

A-12

# 左順序付き柔軟ラベリングの実装 Implementation of Left-part Ordered Flexible Labeling

小池 敦\*  
Atsushi Koike

徳山 豪\*  
Takeshi Tokuyama

## 1 はじめに

地理情報システムではシステムを使うユーザの要求により地図上の指定された標的の集合にラベル(文字情報)を配置する必要がある。この問題はラベル配置問題もしくはマップラベリング問題と呼ばれ、近年、地理情報処理の重要問題として数多くの研究が行われている [4]。標的としては地図上の様々な形状が考えられるが、本論文では各標的が地図上の点である場合に、長方形のラベルを下記の条件を満たすように配置する問題を考える。(1) ラベルは対応する標的の近くに配置する (2) ラベル同士が重ならない (3) 障害物と重ならない。実際には、要請 (1) を数学的に定式化するため、各標的の点に対しラベルの候補の集合を定める。

ラベル候補集合が次節で説明する left-part ordered set であるときのラベル配置問題は Left-part Ordered Flexible Labeling (LOFL) と呼ばれ、多項式時間で解ける数少ないラベル配置問題の定式化の一つである [2]。本論文ではまず、LOFL の解法の red-black tree を用いた高速実装を行い、計算時間の大幅な改善を行った。

しかしながら、実用的なラベル配置問題では、より自由度の高いラベル配置問題に定式化し、それをヒューリスティクスを用いて高速に解く必要がある [3]。そこで、LOFL を二つ組み合わせた two-position LOFL という新しいラベル配置問題の定式化を与え、上記 LOFL のアルゴリズムを用いた高速ヒューリスティクスの設計を行い、出力の品質をラベルのサイズにより評価した。その結果、複数の LOFL を組み合わせることにより、解の品質が向上することが示された。

## 2 LOFL の定式化とアルゴリズムの概略

LOFL はラベル候補集合が left-part ordered set である時のラベル配置問題である。ラベル  $l$  の left part  $l^-$  を

$l^- = \{(x, y) | (x, y) \in l \text{ かつ } x \leq (l \text{ の標的点の } x \text{ 座標})\}$  で定義する。  $l_1 \leq l_2 \Leftrightarrow l_1^- \subseteq l_2^-$  でラベル間の半順序を定義し、このとき  $l_1$  は  $l_2$  より left small であると呼ぶ。ラベル候補集合  $\mathcal{L}$  が条件  $\forall l_1, l_2 \in \mathcal{L}, l_1 \leq l_2 \text{ または } l_1 \geq l_2$  を満たす時、left-part ordered set であると言う。図 1 は left-part ordered set の例である。各ラベルは共通のスケール係数  $\sigma$

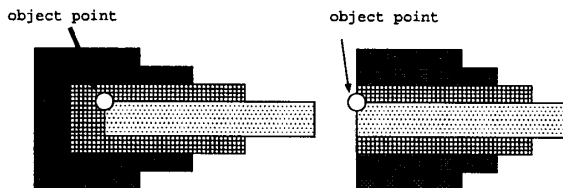


図 1: left-part ordered set の例

\*東北大学大学院情報科学研究科  
{koike,tokuyama}@dais.is.tohoku.ac.jp

で拡大/縮小され、配置される。目的は、全ての標的の点にラベル配置可能であるスケール係数  $\sigma$  の最大化であり、下記の判定問題を用いて、二分探索(理論的にはパラメトリック探索も可能)で  $\sigma$  を最大化する。

判定問題の入力:  $n$  個の標的の点集合  $P = \{p_1, p_2, \dots, p_n\}$ , “left-part ordered set であるようなラベル候補集合” の集合  $\{\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_n\}$ , ラベルのサイズ  $\sigma$ , 合計  $m$  個の辺からなる多角形障害物の集合

出力: ラベル配置可能なら yes, そうでないなら no

## 3 判定問題に対するアルゴリズム

すべてのラベル候補集合が left-part ordered set であるときの判定問題は  $O((n+m) \log(n+m))$  時間で解ける [2]。障害物がない場合のアルゴリズムを示す。標的の点集合  $P = \{p_1, p_2, \dots, p_n\}$  は標的の点の  $x$  座標を基準として降順にソートされているとする。  $p_1$  は最右の点であり、  $p_n$  は最左の点である。

