移動する点に対するラベル配置問題

Label Placement Problem for Moving Points

情報工学専攻 清家 陽佑
SEIKE Yosuke

概 要

地図やグラフなどの図において, 点に対して適切な位置, 大きさの注記を配置する問題を点ラベル配置問題という. カーナビゲーションシステムや航空管制における航空路監視レーダーのシステムなどでは, 移動する点に対してラベル配置を行なう必要がある. このような状況では, ラベルの大きさが変わったり, 配置するラベルの数を減らしたりすることは避けたい. また, 注釈の読みやすさの観点から, ラベル同士がなるべく重ならないことに加え, ラベルの移動はなめらかであることが望ましい. さらに, リアルタイムにラベルの位置を更新する必要があるため, 高速な手法が必要である. そこで本研究では, これらを考慮し, 問題設定を行ない, その問題に対する局所探索を用いたアルゴリズムを複数提案する. そして, それらを計算機実験によって比較する. その結果, 提案した GRASP が多くの場合で有効であることがわかった.

キーワード： ラベル配置問題, 移動する点, 局所探索

1 序論

地図作成, 地理情報システム (GIS) などにおいて, 描かれた内容の説明をするため, 文字や記号による注記が行なわれている. カーナビゲーションシステムや航空管制における航空路監視レーダーのシステムなどでは, 移動する対象に対して注記を行なっている. 注記には重要な情報が載せられるため, 注記の見やすさは重要である. ある注記が他の注記と重なると, 可読性は損なわれる. また, 注記対象が移動する場合, 注記がなめらかに移動すれば見やすいと考えられる.

地図やグラフなどの図において, 点に対して適切な位置, 大きさの注記を配置する問題を点ラベル配置問題という. 注記は水平線と垂直線からなる長方形領域の内部に書かれると考え, この長方形をラベルとする. 点ラベル配置問題には, 点とラベルが接する配置モデルと点に対してラベルを離れた所に配置するモデルが存在する. 点とラベルが接する配置モデルには, 離散的にラベル配置候補が定義される *Fixed-position* モデルと連続的に定義される *Slider* モデルがある. 点に対してラベルを離れた所に配置するモデルでは, 点とラベルを引き出し線で結ぶことで対応付けを明確にする. このモデルを引き出し線を用いたモデルと呼ぶ.

本研究では, Slider モデルと引き出し線を用いたモデルを用いる. Slider モデルにおいては, ラベル同士の重なった面積とラベルの移動距離を小さくするように配置することを考える. 引き出し線を用いたモデルでは, これらに加え, 引き出し線の長さ, 引き出し線同士の交差, 引き出し線とラベルの交差, 点とラベルの交差も考慮する. これら 2 つのモデルにおいて, 移動する点に対してラベルを配置する局所探索を用いたアルゴリズムを複数提案する. また, 計算機実験を行ない, 複数のアルゴリズムを比較する.

2 問題設定

点の移動は, 点が描画された静止画を高速に切り替え続けている現象を指しているといえる. この静止画は映像分野において, 動画におけるフレームと呼ばれる. 本研究では, 各フレームにおける点の移動に対して, 各フレームにおけるすべてのラベルの配置位置を決めることを考える.

移動する点に対するラベル配置問題において, ラベルの見やすさに関係することとしていくつか考慮すべきことがある. 以下のことを満たすとすべてのラベルが見やすいと考えられる.

- ラベルの大きさが一定以上で, かつ各フレームでラベルの大きさが変わらない.
- ラベルが突然, 消失しない.
- ラベル同士が重ならない.
- 点とラベルが重ならない.
- ラベルがなめらかに動く.

また, 引き出し線を用いたモデルにおいては, 以下も考慮する必要がある.

- 引き出し線の長さが短い.
- 引き出し線同士が交差しない.
- 引き出し線とラベルが交差しない.

