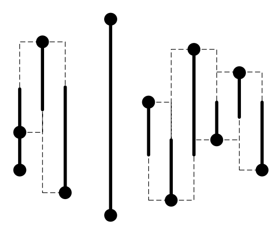

Fig. 5. An NVC consisting of black lines

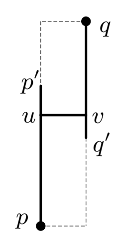

Fig. 6. A switch segment

Now we present an iterative algorithm `CreateNVC` to construct an NVC. In the initialization step, let C_V be the union of segments $[p, q]$ for each degenerate vertical strip $R(p, q)$, whereas N is the set of non-degenerate ones. In addition, let the set X be T_B. The main part of the procedure consists of two loops. Regarding the first loop, a vertical segment of ∂B in some $R(p, q) \in N$ is chosen in each round. Lemma 1 and the definition of strips guarantee that such segments must be connected to some point in X. Let the segment lying in the non-degenerate strip $R(p, q)$ be $[p, p']$, as Fig. 7 shows. Then p' is added to X, and $[p, p']$ is added to C_V. And by invoking `Update`(p'), N is updated to be the set of non-degenerate strips when the new set X is considered as the input point set. Define $\bigcup N := \bigcup_{R(p,q)\in N} R(p, q)$. It is easy to see that the part of $\bigcup N$ adjacent to $[p, p']$ is eliminated in each round, which turns out that $\bigcup N$ becomes smaller. It can be demonstrated that $[p, p']$ is the unique vertical segment excluded from $\bigcup N$ in ∂B. We repeat the operations above until all the vertical segments initially falling in $\partial B \cap \bigcup N$ are excluded from $\bigcup N$ and added to C_V.

In the second loop, we choose $R(p, q) \in N$ arbitrarily and both its left and right edges are added to C_V. Two points $(p.x, q.y), (q.x, p.y)$ are added to X. And N is updated in the similar manner as Fig. 8 shows. The formal description is as follows.

Input: T_B
$C_V \leftarrow \bigcup [p, q]$ where $R(p, q)$ is a degenerate vertical strip;
$X \leftarrow T_B$;
$N \leftarrow \{R(p, q) \mid R(p, q)$ is a non-degenerate vertical strip$\}$;
while *there exists a vertical segment* $[p, p'] \subseteq \partial B \cap R(p, q)$*, where* $R(p, q) \in \mathcal{N}$ **do**
    $X \leftarrow X \cup \{p'\}$;
    $C_V \leftarrow C_V \cup [p, p']$;
    $Update(p')$;
end
while $N \neq \emptyset$ **do**
    Let $R(p, q)$ be an arbitrary vertical strip in N;
    $p' \leftarrow (p.x, q.y)$; $q' \leftarrow (q.x, p.y)$;
    $X \leftarrow X \cup \{p', q'\}$;
    $C_V \leftarrow C_V \cup [p, p'] \cup [q, q']$;
    $Update(p')$; $Update(q')$;
end

Algorithm 1: `CreateNVC`

Lemma 3. `CreateNVC` *takes $O(n)$ time to output an NVC C_V.*

Proof. Since C_V initially contains $[p, q]$ for any degenerate vertical strip $R(p, q)$, a horizontal line ℓ that crosses $R(p, q)$ always intersects C_V. Therefore we only need to consider non-degenerate vertical strips.

```
    Input: v
1   for each R(p,q) ∈ N such that v.x = p.x, [p,v] ∩ R(p,q) ≠ {p} do
2       N ← N\{R(p,q)};
3       if v ∈ R(p,q) and v.y ≠ q.y then N ← N ∪ {R(v,q)};
4   end
```

