Approximating the Generalized Minimum Manhattan Network Problem*

Aparna Das[1], Krzysztof Fleszar[2], Stephen Kobourov[1], Joachim Spoerhase[2], Sankar Veeramoni[1], and Alexander Wolff[2]

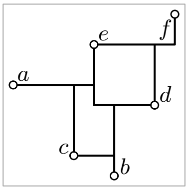
[1] Department of Computer Science, University of Arizona, Tucson, AZ, U.S.A.
[2] Lehrstuhl I, Institut für Informatik, Universität Würzburg, Germany

Abstract. We consider the *generalized minimum Manhattan network problem* (GMMN). The input to this problem is a set R of n pairs of terminals, which are points in $\mathbb{R}^2$. The goal is to find a minimum-length rectilinear network that connects every pair in R by a *Manhattan path*, that is, a path of axis-parallel line segments whose total length equals the pair's Manhattan distance. This problem is a natural generalization of the extensively studied *minimum Manhattan network problem* (MMN) in which R consists of all possible pairs of terminals. Another important special case is the well-known *rectilinear Steiner arborescence problem* (RSA). As a generalization of these problems, GMMN is NP-hard. No approximation algorithms are known for general GMMN.

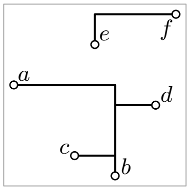
We obtain an $O(\log n)$-approximation algorithm for GMMN. Our solution is based on a stabbing technique, a novel way of attacking Manhattan network problems. Some parts of our algorithm generalize to higher dimensions, yielding a simple $O(\log^{d+1} n)$-approximation algorithm for the problem in arbitrary fixed dimension d. As a corollary, we obtain an exponential improvement upon the previously best $O(n^\varepsilon)$-ratio for MMN in d dimensions [ESA'11]. En route, we show that an existing $O(\log n)$-approximation algorithm for 2D-RSA generalizes to higher dimensions.

1 Introduction

Given a set of terminals, which are points in $\mathbb{R}^2$, the *minimum Manhattan network problem* (MMN) asks for a minimum-length rectilinear network that connects every pair of terminals by a Manhattan path (*M-path*, for short), i.e., a path consisting of axis-parallel segments whose total length equals the pair's M-distance. Put differently, every pair is to be connected by a shortest path in the L_1-norm (M-path). See Fig. 1a for an example.



(a) an MMN for $\{a, b, c, d, e, f\}$



(b) a GMMN for $\{(a,b), (c,d), (e,f)\}$

Fig. 1: MMN versus GMMN

In the *generalized minimum Manhattan network problem* (GMMN), we are given a set R of n unordered terminal *pairs*, and

* This work was supported by the ESF EuroGIGA project GraDR (DFG grant Wo 758/5-1).

L. Cai, S.-W. Cheng, and T.-W. Lam (Eds.): ISAAC2013, LNCS 8283, pp. 722–732, 2013.

the goal is to find a minimum-length rectilinear network such that every pair in R is *M-connected*, that is, connected by an M-path. GMMN is a generalization of MMN since R may contain all possible pairs of terminals. Figure 1b depicts such a network.

We remark that, in this paper, we define n to be the number of terminal *pairs* of a GMMN instance, previous works on MMN defined n to be the number of *terminals*. Moreover, we identify each terminal pair with a rectangle, namely the bounding box of this pair. This is a natural convention as every M-path for this terminal pair lies within the bounding box.

MMN naturally arises in VLSI circuit layout [8], where a set of terminals (such as gates or transistors) needs to be interconnected by rectilinear paths (wires). Minimizing the cost of the network (which means minimizing the total wire length) is desirable in terms of energy consumption and signal interference. The additional requirement that the terminal pairs are connected by *shortest* rectilinear paths aims at decreasing the interconnection delay (see Cong et al. [4] for a discussion in the context of rectilinear Steiner arborescences, which have the same additional requirement; see definition below). Manhattan networks also arise in the area of geometric spanner networks. Specifically, a minimum Manhattan network can be thought of as the cheapest spanner under the L_1-norm for a given set of points (allowing Steiner points). Spanners, in turn, have numerous applications in network design, distributed algorithms, and approximation algorithms, see, e.g., the book [14] and the survey [9].

MMN requires a Manhattan path between every terminal pair. This assumption is, however, not always reasonable. For example, in VLSI design a wire connection is necessary only for an, often comparatively small, subset of terminal pairs, which may allow for substantially cheaper circuit layouts. In this scenario, GMMN appears to be a more realistic model than MMN.

