複雑な路線網に対する鉄道路線図生成手法

Automatic Drawing for Complex Metro Maps

情報工学専攻　恩田 雅大
Information and System Engineering　ONDA Masahiro

概 要

鉄道路線図とは，実際の路線網を基に簡略化した地図である．鉄道路線図を手作業で作る場合，非常に手間と時間がかかる．そのため，鉄道路線図の自動生成の研究が行われている．しかし，複雑な路線網に対して，既存の研究では路線図を自動生成することは難しい．本研究では，複雑な路線網に対して，3 つのステップからなる鉄道路線図生成手法を提案する．路線図を生成するときは，混合整数計画法を用いる．第 1 ステップでは，グラフの簡略化，ラベル領域を表すグラフの追加，乗り換え駅の次数下げを行う．第 2 ステップでは，路線網を部分グラフに分割し，その部分グラフを逐次添加しながら制約を満たす路線図を素早く生成する．第 3 ステップでは，第 2 ステップで生成した路線図に対して，局所探索を行うことで，最適ではないが局所最適となる路線図を生成する．これらを適用することで，複雑な路線網に対しても良い路線図を生成することができる．

キーワード： 鉄道路線図，略地図生成，ラベル配置問題，混合整数計画問題，局所探索法

1　序論

近年，鉄道路線図 (図 1) は世界中の大都市で鉄道を利用する人々にとって重要なツールとなっている．路線網が複雑になればなるほど，出発地から目的地に向かうまでの行き方が複雑になる．もしくは，複数の行き方があることがあり，より路線図が必要となる．鉄道の利用者は，出発する前になるべく乗り換えが少ない経路，もしくはなるべく早く着く経路を設計してから出発する．この時，路線図はどの駅で乗り換えればよいのか，どの路線のどの方向に乗ればよいのか，降りるまで何駅あるのかという情報を与える．このような情報を得るために，駅や路線の地理的位置に基づいた路線の長さ等の情報よりも，路線網の位相幾何学的性質の認識のし易さが重要になる．このような位相幾何学的な認識のし易い路線図を “良い” 路線図という．

“良い” 路線図は Harry Beck が定めた基準 [1] に従って描かれていることが多い．Harry Beck はイギリスの製図技師でロンドンの地下鉄の路線図を作成した．このロンドンの路線図はある一定のルールの下作られており，この基準は Beck の基準と呼ばれている．Beck の基準には，路線の方向が 8 方向に制限されている，駅間の間隔が一様である，路線の折れ曲がりが少ない等のルールがある．しかし，手作業で描くことは難しく，経験のある人でさえ時間が掛かる作業となっている．そのため，多くの路線図自動生成の研究で，Beck の基準を基に考えられた制約で路線図が生成されている．

本研究では，複雑な路線網に対して，路線図を生成する手法を提案する．近年，世界の大都市における路線網はより複雑になっている．Nöllenburg と Wolff [3] は，混合整数計画問題に定式化して解くことで路線図を生成している．しかし，[3] はロンドンや東京などの複雑な路線網では，路線図を生成することは難しく，また路線網を表すグラフの最大次数が 8 より大きいと路線図を生成できない．本研究では，[3] の手法を基に路線図を生成する 3 つのステップからなる手法を提案する．第 1 ステップでは，既存研究によるグラフの簡略化とラベル領域を表すグラフの追加，また最大次数が 8 より大きい駅が存在する場合でも路線図を生成するために乗り換え駅の次数下げを行う．第 2 ステップでは，制約を満たす路線図を素早く生成する初期レイアウト生成手法である．最後，第 3 ステップでは初期レイアウト生成手法で生成した路線図をより良くする反復改善法である．これらを適用することで，複雑な路線網に対しても良い路線図を生成することができる．

図 1. 鉄道路線図 [6]

2　関連研究

路線図の自動生成に関する研究はいくつかある．Hong ら [4] は，力学モデルによるグラフ描画を基にした 5 つの手法を提案している．それらの手法は素早く路線図を生成できる．

Stott ら [2] は，多基準最適化を用いた路線図生成手法を提案している．また，クラスタリングを行い局所最適解を抜け出し，より良い解を見つけることができる．

2.1　混合整数計画問題を用いた鉄道路線図生成手法

Nöllenburg と Wolff [3] は MIP を用いて路線図を生成する．まず，路線図を描画する上でのルールを 7 つ定義し，そのルールをを必ず満たす制約となるべく満たす制約に分ける．それらの制約を MIP として定式化し，解くことで路線図を生成する．

