Julius-Maximilians-Universität Würzburg
Institut für Informatik
Lehrstuhl für Informatik I
Effiziente Algorithmen und wissensbasierte Systeme

Masterthesis

Generalized Minimum Manhattan Networks

Krzysztof Fleszar

May 18, 2012

Thesis advisors:
Dr. Joachim Spoerhase
Prof. Dr. Alexander Wolff

Contents

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.

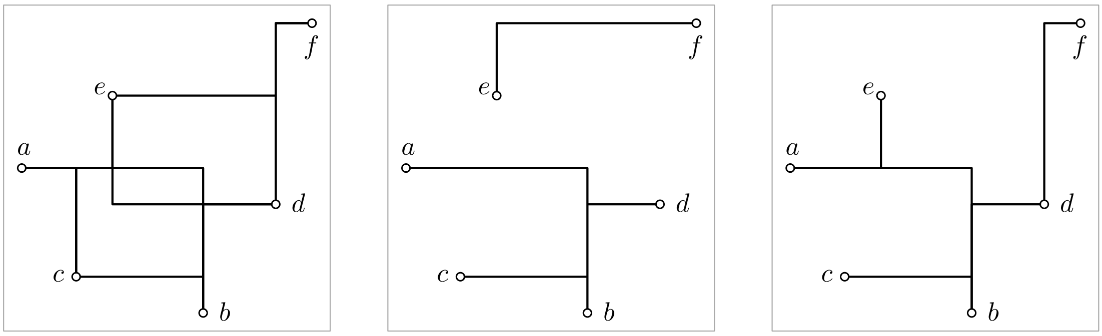
In the *generalized minimum Manhattan network problem* (GMMN), we are given a set R of n unordered terminal *pairs*, and the goal is to find a minimum-length rectilinear network such that every pair in R is *M-connected*, that is, connected by an M-path. GMMN is a generalization of MMN since R may contain all possible pairs of terminals. Figures 1.1a and 1.1b depict examples of both network types.

We remark that in this thesis 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*.

Two-dimensional MMN (2D-MMN) naturally arises in VLSI circuit layout [GLN01], 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. [CLZ93] 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.

MMN requires a Manhattan path between every terminal pair. This assumption is, however, not always reasonable. Specifically 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.

Previous Work. MMN was introduced by Gudmundsson et al. [GLN01] who gave 4- and 8-approximation algorithms for 2D-MMN running in $O(n^3)$ and $O(n \log n)$ time, respectively. Benkert et al. [BWWS06] presented a mixed-integer formulation and a 3-approximation. Fuchs *et al.* [FS08] provide a simpler 3-approximation algorithm. The currently known best approximation algorithms for 2D-MMN have ratio 2; they were obtained independently by Chepoi et al. [CNV08] using an LP-based method, by Nouioua [Nou05] using a primal-dual scheme, and by Guo et al. [GSZ11] using a greedy



(a) A minimum Manhattan network for $\{a, b, c, d, e, f\}$.

(b) A generalized minimum Manhattan network for $\{(a,b), (c,d), (e,f)\}$.

(c) A rectilinear Steiner arborescence for $(\{a, b, c, d, e, f\}, b)$.

Figure 1.1.: MMN versus GMMN versus RSAP in 2D

approach. The complexity of 2D-MMN was settled only recently by Chin et al. [CGS11]; they proved the problem NP-hard. It is not known whether 2D-MMN is APX-hard.

Less is known about MMN in dimensions greater than 2. Muñoz et al. [MSU09] 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. More recently, Das et al. described the first approximation algorithm for MMN in arbitrary, fixed dimension with a ratio of $O(n^{\varepsilon})$ for any $\varepsilon > 0$ [DGK$^+$11].

GMMN was defined by Chepoi et al. [CNV08] who asked whether 2D-GMMN admits an constant-factor approximation. Apart from the formulation of this open problem, only special cases of GMMN (such as MMN) have been considered in the literature so far.

One such special case (other than MMN) that has received significant attention in the past is the *rectilinear Steiner arborescence problem* (RSAP). 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; see Figure 1.1c. Hence, RSAP 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). RSAP was introduced by Nastansky et al. [NSS74] and has mainly been studied in 2D. 2D-RSAP was proved NP-hard by Shi et al. [SS00]. Rao et al. [RSHS92] gave a 2-approximation algorithm for 2D-RSAP. 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 Section 2.4). Lu et al. [LR00] and, independently, Zachariasen [Zac00] described polynomial-time approximation schemes (PTAS) for RSAP, both based on Arora's technique [Aro03]. Zachariasen pointed out that his PTAS can be generalized to the all-quadrant version of RSAP, but that it seems difficult to extend the approach to higher dimensions.

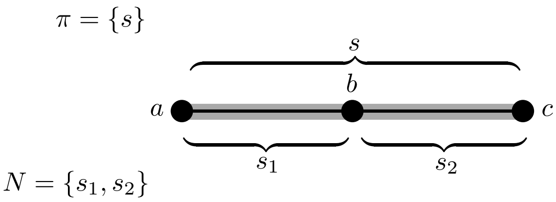
Contribution of the Thesis. The main result achieved in this thesis is an $O(\log^{d+1} n)$- -approximation algorithm for GMMN (and, hence, MMN) in dimension d. For two-dimensional GMMN, we improve the approximation ratio to $O(\log n)$. The key idea of our algorithms is to partition the problem instance and then to reduce the sub-instances to RSAP. As a byproduct we obtain that all-orthant RSAP in fixed dimension d admits an $O(\log n)$-approximation. To the best of our knowledge, this is the currently best result for RSAP in dimension $d \geq 3$. We present these results in Chapter 2.

Besides the general problem, we examine three special cases of GMMN in Chapter 3. We deduce that GMMN in any dimension $d > 0$ admits the same approximation ratio as f-dimensional GMMN when some characteristics of the instance are restricted to dimension f. These characteristic concern the possible positions of the terminals pairs in the Euclidean space, as well as the maximum dimension of their bounding boxes. We also present an $O(\log^2 n)$-approximation algorithm for the case where the aspect ratio of each bounding box that defines a terminal pair is bounded by some constant.

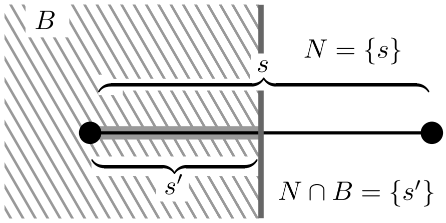
Credits. The algorithms for the general case of GMMN are fruits of joint research. Aparna Das, Stephen Kobourov, Joachim Spoerhase, Sankar Veeramoni, Alexander Wolff and myself developed these ideas together and, recently, solidified them in a joint paper [DFK+12] (yet to be published). Thus, Chapter 2 corresponds in many parts to the paper. Here, my personal contribution especially concerns the final reduction step of GMMN to RSAP (Section 2.2.2), the approximation of d-dimensional RSAP (Section 2.4), as well as the improvements in the two-dimensional case (Section 2.3.3) and its limits (Section 2.3.5). The algorithm in Section 3.3 that regards the aspect ratio of bounding boxes is based on an idea of Esther Arkin and Joseph Mitchell.

Notations. In the following chapters, we will identify each terminal pair with its d-dimensional *bounding box*, that is, the smallest axis-aligned box that contains both terminal pairs. Consequently, we will consider any GMMN instance R also as a set of d-dimensional boxes.

For our use, a *network* is a set of axis-aligned line segments. We say that a network N *contains* a rectilinear path π when the point set of π is entirely contained in the point set of N; see Figure 1.2a. Further, for any (unbounded) box B (for example, a half-plane or an axis-spanned subspace) and any network N, we define their intersection $B \cap N$ as a finite network N' whose point set equals the intersection of B and the point set of N; see Figure 1.2b. For short, we will refer by *GMN* to a rectilinear network that M-connects all terminal pairs of some given instance R and call it a *minimum GMN* if it is an optimum GMMN solution to R. A *rectilinear Steiner arborescence (RSA)* is a feasible but not necessarily minimum solution to RSAP. To emphasize that such a solution is a constant-factor approximation, we will refer to it by *near-optimal RSA*. In Table A.1 (see Appendix A) we summarize the most important notions used in this thesis.



(a) The M-path π connects a and c with one segment s. The network N consists of two segments, s_1 between a and b, and s_2 between b and c. Thus, $\pi \not\subseteq N$, as $s \neq s_1$ and $s \neq s_2$. However, the point set of N contains all points of the point set of π. Hence, π is *contained* in N.



(b) The network N consists of one segment s. Its intersection with the half-plane B (hatched area) yields the network containing only $\{s'\}$, where $s' = s \cap B$.

Figure 1.2.: Point Sets of Networks and Paths.

2. Polylogarithmic Approximation

In this chapter we study the general case of GMMN in dimension two and higher. Our main result is an $O(\log^{d+1} n)$-approximation algorithm for GMMN in dimension d. For the sake of simplicity, we first present our approach in 2D (see Section 2.1) and then show how it can be generalized to higher dimensions (see Section 2.2). We also provide an improved and technically more involved $O(\log n)$-approximation for the special case of 2D-GMMN, but this approach does not seem to generalize to higher dimensions; see Section 2.3. We show the tightness of our analysis in Section 2.3.4 and in Section 2.3.5 we discuss limits of the approach and show that it can't be easily modified to yield a $O(1)$-approximation. Finally, in Section 2.5 we provide a running-time analysis.

To the best of our knowledge, we present the first approximation algorithms for GMMN. Our result for 2D is not quite the constant-factor approximation that Chepoi et al. were asking for, but it is a considerable step into that direction. Note that the poly-logarithmic ratio of our algorithm for GMMN in dimension $d \geq 3$ constitutes an exponential improvement upon the previously only known approximation algorithm, which solves the special case MMN, with a ratio of $O(n^{\varepsilon})$ for any $\varepsilon > 0$ [DGK$^+$11].

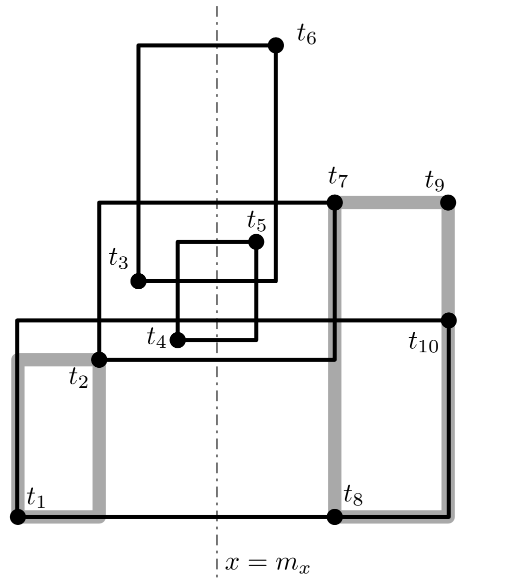
Our algorithm for GMMN is based on divide and conquer. We use $(d-1)$-dimensional hyperplanes to partition R recursively into sub-instances. The base case of our partition scheme consists of GMMN instances where all boxes contain a common point. We solve the resulting special case of GMMN by reducing it to RSAP. Here we heavily depend on the approximability of d-dimensional RSAP. In Section 2.4 we generalize a known algorithm for one-quadrant 2D-RSAP to yield an $O(\log n)$-approximation for the all-orthant case in arbitrary fixed dimension.

2.1. Approximation for Two-Dimensional GMMN

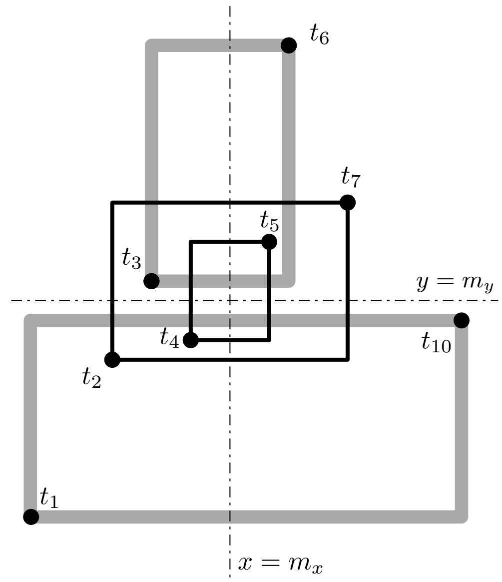
In this section, we present an $O(\log^2 n)$-approximation algorithm for 2D-GMMN. The algorithm consists of a *main algorithm* that recursively subdivides the input instance into instances of so-called *x-separated* GMMN; see Section 2.1.1. We prove that the instances of x-separated GMMN can be solved independently by paying a factor of $O(\log n)$ in the approximation ratio. Then we show how to approximate x-separated GMMN within ratio $O(\log n)$; see Section 2.1.2. This yields an overall ratio of $O(\log^2 n)$.

2.1.1. Main Algorithm

Our approximation 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.



(a) The dashed line is the median m_x that is used to partition the set into $R_{\text{left}}, R_{\text{mid}}, R_{\text{right}}$. The thick gray rectangles belong to $R_{\text{left}} \cup R_{\text{right}}$.



(b) The x-separated instance R_{mid} is shown along with the median line m_y. Again, the thick gray rectangles belong to $R_{\text{left}} \cup R_{\text{right}}$.

Figure 2.1.: The division step of main algorithm on an instance of 2D-GMMN. An instance of GMMN with $R = \{(t_1, t_2), (t_1, t_{10}), (t_2, t_7)(t_3, t_6), (t_8, t_9), (t_4, t_5)\}$.

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 $x = m_x$; see Figure 2.1a.

Our algorithm divides R into three subsets $R_{\text{left}}, R_{\text{mid}}$, and R_{right}. The set R_{left} consists of all rectangles that lie *completely* to the left of the vertical line m_x. Similarly, the set R_{right} consists of all rectangle that lie *completely* to the right of m_x. The set 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 and apply our algorithm recursively to R_{left} and to R_{right}. The union of the two resulting networks is a rectilinear network that M-connects all terminal pairs in $R_{\text{left}} \cup R_{\text{right}}$.

It remains to M-connect the pairs in R_{mid}. We call an 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 2.1.2. Later, in Section 2.3, we improve upon this and describe an $O(1)$-approximation algorithm for x-separated GMMN.

To analyze the performance of our main algorithm, let $\rho(n)$ denote the 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* $\rho(n) \cdot \text{OPT}$, where OPT denotes the cost of an optimum solution N_{opt} to R. Let N_{left} and N_{right} be the parts of N_{opt} to the left and to the right of m_x, respectively.