これらをすべて満たすようにラベルを配置することは困難である. そのため, 何を必ず守るべき制約とするか, なるべく満たすべき制約とするかを決定することが必要である. これを決定するにあたり, ヨーロッパの航空管制における安全規則 [2] を参考にする. [2] では, すべての飛行機に対して常にラベルが配置され, 大きさは変えてはならないことが求められている. よって, すべてのフレームにおいて点に対してラベルが常に配置されることと, 点に対するラベルの大きさが一定以上で, かつ各フレームでラベルの大きさが変わらないことを必ず満たすべき制約として考える. これら 2 つ以外はなるべく満たすべき制約として考えることとする. なお, Slider モデルの場合, ラベル同士が重ならなければ, 点とラベルが重ならないため, 点とラベルの重なりを考慮しなくてよいとする. そして, なるべく満たすべき制約それぞれの重み付きの和を目的関数とし, 必ず守るべき制約を制約条件として考える.

Slider モデルにおける目的関数は, (重なったラベルの総面積) $+ \alpha \times$ (フレーム間のラベルの総移動距離) とし, 引き出し線を用いたモデルにおける目的関数は, (重なったラベルの総面積) $+ \alpha \times$ (フレーム間のラベルの総移動距離) $+ \beta \times$ (引き出し線の長さの総和) $+ \gamma \times$ (引き出し線同士の交差数) $+ \delta \times$ (引き出し線とラベルの交差数) $+ \epsilon \times$ (点とラベルの重なっている数) と

する．$\alpha, \beta, \gamma, \delta, \epsilon$ はそれぞれ非負実数の重みである．これらの重みを変更することで, なるべく満たすべき制約のうち, どれを優先するかを制御する．なお, 本研究では描画領域内にすべての点が存在するとし, ラベルがその描画領域からはみ出したら, その分も重なっている面積として考える．

移動する点に対するラベル配置問題は, 各フレームにおいてその時々でラベルの位置を決定していくことが求められる．すなわち, k フレーム目においては, $k-1$ フレーム目までのラベル配置はすでに決定している必要がある．これを考慮するため, 更新時間という概念を用いることとする．この更新時間以内に各フレームにおけるラベルの位置を決定しなければならないとする．また, 1 フレーム目のラベルの位置はすでに決定されているとし, 2 フレーム目以降のラベル配置を求めることとする．そして, k フレーム目のラベル配置を求める際, $k+1$ フレーム目以降の点の座標は未知であるとする．

本研究で扱う問題は, 点の座標の入力が与えられたのち, ラベル配置を求めることを繰り返す問題であるといえる．しかしながら, 各フレームにおける点の座標の入力がどのようになるのか完全に予想することは難しい．そこで本研究では, 1 つ 1 つのフレームごとにラベル配置を求める問題を考え, それを連続で解くことによって, すべてのフレームのラベル配置を求めることとする．

3　提案手法

局所探索法は, まず初期解が与えられ, 近傍に存在する解に更新する操作を, 終了条件を満たすまで繰り返す方法である．ある解 x の近傍を $N(x)$ とし, 実行可能解の集合を F と表す．そのとき, 近傍とは, ある実行可能解 $x \in F$ に少し変形を加えた解の集合 $N(x) \subseteq F$ のことをいう．局所探索法は初期解の算出方法, 近傍の定義, 解の更新条件, 終了条件を変更することによって様々なアルゴリズムが得られる．終了条件を近傍に存在する解の中で目的関数値が改善されるものがなくなったときとし, 解を更新する際, 近傍に存在する解の中で最も目的関数値が改善される解 (最良解) に更新するとした場合, 本研究では山登り法と呼ぶこととする．

本研究では, 局所探索法の一般化であると考えることができるメタヒューリスティクスを用いた手法を提案する．メタヒューリスティクスとは, 特定のアルゴリズムを指すのではなく, 様々なアルゴリズムの総称のことを指す．メタヒューリスティクスの中でも単純なアルゴリズムである多スタート局所探索法, [1, 3] でよい結果が知られている GRASP(greedy randomized adaptive search procedure), Simulated Annealing を提案する．