また，入力するグラフに対して，ラベル領域を表すグラフを追加する．そのグラフに対して MIP を解く

ことでラベル付き路線図を生成できる．

さらに，Nöllenburg と Wolff は計算時間を削減する 2 つの手法を提案している．1 つ目は，グラフサイズ削減手法である．路線網には，次数 2 の頂点が多くあるため，その次数 2 の頂点を省略しグラフのサイズを小さくする．2 つ目は，MIP サイズの削減である．定式化した MIP では，交差禁止を表す制約式の数が多くなる．そこで，初めは交差禁止制約を考慮せず MIP を解き，解いている途中で交差が発生したら交差禁止制約を追加する．このように，必要に応じて制約を追加しながら MIP を解くことで制約式の数を抑えながら解を求めることができる．

3 Metro Map Layout Problem

本研究で扱う鉄道路線図を描画する問題として，Nöllenburg と Wolff [3] によって定義された Metro Map Layout Problem について述べる．

$G = (V, E)$ を入力グラフである路線網とし，入力グラフが平面グラフでなく交差がある場合は，交差点に仮想ノードを追加し平面グラフとする．l を路線と呼び，L を路線集合とする．(G, L) をメトログラフと呼ぶ．メトログラフ (G, L) の略地図である路線図を Γ とする．路線図を生成する際の制約を必ず満たす制約となるべく満たす制約に以下のように分ける．

必ず満たす制約

(H1) 路線を表す線分の方向が 8 方向である．
(H2) 隣接頂点の循環的順序付けが G と同じである．
(H3) 路線を表す各線分 $e \in E$ の長さが l_e 以上である．
(H4) 端点を共有しない 2 つの線分が交差しない．

なるべく満たす制約

(S1) 路線の折れ曲がりが少ない．
(S2) 隣接頂点の位置関係が描画前後で変わらない．
(S3) 全ての線分の長さの合計が小さい．

これらの制約を用いて，Metro Map Layout Problem を以下のように定義する．

Metro Map Layout Problem

入力： 平面グラフ $G = (V, E)$，G の路線集合 L，各辺 $e \in E$ に対する最小辺長 $l_e > 0$，最小距離 $d_{\min} > 0$
出力： (H1)-(H4) を必ず満たし，(S1)-(S3) をなるべく満たしている (G, L) の略地図 Γ

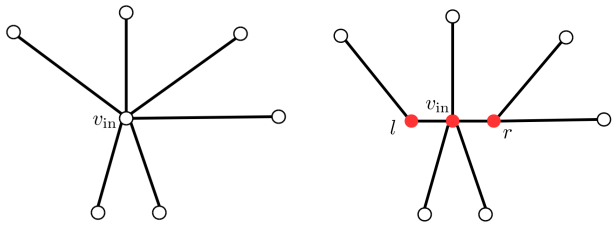
本研究では [3] の手法を基にしているため，Metro Map Layout Problem を MIP として定式化し，解くことで路線図を生成する．目的関数は，各なるべく満たす制約 $\mathrm{S}_i (i = 1, 2, 3)$ に対するコスト関数の総和とし，目的関数が最小となる解を出力する．各コスト関数には非負実数である重み $\lambda_{(\mathrm{S}_i)} (i = 1, 2, 3)$ が乗じられている．

4 提案手法

複雑な路線網に対して路線図を生成するための 3 つのステップについて説明する．

4.1 前処理

まず，入力グラフに対して，Nöllenburg と Wolff [3] によるグラフの簡略化とラベル領域を表すグラフの追加を行う．しかし，[3] では，路線網を表すグラフの最大次数が 8 より大きい場合解くことができないため，乗り換え駅の次数を下げる．本研究では，図 2(a) の頂点 v_{in} のような乗り換え駅に対して，図 2(b) のように頂点 l，r を左右の水平方向に追加し，辺の付け替えを行うことで次数を下げる．このグラフを $\widetilde{G}$ とする．その 3 つの頂点を乗り換え駅とみなす．また，その 3 つの頂点をラベル領域とし，頂点にラベルを埋め込む．