Previous Work and Related Problems. MMN was introduced by Gudmundsson et al. [8] who gave 4- and 8-approximation algorithms for MMN running in $O(n^3)$ and $O(n \log n)$ time, respectively. The currently best known approximation algorithms for MMN have ratio 2; they were obtained independently by Chepoi et al. [2] using an LP-based method, by Nouioua [16] using a primal-dual scheme, and by Guo et al. [10] using a greedy approach. The complexity of MMN was settled only recently by Chin et al. [3]; they proved the problem NP-hard. It is not known whether MMN is APX-hard. Gudmundsson et al. [7] consider a variant of MMN where the goal is to minimize the number of (Steiner) nodes and edges. Using divide-and-conquer they show that there is always a Manhattan network with $O(n \log n)$ nodes and edges. Knauer and Spillner [11] show that MMN is fixed-parameter tractable. More specifically, they show that there is an exact algorithm for MMN taking $O^*(2^{14h})$ time, where h is the number of horizontal lines that contain all terminals and the O^*-notation neglects factors polynomial in n.

Recently, there has been an increased interest in MMN for higher dimensions. Muñoz et al. [13] proved that 3D-MMN is NP-hard to approximate within a factor of 1.00002. They also gave a constant-factor approximation algorithm for a (rather restricted) special case of 3D-MMN. Das et al. [6] described the first approximation algorithm for MMN in arbitrary, fixed dimension. Their algorithm recursively computes a grid and attaches the terminals within a grid cell to grid vertices using RSA as a subroutine. Its ratio is $O(n^{\varepsilon})$ for any $\varepsilon > 0$.

GMMN was defined by Chepoi et al. [2] who posed the question whether it admits an $O(1)$-approximation. Suprisingly, only special cases of GMMN such as MMN have been considered so far—despite the fact that the problem is very natural and relevant for practical applications.

Another special case of GMMN that has received significant attention in the past is the *rectilinear Steiner arborescence problem* (RSA). Here, one is given a set of n terminals in the first quadrant, and the goal is to find a minimum-length rectilinear network that M-connects every terminal to the origin o. Hence, RSA is the special case of GMMN where o is considered a (new) terminal and the set of terminal pairs contains, for each terminal $t \neq o$, only the pair (o, t). Note that RSA is very different from MMN. Although every RSA solution is connected (via the origin), terminals are not necessarily *M-connected* to each other. RSA was introduced by Nastansky et al. [15]. RSA is NP-hard [18]. Rao et al. [17] gave a 2-approximation algorithm for RSA. They also provided a conceptually simpler $O(\log n)$-approximation algorithm based on rectilinear Steiner trees. In the full version of this paper [5], we generalize this algorithm to dimensions $d > 2$. Lu et al. [12] and, independently, Zachariasen [19] described polynomial-time approximation schemes (PTAS) for RSA, both based on Arora's technique [1]. Zachariasen pointed out that his PTAS can be generalized to the all-quadrant version of RSA but that it seems difficult to extend the approach to higher dimensions.

Our Contribution. Our main result is the first approximation algorithm for GMMN. Its ratio is $O(\log n)$ (see Section 3). Our algorithm is based on two ideas. First, we use a simple (yet powerful) divide-and-conquer scheme to reduce the problem to RSA. This yields a ratio of $O(\log^2 n)$. To bring down the ratio to $O(\log n)$ we develop a new *stabbing technique*, which is a novel way to approach Manhattan network problems and constitutes the main technical contribution of this paper.

We also consider higher dimensions. More specifically, we generalize an existing $O(\log n)$-approximation algorithm for RSA to arbitrary dimensions (see the full version [5]). Combining this with our divide-and-conquer scheme yields an $O(\log^{d+1} n)$-approximation algorithm for d-dimensional GMMN (see Section 4). For the special case of d-dimensional MMN, this constitutes an exponential improvement upon the $O(n^{\varepsilon})$-approximation algorithm of Das et al. [6]. Another advantage of our algorithm is that it is significantly simpler and easier to analyze than that algorithm.

Our result is a first step towards answering the open question of Chepoi et al. [2]. In the full version [5] we give indications that it may be difficult to obtain an $O(1)$-approximation algorithm since the problem can be viewed as a geometric rectangle covering problem. There we also argue why existing techniques for MMN seem to fail, which underlines the relevance of our techniques.

2 Divide-And-Conquer Scheme

As a warm-up, we start with a simple $O(\log^2 n)$-approximation algorithm illustrating our divide-and-conquer scheme. This is the basis for (a) an improved $O(\log n)$-approximation algorithm that uses our stabbing technique (see Section 3) and (b) a divide-and-conquer scheme for GMMN in arbitrary dimensions (Section 4). We prove the following.

Theorem 1. *GMMN admits an $O(\log^2 n)$-approximation algorithm running in $O(n \log^3 n)$ time.*

Our algorithm consists of a *main algorithm* that recursively subdivides the input instance into instances of so-called x-*separated* GMMN; see Section 2.1. We prove that the instances of x-separated GMMN can be solved independently by paying a factor of $O(\log n)$ in the overall approximation ratio. Then we solve each x-separated GMMN instance within factor $O(\log n)$; see Section 2.2. This yields an overall approximation ratio of $O(\log^2 n)$. Our analysis is tight; see the full version [5]. Our presentation follows this natural top-down approach; as a consequence, we will make some forward references to results that we prove later.

