The Minimum Manhattan Network Problem: A Fast Factor-3 Approximation

Marc Benkert* Florian Widmann* Alexander Wolff*

For many applications it is desirable to connect the nodes of a transportation or communication network by short paths within the network. In the Euclidean plane this can be achieved by connecting all pairs of nodes by straight line segments. While the complete graph minimizes node-to-node travel time, it maximizes the network-construction costs. An interesting alternative are Euclidean *t-spanners*, i.e. networks in which the ratio of the network distance and the Euclidean distance between any pair of nodes is bounded by a constant $t \geq 1$. Other desirable properties are small node degree, total edge length, and diameter. Euclidean spanners with one or more of these properties can be constructed in $O(n \log n)$ time [1] (n the number of nodes) and have been studied extensively.

Under the Euclidean metric, in a 1-spanner (which is the complete graph) the location of each edge is uniquely determined. This is not the case in the Manhattan (or L_1-) metric, where an edge $\{p, q\}$ of a 1-spanner is a Manhattan p–q path, i.e. a staircase path between p and q. A 1-spanner under the Manhattan metric for a set $P \subset \mathbb{R}^2$ is called a *Manhattan network* and can be seen as a set of axis-parallel line segments whose union contains a Manhattan p–q path for each pair $\{p, q\} \in \binom{P}{2}$.

In this paper we investigate how the extra degree of freedom in routing edges can be used to construct Manhattan networks of minimum total length, so-called *minimum Manhattan networks* (MMN). The MMN problem may have applications in city planning or VLSI layout. It has been considered before, but until now, its complexity status is not known. Gudmundsson et al. [2] have proposed an $O(n \log n)$-time factor-8 and an $O(n^3)$-time factor-4 approximation algorithm. Later Kato et al. [3] have given an $O(n^3)$-time factor-2 approximation algorithm. However, their correctness proof is incomplete.

In this paper we present an $O(n \log n)$-time factor-3 approximation algorithm. We use some of the ideas of [3], but our algorithm is simpler, faster and uses only linear (instead of quadratic) storage. The main novelty of our approach is that we partition the plane into two regions and compare the network computed by our algorithm to an MMN in each region separately.

We use the idea of *generating sets* [3]. A generating set is a subset of $\binom{P}{2}$ with the property that a network containing Manhattan paths for all pairs in the subset is a Manhattan network of P. A generating set Z with $|Z| = O(n)$ is known [3]. We write Z as the union of three sets Z_{hor}, Z_{ver}, and Z_{quad}. Our algorithm first establishes paths for all pairs in $Z_{\text{hor}} \cup Z_{\text{ver}}$, then for those in Z_{quad}.

For the sake of brevity we assume in this abstract that no two input points have the same x- or y-coordinate. Of course our algorithm works for general point sets. Under the above assumption the set Z_{ver} consists of all pairs of points that are neighbors in the lexicographic order. The definition of Z_{hor} is analogous to that of Z_{ver} with the roles of x and y exchanged. We now define Z_{quad}:

Definition 1 *For a point $r \in \mathbb{R}^2$ denote its Cartesian coordinates by (x_r, y_r). Let $Q(r, 1) = \{s \in \mathbb{R}^2 \mid x_r \leq x_s \text{ and } y_r \leq y_s\}$ be the first quadrant of the Cartesian coordinate system with origin r. Define $Q(r, 2)$, $Q(r, 3)$, $Q(r, 4)$ analogously and in the usual order. Then Z_{quad} is the set of all pairs (p, q) with $p, q \in P$ and $q \in Q(p, t)$ for some $t \in \{1, 2, 3, 4\}$ that fulfill*

(a) q is the point that has minimum y-distance from p among all points in $Q(p, t) \cap P$ with minimum x-distance from p, and

(b) there is no $q' \in Q(p, t) \cap P$ with $\{p, q'\} \in Z_{\text{hor}} \cup Z_{\text{ver}}$.