### Algorithm DECIDE

```

1. for  $i \leftarrow 1$  to  $n$ 
2.   do
3.     if  $\mathcal{L}_i$  のすべてのラベル候補が
       既に配置されたラベルと交わる
4.       then return “ラベル配置不可能”
5.     else 既に配置されたラベルと
       交わらないラベルのうち
       最も left small なラベル  $l_i$  を
        $p_i$  に割り当てる
6.   fi
7. end
8. return  $L = (l_1, l_2, \dots, l_n)$  が可能なラベル配置

```

アルゴリズムの計算時間を考察するため、5 行目について詳しく述べる。  $p_1$  から  $p_{i-1}$  までのラベルが配置されている領域  $U$  を  $U = \bigcup_{k=1}^{i-1} U_k$  とし、このとき frontier  $F$  を以下のように定義する。

$$F = \{(x, y) \in U \mid (x_0, y) \in U, x \leq x_0\}$$

即ち、frontier は  $U$  を  $x = (p_i \text{ の } x \text{ 座標})$  から見たとき ( $p_1$  から  $p_{i-1}$  はすべて  $p_i$  の右側にある), 見ることができる点の集合ということになる。従って、既に配置されたラベルとの交差判定のためには、frontier との交差判定を行えばよいことになる。

frontier のデータ構造に red-black tree を用いると各標的の点ごとの交差判定に  $O(\log n)$  時間、frontier の更新に  $O(\log n)$  時間かかる。更新の回数は  $O(n)$  であることが示せるため、計算時間は  $O(n \log n)$  時間となる。障害物がある場合も同様に考えることができ  $O((n+m) \log(n+m))$  時間で解くことができる。

#### 4 LOFL を用いたヒューリスティクス

LOFL 単体では十分な大きさのラベルを配置できない場合もあり、必ずしも他の既存モデルよりも優れているとは言い切れない。そこで、LOFL を広げたラベル配置問題の定式化を考え、LOFL のアルゴリズムを用いて解くアプローチを考える。本論文では 2 種類の LOFL を組み合わせた two-position LOFL を扱う。

Two-position LOFL でのラベル候補集合は次の二つのラベル候補集合の和集合である。まず一つのラベル候補集合  $L$  は left-part ordered set で、集合に属するすべてのラベルについて左上に標的点がくるものを考える。二つ目のラベル候補集合  $M$  は  $L$  のそれぞれの要素にたいし標的点が左下にくるようにしたものである。また、 $l_j \in L$  に対応する (標的点の位置だけが異なる)  $M$  でのラベルを  $m_j$  とする。このとき  $L + M$  は left-part ordered set にならないが  $L$  と  $M$  のそれぞれは left-part ordered set になっている。

各  $j$  について  $L$  からラベル  $l_j$  を選ぶと、各標的点  $p_j$  にラベル候補集合  $\{l_j, m_j\}$  を与えたものは Forman らにより提唱された two-position model の一例となっている [1]。これは 90 度回転すると、LOFL の特殊例と考える事ができる。

残念な事に、two-position LOFL でのラベル配置問題は NP 完全問題になる。従って、two-position と LOFL を以下のように組み合わせてヒューリスティクス解法を設計する。まず  $L$  で LOFL の最適化問題を解く。次に各標的点での求めたラベル  $l_i (i = 1, 2, \dots, n)$  に対し  $l_i$  と  $m_i$  ラベル候補とし、two-position model での最適化問題を (これも LOFL のアルゴリズムを用いて) 解く。そこで  $l_i$  が選択された標的点は次は  $L$  をラベル候補集合として、 $m_i$  が選択された標的点は次は  $M$  をラベル候補集合として、LOFL の最適化問題を解く。これをラベルサイズが増加しなくなるまで (もしくは指定された回数) 繰り返す。この解法は、少なくとも LOFL および two-position より良い解を生成する事を保証し、LOFL の高速アルゴリズムを用いるため、非常に高速なラベル生成を行う。