2.1 Main Algorithm

Our algorithm is based on divide and conquer. Let R be the set of terminal pairs that are to be M-connected. Recall that we identify each terminal pair with its bounding box. As a consequence of this, we consider R, a set of rectangles. Let m_x be the median in the multiset of the x-coordinates of terminals where a terminal occurs as often as the number of pairs it is involved in. We identify m_x with the vertical line at $x = m_x$.

Now we partition R into three subsets R_{left}, R_{mid}, and R_{right}. R_{left} consists of all rectangles that lie *completely* to the left of the vertical line m_x. Similarly, R_{right} consists of all rectangle that lie *completely* to the right of m_x. R_{mid} consists of all rectangles that intersect m_x.

We consider the sets R_{left}, R_{mid}, and R_{right} as separate instances of GMMN. We apply the main algorithm recursively to R_{left} to get a rectilinear network that M-connects terminal pairs in R_{left} and do the same for R_{right}.

It remains to M-connect the pairs in R_{mid}. We call a GMMN instance (such as R_{mid}) x-*separated* if there is a vertical line (in our case m_x) that intersects every rectangle. We exploit this property to design a simple $O(\log n)$-approximation algorithm for x-separated GMMN; see Section 2.2. In Section 3, we improve upon this and describe an $O(1)$-approximation algorithm for x-separated GMMN.

In the following lemma we analyze the performance of the main algorithm, in terms of $\rho_x(n)$, our approximation ratio for x-separated instances with n terminal pairs.

Lemma 1. *Let $\rho_x(n)$ be a non-decreasing function. Then, if x-separated GMMN admits a $\rho_x(n)$-approximation algorithm, GMMN admits a $(\rho_x(n) \cdot \log n)$-approximation algorithm.*

Proof. We determine an upper bound $\rho(n)$ on the main algorithm's approximation ratio for instances with n terminal pairs. Let N^{opt} be an optimum solution to an instance R of size n and let OPT be the cost of N^{opt}. Let $N^{\text{opt}}_{\text{left}}$ and $N^{\text{opt}}_{\text{right}}$ be the parts of N^{opt} to the left and to the right of m_x, respectively. (We split horizontal segments that cross m_x and ignore vertical segments on m_x.)

Due to the choice of m_x, at most n terminals lie to the left of m_x. Therefore, R_{left} contains at most $n/2$ terminal pairs. Since $N^{\text{opt}}_{\text{left}}$ is a feasible solution to R_{left}, we conclude (by induction) that the cost of the solution to R_{left} computed by our algorithm is bounded by $\rho(n/2) \cdot \|N^{\text{opt}}_{\text{left}}\|$, where $\|\cdot\|$ measures the length of a network. Analogously, the cost of the solution computed for R_{right} is bounded by $\rho(n/2) \cdot \|N^{\text{opt}}_{\text{right}}\|$. Since N^{opt} is also a feasible solution to the x-separated instance R_{mid}, we can compute a solution of cost $\rho_x(n) \cdot \text{OPT}$ for R_{mid}.

As the networks $N^{\text{opt}}_{\text{left}}$ and $N^{\text{opt}}_{\text{right}}$ are separated by line m_x, they are edge disjoint and hence $\|N^{\text{opt}}_{\text{left}}\| + \|N^{\text{opt}}_{\text{right}}\| \leq \text{OPT}$. Therefore, we can bound the total cost of our algorithm's solution N to R by

$$\rho(n/2) \cdot (\|N^{\text{opt}}_{\text{left}}\| + \|N^{\text{opt}}_{\text{right}}\|) + \rho_x(n) \cdot \text{OPT} \leq (\rho(n/2) + \rho_x(n)) \cdot \text{OPT} \ .$$

This yields the recurrence $\rho(n) = \rho(n/2) + \rho_x(n)$, which resolves to $\rho(n) \leq \log n \cdot \rho_x(n)$. □

Lemma 1 together with the results of Section 2.2 allow us to prove Theorem 1.

Proof (of Theorem 1). By Lemma 1, our main algorithm has performance $\rho_x(n) \cdot \log n$, where $\rho_x(n)$ denotes the ratio of an approximation algorithm for x-separated GMMN. In Lemma 2 (Section 2.2), we will show that there is an algorithm for x-separated GMMN with ratio $\rho_x(n) = O(\log n)$. Thus overall, the main algorithm yields an $O(\log^2 n)$-approximation for GMMN. See the full version [5] for the running time analysis. □

2.2 Approximating x-Separated and xy-Separated Instances

We describe a simple algorithm for approximating x-separated GMMN with a ratio of $O(\log n)$. Let R be an x-separated instance, that is, all rectangles in R intersect a common vertical line.

