

最小マンハッタンネットワーク問題の近似について

山崎 康行 (Yasuyuki Yamasaki)

九州工業大学大学院情報工学府
Graduate School of Computer Science &
Systems Engineering, Kyushu Institute of
Technology
yamasaki@theory.ces.kyutech.ac.jp

宮野 英次 (Eiji Miyano)

九州工業大学大学院情報工学研究院
Faculty of Computer Science & Systems
Engineering, Kyushu Institute of
Technology
miyano@ces.kyutech.ac.jp

1 はじめに

2次元平面上の2点 p, q を横方向の線分と縦方向の線分で連結しているとき、そのパスを直線パスと呼ぶ。また、2点間の直線パスの長さを横線分と縦線分の長さの合計とすると、パスの長さが最小となる直線パスのことをマンハッタンパスと呼ぶ。すなわち、マンハッタンパスの長さは2点間の L_1 距離に等しい。2次元平面上の n 点集合 T が与えられたとき、すべての2点の対についてマンハッタンパスが存在するネットワーク G をマンハッタンネットワークとする。ネットワーク G について、 G に含まれるすべての線分の長さの合計を G の長さということにし、 $L(G)$ と表す。最小マンハッタンネットワーク問題 (MMN 問題と略記) は、点集合 T が与えられたとき、 $L(G)$ が最小となる T のマンハッタンネットワーク G を求める問題である。

MMN 問題は、文献 [5] で、Gudmundsson, Levcopoulos, Narasimhan により提案された問題であるが、MMN 問題に対して多項式時間アルゴリズムが存在するかどうかについてはまだ分かっていない。一方、Gudmundsson らは、 $O(n^3)$ 時間で動作する4近似アルゴリズム、および、 $O(n \log n)$ 時間で動作する8近似アルゴリズムを示している [5]。文献 [8] では $O(n^3)$ 時

間で動作する2近似アルゴリズムが示されたが、証明が不完全であったという指摘がある [3]。Benkert, Wolff, Widmann は、MMN 問題を混合整数計画問題として定式化し、 $O(n \log n)$ 時間の3近似アルゴリズムを示している [1]。その後、Chepoi, Nouioua, Vaxès は、混合整数計画問題の線形緩和することにより、2近似アルゴリズムを設計している [3]。Guo, Sun, Zhu は、組合せ的な手法により、 $O(n^2)$ 時間で動作する2近似アルゴリズムを文献 [6] で提案し、さらに、文献 [7] で $O(n \log n)$ 時間へと高速化を行っている。文献 [9] では1.5近似アルゴリズムが提案されているが、証明が不完全であり、現在知られている MMN 問題の近似上限は2である。

本稿では、Guo らによって文献 [6] で提案された2近似アルゴリズム (以下ではGSZアルゴリズムと呼ぶ) と文献 [7] で提案された別の2近似アルゴリズム (以下ではF-GSZアルゴリズム) の近似下界について調べる。両アルゴリズムに対して、どのような入力ネットワークが都合が悪く、近似比が大きくなってしまっているのか示さず、近似解析が厳密なものであるのか分からない。実際に、後で述べるように、F-GSZアルゴリズムはGSZアルゴリズムの近似解析でまだ“余裕のある無駄な線分”を追加することにより近似比を大きくすること無く高速化を

実現している。本稿では、GSZ アルゴリズムに対して近似比が 1.5 になってしまう入力例を示す。また、GSZ アルゴリズムを用いた場合には 1.5 近似になるが、F-GSZ を用いた場合には近似比が 2 まで大きくなる例を示す。このことは GSZ アルゴリズムの近似解析の精度を高めることで、より小さい近似上界を求めることができる可能性が残っていることを示している。

2 準備

n 個の 2 次元平面上の点集合 T が与えられたとき、すべての点の対についてマンハッタンパスが存在するネットワークをマンハッタンネットワーク G と呼ぶ。以下では T に対する最小マンハッタンネットワーク (MMN) を G^* とし、GSZ アルゴリズムにより得られた出力を G と表すことにする。

