

鉄道路線に対する略地図生成手法

Automatic Drawing of Brief Railroad Maps

情報工学専攻 傳保 能幸

DEMBO Takayuki

概要

近年、情報技術の発展に伴ない、様々なデータを計算機上で管理、加工する機会が増加している。また、通信技術の発展に伴ない、さまざまな情報端末からこれらのデータを送受信し、利用することが可能になっている。このようなデータとして地図や略地図がある。中でも略地図は描画に関する自由度が高いため、目的に応じて地図を変形、簡略化し、視認性や有用性の高い地図と言える。略地図は従来職人の手によって1つ1つ描画されてきたが、近年、略地図を計算機上で自動描画する研究が行なわれている。本研究では、鉄道路線を対象とした略地図を考える。鉄道路線の略地図描画に関する既存手法では、見易さから線分の描画方向を数方向に限定しようとしているが、その多くで、描画方向を限定できていない。一方、描画方向を限定できる手法では、計算に時間がかかっている。さらに、交通網の発展した都市部では、同じ区間を複数の路線が並走していることがある。このような場合、実際に用いられている略地図では、線分を平行移動して、路線の情報を保持している。しかし、並走している路線を考慮した略地図描画手法は今のところ存在していない。よって本研究では、これらのことを考慮した略地図描画手法を提案し、実装する。

キーワード：略地図、鉄道路線、8方向、MIP、高速描画、並行路線

1 序論

近年、情報技術の発展に伴ない、様々なデータを計算機上で管理、加工する機会が増加している。また通信技術の発展に伴ない、さまざまな情報端末からデータを送受信し、利用することが可能になっている。そのようなデータの1つとして略地図がある。略地図とは、点と線分からなるグラフ(地図)を与えたときに、何らかの意味で、デフォルメ化あるいは単純化した地図のことをいう。

本研究では、鉄道路線に対する略地図描画手法を提案する。略地図描画に関する研究は、数多く行なわれている。例えば、M. Nöllenburgらの手法[1]がある。この手法の特徴として、線分の描画方向を8方向に限定できる点や混合整数計画法(MIP)を用いて定式化を行なう点がある。8方向とは、 x 軸方向と45度刻みの方向とする。この手法は、規則的な略地図を得ることができるが、計算に時間がかかる。

また、高速描画を念頭においた手法も数多く提案されているが、現時点では、線分の描画方向を8方向に限定した手法は存在しない。また、計算時間も[1]に比べれば格段に早い。携帯端末での利用を考えると、まだ高速化の必要がある。

また、同じ区間を複数の路線が並走していることもある。このような路線のことを、本研究では並行路線という。実際に利用されている多くの略地図では、路線の情報を保持するため、並行路線を横に平行移動して描画している。しかし、並行路線を考慮した既存手法は存在していない。また、既存手法を用いた後で、線分を平行移動すると、別の路線と交差する可能性もあるため好ましくない。

よって、本研究では、鉄道路線に対する略地図描画手法を3つ提案する。まず、既存手法[1]をもとに、数理計画法を用いて、並行路線を考慮した手法を提案する。次に、鉄道路線に対する略地図の高速描画手法を提案する。最後に、2つ目の手法をもとに、並行路線を考慮した高速描画手法を提案する。

2 MIPによる並行路線を考慮した略地図描画手法

ここでは、既存手法[1]をもとに、並行路線を考慮した略地図描画手法を提案する。[1]では、略地図として必ず満たすべき条件を以下のように定めている。

- (H1) 各点から見た隣接駅の順番を保持する。
- (H2) 各線分は8方向に描画する。
- (H3) 各隣接駅間 $e \in E$ は l_e 以上離す。
- (H4) 端点を共有しない各2線分は d_{\min} 以上離す。

また、できるだけ満たしたい条件を以下のように定めている。

- (S1) 各路線をできるだけ少ない線分で表現する。
- (S2) 辺長和をできるだけ小さくする。
- (S3) 隣接駅との位置関係をできるだけ保持する。

[1]で扱っている問題を以下に記す。入力として、最大次数8の平面グラフ $G = (V, E)$ と、各路線情報 $L \in \mathcal{L}$ を与える。また、各線分 $\forall e_j \in E$ は少なくとも1つの L に属している。さらに、長さ l_e 、 d_{\min} と重み $\lambda_1, \lambda_2, \lambda_3$ を与える。このとき、条件(H1)から(H4)を全て満たし、条件(S1)から(S3)をできるだけ満たす略地図を求める。

