The Generalized Minimum Manhattan Network Problem (GMMN) – Scale-Diversity Aware Approximation and a Primal-Dual Algorithm

Stefan Funke* Martin P. Seybold*

Abstract

In the d-dimensional Generalized Minimum Manhattan Network (d-GMMN) problem one is interested in finding a minimum cost rectilinear network N connecting a given set R of n pairs of points in $\mathbb{R}^d$ such that each pair is connected in N via a shortest Manhattan path. The problem is known to be $\mathcal{NP}$-complete and does not admit a FPTAS, the best known upper bound is an $\mathcal{O}(\log^{d+1} n)$-approximation for $d > 2$ and an $\mathcal{O}(\log n)$-approximation for $d = 2$ by Das et al. [3].

In this paper we provide some more insight into the problem and develop two new algorithms, a 'scale-diversity aware' algorithm with an $\mathcal{O}(\mathcal{D})$ approximation guarantee for $d = 2$. Here $\mathcal{D}$ is a measure for the different 'scales' that appear in the input, $\mathcal{D} \in \mathcal{O}(\log n)$ but potentially much smaller depending on the problem instance. Moreover, this implies that a potential proof of $\mathcal{O}(1)$-inapproximability for 2-GMMN *requires* gadgets of many different scales in the construction. The other algorithm is based on a primal-dual scheme solving a more general path covering problem. On 2-GMMN it performs pretty well in practice with good a posteriori, instance-based approximation guarantees. Furthermore, it can be extended naturally to deal with obstacle avoiding requirements.

1 Introduction

We study the d-dimensional Generalized Minimum Manhattan Network problem:

Input: A set R of n pairs of points (terminals) in $\mathbb{R}^d$.

Goal: Determine a finite set N of axis parallel line segments of minimum total length $c(N)$ that connects each pair via a shortest Manhattan path.

The problem is closely related to the rectilinear Steiner network problem, where the goal is to connect designated pairs in a minimum cost network but not necessarily on shortest paths. In the context of circuit design ($d = 2$ or $d = 3$) – one of the main application areas of many Steiner-type problems – restricting to shortest paths for interconnection corresponds to keeping the latency low.

*Formale Methoden der Informatik, University of Stuttgart, Germany, {funke, seybold}@fmi.uni-stuttgart.de

d-GMMN is a generalization of the d-dimensional Minimum Manhattan Network problem (d-MMN, all pairs of terminals are present in R) and the d-dimensional Rectilinear Steiner Arborescence problem (d-RSA, each pair in R contains the origin as one of its elements). Already 2-RSA is $\mathcal{NP}$-complete [10] and 2-MMN does not admit a fully polynomial time approximation scheme (FPTAS) unless $\mathcal{P} = \mathcal{NP}$ [2].

For 2-RSA several $\mathcal{O}(1)$-approximation algorithms are known. The algorithm given in [9] is conceptually simple and achieves a 2-approximation. There is also a polynomial time approximation scheme (PTAS) based on Arora's shifting-technique [7].

The situation for MMN problems is slightly different. We know that 3-MMN does not admit a PTAS, unless $\mathcal{P} = \mathcal{NP}$ [4, 8]. [5] gives an $\mathcal{O}(1)$-approximation. Several subsequent papers improve on the running time and the constant factor of the $\mathcal{O}(1)$-approximation. [1] gives the first 2-approximation for 2-MMN. Their approach is based on a multiphase flow ILP formulation of polynomial size and iteratively rounding an optimal fractional solution. In their discussion section they introduce GMMN and point out that their approach might not translate easily.

[3] were the first to come up with an $\mathcal{O}(\log^{d+1} n)$ approximation for d-GMNN. For $d = 2$ they could even prove an $\mathcal{O}(\log n)$ upper bound. Their approach follows the divide and conquer paradigm exhibiting sets of pairs which can be connected via a common point in space; these base cases are solved with a known d-RSA approximation algorithm. They also provide an instance showing the analysis for their algorithm is essentially tight (not excluding other, better algorithms).

The results in this paper are along the lines of [3] but the approximation ratio of our 'scale-diversity aware' algorithm depends rather on the *scale diversity* than the arrangement of the input. For a set $U \subseteq (1, n]$ of numbers, we denote by $g(U) := |\{i \in \mathbb{N}_0 \mid \exists u \in U : 2^i < u \leq 2^{i+1}\}|$ the *scale diversity* of U. Intuitively, $g(U)$ describes how many different magnitudes of numbers appear in the set U. Clearly $g(U) \in \mathcal{O}(\log n)$. In an instance for 2-GMMN, each pair $(p, q) \in R$ naturally gives rise to 2 distance values $d_j(p, q)$ – their distance in the j-th coordinate. In the later we argue that scaling such that the biggest l_1 distance of a pair is exactly n does not affect the form of solutions. After scaling,

pairs with l_1 distance of no more than a constant can be neglected when aiming at a constant approximation. With $U_j := \{\ d_j(p,q) \in (1,n] \mid (p,q) \in R\ \}$ we denote the spread in dimension j of the pairs after scaling. The quantity $\mathcal{D} := \max\{g(U_1), g(U_2)\}$ denotes the *scale diversity* of R and essentially captures how many really different magnitudes of spreads with respect to the maximum extent appear in R.

