巡回セールスマン問題に対する発見的解法の高速化

岩倉　行信（沼田　一道　助教授）

1．はじめに

代表的な組合せ最適化問題の一つである、巡回セールスマン問題(Traveling Salesman Problem : TSP)は訪問対象となるすべての都市を１度ずつ訪問して出発した都市に戻る巡回路の中で、距離や時間に基づく移動費用の総和が最小となる巡回路を求める問題である。

都市数をnとした場合、巡回路の個数は$((n-1)!/2)$個と有限であるが、nが大きくなるとこの数は急速に増大する。したがってnが大きい場合全巡回路を列挙することによって最適解を探索することは事実上不可能となる。このようなＴＳＰに対して、厳密な最適解を少しでも速く求める厳密解法の研究、満足できる準最適解を実用的な計算時間内に求める発見的解法の研究が盛んに行われている。特に後者の場合、初期値を変えるなどして多数回の探索を行うのが普通であり、１回の探索を早くしたいという要求は切実である。

2．研究目的

発見的解法高速化の手段として、Candidate Sets を取り上げる。最も基本的な発見的解法である２－ｏｐｔ法で元の問題と Candidate Sets の枝からなる問題を解き、得られる近似解の精度と実行時間を比較する。

3．巡回セールスマン問題

本研究において対象とされるＴＳＰは、都市間の移動費用が巡回路の方向に無関係である、対称型巡回セールスマン問題とする。nを都市数、S_nを巡回路全体の集合、$\sigma(i)$を巡回路σにおいてi番目に訪問する都市、$(i=1,\cdots,n),\sigma(n+1)=\sigma(1)$、$C_{ij}$を都市$i,j$間の移動費用（$C_{ij}=C_{ji},i,j=1,\cdots,n,i\neq j$）、Ｚを巡回路の総移動費用とすると、巡回セールスマン問題（ＴＳＰ）は、以下のように定式化される。

$$\boxed{\textit{Minimize}\ \ Z=\sum_{i=1}^{n}C_{\sigma(i)\sigma(i+1)}\quad,\quad \textit{Subject To}\quad \sigma\in S_n}$$

4．Candidate Sets

巡回セールスマン問題（ＴＳＰ）を発見的解法を用いて解く場合、原則的には全ての枝（都市間の道）を考慮して準最適解を求める。この場合、明らかに最適解に含まれないと考えられる長い枝についてまで考慮しているので、非常に効率が悪く、かなりの実行時間を必要とする。したがって、最適解に含まれる可能性の薄い、長い枝を除いたものの中で発見的解法を用いれば、計算に必要な時間を減少させることができると考えられる。最適解に含まれる可能性があると判断して選び出された枝の集合を Candidate Set と呼ぶ。

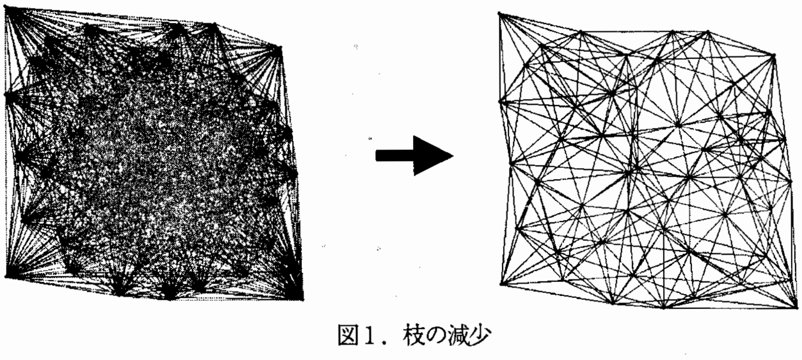

図１．枝の減少

本研究では２種類の Candidate Set について考える。方法[１]の Candidate Set はすべての移動費用の定義に対して適用可能であり、特にユークリッド距離で定義された問題では高速に求めることができる。方法[２]の Candidate Set は、ユークリッド距離で定義された問題についてのみ用いることができる。

４．１　方法［１］k nearest neighbors

最適巡回路では、最も近い２都市を結ぶ枝が多く含まれている。したがって各都市についてその都市となるべく近い２都市を結ぶ枝を Candidate Set として選ぶ。ここで k nearest neighbors をある都市に対して近い方から k 個までの都市と定義し、対応する枝を Candidate Set として選んでいく。

k nearest neighbors による Candidate Set の欠点は、都市がいくつかの集団に分かれている場合、k nearest neighbors が集団内のみで構成され、巡回路を作るのにうまく行かない可能性が生じることである。この欠点は方法［２］で解決される。

k nearest neighbors は全ての都市間の移動費用を列挙することで求められるが、都市数が多くなると相当の実行時間を要する。ここで、移動費用がユークリッド距離で定義される問題については、ドロネーグラフの接続関係を利用することで高速に求めることが可能である。ドロネーグラフはボロノイ図より求められる。

