G-104

整数計画法を用いたデフォルメ路線図改良法

A method for improving a deformed railway map by solving integer programming problems

一ノ瀬 拓海 † 多鹿 雄策 † 増田 澄男 † 斎藤 寿樹 † 山口 一章 †
Takumi Ichinose Yusaku Tashika Sumio Masuda Toshiki Saitoh Kazuaki Yamaguchi

1. まえがき

鉄道の路線図の構造をとらえやすくするために，単純化したデフォルメ路線図がよく用いられており，これらを自動生成するためのアルゴリズムについて研究が行われている [1]~[3]．デフォルメ路線図では，駅同士の相対的な位置関係をある程度保持しつつ，路線を垂直・水平線分からなる折れ線で描いているものがある．各路線をたどることが容易であるようにするためには，路線を示す折れ線の形状は単純であることが望ましい．

デフォルメ路線図は，駅を頂点，駅同士を結ぶ路線を辺とみなすことでグラフ描画としてとらえることが可能である．辺を垂直・水平線分を用いて描いたグラフ描画を直交描画と呼ぶ（各頂点の次数は 4 以下に限定される）．橘ら [3] は，グラフの直交描画アルゴリズムを応用することにより，デフォルメ路線図を自動で作成するアルゴリズムを提案している．この方法は，グラフ描画 $\tilde{G}$ が与えられたときに，頂点の相対的位置関係を概ね保持しながら，$\tilde{G}$ を直交格子上に埋め込み，その後，辺の折れ曲がりや描画面積の削減のための処理を行うものである．

デフォルメ路線図では，各路線に対応する折れ線の折れ曲がりは少ない方が望ましいと考えられるが，橘らの手法による路線図では，折れ曲がりを削減できる余地が残されている場合がある．また，一般にグラフ描画では辺長の総和や辺長分散が不必要に大きくならないことが望まれる [4] が，この点に関しても改善の余地が残されている．

本研究では，垂直・水平線分からなるデフォルメ路線図を改良する手法について検討する．そして，橘らの方法による 2 種類のデフォルメ路線図を用いた計算機実験により，提案手法が上記の問題点の緩和に有効であることを示す．

提案手法は，与えられたデフォルメ路線図から，各路線に対応する折れ線を取り出し，それらの折れ線を平面上に再配置する問題（折れ線配置問題と呼ぶ）を解くものである．折れ線を配置する際に，折れ曲がりが減るような変形を行うこと，及び折れ線の長さの総和を最小にすることにより，デフォルメ路線図の改良を行っている．本研究では，折れ線配置問題を整数計画問題に帰着して解いている．

以下，まず **2.** において折れ線配置問題の説明を行う．**3.** では，デフォルメ路線図から折れ線配置問題の入力を作成する方法と，整数計画問題における制約条件を生成する方法について述べる．**4.** では，橘らの方法によるデフォルメ路線図を用いた計算機実験の結果を示す．最後に **5.** において，本稿の結果をまとめ，今後の課題について述べる．

2. 折れ線配置問題

本章では，折れ線配置問題について説明する．この問題では，折れ線の集合 $S = \{l_1, l_2, ..., l_n\}$ が与えられる．各折れ線 l_i は k_i 個の点からなる系列 $[p_i^1, p_i^2, ..., p_i^{k_i}]$ で表現される．異なる折れ線 l_i, l_j が点を共有しているとき，その点を合流点と呼ぶ．

本研究では，各折れ線 l_i と各整数 $j = 1, 2, ..., k_i - 1$ について，$p_i^j p_i^{j+1}$ は水平線分もしくは垂直線分であるものとする．p_i^j と p_i^{j+1} に対して，相対的な位置関係（例えば，p_i^{j+1} は p_i^j の右にあるなど）が指定される．また，p_i^j, p_i^{j+1} 間のみでなく，連続していない 2 点 p_i^j, p_i^k $(j < k)$ の位置関係，あるいは異なる折れ線上の点 p_i^j, p_l^k の位置関係が指定される場合もあるものとする．