2 次元平面上の点を $p = (p.x, p.y)$ と表し、2 点 p, q 間のマンハッタン距離を $dist(p, q) := |p.x - q.x| + |p.y - q.y|$ とする。また、2 点 p, q について、 $p.x = q.x$ である場合には 2 点の垂直線分を、 $p.y = q.y$ である場合には水平線分を $[p, q]$ で表す。 $R(p, q)$ により p, q を対角の点とする境界線分を含む長方形、 $B_V(p, q)$ により p, q をそれぞれ通る 2 本の垂直直線とそれら 2 本で挟まれた帯、 $B_H(p, q)$ により p, q をそれぞれ通る 2 本の水平直線とそれら 2 本で挟まれた帯を表すことにする。与えられた n 個の点集合 T に対して、 Γ は T に含まれる点を通るすべての垂直直線と水平直線の和集合とする。また、 H, W をそれぞれ T を内包する最小の長方形の縦の長さとする。点 p を原点とする第一象限を $Q_1(p) = \{q | p.x \leq q.x, p.y \leq q.y\}$ とする。同様に、反時計回りに $Q_2(p), Q_3(p), Q_4(p)$ とする。また、アルゴリズムの説明で重要な役割を果たす領域について次節以降で示す。

パレートエンベロープとブロック

入力点集合 T に対して、 $P(T) = \bigcap_{u \in T} \bigcup_{v \in T} R(u, v)$ をパレートエンベロープと呼ぶ (図 1 参照)。 $P(T)$ はいくつかの凸型の多角形に分けることができ、それぞれをブロックと呼ぶ。各ブロックは他のブロックと 1 点のみでしか交わらないことが分かる。その共有点を切断点と呼ぶ。切断点の集合を C で表し、 $T^+ = T \cup C$ とする。ブロック B について、 H_B および W_B をそれぞれ B の縦幅および横幅を表すものとし、 $T_B := T^+ \cap B$ とする。

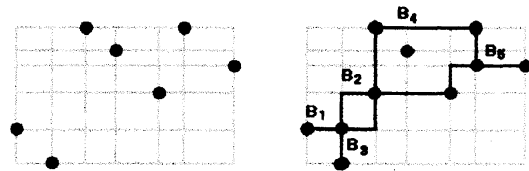


図 1: 点集合 T とパレートエンベロープ

もしブロック B が $|T_B| = 2$ であるような長方形であった場合、 B を自明ブロックと呼ぶ。 B が自明ブロックのとき、 T_B の 2 点は B の対角の 2 点になる。 T^+ の MMN は T の MMN になり、 T^+ の MMN を求めるためには、それぞれのブロック $B \subseteq P(T)$ について T_B の MMN を求めれば十分であることが知られている [3]。以下では、 B のアルゴリズムの出力を G_B とする。 B の周を ∂B とし、 $\Gamma_B = \Gamma \cap B$ とする。自明ブロックのマンハッタンパスは簡単に求めることができるため、自明でないブロックの MMN を求めることが目的となる。

ストリップ

$p.y < q.y$ であるような 2 つの点 $p, q \in T_B$ について、垂直直線 $\{(x, y) | x = p.x, y \leq p.y\}$ および $\{(x, y) | x = q.x, y \leq q.y\}$ を除く $B_V(p, q)$ の範囲内に T_B の点を含まないとき、 $R(p, q)$ を垂

直ストリップと呼ぶ。同様に, $p.x < q.x$ である 2 点 $p, q \in T_B$ について, 水平直線 $\{(x, y) | x \leq p.x, y = p.y\}, \{(x, y) | x \geq q.x, y = q.y\}$ を除く $B_V(p, q)$ の範囲内に他の点を含まないとき, $R(p, q)$ を水平ストリップと呼ぶ。

階段領域

近似アルゴリズムでは以下で定義する階段成分が重要な役割を果たす。階段成分は 4 つの象限により定義される 4 種類のものがある。ここでは, 一般性を失うことなく, 第一象限に相当する階段成分についてのみ述べる。 $q \in Q_1(p)$, $q' \in Q_1(p')$, $p, q \in B_V(p', q')$, $p', q' \in B_H(p, q)$ であるような 2 点 p, q について, $R(p, q)$ を垂直ストリップ, $R(p', q')$ を水平ストリップとする (図 2 参照)。このとき, $R(p, q) \cap R(p', q')$ での右上の点を o とする。 $(Q_3(v) \setminus Q_3(o)) \cap T_B = \{v\}$ であるような $T_B \cap Q_1(o)$ の点 v の集合を $T_{pp'|qq'}$ とする。もし $T_{pp'|qq'} \neq \emptyset$ であるとき, $S_{pp'|qq'} = \bigcup_{v \in T_{pp'|qq'}} R(o, v)$ とし, これを階段領域と呼ぶ。

