

# 地図における道路に対するラベル配置アルゴリズムの実験的評価

## Experimental Evaluation of Label Placement Algorithms for Roads

情報工学専攻 杉本 竣亮

Shunsuke SUGIMOTO

**概要:** ラベル配置問題とは、地図作成において施設や道路の名称やシンボルを地図上の適切な位置に配置する問題である。地図上の建物、道路などの物体をオブジェクトと呼び、道路などの一次元オブジェクトにラベルを配置する場合、ラベルを一次元オブジェクトの上や下に配置する方法や一次元オブジェクトに幅を持たせて埋め込むという方法が用いられる。本論文では、道路に対するラベル配置問題を考える。ラベルは一次元オブジェクトの中に埋め込んで配置する。ここで、隣接する交差点同士の間にある道路を、道路区間と呼ぶ。本論文での目的はラベル同士の重複は禁止し、できるだけ多くのラベルを道路区間に配置することである。この問題は一般的に NP 困難であることが証明されているが、Gemsa らは道路地図が木であれば多項式時間で問題を解決できることを示している。本論文では、計算機実験を通して、彼らのアルゴリズムとヒューリスティクスを加えたアルゴリズムの実験的評価を行う。

**キーワード:** ラベル配置問題, 道路地図

### 1 序論

地図を使用するとき、道路の名前を参照する必要がある。しかし、他の道路の注記と重なったり、注記のない道路部分が多く存在すると、可読性は損なわれる。

本論文では、道路に対するラベル配置問題を扱う。ラベル配置問題とは、建物や道路といったオブジェクトに対し、適当な位置や大きさなどの制約を満たすように注記の書かれたラベルを配置する問題である。地図のラベル配置問題は、地図内の建物を点や領域、道路を辺として扱い、ラベルを配置する。

地図から道路のみを抽出し、各道路に道路の名前などの情報が書かれている地図をここでは道路地図と呼ぶ。道路地図に対するラベル配置問題に関する以前から行われていた研究として、Chirié [1] は、地図作成者らに行ったインタビューに基づき、道路に配置するラベルはなるべく道路の形状に沿っていることが望ましく、異なる 2 つのラベルが重複しない方が良いといった、道路地図に関する評価の基準を提示した。Vaaraniem ら [3] は、2D および 3D の地図やネットワークモデルに対し力学モデルのラベリングアルゴリズムを適用させることで、動的に変化する情報に対し、リアルタイムで衝突の無いラベル配置を可能にした。

Gemsa ら [2] は、道路地図内の道路にラベルを配置する問題を扱っている。道路地図内の隣接する交差点同士を結ぶ道路をここでは道路区間と定義し、各道路は一個以上の道路区間を要素として持つ集合と定義する。幾何学的には、道路地図は道路の交差点や道路の屈折を表現するための形状点を要素とした集合を  $V$ 、各道路区間の集合を  $E$  として、道路グラフ  $G = (V, E)$  と表現できる。 $G$  内の各道路は  $G$  の部分グラフであり、各辺は道路区間である。道路に存在する交差点の性質から、異なる二つの道路が  $G$  の頂点において互いに交差してもよいものとする。十分な幅を持つ道路区間の中に道路名であるラベルを配置して、道路区間が明確に識別されるようにする (図 1)。ラベルが配置された道路区間をラベルによってカバーされていると定義する。Gemsa ら [2] の研究においては、道路地図内

のカバーされている道路区間の数が最大になるようなラベル配置を見つけることを目的としており、この問題を Max Identified Roads と呼んでいる。

一般的な都市の道路地図には、多くの閉路が存在する可能性がある。Gemsa ら [2] は、閉路のある道路地図の道路における Max Identified Roads は NP 困難であることが証明しているが、道路地図が木構造であれば多項式時間で解けるアルゴリズムを提案した。本論文では、[2] のアルゴリズムとヒューリスティクスを組み合わせたアルゴリズムを実装し、計算機実験による実験的評価を行う。

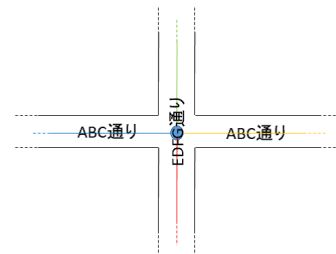


図 1. 道路地図における道路名のラベル配置の例

### 2 木構造の道路地図に対するラベル配置問題

本章では、[2] で提案された Max Identified Roads の問題設定やアルゴリズムについて述べる。

#### 2.1 問題設定

Max Identified Roads では、交差点内でラベルが交差することや、任意の道路区間に異なる道路名のラベルを配置することを禁じており、ラベルは全て道路区間に埋め込むことで配置する。