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_{left} is a feasible solution to R_{left}, we conclude that the cost of the solution to R_{left} computed by our algorithm is bounded by $\rho(n/2) \cdot \|N_{\text{left}}\|$, where $\|\cdot\|$ measures the length of a network. Analogously, the cost of the solution computed for R_{right} is bounded by $\rho(n/2) \cdot \|N_{\text{right}}\|$. Now we assume that we can approximate x-separated instances with a ratio of $\rho_x(n)$. 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}.

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

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

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_{left} and N_{right} are separated by m_x, hence they are edge disjoint and $\|N_{\text{left}}\| + \|N_{\text{right}}\| \leq \text{OPT}$. This yields the recurrence $\rho(n) \leq \rho(n/2) + \rho_x(n)$, which resolves to $\rho(n) = \log n \cdot \rho_x(n)$. Let's summarize this discussion.

Lemma 2.1. *If x-separated 2D-GMMN admits a $\rho_x(n)$-approximation, 2D-GMMN admits a $(\rho_x(n) \cdot \log n)$-approximation.*

Combining this lemma with our $O(\log n)$ approximation algorithm for x-separated instances described below, we obtain the following intermediate result.

Theorem 2.2. *2D-GMMN admits an $O(\log^2 n)$-approximation.*

2.1.2. Approximating x-Separated Instances

In this section, we describe a simple algorithm for approximating x-separated 2D-GMMN instances with a ratio of $O(\log n)$. Let R be our input. Since R is x-separated, all rectangles in R intersect a common vertical line. Without loss of generality, this is the y-axis.

The algorithm works as follows. Analogously to the main algorithm presented in Section 2.1.1, we recursively subdivide the x-separated input instance, but this time according to the *y-coordinate*; see Figure 2.1b. As a result of this, the input instance R is decomposed into *y-separated* sub-instances. Moreover, since each of these sub-instances is (as a subset of R) already x-separated, we call these instances *xy-separated*. In Section 2.1.3, we give a specialized algorithm for xy-separated instances.

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 Section 2.1.3, we show that $\rho_{xy}(n) = O(1)$. Then Lemma 2.1 (by exchanging x- and y-coordinates) implies that $\rho_x(n) = \log n \cdot \rho_{xy}(n) = O(\log n)$.

Lemma 2.3. *x-separated 2D-GMMN admits an $O(\log n)$-approximation.*

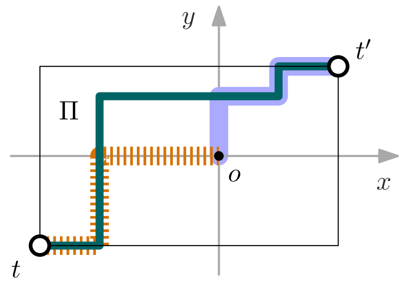


Figure 2.2.: Network N connects all terminals to o.

2.1.3. Approximating xy-Separated Instances

It remains to show that xy-separated GMMN instances can be approximated within a constant ratio. Let R be such an instance. We assume, without loss of generality, that it is the x- and the y-axis that intersect all rectangles in R, that is, all rectangles contain the origin. Let N_{opt} be an optimum solution to R. Let N be the union of N_{opt} and the projections of N_{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_{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 RSAP instance of M-connecting every terminal in R to o.

To see this, consider an arbitrary terminal pair $(t, t') \in R$. Let Π be an M-path connecting t and t' in N_{opt}; see Fig. 2.2. Note that, since the bounding box of (t, t') contains o, Π intersects both x- and y-axis. To obtain an M-path from t to o, we follow Π from t to t' until Π crosses one of the axes. From that point on, we follow the *projection* of Π on this axis. We reach o when Π crosses the other axis; see the dotted path in Fig. 2.2. Analogously, we obtain an M-path from t' to o.

Let T be the set of terminals in R. We have shown above that there is a feasible solution N of cost $O(\text{OPT})$ to the RSAP instance with terminal set T. There is a PTAS for RSAP in dimension two [Zac00, LR00]. Using this PTAS, we can efficiently *compute* a feasible RSAP solution N' for T of cost $O(1) \cdot \|N\| = O(\text{OPT})$. Moreover, N' is also a feasible solution to the GMMN instance R. To see this, note that N' contains, for every terminal pair $(t, t') \in R$, an M-path π from t to o and an M-path π' from o to t'. Concatenating π and π' yields an M-path from t to t' as the bounding box of (t, t') contains o. Thus we obtain the following result.

Lemma 2.4. *xy-separated 2D-GMMN admits a constant-factor approximation.*

2.2. Generalization to Higher Dimensions

In this section, we describe an $O(\log^{d+1} n)$-approximation algorithm for GMMN in d dimensions, which is a generalization of the algorithm for dimension two presented in Section 2.1. Let us view this algorithm from the following perspective. In Section 2.1.1, we reduced GMMN to solving x-separated sub-instances at the expense of a $(\log n)$-factor in the approximation ratio (see Lemma 2.1). Applying the same lemma to the y-coordinates in Section 2.1.2, we further reduced the problem to solving xy-separated sub-instances, that is, to instances that were separated with respect to both coordinates. This caused the second $(\log n)$-factor in our approximation ratio. Finally, we were able to approximate these completely separated sub-instances within constant ratio by solving a related RSAP problem (see Section 2.1.3).

These ideas generalize to higher dimensions. An instance R of d-dimensional GMMN is called *j-separated* for some $j \leq d$ if there exist a set J of j coordinates and a point s such that, for each terminal pair $(t, t') \in R$ and for each coordinate $x_i \in J$ we have that s *separates* t and t' in coordinate x_i (meaning that either $x_i(t) \leq x_i(s) \leq x_i(t')$ or $x_i(t') \leq x_i(s) \leq x_i(t)$). We call such a point s a j-separator of R. Under this terminology, an arbitrary instance of d-dimensional GMMN is always *0-separated*. Without loss of generality, we assume in this chapter that a *j-separated* GMMN instance is separated with respect to coordinates x_1 to x_j.

We first show that if we can approximate j-separated GMMN with ratio $\rho_j(n)$ then we can approximate $(j-1)$-separated GMMN with ratio $\rho_j(n) \cdot \log n$; see Section 2.2.1. Then we show that d-separated GMMN can be approximated within a factor $\rho_d(n) = O(\log n)$; see Section 2.2.2. Combining these two facts and applying them inductively to arbitrary (that is, 0-separated) GMMN instances yields the following central result of this thesis.

Theorem 2.5. *GMMN in dimension d admits an $O(\log^{d+1} n)$-approximation.*

As a byproduct of this algorithm, 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 2.6. *MMN in dimension d admits an $O(\log^{d+1} n)$-approximation, where n denotes the number of terminals.*

2.2.1. Separation

In this section, we show that if we can approximate j-separated GMMN instances with ratio $\rho_j(n)$, we can approximate $(j-1)$-separated instances with ratio $\log n \cdot \rho_j(n)$. The separation algorithm and its analysis work analogously to the main algorithm for 2D where we reduced (the approximation of) 2D-GMMN to (the approximation of) x-separated 2D-GMMN; see Section 2.1.1

Let R be a set of $(j-1)$-separated terminal pairs. Let m_x be the median in the multiset of the j-th coordinates of terminals. We divide R into three subsets R_{left}, R_{mid}, and R_{right}. The set R_{left} consists of all terminal pairs (t, t') such that $x_j(t), x_j(t') \leq m_x$ and R_{right} contains all terminal pairs (t, t') with $x_j(t), x_j(t') \geq m_x$. The set R_{mid} contains the remaining terminal pairs, all of which are separated by the hyperplane $x_j = m_x$. We apply our algorithm recursively to R_{left} and R_{right}. The union of the resulting networks is a rectilinear network that M-connects all terminal pairs $R_{\text{left}} \cup R_{\text{right}}$.

In order to M-connect the pairs in R_{mid}, we apply an approximation algorithm for j-separated GMMN of ratio $\rho_j(n)$. Note that the instance R_{mid} is in fact j-separated by construction. The analysis of the resulting algorithm for $(j-1)$-separated GMMN is analogous to the 2D-case (see Section 2.1.1) and is therefore omitted.

Theorem 2.7. *Let $1 \leq j \leq d$. If j-separated GMMN admits a $\rho_j(n)$-approximation, then $(j-1)$-separated GMMN admits a $(\rho_j(n) \cdot \log n)$-approximation.*

2.2.2. Approximating d-Separated Instances

In this section, we show that we can approximate instances of d-separated GMMN within a ratio of $O(\log n)$ by reducing the problem to RSAP. Let R be a d-separated instance and let T be the set of all terminals in R. As R is d-separated, all bounding boxes defined by terminal pairs in R contain a common point, which is, without loss of generality, the origin.

As in the two-dimensional case (see Section 2.1.3), we M-connect all terminals to the origin by solving an RSAP instance with terminal set T. This yields a feasible GMMN solution to R since for each pair $(t, t') \in R$ there is an M-path from t to the origin and an M-path from the origin to t'. The union of these paths is an M-path from t to t' since the origin is contained in the bounding box of (t, t').

Rao et al. [RSHS92] presented an $O(\log |T|)$-approximation algorithm for 2D-RSAP, which generalizes, in a straight-forward manner, to d-dimensional RSAP; see Section 2.4 for details. Hence, we can use the algorithm of Rao et al. to efficiently compute a feasible GMMN solution. The following lemma shows that this solution is in fact an $O(\log n)$-approximation. The proof is similar to the proof of Lemma 7 in the paper of Das et al. [DGK+11].

Lemma 2.8. *d-separated GMMN admits an $O(\log n)$-approximation for any fixed dimension d.*

Proof. Below we show that there is a solution of cost $O(\mathrm{OPT})$ to the RSAP instance connecting T to the origin. Observing that $|T| \leq 2n$ and using our extension of the $O(\log |T|)$-approximation algorithm of Rao et al. (see Section 2.4), we can efficiently compute a minimum GMN of cost $O(\mathrm{OPT} \cdot \log n)$.

Let N_{opt} be an optimal GMMN solution to R and let N be the projection of N_{opt} onto all subspaces that are spanned by some subset of the coordinate axes. Since there are 2^d such subspaces, which is a constant for fixed d, the cost of N is $O(\mathrm{OPT})$.

It remains to show that N M-connects all terminals to the origin, that is, N is a feasible solution to the RSAP instance. First, note that $N_{\mathrm{opt}} \subseteq N$ since we project on the d-dimensional space, too. Now consider an arbitrary terminal pair (t, t') in R and an M-path π in N_{opt} that connects t and t'. Starting at t, we follow π until we reach the first point p_1 where one of the coordinates becomes zero. Without loss of generality, $x_1(p_1) = 0$. Clearly π contains such a point as the bounding box of (t, t') contains the origin. We observe that p_1 lies in the subspace spanned by the $d-1$ coordinate axes $x_2, \ldots, x_d$. From p_1 on we follow the projection of π onto this subspace until we reach the first point p_2 where another coordinate becomes zero; without loss of generality, $x_2(p_2) = 0$. Hence, p_2 has at least two coordinates that are zero, that is, p_2 lies in a subspace spanned by only $d-2$ coordinate axes. Iteratively, we continue following the projections of π onto subspaces with decreasing dimension until every coordinate is zero, that is, we have reached the origin. An analogous argument shows that N also contains an M-path from t' to the origin. □

2.3. Improved Algorithm for Two-Dimensional GMMN

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 2.1. To this end, we develop a $(6+\varepsilon)$-approximation algorithm for x-separated 2D-GMMN, for any $\varepsilon > 0$. While the algorithm is simple, its analysis turns out to be quite intricate. In Section 2.3.4, we show that our analysis is tight and in Section 2.3.5 we point our problems when generalizing this approach to 2D-GMMN that is not x-separated. Using Lemma 2.1, our new subroutine for the x-separated case yields the following.

Theorem 2.9. *2D-GMMN admits a $((6+\varepsilon)\cdot\log n)$-approximation.*

Let R be the set of terminal pairs of an x-separated instance of 2D-GMMN. We assume, without loss of generality, that each terminal pair $(l, r) \in R$ is separated by the y-axis, such that $x(l) < 0 \leq x(r)$. Let N_{opt} be an optimum solution to R. Let OPT_{ver} and OPT_{hor} be the total costs of the vertical and horizontal segments in N_{opt}, respectively. Hence, $\text{OPT} = \text{OPT}_{\text{ver}} + \text{OPT}_{\text{hor}}$. We first compute a set S of horizontal line segments of total cost $O(\text{OPT}_{\text{hor}})$ such that each rectangle in R is *stabbed* by some line segment in S; see Sections 2.3.1 and 2.3.2. Then we M-connect the terminals to the y-axis so that the resulting network, along with the affected part of the y-axis and the stabbing S, forms a feasible solution to R of cost $O(\text{OPT})$; see Section 2.3.3.

2.3.1. 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 straight line through 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_{opt}, and let N^+_{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. Note that, for every $x \geq 0$, the intersection of R^+ with ℓ_x forms a set $\mathcal{I}_x$ of intervals. The intersection of N^+_{hor} ($\bigcup N^+_{\text{hor}}$ to be precise) with ℓ_x is, at any time, a set of points that constitutes a *piercing* for $\mathcal{I}_x$, that is, every interval in $\mathcal{I}_x$ contains a point in $\ell_x \cap N^+_{\text{hor}}$. Note that $\|N^+_{\text{hor}}\| = \int |\ell_x \cap N^+_{\text{hor}}| dx$.

We imagine that we continuously move ℓ_x from $x = 0$ to the right. At any time, we maintain an inclusion-wise minimal piercing P_x of $\mathcal{I}_x$. With increasing x, we only *remove* points from P_x; we never add points. This ensures that the traces of the points in P_x form horizontal line segments that all touch the y-axis. These line segments form our stabbing of R^+.

The algorithm proceeds as follows. It starts at $x := 0$ with an arbitrary minimal piercing P_0 of $\mathcal{I}_0$. Note that we can even compute an optimum piercing P_0. We must adapt P_x whenever $\mathcal{I}_x$ changes. With increasing x, $\mathcal{I}_x$ decreases inclusion-wise since all rectangles in R^+ touch the y-axis. So it suffices to adapt the piercing P_x 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.

It is clear that the algorithm produces a stabbing of R^+; see the thick solid line segments in Fig. 2.3a. The following lemma is crucial to prove the overall cost of the stabbing.