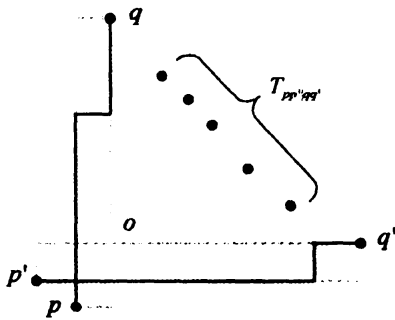


図 2: 階段領域

3 GSZ アルゴリズムと 1.5 近似下界

本節では文献 [6] で提案された 2 近似アルゴリズム (GSZ アルゴリズム) の動作概要と近似

上界の証明概要を示す。

任意の水平直線 ℓ と, ℓ が交差する任意の垂直ストリップ R に対し, 常に $\ell \cap C_V \cap R \neq \emptyset$ が成り立つならば, 垂直線分の和集合 C_V を垂直カバーと呼ぶ。水平カバーも同様に定義される。さらに, 任意の辺が少なくとも T_B の点を 1 つ含むような垂直カバーを良い垂直カバー (NVC) とする。良い水平カバー (NHV) も同様に定義される。最小垂直カバー (MVC) を最小の垂直カバーとし, 最小水平カバー (MHC) を最小の水平カバーとする。さらに, MVC(MHC) の任意の辺が少なくとも T_B の点を 1 つ含むとき, それを良い MVC(MHC) とする。良い MVC と良い MHC の和集合をナイスカバー (NC) とし, E_C で表す。

事実 1 n 点集合 T に対して, $E_C \subseteq \Gamma_B$ であるようなナイスカバー E_C を $O(n \log n)$ 時間で出力するアルゴリズム **CreateNC** が存在する [6]。

領域がつぶれていないストリップ $R(p, q)$ に対して, $R(p, q)$ を含む E_C の線分を $[p, a]$, $[q, b]$ とする。このとき, $[p, a] \cup [q, b] \cup [a, b]$ はマンハッタンパスである。この $[a, b]$ のことをスイッチ線分と呼ぶことにし, スイッチ線分の和集合を E_S とする。長さについて, $L(E_S) \leq H_B + W_B$ が成り立つ。

$M_{p,q}$ を p, q のマンハッタンパスとする。以下の改良階段領域を考えることができる。

定義 1 (改良階段領域) 階段領域 $S_{pp'|qq'}$ について, M_{pq} と $M_{p'q'}$ の交点を o' とする。このとき, M_{pq} と $M_{p'q'}$ を除いて M_{pq} と $M_{p'q'}$ で囲まれた $S_{pp'|qq'}^* = \bigcup_{v \in T_{pp'|qq'}} R(o', v) \setminus \partial B$ の部分のことを改良階段領域と呼び, $S_{pp'|qq'}^*$ で表す。

階段領域 $S_{pp'|qq'}$ が与えられたときに, 全ての $S_{pp'|qq'}$ の組に対してどのようにマンハッタン

GSZ アルゴリズム (入力 T)

1. $\mathcal{P}(T)$ を求める
2. 自明ブロックのマンハッタンパスを得る
3. 非自明ブロックに対して以下を実行
 - (a) **CreateNC** を実行する
 - (b) すべての E_S を追加する
 - (c) すべての $S_{pp'|qq'}^*$ に対して **CompOptNet** を実行する
4. $G := E_C \cup E_S \cup \bigcup_{pp'|qq'} S_{pp'|qq'}^* E_{pp'|qq'}$ を出力

図 3: GSZ アルゴリズム

パスを構成するかを述べる. 例として, $q, q' \in Q_1(o)$ について, $R(p, q)$ を垂直, $R(p', q')$ を水平ストリップとして仮定する.

事実 2 改良階段領域に含まれる点数 $|T_{pp'|qq'}|$ を n とする. 改良階段領域 $S_{pp'|qq'}^*$ が与えられたとき, $E \cup \partial B$ と $T_{pp'|qq'}$ のすべての点と M_{pq} または $M_{p'q'}$ と連結するようなネットワーク E を $O(n^2)$ 間で構成するアルゴリズム **CompOptNet** が存在する [6].

GSZ アルゴリズムは図 3 で示される.