[1]では各条件をMIPを用いて定式化している。また、条件(S1)から(S3)に対するコストを求め、それぞれに重み $\lambda_1, \lambda_2, \lambda_3$ を乗じた値の和を目的関数とし、その最小化を図っている。

提案手法では、基本的に[1]を用いることとする。しかし、[1]は駅を点として捉えている手法である。そのため、並行路線を描画する場合、グラフを変形後、手作業で線分を平行移動しなければならない。この方法では、平行移動した線分が、別の線分と交差することがある。したがって、入力として、さらに S_h, S_w を加え、図1のようにグラフを変形し、駅を長方形として捉えることで、並行路線の描画を考慮した定式化を行なう。

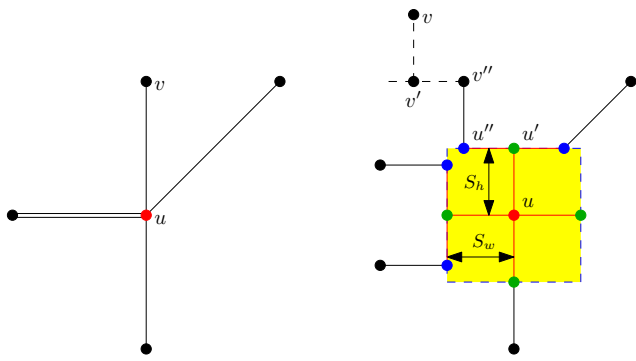


図 1. 駅を長方形 (黄色の領域) として捉える.

計算機実験

実験環境は以下のとおりである.

- OS : Windows XP x64 Professional.
- CPU : Intel Xeon 5160 (3.00 GHz) × 2.
- Memory : 6.00 GB.
- MIP Solver : CPLEX 11.0.

東京 23 区内の JR 鉄道網 (図 2) に対する出力結果の例を図 3 に示す. また, 計算量削減のため, 次数 3 の頂点間にある次数 2 の頂点を, 高々 2 つになるようにあらかじめ削除した. MIP に関する計算は 1 時間行ない, 実行可能解を出力した. 図 3 では, 並行路線を横に平行移動できている. このように, 駅を長方形として捉えることにより, 並行路線も自動描画できた.

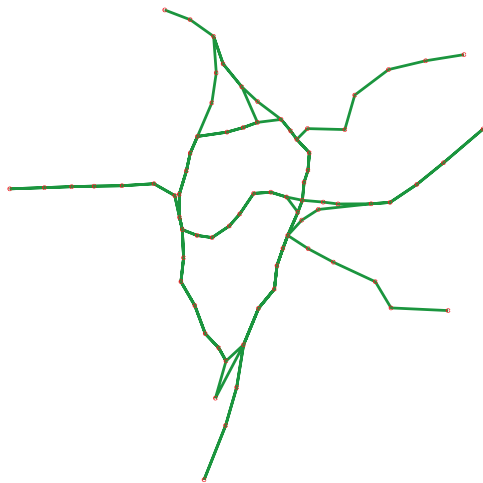


図 2. 東京 23 区内の JR 鉄道網.

3 鉄道路線に対する略地図の高速描画手法

ここでは, 線分の描画方向を 8 方向に限定した, 略地図の高速描画手法を提案する. 先に定義した問題は, できるだけ満たしたい条件が多く, 略地図の高速描画には不向きである. したがって, 略地図として必ず満たすべき条件を以下のように定める.

- (H1) 各点から見た隣接駅の順番を保持する.
- (H2) 各線分は 8 方向に描画する.
- (H3) 端点を共有しない各 2 線分は交差しない.

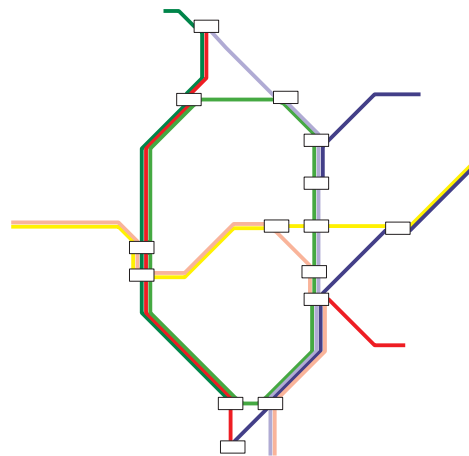


図 3. 出力結果.

