アルゴリズム 79—9
（2001. 7. 27）

引出し線を用いたラベル配置問題

大塚 善仁* 今井 桂子†
*中央大学大学院 理工学研究科 情報工学専攻
†中央大学 理工学部 情報工学科

概要 平面上に n 個の点集合 V が与えられているとき，これらの点に対しラベルを配置する問題を NLP(Node Label Placement) 問題という．ここでは配置できるラベルの数が最大となるような問題を考える．一般にこの最適化問題は NP 困難であり，いくつかのヒューリスティックスや近似アルゴリズムが研究されている．NLP 問題に対してはこれまでに固定位置モデルとスライダーモデルが考えられているが，これらのモデルはどちらも点とラベルが接触していなければならないものである．本稿では点とラベルが引出し線によって結ばれているモデルを提案し，その解法と実験結果を示す．

Node Label Placement Problems with Leader Lines

Yoshihito OHTSUKA* Keiko IMAI†
*Information and System Engineering Course,
Graduate School of Science and Engineering, Chuo University
†Department of Information System and Engineering, Chuo University

Abstract In this paper, we consider the node label placement (NLP) problem. Given a set V of n points in the plane, the problem is maximizing the number of labeled points. For this problem, there are two models, the fixed position model and slider model, and each label touches the corresponding point in both models. In general, this optimization problem is NP-hard, and heuristics and approximation algorithms have been investigated. We give a new approach to the NLP problem in this paper. In our model, each label connects with the corresponding point by means of a leader line. We propose some algorithms for the NLP problem with leader lines, and experimental results are also shown.

1 はじめに

地図, グラフなどにおいて, 描かれている物にラベルを配置する問題をラベル配置問題という．例えば, この問題は地理情報システムの分野で, デジタル化された地図へ自動的にラベルを配置するなどの目的で研究されている．ラベル配置問題はその用途やラベルを配置する対象によって多くの種類が与えられ, 現在までにさまざまな研究が行われてきているが, それらの多くは NP 困難であることが証明されている [5].

ラベルは適切な位置に配置しなければならず, ラベル配置問題を考えるときは以下の制約を満たすようなラベルの配置位置を求めることを目標とする．

- ラベルは他のラベルやラベルを配置する対象と重なってはいけない．
- ラベルはどの対象物を指しているのか明白でなければならない．

ラベルを配置する対象は, 領域, 点, 辺の 3 つに分類される．このうち, 点に対する問題は NLP (Node Label Placement) 問題と呼ばれている．本研究ではこの NLP 問題を主に考える．

NLP 問題では, ラベルを表す長方形のどこと点を一致させるかによって固定位置モデルとスライダーモデルに大きく分けられる．しかし, どちらのモデルも点とラベルが接しているために, 点が密集している場合にはどうしてもラベルを配置できないことがある．このようなとき, 実際の地図ではしばしば線を用いて近くの空白へラベルを引

き出しているという例が見られる. この線のことを地図用語で引出し線 (Leader Line) という.

図 1 は実際の上下水道, ガス, 電気, NTT などの工事計画に関する情報を表す地図であり, 引出し線を用いないと配置できないような例である.

本稿ではこの引出し線をラベル配置問題へ導入することを考える. バケット法に基づくアルゴリズムを提案し, 実際の地図などいくつかの例に対して計算機実験を行う.

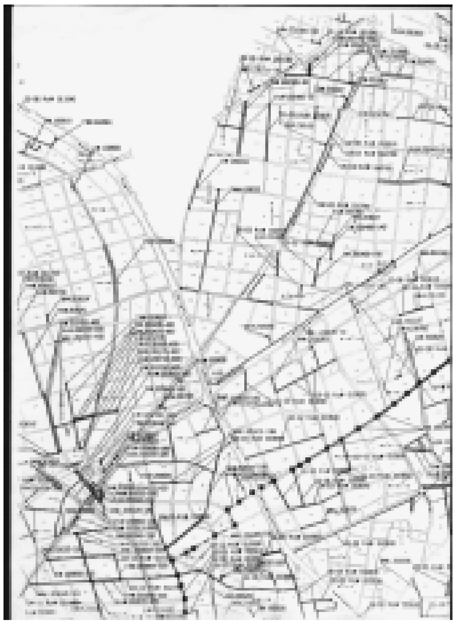


図 1. 工事地図

2 従来のモデル

NLP 問題はラベル配置問題の中でも基本的なものとして研究がなされている. ここではこれまでに研究されてきた一般的な 2 つのモデルについて述べる.