本研究でいうところの解は, すべての点に対する各ラベルの位置であり, 計算時間が更新時間を超えたとき終了させるという終了条件を与える．また, 提案する各手法においてラベルの重なった面積を頻繁に算出する必要がある．ラベルの重なった面積を高速に計算するため, 長方形交差問題という問題を用いる．長方形交差問題とは, 与えられた n 個の水平線と垂直線からなる長方形に対して, 共通部分 (交差) をもつ長方形の対を列挙する問題である．この問題は, n を長方形の数, K を共通部分をもつすべての長方形対の個数とすると, $O(n \log n + K)$ の実行時間, $O(n)$ の記憶領域で動作する McCreight のアルゴリズムが知られている [4]．K は最大でも $O(n^2)$ である．本研究ではこのアルゴリズムを用いて交差するラベルの対を列挙した上で, それぞれの交差している面積を計算することで, ラベルの重なった面積を計算する．

3.1　近傍の定義

近傍は実行可能解, すなわち必ず満たすべき制約をすべて満たす解の中から 1 つのラベルを選び, そのラベルを少し動かした実行可能解を, すべてのラベルに対して求めたものの集合とする．

Slider モデルにおいては, 少し動かすラベルの位置は, 元のラベルの配置箇所におけるラベルの左下の点を中心とする小さな円と, ラベルの左下の点の取りうる軌跡の交わる点を用いて算出する (図 1)．この小さな円の半径はラベルの幅, 高さの最小値より十分小さいものとする．図 1 の青い点が近傍における新たなラベルの左下の点である．

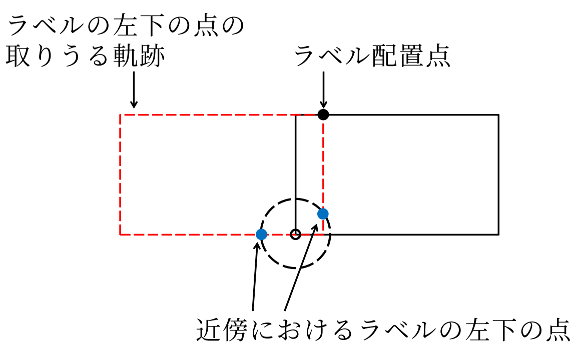

図 1. Slider モデルにおける近傍の定義

引き出し線を用いたモデルの場合はラベルの位置に制限がないため, 少し動かすラベルの位置はラベルに接するようにする必要はない．よって, 引き出し線を用いたモデルにおいては, 少し動かすラベルの位置は, 元のラベルの配置箇所に対して上下左右に微小に動かした 4 つを用いる (図 2)．

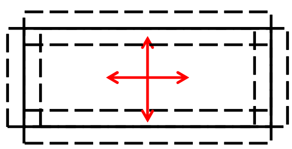

図 2. 引き出し線を用いたモデルにおける近傍の定義

3.2　山登り法

山登り法は近傍の最良解を解の更新に用いるとし, 終了条件を近傍に存在する解の中で解が改善されるものがなくなったときとするものである．本研究ではこの終了条件に加え, 計算時間が更新時間を超えたとき終了させるという終了条件を与える．目的関数を f とし, 解 x に対する目的関数値を $f(x)$ と示すこととする．初期解を x_{init} と表す．ただし, $x_{init} \in F$ とする．

山登り法は以下のように表すことができる.

山登り法

1. $l \leftarrow 1, x_1 \leftarrow x_{init}$ とする.
2. 計算時間が更新時間を超えない, または, $\{x \in N(x_l) | f(x) < f(x_l)\} = \phi$ でない限り, 以下の操作を繰り返す.
   (a) $\{x \in N(x_l) | f(x) < f(x_l)\}$ のうち, 最良解を x' とする.
   (b) $x_{l+1} \leftarrow x'$ とし, $l \leftarrow l+1$ とする.
3. x_l を出力する.