事実 3 n 点集合 T に対して, GSZ アルゴリズムは $O(n^2)$ 時間で T のマンハッタンネットワーク G を求めることができる [6].

定理 1 GSZ アルゴリズムに対して, 近似比が 1.5 となる入力例が存在する.

証明. GSZ アルゴリズムに対して, 最適解の 1.5 倍になる入力例を具体的に示す. 図 4 のような入力点集合 T が与えられたとき, まずアルゴリズムはパレートエンベロープを求めることで,

入力点をいくつかのブロックに分ける. このとき, T は自明ブロック B 一つに分けられる. 次に, B に対し **CreateNC** により, ナイスカバーな路を加える. さらに, スイッチ線分を加える (図 4, 5 参照). B 内の改良階段領域を探し (図 5), 全ての改良階段領域に対して **CompOptNet** で最適なマンハッタンネットワーク E を構成する (図 6 左).

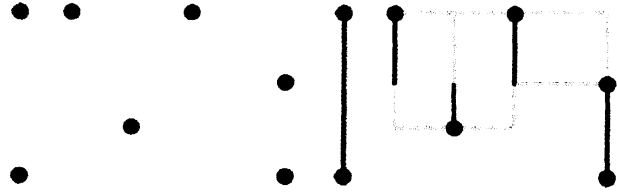


図 4: 入力点集合と垂直ストリップ

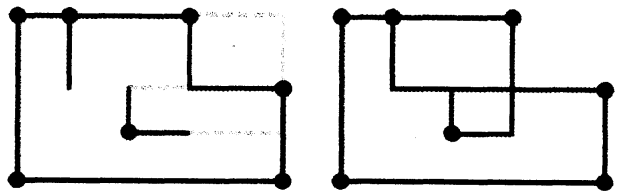
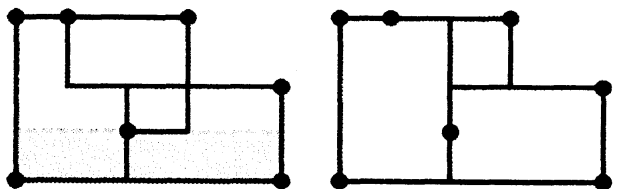


図 5: 水平ストリップとスイッチ線分

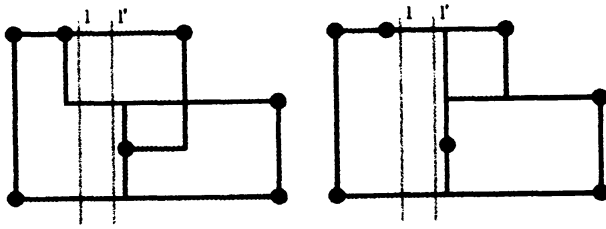
図 6: E と E^*

T に対する最適解 E^* は図 6 右のようになる. E, E^* に対し, 図 7 のような垂直な補助線 ℓ, ℓ' を考える. ℓ, ℓ' 間を B_ℓ とし幅を W_ℓ とする. $L(E \cap B_\ell) = \varepsilon, L(E^* \cap B_\ell) = \varepsilon^*$ とする.

$$L(E^*) = 2W_\ell + \varepsilon^*$$

$$L(E) = 3W_\ell + \varepsilon$$

$$\geq 3W_\ell + \varepsilon^*$$

図 7: 補助線を足した, E と E^*

$$= \frac{3}{2}L(E^*) - \frac{1}{2}\varepsilon^* \quad (1)$$

$W_\ell \gg \varepsilon^*$ とすると, E, E^* の形は W_ℓ の大きさに依存しないため, このときも式 (1) は成り立ち, すなわち, 近似比が 1.5 となることを示すことができる.

$$L(E) \geq \frac{3}{2}L(E^*)$$

□

4 F-GSZ アルゴリズムと 2 近似下界

文献 [7] で示された 2 近似アルゴリズムを F-GSZ アルゴリズムと呼ぶことにする. 前節で示した GSZ アルゴリズムも同じく 2 近似アルゴリズムであるが, 解析において厳密でない部分が多くある. そのため, 前節では 1.5 の近似下界しか示すことができなかった. F-GSZ アルゴリズムは, 近似比 2 を示す際にまだ”余裕のある”線分を GSZ アルゴリズムの出力するネットワークに追加をすることで処理の単純化を行い, 高速化を実現している. 本節では, F-GSZ アルゴリズムの概要を示し, F-GSZ アルゴリズムに対する 2 近似下界を示す.