(a) 乗り換え駅の頂点 v_{in}．　(b) 乗り換え駅の左右に頂点 r，l を追加．

図 2. 乗り換え駅に対するラベル配置．

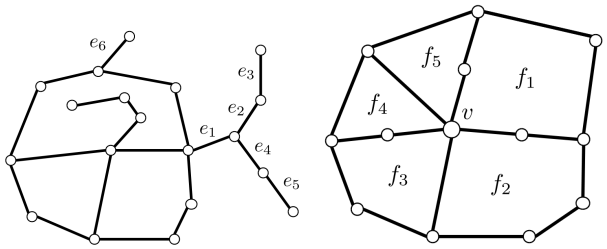
4.2 初期レイアウト生成手法

初期レイアウト生成手法では，制約を満たしてるが最適化されていない路線図を生成する．ここでは，グラフ $\widetilde{G}$ をいくつかの部分グラフに分割する．分割した部分グラフの路線図を生成し，生成した路線図を基にしながら部分グラフを逐次添加することで最終的に路線図を生成する．路線図を生成するときは，Nöllenburg と Wolff [3] の MIP モデルを参考に定式化する．グラフを逐次添加して路線図を生成するとき，新しく追加した部分グラフに属する辺のみ方向と辺長を変更できるように定式化し，それ以外の辺は方向は固定し，辺長のみ変更できるように定式化する．

分割方法

グラフ $\widetilde{G}$ の分割を行う．このとき，乗り換え駅以外のラベルを表す辺については考えない．本研究では，外部木と面近傍グラフに分割する．それぞれの定義は以下で述べる．

外部木を定義する前に外部辺を定義する．**外部辺**とは，非有界な面のみに接している辺とする (図 3(a) の辺 e_1，e_2，e_3，e_4，e_5，e_6)．連結している外部辺の集合を**外部木** ET とする (図 3(a) の $\{e_1, e_2, e_3, e_4, e_5\}$，$\{e_6\}$)．グラフ $\widetilde{G}$ の各有界面 f に対して，1 つの f からなる部分グラフを f の**面部分グラフ**と呼ぶ (図 3(b) の面 f_1，f_2，f_3，f_4，f_5 からなる部分グラフ)．有界な面に接続する各頂点 v に対して，v に接続する全ての有界な面の面部分グラフの和集合を，v の**面近傍グラフ** $N(v)$ と呼ぶ (図 3(b))．ここで，グラフ $G_1 = (V_1, E_1), G_2 = (V_2, E_2), ..., G_k = (V_k, E_k)$ の和集合とは，$\cup_{i=1}^{k} V_i$ の頂点の集合と $\cup_{i=1}^{k} E_i$ の辺の集合からなるグラフのこととする．また，全ての面近傍グラフ $\{N(v)\}$ を入力グラフ G での頂点 v の次数で降順にソートし，$\mathcal{N} = \{N(v_1), N(v_2), ..., N(v_m)\}$ とする．



(a) 外部辺 $e_1, e_2, e_3, e_4, e_5, e_6$ と(b) 面 f_1, f_2, f_3, f_4, f_5 から外部木 $\{e_1, e_2, e_3, e_4, e_5\}, \{e_6\}$.なる面部分グラフと頂点 v の面近傍グラフ $N(v)$.

図 3. 分割方法.

アルゴリズム

初期レイアウトの路線図を生成するアルゴリズムについて述べる．前述した外部木 $\{ET_1, ET_2, ..., ET_l\}$ とソートされた面近傍グラフ $\mathcal{N} = \{N(v_1), N(v_2), ..., N(v_m)\}$ を用いて，路線図を生成していく．これらのデータを面近傍グラフ，外部木の順に逐次添加しながら初期レイアウトとなる路線図を生成する．

路線図を部分的に生成するとき，Nöllenburg と Wolff [3] の MIP の定式化を用いる．また，[3] による交差禁止制約の逐次添加を用いて MIP サイズの削減を行う．

逐次添加しながら路線図を生成するときは，新しく追加した部分グラフに属する辺のみ方向と辺長を変更できるように定式化し，それ以外の辺は方向を固定し，辺長のみ変更できるようにする．新しく追加した部分グラフに属する辺のみ方向と辺長を変更することで，すばやく路線図を生成できる．また，それ以外の辺は方向は固定するが，辺長を変更できるようにすることで，交差しない路線図を生成できる．ここでは，新しく追加した部分グラフに属する辺が，すでに追加されている辺の場合は方向と辺長を変更できるように定式化する．

アルゴリズムは以下の 2 ステップからなる．まずは，ソートされた面近傍グラフを逐次添加しながら路線図を生成する．次に，外部木を逐次添加していく．外部木を追加していく順番については，入力されたものから順に追加していく．

Step 1

G_0 を空集合とする．

For each $i = 1, ..., m$:

$N(v)_i = (V_{N(v_i)}, E_{N(v_i)})$ を G_{i-1} に追加し，得られたグラフを $G_i = (V_i, E_i)$ とする．

ここで，G_i の頂点は $V_i = V_{i-1} \cup V_{N(v_i)}$ とし，辺は $E_i = E_{i-1} \cup E_{N(v_i)}$ とする．