したがって、Max Identified Roads では、以下のことを考慮する。

- ラベルは道路の形状に沿って、埋め込んで配置する。
- なるべく全ての道路区間が識別されるようにする (図 2)。
- 2 つのラベルが重複しないようにする (図 3)。

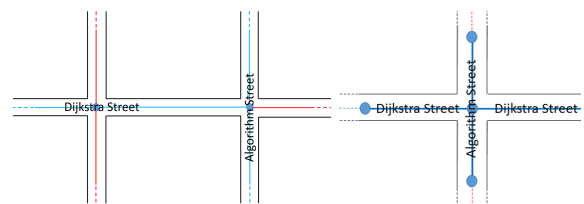


図 2. 赤色の道路区間が識別されていない

図 3. 重複しないラベル配置

## 2.2 アルゴリズム

Gemsa ら [2] では、Max Identified Roads を解くために、道路地図を木構造に変換して、道路名のラベル配置を考えるアルゴリズムを提案した。具体的なアルゴリズム名を与えられていなかったため、本論では Tree Map Labeling と呼ぶ。

Tree Map Labeling の全体の流れは以下の通りである。

- ラベルの配置が明白な道路区間の隔離
- 交差点付近に対する正規頂点の配置
- 鎖状のラベル配置による頂点の追加
- 各頂点に対するラベル配置

### 2.2.1 入出力

Tree Map Labeling の説明に必要な語句を次のように定義する。

- $T = (V, E)$  : 道路地図の木
- $V = \{v_i | i = 1, \dots, N\}$  : 交差点の集合 ( $N$  は点数)
- $E = \{e_j | j = 1, \dots, M\}$  : 道路区間の集合 ( $M$  は辺数)
- $\mathcal{R} = \{R_k | k = 1, \dots, O\}$  : 道路名の集合 ( $O$  は道路名の数)
- $R_k = \{r_l | l = 1, \dots, P\}$  : 道路名  $k$  の道路区間の集合 ( $P$  は道路  $k$  の道路区間数)
- $L = \{l_m | m = 1, \dots, Q\}$  : ラベルの集合 ( $Q$  はラベルの数)
- $h(l)$  : ラベル  $l$  の始点
- $t(l)$  : ラベル  $l$  の終点
- $len(l)$  : ラベル  $l$  の長さ
- $\omega(L)$  : ラベル集合  $L$  によってカバーされている道路区間数

入力は、道路地図における道路名、交差点、道路区間及びその道路区間の名前とする。入力したデータから形成される道路グラフは、木構造でなければならない。任意のラベル集合を  $L'$  とし、 $\omega(L') \leq \omega(L)$  であるようなラベル集合  $L$  を、つまり、道路グラフ内でカバーされている道路区間数が最大であるようなラベル配置を出力とする。

### 2.2.2 ラベルの配置が明白な道路区間の隔離

入力された各道路区間  $r$  の中で、以下のいずれかの条件を満たすような道路区間は、最適なラベル配置においてラベルが配置されているか明白なので、すべてグラフ内から隔離する。

- 道路区間が1つだけの道路を削除する。
- 図4の真ん中の道路区間のように、ラベルを全て挿入できる長さの道路区間であり、隣接する同じ道路名の区間もラベルを全て挿入できる長さであれば、その道路区間を排除する。

また、ラベルを2つ連結して配置するのに十分な長さである道路区間は二等分に分割する。これらの処理により、グラフ内の閉路の数を削減することができる。

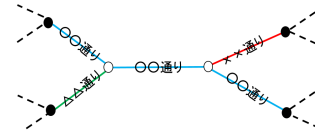


図4. 隔離できる道路区間の例

### 2.2.3 交差点付近に対する正規頂点の配置

道路の行き止まりや、道路を形成する形状点といった、グラフ内の交差点以外の頂点を、ここでは正規頂点と呼ぶ。ラベルを配置するとき、交差点内でラベルの交差が発生しているかを判定するために、入力された頂点の中で交差点である点の近傍に、図5の赤色の頂点のような正規頂点を配置する。これらの追加された正規頂点と交差点を端点に持つ辺をここでは交差辺と呼び、ラベルの端点が、追加された正規頂点を除き、交差辺内に存在してはいけないものとする。