まず, 離散的なモデルとして固定位置モデル (Fixed-position Model) がある. これは長方形の境界上にある有限個の点集合の中の 1 点とラベルを配置したい点とが一致するように長方形を平行移動させるものである. 図 2 はラベルの候補位置が 4 つなので 4-position model と呼ばれる. この 4-position model では全てのラベルが正方形かつ大きさが同じときでさえ, 重ならないように配置できるかどうかを判定する問題は NP 完全であることが証明されている [1].

もう一方のモデルがスライダーモデル (Slider Model) である. これはラベルを配置したい点を長方形のいくつかの辺上の任意の点と一致させるものである. 図 3 は点のスライドを許す辺の数が 4 本なので 4-slider model と呼ばれる. スライダーモデルのラベル数最大化問題は NP 困難であることが証明されている [4]. また, このモデルでは少なくとも最適配置の半分のラベル数が得られる $O(n \log n)$ 時間の近似アルゴリズムが知られている [4].

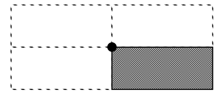


図 2. 4-position model

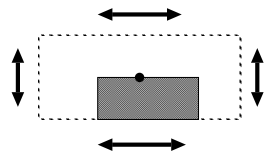


図 3. 4-slider model

3 引出し線の導入

2 節で説明したように従来のモデルでは点が長方形の境界上になくてはいけないので, 点が密集している場合には配置できるラベルの数が限られてしまうことがある. そこで, 引出し線を用いればより多くのラベルが配置できるのではないかと考えて, ラベル配置問題に引出し線を導入することを提案する.

“引出し線を用いてラベルを配置する” とは, 点とラベルを直線分で結びつけるということとする. その結びつけ方は点とラベルの左の辺の中点とを結ぶことにする. ただし, 引出し線がそのラベル自身と交差してしまうようであれば, その交差している部分を消去する (図 4).

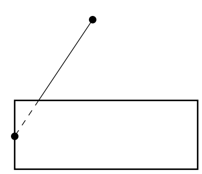


図 4. 引出し線の引き方

さらに, ラベル配置の制約に次の 2 つを追加する.

– 引出し線は他のラベルや点と重なってはいけない.
– 引出し線は他の引出し線と交差してはいけない.

4 NLP 問題に対する解法

今, 有限な平面とそれ上の n 個の点集合 V が与えられているとする. 各点 $v_i \in V$ に配置されるラベルのサイズは, 幅 W_i, 高さ H_i であるとする. このようなとき, 前述の制約を満たすような配置の中で, 配置できるラベル数の最大化問題を考える. アルゴリズムの概要を以下に示す.

(1) 平面をバケット分割する.
(2) バケット内の点の数が少ないとき, ラベルはそのバケット内に配置する.
(3) バケット内の点の数が多いとき, 周囲に空いているバケットがあるかどうか探す.
もしあれば, 次のことを行う.
- 探してきたバケット内に離散的な候補位置を定める.
- 制約を満たすように点と候補位置とを対応させる.

以下で詳細を説明する.

4.1 バケット分割

まず, 平面の領域をバケットと呼ばれる同じ大きさの長方形に分割し, 各バケットごとに解を求めていく. ここではバケットの大きさを, 幅 $2W_{\max}$, 高さ $2H_{\max}$ とする. $W_{\max}, H_{\max}$ とは それぞれ W_i, H_i の最大値である. この大きさのバケットで平面を分割し, バケット内にある点の数を数える (図 5).

点の数が 1 つのとき, ラベルがバケットの外に出ないように配置する. このとき, 配置するモデルは 4-slider model とする. バケットの大きさが $2W_{\max} \times 2H_{\max}$ であるので, この配置が可能なことは明らかである. ただし, 後に述べる探索によってこのバケットが確保された場合にはこのような配置は行わない.

点の数が 2 つのとき, 引出し線を用いてそのバケット内にラベルを配置する. やはりバケットの大きさにより, この配置は可能である.

バケットが $k(>2)$ 個の点を持っていたとき, 隣接しているバケットの点の数を見て空いているかどうかを調べる. 空いていなければさらにその隣のバケット, というように幅優先で探索していく. 図 6 がその例である. 図中の数字はそのバケット内にある点の数を示している. 3 つと 4 つの点を持つバケットがあるので, それぞれ周囲の空きバケットを探し出している.

バケット内には 2 つまでの点が配置できるとしているので, 点を持たないバケットには 2 つ, 点を 1 つ持つバケットには 1 つの点を配置させられる余裕があると考えて, k 個の点をすべて任せられる数の空きバケットを確保しておく. そして, 次のステップへ進む.