この山登り法を提案する他の手法においても利用する. なお, 山登り法を単独で用いる場合, 初期解は前フレームで得られた解から移動距離が最も小さい解とする.

3.3 多スタート局所探索法

山登り法を単独で用いる場合, 前フレームで得られた解から移動距離が最も小さい解を初期解とし, 局所最適解が 1 度得られた時点で終了するとしていた. 多スタート局所探索法は複数の初期解に対して山登り法を適用し, それぞれで局所最適解を求め, そのうちの最良解を結果として得るアルゴリズムである. 本研究では, まず前フレームで得られた解から移動距離が最も小さい解を初期解とし, 解を得る. その後, ランダムに初期解を生成し, 山登り法を用いて解を得ることを繰り返す. 各初期解から局所最適解が 1 つずつ得られるが, これらのうちの最良解を最終的な解とする.

3.4 GRASP

多スタート局所探索法はランダムに複数の初期解を生成していた. しかし, 初期解を生成する際に工夫を加え, ある程度目的関数値のよい解を初期解として生成し, その初期解を用いて山登り法を行なえば, さらによい解を得ることができる可能性がある. GRASP は多スタート局所探索法の一種であると考えられ, Greedy 法に確率的な要素を加えた手法によって初期解を生成し, その初期解を用いて山登り法を行なうことで解を得る手法である. Greedy 法に確率的な要素を加えた手法をランダム化 Greedy 法と呼ぶこととする. 多スタート局所探索法においてランダムなものとしていた初期解の生成方法をランダム化 Greedy 法に置き換えたものが GRASP である.

3.5 Simulated Annealing

これまで説明してきた手法はすべて近傍の最良解を解の更新に用いたため, 局所探索を行なうたびに必ず目的関数値が改善されるものであった. Simulated Annealing はこれらと異なり, 確率的に悪い解に更新することも許す. 悪い解に更新することを許すことで, 局所最適解から脱出する可能性がある. 確率的に悪い解に更新する操作は, 温度を表すパラメータ T によって制御される. T の値は新しい解に更新するごとに徐々に小さくなり, それによって悪い解に更新する確率が小さくなる. Simulated Annealing においても, 前フレームで得られた解から移動距離が最も小さい解から局所探索を行なうこととする.

3.6 Slider モデルの特徴を考慮した計算量の削減

Slider モデルは引き出し線を用いたモデルと異なり, 各ラベルの配置位置は点の周囲に限定されている. ゆえに, 点の位置とラベルの幅, 高さによっては, 明らかに他のラベルと重ならないようなラベルが存在することがある. 他のラベルと重なる可能性のないラベルは, このラベルを少し動かしたとしても, 重なる面積は減少しない. よって, 他のラベルと重なる可能性があるラベルのみに対して近傍を列挙することとする.

また, 近傍の要素のうち 1 つを選ぶことでラベルを移動させたのち, 新たに近傍を列挙する場合に, 重なったラベルの総面積を再計算する必要がある. この再計算の際, 重なったラベルの総面積に影響するものは, 移動したラベルと, そのラベルと重なる可能性のあるラベルとの重なった面積のみであるといえる.

これらのことを考慮するために, ラベルが配置されうる領域を長方形交差問題の長方形とみなすことによって, 重なる可能性のあるラベルの対を列挙する. このことで, Slider モデルにおいて計算量を削減する. なお, この計算量の削減方法は, Slider モデルにおいて点とラベルが接するという特徴を活かした方法で行なっているため, 引き出し線を用いたモデルでは利用することができないことに注意する.