Lemma 2.10. *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 interval* $I_p \in \mathcal{I}_x$ 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 witness intervals of at most two points in P_x. Assume, for the sake of contradiction, that q is contained in the witness intervals of points $p, p', p'' \in P_x$ with strictly increasing y-coordinates. Suppose that q lies above p'. But then the witness interval 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.

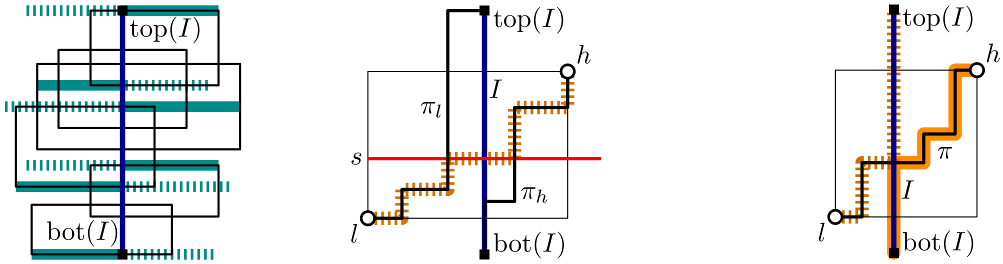
Recall that $\ell_x \cap N^+_{\text{hor}}$ is a piercing of $\mathcal{I}_x$ and, hence, of the $|P_x|$ many witness intervals. Since every point in $\ell_x \cap N^+_{\text{hor}}$ pierces at most two witness intervals, the lemma follows. □

Observe that the cost of the stabbing is $\int |P_x| dx$. By the above lemma, the cost of the stabbing can be bounded by $\int |P_x| dx \leq \int 2 \cdot |\ell_x \cap N^+_{\text{hor}}| dx = 2 \cdot \|N^+_{\text{hor}}\|$, which proves the following lemma.

Lemma 2.11. *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^+.*

2.3.2. Stabbing Both Parts

We now detail how we construct a stabbing of R. To this end we apply Lemma 2.11 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



(a) S^+ stabs R^+, and S stabs R.

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

(c) $N_{\mathrm{opt}} \cup \{I\}$ is feasible for RSAP instances $(L, \mathrm{top}(I))$, $(H, \mathrm{bot}(I))$.

Figure 2.3.: The improved algorithm for x-separated 2D-GMMN.

by one segment. To overcome this difficulty, we mirror S^- and S^+ to the respective other side of the y-axis; see Fig. 2.3a. The total cost of the resulting set S of horizontal line segments is at most $4(\|N^-_{\mathrm{hor}}\| + \|N^+_{\mathrm{hor}}\|) = 4 \cdot \mathrm{OPT}_{\mathrm{hor}}$. 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 $s \cup s'$. Let us summarize.

Lemma 2.12. *Given a set R of rectangles intersecting the y-axis, we can compute a set of horizontal line segments of cost at most $4 \cdot \mathrm{OPT}_{\mathrm{hor}}$ that stabs R.*

2.3.3. 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 $\mathrm{top}(I)$ and $\mathrm{bot}(I)$ be the top and bottom endpoints of I, respectively. Let L be the set containing every terminal t with $(t, t') \in R$ and $y(t) \le y(t')$. Symmetrically, let H be the set containing every terminal t with $(t, t') \in R$ and $y(t) > y(t')$. Note that L and H are not necessarily disjoint.

Using a PTAS for RSAP [LR00, Zac00], we compute a near-optimal RSA A_{up} connecting the terminals in L to $\mathrm{top}(I)$ and a near-optimal RSA 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 2.3.2.

We now show that this network is a feasible solution to R. Let $(l, h) \in R$. Without loss of generality, $l \in L$ and $h \in H$[1]. Hence, A_{up} contains a path π_l from l to $\mathrm{top}(I)$, see Fig. 2.3b. This path starts inside the rectangle (l, h) leaving (l, h), the path intersects a line segment s in S that stabs (l, h). This line segment 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.

[1] If $l, h \in L$ then $y(t) = y(t')$ and the line segment stabbing (l, h) is already an M-path for them.

Now, let us analyze the cost of N. Clearly, the projection of N_{opt} onto the y-axis yields the line segment I. Hence, $|I| \leq \text{OPT}_{\text{ver}}$. Observe that $N_{\text{opt}} \cup \{I\}$ constitutes a solution to the RSAP instance $(L, \text{top}(I))$ connecting all terminals in L to $\text{top}(I)$ and to the RSAP instance $(H, \text{bot}(I))$ connecting all terminals in H to $\text{bot}(I)$. This holds since, for each terminal pair, its M-path π in N_{opt} crosses the y-axis in I; see Fig. 2.3c. Since A_{up} and A_{down} are near-optimal solutions to these RSAP instances, we obtain, for any $\varepsilon > 0$, that $\|A_{\text{up}}\| \leq (1+\varepsilon) \cdot \|N_{\text{opt}} \cup I\| \leq (1+\varepsilon) \cdot (\text{OPT} + \text{OPT}_{\text{ver}})$ and analogously $\|A_{\text{down}}\| \leq (1+\varepsilon) \cdot (\text{OPT} + \text{OPT}_{\text{ver}})$.

By Lemma 2.12, we have $\|S\| \leq 4 \cdot \text{OPT}_{\text{hor}}$. Assuming $\varepsilon \leq 1$, this yields

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

for $\varepsilon' = \varepsilon/2$, which we can make arbitrarily small by making ε arbitrarily small. We summarize our result as follows.

Lemma 2.13. *x-separated 2D-GMMN admits, for any $\varepsilon > 0$, a $(6+\varepsilon)$-approximation.*

2.3.4. Tightness of Our Analysis

In this section we prove that our analysis of the approximation ratio for the improved algorithm is indeed a tight result. We show this by providing an example.

Observation 2.14. *There are infinitely many instances where the $O(\log n)$-approximation algorithm for 2D-GMMN has approximation performance $\Omega(\log n)$.*

Proof. We recursively define an arrangement $A(n)$ of n rectangles each of which represents a terminal pair; the lower left and upper right corner of the rectangle. By $\alpha \cdot A(n)$ we denote the arrangement $A(n)$ but uniformly scaled in both dimensions so that it fits into an $\alpha \times \alpha$ square. Let $\varepsilon > 0$ be a sufficiently small number.

The arrangement $A(0)$ is empty. The arrangement $A(n)$ consists of a unit square S_n whose upper right vertex is the origin. We add the arrangement $A_{\text{right}} := \varepsilon \cdot A((n-1)/2)$ and place it in the first quadrant at distance ε to the origin. Finally, we add the arrangement $A_{\text{left}} := (1-\varepsilon) \cdot A((n-1)/2)$ inside the square S_n so that it does not touch the boundary of S_n. See Fig. 2.4 for an illustration.

Let $\rho(n)$ denote the cost produced by our algorithm when applied to $A(n)$. Observe that our algorithm partitions $A(n)$ into subinstances $R_{\text{left}} = A_{\text{left}}$, $R_{\text{mid}} = \{S_n\}$, and $R_{\text{right}} = A_{\text{right}}$. Solving the x-separated instance R_{mid} by our stabbing subroutine costs 1. Let $\rho(n)$ be the cost of the solution to $A(n)$ that our algorithm computes. Recursively solving R_{left} costs $(1-\varepsilon) \cdot \rho((n-1)/2)$. Recursively solving R_{right} costs $\varepsilon \cdot \rho((n-1)/2)$. Hence, the cost of the solution of our algorithm is $\rho(n) \geq 1 + \rho((n-1)/2)$. This resolves to $\rho(n) = \Omega(\log n)$.

Finally, observe that the optimum solution is a single M-path π_n of length $1+2\varepsilon$ going from the third to the first quadrant through the origin, see Fig. 2.4. □

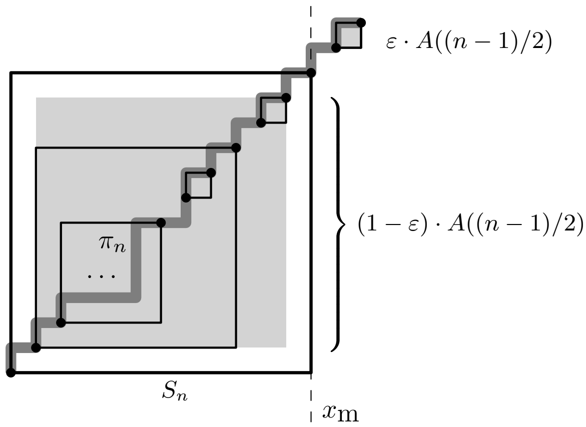


Figure 2.4.: Recursive construction of the arrangement $A(n)$. The gray M-path π_n shows an optimum solution. The dashed vertical line marks where the algorithm separates.

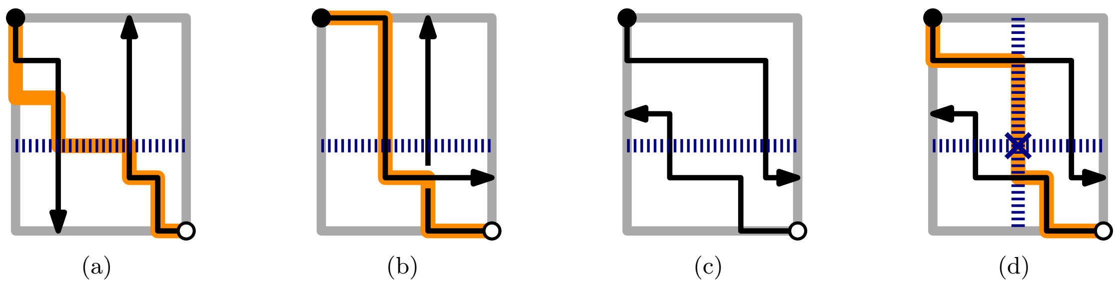


Figure 2.5.: Solid arrows M-connect each terminal to one of its opposite edges. Dashed line segments stab their bounding box. Thick solid paths indicate M-paths between the terminals.

2.3.5. Limits of the Stabbing Technique

The GMMN problem requires to connect each terminal pair with an M-path. In other words, we are asked to M-connect each terminal to the *opposite vertex* of the bounding box it belongs to. In Sections 2.3.1 to 2.3.3 we split the x-separated 2D-GMMN problem into two easier tasks. One of them is to M-connect each terminal to the *opposite horizontal edge*. The problem turned out to be indeed simpler and we solved it with a constant-factor approximation by generating two near-optimal RSAs. Obviously, a solution to this problem alone does not yield a GMN, though. Both M-paths of a rectangle, each connecting one terminal to the opposite horizontal edge, may not cross each other (see Figure 2.5a). However, the key observation is that both M-paths have to cross any horizontal line segment that is stabbing the rectangle. Hence, the existence of any horizontal stabbing line implies an M-path between both terminals; see Figure 2.5a. On this account, the other task regards finding a cost-efficient horizontal stabbing of all rectangles. By exploiting the fact that all rectangles are separated by a vertical line, we achieved a constant approximation, too.

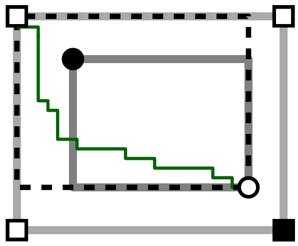
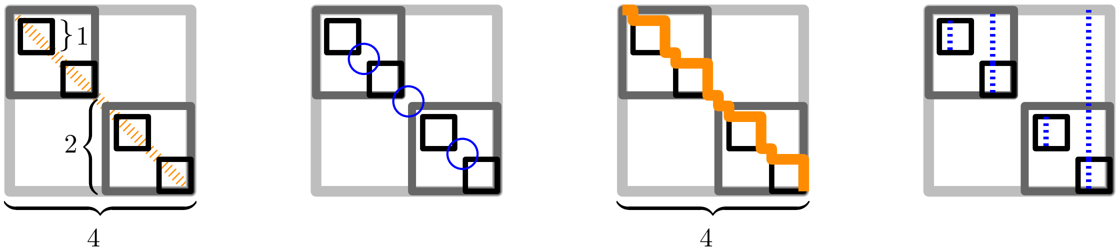


Figure 2.6.: We M-connected the white terminal (white circle) to the upper-left vertex (white square) of the overall bounding box, since the small rectangle lies in the (dashed) bounding box spanned by the terminal and the vertex. Clearly, the black terminal has to be M-connected to the black vertex.

Now, an interesting question is whether this procedure can be adapted to general instances that are not separated by any axis-parallel line. Observe that our solutions to both tasks, stabbing and the RSAs, heavily depend on such a separating line. As for the RSAs, we place the origins on the separating line and guarantee that the generated M-paths cross the respective opposite horizontal edges. In the general case of 2D-GMMN, though, two RSAs would clearly not be sufficient to M-connect each terminal to the opposite edge and we would arrive at considering each x-separated instances separately. However, if we further simplify the task we might obtain a cheaper solution. Instead of M-connecting each terminal to its opposite horizontal edge, let's ask for an M-path to the *horizontal or vertical opposite edge*; see Figures 2.5a, 2.5b and 2.5c. A feasible (but not necessarily cheap) solution can be obtained by solving no more than four RSAP instances. Consider the overall bounding box that contains all rectangles. We M-connect each of its four vertices to every terminal whose rectangle is contained in the bounding box spanned by the vertex and the terminal; see Figure 2.6. Clearly, the M-path between a vertex and a terminal has to leave the terminal's rectangle by crossing the opposite horizontal or vertical edge. Now, if two terminal partners are simultaneously M-connected to their opposite vertical edges and their M-paths do not cross each other (as in Figure 2.5c), there is no M-path between the terminals. Thus, we additionally need a vertical stabbing to *patch* our solution; see Figure 2.5d. Assuming that we are able to compute a cheap horizontal stabbing, by symmetry we also can generate a cheap vertical stabbing.

One way to compute a stabbing is to consider all x-separated instances and to unite their horizontal stabbings. The length of the resulting stabbing is in $O(\mathrm{OPT} \cdot \log n)$, where OPT denotes the length of a minimum GMN. Another way is to take the $\log n$ vertical lines that separate all x-separated instances and to intersect them with the overall bounding box. The resulting line segments constitute a vertical stabbing with the cost factor still being in $O(\log n)$. However, if we want to improve Lemma 2.9 significantly, we need to find a stabbing with total length in $O(\mathrm{OPT})$. Unfortunately, no matter how we solve both tasks, the stabbing, as well as the M-connection of terminals to opposite edges, our obtained GMN will not constitute a constant approximation. For infinite many n there is an GMMN instance with n terminal pairs where an optimum stabbing costs $\Omega(\mathrm{OPT} \cdot \log n / \log \log n)$.