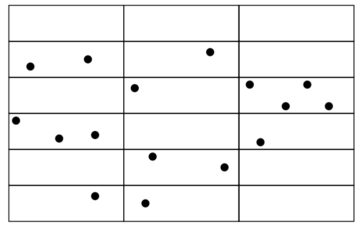


図 5. バケット分割

0	0	0
2	1	0
0	1	4
3	0	1
0	2	0
1	1	0

図 6. 空いているバケットの探索

4.2 マッチング

$k(>2)$ 個の点をもつバケットに対して, 探索で得られたバケットの集合を B_j, それらのバケット内にある点の集合を $V_{B_j} \subset V$ とする. 3 個以上の点を持つバケットが m 個あった場合, $j = 1, \ldots, m$ である. B_j の各バケット内に, 図 7 のように 4 つの離散的なラベル候補位置を設定する (ただし, その候補位置と V_{B_j} の点が重なってしまう場合は除く). この候補位置の集合を Λ_j とする. B_j において制約を満たすような Λ_j と V_{B_j} とのマッチングを求め, これをすべての j について行うことによってラベルを配置する.

ここではこのマッチングをバックトラックで総当たり的に求める. 距離が近いものから組合わせていき, 制約を満たすような解が得られた時点で終了する. 最悪の場合は全ての組合せを試したこ

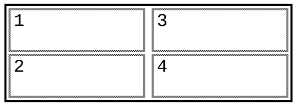


図 7. バケット内の候補位置

とと同じなので効率は悪いが, 比較的速い時間でそれなりによい解が得られることが実験でわかっている [2]. 密集の度合が低いときに限れば有効であると考えられる.

また, 引出し線の長さの最小化を最適解とすれば, 以下のような問題を考えることができる.

引出し線の総長和最小化問題

$v \in V_{B_j}$ に Λ_j の中から候補位置を割り当てる関数を $\lambda : V_{B_j} \to \Lambda_j$ とする. また, v に割り当てられた候補位置 $\lambda(v)$ と v を結んだ際の引出し線の長さを $length(v, \lambda(v))$ と表すことにする.

ここで, $v \in V_{B_j}$ と $l \in \Lambda_j$ に対し, $x_{vl}, y_{vl} \in \{0, 1\}$ という変数を用意する. x_{vl} は v に配置されるラベル $\lambda(v)$ が $l \in \Lambda_j$ だったとき 1, そうでなければ 0 となる変数であり, y_{vl} は v と $\lambda(v)$ を結んだ引出し線が $l \in \Lambda_j$ 上にあったとき 1, そうでなければ 0 となる変数である.

以上のようにして V_{B_j} と Λ_j のマッチングを整数計画問題として定式化する.

$$\min_{\lambda} \quad \sum_{v \in V_{B_j}} \sum_{l \in \Lambda_j} length(v, \lambda(v)) \; x_{vl}$$

$$\text{s.t.} \quad \sum_{l \in \Lambda_j} x_{vl} = 1 \qquad {}^{\forall} v \in V_{B_j}$$

$$\sum_{v \in V_{B_j}} x_{vl} \le 1 \qquad {}^{\forall} l \in \Lambda_j$$

$$\sum_{l \in \Lambda_j} x_{vl} + \sum_{v' \in V_{B_j}} \sum_{l \in \Lambda_j} y_{v'l} \le 1 \qquad {}^{\forall} v \in V_{B_j}$$

4.3 後処理

以上のアルゴリズムでラベルを配置すると, 図 7 のような離散的な候補位置しか用意していないため整然とラベルが並んでしまい, 人間の目には見やすく映らないことがある. これを改善するために後処理を行う.

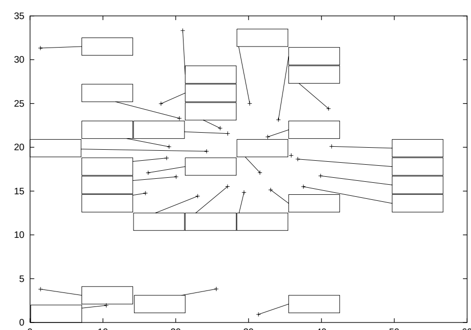
今, 仮に配置されているラベルは幅 $W_{\max}$, 高さ $H_{\max}$ であるので, それぞれ元のラベルのサイズ W_i, H_i に戻す. 仮に配置されているラベルのサイズは最大のものなので, この操作によって新しくラベルの重なりが起こってしまうことはない.

ラベルはそのラベルが配置されている点の方向に縮めることにする. これにより引出し線の長さが長くなることはない.