4 計算機実験

提案したアルゴリズムを実装した上で, 計算機実験により評価を行なう. 計算機実験を行なう際の設定について述べる. まず, 描画領域は幅 800, 高さ 480 であるとする. 各ラベルの大きさは, 幅が 50 以上 100 以下, 高さが 20 以上 50 以下でランダムに決定する. 実験はすべて, 100 フレーム行なう. 点の位置は, 1 フレーム目と 2 フレーム目はランダムとする. 3 フレーム目以降は, 比較的大きな確率で 2 フレーム前の位置から 1 フレーム前の位置への移動方向と距離に近い位置になるようにする. このようにした理由は, 例えば航空路監視レーダーのシステムなどにおいて, 移動方向や距離が大きく変わることは少ないと考えられるためである. 確率は 90% とする. また, 同様の理由で各フレーム間の点の移動距離は 20 以下となるようにする. 1 フレーム目は点に対して右上にラベルを配置することとする (図 3).

Slider モデルにおいて, ラベル数を 30, 重み $\alpha = 5.0$ として実験した結果を表 1 に示す. 各値は, 1 つのインスタンスに対して各フレームにおける目的関数値の平

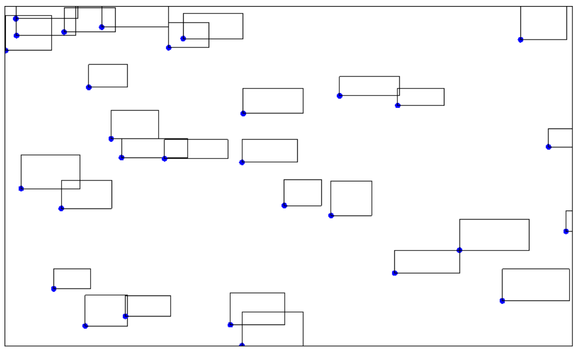

図 3. 1 フレーム目の点とラベルの位置の例

均を求め, その値が最もよくなったアルゴリズムに対して 1 を加えたものである. すなわち, 各アルゴリズムが 100 回中何回最もよい結果となったか数えたものである. 各フレームにおけるラベルの移動距離が常に最小であるときを比較に用い, 移動距離最小と表す.

GRASP がよい結果となる場合が多かった. また, 更新時間が短いほど, 山登り法や Greedy 法がよい結果になっていることから, 更新時間が短いときは単純な手法を選択するべきではないかと考えられる. また, どの更新時間でも Simulated Annealing の結果が最もよくなることがなかったため, ラベル数が 30 の場合, Simulated Annealing は 2[s] 程度では有効な結果が得られないのではないかと考えられる.

更新時間が 1[s] の 100 個のインスタンスのうちの 1 つにおける GRASP の 10 フレーム目における点とラベルの位置は図 4 のようになった.

表 1. Slider モデルの計算機実験結果

アルゴリズム	更新時間			
	2[s]	1[s]	0.1[s]	0.03[s]
山登り法	4 回	6 回	16 回	28 回
多スタート局所探索法	11 回	20 回	44 回	26 回
GRASP	84 回	73 回	20 回	1 回
Simulated Annealing	0 回	0 回	0 回	0 回
Greedy 法	1 回	1 回	20 回	45 回
移動距離最小	0 回	0 回	0 回	0 回

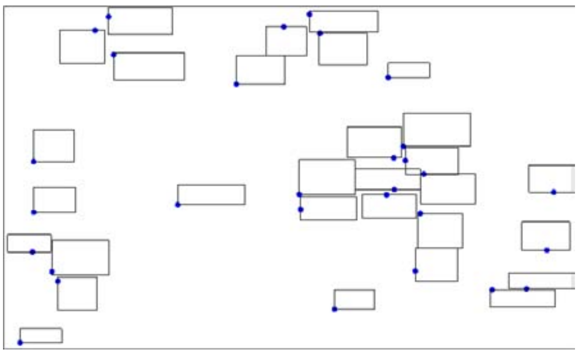

図 4. Slider モデルで得られた描画結果

表 2. 引き出し線を用いたモデルの計算機実験結果

アルゴリズム	更新時間			
	2[s]	1[s]	0.1[s]	0.03[s]
山登り法	5 回	4 回	10 回	2 回
多スタート局所探索法	0 回	2 回	81 回	98 回
GRASP	95 回	94 回	9 回	0 回
Simulated Annealing	0 回	0 回	0 回	0 回
Greedy 法	0 回	0 回	0 回	0 回
移動距離最小	0 回	0 回	0 回	0 回