1.1 Our Contribution

The paper presents two algorithms for GMMN problems. The approximation ratio for the first algorithm depends on the *scale diversity* of the terminal pairs in R. More concretely, we show that our algorithm computes an $\mathcal{O}(\mathcal{D})$ approximation to the 2-GMMN problem. This result always matches the result in [3], but is better if the scale diversity of R is small, e.g. $\mathcal{D} \in o(\log n)$. Moreover, this implies that a potential proof of $\mathcal{O}(1)$-inapproximability for 2-GMMN *requires* gadgets of many different scales in the construction.

While we consider the first algorithm to be of rather theoretical interest only, our second algorithm – following a primal-dual approach based on an ILP formulation of the d-GMMN problem – seems more practical. While we could not prove an a priori approximation ratio, it inherently produces lower bounds during its execution which – at least in our experiments with an implementation for $d = 2$ – turn out to be very close to the costs of the computed networks.

2 Scale-Diversity Aware Approximation

2.1 Decomposition Properties

Let us first make some general observations about the d-GMNN problem.

Lemma 1 *The cost of an optimal solution to any subset R' of a GMMN instance R is a lower bound to the cost of an optimal solution for R.*

Proof. Consider an optimal solution N for R having a strictly lower cost than an optimal solution N' for R'. However, N contains a shortest path for each pair in R'. A contradiction to the optimality of N'. □

This simple lemma gives rise to the following decomposition property of GMMN instances:

Lemma 2 *Let $R = R_1 \cup \ldots \cup R_k$ (not necessarily disjoint). If each N_i is an α_i-approximation for R_i, then $N = \bigcup_i N_i$ is a solution for R with $c(N) \leq \mathrm{OPT}(R) \cdot \sum_i \alpha_i$.*

Proof. N clearly connects each pair in R. By Lemma 1 we have $\mathrm{OPT}(R_i) \leq \mathrm{OPT}(R)$ and $c(N) \leq \sum_i c(N_i) \leq \sum_i \alpha_i \cdot \mathrm{OPT}(R_i) \leq \sum_i \alpha_i \cdot \mathrm{OPT}(R)$. □

2.2 Shape Properties

Let us now turn to more shape-dependent properties of GMMN. We first show that if all terminal pairs $(p,q) \in R$ exhibit about the same shape, that is, if in one dimension every terminal pair has about the same extent, then we can decompose R into constantly many, not necessarily disjoint instances. Each of these instances has a very special structure which allows for a constant approximation. To simplify notation, we associate with each pair $(p,q) \in R$ the minimum area axis-parallel **box** r having p and q as corners.

Lemma 3 (Shapes in 2-GMMN) *Let $\gamma \geq 0$ be constant and R a 2-GMMN instance. If for some dimension $j \in \{1,2\}$ each pair $(p,q) \in R$ has $\gamma < d_j(p,q) \leq 2\gamma$, then R can be decomposed into a constant number of (not necessarily disjoint) instances $R = R_0 \cup R_1 \cdots \cup R_5$. Boxes in each R_i have either a common axis parallel intersection line or no intersection at all.*

Proof. Let j be the dimension fulfilling the shape property. Consider axis parallel lines with distance γ in dimension j. Each box of a pair $(p,q) \in R$ intersects at least one and at most three lines. If boxes r and r' contain lines i and i' respectively, then $|i - i'| \geq 6$ implies that $r \cap r' = \emptyset$. This gives rise to the decomposition where $R_i \subseteq R$ consists of all pairs that contain a line k with $k \equiv i \mod 6$ in their box. □

2.3 Scale Properties

We call a d-GMMN instance R' *scaled*, if it is derived from an instance R by dividing each coordinate of a terminal by a fixed $\sigma > 0$. We have an one-to-one correspondence between their solutions, because a line segment between two points can be scaled up or down in the same way. Moreover, the cost of a network is σ times the cost of the corresponding network. If we have an instance of d-GMMN where some of the boxes associated with terminal pairs are very 'small' compared to the other boxes, we can essentially connect them naively without losing more than a constant factor in the total connection cost.