While the pairs in Z are 2-sets, it helps to view those in the subset Z_{quad} as ordered pairs. We need a few simple definitions. Let $\mathcal{R}_{\text{ver}} = \{\text{BBox}(p, q) \mid \{p, q\} \in Z_{\text{ver}}\}$ (see rectangles in Fig. 1), where $\text{BBox}(p, q)$ is the smallest axis-parallel closed rectangle that contains p and q. Define $\mathcal{R}_{\text{hor}}$ and $\mathcal{R}_{\text{quad}}$ analogously. Let $\mathcal{A}_{\text{ver}}$, $\mathcal{A}_{\text{hor}}$, and $\mathcal{A}_{\text{quad}}$ be the subsets of the plane that are defined by the union of the rectangles in $\mathcal{R}_{\text{ver}}$, $\mathcal{R}_{\text{hor}}$, and $\mathcal{R}_{\text{quad}}$, respectively.

For a horizontal line ℓ consider the graph $G_\ell(V_\ell, E_\ell)$, where V_ℓ is the intersection of ℓ with the vertical edges of rectangles in $\mathcal{R}_{\text{ver}}$, and there is an edge in E_ℓ if two intersection points belong to the same rectangle. We say that a point v in V_ℓ is *odd* if

*Faculty of Computer Science, Karlsruhe University, P.O. Box 6980, D-76128 Karlsruhe, Germany. WWW: `i11www.ilkd.uka.de/people/`, Email: `firstname.lastname@ilkd.uka.de`

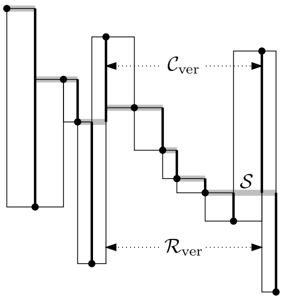

Figure 1: $\mathcal{R}_{\mathrm{ver}}$, $\mathcal{C}_{\mathrm{ver}}$, $\mathcal{S}$.

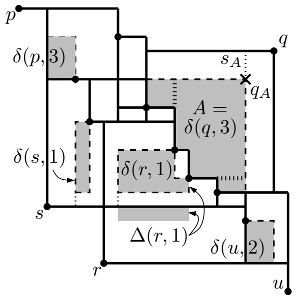

Figure 2: Network N.

the number of points to the left of v that belong to the same connected component of G_ℓ is odd, otherwise v is *even*. For a vertical line g with $g \cap P \neq \emptyset$ let an *odd segment* be an inclusion-maximal connected set of odd points on g. We show that the set $\mathcal{C}_{\mathrm{ver}}$ of all odd segments (bold black in Fig. 1) is a *vertical cover* [3]: for each $R \in \mathcal{R}_{\mathrm{ver}}$ and each horizontal line h that intersects R there is a segment $s \in \mathcal{C}_{\mathrm{ver}}$ with $s \cap h \cap R \neq \emptyset$. The set $\mathcal{C}_{\mathrm{hor}}$ defined analogously is a *horizontal cover*, and thus by [3, Lemma 2]:

Lemma 1 *The total length of* $\mathcal{C} = \mathcal{C}_{\mathrm{ver}} \cup \mathcal{C}_{\mathrm{hor}}$ *is bounded by the length of an* MMN.

The set $\mathcal{C}$ can be computed in $O(n \log n)$ time by a plane sweep. We say that a cover is *nice* if each cover segment contains an input point. We show that $\mathcal{C}$ is nice.

Our algorithm proceeds in three phases. In phase I we compute the generating set Z. In phase II we connect all pairs in $Z_{\mathrm{ver}} \cup Z_{\mathrm{hor}}$ by computing $\mathcal{C}$ and by then adding at most one additional line segment for each rectangle in $\mathcal{R}_{\mathrm{ver}} \cup \mathcal{R}_{\mathrm{hor}}$. Since each rectangle $R = \mathrm{BBox}(p,q) \in \mathcal{R}_{\mathrm{ver}}$ ($\mathcal{R}_{\mathrm{hor}}$) is covered nicely, it suffices to add a horizontal (vertical) segment whose length is the width (height) of R in order to connect p and q by a Manhattan path. Let $\mathcal{S}$ be the set of these additional segments (gray in Fig. 1). Consider the vertical strip that is defined by a rectangle $R \in \mathcal{R}_{\mathrm{ver}}$. By definition of $\mathcal{R}_{\mathrm{ver}}$, R is the only rectangle in $\mathcal{R}_{\mathrm{ver}}$ that intersects the interior of the strip. Thus the total length of the additional horizontal (vertical) segments is the width W (height H) of $\mathrm{BBox}(P)$. By Lemma 1 the network $N_1 = \mathcal{C} \cup \mathcal{S}$ (solid in Fig. 2) has length at most $|N_{\mathrm{opt}}| + H + W$, where N_{opt} is a fixed MMN and $|M|$ is the total length of a set M of line segments. We will abuse notation by identifying a set M of line segments with the corresponding set of points $\bigcup M$.