Algorithm 2: `Update`

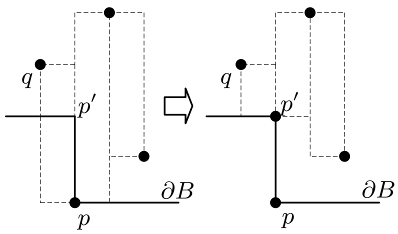

Fig. 7. The change in the first loop

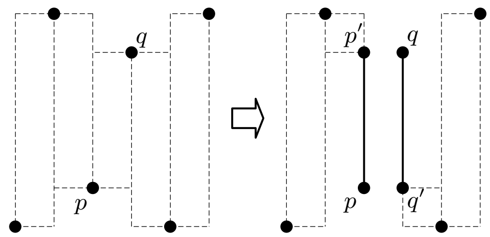

Fig. 8. The change in the second loop

We prove the following invariant maintains: let $R(p,q)$ be a vertical strip in the original N and ℓ be a horizontal line that intersects $R(p,q)$, then at any stage of the algorithm, either $\ell \cap R(p,q) \cap C_V \neq \emptyset$ or $\ell \cap R(p,q) \subseteq \bigcup N$ holds.

At the beginning obviously $\ell \cap R(p,q) \subseteq \bigcup N$ holds. Each time when N is updated, the part of $R(p,q)$ eliminated from $\bigcup N$ (if existing) must be adjacent to some segment which is added to C_V, so either $\ell \cap R(p,q) \cap C_V \neq \emptyset$ or $\ell \cap R(p,q) \subseteq \bigcup N$ still holds for the updated N. The set N will be updated iteratively until $\bigcup N = N = \emptyset$, which implies $\ell \cap R(p,q) \cap C_V \neq \emptyset$.

Secondly, we consider the running time of the procedure.

Line 1 takes $O(n)$ time since $O(n)$ degenerate vertical strips exist. Initially N contains $O(n)$ non-degenerate vertical strips and $\bigcup N$ contains $O(n)$ vertical segments of ∂B. The first loop reduces one such vertical segment in each round, whereas the second loop eliminates at least one strip in N in each round. Moreover, each invoking of the procedure `Update` takes $O(1)$ time since when a point is added to X, $O(1)$ strips need to be removed or replaced. Therefore the overall time complexity is $O(n)$. $\square$

After invoking `CreateNVC`, we add the topmost and bottommost switch segments for each non-degenerate vertical strip, as Fig. 9 shows. Then for each vertical strip $R(p,q)$, at least one Manhattan path between p and q is built. Symmetrically, we can use the algorithm `CreateNHC` to compute an NHC. Furthermore, for each horizontal strip, the leftmost and the rightmost switch segments are added. All these procedures guarantee that the Manhattan paths for all the strip pairs have been constructed.

Now we turn to the discussion of staircases. For simplicity, we only describe the definition of the staircase with respect to the third quadrant. The other cases are symmetric.

Definition 1 (staircase). *For a staircase component $T_{pp'|qq'}$ with respect to the third quadrant, assume $R(p, q)$ is a vertical strip and $R(p', q')$ is a horizontal strip. Let M_{pq} be the Manhattan path between p and q which passes through the bottommost switch segment. Let $M_{p'q'}$ be the Manhattan path between p' and q' which passes through the leftmost switch segment. The part of $\bigcup_{v \in T_{pp'|qq'}} \mathcal{Q}_3(v)$ bounded by M_{pq} and $M_{p'q'}$, excluding $M_{pq}, M_{p'q'}, C_V, C_H$ is said to be a staircase, denoted by $S_{pp'|qq'}$.*

Fig. 10 gives an example of staircase.

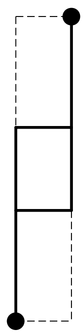

Fig. 9. Adding switch segments

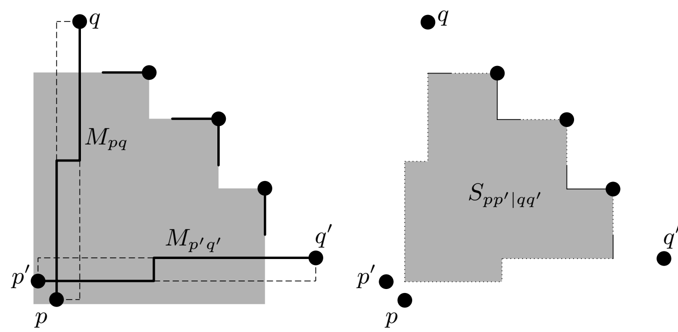

Fig. 10. The definition of the staircase $S_{pp'|qq'}$. The dotted lines in the right picture is not included in $S_{pp'|qq'}$.