Lemma 4 *Let R be a scaled d-GMMN instance such that $n = \max_{(p,q)\in R} ||p - q||_1$ and $R_\varepsilon := \{(p,q) \in R \mid ||p - q||_1 \leq d\}$. If N' is an α-approximation for the instance $R - R_\varepsilon$, then R can be approximated within $\mathcal{O}(\alpha)$.*

Proof. As there exists a (p,q) with $||p - q||_1 = n$, we know that the cost of the optimal solution to R must be at least n. Connecting all terminal pairs in R_ε has cost at most nd. The lemma follows. □

2.4 An Algorithm for 2-GMMN

The algorithm in this section is based on the combination of decomposition, shape and scale properties. We borrow the following lemma to solve the partitions in each of the six covers of one shape class.

Lemma 5 (Lemma 8 in [3]) *Let R be a 2-GMMN instance. If all boxes of R have a common, axis-parallel intersection line, then R can be approximated within $\mathcal{O}(1)$.*

Algorithm 1 Scale-Diversity Aware Approximation

1. Scale the instance R such that the biggest l_1 metric of a pair is exactly n.
2. Partition R into
   $R_\varepsilon = \{(p,q) \in R \mid ||p-q||_1 \leq 2\}$,
   $R_1 = \{(p,q) \in R - R_\varepsilon \mid d_1(p,q) \geq d_2(p,q)\}$,
   and $R_2 = R - (R_1 \cup R_\varepsilon)$.
3. Partition R_1 into $g(U_1)$ shape classes and solve.
4. Partition R_2 into $g(U_2)$ shape classes and solve.
5. Solve R_ε trivially.

Essentially, after scaling we partition the problem instance into three instances R_ε, R_1 and R_2, where R_ε contains terminal pairs with 'very small' boxes, R_1 all terminal pairs whose box is wider than tall, and R_2 the remaining ones. Then for each R_i we consider the scale classes in dimension i and solve each of them using Lemmas 3 and 5. The R_ε are solved trivially.

Theorem 6 (2-GMMN) *If R is a 2-GMMN instance with scale diversity $\mathcal{D} = \max\{g(U_1), g(U_2)\}$, then Algorithm 1 computes an $\mathcal{O}(\mathcal{D})$ approximate solution.*

Proof. The scaling property preserves optimality of solutions. The pairs in R_ε can essentially be ignored according to Lemma 4. R_1 can be decomposed into $g(U_1)$ shape classes of boxes. Using Lemma 3, each shape class in R_1 can again be decomposed in a constant number of instances each of which allows a partition in boxes sharing a common intersection line or not having a intersection at all. The disjoint parts of these instances with the common intersection line are solved with Lemma 5 within a constant factor of their optimum – their disjoint union remains within a constant factor of their optimum. Using Lemma 2 the approximation ratio of the possibly not disjoint union follows. The same argument holds for R_2. □

Clearly $\mathcal{D} \in \mathcal{O}(\log n)$ but might be smaller in some applications. We give an example that occupies many shape classes. Let p_x denote the point $(x, 0)$ and q_y the point $(0, y)$. The arrangement $\{p_{n/2}, p_{n/4}, \ldots, p_{1/2}\} \times \{q_{n/2}, q_{n/4}, \ldots, q_{1/2}\}$ has an optimal solution of cost n. Consider the 2-GMMN instance R that contains $n/\log_2^2(n)$ disjoint copies of this arrangement ($|R| = n$). We have an optimal solution of cost $n^2/log_2^2(n)$ and all shape classes in dimension 1 and 2 are occupied – e.g. $g(U_1), g(U_2) \in \Theta(\log n)$.

3 Practical Algorithms

Let us first make the following observation which reduces the potentially very large number of line segments to consider for the network N.

3.1 Restriction to Hanan Grid

We consider the undirected, simple graph $\mathcal{H}(R)$ induced by the instance R. Let P_i be the projection of the terminals in R on the i-th coordinate. The vertices are the Cartesian product $\prod_{i=1}^{d} P_i$ and have an edge if and only if they are identical in all but neighbored in one coordinate. We call the difference in this coordinate the cost c_e of the edge e. Any d-GMMN instance with n pairs has a Hanan Grid of size at most $(2n)^d$ vertices and $\mathcal{O}(d(2n)^d)$ edges. This is polynomial for fixed d. Given two vertices we call a simple path connecting them a **m-path** if the sequence of each coordinate of the vertices along the path is monotonous.

The following property of the Hanan Grid is common with many rectilinear problems. To make the paper more self contained, we give a formulation of a simple proof that follows the argument for Abstract Steiner Minimal Trees[13] and yet has the generality of d dimensions as in [11].

