平面上の点施設の最小費用連結問題

鈴木 光雄（沼田 一道助教授, 池辺 淑子助手）

1．はじめに

平面上に点（施設）が与えられたとき，各々の点間を直線の連絡路で結び，連絡路を辿って相互に到達できるようなネットワークを構成することを考える。このようなネットワークの中で連絡路の総延長を最小にするものを求める。この問題を最小費用連結問題と呼ぶ。最小費用連結問題は与えられた点と，全２点間を結ぶ直線の枝から成る完全グラフにおいて最小全域木（minimum spanning tree：最小木とも呼ぶ）を求める問題に他ならない。このグラフにおいて枝の長さは，２点間のユークリッド距離である。例えば，通信施設を通信回線で結ぶ通信網の建設において通信回線は障害物の影響に関係なく直線で結べるものとし，かつ建設費用が通信回線の総延長に比例するとすれば経済的に通信網を建設する（総延長を最小化する）問題は，最小費用連結問題となる。

1図

2．研究目的

本研究では，点（施設）を連結する最小全域木を求める際に，全２点間を直線の枝で結んだ完全グラフの全枝を対象として求める方法〔方法１〕と，勢力圏の考え方をもとに，完全グラフから絶対使われることのない枝を除いたグラフ（勢力圏の隣接関係を示すドローネ網（Delaunay net）［１］）の枝を候補として求める方法〔方法２〕の２つの方法を取り上げた。２つの方法のうち〔方法１〕は，対象とする枝の本数（n点の時，n(n−1)／２本の枝が対象）が多いため，完全グラフの枝候補から最小全域木を構成するのに時間を要してしまう。また，〔方法２〕では対象とする枝が少なくなる分，有利ではあるがドローネ網を求めるのに余分の時間を要する。そこで，実際に２つの方法を用いて最小全域木を求め，実行時間を計測して，比較・検討する。

3．勢力圏とドローネ網

3．1 勢力圏

平面上にn個の点$P_i=(x_i, y_i)$（i＝1，・・・，n）が与えられたとき，(1)式で与えられる点の集合（平面上の領域）$V(P_i)$を点P_iの勢力圏［２］と呼ぶ。

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

（ただし，$d(P_i,P)$は点P_iと点Pのユークリッド距離）

つまり，$V(P_i)$は「P_1，・・・，P_nの中でP_iが最近点であるような点の集合」ということができる。従って（１）式より，全平面（境界を除く）は$V(P_i)$に分割される。これを示す図（図１）を勢力圏図（ボロノイ図 Voronoi diagram）と呼ぶ。また，点P_i（i＝1，・・・，n）を勢力圏図の母点と呼ぶ。 勢力圏は平面上にn個の母点P_i（i＝1，・・・，n）が与えられたとき，次のようにして構成できる。まず，母点P_1と母点P_2を結ぶ線分を垂直二等分する直線を引く。母点P_1のある方の半平面（図２の網点部の領域）をH_{12}と記す。明らかに，この領域H_{12}の点については，母点P_2より母点P_1の方が近い。同様にして，母点P_1と母点P_j（j＝2，・・・，n）に関してH_{1j}を定義すると，領域H_{12}，H_{13}，H_{14}，・・・，H_{1n}の共通部

分（図2の太い実線で囲まれた領域）が母点P_1の勢力圏$V(P_1)$となる。

一般に，母点P_i側の平面をH_{ij}と記すと，母点P_iの勢力圏$V(P_i)$は，

$$V(P_i) = H_{i1} \cap \cdots \cap H_{ii-1} \cap H_{ii+1} \cap \cdots \cap H_{in} \qquad (2)$$

で与えられる。勢力圏は半平面の共通部分からなる凸多角形

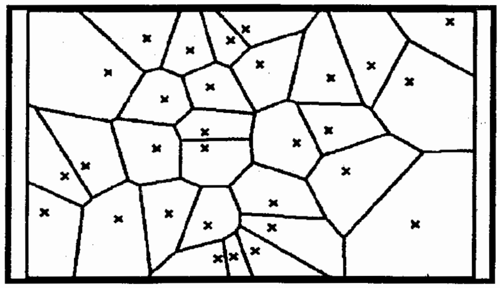

図1 勢力圏図（ボロノイ図）（点数＝３０）

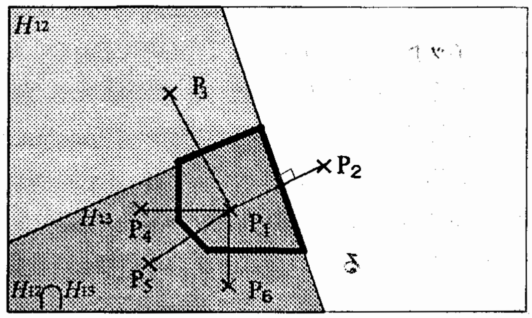

図2 半平面の共通重なり部分としての勢力圏

３．２ ドローネ網

勢力圏図において母点P_iと母点P_jの勢力圏が共有辺を持つとき，母点P_iと母点P_jとを枝（線分）で結ぶことによってできるグラフがドローネ網（図３）である。従って，ドローネ網の枝によって，勢力圏の母点P_iと隣接している母点P_j（$i \neq j$）の隣接関係が表され，各母点についての最近隣母点を調べるのに有用である。