点の位置関係に関する制約以外に，線分の長さに関しても制約を設ける．各線分 $p_i^j p_i^{j+1}$ について，長さの下限 ll_i^j が指定される．$p_i^j p_i^{j+1}$ の長さ $\overline{p_i^j p_i^{j+1}}$ は ll_i^j 以上でなければならないものとする．同一折れ線上の連続する線分 $p_i^j p_i^{j+l}, p_i^{j+1} p_i^{j+2}, \ldots, p_i^{j+l-1} p_i^{j+l}$ $(l \leq k_i - j)$ に対して，それらの線分長の合計の下限が指定される場合もあるものとする．

以上に述べた制約をすべて満たすようにして，S の折れ線のすべての線分の端点及び合流点の座標を決定する問題を折れ線配置問題と呼び，決定した配置を解と呼ぶ．折れ線配置問題に対する解のうち，すべての線分の端点及び合流点の座標が整数であるものを整数解と呼ぶ．また，異なる点（線分の端点あるいは合流点）が異なる座標をもち，合流点以外で異なる線分同士が重ならない整数解を正当解と呼ぶ．一般に，折れ線配置問題の中には解をもたないものや，解が複数あるものが存在する．

図 1 に折れ線配置問題の入力とその正当解の例を示す．入力は図 1(a) であり，3 本の折れ線からなる．図中に示

† 神戸大学，Kobe University

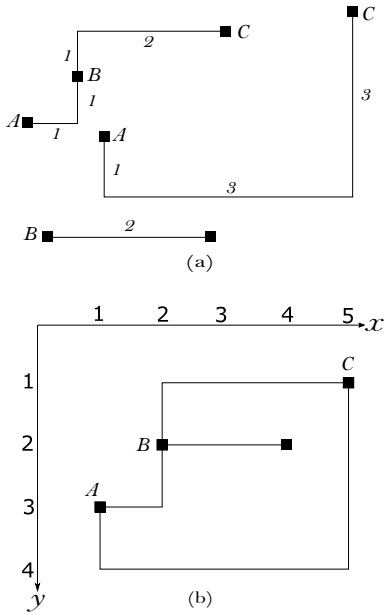


図 1 折れ線配置問題の入力とその正当解の例

した折れ線上の点 $A \sim C$ は合流点であり，同じ文字で表した合流点には同じ座標をもたせる．また各線分の近傍に示した数 $1 \sim 3$ はその線分の長さの下限を表している．図 1(b) は解の一例であり，これは正当解である．なお本稿では，y 軸の向きを図 1(b) に示しているように定めている．

本研究では，各路線に対応する折れ線が短く簡潔となるように，折れ線の長さの総和が最小となる解を選ぶこととする．折れ線配置問題の整数解のうち，折れ線の長さの総和を最小とするものを求める問題は，次章で述べるように，整数計画問題に帰着することが可能である．

3. デフォルメ路線図の改良

本章では，直交描画のデフォルメ路線図が与えられたときに，その改良を行うため，折れ線配置問題の入力と整数計画問題の目的関数及び制約条件を生成する方法について述べる．橘ら [3] のデフォルメ路線図作成法は，基本的に頂点と辺を処理単位としたものであるが，本稿で提案するデフォルメ路線図改良法は，各路線に対応する折れ線を直接扱うものである．

提案手法は，以下の 4 段階の処理を実行する．

第 1 段階： 与えられたデフォルメ路線図の各路線に対し，線分の端点と合流点以外の点を間引くことによって，1 本の折れ線を生成する．

第 2 段階： 折れ線配置問題を整数計画問題に帰着するときの目的関数と，点の位置関係及び線分長に関する制約条件を生成する．その整数計画問題を解くことで，各折れ線上の各点（線分の端点及び合流点）の座標を決定する．

