# Polylogarithmic Approximation for Generalized Minimum Manhattan Networks

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**

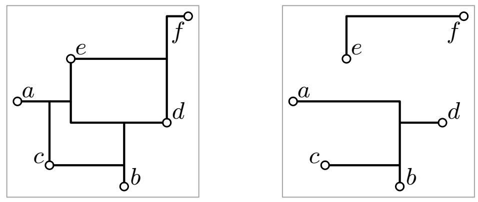
We consider the *generalized minimum Manhattan network problem* (GMMN). The input to this problem is a set $R$ of $n$ pairs of terminals, which are points in $\mathbb{R}^d$. The goal is to find a minimum-length rectilinear network that connects every pair in $R$ by a Manhattan path, that is, a path of axis-parallel line segments whose total length equals the pair's Manhattan distance. This problem is a 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 but approximation algorithms are only known for MMN and RSA.

We give the first approximation algorithm for GMMN; our algorithm has a ratio of $O(\log^{d+1} n)$ for the problem in arbitrary and fixed dimension $d$. This is an exponential improvement upon the $O(n^{\varepsilon})$-ratio of an existing algorithm for MMN in $d$ dimensions [ESA'11]. For the important case of dimension $d = 2$, we derive an improved bound of $O(\log n)$. Finally, we show that an existing $O(\log n)$-approximation algorithm for two-dimensional RSA generalizes to higher dimensions.

## 1 Introduction

Given a set of terminals, which are points in $\mathbb{R}^d$, 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), that is, a path consisting of axis-parallel segments whose total length equals the pair's Manhattan distance.

In the *generalized minimum Manhattan network problem* (GMMN), we are given a set $R$ of $n$ unordered terminal *pairs*, and 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 1 depicts examples of both network types. We remark that, in this paper, we define $n$ to be the number of terminal *pairs* of a GMMN instance, whereas previous works on MMN defined $n$ to be the number of *terminals*.




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

**Figure 1:** MMN versus GMMN in 2D.

Two-dimensional MMN (2D-MMN) naturally arises in VLSI circuit layout; higher-dimensional MMN has applications in the area of computational biology. 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 a, 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.

The currently best known approximation algorithms for 2D-MMN have ratio 2; for example, the $O(n \log n)$-time algorithm of Guo et al. [5]. The complexity of 2D-MMN was settled only recently by Chin et al. [2]; they proved the problem NP-hard. It is not known whether 2D-MMN is APX-hard.

Recently, there has been an increased interest in (G)MMN for higher dimensions. Muñoz et al. [7] 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. described the first approximation algorithm for MMN in arbitrary, fixed dimension with a ratio of $O(n^{\varepsilon})$ for any $\varepsilon > 0$ [4].

GMMN was defined by Chepoi et al. [1] who asked whether 2D-GMMN admits an $O(1)$-approximation. Apart from the formulation of this open problem, only special cases of GMMN such as MMN have been considered so far.

Another special case of GMMN that has received significant attention in the past is the *rectilinear Steiner arborescence problem* (RSA). Here, we are given $n$ terminals lying 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 $(o, t)$. RSA is NP-hard [9] even in dimension $d = 2$. Rao et al. [8] gave a 2-approximation algorithm for 2D-RSA. They also provided a conceptually simpler $O(\log n)$-approximation algorithm based on rectilinear Steiner trees. That algorithm generalizes quite easily to dimensions $d > 2$ (as we show in the long version of this paper [3]). Lu and Ruan [6] described a polynomial-time approximation schemes (PTAS) for 2D-RSA based on Arora's technique.

Our contribution is two-fold. First, we provide an $O(\log^{d+1} n)$-approximation algorithm for GMMN (and, hence, MMN) in $d$ dimensions. For the sake of simplicity, we first present our approach in 2D (see Section 2). Second, we provide an improved and technically more involved $O(\log n)$-approximation for the special case of 2D-GMMN; see Section 3.

## 2 Polylogarithmic Approximation

We present an $O(\log^2 n)$-approximation algorithm for 2D-GMMN and prove the following theorem.