The algorithm works as follows. Analogously to the main algorithm we subdivide the x-separated input instance, but this time using the line $y = m_y$, where m_y is the median of the multiset of y-coordinates of terminals in R. This yields sets R_{top}, R'_{mid}, and R_{bottom}, defined analogously to the sets R_{left}, R_{mid}, and R_{right} of the main algorithm, using m_y instead of m_x. We apply our x-separated algorithm to R_{top} and then to R_{bottom} to solve them recursively. The instance R'_{mid} is a *y-separated* sub-instance with all its rectangles intersecting the line m_y. Moreover, R'_{mid} (as a subset of R) is already x-separated, thus we call R'_{mid} an *xy-separated* instance. Below, we describe a specialized algorithm to approximate xy-separated instances within a constant factor. Assuming this for now, we prove the following.

Lemma 2. *x-separated GMMN admits an $O(\log n)$-approximation algorithm.*

Proof. Let $\rho_x(n)$ be the ratio of our algorithm for approximating x-separated GMMN instances and let $\rho_{xy}(n)$ be the ratio of our algorithm for approximating xy-separated GMMN instances. In Lemma 3, we show that $\rho_{xy}(n) = O(1)$.

Following the proof of Lemma 1 (exchanging x- and y-coordinates and using R_{top}, R'_{mid}, R_{bottom} in place of $R_{\text{left}}, R_{\text{mid}}, R_{\text{right}}$), yields $\rho_x(n) = \log n \cdot \rho_{xy}(n) = O(\log n)$. □

It remains to show that xy-separated GMMN can be approximated within a constant ratio. Let R be an instance of xy-separated GMMN. We assume, w.l.o.g., that it is the x- and the y-axes that intersect all rectangles in R, that is, all rectangles contain the origin o. To solve R, we compute an RSA network that M-connects the set of terminals in R to o. Clearly, we obtain a feasible GMMN solution to R. In the full version [5] we prove that this is a constant-factor approximation algorithm.

Lemma 3. *xy-separated GMMN admits a constant-factor approximation algorithm.*

3 An $O(\log n)$-Approximation Algorithm via Stabbing

In this section, we present an $O(\log n)$-approximation algorithm for GMMN, which is the main result of our paper. Our algorithm relies on an $O(1)$-approximation algorithm for x-separated instances and is based on a novel stabbing technique that computes a cheap set of horizontal line segments that stabs all rectangles. Our algorithm connects these line segments with a suitable RSA solution to ensure feasibility and approximation ratio. We show the following (noting that our analysis is tight up to a constant factor; see the full version [5]).

Theorem 2. *For any $\varepsilon > 0$, GMMN admits a $((6+\varepsilon)\cdot\log n)$-approximation algorithm running in $O(n^{1/\varepsilon}\log^2 n)$ time.*

Proof. Using our new subroutine for the x-separated case given in Lemma 7 below, along with Lemma 1 yields the result. See the full version [5] for the run-time analysis. □

We begin with an overview of our improved algorithm for x-separated GMMN. Let R be the set of terminal pairs of an x-separated instance of GMMN. We assume, w.l.o.g., that each terminal pair $(t, t') \in R$ is separated by the y-axis, that is, $x(t) \leq 0 \leq x(t')$ or $x(t') \leq 0 \leq x(t)$. Let $N^{\rm opt}$ be an optimum solution to R. Let $\mathrm{OPT_{ver}}$ and $\mathrm{OPT_{hor}}$ be the total costs of the vertical and horizontal segments in $N^{\rm opt}$, respectively. Hence, $\mathrm{OPT} = \mathrm{OPT_{ver}} + \mathrm{OPT_{hor}}$. We first compute a set S of horizontal line segments of total cost $O(\mathrm{OPT_{hor}})$ such that each rectangle in R is *stabbed* by some line segment in S; see Sections 3.1 and 3.2. Then we M-connect the terminals to the y-axis so that the resulting network, along with S, forms a feasible solution to R of cost $O(\mathrm{OPT})$; see Section 3.3.

3.1 Stabbing the Right Part

We say that a horizontal line segment h *stabs* an axis-aligned rectangle r if the intersection of r and h equals the intersection of r and the supporting line of h. A set of horizontal line segments is a *stabbing* of a set of axis-aligned rectangles if each rectangle is stabbed by some line segment. For any geometric object, let its *right part* be its intersection with the closed half plane to the right of the y-axis. For a *set* of objects, let its right part be the set of the right parts of the objects. Let R^+ be the right part of R, let N^+ be the right part of $N^{\rm opt}$, and let $N^+_{\rm hor}$ be the set of horizontal line segments in N^+. In this section, we show how to construct a stabbing of R^+ of cost at most $2 \cdot \|N^+_{\text{hor}}\|$.