４．１．１　ボロノイ図

平面上に n 個の点 $P_i=(x_i,y_i)\ (i=1,\cdots,n)$ が与えられたとき、次の式で与えられる点の集合 $V(P_i)$ を点 P_i のボロノイ領域（勢力圏）と呼ぶ。

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

（ただし、$d(P_i,P)$ は点 P_i と点 P のユークリッド距離）

ここで、点 P_i をボロノイ領域 $V(P_i)$ の母点と呼び、全ボロノイ領域を示す幾何図形をボロノイ図と呼ぶ。

ボロノイ図は、定義にしたがって次のようにして構成できる。まず、母点 P_1 と母点 P_2 の垂直２等分線を引く。母点 P_1 のある方の半平面を H_{12} とすると、明らかに、領域 H_{12} では母点 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 $P_j\ (j=2,\cdots,n)$ の垂直２等分線よりなる領域を H_{1j} とすると、領域 $H_{12}, H_{13}, H_{14},\cdots,H_{1n}$ の全共通部分が母点 P_1 のボロノイ領域 $V(P_1)$ となる。

４．１．２　ドロネーグラフ

ボロノイ図において母点 P_i と母点 P_j のボロノイ領域が同じ辺を共有しているとき、母点 P_i と母点 P_j のボロノイ領域は隣接関係にあるという。

ボロノイ領域（母点）を点、隣接するボロノイ領域間を枝で結んだグラフがドロネーグラフである。

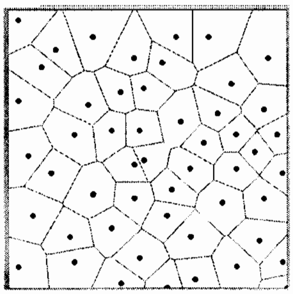

図２．ボロノイ図（ｅｉｌ５１）

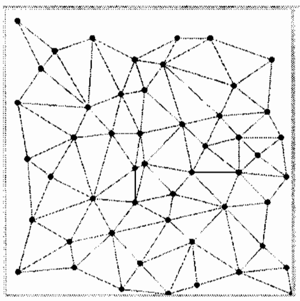

図３．ドロネーグラフ（ｅｉｌ５１）

4．1．3　ドロネーグラフの接続関係を利用してのk nearest neighbors

Step1：ドロネーグラフを求める

Step2：都市P_iにドロネーグラフにおいて接続している都市の集合をL_iとし、$s=1$とする。L_iの中でP_i に最も近い都市を$X[s]$とし、L_iから$X[s]$を除く。sに1を加える。

Step3：都市$X[s-1]$ にドロネーグラフにおいて接続している都市をL_iに加える。L_iの中でP_iに最も近い都市を$X[s]$とし、L_iから$x[s]$を取り除く。

Step4：$s=k$であればStep5へ、そうでなければsに1を加えてStep3へ

Step5：$X[r]\,(r=1,\cdots,k)$がP_iのk nearest neighborsである。

Step2～Step5を$i=1,2,\cdots,n$について行う。

4．2　方法［2］Delaunay Candidate Set

ドロネーグラフは連結グラフなので、その枝をすべて採用すれば方法［1］の欠点は解決できる。そこで、ドロネーグラフの枝をまずCandidate Setに選ぶが、ドロネーグラフの枝のみではCandidate Setとしては、枝の数が少なく十分ではない。そこで次の手順を用い枝を追加する。

都市$P_i\,(i=1,\cdots,n)$にドロネーグラフにおいて接続している2都市P_j, P_kについて枝P_jP_kが Candidate Set に含まれていなければ、その枝を Candidate Set に加える。この操作によって得られる Candidate Set は Delaunay Candidate Setと呼ばれる。この集合には、やや長い枝を含む可能性がある。

5．　2－ｏｐｔ法

本研究では、発見的解法として2－ｏｐｔ法を用いる。2－ｏｐｔ法とは下図のような巡回路において、枝ａｂ，ｃｄを除去し、枝ａｃ，ｂｄを加えたとき巡回路長が減少すれば、枝の交換を行う。この操作を、長さを減少させる交換がなくなるまで繰り返す方法である。

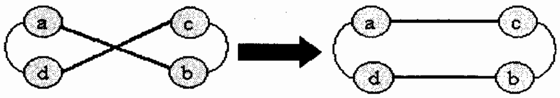

図4．　2－ｏｐｔ法

6．数値実験

6．1実験方法

実験に使うＴＳＰの問題は、ＴＳＰの問題集ＴＳＰＬＩＢのデータを使用する。

Candidate Setを求めるのに要する時間と、2－ｏｐｔ法に要する時間を調べる。また、Candidate Setの有効性を調べるために、ＴＳＰＬＩＢにある解答と、2－ｏｐｔ法による準最適解の比（相対誤差）を求める。

6．2実験結果

計算機による実験結果を以下の表に示す。使用した計算機は FMV-DeskPower/S(pentium100Mhz、１６ＭＢ)。プログラムは、Boland 社の Delphi3.1を用いて作成した。表1には、ドロネーグラフと Candidate Set を求めるのに要する時間を示す。結果は１０回の試行の平均による。