**Theorem 1** *2D-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 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. We identify each terminal pair with its bounding box, that is, the smallest axis-aligned rectangle that contains both terminals. 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. We identify $m_x$ with the vertical line at $x = m_x$.

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

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

It remains to M-connect the pairs in $R_{\mathrm{mid}}$. We call a GMMN instance (such as $R_{\mathrm{mid}}$) *$x$-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. Later, in Section 3, we improve upon this and describe an $O(1)$-approximation algorithm for $x$-separated GMMN when terminals lie in 2D.

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 2** *If $x$-separated 2D-GMMN admits a $\rho_x(n)$-approximation, 2D-GMMN admits a $(\rho_x(n) \cdot \log n)$-approximation.*

**Proof.** Let $\rho(n)$ denote the main algorithm's worst-case approximation ratio for instances with $n$ terminal pairs. Now assume that our input instance $R$ is a worst case. More precisely, the cost of the solution of our algorithm *equals* $\rho(n) \cdot \mathrm{OPT}$, where OPT denotes the cost of an optimum solution $N^{\mathrm{opt}}$ to $R$. Let $N^{\mathrm{opt}}_{\mathrm{left}}$ and $N^{\mathrm{opt}}_{\mathrm{right}}$ be the parts of $N^{\mathrm{opt}}$ 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, $R_{\mathrm{left}}$ contains at most $n/2$ terminal pairs. Since $N^{\mathrm{opt}}_{\mathrm{left}}$ is a feasible solution to $R_{\mathrm{left}}$, we conclude that the cost of the solution to $R_{\mathrm{left}}$ computed by our algorithm is bounded by $\rho(n/2) \cdot \|N^{\mathrm{opt}}_{\mathrm{left}}\|$, where $\|\cdot\|$ measures the length of a network. Analogously, the cost of the solution computed for $R_{\mathrm{right}}$ is bounded by $\rho(n/2) \cdot \|N^{\mathrm{opt}}_{\mathrm{right}}\|$. Since $N^{\mathrm{opt}}$ is also a feasible solution to the $x$-separated instance $R_{\mathrm{mid}}$, we can compute a solution of cost $\rho_x(n) \cdot \mathrm{OPT}$ for $R_{\mathrm{mid}}$.

Therefore, we can bound the total cost of our algorithm's solution $N$ to $R$ by

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

Note that this inequality does not necessarily hold if $R$ is *not* a worst case since then $\rho(n) \cdot \text{OPT} > \|N\|$. The networks $N^{\text{opt}}_{\text{left}}$ and $N^{\text{opt}}_{\text{right}}$ are separated by line $m_x$, hence they are edge disjoint and $\|N^{\text{opt}}_{\text{left}}\| + \|N^{\text{opt}}_{\text{right}}\| \leq \text{OPT}$. This yields the recurrence $\rho(n) \leq \rho(n/2) + \rho_x(n)$, which resolves to $\rho(n) = \log n \cdot \rho_x(n)$. ☐

Lemma 2 together with the results of Section 2.2 allow us to prove Theorem 1.

**Proof.** By Lemma 2, 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 2D-GMMN. In Lemma 3 (Section 2.2), we will show that there is an algorithm for $x$-separated 2D-GMMN with ratio $\rho_x(n) = O(\log n)$. Thus overall, the main algorithm yields an $O(\log^2 n)$-approximation for 2D-GMMN. See the long version [3] for the running time analysis. ☐

## 2.2 Approximating $x$-Separated Instances

We describe a simple algorithm for approximating $x$-separated 2D-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 $R_{\text{top}}$, $R'_{\text{mid}}$, and $R_{\text{bottom}}$, defined analogously to the sets $R_{\text{left}}$, $R_{\text{mid}}$, and $R_{\text{right}}$ of the main algorithm, using $m_y$ instead of $m_x$. We apply our $x$-separated algorithm to $R_{\text{top}}$ and then to $R_{\text{bottom}}$ to solve them recursively. The instance $R'_{\text{mid}}$ is a *$y$-separated* sub-instance with all its rectangles intersecting the line $m_y$. Moreover, $R'_{\text{mid}}$ (as a subset of $R$) is already $x$-separated, thus we call $R'_{\text{mid}}$ an *$xy$-separated* instance. In Section 2.3, we give a specialized algorithm to approximate $xy$-separated instances within a constant factor. Assuming this for now, we show (in the long version [3]), analogously to Lemma 2, the following.