第 3 段階： 第 2 段階で得られた点の配置が，折れ線配置問題の正当解を構成しているか否かを判定し，正当解になっていない場合には問題点を回避するための適切な制約条件を追加して，再び整数計画問題を解く．この処理は，正当解が得られるまで繰返し実行する．

第 4 段階： 第 3 段階までの処理で配置した折れ線に対して，第 1 段階で間引いた点を，それが存在していた各線分上に均等な間隔で再配置する．

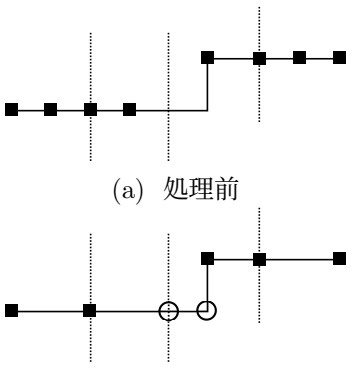
以下，各段階の詳細について述べる．

3.1 第 1 段階

与えられたデフォルメ路線図の路線に対応する各折れ線に対して，その上の次数 2 の点（駅に対応するもの）で線分の端点でないものを一旦削除する．そして，第 2, 3 段階で線分の端点及び合流点の座標を決定した後，第 4 段階において，削除していた点の再配置を行う．このようにすることによって，第 2，3 段階で解く整数計画問題の変数を減らすことができる．また，第 4 段階において，辺長のバランスを考慮しながら，間引いていた点の座標を決めることができる．

第 1 段階終了時点で残っている点のうち，駅に対応するものを実頂点と呼ぶことにする．合流点はすべて実頂点である．線分の端点のうち実頂点でないものは，折れ線の折れ曲がり点である．そのような点には仮の頂点を置く．また，異なる路線の交点で駅が存在しない点にも仮の頂点を置く．これ以降，実頂点と仮の頂点を併せて頂点と呼ぶ．路線の交点に仮の頂点を置くことによって，最終的なデフォルメ路線図でも，対応する 2 本の路線が交差している状態を保持することができる．

例として，図 2(a) に実線で示した折れ線について考える．点線はこの折れ線と交わる他の折れ線を示している．第 1 段階の処理をこの折れ線に対して行うと図 2(b) のようになる．3 つの点が間引かれており，丸で示したような 2 つの仮の頂点が作られている．



(a) 処理前

(b) 処理後

図 2 第 1 段階の処理の例

3.2 第 2 段階

第 1 段階で作成した折れ線の集合を $S = \{l_1, l_2, ..., l_n\}$ とし，各折れ線 l_i を点の系列 $[p_i^1, p_i^2, ..., p_i^{k_i}]$ で表現するものとする．提案手法の第 2，3 段階で解く整数計画問題の変数は，各頂点 p_i^j の x 座標と y 座標であり，それぞれ x_i^j, y_i^j と表す．

各折れ線 l_i 上の各水平（垂直）線分 $p_i^j p_i^{j+1}$ について，$x_i^j < x_i^{j+1}$ $(y_i^j < y_i^{j+1})$ であるならば

$$\overline{p_i^j p_i^{j+1}} = x_i^{j+1} - x_i^j \ \ (y_i^{j+1} - y_i^j) \tag{1}$$

であり，$x_i^j > x_i^{j+1}$ $(y_i^j > y_i^{j+1})$ であるならば

$$\overline{p_i^j p_i^{j+1}} = x_i^j - x_i^{j+1} \ \ (y_i^j - y_i^{j+1}) \tag{2}$$

