遺伝的アルゴリズムによる巡回セールスマン問題の並列化を考慮した近似解法

荒木智行・山本富士男・巽　久行

情報工学科

An Approximate Solution Considering Parallelism for Traveling Salesman Problem Using Genetic Algorithm.

Tomoyuki ARAKI , Fujio YAMAMOTO and Hisayuki TATSUMI

Abstract

It is well known that traveling salesman problem (for short, TSP) is one of most important problems for optimization, and almost all optimization problems result in TSP. This paper describes on an effective solution of TSP using genetic algorithm.

The features of our method are summarized as follows: (1) By using division and unification method, a large problem is replaced with some small ones. (2) Smoothing method proposed in this paper enables us to obtain a fine approximate solution globally. Accordingly, demerits caused by division and unification method are decreased. (3) Parallel operation is available because all divided problems are independent of each other.

Keywords: traveling salesman problem, genetic algorithm, parallelism, smoothing method

１．まえがき

巡回セールスマン問題（以下，ＴＳＰと略す．）を効率的に解くアルゴリズムに関して古くから多くの研究がなされている[6]．巡回セールスマン問題は，組み合わせ最適化問題のうち最も基本的な問題の一つであり，また，ほとんど全ての組み合わせ最適化問題は，ＴＳＰに帰着されることは良く知られている．組み合わせ最適化問題の代表的なものとしては，大規模な基盤配線，配送計画，スケジューリング，X線による結晶実験，蛋白質の構造解析，ＶＬＳＩ設計などがある．ＴＳＰを効率良く解くアルゴリズムは，これらの問題に適用可能であり，重要である．

本論文では，遺伝的アルゴリズム（以下，ＧＡと略す．）を基礎とし，ＴＳＰを解くための効率の良い近似解法の提案を行っている．本論文で提案する手法の特徴としては，

(1) 分割統治法により，大規模な問題を複数の小規模な問題に分割し，問題規模の削減を行っている．
(2) 問題の分解を行うことにより起こる弊害として局所解に陥る問題がある．これを改善し，大域的に良い近似解を得るために，平滑法（Smoothing Method: 本論文で提案）を導入している．
(3) 分割された各問題は，互いに独立に計算することができるため，これらの処理を同時並行に行うことができる．このことは，本論文で提案する手法が，並列計算機向きのアルゴリズムとして利用できることを示している．

ＴＳＰには対称的（A点とB点の間の移動コストを，A->B，B->Aいずれの場合も同じとするもの）と非対称的（A->BとB->Aで移動コストが異なるもの）な問題がある．一般に対称的なＴＳＰは非対称的なＴＳＰよりも，その解法が困難である．本研究では対称的なＴＳＰを取り扱う．

対称的なＴＳＰの近年の成果としては，１９８１年に３２５都市（Balas and Christofides），１９９２年に２３９２都市（Padberg and Rinaldi），１９９３年に４４６１都市（Applegate, Bixby, Chvatal and Cool），１９９４年に７３９７都市(Applegate, Bixby, Chvatal and Cool)の問題が解かれており，Applegateらの１９９４年の結果が，著者らの知る限り，現在解かれている問題の中で最も大規模なものである[6]．これらのうち１９９３年以降にApplegateらにより解かれた問題は，超並列計算機を用いて解かれている．

本研究では，将来的に超並列計算機（例えば，神奈川工科大学のＳＲ２２０１）に実装するための予備的な考察としてシングルＣＰＵのワークステーション(SUN SPARCStation5: MicroSPARC Ⅱ / 110MHz)を用いてベンチマークデータ集ＴＳＰＬＩＢ[7)8)]に含まれる２３９２都市までの問題について実験的検証を行った．著者らの行った実験ではApplegateらが用いた計算機よりも，非常に低速な計算機を用いており並列処理も行っていないが，２３９２都市の問題に関して，実用時間内に実用十分な精度の結果を得ている．