引き出し線を用いたモデルにおいて, ラベル数を 30, 重み $\alpha = 5.0, \beta = 15.0, \gamma = 100.0, \delta = 100.0, \epsilon = 1000.0$ として 100 個のインスタンスに対して実験した結果を表 2 に示す. 2[s], 1[s] の場合, GRASP が最もよい結果となった. 0.1[s], 0.03[s] の場合, 多スタート局所探索法が最もよい結果となった. GRASP は, Slider モデルと比較して Greedy 法における計算時間が増大したため, 更新時間が 0.1[s] 以下という短い時間ではよい結果とならなかったと考えられる. 更新時間が 1[s] の 100 個のインスタンスのうちの 1 つにおける GRASP の 100 フレーム目の点とラベルの位置は図 5 のようになった.

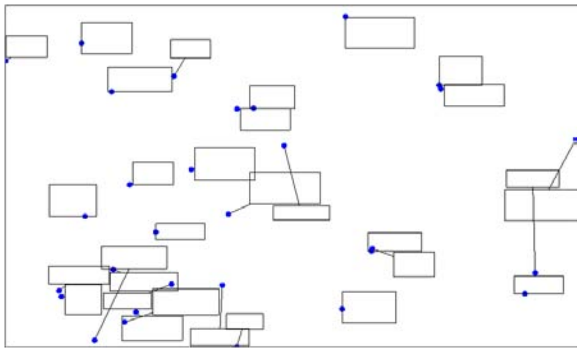

図 5. 引き出し線を用いたモデルで得られた描画結果

5 結論

Slider モデル, 引き出し線を用いたモデルにおいて, 移動する点に対するラベル配置問題を扱った. この問題を, 常にラベルの大きさを変えず, すべてのラベルを配置することを制約条件とし, ラベル同士の重なった面積, ラベルの移動距離, 点とラベルの重なった数, 引き出し線の長さ, 引き出し線同士の交差数, 引き出し線とラベルの交差数の重み付きの和を最小化する問題とした. また, 各フレームにおいて更新時間以内にラベル配置を求めなければならないとした. そして, この問題に対する局所探索を行なう手法を複数提案し, 計算機実験をすることでこれらの手法を比較した. GRASP によって得られた結果が総じてよいものとなった. また, 更新時間が短い場合は単純な手法の結果がよくなった.

今後の課題としては, Greedy 法を改善することで GRASP を改善することや過去に探索した解をうまく利用する新たなアルゴリズムを提案することが挙げられる.

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, 多くの助言をしてくれた今井研究室の学生各位に感謝致します.

参考文献

[1] G. L. Cravo, G. M. Ribeiro and L. A. N. Lorena, "A greedy randomized adaptive search procedure for the point-feature cartographic label placement," *Computers and Geosciences*, 34:373-386, 2008.

[2] A. Dorbes, "Requirements for the implementation of automatic and manual label anti-overlap functions," *EEC Note No. 21/00*, EUROCONTROL Experimental Centre, 2000.

[3] D. Ebner, G. W. Klau and R. Weiskircher, "Label number maximization in the slider model," In János Pach (editor), *Proceedings of the 12th International Symposium on Graph Drawing (GD'04)*, 144-154, Springer-Verlag, 2005.

[4] E. M. McCreight, "Priority search trees," *The SIAM Journal on Computing*, 14:257-276, 1985.