For $x' \geq 0$, let $\ell_{x'}$ be the vertical line at $x = x'$. Our algorithm performs a left-to-right sweep starting with ℓ_0. For $x \geq 0$, let $\mathcal{I}_x = \{r \cap \ell_x \mid r \in R^+\}$ be the "traces" of the rectangles in R^+ on ℓ_x. The elements of $\mathcal{I}_x$ are vertical line segments; we refer to them as *intervals*. A set P_x of points on ℓ_x constitutes a *piercing* for $\mathcal{I}_x$, if every interval in $\mathcal{I}_x$ contains a point in P_x.

Our algorithm continuously moves the line ℓ_x from left to right starting with $x = 0$. In doing so, we maintain an inclusion-wise minimal piercing P_x of $\mathcal{I}_x$ in the following way: At $x = 0$, we start with an arbitrary minimal piercing P_0. (Note that we can even compute an optimum piercing.) We update P_x whenever $\mathcal{I}_x$ changes. Observe that with increasing x, the set $\mathcal{I}_x$ can only inclusion-wise decrease as all rectangles in R^+ touch the y-axis. Therefore, it suffices to update the piercing P_x only at *event points*; x is an event point if and only if x is the x-coordinate of a right edge of a rectangle in R^+. Let x' and x'' be consecutive event points. Let x be such that $x' < x \leq x''$. Note that $P_{x'}$ is a piercing for $\mathcal{I}_x$ since $\mathcal{I}_x \subset \mathcal{I}_{x'}$. The piercing $P_{x'}$ is, however, not necessarily *minimal* w.r.t. $\mathcal{I}_x$. When the sweep line passes x', we therefore have to drop some of the points in $P_{x'}$ in order to obtain a new minimal piercing. This can be done by iteratively removing points from $P_{x'}$ such that the resulting set still pierces $\mathcal{I}_x$. We stop at the last event point (afterwards, $\mathcal{I}_x = \emptyset$) and output the *traces* of the piercing points in P_x for $x \geq 0$ as our stabbing.

Note that with increasing x, our algorithm only *removes* points from P_x but never add points. Thus, the traces of P_x form horizontal line segments that touch the y-axis. These line segments form a stabbing of R^+; see the thick solid line segments in Fig. 2a. The following lemma is crucial to prove the overall cost of the stabbing.

Lemma 4. *For any $x \geq 0$, it holds that $|P_x| \leq 2 \cdot |\ell_x \cap N^+_{\text{hor}}|$.*

Proof. Since P_x is a minimal piercing, there exists, for every $p \in P_x$, a *witness* $I_p \in \mathcal{I}_x$ that is pierced by p but not by $P_x \setminus \{p\}$. Otherwise we could remove p from P_x, contradicting the minimality of P_x.

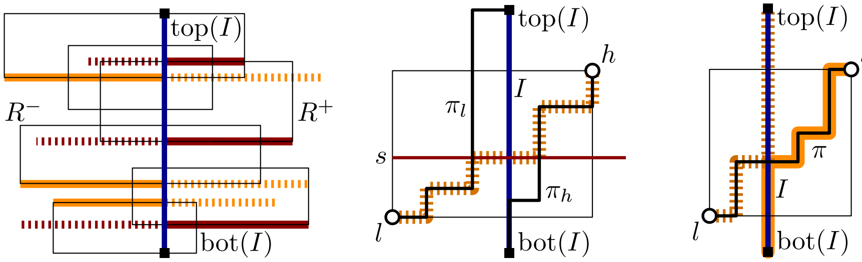
Now we show that an arbitrary point q on ℓ_x is contained in the witnesses of at most two points in P_x. Assume, for the sake of contradiction, that q is contained in the witnesses of points $p, p', p'' \in P_x$ with strictly increasing y-coordinates. Suppose that q lies above p'. Then the witness I_p of p, which contains p and q, must also contain p', contradicting the definition of I_p. The case q below p' is symmetric.

Observe that $\ell_x \cap N^+_{\text{hor}}$ is a piercing of $\mathcal{I}_x$ and, hence, of the $|P_x|$ many witnesses. Since every point in $\ell_x \cap N^+_{\text{hor}}$ pierces at most two witnesses, the lemma follows. □

Next, we analyze the overall cost of the stabbing.