２．巡回セールスマン問題（ＴＳＰ）

ＴＳＰとは，セールスマンがｎ個の都市の，どこかから出発して，ｎ個の都市すべてを一巡する際の最短（最小コスト）経路を求める問題である．都市数をｎとする

と，セールスマンの選べるすべての巡回路の数は $\frac{(n-1)!}{2}$ 通りあり，なんらかの制約条件を満たさない限りは，一般にその解法は，計算量的に困難な問題として知られている．

3．遺伝的アルゴリズム（ＧＡ）

ＧＡは，生物進化（選択淘汰，突然変異）の原理に着創を得たアルゴリズムである．ＧＡは交叉，選択，突然変異と呼ばれる三つの操作を順に繰り返し，最適な個体群を見い出すよう実行される．

4．ＧＡのＴＳＰへの適用

4.1　コード化および交叉の評価規範[1)]

ＴＳＰにおけるコード化および交叉の適切さを判定するのに，以下の四つの評価規範が考えられる．

［コード化の判定規範］

(1) 完備性 (completeness)
問題空間上の解候補はすべて染色体として表現できること．

(2) 健全性 (soundness)
ＧＡ空間上の染色体は，すべて問題空間の解候補に対応づけられること．

(3) 非冗長性 (nonredundancy)
染色体と解候補は一対一に対応づけられること．

［交叉の評価規範］

(4) 形質遺伝性 (charactor preservingness)
親の形質は交叉によって子に適切に継承されること．

以下の4.2項では，これらの規範を満たすコード化の方法を，また4.3項において交叉の方法（サブツアー交換交叉）について述べる．

4.2　コード化

都市名を０以上の整数で表現し，都市名からなる順序対を遺伝子とする．起点となる都市から巡回する順番に都市名を列挙した文字列を染色体とするコード化をパス表現と呼ぶ．このコード化においては，同一のツアーに右回りと左回りの二つの表現が可能である．

右回り：(0, 2, 3, 4, 5, 1, 7, 6, 8, 9)
左回り：(0, 9, 8, 6, 7, 1, 5, 4, 3, 2)

このコード化は完備性，健全性の評価規範を満たしている．さらに非冗長性を満たすために，片方を選び両方の表現を代表させるものとする．ただし，これらは文字列として異なるので，交叉を行う際には両方の組み合わせを考慮しなければならない．

本論文で用いる交叉では二つの親Ａ，Ｂに次の二通りの文字列の組み合わせを同時に考える．

・親Ａ（右回り）と親Ｂ（右回り）での交叉
・親Ａ（右回り）と親Ｂ（左回り）での交叉

なお，このとき，次の組み合わせによって生成される子は，上の操作で生成される子と同じである．

・親Ａ（左回り）と親Ｂ（右回り）での交叉
・親Ａ（左回り）と親Ｂ（左回り）での交叉

4.3　サブツアー交換交叉[1)]

解候補となる巡回路をツアーという．本論文では形質遺伝性の規範を満たす交叉として，遺伝する形質としてツアーの部分集合であるサブツアーを対象とするサブツアー交換交叉を用いる．パス表現では，サブツアーは染色体上の２点で囲まれた範囲を交換する２点交叉となる．しかしながら，２点交叉をＴＳＰに適用した場合には，一般的に致死遺伝子を持つ子を生成し得る．なぜならば，子の遺伝子が巡回路を形成するとは限らないからである．コード化は健全性を満たす必要があるので，今回採用するサブツアー交換交叉は，交換されるサブツアーに含まれる都市の集合が一致する場合に限り交叉するものである．

サブツアー交換交叉は，形質遺伝性に関して優れた交叉であるが，都市集合が一致するときのみ交叉が許されることから，そのようなサブツアーを見い出すための探索時間が大きいことが短所である．したがって，大規模な問題では，サブツアー探索コストを軽減するための工夫が必要である．

4.4　改良されたサブツアー交換交叉

本項では，従来から提案されているサブツアー交換交叉を改良し，実行時間の削減する方法について述べる．

4.4.1　サブツアー交換交叉の適用範囲の制限

サブツアー交換交叉では，図１のような場合，親Ａと親Ｂの下線部で都市集合が一致するため交叉ができる．このとき，それぞれの親の下線部以外も都市集合が一致する．しかし，どちらの交叉も生成される子は同じである．そこで，本論文では，都市数の半分より長いサブツアーは交叉させないことにする．これにより計算時間を短縮可能である．