また, できるだけ満たしたい条件を以下のように定める.

(S1) 各路線をできるだけ少ない線分で表現する.

ここでは, 以下の問題を扱う. まず入力として, 最大次数 8 の平面グラフ $G = (V, E)$ と, 各路線情報 $L \in \mathcal{L}$ を与える. このとき, 出力する略地図は, 条件 (H1) から (H3) を全て満たし, 条件 (S1) をできるだけ満たす略地図を出力する. アルゴリズムの概要を以下に示す.

1. 仮頂点を追加し, 線分を 8 方向に埋め込む.
2. 次数 2 の頂点の移動.
 - i. スライド.
 - ii. フリップ.
 - iii. 平滑化.
3. 次数 3 以上の頂点の移動.

3.1 グラフの 8 方向化

ここでは, 入力されたグラフの線分の 8 方向化について述べる. 一般に, 入力されたグラフの線分の描画方向は 8 方向に限定されていない. したがって, 線分上に仮頂点を加え, 分割した線分をそれぞれ 8 方向のいずれかの方向に埋め込む. まず, 各線分 $uv \in E$ の両端点 $u, v \in V$ の位置関係を考える. つまり, 8 方向のうち, 点 u からみた v の方向に最も近い方向を求める. この方向を u から v への描画方向とする. 例えば, 図 4 では, 点 u から v への描画方向は 5, 点 v から u への描画方向は 1 となる. ただし, 各点 u において, あ

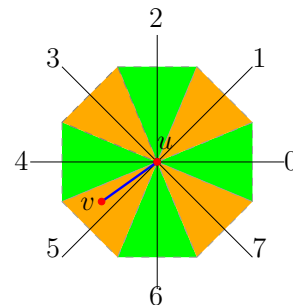


図 4. 8 方向.

る描画方向に対して複数の線分が競合する場合, 各描画方向に対応する線分が高々 1 つとなるように, 各線分の描画方向を隣の方向に変更する.

次に、線分 (u, v) を 8 方向に埋め込む。各頂点 u, v で求めた描画方向の組合せに対応するテンプレート (図 5) を用いて 8 方向に埋め込む。また、線分が交差してしまう場合には、テンプレートを使用せず、交差しないように 8 方向に埋め込む。

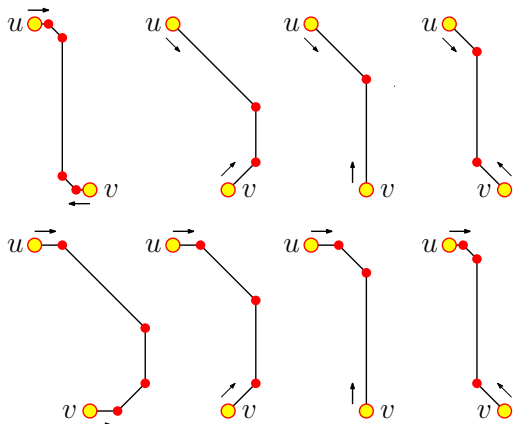


図 5. 8 方向化のためのテンプレート。

3.2 次数 2 の頂点の移動

ここでは、次数 2 の頂点の移動による、グラフの単純化について述べる。また、次数 2 の頂点 v を端点とする線分 uv, vw が、単純化により一直線になった場合、それ以降 uw として処理する。

まず、スライドについて述べる。鉄道路線は、急にカーブすることがあまりなく、一定方向に進んでいることが多い。そのため、路線図でも 45 度や 90 度のような急カーブを描画することは好ましくない。よって本研究では、図 6, 7 のように、点をスライドさせ、できるだけ緩やかにカーブするような処理を行なう。

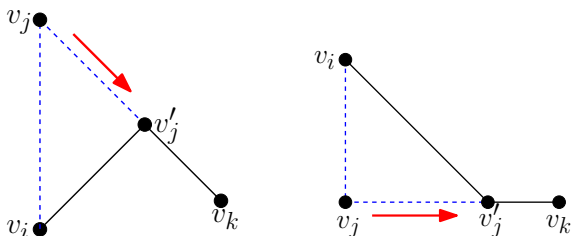


図 6. 45 度のスライド。

図 7. 90 度のスライド。

次に、フリップについて述べる。8 方向化を行なう際、多くの仮頂点を加えるため、グラフの頂点や線分の数が増え、グラフが複雑になる。また、多くの場合、図 8 のように、頂点を移動することでカーブを減らし、グラフを単純化することができる。