しかし，$N(v_i)$ の全ての頂点と辺が G_{i-1} に追加されていた場合，$N(v_i)$ は G_{i-1} に追加せず，G_{i-1} を G_i とする．

G_i に対して MIP を用いて路線図を生成し，その路線図を Γ_i とする．

このとき，$N(v_i)$ に属する辺は方向と辺長を変更できるように定式化し，それ以外の辺は，方向は Γ_{i-1} の方向で固定し，辺長のみ変更できるように定式化する．

Step 2

For each $j = 1, ..., l$:

ET_j を G_{m+j-1} に追加し，G_{m+j} とする．

G_{m+j} に対して MIP を用いて路線図を生成し，その路線図を Γ_{m+j} とする．

このとき，ET_j に属する辺は方向と辺長を変更できるように定式化し，それ以外の辺は，方向は Γ_{m+j-1} の方向で固定し，辺長のみ変更できるように定式化する．

これらの 2 ステップ後，制約を満たした路線図が生成され，反復改善法の入力となる．

4.3 反復改善法

反復改善法では，初期レイアウト生成手法で生成した路線図をより良くする．反復改善法では，路線図を分割し，部分的に最適化を行うことで，局所最適解となる路線図を生成する．

分割方法

反復改善法では，路線図をインターチェンジパスに分割する．**インターチェンジパス**とは，次数 2 以外の頂点間の辺の集合とする (図 4).

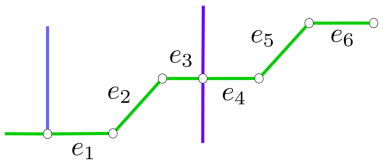


図 4. インターチェンジパス $\{e_1, e_2, e_3\}$, $\{e_4, e_5, e_6\}$.

アルゴリズム

反復改善法では，全てのインターチェンジパスに対して，そのインターチェンジパスのみが辺の方向と長さを変更できるように定式化する．一方，それ以外の辺の方向は固定し，辺長のみが変更できるようにする．全てのインターチェンジパスに対して，最適解となる路線図を生成し，局所最適解となる路線図が生成されるまで繰り返す.

目的関数

既存研究 [3] の MIP モデルでは目的関数の辺長が線形であるため，頂点が偏って配置されることがある．そこで反復改善法では，以下のようななるべく満たす制約を追加する．

(S4) 頂点がなるべくバランスよく配置される．

また，この制約に対応する目的関数を [3] の MIP モデルに追加する．本研究では，接続している 2 辺 uv, vw の頂点 u, w がともに次数 2 でなく，次数 2 の各頂点 v に対して，その 2 辺の差の絶対値の総和をとる．頂点 v に対して $\deg(v)$ は v の次数，$\lambda(uv)$ を辺 uv の長

さとし，以下のような式に重みを乗じ，目的関数に加える．

$$\mathrm{cost}_{S4} = \sum_{v \in V, \deg(v)=2, \deg(u)\neq 2, \deg(w)\neq 2} |\lambda(uv) - \lambda(vw)|$$

5　計算機実験

これらの手法を実装し，実際の路線網に対して計算機実験を行った．実験環境は次のとおりである．MIP を解くためのソルバは CPLEX12.7.1 を用いる．

- CPU : Intel(R) Xeon(R) 3.60GHz × 6.
- Memory : 128.0 GB.
- MIP Solver : CPLEX 12.7.1.

実験データは [5] より取得し，今回は東京の地下鉄のデータを使用し実験を行った．東京の路線網は，次数が 8 より大きい駅が存在し，グラフサイズも大きいため [3] の手法では路線図を生成できない．

初期レイアウト生成手法を用いて生成した路線図が図 5 である．初期レイアウト生成手法では，目的関数の各コストの重みによって計算時間が大きく変化する．経験的に位置関係の重みを大きくすることですばやく路線図を生成できるため，初期レイアウト生成手法では $(\lambda_{(S_1)}, \lambda_{(S_2)}, \lambda_{(S_3)}) = (1, 30, 1)$ とした．計算時間は 13 分であった．

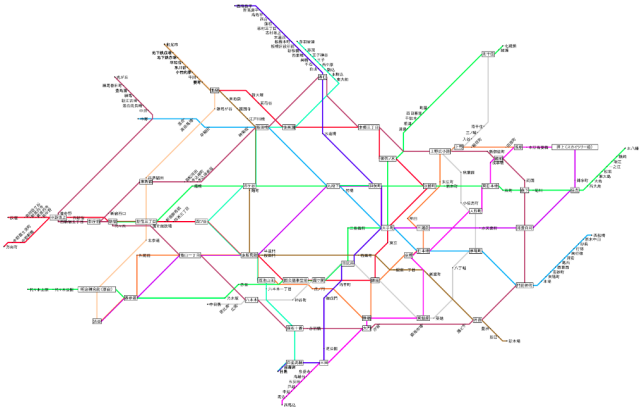


図 5. 初期レイアウト生成手法による東京の路線図.