Lemma 4. *There exists a procedure* `CreateStaircasePath` *such that for the given staircase $S_{pp'|qq'}$ with the staircase component $T_{pp'|qq'}$, $|T_{pp'|qq'}| = n$, the procedure takes $O(n \log n)$ time to construct a network $G_{pp'|qq'} \subseteq S_{pp'|qq'}$ such that $G_{pp'|qq'} \cup C_H \cup C_V$ connects each point in $T_{pp'|qq'}$ to either M_{pq} or $M_{p'q'}$.*

Proof. Without loss of generality, assume $R(p, q)$ is a vertical strip and $R(p', q')$ is a horizontal strip where $q \in \mathcal{Q}_1(p)$, $q' \in \mathcal{Q}_1(p')$, as Fig. 10 shows. Let $t_0 := q, t_{n+1} := q'$. Express the points in $T_{pp'|qq'}$ as $t_1, t_2, \cdots, t_n$ in the order from the topmost and leftmost one to the bottommost and rightmost one.

For $1 \leq i \leq n$, define the horizontal segment $h_i := \{(x, y) \mid y = t_i.y\} \cap S_{pp'|qq'}$ and the vertical segment $v_i := \{(x, y) \mid x = t_i.x\} \cap S_{pp'|qq'}$, as Fig 11 shows. We use $\mathrm{Right}_i(S_{pp'|qq'})$ to represent the staircase polygon on the right of v_i whereas $\mathrm{Top}_i(S_{pp'|qq'})$ represents the one on the top of h_i. Note that $\mathrm{Right}_i(S_{pp'|qq'})$ and $\mathrm{Top}_i(S_{pp'|qq'})$ are all smaller staircase polygons. Assume S is a general staircase polygon in $S_{pp'|qq'}$. Let $h_i^S := h_i \cap S, v_i^S := v_i \cap S$. It can be observed that $\langle h_i^S \rangle$ is ascending whereas $\langle v_i^S \rangle$ is descending. Define $\mathrm{Right}_i(S)$ and $\mathrm{Top}_i(S)$ in a similar way. The partial network $G_{pp'|qq'} \cap S$ is constructed in a recursive manner.

Initially we invoke `CreateStaircasePath`$(S_{pp'|qq'})$. For any non-empty S one of the three branches is chosen. In the third case, binary search guarantees the

Input: S

if $S = \emptyset$ **then** return $\emptyset$;

else if $L(h_1^S) \geq L(v_1^S)$ **then** return $v_1^S \cup$ `CreateStaircasePath`$(\text{Right}_1(S))$;

else if $L(h_n^S) \leq L(v_n^S)$ **then** return $h_n^S \cup$ `CreateStaircasePath`$(\text{Top}_n(S))$;

else

  Choose k such that $L(h_k^S) \leq L(v_k^S)$ and $L(h_{k+1}^S) \geq L(v_{k+1}^S)$;

  return $h_k^S \cup v_{k+1}^S \cup$ `CreateStaircasePath`$(\text{Top}_k(S)) \cup$ `CreateStaircasePath`$(\text{Right}_{k+1}(S))$;

end

Algorithm 3: `CreateStaircasePath`

proper k can be obtained with running time $O(\log n)$ whereas the procedure is invoked recursively at most $O(n)$ times, which results in the total running time $O(n \log n)$.