Theorem 7 (Hanan Grid for d-GMMN) *For any d-GMMN instance R, there is an optimal solution $\overline{N}$ using only edges of $\mathcal{H}(R)$ as line segments.*

Given a solution N of line segments for R. We call a point in the cut of at least two segments a *node*. This prove constructs a set of line segments where nodes coincide with vertices of $\mathcal{H}(R)$. We apply a sweeping hyperplane argument over one dimension after the other:

Proof. Let N be an optimal solution to R with minimal number of non-vertex nodes. We describe the sweep over x_1. After the sweep the x_1 coordinate of each node in N will be identical to one in P_1.

Consider the non-vertex nodes of N with maximum x_1 coordinates and their hyperplane h containing them. Now above or below h means that a point has a higher or respectively lower x_1 coordinate than h. Inductively, every node above h already has x_1-coordinates as desired. Let ε^+ denote the distance in x_1 to the next above coordinate in P_1 and ε^- the distance in x_1 to the smaller of either the next lower set of such nodes of N

or to the next lower point in P_1. Let also S^+ and S^- denote the set of line segments parallel to x_1 and incident to a node in h above and below respectively. The change in total cost for moving line segments of N contained in h along the x_1 direction for an $-\varepsilon^- < \varepsilon < \varepsilon^+$ is

$$\Delta(\varepsilon) = \varepsilon \cdot (|S^-| - |S^+|) \geq 0.$$

If $|S^+| \geq |S^-|$ we simply move the segments in h upwards by ε^+ otherwise we move them downwards by ε^-. In both cases we would only lower the total costs. Moreover, by moving all nodes in h an arbitrary m-path of N crossing h remains monotonous in all x_i. After this sweep, the x_1 coordinates of a node in N is identical to one in P_1 and the total number of non-vertex nodes did not increase. After d sweeps no non-vertex nodes are left. □

The Hanan Grid enables us to compute an optimal solution to d-GMMN with a brute-force approach. Every feasible network consists of a covering of n *m-paths* connecting a pair of terminals each. The number of m-paths for two vertices is generally exponential in n. For small instances however, one can enumerate all feasible networks living on the Hanan Grid to find an optimal GMMN solution.

3.2 Lower Bounds for Path Covering Problems

A canonical framework to derive lower bounds for discrete problems is the theory of Integer Linear Programs [12]. Unfortunately, the natural cut formulation of the GMMN problem does not exhibit a *weakly supermodular* function for the number of required edges in a cut. Therefore, we cannot simply apply the known iterative rounding procedure of [6]. We developed several formulations to the problem. However, only the following shows 'good' (c.f. Section 4) behavior. Consider $\mathcal{H}\big(\{p^{(1)}, q^{(1)}\}, \ldots, \{p^{(n)}, q^{(n)}\}\big) = (V, E)$ of a given instance. Note that some $p^{(i)}$ and $q^{(j)}$ might be identical. We use the superscript to denote their context.

By $\Pi_{p^{(i)}}$ we denote the set of prefixes of all m-paths connecting $p^{(i)}$ to $q^{(i)}$. Given a subset $E' \subseteq E$ and a pair's terminal $p^{(i)}$ we can define a subnetwork

$$S(p^{(i)}, E') = \{e \in E' \mid \exists \pi \in \Pi_{p^{(i)}} \text{ with } e \in \pi \subseteq E'\}.$$

The partial networks for a pair's terminal are

$$\mathcal{S}_{p^{(i)}} = \bigcup_{E' \subseteq E} \{\, S(p^{(i)}, E') \mid q^{(i)} \text{ no vertex in } S(p^{(i)}, E')\}.$$

They essentially capture all situations where at least one edge is missing for a m-path from $p^{(i)}$ to $q^{(i)}$. We also define a multiset to capture all *unfinished situations*

$$\mathcal{S} = \biguplus_i \mathcal{S}_{p^{(i)}} \cup \mathcal{S}_{q^{(i)}}$$

in which each network is rooted with one terminal and i. The *boundary* of a rooted partial network in the multiset $\mathcal{S}$ is given by

$$\delta(S_{p^{(i)}}) = \Big\{ e \in E - S_{p^{(i)}} \Big| e \text{ extends a m-path in } S_{p^{(i)}} \Big\}.$$

Note that one could easily integrate obstacle avoiding requirements for *some* of the pairs in this boundary function.

We define an indicator variable $x_e \in \{0, 1\}$ for every edge of the Hanan Grid. We have an one-to-one correspondence between optimal solutions on $\mathcal{H}(R)$ and optimal solutions of the following primal ILP:

$$\begin{aligned} \min : & \sum_{e \in E} c_e \cdot x_e \\ \text{s.t.} & \sum_{e \in \delta(S)} x_e \geq 1 \qquad \forall S \in \mathcal{S} \\ & x_e \in \{0, 1\} \end{aligned}$$

The primal LP is given by relaxing to non-integral variables $\tilde{x}_e \geq 0$. Any optimal solution $\tilde{\mathbf{x}}$ to the LP is a lower bound to the ILP since any optimal ILP solution $\mathbf{x}$ is also feasible for the LP.