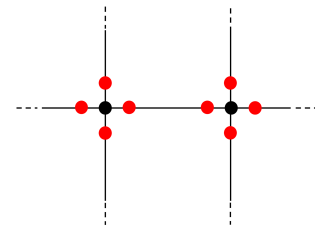


図5. 正規頂点 (赤色の点) の追加

### 2.2.4 Chain Vertex の追加

最適なラベル配置を考えるために、同じ道路名のラベルが連結して配置されている場面を想定する必要があるため、そのようなラベルの連結点をグラフ内に追加する。まず、交差点を除く、各正規頂点  $v$  を一つずつ見ていく。その頂点を始点とし、以下の条件をすべて満たす場合、図6のように、根  $\rho$  に向かって配置できる赤色の垂直ラベルの終点  $t(1)$  をグラフ内に追加する。

- 根までの距離がラベルの長さよりも短い
- 配置するラベル内の道路名が全て同じであること
- 配置されるラベルによってカバーされる道路区間数が増えること

頂点を追加した場合、図6の  $h(2)$  のように、配置したラベルの終点を新たなラベルの始点とし、同様に垂直ラベルの配置を考える。条件を満たさなかった場合、その正規頂点を始点にした頂点の追加を終了する。

### 2.2.5 各頂点に対するラベル配置

最適なラベル配置を見つけるために、各頂点を根にした部分グラフのラベル配置を考えていく。頂点を見ていく順番としては、根からの最短経路に含んでいる頂点の数が最も多い頂点から見ていき、最終的には、木  $T$  の根を端点としたラベル配置を考える。具体的には、 $T$  内の各頂点  $u$  について、ラベル候補である集合  $C(u)$  を考える。 $u$  が  $T$  の正規頂点である場合、集合  $C(u)$  は点  $u$  を端点を持つ全てのラベル  $l$  の集合となる (図7)。 $u$  が交差点である場合、 $C(u)$  は、 $T$  内の正規頂点で開始または終了し、最も根に近い点  $u$  である全てのラ

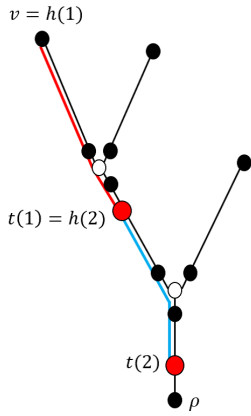


図 6. Chain Vertex の追加例

ベルの集合となる (図 8). また, 葉までの道路区間の長さが短すぎて, ラベルを配置できないといった可能性を考えるため, ラベルを配置しない  $\perp_u$  というラベル候補も  $C(u)$  に追加する. 例えば,  $u$  が道路グラフの葉である場合,  $u$  を根にした部分木に道路区間が存在しないことは明らかであり, ラベル候補  $C(u)$  の要素は,  $\perp_u$  のみとなる.

ラベル候補  $c \in C(u)$  を決定した際,  $u$  を根とする部分木のカバーできる道路区間数は, ( $c$  でカバーした道路区間数) + ( $c$  に含まれる頂点及び入射している辺を削除すると発生する各部分木の最適ラベル配置によってカバーされる道路区間数) である. 決定した  $c$  が  $\perp_u$  の場合にカバーできる道路区間数は,  $u$  の子を根にした部分木各部分木の最適ラベル配置によってカバーされる道路区間数となる. この式から最もカバー数の多い候補  $c$  を選び, 最終的に  $u$  を根としたラベル候補を求めることで,  $T$  全体の最適なラベル配置を求めることができる. 任意のラベル  $l$  とその子集合を根にした部分木の最適なラベリング  $L(l)$  を, 以下のように定義する.

$$L(l) = \cup_{v \in N(l)} L(v) \cup \{l\} \quad (1)$$

また,  $C(u)$  の中の最適なラベル  $l$  は以下の通りに定義する.

$$\bar{l} \in C(u) \quad \bar{l} = \operatorname{argmax}\{\omega(L(l)) \mid l \in C(u)\} \quad (2)$$

このように, 頂点  $u$  を根とした最適なラベルを求めていき, 最後は  $\rho$  を根とした  $T$  の最適ラベリング  $L(\rho)$  を出力する.