良い垂直カバール C_V に対して, もし $R(p, q)$ がつぶれてしまっていたら, 線分 $[p, q] \subseteq C_V$ となる. もし $R(p, q)$ がつぶれていなかった場合に, $p.y < q.y$ であれば, $u.x = p.x$, $v.x = q.x$,

$q'.y \leq u.y = v.y \leq p'.y$ であるようなスイッチ線分 $[u, v]$ が必要になる.

GSZ アルゴリズムでは, **CreateNC** により最小の垂直と水平カバールからなるナイスカバール NC を求めたが, F-GSZ では, 最小である必要の無い良い垂直カバールと水平カバールをまず求めるように条件を緩めている. これにより **CreateNC** では $O(n \log n)$ 時間必要であったものを $O(n)$ 時間へと高速化できている (細部については文献 [7] を参照).

事実 4 n 点集合 T に対して, 良い垂直カバール C_V を $O(n)$ 時間で出力するアルゴリズム **CreateNVC** が存在する [7].

CreateNVC の実行後, 後で示すように, つぶれていない垂直ストリップに対して最上と最下の 2 本のスイッチ線分を加える. 水平方向については **CreateNVC** とまったく同様に **CreateNHC** を考えて, 良い水平カバール C_H を構成する. この時点ですべてのストリップ対に対してマンハッタンパスが存在することが保証できる. 残りは改良階段領域にネットワークを構成すれば良い.

事実 5 階段領域に含まれる点数 $|T_{pp'|qq'}|$ を n とする. 階段領域 $S_{pp'|qq'}$ が与えられたとき, $G_{pp'|qq'} \cup C_H \cup C_V$ が $T_{pp'|qq'}$ のすべての点と M_{pq} または $M_{p'q'}$ と連結するようなネットワーク $G_{pp'|qq'} \subseteq S_{pp'|qq'}$ を $O(n \log n)$ 時間で構成するアルゴリズム **CreateStaircasePas** が存在する [7].

$O(n)$ 時間で動作するアルゴリズム **CreateNVC** および **CreateNHC** と **CreateStaircasePath** をサブルーチンとする F-GSZ アルゴリズム全体を図 8 に示す.

事実 6 n 点集合 T に対して, F-GSZ アルゴリズムは $O(n \log n)$ 時間で T のマンハッタンネットワーク G を求めることができる [7].

F-GSZ アルゴリズム (入力 T)

1. $\mathcal{P}(T)$ を求める
2. 自明ブロックのマンハッタンパスを得る
3. 非自明ブロックに対して以下を実行
 - (a) **CreateNVC** を実行
 - (b) すべての垂直ストリップ $R(p, q)$ について, $R(p, q)$ の最上と最下のスイッチ線分を追加
 - (c) **CreateNHC** を実行
 - (d) すべての水平ストリップ $R(p, q)$ について, $R(p, q)$ の最左と最右のスイッチ線分を追加
 - (e) すべての $S_{pp'|qq'}^*$ に対して **CreateStaircasePath**($S_{pp'|qq'}$) を実行

図 8: F-GSZ アルゴリズム

定理 2 F -GSZ アルゴリズムに対して, 近似比が 2 となる入力例が存在する.

証明. ある T に対して, F -GSZ アルゴリズムが最適解の 2 倍のコストの出力をする例を示すことで, F -GSZ アルゴリズムの近似下限 2 を示す. その例として次のような出力 T を与える. 水平に 2 列点が並んでおり, 下部の点を左から $q_a (a = 0, 1, 2, \dots, n+1)$, 上部の点を左から $q_b (b = 1, 2, 3, \dots, n+1)$ とする. また, $p_0.x = q_0.x, p_{n+1}.x = q_{n+1}.x$ とし, $p_1.x > q_1.x, p_2.x < q_2.x, \dots$ と p_a, q_a の水平成分が交互に大きくなるように並んでいる. このとき, T は自明でないブロック B 一つでできている. また, $R(p_a, q_a)$ が垂直ストリップとなっている (図 9). T に対する F -GSZ の出力を E , 最適解を E^* , $p'_a = (p_a.x, q_a.y), q'_a = (q_a.x, p_a.y)$ とすると, **CreateNVC** は, T に対し $[p_a, p'_a], [q_a, q'_a]$