#### 5 実験

実験では以下の 4 通りのモデルの比較を行った。ラベルサイズ (スケール係数) を  $\sigma$  として (1) Fixed model: 高さ  $3\sigma$ , 幅  $4\sigma$  でラベルの左上に標的点がくるラベル (2) Two-position model: 高さ  $3\sigma$ , 幅  $4\sigma$  でラベルの左上に標的点がくるラベルと左下に標的点がくるラベルの 2 つ (3) LOFL: ラベルの左上に標的点がくる以下の 6 種類のラベル (高さ  $1\sigma$ , 幅  $12\sigma$ ), (高さ  $2\sigma$ , 幅  $6\sigma$ ), (高さ  $3\sigma$ , 幅  $4\sigma$ ), (高さ  $4\sigma$ , 幅  $3\sigma$ ), (高さ  $6\sigma$ , 幅  $2\sigma$ ), (高さ  $12\sigma$ , 幅  $1\sigma$ ) (4) Two-position LOFL: (3) の各ラベルにおいてラベルの左下に標的点がくるものを追加した 12 種類のラベル。

各実験において  $50000 \times 50000$  の障害物の無い平面上にランダムに  $n$  個の標的点を置く。  $n$  としては 20, 40, 60, 80, 100, 200, 400, 1600 を採った。

(1)(2)(3) は LOFL のアルゴリズムを用いて解を求め、(4) には 4 節で示したアルゴリズムを用いて、反復回数は 6 回とした。また、3 節で述べたように frontier のデータ構造には red-black tree を用いた。

図 2 は 100 回実験を行ったときのラベルサイズの平均値である。Two-position model と LOFL では  $n$  が小さいときは同等なサイズであるが  $n$  が大きいと two-position model の方が大きいサイズになることが判る。Two-position LOFL は two-position model や LOFL に比べ、 $n$  に関わらず 1.5

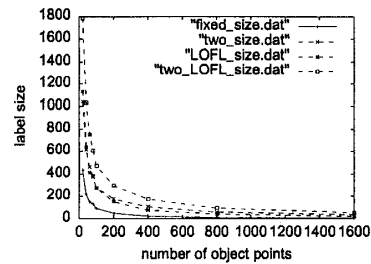


図 2: 各モデルにおけるラベルサイズの平均

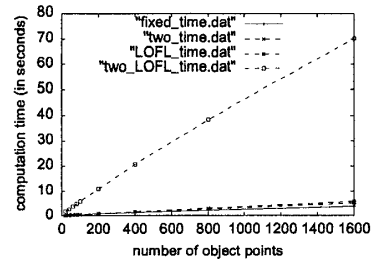


図 3: 100 回の合計計算時間 (秒)

倍以上の大きさのラベルが配置され、2 つの方法の組み合わせは有効であるといえる。

図 3 は 100 回実験を行ったときの計算時間の合計値である。どのモデルに関しても sub linear な曲線になっており、理論値より良いが、これは frontier のサイズが平均的には  $n$  より小さい事が影響している。なお、frontier のデータ構造に単純な list を用いた場合は  $n = 1600$  の時、LOFL で 299.92 秒、two-position LOFL で 2093.68 秒であり、red-black tree を用いることにより、大幅に高速化されることが判る。

#### 6 まとめ

実際のラベル配置問題においてはヒューリスティクスをもちいてより品質のよいラベル配置を見つけることが必要になるが、ラベルサイズで評価した場合、LOFL は強力な道具になることが言えた。また、red-black tree を用いて高速実装する事により計算時間が大きく向上した。今後はスライダーモデルなどを含め、LOFL のより高度な組合せの実験も行う予定である。

#### 参考文献

- [1] M. Formann and F. Wagner, A packing problem with applications to lettering of maps, *Proc. 7th ACM SOCG* (1991), pp. 281-290.
- [2] A. Koike, S. Nakano, T. Nishizeki, T. Tokuyama, and S. Watanabe, Labeling Points with Rectangles of Various Shapes, *Int. J. of Comp. Geom. & Appl.*, to appear.
- [3] F. Wagner and A. Wolff, A practical map labeling heuristics algorithm *Comp. Geom. Theory and Appl.* 7 (1997), pp. 387-404.
- [4] <http://liinwww.ira.uka.de/bibliography/Theory/map.labeling.html>