である．このとき折れ線 l_i の長さは

$$\overline{l_i} = \sum_{j=1}^{k_i - 1} \overline{p_i^j p_i^{j+1}} \tag{3}$$

と表すことができる．折れ線の長さの総和を最小とするため，整数計画問題の目的関数は次のように定める．

$$\text{minimize} \sum_{i=1}^{n} \overline{l_i}. \tag{4}$$

第 1 段階終了時に存在するすべての 2 頂点について，それらの位置関係に関する制約条件を作ることも可能である．しかし，制約条件を増やしすぎると，折れ線配置の際の自由度が小さくなってしまい，よい解を見逃す可能性がある．そこで第 2 段階では，基本的に，以下を保証する制約条件のみを加える．

- 各折れ線上で連続する 2 頂点の位置関係と，それらの間の長さの下限
- 合流点が，それが存在する折れ線において同一の座標をもつこと

ただし，後述する規則 4 により，同一折れ線上の連続していない 2 頂点に対して制約条件を作ることもある．

第 2 段階で制約条件を作成する規則は以下の 5 つである．

規則 1: 各線分 $p_i^j p_i^{j+1}$ が水平であれば

$$y_i^j = y_i^{j+1} \tag{5}$$

とし，垂直であれば

$$x_i^j = x_i^{j+1} \tag{6}$$

とする．

規則 2: 各線分 $p_i^j p_i^{j+1}$ の長さの下限 ll_i^j を，p_i^j, p_i^{j+1} 間で間引いた点の個数に 1 加えた値とする．例えば 2 点 p_i^j, p_i^{j+1} の間で 4 つの点を間引いていた場合には，ll_i^j は 5 とする．

規則 3: 各線分 $p_i^j p_i^{j+1}$ に対し，p_i^j の x 座標が p_i^{j+1} の x 座標より小さければ

$$x_i^{j+1} - x_i^j \geq ll_i^j \tag{7}$$

とする．また，p_i^j の y 座標が p_i^{j+1} の y 座標より小さければ

$$y_i^{j+1} - y_i^j \geq ll_i^j \tag{8}$$

とする．座標の大小が逆の場合，不等号の向きを逆にする．

規則 4: 隣接する 2 頂点 p_i^j, p_i^{j+1} がどちらも折れ曲がり点であり，第 1 段階において p_i^j と p_i^{j+1} の間にあった点が間引かれておらず，かつ p_i^j と p_i^{j+1} の少なくとも一方が仮の頂点である場合に本規則を適用する．規則 2 では，p_i^j, p_i^{j+1} とそれぞれに隣接する p_i^{j-1}, p_i^{j+2} についての長さの下限 $ll_i^{j-1}, ll_i^j, ll_i^{j+1}$ の値を個別に定めていたが，本規則では，$ll_i^{j-1} + ll_i^{j+1}$ を計算し，これを p_i^{j-1} と p_i^{j+2} の間の距離の下限とする．さらに，ll_i^j の値を 0 に変更する．本規則の目的は路線の折れ曲がりを削除することである．

規則 2, 4 の適用例を図 3 に示す．三角形で示した B が実頂点である場合には，規則 2 により

$$x_B - x_A \geq 3, \ y_A = y_B,$$
$$x_B = x_C, \ y_C - y_B \geq 1,$$
$$x_D - x_C \geq 3, \ y_C = y_D$$

という制約条件を作る．一方，B が仮の頂点である場合には，

$$x_B - x_A \geq 3, \ y_A = y_B,$$
$$x_B = x_C, \ y_C - y_B \geq 0,$$
$$x_D - x_C \geq 3, \ y_C = y_D$$

とした上で，

$$x_D - x_A \geq 6$$

という制約条件を追加する．こうすることで，図 3 右下のように折れ曲がりを削減できる可能性が生まれることになる．

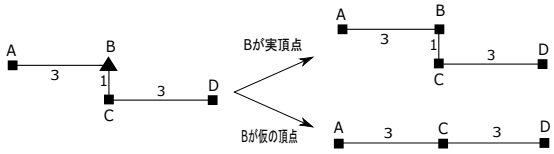


図 3 規則 2, 4 の適用例

規則 5: 異なる折れ線上の 2 頂点 p_i^j, p_k^l が合流点であるとき，

$$x_i^j = x_k^l,\ y_i^j = y_k^l \tag{9}$$

という制約条件を加える．

提案手法の第 2 段階では，これまでに述べた整数計画問題の最適解を求める．本研究においては，入力が直交描画のデフォルメ路線図であり，第 2 段階までで考えている折れ線配置問題の正当解の 1 つに対応するものであることから，この整数計画問題が解をもつことは容易に分かる．