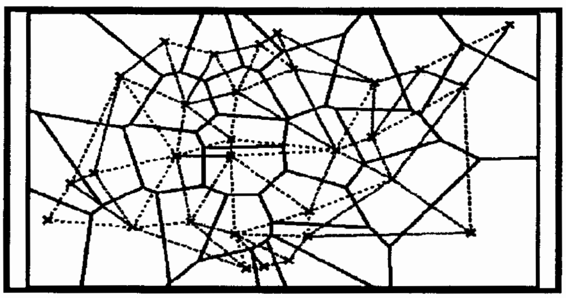

図3 勢力圏図（太線）とドローネ網（点線）の関係 （n＝３０）

4. 勢力圏図を求める算法

n個の母点P_1，・・・，P_nに対する勢力圏図を構成する算法の１つとして逐次切断法がある。逐次切断法は平面上にn個の母点が与えられた時，はじめに各母点に対する勢力圏をn個の母点をすべてを含む長方形であると設定し，次に自分の母点と自分以外の各母点との垂直２等分線により，長方形を次々に切断していく。すべての母点に関する切断を終ると１つの勢力圏が構成できる。従ってこの操作をすべての母点について繰り返すとn個の母点からなる勢力圏図が構成できる。

以下に，逐次切断法のアルゴリズムを示す。

ｓｔｅｐ０：平面上に母点P_i（$i=1$，・・・，n）の配置が与えられる。

ｓｔｅｐ１：各母点P_iに対する勢力圏を長方形と設定し，辺をそれぞれ登録する。

ｓｔｅｐ２：他の母点P_j（$j \neq i$，$j = 1, 2, \cdots, n$）について以下（ａ）から（ｅ）を繰り返し，母点P_iの勢力圏を求める。

ｓｔｅｐ２－（ａ）：母点P_iと母点P_jの垂直２等分線を引き，現在登録されている線分と交わるかを調べる。

ｓｔｅｐ２－（ｂ）：交わらない垂直２等分線は消去し，交わる線分がある時，交点を求める。

ｓｔｅｐ２－（ｃ）：交わった線分を，その端点のうちH_{ij}内にあるものと，交点を結ぶ線分で置き換える。

ｓｔｅｐ２－（ｄ）：両端点ともH_{ij}内にない線分は消去する。

ｓｔｅｐ２－（ｅ）：求めた交点を端点とする線分を新たに登録する。

ｓｔｅｐ３：この時点で登録されている線分が，勢力圏$V(P_i)$の境界線分を与える。

５．最小全域木を求める算法

５．１ Prim の算法

最小全域木を求める算法［３］はいくつか存在するが本研究では，Prim の算法を採用した。このアルゴリズムは全ての点からなる点集合Ｖから点を１つずつ選んで，点集合Ｖの部分集合である点集合Ｕに加えていく。これによって，出発点を中心とする部分最小全域木Ｔを少しずつ成長させていく。

以下に，Prim の算法のアルゴリズムを示す。

ｓｔｅｐ０：点集合Ｖから出発点ｓ（$s \in V$）を任意に選び，$U = \{s\}$，部分最小全域木$T \neq \varnothing$

ｓｔｅｐ１：$V \neq U$である限り以下（ａ），（ｂ）を繰り返す。

ｓｔｅｐ１－（ａ）：ＵとＶ－Ｕを連結する重み最小の枝ｅを求める。

ｓｔｅｐ１－（ｂ）：枝ｅの端点でＶ－Ｕに含まれる点ｗを求め，$U = U \cup \{w\}$とする。さらに$T = T \cup \{e\}$とする枝ｅを最小全域木の一部として確定する