In our primal LP ($\min \mathbf{c} \cdot \tilde{\mathbf{x}}$, $\mathbf{A} \cdot \tilde{\mathbf{x}} \geq \mathbf{1}$) all variables and objective-coefficients are non-negative. The constraint-coefficients are either 0 or 1. Consider a single row constraint of $\mathbf{A}$ multiplied by $\varepsilon \geq 0$. If ε is small enough (e.g. $\varepsilon = \min\{c_e | e \in E\}$) the multiplied constraint-coefficients are component wise smaller than the objective-coefficients. Therefore, the right hand side $1 \cdot \varepsilon$ is a lower bound to the objective for arbitrary non-negative $\tilde{\mathbf{x}}$. This canonical method gives the dual LP with the objective to find a maximal lower bound using a conical combination of constraints:

$$\begin{aligned} \max : & \sum_{S \in \mathcal{S}} 1 \cdot \tilde{y}_S \\ \text{s.t.} & \sum_{S \in \mathcal{S} \,:\, e \in \delta(S)} \tilde{y}_S \leq c_e \qquad \forall e \in E \\ & \tilde{y}_S \geq 0 \end{aligned}$$

This described property is known as *weak duality*:

Lemma 8 *For any dual feasible assignment $\tilde{\mathbf{y}}$ we have a lower bound to the objective of an optimal solution for the primal LP and ILP for d-GMMN:*

$$\mathbf{1} \cdot \tilde{\mathbf{y}} \leq \mathbf{c} \cdot \tilde{\mathbf{x}} \leq \mathbf{c} \cdot \mathbf{x}$$

The maximum ratio $(\mathbf{c} \cdot \mathbf{x})/(\mathbf{c} \cdot \tilde{\mathbf{x}})$ observed on instances of a problem is called the *integrality gap* of a particular formulation.

3.3 A Primal-Dual Algorithm for Path Covering

The idea of a primal-dual scheme is to start with a pair of assignments. A non-feasible, integral assignment of the primal ILP ($\mathbf{x} = \mathbf{0}$) and a complementary feasible, however far from optimal relaxed assignment ($\tilde{\mathbf{y}} = \mathbf{0}$) of the dual LP. By alternately improving the dual assignment and making the primal assignment more feasible one obtains a lower bound that *might* have a close relation to the eventually constructed primal feasible assignment. We now describe the process more precisely:

Given the above primal ILP and a not feasible set of edges $\mathbf{x}$ – e.g. lacking a m-path for the pair $\{p^{(i)}, q^{(i)}\}$. A maximal partial network $S(p^{(i)}, \mathbf{x})$ can be found with a simple depth-first search on m-paths from $p^{(i)}$ to $q^{(i)}$ that only traverses edges that are given in $\mathbf{x}$ and the box of $\{p^{(i)}, q^{(i)}\}$. Therefore this generally exponential number of constraints does not need to be stored – each partial network can be constructed if needed.

The algorithm first creates an ordered sequence of edges and increases the lower bound $\sum_{S \in \mathcal{S}} \tilde{y}_S = \tilde{Y}$. After the sequence is feasible, unnecessary edges are pruned in reverse order.

Algorithm 2 Primal-Dual Scheme for Path Covering

```
x := ∅                                          ▷ The primal assignment
Ỹ := 0                                          ▷ The dual's objective
c' := c                                         ▷ Copy of edge costs
while x not primal feasible do
    I := {i | {p^(i), q^(i)} not m-connected in (V, x)}
    N := ⊎_{i∈I} {S(p^(i), x), S(q^(i), x)}
    Let ν(e) = |{S ∈ N | e ∈ δ(S)}|
    ε := min{ c'[e]/ν(e) | ν(e) > 0 }
    for e with ν(e) > 0 do
        c'[e] = c'[e] − ν(e) · ε
        if c'[e] = 0 then
            x.append(e)
        end if
    end for
    Ỹ := Ỹ + ε · |N|
end while
for e ∈ x in LiFo order do
    if x − {e} primal feasible then
        x := x − {e}
    end if
end for
```

Consider the pseudo code of Algorithm 2. Termination of the algorithm is given since at least one of the polynomially many edges is added in each step. The variable $\tilde{Y}$ stores the dual's objective: $\tilde{\mathbf{y}}$ is feasible for the relaxed dual problem at the beginning. If a dual constraint becomes tight, the edge e is added and no set containing $e \in \delta(S)$ will be increased in any subsequent loop. Hence $\tilde{\mathbf{y}}$ remains feasible for the dual LP and the pruned set $\mathbf{x}$ remains primal feasible by construction.