3.3 第 3 段階

第 2 段階で得られる各頂点の座標を用いてデフォルメ路線図を作成すると，異なる頂点同士が重なる，異なる折れ線上の線分が重なるなどの問題点がでてきて，折れ線配置問題の正当解が得られないことがある．そこで，提案手法の第 3 段階では，そのような問題点を解決するための制約条件を加えて，再度，整数計画問題を解く．このとき加える制約条件は以下の手順で作成する．

1. 重なっている頂点（あるいは線分）について，入力のデフォルメ路線図における座標を参照する．
2. 参照した元の座標の大小関係を保存するような制約条件を作成する．

第 3 段階では，以上の処理を，折れ線配置問題の正当解が得られるまで繰返し実行する．

3.4 第 4 段階

第 1 段階において間引いた各点を，それが存在していた線分上に再配置する．例えば水平線分 $p_i^j p_i^{j+1}$ 間で 3 つの頂点を間引いていた場合には，$x_i^j < x_i^{j+1}$ であるとすると，距離 $(x_i^{j+1} - x_i^j)/(3+1)$ ごとに $p_i^j p_i^{j+1}$ 上に 3 つの点を配置する．このとき配置した点の座標は必ずしも整数値にはならない．点を均等に再配置することで，**1.** で述べた辺長分散の問題がある程度解消されると考えられる．

4. 計算機実験

橘ら [3] の手法により求めた大阪市営地下鉄と名古屋市営地下鉄のデフォルメ路線図に対して提案手法を適用した．本章では，この計算機実験の結果について述べる．なお，本研究では，整数計画問題を解くためにソルバー SCIP[5] を用いた．

4.1 大阪市営地下鉄

入力としたデフォルメ路線図を図 4 に示す．これに第 1 段階の処理を実行した結果が図 5 である．第 2 段階で，**3.2** で述べた方法により整数計画問題の目的関数と制約条件を作成し，その問題を解いたところ，図 6 の結果が得られた．図 4 と見比べると分かるように，図 6 中に A と示した部分で異なる線分の重なりが起こっており，折れ線配置問題の解として正当でない．

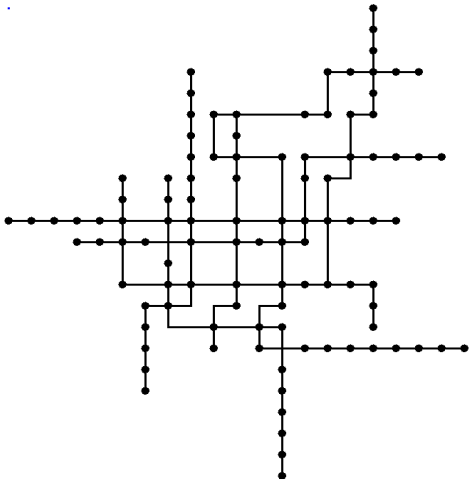


図 4 入力のデフォルメ路線図（大阪市営地下鉄）[3]