In phase III we connect the pairs in Z_{quad}. For a set $M \subseteq \mathbb{R}^2$ we denote by ∂M the boundary of M and by $\mathrm{int}(M) = M \setminus \partial M$ the interior of M. Let $P(q,t) = \{p \in P \cap Q(q,t) \mid (p,q) \in Z_{\mathrm{quad}}\}$ for $t = 1, \ldots, 4$. For each q and t we have to connect q to the points in $P(q,t)$. Let $\Delta(q,t) = \bigcup_{p \in P(q,t)} \mathrm{BBox}(p,q) \setminus \mathrm{int}(\mathcal{A}_{\mathrm{hor}} \cup \mathcal{A}_{\mathrm{ver}})$, see the gray areas in Fig. 2. Let $\delta(q,t)$ be the union of those connected components of $\Delta(q,t)$ that are actually incident to some $p \in P(q,t)$. Each connected component A of $\delta(q,t)$ is a staircase polygon.

We give an algorithm $B^\star$ that computes for each A a set $B^\star(A)$ of line segments in $\mathrm{int}(A)$ with $|B^\star(A)| \leq 2|N_{\mathrm{opt}} \cap \mathrm{int}(A)|$ (thickly dotted in Fig. 2). The set $B^\star(A)$ connects the points in $P \cap A$ to the point $q_A \in \partial A$ closest to q. Our algorithm $B^\star$, which runs in $O(k \log k)$ time ($k = |A \cap P|$), is a non-trivial modification of an $O(k)$-time factor-2 approximation algorithm B for rectangulating staircase polygons [2]. The algorithm in [3] needs an exact rectangulation, which takes $O(k^3)$ time.

In case the point q_A is not yet connected to q via N_1 we need an extra vertical segment s_A (thinly dotted in Fig. 2) to connect q_A to segments in N_1 that lead to q. The segment s_A, the boundary of A (dashed in Fig. 2) and the set $B^\star(A)$ connect q to the points in $P(q,t)$ via N_1. Let N_2 be the set of all segments s_A and of the pieces of ∂A that are not in N_1, both over all sets of type A. Let N_3 be the union of all $B^\star(A)$ over all sets of type A. Our algorithm returns the line segments in $N = N_1 \cup N_2 \cup N_3$, see Fig. 2. By the discussion above, N is a Manhattan network.

To bound the length of N we partition the plane into two sets $\mathcal{A}_{12}$ and $\mathcal{A}_3$, and compare N to N_{opt} in each region separately. Set $\mathcal{A}_3$ is the union of $\mathrm{int}(A)$ over all sets of type A, while $\mathcal{A}_{12} = \mathbb{R}^2 \setminus \mathcal{A}_3$. We have $N_1 \cup N_2 \subseteq \mathcal{A}_{12}$ and $N_3 \subseteq \mathcal{A}_3$, and the interiors of different regions of type A do not intersect. Algorithm $B^\star$ guarantees $|N \cap \mathcal{A}_3| \leq 2|N_{\mathrm{opt}} \cap \mathcal{A}_3|$. We show that $|N_2| \leq 2|N_{\mathrm{opt}}| - (H + W)$. Thus $|N \cap \mathcal{A}_{12}| = |(N_1 \cup N_2) \cap \mathcal{A}_{12}| \leq 3|N_{\mathrm{opt}} \cap \mathcal{A}_{12}|$, which in turn yields $|N| \leq 3|N_{\mathrm{opt}}|$.

Theorem 1 *An* MMN *can be 3-approximated in* $O(n \log n)$ *time and* $O(n)$ *space.*

References

[1] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. Smid. Euclidean spanners: Short, thin, and lanky. In *Proc. STOC'95*, pp. 489–498, 1995.

[2] J. Gudmundsson, C. Levcopoulos, and G. Narasimhan. Approximating a minimum Manhattan network. *Nordic J. Comput.*, 8:219–232, 2001.

[3] R. Kato, K. Imai, and T. Asano. An improved algorithm for the minimum Manhattan network problem. *Proc. ISAAC'02*, vol. 2518 of *LNCS*, pp. 344–356, 2002. Springer-Verlag.