(a) All terminals on a diagonal.

(b) Gaps between the squares.

(c) A minimum GMN.

(d) An optimum stabbing.

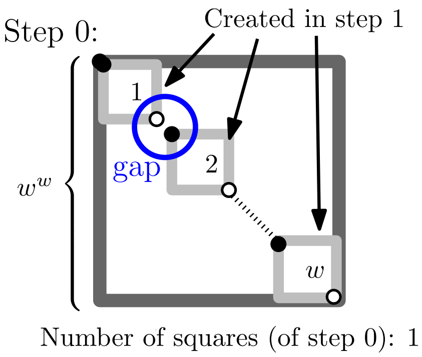
Figure 2.7.: Counterexample for $w = 2$.

Theorem 2.15. *For any integer $w > 0$ there exists a 2D-GMMN instance with $n = \Theta(w^w)$ terminal pairs where the length of a minimum GMN is in $O(w^w)$ and the length of an optimum stabbing is in $\Omega(w \cdot w^w)$. Especially, the length of an optimum stabbing is in $\Omega(\mathrm{OPT} \cdot \log n / \log\log n)$ for each such GMMN instance.*

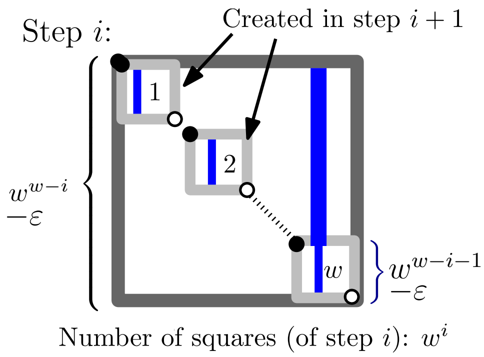
Proof. We begin with an counterexample for $w = 2$. Consider Figure 2.7a. Here we have a recursively defined square structure with initial edge length $2^2 = 4$. Each square represents the bounding box of a terminal pair and all terminals lie on a dotted diagonal. The infinitesimal gaps between the squares (denoted by circles in Figure 2.7b) enforce that no two consecutive squares of the same size share a common stabbing line. A minimum GMN is shown in Figure 2.7c. Its horizontal length is 4 and, hence, the total cost is $2 \cdot 4$. Figure 2.7d depicts an optimum stabbing: Each of the smallest squares has to be stabbed separately. The next bigger squares may share their stabbing lines with at most one embedded square. In total, we pay $8 = 2 \cdot 2^2$ for the stabbing (neglecting the arbitrarily small gaps).

Now let us consider an arbitrary value for w and create a problem instance analogously to the case $w = 2$. We construct the instance *step by step*, beginning with step 0 and finishing with step w. In step 0 we place a square with edge length w^w; see Figure 2.8a. Then, in each step $i > 0$ we place w (smaller) squares into each square created at step $i - 1$. We require that all terminals (of all squares) lie on a diagonal and that there are small gaps between the new created squares such that they don't overlap (as in the example for $w = 2$). Summarized, in each step $i \geq 0$ we place w^i squares with edge length at least $w^{w-i} - \varepsilon$ (where ε accounts for the gap); see Figure 2.8b.

Now, we inductively generate an optimum vertical stabbing and analyze its length. We begin by stabbing each square *of* step w (that has been created in step w) by placing a vertical line segment with length $w^{w-w} - \varepsilon = 1 - \varepsilon$ in each of them. In total, we pay $cost_w = w^w(1-\varepsilon)$. Observe that all squares created at the same step are non-overlapping due to the gaps. Hence, our stabbing is optimal for the squares of step w. Next, we assume that we have an optimum stabbing of all boxes of steps $i+1, i+2, \ldots, w$. We stab each box of step i independently by placing a line segment such that it intersects with any line segment that stabs a box of step $i + 1$ (see Figure 2.8b). As the first line



(a) The big square is created in step 0. The circle indicates a gap between two squares created in step 1. All terminals lie on a diagonal.



(b) A square created in step $i < m$. The vertical line segments depict an optimum stabbing. The thick segment stabbing the big box reuses the stabbing segment of a smaller box.

Figure 2.8.: Counterexample for arbitrary w.

segment has length $w^{w-i} - \varepsilon$ and the second one $w^{w-i-1} - \varepsilon$, we pay only $w^{w-i} - w^{w-i-1}$ for stabbing a box of step i, and $cost_i = w^i \cdot (w^{w-i} - w^{w-i-1}) = w^w - w^{w-1}$ for all the w^i boxes of this step. Thus, for the total cost of an optimum stabbing we obtain that

$$\sum_{i=0}^{w} cost_i = \sum_{i=0}^{w-1} (w^w - w^{w-1}) + cost_w = w \cdot (w^w - w^{w-1}) + w^w(1 - \varepsilon) = \Omega(w \cdot w^w)$$

for constant $\varepsilon > 0$. Given $w = \frac{w \cdot \log w}{\log w} \geq \frac{\log w^w}{\log(w \cdot \log w)} = \frac{\log w^w}{\log \log w^w}$ we even have $\Omega(w^w \frac{\log w^w}{\log \log w^w})$ for the cost. Since the number of terminal pairs n is $\sum_{i=0}^{w} w^i = \Theta(w^w)$ and the length of a minimum GMN is clearly $2 \cdot w^w$ (all terminals lie on a diagonal), an optimum stabbing costs at least $\Omega(\mathrm{OPT} \frac{\log n}{\log \log n})$. □

Note that the counterexample is an instance that is separated by a diagonal. By taking rectangles instead of squares, we can set the angle between the diagonal and any axis arbitrarily (as long as the diagonal remains non-axis-parallel). Hence, the counterexample even shows that GMMN instances separated by any line that is not axis-parallel cannot be stabbed in $O(\mathrm{OPT})$.

2.4. Solving RSAP in Higher Dimension

In this section, we show that we can approximate d-dimensional RSAP with a ratio of $O(\log n)$ even in the all-orthant case where every orthant may contain terminals. In this section, n denotes the number of *terminals*. We generalize the algorithm of Rao et al. [RSHS92] who give an $O(\log n)$-approximation algorithm for the one-quadrant version of 2D-RSAP.

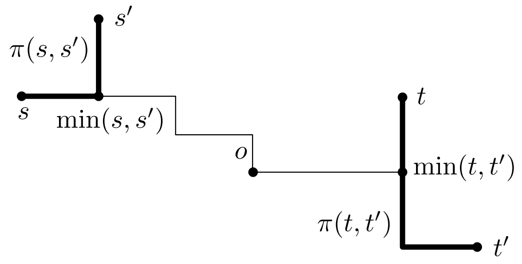


Figure 2.9.: Illustration of Lemma 2.18. For each terminal pair t, t', we compute a suitable point $\min(t, t')$ and an M-path $\pi(t, t')$ containing $\min(t, t')$. Adding an arbitrary M-path from $\min(t, t')$ to o also M-connects t and t' to o.

It is not hard to verify that the $O(\log n)$-approximation algorithm of Rao et al. carries over to higher dimensions in a straightforward manner if all terminals lie in the *same* orthant. We can therefore obtain a feasible solution to the all-orthant version by applying the approximation algorithm to each orthant separately. This worsens the approximation ratio by a factor no larger than 2^d since there are 2^d orthants. Hence, we can quite easily give an $O(\log n)$-approximation algorithm for the all-orthant version since 2^d is a constant for fixed dimension d.

In what follows, we present a tailor-made approximation algorithm for the all-orthant version of d-dimensional RSAP that avoids the additional factor of 2^d. Our algorithm is an adaption of the algorithm of Rao et al., and our presentation closely follows their lines, too.

Consider an instance of RSAP given by a set T of terminals in $\mathbb{R}^d$ (without restriction of the orthant). Let o denote the origin. The algorithm relies on the following lemma, which we prove below.

Lemma 2.16. *Given a rectilinear Steiner tree B for terminal set $T \cup \{o\}$, we can find an RSA A for T of length at most $\lceil \log_2 n \rceil \cdot \|B\|$.*

Every RSA is also a rectilinear Steiner tree. Since the rectilinear Steiner tree problem (RST) admits a PTAS for any fixed dimension d [Aro97], we can generate a $(1 + \varepsilon)$-approximate RST network B that connects T and the origin. By means of Lemma 2.16, we get a $(1 + \varepsilon)\lceil \log_2 n \rceil$-approximation for the RSAP instance T.

Theorem 2.17. *The all-orthant version of d-dimensional RSAP admits a $(1+\varepsilon) \cdot \lceil \log_2 n \rceil$ approximation for any $\varepsilon > 0$.*

Our proof of Lemma 2.16 relies on the following technical lemma, which constitutes the main modification that we make to the algorithm of Rao et al. See Fig. 2.9 for an illustration.

Lemma 2.18. *Let t, t' be two terminals. Then we can compute in constant time a point $\min(t, t')$ and an M-path $\pi(t, t')$ from t to t' containing $\min(t, t')$ with the following property. The union of $\pi(t, t')$ with an M-path from $\min(t, t')$ to o M-connects t and t' to o.*

Proof. We start with the following simple observation. If s and s' are points and p is a point in the bounding box $B(s, s')$ of s and s', then M-connecting s to p and p to s' also M-connects s to s'.

Observe that the three bounding boxes $B(o, t)$, $B(o, t')$, and $B(t, t')$ have pairwise non-empty intersections. By the Helly property of axis-parallel d-dimensional boxes, there exists a point $\min(t, t')$ that simultaneously lies in all three boxes.

M-connecting $\min(t, t')$ with t and t' yields $\pi(t, t')$, and M-connecting $\min(t, t')$ with o yields an M-path between any two of the points t, t', o by a repeated application of the above observation. This completes the proof. □

Proof of Lemma 2.16. We double the edges of B and construct a Eulerian cycle C that traverses the terminals in $T \cup \{o\}$ in some order $t_0, t_1, \ldots, t_n$. The length of C is at most $2\|B\|$ by construction. Now consider the shortcut cycle $\tilde{C}$ in which we connect consecutive terminals t_i, t_{i+1} by the M-path $\pi(t_i, t_{i+1})$ as defined in Lemma 2.18; we set $t_{n+1} := t_0$. Clearly $\|\tilde{C}\| \leq \|C\|$. We partition $\tilde{C}$ it into two halves; $C_0 = \{\pi(t_{2i}, t_{2i+1}) \mid 0 \leq i \leq n/2\}$ and $C_1 = \{\pi(t_{2i+1}, t_{2i+2}) \mid 0 \leq i \leq n/2 - 1\}$. For at least one of the two halves, say C_0, we have $\|C_0\| \leq \|B\|$.

We use C_0 as a partial solution and recursively M-connect the points in the set $T' := \{\min(t_{2i}, t_{2i+1}) \mid 0 \leq i \leq n/2\}$, which lie in C_0 (see Lemma 2.18), to the origin by an arboresence A'. Lemma 2.18 implies that the resulting network $A = C_0 \cup A'$ is in fact a feasible RSAP solution. The length of A is at most $\|C_0\| + \|A'\| \leq \|B\| + \|A'\|$. Note that $|T'| \leq (|T| + 1)/2$.

To summarize, we have described a procedure that, given the rectilinear cycle C traversing terminal set $T \cup \{o\}$, computes a shortcut cycle $\tilde{C}$, its shorter half C_0, and a new point set T' that still has to be M-connected to the origin. We refer to this procedure as *shortcutting*.

To compute the arboresence A', observe that $\tilde{C}$ is a rectilinear cycle that traverses the points in T'. Shortcutting yield a new cycle $\tilde{C}'$ of length at most $\|\tilde{C}\| \leq \|C\|$, a half C_0' no longer than $\|B\|$, which we add to the RSA, and a new point set T'' of cardinality $|T''| \leq |T'|/2 \leq (|T| + 1)/4$, which we recursively M-connect to the origin.

We repeat the shortcutting and recurse. Each iteration halves the number of new points, so the process terminates in $O(\log n)$ iterations with a single point t. Since $\min(o, p) = o$ for any point p (see proof of Lemma 2.18) and our original terminal set $T \cup \{o\}$ contained o, we must have that $t = o$. This shows that the computed solution is feasible. As each iteration adds length at most $\|B\|$, we have $\|A\| \leq \lceil \log_2 n \rceil \cdot \|B\|$. □

2.5. Running Time Analysis

We first analyze the running times of the algorithm for $d > 2$ in Section 2.2. Given an instance R of 0-separated d-dimensional GMMN, the algorithm uses d recursive procedures to subdivide the problem into d-separated instances. For $j \in \{0, \ldots, d-1\}$, let $T_j(n)$ denote the running time of the j-th recursive procedure. The j-th recursive procedure takes a j-separated instance R as input and partitions it into two j-separated instances, each of size at most $|R|/2$, and one $(j + 1)$-separated instance of size at most

$|R|$. The partitioning requires $O(n)$ steps for finding the median of the j-th coordinate value of terminals in R. The two j-separated instances are solved recursively and the $(j+1)$-separated instance is solved with the $(j+1)$-th recursive procedure. Let $T_d(n)$ denote the running time to solve a d-separated instance. In Section 2.4 we approximated such instances by applying a PTAS for the rectilinear Steiner tree problem in any fixed dimension d [Aro97]. Though the probabilistic running time is nearly linear, the deterministic running time is in $O(n^{O(d)})$. We can improve our running time by solving a *rectilinear minimum spanning tree* in time $O(n^2 \log n)$ (see Yao [Yao82]). Observe that such a tree is always a 2-approximation of a minimum rectilinear Steiner tree [GP66]. This worsens our approximation ratio for d-separated instances by only a constant factor and doesn't change it in the order of magnitude. Thus we have

$$\begin{aligned} T_d(n) &= O(n^2 \log n) \\ T_j(n) &= 2T_j(n/2) + T_{j+1}(n) \qquad\qquad \text{for } j \in \{0, \ldots, d-1\} \end{aligned}$$

The running time of our overall algorithm is given by $T_0(n)$. Solving the recurrences above yields $T_0(n) = O(n^2 \log^{d+1} n)$.

Now we analyze the running time of the improved algorithm of Section 2.3. Stabbing x-separated instances can be done with a sweep-line algorithm in $O(n \log n)$ time. The PTAS for RSAP requires time $O(n^{1/\varepsilon} \log n)$ for any ε with $0 < \varepsilon \leq 1$. Hence, we have that $T_1(n) = O(n^{1/\varepsilon} \log n)$, and $T_0(n) = 2T_0(n/2) + T_1(n)$. Solving the recursion yields a running time of $T_0(n) = O(n^{1/\varepsilon} \log^2 n)$ for the improved algorithm.