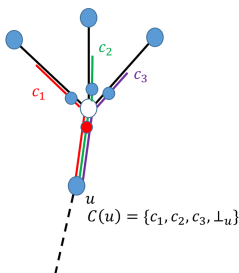


図 7. 正規頂点  $u$  のラベル候補

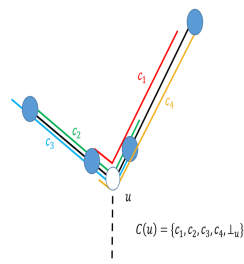


図 8. 交差点  $u$  のラベル候補

### 2.3 計算量

木  $T$  では, Chain Vertex などで追加された頂点を含めると,  $O(n^2)$  個の  $C(u)$  を考慮する.  $u$  が交差点である場合,  $C(u)$  には  $O(n^2)$  個のラベルが含まれることがある. 各ラベルは, 幅優先探索を用いて  $O(n)$  時間で構築することができる. したがって, 各頂点  $u$  について, OptCandidate の計算時間は  $O(n^3)$  となり, 合計で  $O(n^5)$  の実行時間となる. さらに,  $T$  の各頂点を根にしたラベル配置の記録を保存するために,  $O(n^2)$  の記憶域が必要である.

### 2.4 閉路を一つ含む道路グラフに対するラベル配置

Tree Map Labeling を, 閉路を一つだけ含むような道路グラフ及び道路グラフの部分グラフに適用しても, 最適なラベル配置が可能である. そのような道路グラフに適用すると, 三つのパターンがある.

- (1) 閉路内の道路区間をカバーするラベルが存在しない.
- (2) 閉路内の道路区間のみをカバーしているラベル  $l$  が存在する.
- (3) 閉路内の道路区間と閉路外の道路区間をカバーするラベル  $l$  が存在する.

(1) の場合, 道路グラフ及び道路の部分グラフから閉路の部分を除き, 閉路を木構造のグラフに変形させ, Tree Map Labeling を適用させる. (2), (3) の場合, ラベル  $l$  によってカバーされていない閉路内の道路区間を部分グラフとして分割し, その部分グラフに対し Tree Map Labeling を適用させる. どちらの場合も, 閉路内に交差点が存在するとき, そのような交差点を通過するようなラベルが配置されている場合, 閉路からその交差点と隣接する辺を閉路から削除する.

## 3 グラフから閉路を除くヒューリスティクス

[2] の手法は, 道路グラフが木構造の場合最適なラベル配置ができる. しかし, 世界の主要な都市の道路グラフでは, 前述したラベルの配置が明白な道路区間の隔離及び分解を用いても, 複数の閉路を含む部分グラフが残る可能性がある. そこで本論文では, 複数の閉路を含む部分グラフを木構造になるように, 道路区間を取り除くことで, [2] のアルゴリズムを適用可能にする.

### 3.1 道路区間の短い方を隔離するヒューリスティクス

道路グラフにおいて, 根から幅優先探索を行い, 探索中に既に調査済みに頂点に到着したとき, それは閉路が生じている事になる. このとき最後に辿った道路区間を  $r_1$ , 同じく過去にその点に到着した道路区間を  $r_2$  とし, 長さが長い方の道路区間を排除する. 以上を実現するために, Algorithm 1 を用いる.

### 3.2 道路区間のカバー数の多い方をカバーするヒューリスティクス

3.1 のヒューリスティクスを用いる場合, 閉路を形成する 2 つの道路区間の内, 長い道路区間を排除し, 全ての点を探査した時点で終了していた. 本ヒューリスティクスは, 探索中に閉路を形成したとき, 最後に辿った道路区間を  $r_1$ , 同じく過去にその点に到着した道路



**Algorithm 1** 道路区間の短い方を隔離するヒューリスティックを用いた部分グラフ  $T'$  の部分木を出力するアルゴリズム

```

1:  $Q \leftarrow$  空のキュー
2:  $v_s$  にチェックを入れる
3:  $Q \leftarrow v_s$ 
4: while  $Q$  が空になるまで
5:    $v \leftarrow$  キューから取り出す
6:   foreach  $v$  に隣接している頂点  $v_i$  do
7:     if  $i$  にチェックが付いてない場合 then
8:       点  $v_i$  にチェックを入れる
9:        $e \leftarrow v$  と  $v_i$  を結ぶ辺
10:       $e$  にチェックを入れる
11:     else
12:        $e_a \leftarrow v$  と  $v_i$  を結ぶ辺
13:        $e_b \leftarrow$  点  $i$  を端点として持ち, なおかつ削除
         されていない辺
14:       if  $e_a > e_b$  then
15:          $e_a$  を部分グラフ  $T'$  から削除
16:       else
17:          $e_b$  を部分グラフ  $T'$  から削除
18:       end if
19:     end if
20: end while

```

区間を  $r_2$  とし, それぞれの道路区間を除いたときの木に対して Tree Map Labeling を用いてラベル配置を行い, カバーしている道路区間数の多い方を選択し, すべての道路区間が探索済みになるまでこれを続ける.