The correctness proof simply follows from the induction method. □

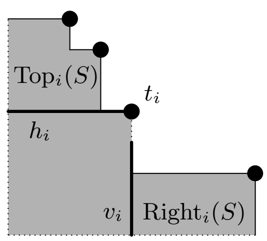

Fig. 11. The definition of h_i and v_i

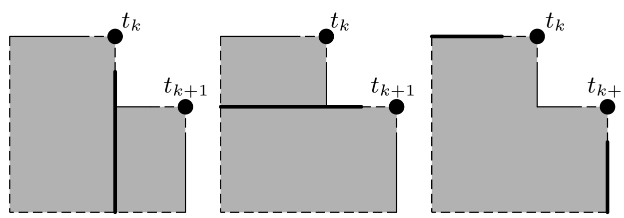

Fig. 12. One of the three connections for t_k and t_{k+1} is optimal

In the following, we present the global algorithm `CreateMMN`.

Theorem 2. *For the given point set T of size n,* `CreateMMN` *takes $O(n \log n)$ time to compute a Manhattan network G on T.*

Proof. For any non-trivial block, NVC, NHC and switch segments form the Manhattan paths for strip pairs, whereas some segments are added in staircases such that there exist Manhattan paths for staircase pairs. By Theorem 1, the final network is a Manhattan network.

Regarding the running time, it is well-known that computing the Pareto envelope and constructing the networks in staircases can be implemented in $O(n \log n)$ time, and the time required for decomposing each block into staircases and strips is also $O(n \log n)$ using the method similar to [1]. The other steps, including computing NVC, NHC and adding switch segments, can be implemented in linear time. Thus the overall time complexity is $O(n \log n)$. □

Input: T
Compute $\mathcal{P}(T)$.
for *each trivial block* $B \subseteq \mathcal{P}(T)$ **do**
  connect the two points in T_B with a Manhattan path.
for *each non-trivial block* $B \subseteq \mathcal{P}(T)$ **do**
  `CreateNVC`;
  for *each vertical strip* $R(p, q)$ **do**
    add the topmost and bottommost switch segments of $R(p, q)$;
  `CreateNHC`;
  for *each horizontal strip* $R(p, q)$ **do**
    add the leftmost and rightmost switch segments of $R(p, q)$;
  for *each staircase* $S_{pp'|qq'}$ **do** `CreateStaircasePath`($S_{pp'|qq'}$);
end

Algorithm 4: `CreateMMN`

4 Approximation Analysis

The rest of this paper is devoted to the approximation analysis of this problem. Let G denote the Manhattan network constructed by our algorithm, whereas $G^\star$ is the optimal one demonstrated by Lemma 2 with the property that $\partial B \subseteq G^\star \cap B \subseteq \Gamma_B$ for every non-trivial block B. For any block B, let $G_B := G \cap B$ and $G^\star_B := G^\star \cap B$.

Let B be a non-trivial block. Denote by G_S the switch segments our algorithm adds when computing G_B. Let $\mathcal{S} := \bigcup S_{pp'|qq'}, G_U := G_B \cap \mathcal{S}$. From the algorithm description obviously $G_B := C_V \cup C_H \cup G_S \cup G_U$.

Let $G^\star_C := G^\star_B \cap (C_V \cup C_H)$, whereas $G^\star_U := G^\star_B \cap \mathcal{S}$.

Lemma 5. *$L(C_V \cup C_H) \le 2L(G^\star_C) - 2H_B - 2W_B$.*

Proof. We divide $C_V \cup C_H$ into two parts: let C_1 be the set of segments for each degenerate vertical and horizontal strip, as well as the segments added in the first loop of procedures `CreateNVC` and `CreateNHC`. Let C_2 represent the union of the segments added in the second loop. In addition, denote $C^\star_1 := G^\star_C \cap C_1$, and $C^\star_2 := G^\star_C \cap C_2$.

Observing that C_1 is the union of the segments in degenerate strips and ∂B, it is easy to show that $\partial B \subseteq C_1 = C^\star_1$. Therefore $L(C_1) \le 2L(C^\star_1) - L(\partial B) = 2L(C^\star_1) - 2H_B - 2W_B$.

On the other hand, let us consider the second loop of the procedure `CreateNVC` and `CreateNHC`. By symmetric property, we only analyze the procedure `CreateNVC`. In a round, two segments $[p, p']$, $[q, q']$ of length ℓ are added into C_V. By our algorithm, $R(p, q)$ is contained in some vertical strip $R(s, t)$. Since $G^\star_B$ is a Manhattan network, $C^\star_2 \cap ([p, p'] \cup [q, q'])$ contains segments of length at least ℓ to connect s and t. Since the relation holds for each round and also the procedure `CreateNHC`, we obtain $L(C_2) \le 2L(C^\star_2)$.