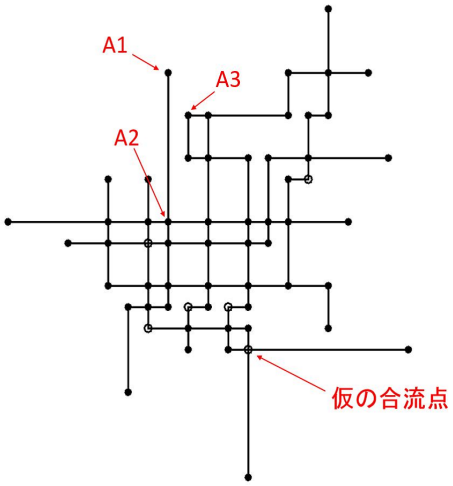


図 5 第 1 段階の結果（大阪市営地下鉄）

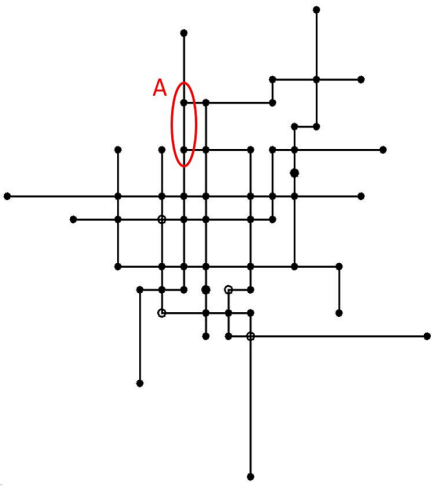


図 6 第 2 段階の結果（大阪市営地下鉄）

この問題点を解決するために，第 3 段階において新たな制約条件を追加した．図 5 で A1∼A3 としている 3 つの頂点の座標をそれぞれ $(x_{A1}, y_{A1}), (x_{A2}, y_{A2}), (x_{A3}, y_{A3})$ としたとき，加えた制約条件は以下の 2 つである．

$$x_{A1} < x_{A3},$$

$$x_{A2} < x_{A3}.$$

（規則 1 によって $x_{A1} = x_{A2}$ という制約条件を設けているので，実際には一方の制約条件を追加するだけで十分である．）

第 3 段階の結果を図 7 に示す．さらに，間引いていた頂点を再配置した結果のデフォルメ路線図（提案手法の出力）を図 8 に示す（図 5∼7 で示していた仮の頂点は，図 8 では削除している）．

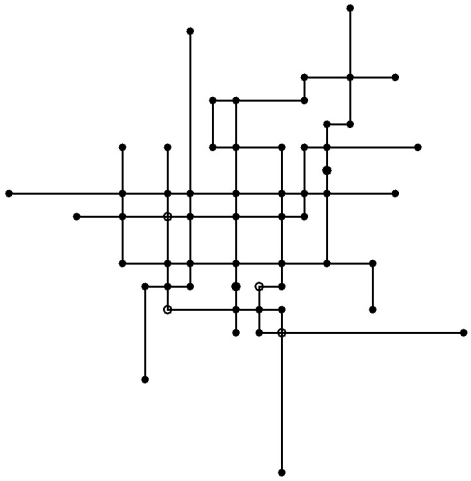


図 7　第 3 段階の結果（大阪市営地下鉄）

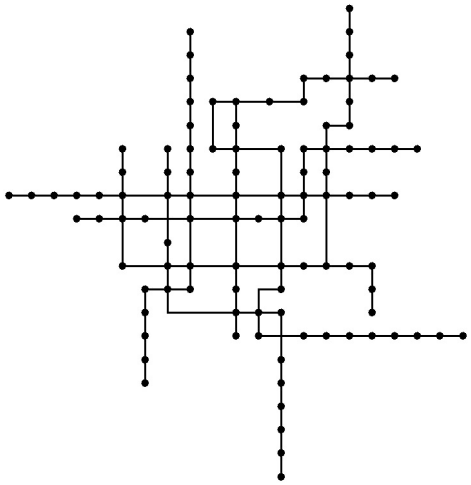


図 8　提案手法により得られた大阪市営地下鉄のデフォルメ路線図

提案手法では，第 2 段階と第 3 段階において整数計画問題を 1 回ずつ解いているが，いずれも非常に高速に解くことができた．

図 4 と図 8 の路線図を，最短の辺長が 1 で，x, y 座標の最小値が 0 のグラフ描画とみなし，以下の評価基準に関して比較を行った．

- 路線に対応する道の折れ曲がり数の総和
- 辺長の総和
- 辺長の分散
- 描画の幅：頂点の x 座標の最大値
- 描画の高さ：頂点の y 座標の最大値
- 描画面積：幅と高さの積