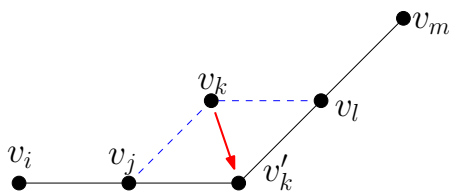


図 8. フリップ。

最後に、平滑化について述べる。鉄道路線は少し蛇行していても、一定方向に進んでいることが多い。そのため、蛇行はしているが、一定方向に進んでいるこ

とが明らかな場合は、図 9 のように単純化する必要がある。

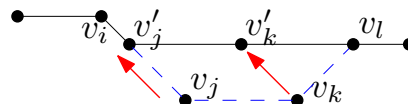


図 9. 平滑化。

3.3 次数 3 以上の頂点の移動

ここでは、次数 3 以上の頂点の移動による、グラフの単純化について述べる。8 方向化の際に追加された仮頂点の多くは、本稿の第 3.2 節で提案した単純化手法を適用することで消去される。さらに、各路線の線分の数が減り、より長い線分で描画されている。しかし、次数 3 以上の頂点も移動しなければ、単純化ができないこともある。図 10 に例を示す。次数 3 の頂点 (青点) が一直線上に並んでいないため、次数 2 の頂点を移動しても、これ以上単純化はできない。

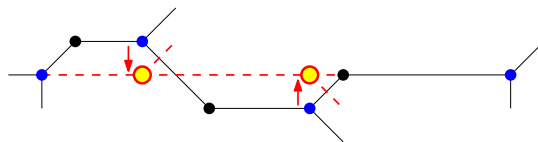


図 10. 次数 3 以上の頂点を移動しなければならない例。

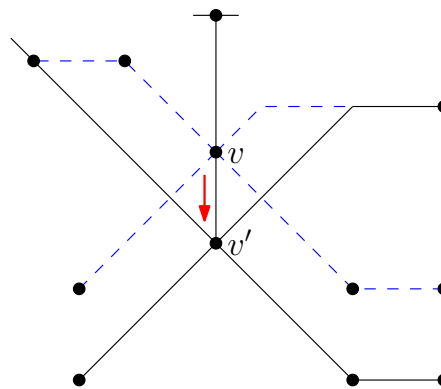


図 11. 次数 3 以上の頂点の移動。

よって、次数 3 以上の頂点の移動を行なう。頂点は、各線分が交差しなければ、任意の位置に移動できるが、本手法では、以下の条件を満たす位置に移動する。

1. 路線のカーブの数をできるだけ少なくする。
2. 移動距離をできるだけ短くする。

計算機実験

実験環境は以下のとおりである。

- OS : Windows XP Professional.
- CPU : Intel Pentium M (1.70 GHz).
- Memory : 1.00 GB.

入力 (図 2) に対する出力結果を図 12 に示す。計算時間は 0.109 [sec] である。本提案手法は単純なものであるが、実用に耐えうる出力が得られたと考えている。

4 並行路線を考慮した略地図の高速描画手法

ここでは、本稿の第 3 節の手法をもとに、並行路線を考慮した高速描画手法を提案する。まず、並行路線

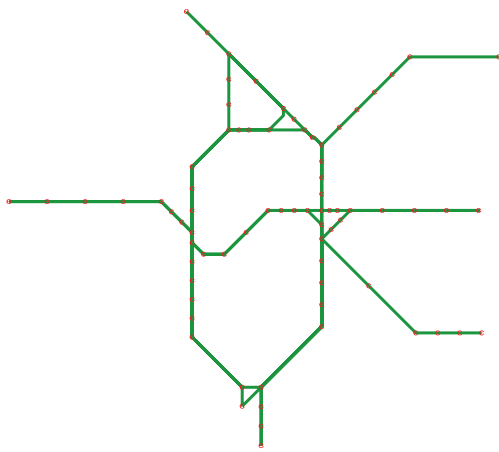


図 12. 出力結果.

を考慮するため、駅を図 13 のように同心円として捉える。例えば、並行路線が 3 路線の場合、線分の幅を l をとすると、同心円の直径は $3l$ となる。



図 13. 駅を同心円として捉える.