Combining the two inequalities above, we obtain the lemma. □

For any staircase $S_{pp'|qq'}$, let $G^\star_{pp'|qq'} := G^\star_U \cap S_{pp'|qq'}$.

Lemma 6. *For any staircase $S_{pp'|qq'}$, it holds $L(G_{pp'|qq'}) \leq 2L(G^\star_{pp'|qq'})$.*

Proof. Without loss of generality, let $S_{pp'|qq'}$ be a staircase with respect to the third quadrant, as Fig. 10 shows. Let S be a staircase polygon in $S_{pp'|qq'}$ such that `CreateStaircasePath`(S) is invoked. We will prove $L\left(G_{pp'|qq'} \cap S\right) \leq 2L\left(G^\star_{pp'|qq'} \cap S\right)$ using induction.

The inequality obviously holds in the trivial case $S = \emptyset$. Assume the relation holds for smaller staircase polygons in S. For the case $L\left(h_1^S\right) \geq L\left(v_1^S\right)$, t_1 is connected down and the original problem is reduced to the small one with region $\mathrm{Right}_1(S)$. Let $S_R := S \backslash \mathrm{Right}_1(S)$. By assumption, we only need to prove $L\left(G_{pp'|qq'} \cap S_R\right) \leq 2L\left(G^\star_{pp'|qq'} \cap S_R\right)$. Note that $L\left(G_{pp'|qq'} \cap S_R\right) = L\left(v_1^S\right)$, and in $G_{pp'|qq'} \cap S_R$ segments of length $\min\left\{L\left(h_1^S\right), L\left(v_1^S\right)\right\} = L\left(v_1^S\right)$ is necessary to connect t_1 to either the left or the bottom boundary of S. Thus the relation holds. The analysis for the case $L\left(h_n^S\right) \leq L\left(v_n^S\right)$ is analogous.

Regarding the last case, let $S_R := S \backslash \left(\mathrm{Top}_k(S) \cup \mathrm{Right}_{k+1}(S)\right)$. We only need to prove $L\left(G_{pp'|qq'} \cap S_R\right) \leq 2L\left(G^\star_{pp'|qq'} \cap S_R\right)$. As Fig. 12 shows, segments of length at least $\min\left\{L\left(v_k^S\right), L\left(h_{k+1}^S\right), L\left(h_k^S\right) + L\left(v_{k+1}^S\right)\right\}$ are necessary to connect t_k and t_{k+1} to either the left or the bottom boundary. By monotonicity, $L\left(v_{k+1}^S\right) \leq L\left(v_k^S\right)$ and $L\left(h_k^S\right) \leq L\left(h_{k+1}^S\right)$. By the choice of k, we obtain $L\left(v_{k+1}^S\right) \leq L\left(h_k^S\right)$ and $L\left(h_k^S\right) \leq L\left(v_{k+1}^S\right)$. Therefore $L\left(G_{pp'|qq'} \cap S_R\right) = L\left(h_k^S\right) + L\left(v_{k+1}^S\right) \leq 2L\left(G^\star_{pp'|qq'} \cap S_R\right)$. $\square$

Now we estimate $L(G_U)$. Note that it is possible that some segments of $G^\star_U$ lie in two different staircases. Let $G^\star_D$ denote the union of these segments. Fig. 13 illustrates this special condition.

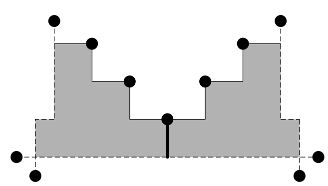

Fig. 13. The segment lying in two different staircases

Lemma 7. $L(G_U) \leq 2L(G^\star_U) + 2L(G^\star_D)$.