Lemma 5. *Given a set R of rectangles intersecting the y-axis, we can compute a set of horizontal line segments of cost at most $2 \cdot \mathrm{OPT}_{\text{hor}}$ that stabs R^+.*

Proof. Observe that $\|N^+_{\text{hor}}\| = \int |\ell_x \cap N^+_{\text{hor}}| \, dx$. The cost of our stabbing is $\int |P_x| \, dx$. By Lemma 4, this can be bounded by $\int |P_x| \, dx \leq \int 2 \cdot |\ell_x \cap N^+_{\text{hor}}| \, dx = 2 \cdot \|N^+_{\text{hor}}\|$. □



(a) The dark (light) segments S^+ (S^-) stab R^+ (R^-). The dotted segments are mirror images of $S^+ \cup S^-$.

(b) $N = A_{\text{up}} \cup A_{\text{down}} \cup S$ is feasible for R.

(c) $N^{\text{opt}} \cup \{I\}$ is feasible for RSA instances $(L, \text{top}(I))$, $(H, \text{bot}(I))$.

Fig. 2: The improved algorithm for x-separated GMMN

3.2 Stabbing the Right and Left Parts

We now detail how we construct a stabbing of R. To this end we apply Lemma 5 to compute a stabbing S^- of cost at most $2 \cdot \|N^-_{\text{hor}}\|$ for the left part R^- of R and a stabbing S^+ of cost at most $2 \cdot \|N^+_{\text{hor}}\|$ for the right part R^+. Note that $S^- \cup S^+$ is not necessarily a stabbing of R since there can be rectangles that are not *completely* stabbed by one segment (even if we start with the same piercing on the y-axis in the sweeps to the left and to the right). To overcome this difficulty, we mirror S^- and S^+ to the respective other side of the y-axis; see Fig. 2a. Let S denote the union of $S^- \cup S^+$ and the mirror image of $S^- \cup S^+$.

Lemma 6. *Given a set R of rectangles intersecting the y-axis, we can compute a set of horizontal line segments of cost at most $4 \cdot \text{OPT}_{\text{hor}}$ that stabs R.*

Proof. Let S be the set of horizontal line segments described above. The total cost of S is at most $4(\|N^-_{\text{hor}}\| + \|N^+_{\text{hor}}\|) = 4 \cdot \text{OPT}_{\text{hor}}$. The set S stabs R since, for every rectangle $r \in R$, the larger among its two (left and right) parts is stabbed by some segment s and the smaller part is stabbed by the mirror image s' of s. Hence, r is stabbed by the line segment $s \cup s'$. □

3.3 Connecting Terminals and Stabbing

We assume that the union of the rectangles in R is connected. Otherwise we apply our algorithm separately to each subset of R that induces a connected component of $\bigcup R$. Let I be the line segment that is the intersection of the y-axis with $\bigcup R$. Let $\text{top}(I)$ and $\text{bot}(I)$ be the top and bottom endpoints of I, respectively. Let $L \subseteq T$ be the set containing every terminal t with $(t, t') \in R$ and $y(t) \le y(t')$ for some $t' \in T$. Symmetrically, let $H \subseteq T$ be the set containing every terminal t with $(t, t') \in R$ and $y(t) \ge y(t')$ for some $t' \in T$. Note that, in general, L and H are not disjoint.

Using a PTAS for RSA [12,19], we compute a near-optimal RSA network A_{up} connecting the terminals in L to $\text{top}(I)$ and a near-optimal RSA network A_{down} connecting the terminals in H to $\mathrm{bot}(I)$. Then we return the network $N = A_{\mathrm{up}} \cup A_{\mathrm{down}} \cup S$, where S is the stabbing computed by the algorithm in Section 3.2.

We prove in the following lemma that the resulting network is a feasible solution to R, with cost at most constant times OPT.

Lemma 7. *x-separated GMMN admits, for any $\varepsilon > 0$, a $(6+\varepsilon)$-approximation algorithm.*

Proof. First we argue that the solution is feasible. Let $(l,h) \in R$. W.l.o.g., $y(l) \leq y(h)$ and thus $l \in L$ and $h \in H$. Hence, A_{up} contains a path π_l from l to $\mathrm{top}(I)$, see Fig. 2b. This path starts inside the rectangle (l,h). Before leaving (l,h), the path intersects a line segment s in S that stabs (l,h). The segment s is also intersected by the path π_h in A_{down} that connects h to $\mathrm{bot}(I)$. Hence, walking along π_l, s, and π_h brings us in a monotone fashion from l to h.

Now, let us analyze the cost of N. Clearly, the projection of (the vertical line segments of) N^{opt} onto the y-axis yields the line segment I. Hence, $\|I\| \leq \mathrm{OPT}_{\mathrm{ver}}$. Observe that $N^{\mathrm{opt}} \cup \{I\}$ constitutes a solution to the RSA instance $(L, \mathrm{top}(I))$ connecting all terminals in L to $\mathrm{top}(I)$ and to the RSA instance $(H, \mathrm{bot}(I))$ connecting all terminals in H to $\mathrm{bot}(I)$. This holds since, for each terminal pair, its M-path π in N^{opt} crosses the y-axis in I; see Fig. 2c. Since A_{up} and A_{down} are near-optimal solutions to these RSA instances, we obtain, for any $\delta > 0$, that $\|A_{\mathrm{up}}\| \leq (1+\delta) \cdot \|N^{\mathrm{opt}} \cup I\| \leq (1+\delta)\cdot(\mathrm{OPT}+\mathrm{OPT}_{\mathrm{ver}})$ and, analogously, that $\|A_{\mathrm{down}}\| \leq (1+\delta)\cdot(\mathrm{OPT}+\mathrm{OPT}_{\mathrm{ver}})$.