５．２ ドローネ網と最小全域木

勢力圏の隣接関係を表わすドローネ網の枝のみを候補とすることで最小全域木を構成することができる。その理由は母点P_iの勢力圏$V(P_i)$に隣接する勢力圏を$V(P_{jk})$（$i \neq j$，$k = 1, 2, \cdots, s$）とした時，母点P_iを他の母点に連結する枝として$P_i P_{jk}$以外の$P_i P_x$の枝を採用したとすると母点P_iの勢力圏に隣接する勢力圏の母点以外の母点P_xに行くのには，必ず母点P_{jk}の勢力圏を横切らなければならず，枝$P_i P_x$は，枝$P_i P_{jk}$とのいずれよりも長い。これと Prim の算法のｓｔｅｐ１－（ａ）における枝の選び方より枝$P_i P_x$は最小全域木に採用されることがない。

６．数値実験

６．１ 実験方法

２つの方法を用いて，最初に与える点数を５０から１０００まで５０ずつ変化させ，各点数について得られた最小全域木を構成する実験を１０回行い，実行時間を計測した。各点数についての実験データから実行時間の平均をプロットし，実行時間の比較を行う。使用したプログラム言語はＣ言語，使用計算機は日本電算機社のＪＳ

5／70である。

6．2 結果及び考察

予想していた結果と数値実験から得られた結果との間に大きな差が生じてしまった。図4の〔方法1〕と〔方法2〕での実行時間の比較から，〔方法1〕の完全グラフの枝を対象として最小全域木を構成する実行時間の方が短いことが分かる。さらに，図5での2つの方法における実行時間の比率Dはほぼ一定であり，〔方法1〕の方が〔方法2〕より約40％も実行時間が速いというものであった。また，実験結果を両対数でプロット（点数－実行時間）し，それぞれの直線の傾きから計算量を見積もると〔方法1〕，〔方法2〕共に大体$O(n^2)$と推定できた。

当初は，〔方法1〕で求めた枝の候補数より〔方法2〕で求めた枝の候補数の方が少ない点を重視し，2つの方法から求めた枝の候補数から最小全域木を構成するまでの実行時間が〔方法2〕で求める方が有利であると予想していた。この予想と反対の結果が得た理由として勢力圏を構成する方法である逐次切断法のアルゴリズムが1つの勢力圏を構成するのに計算量が多い為に，ドローネ網の枝から最小全域木を構成するまでの計算量が少なくても全体的に相殺されてしまったからではないかと考える。

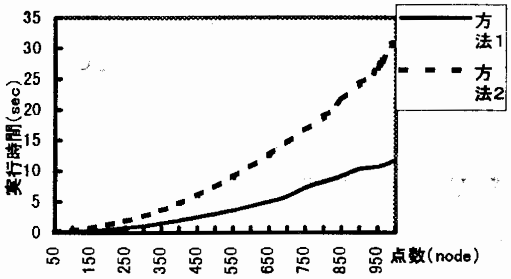

図4 実行時間の比較

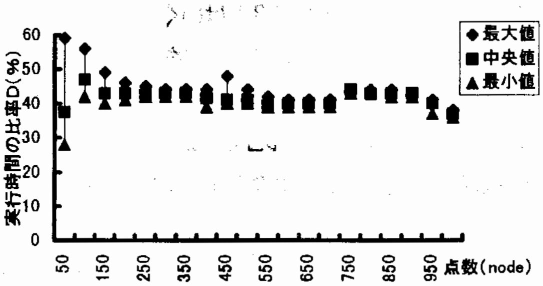

図5 〔方法1〕と〔方法2〕の実行時間の比率Dの分布
（D＞0の時，〔方法1〕の方が良い事を示す）

7．まとめ

本研究では，平面上の点施設の最小費用連結問題に対して最小全域木を構成する際に，全2点間を直線の枝で結んだ完全グラフの全枝を対象とする方法と勢力圏の隣接関係を示すドローネ網の枝を候補とする方法の2つの方法を考えた。そして2つの方法を用いて最小全域木を求め，実行時間を計測し，比較した。

実験結果では，〔方法1〕における完全グラフの全枝から最小全域木を構成する方が実行時間が短くなる。〔方法2〕の問題点はドローネ網を求める算法であり，勢力圏図を構成する良い算法として1点ずつ母点を付け加えていく逐次添加法や母点の集合を半分に分けてそれぞれの勢力圏図を作って併合することを再帰的に繰り返す再帰二分法を用いれば，どちらも計算量が逐次切断法の計算量より下回るので〔方法1〕と〔方法2〕の実行時間は逆転するのではないかと思われる。

【参考文献】

［1］岡部篤行，鈴木敦夫：「最適配置の数理」，朝倉書店，1992.

［2］夷隅嘉晃：「調布市内の小学校のVoronoi(勢力圏)図と通学区」，電気通信大学卒業研究論文，1991.

［3］浅野孝夫：「情報の構造 上・下巻」，日本評論社，1993.