5 計算機実験

前節で述べたアルゴリズムについて計算機実験を行った. 比較の対象として [4] を元にした 1-slider model の近似アルゴリズムを用いた. 3 つのデータについての結果を示す. なお, 点と候補位置とのマッチングはバックトラックで総当たり的に組合わせる方法で行った.

1 つめの実験は人の手で点を適当に密集させたデータで行った. 1-slider model では 30 点中 18 点にしか配置できなかったが, 引出し線を用いることによって 29 点に配置することができた. 図 8 がその結果である.



ばならない点の数 $|V_{B_j}|$ の最大数は 6 であった.

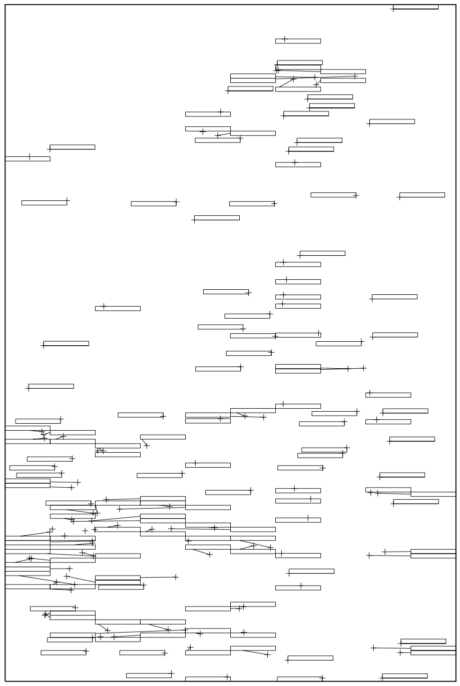


図 9. 工事地図に対する実験

3 つめの例は昭文社の MAPLLE 2500 のデータを用いた. 中央大学理工学部キャンパス周辺の地図を (株) インフォマティクスのソフト SIS で取り出し, 座標とその場所の名前を入力として与えた. 1-slider model では 23 点中 17 点にしか配置できなかったが, 引出し線を用いることによってすべての点に配置することができた. 図 10, 11 がその結果である.

この例では後処理を行ってラベルを文字数に応じたサイズにした. いくつかのフォントサイズで実験をしたが, 図 10 のサイズが一番バランスのよい結果であった. 1 つのバケットに入る点の最大数は 4 であり, $|V_{B_j}|$ の最大数も 4 であった.

6 おわりに

ラベル配置問題に対して, これまでに研究されていない引出し線を用いて解く手法を研究した. NLP 問題についてバケット法に基づくアルゴリズムを提案し, 計算機実験を行った結果, 1-slider model の近似アルゴリズムよりもよい結果が得られた.

しかし, 図 11 を見てもわかるように, もう少し引出し線を短くできる部分がある. マッチングの際に組合わせていく順番にもよるが, ラベルを点の近くに移動させるという後処理によっても改善されると思われる.

また, 今回のアルゴリズムは最初のバケットの分割の仕方によって大きく結果が異なってしまうという欠点がある. 1 つのバケットに入る点の数がなるべく小さくなるような分割をすることができればさらによい結果が得られると考えられる. 現在, バケット法以外で点の密集を考慮しながら平面を分割する手法を検討中である.

今後の課題として, 実験結果を 1-slider model の近似アルゴリズムと比較するのではなく, それよりも多くのラベルが配置できる 4-slider model の最適解を求めるアルゴリズム [3] との比較, また, 4.2 節で定式化した整数計画問題を解いて得られた最適解との比較をすることなどが挙げられる.

参考文献

[1] M. Formann and F. Wagner: A Packing Problem with Applications to Lettering of Maps, *Proc. 7th ACM Symp. on Computational Geometry,* pp.281-290, 1991.

[2] 今井 桂子, 亀田 貴之, 大塚 善仁, 佐竹 直也: 地図におけるラベル配置問題 – 点と辺に対する解法の実験的評価, 第 7 回 "統合型地理情報システム" シンポジウム予稿集, pp.23–30, 2001.

[3] G. W. Klau and P. Mutzel: Optimal labelling of point features in the slider model, *Proc. 6th Annual International Computing and Combinatorics Conference (COCOON'00)*, Lecture Notes in Computer Science 1858, pp. 340-350, 2000.

[4] M. van Kreveld, T. Strijk and A. Wolff: Point Set Labeling with Sliding Labels, *Proc. 14th Annu. ACM Symp. on Computational Geometry,* pp.337-346, 1998.