図 5 に対して，反復改善法を用いた路線図が図 6 である．初期レイアウト生成手法による路線図と比べて，全体的に辺長が短くなっている．反復改善法も初期レイアウト生成手法同様に目的関数の各コストの重みで計算時間が大きく変わる．ここでは，$(\lambda_{(S_1)}, \lambda_{(S_2)}, \lambda_{(S_3)}, \lambda_{(S_4)}) = (1, 20, 5, 1)$ とした．計算時間は 5 時間であった．

また反復改善法では，目的関数を追加したことにより，頂点がバランス良く配置されている．

6　まとめと今後の課題

本研究では，複雑な路線網に対して，3 つのステップからなる路線図生成手法を提案した．第 1 ステップでは，乗り換え駅に対するラベル配置では，次数が 8 より大きい駅が存在する路線網に対しても [3] の MIP モデルを解くことができるようになった．第 2 ステップ

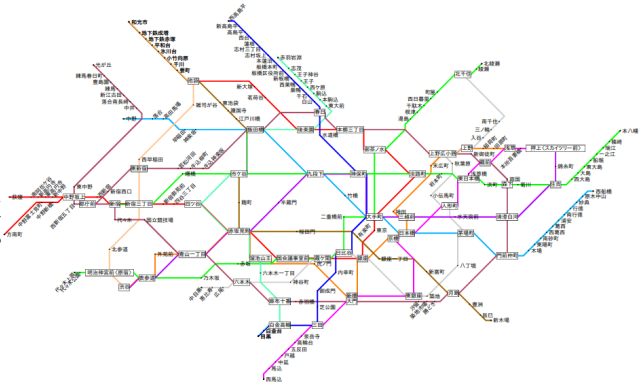


図 6. 反復改善法による東京の路線図.

である初期レイアウト生成手法では，路線網を分割し部分的に路線図を生成していくことで，制約を満たしている路線図を素早く生成することができた．第 3 ステップである反復改善法では，初期レイアウト生成手法で生成した路線図に対して局所探索を行うことで，複雑な路線網に対しても最適ではないが，局所最適解となる路線図を生成することができた．さらに，反復改善法では，目的関数を追加したため，偏って配置される頂点を少なくすることができた．

今後の課題として，各コストの重みによって計算時間が大きく異なるため，計算時間が早い重みの条件を見つける必要がある．現状では，経験的に位置関係の重みを大きくすることで計算時間が短くなる傾向がある．また，局所探索の計算時間が長いため，計算時間を短縮する工夫が必要である．

謝辞

本研究を進めるにあたり, 適切な御指導, 御指摘をしていただきました中央大学理工学部情報工学科の今井桂子教授，明治大学先端数理科学インスティテュートの森口昌樹特任講師に心から感謝致します．また，今井研究室および情報工学専攻の学生各位には，研究に対するアドバイスを与えていただくなど，大変お世話になりました．最後に，今井研究室の院生各位，学生各位の協力なしには，この論文は完成しませんでした．深く感謝致します．

参考文献

[1] K. Garland, “Mr Beck’s Underground Map,” Capital Transport, 1994.

[2] J. Stott , P. Rodgers, J. C. Martinez-Ovando and S. G. Walker, “Automatic metro map layout using multicriteria optimization,” *IEEE Transactions on Visualization and Computer Graphics*, Vol. 17(1), pp. 101–114, 2011.

[3] M. Nöllenburg and A. Wolff, “Drawing and labeling high-quality metro maps by mixed-integer programming,” *IEEE Transactions on Visualization and Computer Graphics*, Vol. 17(5), pp. 626–641, 2011.

[4] S. -H. Hong, D. Merrick and H. A. D. do Nascimento, “Automatic visualization of metro maps,” *J. Visual Languages and Computing*, Vol. 17(3), pp. 203–224, 2006.

[5] 株式会社コードプラス, “ 駅データ.jp,” `http://www.ekidata.jp`, 2016.

[6] 東京メトロ, “ 路線・駅の情報,” `http://www.tokyometro.jp/station/index.html`, 2016.