親Ａ　($\underline{0,\ 1,\ 2,\ 3}$, 4, 5, 6, 7, 8)

親Ｂ　($\underline{0,\ 2,\ 3,\ 1}$, 8, 4, 6, 7, 5)

図１　サブツアー交換交叉の適用範囲

4.4.2　染色体の縮約による計算時間の短縮[1)]

世代交代が進行して集団に同じようなサブツアーを持つ個体が蓄積されてくると，サブツアー交換交叉によって同一の個体が生成されやすくなる．そこで以下に示す手順で親の染色体を縮約し，同一の染色体を持つ子を生成しにくくする．

(STEP1) 二つの親に共通するサブツアーを見つける．
(STEP2) 共通のサブツアーを一つの都市として，染色体の表現を変更し，縮約する．
(STEP3) 縮約された染色体による交叉を行い，子を生成する．
(STEP4) 生成された子の染色体に含まれる都市のうち，縮約されているものを元に戻す．

［例１］以下の二つの親を考える．

親Ａ (0, 9, 7, 8, 5, $\underline{3,\ 1,\ 2}$, $\underline{4,\ 6}$)
親Ｂ ($\overline{0}$, 5, 8, 7, $\underline{3,\ 1,\ 2}$, 9, $\underline{4,\ 6}$)

親Ａと親Ｂ：の間では，サブツアー[3, 1, 2]と[4, 6, 0]が共通しており，縮約の対象となる．□

世代交代が進むにつれて，個体間で共通するサブツアーは多くなる傾向になる．このサブツアーの縮約により，全体の計算量の削減ができる．

4.4.3　都市番号の加算による計算時間の短縮

サブツアー交換交叉は，二つの親の染色体の部分集合が一致したときのみ交叉が許される．このため，二つの親の染色体の部分集合をすべて確認する必要があり，こ

れを行うには多くの計算時間を要する．この問題を改善するため，都市番号の加算を用いた高速な方法を導入する．以下，例にしたがって述べる．

［例２］ ９都市で部分集合の要素数を４とし，選ばれた部分集合を下線で示す．

親A (0, 1, 2, 3, 4, 5, 6, 7, 8)
親B (5, 3, 0, 2, 1, 7, 4, 6, 8)

このときの都市番号の総和は，以下のようになる．

親A 0 + 1 + 2 + 3 = 6
親B 5 + 3 + 0 + 2 = 10

部分集合に含まれる都市が一致したときには必ず都市番号の総和は一致する．この場合総和が一致しないので，部分集合としては異なっている．

次に親Bの下線の部分を右に移動すると以下のようになる．

親B (5, 3, 0, 2, 1, 7, 4, 6, 8)

-5 + 10 + 1 = 6

このときの総和は，親Aの総和と一致する．　□

以上のように，部分集合内の都市番号の総和が一致した場合に限り，部分集合内の都市が一致することを確認するための処理を行う．これにより処理時間の短縮が可能となる．

4.5　相違選択法の提案

従来から提案されているサブツアー交換交叉では，次世代の親を経路の総距離が短い順に選択していくと，世代交代を重ねるにしたがい，それぞれの遺伝子の順番が似てきてしまい，局所解に陥ると，そこから退避できなくなる欠点があった．この問題を回避するためには，次世代の親の選択の際には，各世代の各々の個体に多様性を持たせることが重要となる．本項では，サブツアー交換交叉における親の選択法として相違選択法を提案する．

相違選択法とは以下の基準で次世代の親の選択をする手法である．

(1) 最初の親は，総距離が最小である遺伝子を選択する．
(2) 次世代以降の親は，それまでに選んだ遺伝子を除いて，前世代の親と比べ距離最小で，かつ，似てない遺伝子を選択する．

ただし，サブツアーが都市数の１／２以上同じである場合を似ていると判断し，それ以外の場合を似てないと判断する．

［例３］ ８都市の場合の親と子を以下に示す．

親　(0, 1, 2, 3, 4, 5, 6, 7)