By Lemma 6, we have $\|S\| \leq 4 \cdot \mathrm{OPT}_{\mathrm{hor}}$. Assuming $\delta \leq 1$, this yields

$$\begin{aligned}\|N\| &= \|A_{\mathrm{up}}\| + \|A_{\mathrm{down}}\| + \|S\| \leq (2+2\delta)\cdot(\mathrm{OPT}+\mathrm{OPT}_{\mathrm{ver}}) + 4\cdot\mathrm{OPT}_{\mathrm{hor}} \\ &\leq (2+2\delta)\cdot\mathrm{OPT} + 4\cdot(\mathrm{OPT}_{\mathrm{ver}}+\mathrm{OPT}_{\mathrm{hor}}) = (6+2\delta)\cdot\mathrm{OPT}\ .\end{aligned}$$

Setting $\delta = \varepsilon/2$ yields the desired approximation factor. □

4 Generalization to Higher Dimensions

In this section, we describe an $O(\log^{d+1} n)$-approximation algorithm for GMMN in d dimensions and prove the following result (see below for the proof). In the full version [5] we show that the analysis of the algorithm is essentially tight (up to one log-factor).

Theorem 3. *In any fixed dimension d, GMMN admits an $O(\log^{d+1} n)$-approximation algorithm running in $O(n^2 \log^{d+1} n)$ time.*

In Section 2 we reduced GMMN to x-separated GMMN and then x-separated GMMN to xy-separated GMMN. Each of the two reductions increased the approximation ratio by a factor of $O(\log n)$. The special case of xy-separated GMMN was approximated within a constant factor by solving a related RSA problem. This gave an overall $O(\log^2 n)$-approximation algorithm for GMMN. We generalize this approach to higher dimensions.

An instance R of d-dimensional GMMN is called *j-separated* for some $j \leq d$ if there exist values $s_1, \ldots, s_j$ such that, for each terminal pair $(t,t') \in R$ and for each dimension $i \leq j$, we have that s_i *separates* the i-th coordinates $x_i(t)$ of t and $x_i(t')$ of t' (meaning that either $x_i(t) \leq s_i \leq x_i(t')$ or $x_i(t') \leq s_i \leq x_i(t)$). Under this terminology, an arbitrary instance of d-dimensional GMMN is always *0-separated*.

The following lemma reduces j-separated GMMN to $(j-1)$-separated GMMN at the expense of a $(\log n)$-factor in the approximation ratio. The proof is similar to the 2D case; see the full version [5].

Lemma 8. *Let $1 \leq j \leq d$. If j-separated GMMN admits a $\rho_j(n)$-approximation algorithm, then $(j-1)$-separated GMMN admits a $(\rho_j(n) \cdot \log n)$-approximation algorithm.*

Analogously to dimension two we can approximate instances of d-separated GMMN by reducing the problem to RSA. Rao et al. [17] presented an $O(\log |T|)$-approximation algorithm for 2D-RSA, which generalizes to d-dimensional RSA as we show in the full version [5]. Using this, we derive there the following result.

Lemma 9. *d-separated GMMN admits an $O(\log n)$-approximation algorithm for any fixed dimension d.*

We are now ready to give the proof of Theorem 3.

Proof (Proof of Theorem 3). Combining Lemmata 8 and 9 and applying them inductively to arbitrary (that is, 0-separated) GMMN instances yields the claim. See the full version [5] for the run-time analysis. □

As a byproduct of Theorem 3, we obtain an $O(\log^{d+1} n)$-approximation algorithm for *MMN* where n denotes the number of *terminals*. This holds since any MMN instance with n terminals can be considered an instance of GMMN with $O(n^2)$ terminal pairs.

Corollary 1. *In any fixed dimension d, MMN admits an $O(\log^{d+1} n)$-approximation algorithm running in $O(n^4 \log^{d+1} n)$ time, where n denotes the number of terminals.*

5 Conclusions

In 2D, there is quite a large gap between the currently best approximation ratios for MMN and GMMN. Whereas we have presented an $O(\log n)$-approximation algorithm for GMMN, MMN admits 2-approximation algorithms [2,10,16]. In the full version [5], we give indications that this gap might not only be a shortcoming of our algorithm. It would be interesting to derive *some* non-approximability result for GMMN. So far, the only such result is the APX-hardness of 3D-MMN [13].