**Lemma 3** *$x$-separated 2D-GMMN admits an $O(\log n)$-approximation.*

## 2.3 Approximating $xy$-Separated Instances

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 $o$. We use a 2-approximation algorithm for RSA, for example, the one of Rao et al. [8]. This yields the following.

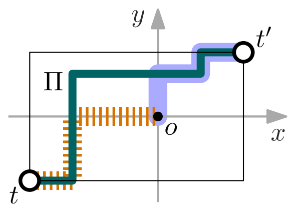



**Figure 2:** Network $N$ connects $t$ to $o$ (dashed path) and $t'$ to $o$ (solid path).

**Lemma 4** *$xy$-separated 2D-GMMN admits a constant-factor approximation.*

**Proof.** Let $T$ be the set of terminals in the $xy$-separated GMMN instance $R$ and $N'$ be an RSA network M-connecting $T$ to $o$, which we compute using the 2-approximation for RSA.

We first claim that $N'$ is a feasible GMMN solution for $R$. To see this, note that $N'$ contains, for every terminal pair $(t, t') \in R$, an M-path $\pi$ from $t$ to $o$ and an M-path $\pi'$ from $o$ to $t'$. Concatenating $\pi$ and $\pi'$ yields an M-path from $t$ to $t'$ as the bounding box of $(t, t')$ contains $o$.

Let OPT denote the cost of an optimal GMMN solution for $R$. We claim there is a solution $N$ for the RSA instance of M-connecting $T$ to $o$ of cost $O(\text{OPT})$.

Let $N^{\text{opt}}$ be an optimum solution to $R$. Let $N$ be the union of $N^{\text{opt}}$ and the projections of $N^{\text{opt}}$ to the $x$-axis and to the $y$-axis. The total length of $N$ is $\|N\| \leq 2 \cdot \text{OPT} = O(\text{OPT})$ since every line segment of $N^{\text{opt}}$ is projected either to the $x$-axis or to the $y$-axis but not to both. The crucial fact about $N$ is that this network contains, for every terminal $t$ in $R$, an M-path from $t$ to the *origin* $o$. In other words, $N$ is a feasible solution to the RSA instance of M-connecting $T$ to $o$.

To see this, consider an arbitrary terminal pair $(t, t') \in R$. Let $\Pi$ be an M-path connecting $t$ and $t'$ in $N^{\text{opt}}$; see Fig. 2. Note that, since the bounding box of $(t, t')$ contains $o$, $\Pi$ intersects both $x$- and $y$-axis. To obtain an M-path from $t$ to $o$, we follow $\Pi$ from $t$ to $t'$ until $\Pi$ crosses one of the axes. From that point on, we follow the *projection* of $\Pi$ onto this axis. We reach $o$ when $\Pi$ crosses the other axis; see the dotted path in Fig. 2. Analogously, we obtain an M-path from $t'$ to $o$.

Finally, as there is a feasible RSA solution $N$ for terminals $T$ of cost $O(\text{OPT})$, the RSA solution $N'$ that we compute costs at most $2\|N\| = O(\text{OPT})$. ☐

Our algorithm for $d$ dimensions is a generalization of the algorithm for 2D-GMMN. For each dimension, we loose a $(\log n)$-factor in the approximation ratio. The final problem instances can be solved using an adaptation to $d$ dimensions of the $O(\log n)$-approximation algorithm of Rao et al. [8] for 2D-RSA; see the long version [3].

**Theorem 5** *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.*

## 3 Improved Algorithm for Two Dimensions

In this section, we show that 2D-GMMN admits an $O(\log n)$-approximation, which improves upon the $O(\log^2 n)$-result of Section 2. To this end, we develop a $(6+\varepsilon)$-approximation algorithm for $x$-separated 2D-GMMN, for any $\varepsilon > 0$; see Lemma 8. Together with Lemma 2, we obtain the following.

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

Our constant-factor approximation for $x$-separated 2D-GMMN and its analysis are technically more involved. Therefore, we only sketch the underlying approach; see the long version [3] for details.