[5] A. Wolff: A Map Labeling Bibliography, 2000. `http://www.math-inf.uni-greifswald.de/map-labeling/bibliography/`

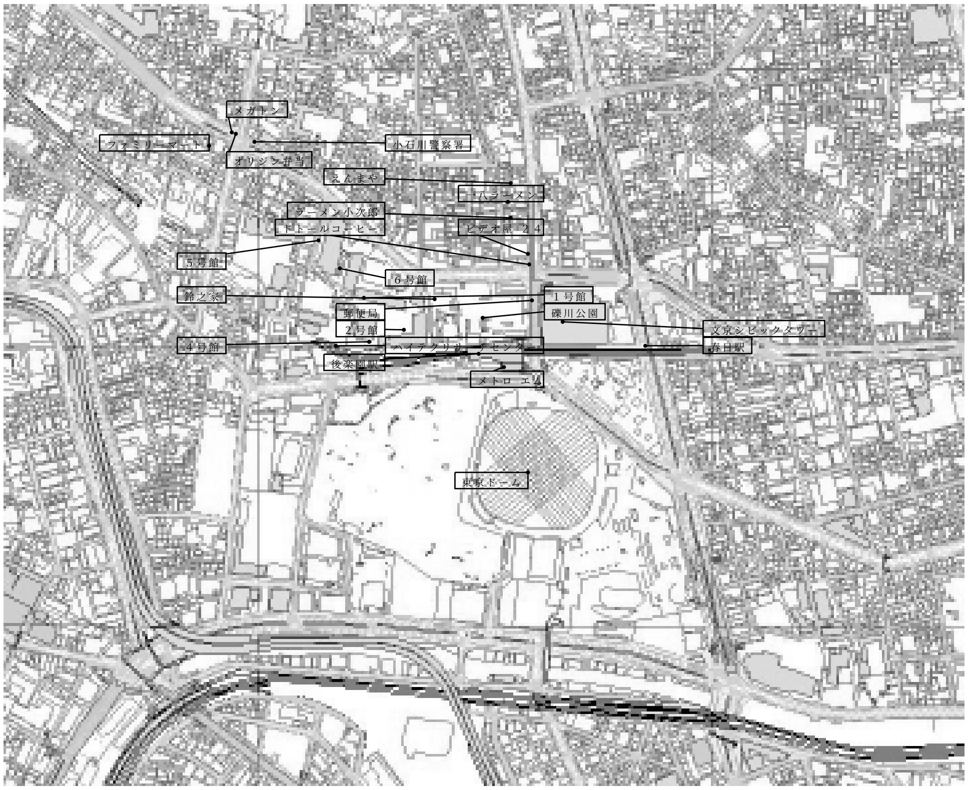


図 10.　中央大学理工学部周辺の地図

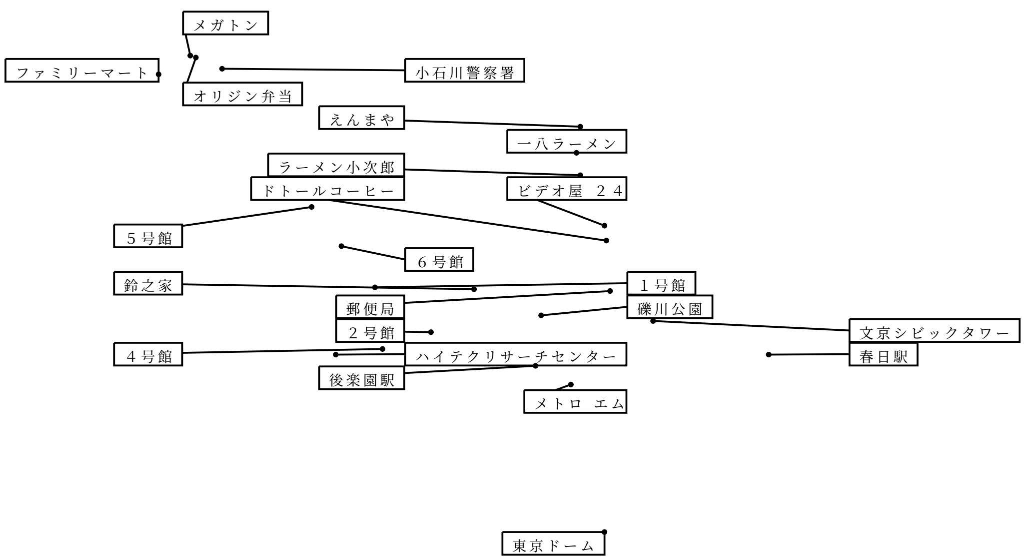


図 11.　中央大学理工学部周辺の地図 (ラベルのみ)