本稿の第 3 節の手法は、グラフ変形の過程で、常にグラフの平面性を保持する手法となっている。しかし、幅 l を考慮すると、グラフの平面性が失われることがある。よって、路線の幅を l としても平面性を保てるようにグラフを変形する。本研究では、まず主座標分析 (PCM) を用いる。さらに、同心円や線分の交差を除去するための点移動アルゴリズムを提案し、実装する。点移動アルゴリズムの概要を以下に記す。

1. 同心円や線分が交差している部分を見つける。
2. 交差部に関連する同心円 p に対し、以下を行なう。
 - a. 同心円 p の隣接駅の重心 g を求める。
 - b. 交差が改善するならば、 p を g に移動する。

計算機実験

実験環境は、本稿の第 3 節と同様である。ここで、 d_{ij} は 2 点 p_i, p_j 間の L^2 -距離、 d_{\max} は各線分 $p_i p_j \in E$ の中で最長の d_{ij} を表わし、PCM で用いる類似度を $\frac{d_{ij} + d_{\max}}{2}$ とする。入力 (図 2) に対して、PCM と点移動アルゴリズムを用いた結果を図 14 に示す。都市部の駅の密集している部分が拡大され、駅の間隔が広がっている。逆に郊外では、駅の間隔が若干縮まっている。さらに、図 14 を入力とし、前章で提案した略地図の高速描画手法を適用した結果を図 15 に示す。計算時間は 0.151 [sec] である。並行路線の多い部分では、線分を描画する領域が多くとられている。出力された形状は、第 2 節の出力 (図 3) とは異なるが、実用に耐えうる出力が得られたと考えている。また、MIP を用いた提案手法は計算に 1 時間以上かかったが、本手法は 0.151 [sec] と非常に高速に出力を得られている。

5 結論

本研究では、線分の描画方向を 8 方向に限定した、鉄道路線に対する略地図描画手法を 3 つ提案した。既存

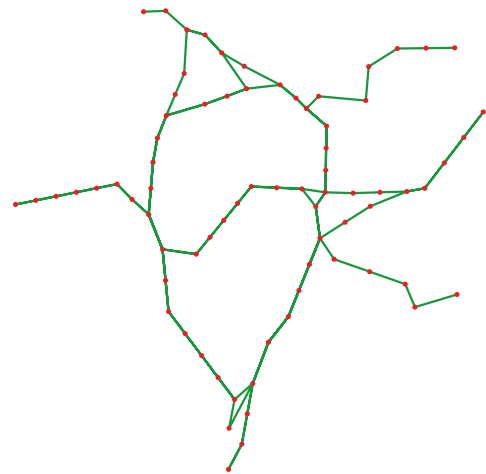


図 14. PCM と点移動アルゴリズム適用後.

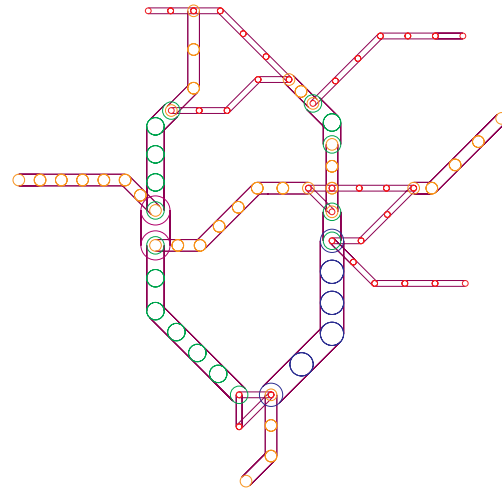


図 15. 出力結果.

研究では、並行路線を考慮した略地図描画手法は存在しておらず、線分の描画方向を 8 方向に限定し、かつ高速に略地図を描画する手法も存在していなかった。

よって、まず、駅を長方形として捉えることにより、MIP による並行路線を考慮した略地図描画手法を提案した。次に、線分の描画方向を 8 方向に限定した、単純な高速描画手法を提案した。さらに、駅を同心円として捉えることにより、並行路線を考慮した高速描画手法も提案した。これらの提案手法は、それぞれ特徴が異なるため、ユーザの用途や目的に応じて使い分け、略地図を描画することが望ましい。

謝辞

本研究を進めるにあたり、適切な御指導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に心から感謝いたします。また、たくさんの助言をしてくれた今井研究室の友人達に感謝いたします。

参考文献

- [1] M. Nöllenburg and A. Wolff, "A Mixed-Integer Program for Drawing and Labeling High-Quality Metro Maps," *Proceedings Graph Drawing*, pp. 321-333, 2006.