3. Special Cases of GMMN

Now that we have worked out an approximation ratio for GMMN, the natural question arises whether there are interesting subproblems of GMMN that allow better approximation ratios. We adapt two subproblems known for 3D-MMN and make use of *axis-aligned flats* to either bound the dimensions of bounding boxes (Section 3.1), or to restrict the positions of terminals (Section 3.2). The third problem we look at concerns the aspect ratio of bounding boxes (Section 3.3). In all three cases, we achieve improved approximation ratios.

We define an *axis-aligned flat* by a point c and a non-empty subset C of the coordinates. An *axis-aligned flat* is the set of all points that have the same value as c in each coordinate in C. We say that these coordinates are *fixed* and the remaining coordinates are *free*. The *dimension* of an axis-aligned flat is the number of free coordinates. For short, we refer to an axis-aligned flat as *flat* and call it a (v, d)-*flat* if v is the number of fixed coordinates and d is the number of all coordinates. For instance, all axis-aligned planes and lines, as well as all points, constitute the flats of the three-dimensional Euclidean space.

The algorithms that we present in the following call approximation algorithms for GMMN in a subroutine. In order to keep our results general, we regard these algorithms as black boxes and present our results as a function of their approximation ratios. Thus, applying any concrete approximation algorithm in a subroutine, we obtain a concrete approximation ratio of our overall algorithm. For any dimension d and any $j \leq d$, we assume that we have a $\rho_{d,j}(n)$-approximation algorithm for j-separated GMMN in dimension d. For example, we might use our results of Chapter 2 and yield $\rho_{d,j}(n) = O(\log^{d+1-j} n)$. If $j = 0$, we will use $\rho_d(n)$ to denote the approximation ratio.

3.1. Restricted Dimension of Bounding Boxes

In the introduction we mentioned that MMN is APX-hard for any dimension $d \geq 3$. In fact, Muñoz et al. [MSU09] deduced this lower bound for *3D-MMN-2D*, a special case of three-dimensional MMN. In 3D-MMN-2D two terminals lie in an axis-aligned plane if their bounding box contains no other terminal. Using this property, Muñoz et al. presented a 2ρ-approximation for 3D-MMN-2D where ρ is any approximation ratio achievable for 2D-MMN. Since there exists a 2-approximation for 2D-MMN, this yields a better ratio than $O(\log^4 n)$, the currently best ratio for 3D-MMN (see Corollary 2.6). Straightforwardly, we can extend the notion of 3D-MMN-2D to GMMN and higher dimensions. For every $1 \leq f \leq d$, we consider the special case of d-dimensional GMMN, *dD-GMMN-fD*, where each terminal pair lies in an f-dimensional flat, or, equivalently, all bounding boxes have dimension f. By generalizing the algorithm of Muñoz et al. for 3D-MMN-2D, we obtain a similar result for our special case.

Lemma 3.1. *dD-GMMN-fD admits a $\binom{d-1}{f-1} \cdot \rho_f(n)$-approximation.*

Proof. Run the given $\rho_f(n)$-approximation algorithm on any f-dimensional flat that contains a terminal pair. The union over all these networks yields a feasible GMN for our problem as every bounding box is contained in an f-dimensional flat. Since we consider at most n flats, the running time remains polynomial.

Regarding the cost, observe that the intersection of an optimum solution N and any flat F constitutes a GMN for the terminal pairs in F. Thus, the length of our network for F is bounded by $\rho_f(n) \cdot \|F \cap N\|$. Recall that an f-dimensional flat is defined by a set of $d-f$ fixed coordinates and their values. Thus, any point lies in exactly $\binom{d}{d-f} = \binom{d}{f}$ flats of dimension f. Similarly, any axis-parallel line appears in just $\binom{d-1}{d-f} = \binom{d-1}{f-1}$ f-dimensional flats. The same holds for any line segment in N. Hence, we can bound the length of our overall network by $\sum_F \rho_f(n) \cdot \|F \cap N\| \leq \binom{d-1}{f-1} \cdot \rho_f(n) \|N\|$. □

Now we can apply our approximation algorithms from Section 2.2 and Section 2.3 to solve f-dimensional GMMN. The obtained approximation for dD-GMMN-fD can be viewed as a generalized approximation ratio for d-dimensional GMMN: It does not depend on d (in the order of magnitude) but solely relies on the maximum dimension of all bounding boxes (which in turn is bounded by d, though).

Corollary 3.2. *dD-GMMN-fD admits*

(i) a polynomial-time solution for $f = 1$,

(ii) a $6(d-1) \cdot \log n$-approximation for $f = 2$, and

(iii) an $O(\log^{f+1} n)$-approximation for $f > 2$.

for any fixed dimension d.

3.2. Terminals in Two or More Flats

In lieu of restricting the terminals of a terminal pair to lie in the same flat, in this section we limit the number of flats where all terminals may lie. That is, the union of a bounded number of flats contains all terminals. Such a problem has been already studied in the context of 3D-MMN. Das et al. [DGK+11] presented a $4(k-1)$-approximation algorithm for the case where all terminals are contained in the union of k axis-aligned planes. Unfortunately, their proof does not generalize well to GMMN as it heavily relies on the fact that there is an M-path between every possible pair of terminals.

At first we take a look at a special case where all terminals lie in the union of two (v, d)-flats and each terminal pair has one terminal in each of them. We denote this problem class by *Two-(v,d)-FGMMN.* In the following we make some observations regarding this problem and yield a first approximation ratio. Then, in Sections 3.2.1 to 3.2.3 we improve the ratio for the case of two-dimensional flats (planes) that we denote by *Two-Planes-GMMN*. Finally, in Section 3.2.4 we extend our results to account for instances with arbitrary many flats.

Observations for Two Flats. Now we deduce some basic properties of *Two-(v,d)-FGMMN*. They will help us to understand the problem better and ease the design of our approximation algorithms.

Consider a (v, d)-flat F_1. All points in F_1 have the same values in each of the v fixed coordinates. Consequently, F_1 is v-separated from any set of points, especially from any other (v, d)-flat F_2. Altogether, F_1 is separated from F_2 in any coordinate that is fixed in F_1 or F_2.

Observation 3.3. *Any instance of Two-(v,d)-FGMMN is w-separated where w is the number of coordinates that are fixed in at least one of the two flats.*

As an immediate consequence we obtain that every d-dimensional GMMN instance has the same approximation ratio as v-separated GMMN if two (v, d)-flats separate each terminal pair.

Corollary 3.4. *Two-(v,d)-FGMMN admits a $\rho_{d,v}(n)$-approximation and even a $\rho_{d,w}(n)$-approximation where w is the number of coordinates that are fixed in at least one of the two flats.*

Now, let's project F_1 onto F_2. Then we obtain a set of points that have a same value in each coordinate that is fixed in F_1 or in F_2. Hence, all these points are separating F_1 from F_2 in these coordinates.

Observation 3.5. *The projection of one (v, d)-flat onto another is a set of w-separators where w is the number of coordinates that are fixed in at least one of the two flats.*

Next, let us consider two points, $p_1 \in F_1$ and $p_2 \in F_2$, and let us assume that they are projections of each other. Then clearly, both points are equal in each free coordinate of F_1 and F_2. Thus, their Manhattan distance depends only on coordinates that are fixed in both of the flats. Let's call such a (ordered) pair (p_1, p_2) a *projection pair* of F_1 and F_2.

Observation 3.6. *An M-path between any projection pair of two flats is a shortest rectilinear path between the flats. It uses only line segments that are parallel to axes of coordinates fixed in both flats.*

We finish this paragraph with an observation that will play a central role in our discussion of two-dimensional flats.

Observation 3.7. *Let $t_1 \in F_1$ and $t_2 \in F_2$ and let (p_1, p_2) be a projection pair of F_1 and F_2 inside the bounding box of (t_1, t_2). If there are M-paths between t_1 and p_1, between p_1 and p_2, and between p_2 and t_2, then t_1 and t_2 are M-connected.*

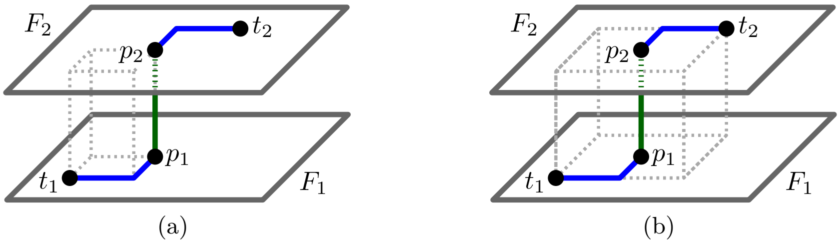
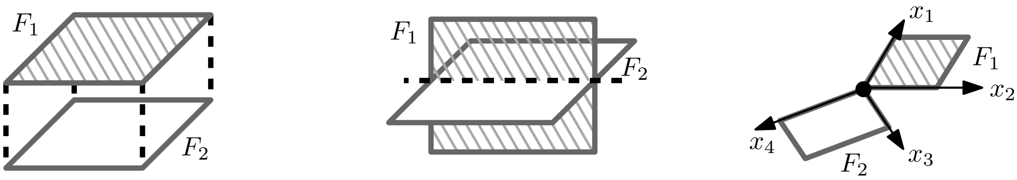


Figure 3.1.: Two parallel planes (2D-flats) in 3D: The terminals t_1 and t_2 are M-connected via an M-path going through p_1 and p_2.



(a) Two parallel planes ($h = 2$).

(b) Two planes that are neither parallel nor orthogonal ($h = 1$).

(c) Two orthogonal planes ($h = 0$). Only possible in dimension $d \geq 4$.

Figure 3.2.: The three cases of h.

Proof. See Figure 3.1a. Clearly, p_1 is inside the bounding box of t_1 and p_2. Given the M-paths between t_1 and p_1, and between p_1 and p_2, we thus conclude that t_1 and p_2 are M-connected. To complete our observation we repeat this argument: Since p_2 is M-connected to t_1 and t_2 and inside their bounding box, there is an M-path between t_1 and t_2 (see Figure 3.1b). □

Note that a projection pair is inside a bounding box of a terminal pair if and only if it is d-separating the terminal pair.

Problems with Two Planes. Now, having established some basic observations we are ready to improve Corollary 3.4 for Two-Planes-GMMN, the case when the flats are two-dimensional planes. Let R denote an instance of this problem class and F_1 and F_2 the respective axis-aligned planes. Recall that each plane contains one terminal from every terminal pair. Then, let N be an optimum solution to R and let N_{x_i} denote the subset of N that contains only line segments parallel to the x_i-axis. We define N_X in the same manner for any set X of coordinates. Further, let X_1 and X_2 denote the sets of free coordinates of F_1 and F_2, respectively. Their complements, $\overline{X_1}$ and $\overline{X_2}$, are the sets of fixed coordinates and $\overline{X_1 \cup X_2}$ is consequently the set of all coordinates that are fixed in both planes. Observe that there are three possible cases for $h = |X_1 \cap X_2|$. Each of the cases determines how the planes are aligned to each other; see Figures 3.2a, 3.2b and 3.2c. We will obtain the following result: Two-Planes-GMMN admits, for any $\varepsilon > 0$,

(i) a $((12 + \varepsilon) \cdot \log n)$-approximation if both planes are parallel,

(ii) a $(8 + \varepsilon)$-approximation if both planes are neither parallel nor orthogonal, and

(iii) a $(1 + \varepsilon)$-approximation if both planes are orthogonal.

In all three cases we will proceed similarly: Our key idea is to identify a small set of projection pairs of F_1 and F_2 such that each bounding box in R contains at least one of them. Then we will interconnect each projection pair and let them act as *bridges* between both planes. By solving RSAP instances and, to some extent, applying the stabbing technique of Section 2.3, we M-connect each terminal pair to at least one projection pair. By Observation 3.7 we will infer that the resulting network contains an M-path for each terminal pair. Note that for applying Observations 3.3 and 3.5 we will use that $d - h$ is the number of coordinates that are fixed in at least one of the two planes.

Regarding the running time, for each case we will generate at most $O(\log n)$ stabbings and 2D-RSAs, each in time polynomial to $O(n)$ terminals. The projection pairs will be either already defined by the problem instance (Section 3.2.3), or automatically obtained as a byproduct when solving the stabbing problems (Sections 3.2.1 and 3.2.2). Hence, the overall running time remains polynomial.

3.2.1. Two Parallel Planes

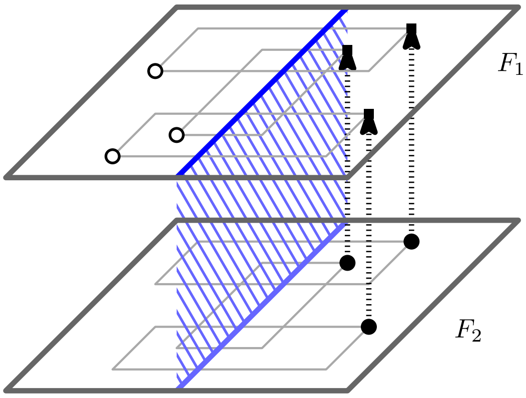
In this case we have $X_1 = X_2$, that is, both planes, F_1 and F_2, are parallel. Though this seems to constitute the case that is least complex to describe, the approximation ratio we obtain is the worst of all the three cases. This is due to the fact that parallel planes are only $(d - 2)$-separated.

Lemma 3.8. *Two-Planes-GMMN admits, for any $\varepsilon > 0$, a $((12 + \varepsilon) \cdot \log n)$-approximation if both planes that define the instance are parallel.*

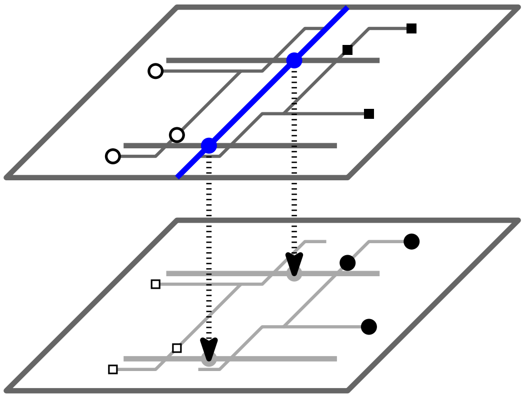
Proof. Recall that we are interested in a set of projection pairs that d-separate all terminal pairs in R. By Observation 3.3 R is already $(d - 2)$-separated with respect to all fixed coordinates. Choosing any free coordinate, we partition R into $\log n$ subsets as described in Section 2.2.1. Each of the subsets is separated in this free coordinate, and hence is $(d - 1)$-separated.