### 3.3 複数の連結した同じ道路区間を隔離するヒューリスティクス

3.1, 3.2 のヒューリスティクスでは, 閉路を形成している2つの道路区間の一方のみを隔離している. 本ヒューリスティクスは, 最初のヒューリスティクスの考え方を基に, 一度に複数の道路区間を隔離する. 根から幅優先探索を行い, 閉路を形成したとき, 閉路探索済みの道路区間の中で,  $r_1, r_2$  と隣接している同じ道路名の道路区間がある場合, それらを連結させ, 経路  $k_1, k_2$  にする. その後, 2つのそれぞれ経路の合計の長さを比較し, 長いほうの経路を隔離する. 隔離した各経路は, 部分木としてラベル配置を行う.

## 4 計算機実験

これらの既存手法と提案したヒューリスティクスを実装し, 計算機実験を行った. 道路地図は, 昭文社の MAPPLE10000 より後楽園周辺のデータを取得し, 道路区間は563個存在した. 3.1, 3.2 のヒューリスティクスを用いてカバーできた道路区間は554個であり, 3.3 のヒューリスティクスを用いてカバーできた道路区間は555個であった. 図9は3.3のヒューリスティクスを用いて道路名のラベル配置を行った結果であり, 黒い線は道路区間, 赤い線は配置されたラベルを表している.

## 5 結論と今後の課題

本論文では, 閉路が複数存在する道路グラフに対して, 木構造になるように, 道路区間を取り除くことで, [2] のアルゴリズムを適用可能にするようなヒューリス

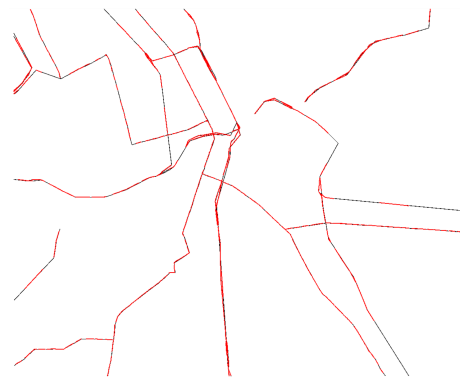


図9. 3.3のヒューリスティクスを用いたラベル配置

ティクスを提案し, 実際の地図データを用いて実験を行った. 3.1, 3.2で提案したヒューリスティクスでは, 道路グラフを木構造にすることができ, 分解された道路グラフに対し, Tree Map Labeling を適用することができた. 3.3で提案したヒューリスティクスでは, 前述した二つのヒューリスティクスより, わずかではあるがカバーする道路区間の多いラベル配置を見つけることができた.

今後の課題として, 追加する頂点数によって計算時間が大きく異なるため, 追加する頂点数を減らす手法を見つける必要がある. 今回の実験の過程で, 道路グラフ内で追加された頂点数は, 最初に入力した頂点数の約三倍に近い数が存在する. これは, 多くの交差点を持つ長い経路で Chain Vertex が生成されやすいことが原因だと考えられる. また, 提案した各ヒューリスティクスで閉路の道路区間の隔離を考えると, 未探索部分の道路区間を配慮していないことが原因で, カバーできる道路区間が減る可能性がある. それらを考慮できるようなヒューリスティクスを提案できれば, さらに良いラベル配置ができる.

## 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] F. Chirié, “Automated name placement with high cartographic quality: City street maps,” *Cartography and Geo. Inf. Science*, Vol. 27(2), pp. 101–110, 2000.
- [2] A. Gemsa, B. Niedermann, and M. Nöllenberg, “Label Placement in Road Maps,” *30th European Workshop on Computational Geometry*, 2014.
- [3] M. Vaaraniemi, M. Treib, and R. Westermann, “Temporally coherent real-time labeling of dynamic scenes” *Proc. 3rd Int. Conf. Comput. Geospatial Research Appl.*, COM.Geo’12, pp. 1–17, 2012.