G-15

勢力圏分割を用いた高速な探索の簡略化

A Simplified Rapid Search Restricted to Voronoi Diagrams

辻野 弘章†　　今井 敏行‡
Hiroaki Tsujino　　Toshiyuki Imai

1. はじめに

質問点がどの郵便局(母点)に最も近いかを答える問題は郵便局問題と呼ばれている. 母点の数が n 個である郵便局問題に対して, 各母点と質問点との距離を順に探索していき答えた場合, 探索時間は $O(n)$ である. だが, 事前に勢力圏分割を行い, 探索を行うことで探索時間が $O(\log(n))$ と, 高速になる. この高速化には, 一般の平面分割に対する汎用算法が利用される. 実際, 計算幾何学の教科書でもそのように記述されている[1]. 本研究で参考にした Edelsblunner の算法も, 探索時間 $O(\log(n))$ で返答することができるが, 探索を行うまでに行う前処理が一般平面グラフに対応したもので, 勢力圏分割において不必要な処理が行われている. そこで, 勢力圏分割に限れば, 勢力圏分割やドロネー図の性質を用いることで, 元の方法では複雑であった処理を簡略化できるのではないかと考え, 探索法の簡略化を行うことを目的とした.

2. 準備

2.1 勢力圏分割

平面上に n 個の点 P_i (x_i, y_i) $(i = 1, 2, \cdots, n)$ が与えられたとき, 点 P_i の勢力圏 $V(P_i)$ を点 P と点 P_i のユークリッド距離を表す $d(P, P_i)$ を用いて表すと

$$V(P_i) = \bigcap_{j \neq i} \{P \mid d(P, P_i) < d(P, P_j)\} \qquad (1)$$

となる[1]. この $V(P_i)$ $(i = 1, 2, \cdots, n)$ による分割を勢力圏分割と呼ぶ. $n = 30$ の時の勢力圏分割の例を図 1 に示す.

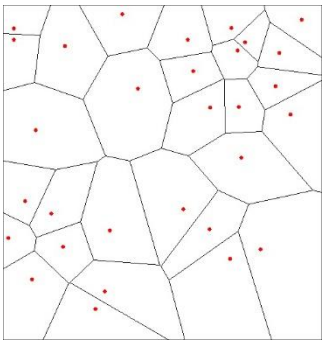

図 1 : 勢力圏分割の例

2.2 ドロネー図

式(1) において, 点 Pi と点 Pj の対応するボロノイ辺が共通のものであった場合, 点 Pi と点 Pj とを線分で結ぶことにより, Pi $(i = 1; 2; \cdots; n)$ の三角形分割が得られる. この三角形分割をドロネー図と呼ぶ. ドロネー図は, ボロノイ領域の隣接関係を表す[2]. 各母点同士を結ぶ線分をドロネー辺と呼ぶ. ドロネー図の例を図 2 に示す.

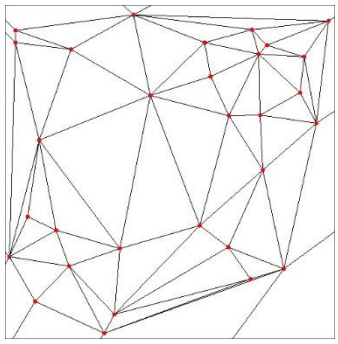

図 2 : ドロネー図の例

2.3 パスについて

平面グラフの辺を, 条件に従い, 各辺の集合に分解する手順が本研究では存在する. 各辺の集合は折れ線のようになる. 本論文では, その折れ線のことをパスと呼ぶ.

3. Edelsbrunner の算法

Edelsbrunner の算法では, 直線分を辺とする n 点の平面グラフ上に存在する質問点を, 探索時間 $O(\log(n))$, 領域量 $O(n)$ で探索することができる. 本節ではその方法について文献[1] に準拠して, 手順を以下に示す.

1. 平面グラフの正規化
2. 平面グラフのパス分け
3. パスの重複している区間をなくす
4. 二分木の子から親へ Y 座標値を付加
5. 二分探索木を用いて探索

3.1 平面グラフの正規化

まず, 正規でない平面グラフを正規な平面グラフにする.下記が正規な平面グラフである条件である.