Let's consider such a subset R_x and its projection R_x^1 onto F_1 (see Figure 3.3a). Note that the projection of N onto F_1 is a feasible GMN N^1 for R_x^1. Thus we can use N^1 to bound he length of an optimum solution to R_x^1. Since R_x^1 can be viewed as a 1-separated 2D-GMMN instance, we solve it according to Section 2.3, and, without loss of generality, use the same set of piercing points P_0 for stabbing the right and left part of R_x^1. Next we project our solution onto F_2 and call our solution A^1 and its projection A^2 (see Figure 3.3b). With Theorem 2.9 we bound the the length of A^1, and hence A^2, by $\|A^1\| = \|A^2\| \leq (6 + \varepsilon) \cdot \|N^1\|$.

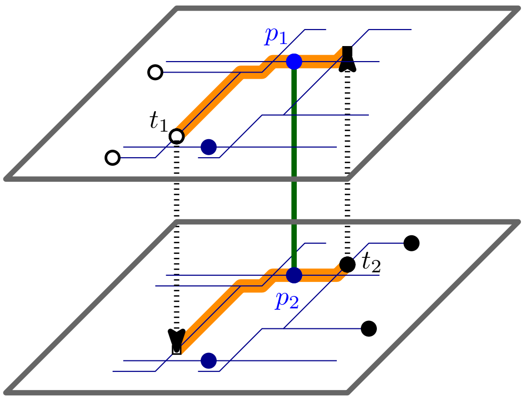
Now we will see that each bounding box contains a projection pair. Consider any terminal pair $(t_1, t_2) \in R_x$ with $t_1 \in F_1$. According to our construction there is an



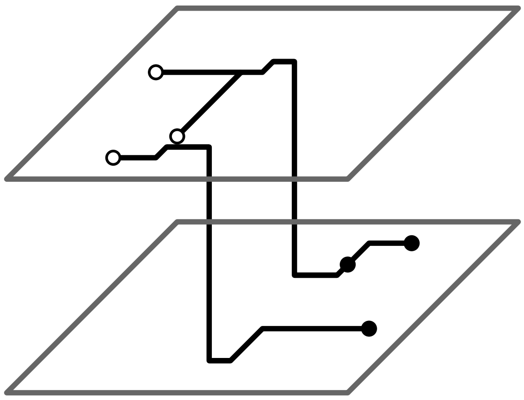
(a) A plane separates R_x. We project terminals of F_2 onto F_1 and obtain the 2D-instance R_x^1.



(b) We generate a 2D-GMN for R_x^1 and project the network onto F_2.



(c) There is a thick M-path between t_1 and the projection of t_2 (in the upper plane), and between t_2 and the projection of t_1 (in the lower plane). When we M-connect p_1 and p_2, we obtain an M-path between t_1 and t_2.



(d) Interconnecting all piercing points yields a 3D-GMN (black lines) for R_x.

Figure 3.3.: Solving R_x in 3D ($h = 2$).

M-path in A^1 connecting t_1 to the projection of t_2 (in F_1) by crossing some piercing point p_1 (see Figure 3.3c). Similarly, the projection of this M-path (onto F_2) connects t_2 to the projection of t_1 by crossing p_2, the projection of p_1. Hence, t_1 is M-connected to p_1 and t_2 is M-connected to p_2. Note that (p_1, p_2) constitutes a projection pair of F_1 and F_2 and lies in the bounding box of (t_1, t_2). Now, if we M-connect p_1 and p_2 then, by Observation 3.7, we obtain an M-path between t_1 and t_2. Therefore, let's M-connect each piercing point in A^1 to its projection in A^2 and denote the union of these M-paths by A_{Con}. We conclude that $A^1 \cup A^2 \cup A_{\text{Con}}$ is a GMN for R_x (see Figure 3.3d).

Let us estimate the cost of A_{Con}. Since the length of an M-path in A_{Con} equals the distance l of both planes (see Observation 3.6), we pay $\|A_{\text{Con}}\| = |P_0| \cdot l$. On the other side, we have $|P_0| \cdot l \leq 2 \cdot \left\|N_{\overline{X_1}}\right\|$. To see this, consider the set P_0' that we obtain by taking every other piercing point from P_0 (with increasing coordinate values on the separating line). Now any two witness intervals of distinct piercing points $p, p' \in P_0'$ are disjoint. Otherwise any point in P_0 between p and p' would pierce one of the witness intervals of p or p'. However, this would contradict the definition of witness intervals. Thus, for each piercing point in P_0' there is a *witness bounding box* that does not overlap with witness bounding boxes of other points in P_0'. Consequently N is forced to connect each terminal pair of such a witness bounding box by a separate M-path. Hence, N connects both planes by at least $|P_0'|$ independent M-paths. Clearly, each such M-path has the same length l and uses only line segments parallel to axes of fixed coordinates. On this account we obtain that $|P_0'| \cdot l \leq \left\|N_{\overline{X_1}}\right\|$. Now, without loss of generality, we have $|P_0'| = \lceil |P_0| / 2 \rceil$ and we conclude that $\|A_{\text{Con}}\| = |P_0| \cdot l \leq 2 \cdot \lceil |P_0| / 2 \rceil \cdot l \leq 2 \cdot \left\|N_{\overline{X_1}}\right\|$.

With $\|N^1\| \leq \|N_{X_1}\|$ we bound the cost of our solution to R_x by

$$\begin{aligned} \left\|A^1 \cup A^2 \cup A_{\text{Con}}\right\| &\leq 2 \cdot ((6+\varepsilon) \cdot \|N_{X_1}\|) + 2 \cdot \left\|N_{\overline{X_1}}\right\| \\ &\leq (12 + \varepsilon') \cdot \|N\| \end{aligned}$$

for $\varepsilon' = \varepsilon/2$. Summarized, R is approximable with the approximation ratio

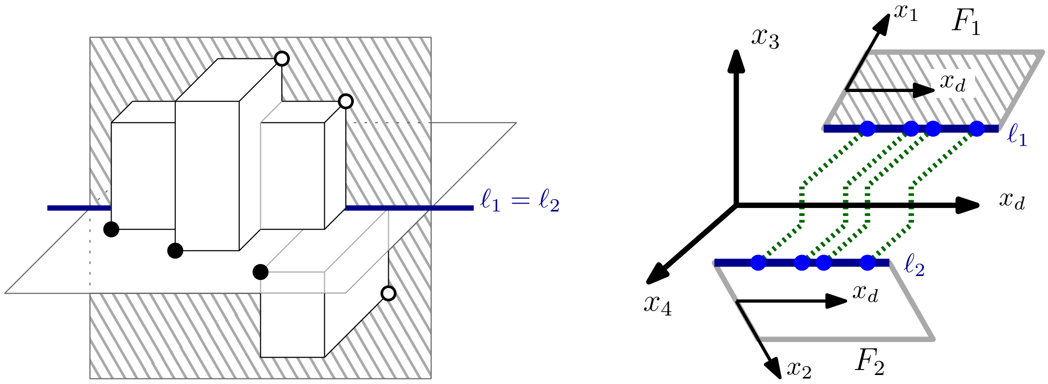
$$(12 + \varepsilon') \cdot \log n \cdot \|N\|.$$

□

3.2.2. Two Planes That Are Neither Parallel Nor Orthogonal

In this section both planes share exactly one free coordinate. Thus, they are neither parallel nor orthogonal. However, the following proof is very similar to the case of parallel planes in Section 3.2.1. We compute a stabbing and some RSAs to connect all terminals to a set of projection pairs that are composed of piercing points. The key difference regards the *mirror step* of stabbing (see Section 2.12). Here we have to *mirror* the stabbing line segments across the planes.

Lemma 3.9. *Two-Planes-GMMN admits, for any $\varepsilon > 0$, a $(8 + \varepsilon)$-approximation if both planes that define the instance are neither parallel nor orthogonal.*



(a) In 3D, both planes interesect in the line $\ell_1 = \ell_2$. Circles and disks represent terminals on the vertical and horizontal plane, respectively. Each terminal pair (represented by a bounding box) has one terminal in each of the planes.

(b) In 5D, consider the subspace spanned by axes x_3, x_4 and $x_d = x_5$. Then F_1 and F_2 are thick lines (ℓ_1 and ℓ_2) parallel to x_d. The dashed interconnections between the piercing points are all parallel to $\overline{X_1 \cup X_2} = \{x_3, x_4\}$.

Figure 3.4.: Case $h = 1$ in 3D and 5D.

Proof. There is exactly one coordinate that is free in both planes, say it's x_d. Thus, the projection of one plane onto the other is a x_d-axis parallel line. Let ℓ_1 denote the projection line in F_1 and ℓ_2 the projection line in the other plane (see Figures 3.4a and 3.4b). By Observation 3.5 both lines are $(d-1)$-separators for R. For $i \in \{1, 2\}$, we project R and N onto F_i and call their projections R^i and N^i, respectively. Note that N^i is a feasible GMN for R^i and we can use its length as an upper bound of an optimum solution to R^i.

At first, let's establish a set of projection pairs that d-separate R. Since ℓ_1 and ℓ_2 are parallel, their intersection with R (or with R^1 and R^2, respectively) yields the same set of intervals. Let's compute a minimal piercing of these intervals in ℓ_1. Observe that projecting the piercing onto F_2, we obtain the same minimal piercing of the intervals in ℓ_2. Clearly, each piercing point in ℓ_1 and its projection in ℓ_2 form a projection pair of F_1 and F_2. Further, they both are contained in the same bounding boxes of R. Let P_{pair} denote the set of all these projection pairs. We M-connect each of them and call A_{Con} the union of these M-paths; see Figure 3.4b. By Observation 3.6 and a similar discussion as in Section 3.2.1 (parallel planes) we infer that

$$\|A_{\text{Con}}\| \leq 2 \cdot \left\| N_{\overline{X_1 \cup X_2}} \right\| .$$

Now all that remains is to M-connect every terminal to at least one projection pair in P_{pair}.

We will use our minimal piercing to generate a stabbing of R^1 in F_1 and of R^2 in F_2. As in Section 3.2.1 we regard R^1 as a 1-separated 2D-GMMN instance. Following Section 2.3.1 we first generate a *horizontal* stabbing of the right part of R^1 and then for its left part, each time beginning with our minimal piercing. In our case *horizontal*

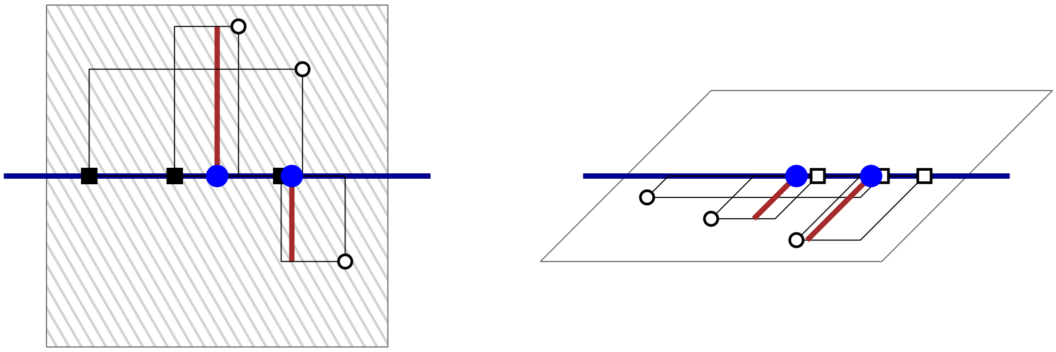


Figure 3.5.: Though we generate the stabbing on both planes independently, we begin with the same minimal piercing (large disks on the thick line).

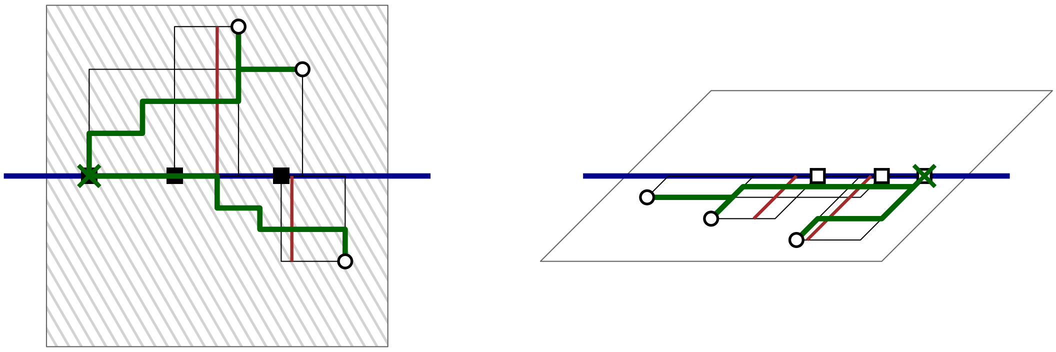


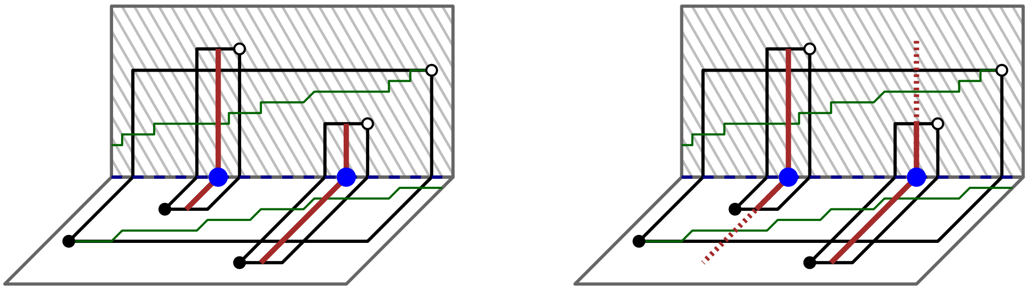
Figure 3.6.: RSAs connecting terminals to stabbing line segments.

means parallel to the axis of the unique coordinate in $X_1 \setminus \{x_d\}$, say x_1. Since every terminal pair has one terminal on ℓ_1, the resulting stabbing is already a stabbing of R^1. Thus we can omit the mirror step of Section 2.3.2 (for now) and bound the length of the stabbing by $2 \cdot \|N^1_{x_1}\| \leq 2 \cdot \|N_{x_1}\|$. Repeating this procedure on F_2, we obtain a stabbing of R^2 with line segments parallel to the axis of the unique coordinate in $X_2 \setminus \{x_d\}$, say x_2. Analogously, the length is bounded by $2 \cdot \|N_{x_2}\|$. Let A^*_{Stab} denote the union of both stabbings. Then it holds that $\|A^*_{\text{Stab}}\| \leq 2 \cdot (\|N_{x_1}\| + \|N_{x_2}\|)$. Figure 3.5 shows a stabbing of the instance of Figure 3.4a.