2 つの路線図の評価値を表 1 に示す．

表 1　大阪市営地下鉄の路線図の評価値

	折れ曲がり数	辺長総和	辺長分散	幅	高さ	描画面積
橘らの手法	21	153	0.276	20	22	440
提案手法	18	143	0.231	20	20	400

提案手法により得られた路線図では，橘らの路線図と比べて，路線の折れ曲がりが少なくなっている．また，折れ線長の総和最小を目的関数とした整数計画問題を解くことにより，辺長総和を小さくすることができており，一旦間引いた点を均等に再配置していることもあって，辺長分散の値も小さくすることができている．全部で 118 本ある辺のうち辺長 1 のものは，橘らの路線図では 87 本であったが，提案手法の路線図では 95 本に増えていた．2 つの路線図の幅は同じであったが，高さに関しては提案手法の方が減っており，若干ではあるがよりコンパクトな路線図が得られている．

4.2　名古屋市営地下鉄

名古屋市営地下鉄については，図 9 のデフォルメ路線図を入力とした．この路線図は，文献 [3] で用いているのとは違う閾値を使って，橘らの手法を実行して得られたものである．この入力に対して提案手法の第 1 段階を実行した結果を図 10 に示す．

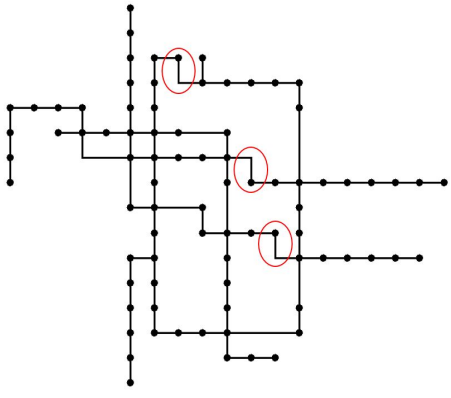


図 9　入力のデフォルメ路線図（名古屋市営地下鉄）

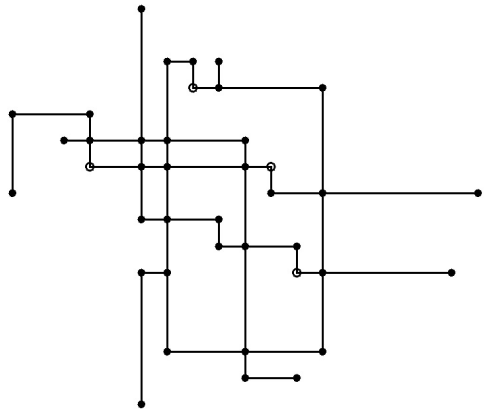


図 10　第 1 段階の結果（名古屋市営地下鉄）

さらに，提案手法の第 2 段階を実行した結果を図 11 に示す．図 9 中に赤丸で示した部分の折れ曲がりが，規則 4 の効果によって消えている．図 11 の段階で折れ線配置問題の正当解が得られているため，第 3 段階において再度整数計画問題を解く必要はなかった．