Concerning the positive side, for $d \geq 3$, a constant-factor approximation algorithm for d-dimensional RSA would shave off a factor of $O(\log n)$ from the current ratio for d-dimensional GMMN. This may be in reach since 2D-RSA admits even a PTAS [12,19]. Alternatively, a constant-factor approximation algorithm for $(d-k)$-separated GMMN for some $k \leq d$ would shave off a factor of $O(\log^k n)$ from the current ratio for d-dimensional GMMN.

Acknowledgments. We thank Michael Kaufmann for his hospitality and his enthusiasm during our respective stays in Tübingen. We thank Esther Arkin, Alon Efrat, Joe Mitchell, and Andreas Spillner for discussions.

References

1. Arora, S.: Approximation schemes for NP-hard geometric optimization problems: A survey. Math. Program. 97(1-2), 43–69 (2003)
2. Chepoi, V., Nouioua, K., Vaxès, Y.: A rounding algorithm for approximating minimum Manhattan networks. Theor. Comput. Sci. 390(1), 56–69 (2008)
3. Chin, F., Guo, Z., Sun, H.: Minimum Manhattan network is NP-complete. Discrete Comput. Geom. 45, 701–722 (2011)
4. Cong, J., Leung, K.S., Zhou, D.: Performance-driven interconnect design based on distributed RC delay model. In: 30th IEEE Conf. Design Automation (DAC 1993), pp. 606–611. IEEE Press, New York (1993)
5. Das, A., Fleszar, K., Kobourov, S.G., Spoerhase, J., Veeramoni, S., Wolff, A.: Approximating the generalized minimum Manhattan network problem. Arxiv report (2012), http://arxiv.org/abs/1203.6481
6. Das, A., Gansner, E.R., Kaufmann, M., Kobourov, S., Spoerhase, J., Wolff, A.: Approximating minimum Manhattan networks in higher dimensions. In: Demetrescu, C., Halldórsson, M.M. (eds.) ESA 2011. LNCS, vol. 6942, pp. 49–60. Springer, Heidelberg (2011), to appear in Algorithmica, http://dx.doi.org/10.1007/s00453-013-9778-z
7. Gudmundsson, J., Klein, O., Knauer, C., Smid, M.: Small Manhattan networks and algorithmic applications for the Earth Mover's Distance. In: 23rd Europ. Workshop Comput. Geom (EuroCG 2007), Graz, Austria, pp. 174–177 (2007)
8. Gudmundsson, J., Levcopoulos, C., Narasimhan, G.: Approximating a minimum Manhattan network. Nordic J. Comput. 8, 219–232 (2001)
9. Gudmundsson, J., Narasimhan, G., Smid, M.: Applications of geometric spanner networks. In: Kao, M.Y. (ed.) Encyclopedia of Algorithms, pp. 1–99. Springer (2008)
10. Guo, Z., Sun, H., Zhu, H.: Greedy construction of 2-approximate minimum Manhattan networks. Int. J. Comput. Geom. Appl. 21(3), 331–350 (2011)
11. Knauer, C., Spillner, A.: A fixed-parameter algorithm for the minimum Manhattan network problem. J. Comput. Geom. 2(1), 189–204 (2011)
12. Lu, B., Ruan, L.: Polynomial time approximation scheme for the rectilinear Steiner arborescence problem. J. Comb. Optim. 4(3), 357–363 (2000)
13. Muñoz, X., Seibert, S., Unger, W.: The minimal Manhattan network problem in three dimensions. In: Das, S., Uehara, R. (eds.) WALCOM 2009. LNCS, vol. 5431, pp. 369–380. Springer, Heidelberg (2009)
14. Narasimhan, G., Smid, M.: Geometric Spanner Networks. Cambridge University Press (2007)
15. Nastansky, L., Selkow, S.M., Stewart, N.F.: Cost-minimal trees in directed acyclic graphs. Zeitschrift Oper. Res. 18(1), 59–67 (1974)
16. Nouioua, K.: Enveloppes de Pareto et Réseaux de Manhattan: Caractérisations et Algorithmes. PhD thesis, Université de la Méditerranée (2005), http://www.lif-sud.univ-mrs.fr∼karim/download/THESE_NOUIOUA.pdf
17. Rao, S., Sadayappan, P., Hwang, F., Shor, P.: The rectilinear Steiner arborescence problem. Algorithmica 7, 277–288 (1992)
18. Shi, W., Su, C.: The rectilinear Steiner arborescence problem is NP-complete. SIAM J. Comput. 35(3), 729–740 (2005)
19. Zachariasen, M.: On the approximation of the rectilinear Steiner arborescence problem in the plane (2000) (Manuscript), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.4529