Next, we M-connect each terminal in F_1 to each line segment that is stabbing the bounding box to which the terminal belongs. We do this analogously to Section 2.3.3 by solving two 2D-RSAP problems, then we rerun the algorithm with the terminals in F_2. Figure 3.6 depicts two possible RSAs for the instance of Figure 3.4a. Let A_{RSA} denote the union of these RSAs. For any $\varepsilon > 0$, we obtain that

$$\begin{aligned}\|A_{\text{RSA}}\| &\leq (2+2\varepsilon) \cdot \left(\left\|N^1\right\| + \left\|N^1_{x_d}\right\|\right) + (2+2\varepsilon) \cdot \left(\left\|N^2\right\| + \left\|N^2_{x_d}\right\|\right) \\ &\leq (2+2\varepsilon) \cdot (\|N_{x_1}\| + \|N_{x_2}\| + 4 \cdot \|N_{x_d}\|)\end{aligned}$$

by applying $\|N^i\| = \left\|N^i_{x_i} \cup N^i_{x_d}\right\| \leq \|N_{x_i}\| + \|N_{x_d}\|$.



(a) The big box is stabbed in each plane at a different piercing point. There is no M-path connecting the terminals of the big box.

(b) After elongating the segments, the big box is now stabbed in both planes at a same piercing point. Hence, now its terminals are M-connected.

Figure 3.7.: A big bounding box and two smaller bounding boxes are projected onto both planes. They are stabbed by line segments that are incident on one of the two piercing points each representing a projection pair. (Note that in 3D a projection pair consists of one point.) An RSA in each plane connects the terminals to some origin. (Here depicted only for the terminals of the big box.)

Our objective is to M-connect each terminal pair to a projection pair in P_{pair} that is inside the bounding box of the terminal pair. However, our current network, $A^*_{\text{Stab}} \cup A_{\text{RSA}}$, does not guarantee such connections yet. In Figure 3.7a we depict a counterexample. Clearly, we need to modify the stabbing such that both projections of each bounding box (one in R^1 and the other in R^2) are stabbed by line segments that are incident on a same projection pair. We will do this by adapting the mirror step from Section 2.3.2. For each projection pair $(p_1, p_2) \in P_{\text{pair}}$ we identify all line segments that are incident on p_1 in F_1 and on p_2 in F_2 and denote their sets as S^p_1 and S^p_2, respectively. In Figure 3.5 each piercing point has a solid horizontal segment in one of the sets, and a solid vertical segment in the other one (In 3D p_1 and p_2 fall together and each projection pair is thus a piercing point). One way to achieve our objective is to elongate all line segments in $S^p_1 \cup S^p_2$ such that they reach the length of a longest segment in $S^p_1 \cup S^p_2$. However, in worst case this would quadruple our costs. Therefore we conduct a cheaper alternative. We take a longest line segment $s_1 \in S^p_1$ and a longest line segment $s_2 \in S^p_2$ and elongate all the remaining line segments such that the elongated line segments in S^p_1 are not shorter than s_2 and the elongated line segments in S^p_2 are not shorter than s_1. In fact we do nothing else than *mirroring* the line segments of F_1 onto F_2 and vice versa (see Figure 3.8). One such operation increases the cost by at most $2 \cdot l(s_1) + 2 \cdot l(s_2)$, and all operations together by no more than $2 \cdot (2 \|N_{x_1}\|) + 2 \cdot (2 \|N_{x_2}\|)$. Denoting A_{Stab} the new stabbing, in total we have

$$\|A_{\text{Stab}}\| \leq 6 \cdot (\|N_{x_1}\| + \|N_{x_2}\|)\,.$$

To see that our new stabbing meets our requirements, we perform a similar discussion as in Section 2.3.2. Consider any bounding box in R and its two projections, B_1 in R^1 and B_2 in R^2. Without loss of generality, B_1 is at least as long in x_1-direction as B_2 in

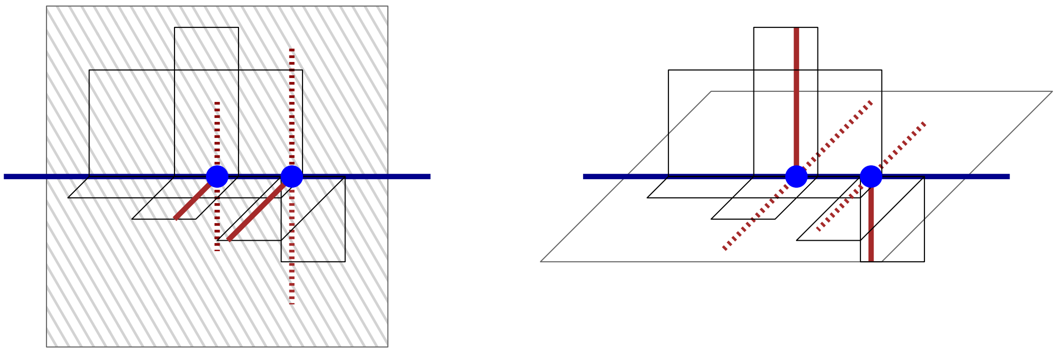
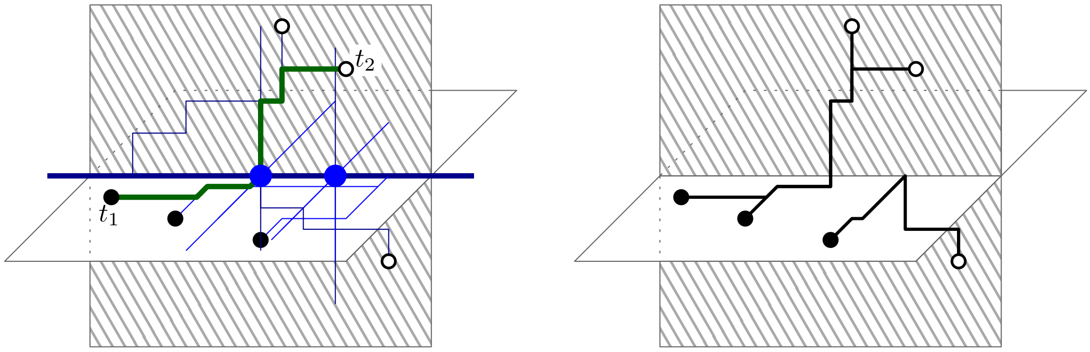


Figure 3.8.: We *mirror* the stabbing line segments of one plane onto the other.



(a) The RSAs and the stabbing line segments (thin) contain an M-path (thick) for t_1 and t_2 over a piercing point.

(b) The obtained GMN for R.

Figure 3.9.: The stabbing and the RSAs together.

x_2-direction. By construction, B_1 is stabbed by a segment $s_1 \in A^*_{\text{Stab}}$ that is incident on some piercing point $p_1 \in \ell_1$. Let $p_2 \in \ell_2$ denote the projection of p_1 onto ℓ_2. If B_2 is stabbed by the segment $s_2 \in A^*_{\text{Stab}}$ that is incident on p_2, we are done. Otherwise, s_1 is longer than s_2. But then we have elongated s_2 at least to the length of s_1 and hence B_2 is stabbed in A_{Stab} with a line segment that is incident on p_2 (see Figure 3.7b).

It remains to show that $A_{\text{Stab}} \cup A_{\text{RSA}} \cup A_{\text{Con}}$ is a GMN for R. Recall, that our stabbing A_{Stab} guarantees two line segments for each bounding box in R, one of them stabs the projection of the bounding box in R^1, the other one stabs the projection in R^2, and both line segments are incident on a same projection pair on the respective lines (ℓ_1 or ℓ_2). Consider any terminal pair (t_1, t_2) in R (with $t_1 \in F_1$ and $t_2 \in F_2$) and such a projection pair $(p_1, p_2) \in P_{\text{pair}}$ (that is necessarily in the bounding box of (t_1, t_2)); see Figure 3.9a. According to the discussion above, there exists an M-path in A_{RSA} connecting t_1 to the line segment incident on p_1, hence t_1 is M-connected to p_1 in $A_{\text{Stab}} \cup A_{\text{RSA}}$. The same holds for t_2 and p_2, and given the M-path between p_1 and p_2 in A_{Con}, we conclude with Observation 3.7 that t_1 and t_2 are M-connected in $A_{\text{Stab}} \cup A_{\text{RSA}} \cup A_{\text{Con}}$. Hence,

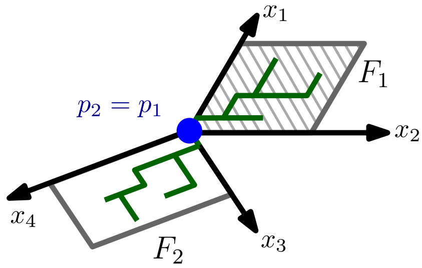


Figure 3.10.: In 4D p_1 and p_2 fall together (large disk) but may be distinct in higher dimensions. The RSAs in each plane connect all terminals to the large disk.

this network constitutes a GMN for R (see Figure 3.9b). We complete our proof by estimating the overall length of our network.

$$\begin{aligned}
&\|A_{\text{Stab}} \cup A_{\text{RSA}} \cup A_{\text{Con}}\| \\
&\quad\leq \|A_{\text{Stab}}\| + \|A_{\text{RSA}}\| + \|A_{\text{Con}}\| \\
&\quad\leq 6 \cdot (\|N_{x_1}\| + \|N_{x_2}\|) + (2 + 2\varepsilon) \cdot (\|N_{x_1}\| + \|N_{x_2}\| + 4 \cdot \|N_{x_d}\|) + \left\|N_{\overline{X_1 \cup X_2}}\right\| \\
&\quad\leq (8 + 2\varepsilon) \cdot (\|N_{x_1}\| + \|N_{x_2}\| + \|N_{x_d}\|) + \left\|N_{\overline{X_1 \cup X_2}}\right\| \\
&\quad= (8 + 2\varepsilon) \cdot \|N_{X_1 \cup X_2}\| + \left\|N_{\overline{X_1 \cup X_2}}\right\| \\
&\quad\leq (8 + \varepsilon') \cdot \|N\|
\end{aligned}$$

for $\varepsilon < 1$ and $\varepsilon' = \varepsilon/2$. □

3.2.3. Two Orthogonal Planes

In this case, any coordinate is fixed in at least one of the two planes. Thus both planes are d-separated and admit an $O(\log n)$-approximation (see Lemma 2.8). However, taking into consideration the *planar nature* of the problem, we achieve a significantly better approximation ratio.

Lemma 3.10. *Two-Planes-GMMN admits a polynomial-time approximation scheme if both planes that define the instance are orthogonal.*

Proof. Both planes, F_1 and F_2, share no free coordinates. Therefore each coordinate is fixed in F_1 or F_2 and the projection of one plane onto the other yields a singleton. We call its unique point p_1 for the projection of F_2 onto F_1, and p_2 for the other direction (see Figure 3.10). By Observation 3.5 both points are d-separators for R and by this contained in all bounding boxes of R.

We construct a rectilinear network A by solving two 2D-RSAP instances by applying a PTAS [LR00, Zac00], in one we M-connect the terminals of F_1 to p_1 and in the other the terminals of F_2 to p_2 (compare the RSAs in Figure 3.10). Further, we connect p_1 and p_2 with an M-path π. Now with Observation 3.7 each terminal pair is M-connected. Hence, A is a GMN for R.

Let's consider the length of A. Since π is a shortest M-path between F_1 and F_2 (see Observation 3.6) its length is covered by $\left\|N_{\overline{X_1 \cup X_2}}\right\|$. To bound the cost of the near-optimal RSAs, let's project N onto F_1 and observe that the projection of any M-path between a terminal $t_1 \in F_1$ and a terminal $t_2 \in F_2$ constitutes an M-path between t_1 and p_1. Hence, the projection of N is an RSA for the terminals in F_1 with origin p_1. The same holds for p_2 and the projection of N onto F_2. As F_1 and F_2 share no line segments, both projections of N cost no more than $\|N_{X_1}\| + \|N_{X_2}\| = \|N_{X_1 \cup X_2}\|$ together. Consequently we can bound the length of the two RSAs by $(1+\varepsilon) \cdot \|N_{X_1 \cup X_2}\|$ for any $\varepsilon > 0$. Hence, altogether we obtain that $\|A\| \leq (1+\varepsilon) \cdot \|N\|$. □

3.2.4. Problems With More Than Two Flats

In this section we provide some straightforward observations for the case of more than two flats. Let's consider a GMMN instance R where all terminals are contained in the union of k flats each of dimension f.

First let us assume that each terminal pair is entirely contained in one flat. Then, obviously, we can generate a solution of cost $k \cdot \rho_f(n) \cdot \mathrm{OPT}$ where OPT is the length of an optimum solution to R. By taking Lemma 3.1 into account, we can even optimize it to $\min\left(k, \binom{d-1}{f-1}\right) \cdot \rho_f(n) \cdot \mathrm{OPT}$.

Next, suppose that both terminals of each terminal pair are in two distinct flats. By considering each pair of flats separately we can reuse our result from Corollary 3.4. Thus, Summing up the cost we obtain $k \cdot (k-1) \cdot \rho_{d,d-f}(n) \cdot \mathrm{OPT}$. Recall that an f-dimensional flat has $d - b$ fixed coordinates.

Finally, if the terminals lie anywhere in the union of the k flats we just solve the two cases independently and add up the costs. The running time is polynomial since we can assume that $k \leq 2n$.

Lemma 3.11. *d-dimensional GMMN admits an approximation ratio of*

$$\min\left(k, \binom{d-1}{f-1}\right) \cdot \rho_f(n) + k(k-1) \cdot \rho_{d,d-f}(n)$$
$$\leq k \cdot \rho_f(n) + k^2 \cdot \rho_{d,d-f}(n)$$

if all terminals are contained in the union of k flats each of dimension f with $1 \leq f \leq d$.

Let's apply Theorem 2.5 for d-dimensional GMMN, and Theorem 2.9 and Lemma 3.8 for the two-dimensional case.