Let $R$ be the set of terminal pairs of an $x$-separated instance of 2D-GMMN. We assume, w.l.o.g., that each terminal pair $(l, r) \in R$ is separated by the $y$-axis, such that $x(l) < 0 \le x(r)$. Let $N^{\text{opt}}$ be an optimum solution to $R$. Let $\text{OPT}_{\text{ver}}$ and $\text{OPT}_{\text{hor}}$ be the total costs of the vertical and horizontal segments in $N^{\text{opt}}$, respectively. Hence, $\text{OPT} = \text{OPT}_{\text{ver}} + \text{OPT}_{\text{hor}}$.

The algorithm consists of two stages: a *stabbing* stage and a *connection* stage. In the stabbing stage, we compute a set $S$ of horizontal line segments such that each rectangle in $R$ is completely *stabbed* by some line segment in $S$. More precisely, for each rectangle $r$ there is a horizontal line segment $h \in S$ such that the intersection of $r$ and $h$ equals the intersection of $r$ with the supporting line of $h$. We can show the following [3].

**Lemma 7** *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 $R$.*

The *connection* stage M-connects the terminals to the $y$-axis so that the resulting network, along with the stabbing $S$, forms a feasible solution to $R$ of cost $O(\text{OPT})$. To this end, 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 $y(t) > y(t')$ for some $t' \in T$. Note that, in general, $L$ and $H$ are not disjoint.

Using a PTAS for 2D-RSA [6], we compute a near-optimal RSA network $A_{\text{up}}$ connecting the terminals in $L$ to $\text{top}(I)$ and a near-optimal RSA network $A_{\text{down}}$ connecting the terminals in $H$ to $\text{bot}(I)$. Then we return the network $N = A_{\text{up}} \cup A_{\text{down}} \cup S$, where $S$ is the stabbing computed by the stabbing stage.

In the long version [3], we show that the resulting network is a feasible solution to $R$, with cost at most constant times OPT.

**Lemma 8** *$x$-separated 2D-GMMN admits, for any $\varepsilon > 0$, a $(6 + \varepsilon)$-approximation.*

## References


[1] V. Chepoi, K. Nouioua, and Y. Vaxès. A rounding algorithm for approximating minimum Manhattan networks. *Theor. Comput. Sci.*, 390(1):56–69, 2008.

[2] F. Chin, Z. Guo, and H. Sun. Minimum Manhattan network is NP-complete. *Discrete Comput. Geom.*, 45:701–722, 2011.

[3] A. Das, K. Fleszar, S. G. Kobourov, J. Spoerhase, S. Veeramoni, and A. Wolff. Polylogarithmic approximation for generalized minimum Manhattan networks. Arxiv report, Apr. 2012. Available at http://arxiv.org/abs/1203.6481.

[4] A. Das, E. R. Gansner, M. Kaufmann, S. Kobourov, J. Spoerhase, and A. Wolff. Approximating minimum Manhattan networks in higher dimensions. In *Proc. 19th Annu. Europ. Symp. on Algorithms (ESA'11)*, volume 6942 of *LNCS*, pages 49–60. Springer, 2011.

[5] Z. Guo, H. Sun, and H. Zhu. Greedy construction of 2-approximate minimum Manhattan networks. *Int. J. Comput. Geom. Appl.*, 21(3):331–350, 2011.

[6] B. Lu and L. Ruan. Polynomial time approximation scheme for the rectilinear Steiner arborescence problem. *J. Comb. Optim.*, 4(3):357–363, 2000.

[7] X. Muñoz, S. Seibert, and W. Unger. The minimal Manhattan network problem in three dimensions. In *Proc. 3rd Int. Workshop Algorithms Comput. (WALCOM'09)*, volume 5431 of *LNCS*, pages 369–380. Springer, 2009.

[8] S. Rao, P. Sadayappan, F. Hwang, and P. Shor. The rectilinear Steiner arborescence problem. *Algorithmica*, 7:277–288, 1992.

[9] W. Shi and C. Su. The rectilinear Steiner arborescence problem is NP-complete. *SIAM J. Comput.*, 35(3):729–740.