Two implementations for this algorithm are obvious. We can perform at most $2n$ depth-first searches during each of the while and the for loop. This shows polynomial runtime for a fixed dimension d. Here we do not even need to keep $\mathcal{H}(R)$ in memory but can work directly on the sparse set of coordinates (P_i) occurring in R. In the other approach we annotate the edges of $\mathcal{H}(R)$ with additional $\mathcal{O}(n)$ bits for the context an edge is used. Further a min heap can be used to point to the currently active boundary. The naive implementation for 2-GMMN enabled us to find solutions and lower bounds for instances up to a size of $n = 128$, which is considerably more than $n = 7$ with the brute-force approach. See Figure 1 for execution times.

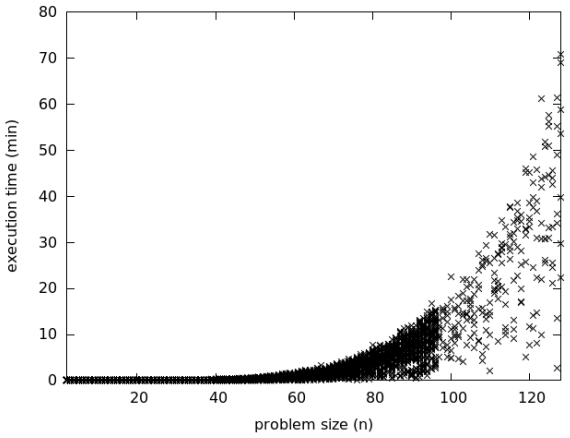

Figure 1: Execution times of the primal-dual algorithm in minutes on each of the sampled random instances of 2-GMMN. We implemented the naive deep first search approach without min-heaps or annotation of the Hanan Grid. Calculations were performed on an Intel Core i7-3930K 3.2 GHz. Each problem instance was solved single threaded and we solved up to 4 problem instances concurrently.

4 Experimental Results

The proposed primal-dual strategy with reverse pruning has the advantage of also constructing a lower bound to the optimal cost of a solution. Let $\mathbf{x}_{\mathrm{OPT}}$ and $\tilde{\mathbf{x}}_{\mathrm{OPT}}$ be optimal solutions for the primal ILP and LP respectively. Consider $\mathbf{x}$ and $\tilde{Y}$ as calculated from Algorithm 2. Given Lemma 8, we have that the actual approximation ratio is bounded:

$$\frac{\mathbf{c} \cdot \mathbf{x}}{\mathbf{c} \cdot \mathbf{x}_{\mathrm{OPT}}} \leq \frac{\mathbf{c} \cdot \mathbf{x}}{\mathbf{c} \cdot \tilde{\mathbf{x}}_{\mathrm{OPT}}} \leq \frac{\mathbf{c} \cdot \mathbf{x}}{\tilde{Y}}.$$

The rightmost term – we call it *approximation guarantee* in the following – is explicitly calculated on an evaluated instance.

Along with the $\mathcal{O}(\log n)$ approximation algorithm, Das et al. also provided a family of recursively defined

instances that show the tightness of the approximation ratio of their algorithm. We evaluated several of those instances and found them all to be solved optimally with a tight lower bound by Algorithm 2 (data not shown).

4.1 Random Sampled Instances

We evaluated Algorithm 2 on random 2-GMMN instances. To rule out solely good behavior on uniformly chosen points of some square, we sampled different aspect ratios (x-range / y-range) and densities (number of vertices in x / x-range). For a given number of pairs n of a instance, we first choose an aspect ratio $a \in \{1, \ldots, 9\}$ uniformly at random. Second we choose a density $d \in \{1n, 2n, \ldots, 10n\}$ uniformly at random. Finally we choose n times the two x-coordinates in $\{1, \ldots, d \cdot a\}$ and the two y-coordinates in $\{1, \ldots, d\}$ uniformly at random. Out of these, we evaluated a total of 11274 random instances of several sizes (150 replicates for each $n \in \{2, \ldots, 64\}$, 50 for each $n \in \{65, \ldots, 96\}$ and 7 for each $n \in \{97, \ldots, 128\}$). Figure 2 shows the achieved distribution of the sampling strategy in the parameter space. See Figure 3 for the observed approximation guarantees.

On the positive side, Algorithm 2 seems to find solutions that are optimal and close to optimum in many cases. Irrespective of size n, none of the instances arranged in a rather quadratic bounding box show an approximation guarantee bigger than 2.0, which also holds for the majority of other instances we evaluated.

On the negative side, we found approximation guarantee worse than 2.0 on instances having many terminals and a *very high* degeneracy in the aspect ratio and density at the same time. Using this observation, the worst experimentally observed approximation guarantee on a random 2-GMMN instance (n=70, aspect ratio=10000/1, density=2/100000) was 3.385.