Corollary 3.12. *GMMN with fixed dimension d admits an approximation ratio of*

- $O(k^2 \cdot \log n)$ *for* $f = 2$, *and*
- $O(k^2 \cdot \log^{f+1} n)$ *for any* $f > 2$,

if all terminals are contained in the union of k flats each of dimension f.

As long as k is in $o(\log^{d/2} n)$ for $f = 2$, and in $o(\log^{(d-f)/2} n)$ for $f > 2$, we achieve a better approximation ratio than in Theorem 2.5.

3.3. Restricted Aspect Ratio of Bounding Boxes

In this section we examine the case where all edges have similar lengths in each bounding box. As we will see, this restriction allows us to obtain a much better approximation ratio than in the general case where the aspect ratios are unbounded. Our study of this problem is motivated by an idea of Arkin and Mitchell [AM12] to *somehow* transform a given GMMN instance such that it meets the above requirements, then to solve it with a good approximation ratio, and finally to conduct a cost-retaining inverse-transformation in order to obtain a feasible solution to the original problem. Such a proceeding would greatly improve our results of Theorem 2.5. However, it is an open problem whether such a transformation is possible. Now we present an algorithm for the *transformed* GMMN instances that has its origin in an idea of Arkin and Mitchell.

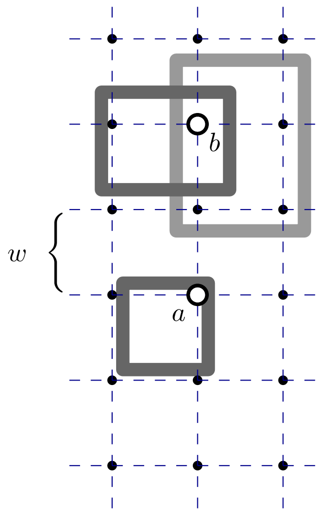
Lemma 3.13. *In any fixed dimension d, GMMN admits an $O\left(c^d \cdot \log n \cdot \rho_{d,d}(n)\right)$ approximation if the ratio between the longest and shortest edge of each bounding box is smaller than c.*

Proof. Let R be a GMMN instance in fixed dimension d where the aspect ratio of each bounding box is bounded by c. To simplify matters, we scale our instance such that the longest edge of all bounding boxes has length n.

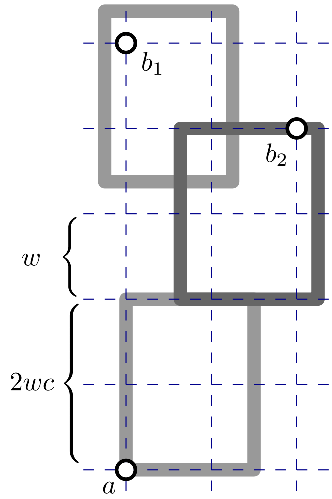
We begin by M-connecting every terminal pair whose bounding box has an edge with length smaller than 1. Since all other edges of these boxes are shorter than c, we pay less than $d \cdot c$ for each such box, and not more than $d \cdot c \cdot n$ for all of them. This is negligible for our overall approximation ratio as every solution costs n at least.

Now all remaining bounding boxes have edge lengths between 1 and n. We partition all bounding boxes into $\log_2 n$ *size classes*, where each *size class* contains only boxes whose shortest edge length is between 2^p and 2^{p+1} for some $1 \leq p \leq \log_2 n - 1$.

Consider a particular size class, thus all edges have lengths between w and $2wc$ for some w. Lay down a d-dimensional regular grid with spacing w between the grid lines. Clearly, each bounding box contains one grid point at least. Further, each grid point is a d-separator of the set of bounding boxes containing it; see Figure 3.11a. Hence, for each grid point we solve a d-separated GMMN instance by applying the $\rho_{d,d}(n)$-approximation algorithm. Taking the union over these networks yields a GMN for our size class. Allowing each terminal pair to appear in at most one d-separated instance, we immediately conclude that our algorithm runs in polynomial time. Regarding the length of our solution, consider any two grid points, a and b; see Fig 3.11b. If for any coordinate x_i it holds that $|x_i(a) - x_i(b)| > 2wc$, then the bounding boxes containing a cannot overlap with bounding boxes containing b. That is, d-separated GMMN instances of a and b are overlapping only if $|x_i(a) - x_i(b)| \leq 2wc$ for any $i \leq d$. Given grid spacing w, we infer that each such instance overlaps with at most $(2 \cdot 2c + 1)^d - 1 = O(c^d)$ other instances. Thus we can partition the set of the d-separated instances into $O(c^d)$ subsets in which all instances are independent. (In Figure 3.11a all boxes would be in one subset, whereas in Figure 3.11b there would be two subsets, as the box in the middle overlaps with the other two.) Clearly, the union of all instances in a subset is $\rho_{d,d}(n)$-approximated by the union of our networks for these instances. Hence, all our networks



(a) The box at the bottom is a square with minimum edge length w, however, it contains the grid point a. The two boxes at the top compose the 2-separated instance of grid point b.



(b) The boxes containing grid points a and b_1 have maximum height $2wc$. However, they do not overlap as the vertical distance between a and b_1 is greater than $2 \cdot 2wc$. On the other side, the box containing b_2 overlaps with the other two boxes; not surprisingly since the respective distances are smaller or equal to $2wc$ in each coordinate.

Figure 3.11.: Two size-classes ($c = 1$) in 2D.

together form an $O(c^d \cdot \rho_{d,d}(n))$-approximation for the size class. Since we have $\log_2 n$ size classes, our claim follows. $\square$

Now, let's apply Lemma 2.8 to approximate d-separated GMMN. Then for all $c = o(\log n)$, especially for $c = O(1)$, Lemma 3.13 constitutes an enhancement of Theorem 2.5.

Theorem 3.14. *In any fixed dimension, GMMN admits an $O(\log^2 n)$-approximation if the ratio between the longest and shortest edge of each bounding box is smaller than some constant.*

4. Conclusion and Open Problems

We have presented an $O(\log^{d+1} n)$-approximation algorithm for d-dimensional GMMN, which implies the same ratio for MMN. Prior to our work, no approximation algorithm for GMMN was known. For $d \geq 3$, our result is a significant improvement over the ratio of $O(n^\varepsilon)$ which was the only approximation algorithm for d-dimensional MMN known so far. If we assume that the aspect ratio of all bounding boxes is bounded by some constant, we even obtain an $O(\log^2 n)$-approximation.

In 2D, there is still 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 2D-GMMN, 2D-MMN admits 2-approximations [CNV08, GSZ11, Nou05]—but is 2D-GMMN really harder to approximate than 2D-MMN? Indeed, given that GMMN is more general than MMN, it may be possible to derive stronger non-approximability results for GMMN. So far, the only such results are that 2D-MMN admits no FPTAS [CGS11] and that 3D-MMN cannot be approximated beyond a factor of 1.00002 [MSU09]. We summarize the currently known best approximation ratios and hardness bounds in Table 4.1.

	2D		Dimension	$d \geq 2$
RSAP	PTAS		$O(\log n)$	
MMN	2,	no FPTAS	$O(\log^{d+1} n)$,	no PTAS
GMMN	$O(\log n)$,	no FPTAS	$O(\log^{d+1} n)$,	no PTAS

Table 4.1.: Approximability and Hardness of RSAP, MMN and GMMN.

Concerning the positive side, for $d \geq 3$, a constant-factor approximation for d-dimensional RSAP would shave off a factor of $O(\log n)$ from the current ratio for d-dimensional GMMN. This may be in reach given that 2D-RSAP admits even a PTAS [LR00, Zac00]. Alternatively, a constant-factor approximation 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. However, note that it would be easier to find a constant-factor approximation for k-dimensional GMMN. This holds by the following observation: Suppose that we have a constant-factor approximation algorithm for $(d-k)$-separated GMMN in any dimension $d \geq 2$ and some $k < d$. Then, given any k-dimensional GMMN instance R, we introduce k additional coordinates and, for each terminal in R, we set their values to zero. Hence, R is now a $2k$-dimensional and k-separated instance. Setting $d = 2k$, we can approximate it within a constant-factor. Regarding the class of GMMN instances where the aspect ratio of bounding boxes is bounded by some constant, it is an interesting question whether general GMMN can be reduced to this special case.

Bibliography

[AM12] Esther Arkin and Joseph Mitchell. personal email communication, 2012.

[Aro97] Sanjeev Arora: Nearly linear time approximation schemes for Euclidean TSP and other geometric problems. In *Proc. 38th Annu. IEEE Symp. Foundat. Comput. Sci. (FOCS'97)*, pages 554–563, 1997. http://dx.doi.org/10.1109/SFCS.1997.646145.

[Aro03] Sanjeev Arora: Approximation schemes for NP-hard geometric optimization problems: A survey. *Math. Program.*, 97(1–2):43–69, 2003. http://www.springerlink.com/openurl.asp?genre=article&issn=0025-5610&volume=97&issue=1&spage=43.

[BWWS06] Marc Benkert, Alexander Wolff, Florian Widmann, and Takeshi Shirabe: The minimum Manhattan network problem: Approximations and exact solutions. *Comput. Geom. Theory Appl.*, 35(3):188–208, 2006. http://dx.doi.org/10.1016/j.comgeo.2005.09.004.

[CGS11] Francis Chin, Zuyu Guo, and He Sun: Minimum Manhattan network is NP-complete. *Discrete Comput. Geom.*, 45:701–722, 2011. http://dx.doi.org/10.1007/s00454-011-9342-z.

[CLZ93] Jason Cong, Kwok Shing Leung, and Dian Zhou: Performance-driven interconnect design based on distributed RC delay model. In *Proc. 30th IEEE Conf. Design Automation (DAC'93)*, pages 606–611, 1993. http://dx.doi.org/10.1109/DAC.1993.204019.

[CNV08] Victor Chepoi, Karim Nouioua, and Yann Vaxès: A rounding algorithm for approximating minimum Manhattan networks. *Theor. Comput. Sci.*, 390(1):56–69, 2008, ISSN 0304-3975.

[DFK$^+$12] Aparna Das, Krzysztof Fleszar, Stephen G. Kobourov, Joachim Spoerhase, Sankar Veeramoni, and Alexander Wolff: Polylogarithmic Approximation for Generalized Minimum Manhattan Networks. *ArXiv e-prints*, March 2012.

[DGK$^+$11] Aparna Das, Emden R. Gansner, Michael Kaufmann, Stephen Kobourov, Joachim Spoerhase, and Alexander Wolff: Approximating minimum Manhattan networks in higher dimensions. In Camil Demetrescu and Magnús M. Halldórsson (editors): *Proc. 19th Annu. Europ. Symp. on Algorithms (ESA'11)*, volume 6942 of *LNCS*, pages 49–60. Springer, 2011. http://dx.doi.org/10.1007/978-3-642-23719-5_5.

[FS08] Bernhard Fuchs and Anna Schulze: A simple 3-approximation of minimum Manhattan networks. Technical Report 570, Zentrum für Angewandte Informatik Köln, February 2008. http://e-archive.informatik.uni-koeln.de/570, http://e-archive.informatik.uni-koeln.de/570.

[GLN01] Joachim Gudmundsson, Christos Levcopoulos, and Giri Narasimhan: Approximating a minimum Manhattan network. *Nordic J. Comput.*, 8:219–232, 2001.

[GP66] E. N. Gilbert and H. O. Pollak: Steiner minimal trees. 16:1–29, 1966.

[GSZ11] Zeyu Guo, He Sun, and Hong Zhu: Greedy construction of 2-approximate minimum Manhattan networks. *Internat. J. Comput. Geom. Appl.*, 21(3):331–350, 2011.

[LR00] Bing Lu and Lu Ruan: Polynomial time approximation scheme for the rectilinear Steiner arborescence problem. *J. Comb. Optim.*, 4(3):357–363, 2000.

[MSU09] Xavier Muñoz, Sebastian Seibert, and Walter Unger: The minimal Manhattan network problem in three dimensions. In Sandip Das and Ryuhei Uehara (editors): *Proc. 3rd Int. Workshop Algorithms Comput. (WALCOM'09)*, volume 5431 of *LNCS*, pages 369–380. Springer, 2009.

[Nou05] Karim Nouioua: *Enveloppes de Pareto et Réseaux de Manhattan: Caractérisations et Algorithmes.* PhD thesis, Université de la Méditerranée, 2005.

[NSS74] Ludwig Nastansky, Stanley M. Selkow, and Neil F. Stewart: Cost-minimal trees in directed acyclic graphs. *Zeitschrift Oper. Res.*, 18(1):59–67, 1974.

[RSHS92] Sailesh Rao, P. Sadayappan, Frank Hwang, and Peter Shor: The rectilinear Steiner arborescence problem. *Algorithmica*, 7:277–288, 1992. http://dx.doi.org/10.1007/BF01758762.

[SS00] Weiping Shi and Chen Su: The rectilinear Steiner arborescence problem is NP-complete. In *Proc 11th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA'00)*, pages 780–787, 2000. http://dl.acm.org/citation.cfm?id=338219.338639.

[Yao82] Andrew Chi Chih Yao: On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM J. Comput.*, 11(4):721–736, 1982, ISSN 0097-5397 (PRINT), 1095-7111 (ELECTRONIC).

[Zac00] Martin Zachariasen: On the approximation of the rectilinear Steiner arborescence problem in the plane. Unpublished manuscript, see http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.4529, 2000.

A. List of Important Notations

Notion	Meaning
RSAP	Rectilinear Steiner arborescene problem
RSA	Rectilinear Steiner arborescence (feasible network)
near-optimal RSA	Near-optimum solution to RSAP (network)
MMN	Minimum Manhattan Network problem
GMMN	Generalized minimum Manhattan network problem
GMN	Generalized Manhattan network (feasible network)
minimum GMN	Optimum solution to GMMN (network)
n	Number of terminal pairs of a GMMN instance
x_i	i-th coordinate
$x_i\left(p\right)$	Value of the x_i-coordinate of a point p
$\rho_d(n)$	Approximation ratio for d-dimensional GMMN
$\rho_{d,j}(n)$	Approximation ratio for j-separated d-dimensional GMMN
$\|N\|$	Length of a network N
$\lvert P\rvert$	Cardinality of a set P
$\lvert x_i\left(p\right), x_i\left(q\right)\rvert$	Distance of points p and q along axis x_i
flat	Axis-Aligned Flat
(v, d)-flat	Flat with v fixed coordinates and $d - v$ free coordinates
f-dimensional flat	Flat with f free coordinates

Table A.1.: List of notations