をパスとして出力する (図 10). E および E^* はそれぞれ図 11 の左および右になり, 次の式が成り立つ.

$$L(E) = (n+2)H + 2W$$

$$L(E^*) = \frac{1}{2}(n+4)H + 2W$$

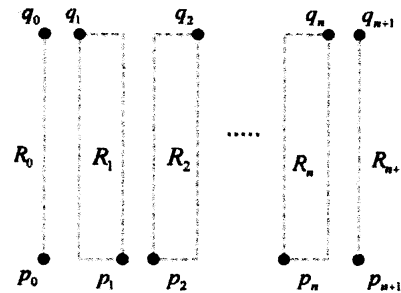
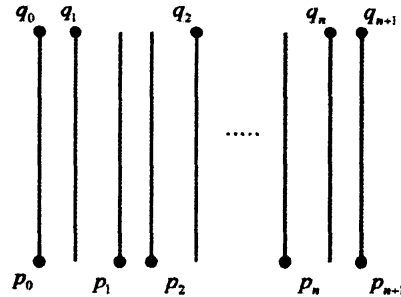
以上の式から以下が成り立つ.

$$\frac{L(E)}{L(E^*)} = \frac{2(n+2)H + 4W}{(n+4)H + 4W}$$

n が十分に大きい場合には十分小さい定数 ε について以下の式が成り立つ.

$$\frac{L(E)}{L(E^*)} \geq 2 - \varepsilon$$

□

図 9: 入力 T 図 10: T に対する C_V

参考文献

- [1] M. Benkert, A. Wolff, and F. Widmann. The minimum Manhattan network problem: A fast factor-3 approximation. In

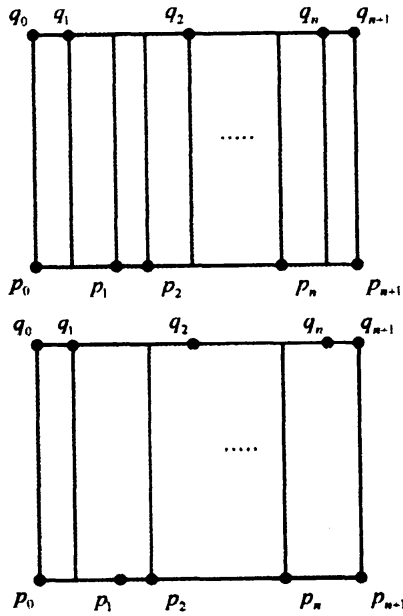


図 11: E と E^*

- Proceedings of the 8th Japanese Conference on Discrete and Computational Geometry*, pp.16–28, 2004.
- [2] M. Benkert, T. Shirabe, A. Wolff. The minimum Manhattan network problem: Approximation and exact solution. In *Proceedings of the 20th European Workshop on Computational Geometry*, pp.209–212, 2004.
- [3] V. Chepoi, K. Nouioua, and Y. Vaxés. A rounding algorithm for approximating minimum Manhattan networks. *Theoretical Computer Science*, 390, pp.56–69, 2008.
- [4] B. Fuchs and A. Schulze. A simple 3-approximation of minimum Manhattan networks. In *Proceedings of the 7th Cologne Twente Workshop on Graphs and Combinatorial Optimization*, pp.26–29, 2008.
- [5] J. Gudmundsson, C. Levcopoulos, and G. Narasimhan. Approximating a minimum Manhattan network. *Nordic Journal of Computing*, 8, pp.216–229, 2001.
- [6] Z. Guo, H. Sun, and H. Zhu. A fast 2-approximation algorithm for the minimum Manhattan network problem. In *Proceedings of the 4th International Conference on Algorithmic Aspect in Information Management*, pp.212–223, 2008.
- [7] Z. Guo, H. Sun, and H. Zhu. Greedy construction of 2-approximation minimum Manhattan network. In *Proceedings of the 19th International Symposium on Algorithm and Computation*, pp.4–15, 2008.
- [8] R. Kato, K. Imai, and T. Asano. An improved algorithm for the minimum Manhattan network problem. In *Proceedings of the 13th International Symposium on Algorithms and Computation*, pp.344–356, 2002.
- [9] S. Seibert, and W. Unger. A 1.5-approximation of the minimum Manhattan network problem. In *Proceedings of the 16th International Symposium on Algorithms and Computation*, pp.246–255, 2005.