4.2 Looseness of the Lower Bound

Despite the experimental close relation between lower bound $\tilde{\mathbf{Y}}$ and cost $\mathbf{c} \cdot \mathbf{x}$ in many situations, one cannot prove this ratio to be constant. In the following we describe a family of instances that are solved optimally by Algorithm 2. However, the lower bound calculated can be off arbitrarily. Consider the 2-GMMN instance $T_k = \{(0,0)\} \times \{(x,y) \in \mathbb{N}_0^2 \mid 0 < x + y < k\}$ having $n_k = k(k+1)/2 - 1$ pairs. Each edge in $\mathcal{H}(T_k)$ has cost 1 and each right hand side of a pair requires at least one incoming edge for this pair. This implies the cost of an optimal solution to T_k is at least n_k. In fact the pruning step in Algorithm 2 already suffices that the cost of the result is n_k, too. The lower bound as calculated of Algorithm 2 for this family of instances is not tight and the approximation guarantee $\mathbf{c} \cdot \mathbf{x}/\tilde{Y}$ grows asymptotically with k (c.f. Figure 4).

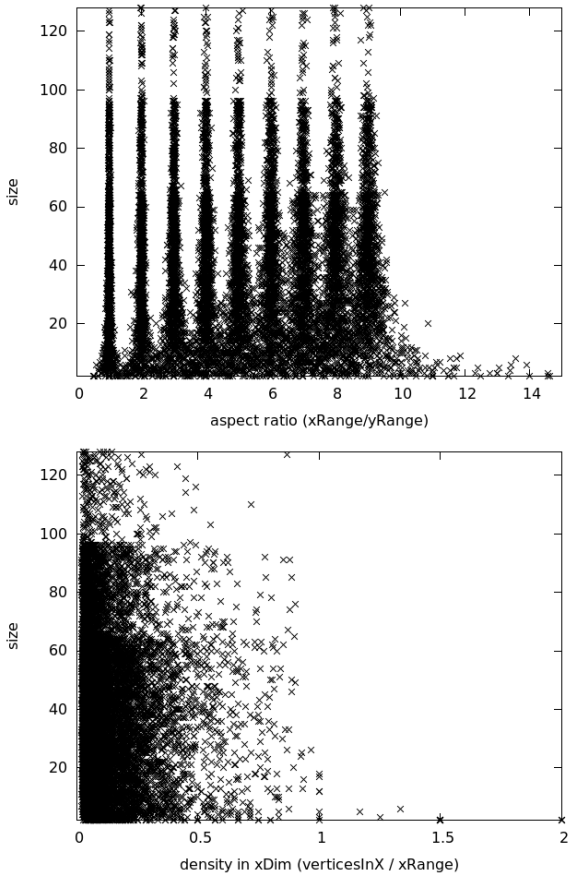

Figure 2: Distribution of sampled random instances in the parameter space. Instance size by aspect ratio (top) and by density (bottom).

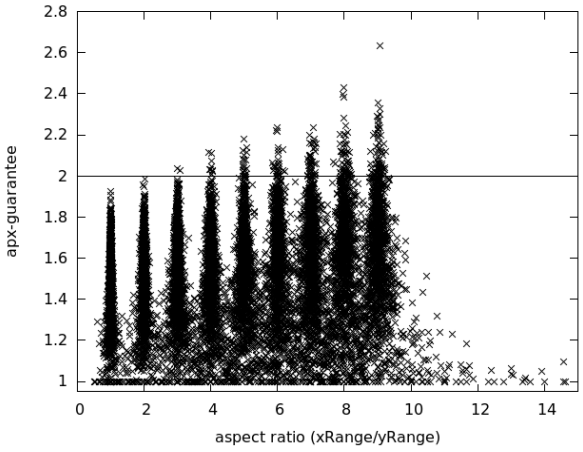

Figure 3: Approximation guarantees of sampled random instances by aspect ratio. Each aspect ratio class contains instances of almost each size $n \in \{2, \ldots, 128\}$.

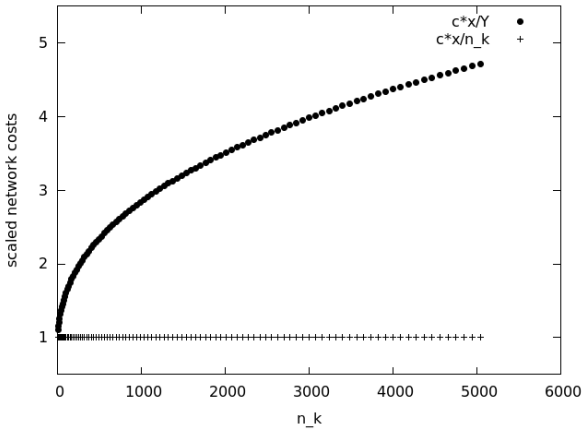

Figure 4: Asymptotic growth of the approximation guarantee of the instances T_k. We spare a lengthy recursive and non intuitive derivation of an upper bound for $\tilde{\mathbf{Y}}$ on the instances T_k and give a picture of the results. Note that the solution of algorithm 2 is optimal in this case.