表1．数値実験の結果（ドロネーグラフと Candidate Set を求める時間［秒］）

問題名	都市数	ドロネーグラフ	方法［1］(列挙)（ｋ＝１０）	方法［1］（ドロネーグラフ）（ｋ＝１０）	方法［2］
ａｔｔ４８	４８	０．０１５２	０．０１４０	０．００５６	０．００２８
ｂｅｒｌｉｎ５２	５２	０．０４４０	０．０２３０	０．０１１２	０．００７０
ｅｉｌ１０１	１０１	０．０６８６	０．１６９４	０．０２１８	０．００３４
ｃｈ１５０	１５０	０．１７２６	０．５５６８	０．０４６４	０．０１６０

表２～４には、２－ｏｐｔ法による数値実験の結果を示す。結果は１０００回の試行の平均による。

表２．数値実験の結果（２－ｏｐｔ法（全枝））

問題名	都市数	時間［秒］	最適解	準最適解	準最適解／最適解	最良の解	最悪の解
ａｔｔ４８	４８	０．０７５０	１０６２８	１１０５２．８	１．０３９９	１０６２８	１１７８９
ｂｅｒｌｉｎ５２	５２	０．０７７８	７５４２	８１３５．２	１．０７８７	７５４２	８９９９
ｅｉｌ１０１	１０１	０．６４２０	６２９	６７３．０	１．０６９９	６４４	７１４
ｃｈ１５０	１５０	３．３８１	６５２８	７０２６．４	１．０７６３	６６４２	７６１３

表３．数値実験の結果（２－ｏｐｔ法（方法［１］（ｋ＝１０）））

問題名	都市数	時間［秒］	最適解	準最適解	準最適解／最適解	最良の解	最悪の解
ａｔｔ４８	４８	０．０３１９	１０６２８	１１１３３．７	１．０４７６	１０６２８	１１９３４
ｂｅｒｌｉｎ５２	５２	０．０３９９	７５４２	８３５６．０	１．１０７９	７５４２	９３７３
ｅｉｌ１０１	１０１	０．１４８６	６２９	６８４．０	１．０８７５	６４６	７２８
ｃｈ１５０	１５０	０．４４７９	６５２８	７３９２．３	１．１３２４	６８００	８１４８

表４．数値実験の結果（２－ｏｐｔ法（方法［２］））

問題名	都市数	時間［秒］	最適解	準最適解	準最適解／最適解	最良の解	最悪の解
ａｔｔ４８	４８	０．０３４７	１０６２８	１１２２６．０	１．０５６３	１０６３８	１２３１６
ｂｅｒｌｉｎ５２	５２	０．０４０８	７５２４	８３２５．６	１．１０３８	７５４２	９３８５
ｅｉｌ１０１	１０１	０．１８０２	６２９	６７８．９	１．０７９２	６４８	７１８
ｃｈ１５０	１５０	０．５６４４	６５２８	７１６９．１	１．０９８２	６６８６	７８０９

６.３．考察

方法［１］で Candidate Set を求める時間は、ドロネーグラフの接続関係を利用することにより高速になることが確かめられた。また、方法［２］は、ドロネーグラフを基にしているので、高速である。しかし、軽減された計算時間は、ドロネーグラフを求める時間によって相殺され、効果があまりない。本研究では、ボロノイ図を定義にしたがって求めているので計算量が$O(n^2)$であり、時間がかかる。四分逐次添加法（$O(n)$）・再帰二分法（$O(n\log n)$）などのボロノイ図を求める効率的算法が提案されているので、これらの方法を利用すれば、ドロネーグラフを用いる効果が現れると思われる。表２～４の結果より Candidate Set を利用しての２－ｏｐｔ法は、全枝を対象とした２－ｏｐｔ法と比べて、方法［１］，方法［２］とも、かなり短い時間でほぼ同程度の精度の解を求めることができるということが示された。またａｔｔ４８（方法［１］）、ｂｅｒｌｉｎ５２（方法［１］、方法［２］）では最適解を求めることができた。

７．おわりに

本研究では、Candidate Set を利用することで発見的解法の高速化を図る方法を試みた。この方法で、２－ｏｐｔ法を用いてＴＳＰを解いたが、かなりの時間の軽減がなされた。２－ｏｐｔ法以外の発見的解法でも実験を行う必要はあるが、本研究により Candidate Set の有効性を確かめることができた。

本研究では、四分逐次添加法によってボロノイ図及びドロネーグラフを作成するプログラムの作成を試みていたが、完成まで至らず計算量の多い方法を採用することになってしまったことは残念である。

【参考文献】

［１］山本芳嗣、久保幹雄：“巡回セールスマン問題への招待”、シリーズ現代人の数理、朝倉書店, １９９７.

［２］Reinelt,G："The Traveling Salesman Computational Solutions for TSP Applications", Springer-Verlag, pp.64－72, １９９４.

［３］岡部篤行、鈴木敦夫：“最適配置の数理”、朝倉書店, １９９２.