第 4 段階を実行して得られたデフォルメ路線図を図 12 に示す．

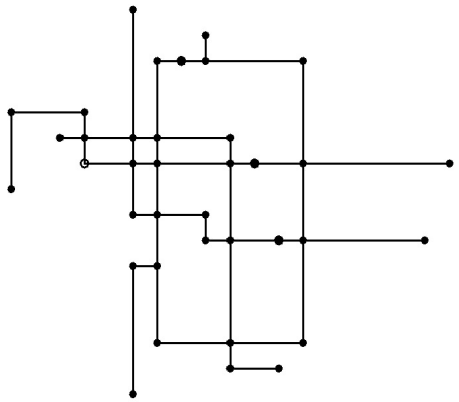


図 11　第 2 段階の結果（名古屋市営地下鉄）

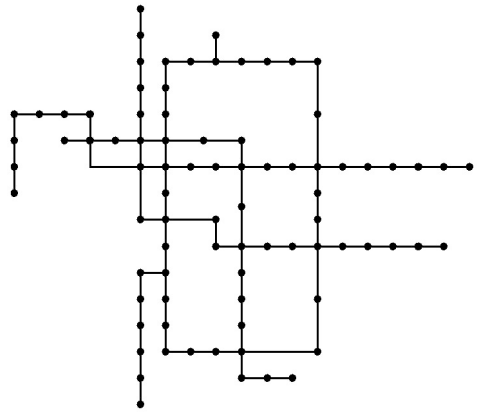


図 12　提案手法により得られた名古屋市営地下鉄のデフォルメ路線図

4.1 で述べた評価基準を用いて，図 9 と図 12 のデフォルメ路線図を比較した結果を表 2 に示す．

表 2　名古屋市営地下鉄の路線図の評価値

	折れ曲がり数	辺長総和	辺長分散	幅	高さ	描画面積
橘らの手法	19	105	0.196	18	15	270
提案手法	13	103	0.147	18	15	270

表 2 を見ると，提案手法による路線図では，橘らの路線図に比べて，路線の折れ曲がりを大きく減らすことができている．描画の幅，高さ及び描画面積は同じであったが，辺長総和と辺長分散の値は改善することができている．

5. あとがき

本稿では，直交描画であるデフォルメ路線図が与えられたときに，それを改良する方法について考察した．提案手法は，与えられた路線図から作成した折れ線の配置を，整数計画問題を解くことにより定めるものである．元の路線図に存在していた路線の折れ曲がりを削減することができるように，整数計画問題の制約条件の作成方法を工夫している．橘ら [3] の手法によるデフォルメ路線図を入力として計算機実験を行ったところ，提案手法により，辺長総和及び辺長分散を改善することができ，また路線の折れ曲がりを削減することができて，より分かりやすいデフォルメ路線図を得ることができた．

今後の課題としては

- デフォルメ路線図の改良に有効な新たな目的関数について検討すること，
- 図 12 の路線図には辺長分散をさらに改善できる余地が残されているため，整数計画問題の制約条件の作成方法を改良すること，
- 本研究の計算機実験で扱ったものよりさらに複雑な路線図を扱うことができるように，提案手法を，斜め 45° の線分を用いたデフォルメ路線図の作成・改良方法に拡張していくこと

などが考えられる．

参考文献

[1] 山守一徳, 海野祐史, 河合敦夫, 椎野　努, “デフォルメ路線図のインタラクティブ生成システムの開発,” 情報処理学会論文誌, vol.43, no.9, pp.2928-2938, 2002.

[2] 浅田信浩, 増田澄男, 山口一章, “グラフ描画アルゴリズムに基づいたデフォルメ路線図作成法,” 2005 年電子情報通信学会総合大会講演論文集, D-1-2, 2005.

[3] 橘　一行, 増田澄男, 山口一章, 鈴木智貴, “グラフ描画の直交格子への埋め込みアルゴリズムとデフォルメ路線図作成への応用,” 電子情報通信学会論文誌 (A), vol.J94-A, no.3, pp.180-191, 2011.

[4] 杉山公造, グラフ自動描画法とその応用 ― ビジュアルヒューマンインターフェース, 計測自動制御学会, 1993.

[5] 宮代隆平, “整数計画ソルバー入門,” オペレーションズ・リサーチ, vol.57, no.4, pp.183-189, 2012.