However, this does not give implications on the integrality gap of our formulation. It might well exist a dual assignment with a higher lower bound than the calculated $\tilde{\mathbf{Y}}$. Unfortunately we were not able to prove a bound of $\mathbf{c} \cdot \mathbf{x}$ against the cost of an optimal solution.

5 Conclusion and Open Problems

We have presented an algorithm that achieves an approximation factor constant in the *scale diversity* $\mathcal{D}$ of a 2-GMMN problem. This result always matches the $\mathcal{O}(\log n)$-approximation given in [3], but is considerably better if the scale diversity is small. Moreover, this algorithm implies that a potential $\mathcal{O}(1)$-inapproximability proof for 2-GMMN *must* use gadgets of many different scales. This is in contrast to the proof in [2]. There the existence of a FPTAS for 2-MMN is disproved (unless $\mathcal{P} = \mathcal{NP}$) by arranging gadgets of about the same shape. On the practical side we have presented a primal-dual algorithm for a more general path covering problem in graphs. The algorithm is applicable for d-GMMN and generalizations that require obstacle avoiding for all *or some* of the pairs. The approach performs well on instances of 2-GMMN and produces instance based lower bounds that are close to the cost of the solutions. We could not prove a priori guarantees in the d-GMMN setting. However, we show that the lower bounds are not tight in some cases. This actually means that in some cases the approximation ratio is even better than guaranteed by the lower bound. The primal-dual algorithm also solves instances optimal that are worst case instances for [3].

The $\mathcal{O}(1)$ approximability of 2-GMMN is still unsolved. Given the good practical approximations achievable and the rather specific requirements for $\mathcal{O}(1)$-inapproximability proofs for 2-GMMN – we still conjecture that 2-GMMN admits an $\mathcal{O}(1)$ approximation.

References

[1] Victor Chepoi, Karim Nouioua, Yann Vaxès. *A rounding algorithm for approximating minimum Manhattan networks.* Theoretical Computer Science, 390 , p56–69, 2008.

[2] Francis Y. L. Chin, Zeyu Guo, He Sun. *Minimum Manhattan Network is NP-Complete.* Discrete & Computational Geometry, 45(4), p701–722, 2011.

[3] Aparna Das, Krzysztof Fleszar, Stephen G. Kobourov, Joachim Spoerhase, Sankar Veeramoni, Alexander Wolff. *Approximating the Generalized Minimum Manhattan Network Problem.* Proc. 24st International Symposium on Algorithms and Computation (ISAAC'13), p722–732, 2013.

[4] Birgit Engels. *The Transitive Minimum Manhattan Subnetwork Problem in 3 Dimensions.*, Discrete Applied Mathematics, 158(4), p298–307, 2010.

[5] Joachim Gudmundsson, Christos Levcopoulos, Giri Narasimhan. Approximating a Minimum Manhattan Network. Nordic J. Comput., 8, p219–232, 2001.

[6] Kamal Jain. *A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem.* Combinatorica, 21, p39–60, 1998.

[7] Bing Lu, Lu Ruan. *Polynomial Time Approximation Scheme for the Rectilinear Steiner Arborescence Problem.* Journal of Combinatorial Optimization, 4(3), p357–363, 2000.

[8] Xavier Muñoz, Sebastian Seibert, Walter Unger. *The Minimal Manhattan Network Problem in Three Dimensions.* Workshop on Algorithms and Computation, p369–380, 2009.

[9] Sailesh K. Rao, P. Sadayappan, Frank K. Hwang, Peter W. Shor. *The Rectilinear Steiner Arborescence Problem.* Algorithmica, 7, p277–288, 1992.

[10] Weiping Shi, Chen Su. *The Rectilinear Steiner Arborescence Problem is NP-Complete.* SIAM Journal on Computing, 35(3), p729–740, 2006.

[11] Timothy Law Snyder. *On the Exact Location of Steiner Points in General Dimension.* SIAM Journal on Computing, 21(1), p163–180, 1992.

[12] David P. Williamson, David B. Shmoys. *The Design of Approximation Algorithms.* http://www.designofapproxalgs.com, 2011.

[13] Martin Zachariasen. *A Catalog of Hanan Grid Problems.* Networks, 38, p200–1, 2001.