Proof. Since the segments of $G^\star_D$ are counted twice, we obtain that $L(G_U) \leq \sum L(G_{pp'|qq'}) \leq 2\sum L(G^\star_{pp'|qq'}) \leq 2L(G^\star_U) + 2L(G^\star_D)$. $\square$

Lemma 8. $2L(G^\star_D) + L(G_S) \leq 2H_B + 2W_B$.

Proof. The lemma can be obtained by the following fact: let ℓ be a vertical or horizontal line such that $\ell \not\subseteq \Gamma$, then ℓ may cross at most one segment in $G^\star_D$, and at most two segments in G_S. Furthermore, due to the definitions of strips and staircase components, ℓ cannot intersect both of $G^\star_D$ and G_S. We omit the details here. $\square$

Theorem 3. *For any block B, $L(G_B) \le 2L(G^\star_B)$.*

Proof. For any trivial block B, the relation obviously holds since $L(G_B) = H_B + W_B$. Let B be a non-trivial block, $L(G_B) \le L(C_V \cap C_H) + L(G_S) + L(G_U) \le 2L(G^\star_C) + 2L(G^\star_U) + 2L(G^\star_D) + L(G_S) - 2H_B - 2W_B \le 2L(G^\star_C) + 2L(G^\star_U)$.

Recall that $G^\star_C = G^\star_B \cap (C_V \cup C_H), G^\star_U = G^\star_B \cap S_U$. By the definition of staircases, it holds that $(C_V \cup C_H) \cap S_U = \emptyset$. This means $G^\star_C$ and $G^\star_U$ are disjoint parts of $G^\star_B$. Therefore $L(G_B) \le 2L(G^\star_C) + 2L(G^\star_U) \le 2L(G^\star_B)$. $\square$

Corollary 1. $L(G) \le 2L(G^\star)$. $\square$

References

1. M. Benkert, A. Wolff and F. Widmann. The minimum Manhattan network problem: a fast factor-3 approximation. Technical Report 2004-16, Fakultät für Informatik, Universität Karlsruhe. A short version appeared in *Proceedings of the 8th Japanese Conference on Discrete and Computational Geometry*, 2005, pages 16-28.
2. M. Benkert, T. Shirabe, and A. Wolff. The minimum Manhattan network problem: approximations and exact solution. In *Proceedings of the 20th European Workshop on Computational Geometry*, 2004, pages 209-212.
3. V. Chepoi, K. Nouioua, and Y. Vaxès. A rounding algorithm for approximating minimum Manhattan networks. *Theoretical Computer Science*, 390(2008):56-69. Preliminary version appeared in *Proceedings of the 8th International Workshop on Approximation Algorithms for Combinatorial Optimization*, 2005, pages 40-51.
4. B. Fuchs and A.Schulze. A simple 3-approximation of minimum Manhattan networks. Technical Report, 2008. Available at `http://www.zaik.uni-koeln.de/~paper/unzip.html?file=zaik2008-570.pdf`
5. J. Gudmundsson, C. Levcopoulos, and G. Narasimhan. Approximating a minimum Manhattan network. *Nordic Journal of Computing*, 8(2001):219-232. Preliminary version appeared in *Proceedings of the 2nd International Workshop on Approximation Algorithms for Combinatorial Optimization*, 1999, pages 28-37.
6. Z. Guo, H. Sun, and H. Zhu. A fast 2-approximation algorithm for the minimum Manhattan network Problem. In *Proceedings of 4th International Conference on Algorithmic Aspect in Information Management*, pages 212-223, 2008.
7. R. Kato, K. Imai, and T. Asano. An improved algorithm for the minimum Manhattan network problem. In *Proceedings of the 13th International Symposium on Algorithms and Computation*, 2002, pages 344-356.
8. K. Nouioua. Enveloppes de Pareto et Réseaux de Manhattan: Caractérisations et algorithmes, Ph.D. thesis, Université de la Méditerranée, 2005.
9. S. Seibert and W. Unger. A 1.5-approximation of the minimal Manhattan network problem. In *Proceedings of the 16th International Symposium on Algorithms and Computation*, 2005, pages 246-255.