1. Y 座標値が最大の点と最小の点以外の点に対して, その点より Y 座標値が大きい点と小さい点とを結ぶ辺が存在する.

条件を満たしていない点に対して, 他の辺と交差しないように線分を 1 本付加することにより,条件を満たすようにし,正規化していく

†和歌山大学大学院システム工学研究科, Graduate School of Systems Engineering, Wakayama University

‡ 和歌山大学システム工学部, Facuity of Systems Engineering, Wakayama University

3.2 平面グラフのパス分け

正規化された平面グラフを各パス(折れ線)で分けていく．分けるとき，以下の 3 つの条件を満たすように分けていく．

1. グラフの各辺は分けられたパスのどれかに含まれる．
2. 各パスは Y 軸に関して単調である．
3. 任意の 2 つのパスが X 字型に交わることがない．

条件の2つ目の Y 軸に対して単調であるとは，Y 座標値最大の点から Y 座標値最小の点まで Y 座標値が常に減りつつけることを表している．3番目の条件から，最後に二分探索を行う際に各パスごとの左右関係がわかる．これにより，質問点の探索を行った場合，探索時間 $O((\log(n))^2)$ かかり，$O(n^2)$ の領域量を必要とする．

3.3 パスの重複している区間をなくす

前節で述べたように，現段階では，領域量 $O(n^2)$ かかってしまう．これを領域量を $O(n)$ に落とす．まず，最後に行う二分探索のために 二分木を作成する．木の各点に1本のパスを対応させておく．このとき，木の深さが最小になるよう にかつ，パス同士の左右関係を保ったまま作成する．また，複数のパスに含まれている区間は，作成した二分木の最も根に近い部分で覚えているパスでのみ覚えておく．そうすることで重複して覚えている区間を1つのパスでのみ覚えているようになる．最も根に近い部分以外の重複していた区間 は，同じ点を結ぶ空の辺として処理する．空の辺が連続する部分は，1本の空の辺として処理する (図3)．このようにして，各パスの区間を覚えておくことにより，領域量を $O(n)$ にすることができる．

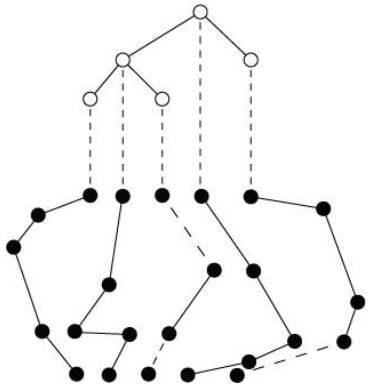

図3: パスの重複をなくした二分木

3.4 二分木の子から親へ Y 座標値を付加

ここでは探索時間を $O(\log(n))$ に減らす方法を述べる．探索時間が $O((\log(n))^2)$ である原因に， パスの左右判定をするとき，パスのすべての区間で探索を行っていることが挙げられる．この原因を解消するために， 二分木に格納されているパスに対して事前に処理を行う． 先ほど作成した二分木の子に格納されているパスの Y 座標値を親に格納されているパスに1つおきに付加していく． 以下に Y 座標値を付加していく時の条件を示す．下記2は付加する値が1つおきなので実現されている．

1. 二分木の親の各区間は， 子の区間と高々2 つずつしか交わらない．
2. Y 座標値を付加した後の全体で覚えている Y 座標値の総数が $O(n)$ である．

これらの条件に従って子から親へと次々に Y 座標値を付加していくことにより，根の部分に格納されているパスの各区間が細分化される．子から親へと Y 座標値を付加していく時，元から親に付加しようとしている Y 座標値がある場合は，何もしない．図4に条件に従い，子から親へ Y 座標値を付加した例を示す． これにより，最初に探索を行う根の部分に格納されているパスでは，$O(\log(n))$ の探索時間がかかるが，その後の探索には，条件の2より，探索時間 $O(1)$ で探索することができる． 故に，全体としては探索時間 $O(\log(n))$ で探索を行える．

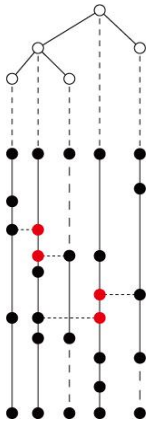

図4: 子から親へ Y 座標値を付加した平面グラフ

4. 提案手法

Edelsbrunnerの算法では，一般平面のグラフに対応していたので，グラフを正規化する必要があった．しかし，今回用いたのは母点数が n 個の勢力圏分割である．勢力圏分割に対し，無限遠点上にある 3 点を用いて，無限に伸びるボロノイ辺を止めることで正規のグラフとすることができる．ゆえに，正規化する処理を省くことができる．また，入力する母点の座標は，X 座標，Y 座標ともに 0 以上 1 以下とする．以下に今回提案する探索法の手順を示す．

1. 母点を X 座標順にソートする
2. 各隣接母点間の X 座標の中間値を求める
3. ドロネー図を用いて，パス分けを行う
4. 探索順の二分木を作成し，パスの区間の重複をなくす
5. 二分木の子から親へ Y 座標値を付加する
6. 二分木探索を行う

4.1 母点を X 座標順にソートする

n 個の母点を，事前に X 軸の小さいほうから順にソートしておく．ソートしておくことにより，後の処理で行う母点同士の中間値を求める際に都合がよい．

4.2 各隣接母点間の X 座標の中間値を求める

ソート済みの母点に対し，隣接母点間の X 座標の中間値を求めていく．母点数が n 個であるため，$n-1$ 個の中間値が求められる．

4.3 ドロネー図を用いて，パス分けを行う

本提案手法では，求めた中間値とドロネー図を用いて，勢力圏分割を各パスごとに分けていくことで簡略化しようと考えた．中間値と交差するドロネー辺の走査を行い，交差していたドロネー辺に対応しているボロノイ辺をパスの区間として抽出を行っていく．この処理を n - 1個の中間

値で行うことで，節5. 2に示した3つの条件に従い, パス分けを行うことができる.

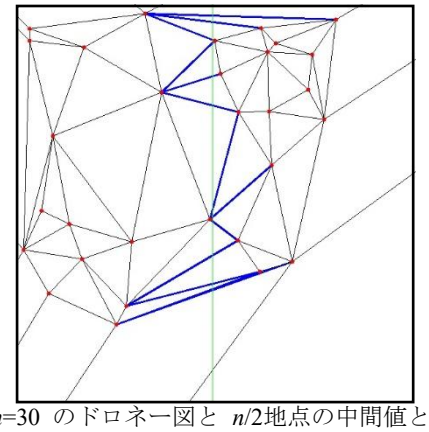

図5：n=30 のドロネー図と n/2地点の中間値との交差

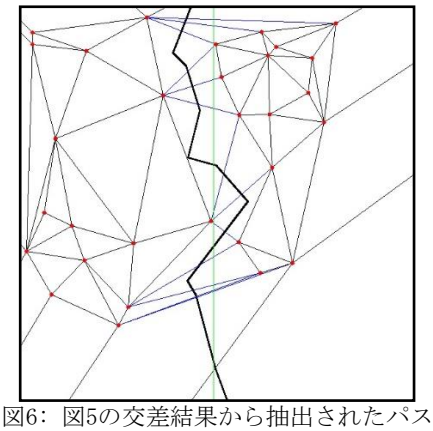

図6：図5の交差結果から抽出されたパス

4. 4　探索順の二分木を作成し, パスの区間の重複をなくす

抽出されたパスを探索順に沿って作成された二分木の各点に格納していく．この時，区間に重複があったならその重複している区間をパスの区間として覚えずに，次の重複していない区間を覚えるようにする．つまり，より根に近い部分でのみ，パスの重複区間を覚えておく．パスの重複をなくすのは，使用領域量を$O(n)$にするためである.

4. 5　二分木の子から親へ Y 座標値を付加する

二分木の子に格納されているY 座標値を1つ置きに親に格納されているパスに付加していく．付加する際，もともと親に付加しようとしているY座標値が存在する場合，何も行わず，次のY座標値の付加を行っていく．付加することにより，親に格納されているパスの1つの区間から参照されるこの区間が2つ以下となる．子に格納されているパスのY座標値を親に格納されているパスに付加しても，1つおきに付加しているため，使用領域量の$O(n)$は保ったままである.

4. 6　二分木探索を行う

これらの処理を行った後，作成した二分木を用いて探索を行うことで，探索時間$O(\log(n))$で探索することができる. Edelsbrunner の算法と大きく異なる点は, パス分けを行うのに, ドロネー図を用いた点である.

5. 結果

今回提案した手法で, 母点数が 30 個の勢力圏分割のパス分けを行った結果, 条件を満たしパスに分けることができた. しかし, 中間値を求める際, すべての隣接母点間で中間値を求め, パス分けに用いたことで, より根に近い部分で多くの区間を覚えたため,すべての区間が重複していてパスを持たない二分木の点が 7 個発生した.また, 母点の位置を中央に偏らせて実験を行った結果を以下の表 1 に示す.

表 1: 母点の座標を偏らせた場合の結果

母点の偏り	パスを持たない点の数
X 座標のみ中央	12 個
X 座標, Y 座標ともに中央	5 個
Y 座標のみ中央	0 個

この結果より,本提案手法では母点の分布の偏りにより, 空のパスのできやすさに違いがあることが分かった. 計算量としては $O(\log(n))$が達成できているものの, 実探索時間の観点からは, パスのとり方により一層の工夫が必要と思われる.

6. おわりに

今回は Edelsbrunner の算法の中でもパス分けの部分のみではあるが簡略化できたといえる. しかし, 探索全体を考えると,まだ簡略化したとは言い切れない部分もあるため, 今後の課題としては座標値の付加の部分などの簡略化を行うことが挙げられると考える. また, 実探索時間の高速化も課題である.

本研究の一部は科学研究費補助金の助成を受けている.

参考文献

[1] 計算幾何学と地理情報処理(伊理正夫 監修, 塚越武志 編集, 共立出版)

[2] ボロノイ図を作る (http://qiita.com/yellow_73/items/abfca17297124785fe28(2017 年 2 月 14 日アクセス))