子１ (0, 1, 2, 3, 4, 7, 6, 5)　…５個
子２ (1, 7, 5, 0, 6, 4, 3, 2)　…４個
子３ (0, 1, 2, 7, 5, 6, 4, 3)　…３個
⋮

子１と子２は親と比べ同じ経路を示すサブツアーが４(=8/2) 以上あるので親と似ている．しかしながら子３は親と似ていない．　□

従来のサブツアー交換交叉では，次世代に親の選択が，距離最小という基準だけでなされていたため，局所解に陥った場合，そこからの回避が困難であったが，ここで提案した相違選択法では，遺伝子間の関係として，似ているという概念を導入し，似てない子を次世代の親と選ぶことにより，局所解を回避するための多様性を作り出すことができる．

4.6　2チェンジ突然変異

単純ＧＡでは，突然変異は，ＧＡにおける操作とは通常見なされないが，一般的には，突然変異はＧＡの操作の一つとして用いられている．突然変異は，交叉だけでは生成できない個体を突然変異を用いて生成し，多様性を持たせることによって，探索空間を広げるために用いられる．本項では，本研究において採用した２チェンジ突然変異について述べる．

２チェンジ突然変異は，図２に示すように，遺伝子の中からランダムに二つの都市を選びだし交換する．これにより交叉だけでは生成できない個体を生成する．

(0, 1, 2, 3, 4, 5, 6, 7, 8)

下線部を交換
⇓
(8, 1, 2, 3, 4, 5, 6, 7, 0)

図２　２チェンジ突然変異

また２チェンジ突然変異の適用範囲については，図３に示すとおりである．ここで，各世代の親を距離の大きさにしたがい昇順にソートし，短い順に親０，親１，…とする．各世代においての親０の距離が，前世代の親０と同じだった場合には，収束状態にあるとし，それ以外を通常状態とする．突然変異は，本質的に良い染色体の状態を壊す可能性もあるので，良い染色体が多数存在すると考えられる親０から親４までは２チェンジ突然変異を実行しないものとする．親５以降は，通常状態では１０％，収束状態では１００％の確率で２チェンジ突然変異を実行する．

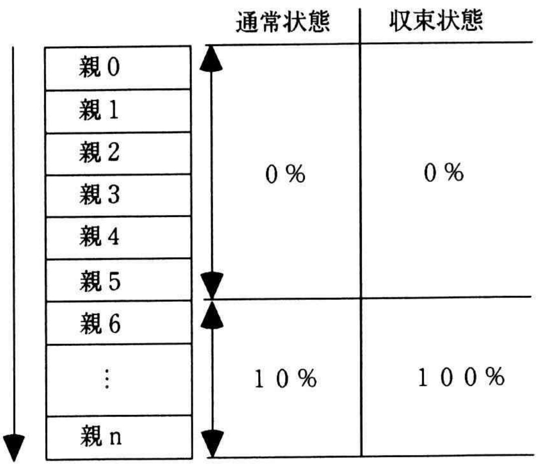

図３　２チェンジ突然変異の適用範囲

4.7　提案したＧＡの評価

前項までで提案したＧＡについて２０都市の問題での最適解収束率を表１に示す．

表１　２０都市における最適解収束率（1000回試行）

交叉確率[%]	最適解収束率[%]	
	親の数:１０	親の数:２０
３０	３０．８	４５．８
３５	３４．２	４９．４
４０	４４．２	５５．２
４５	４５．２	５３．６
５０	４１．８	５３．６
５５	５０．２	<u>５９．４</u>
６０	５２．８	５４．８
６５	５０．０	４９．４

表１からわかるように，交叉確率５５％，親の数１２のときが最も良い最適解収束率となっている．このことから以降，本論文での交叉確率は５５％，親の数は１２として評価を行う．また，この条件で２０都市および３０都市の問題に適用した際の実行時間の平均を表２に示す．

表２　２０都市，３０都市の実行時間（1000回試行）

都市数	実行時間（秒）
２０	１４
３０	６５

表２から３０都市程度の問題までは，１分程度の実行時間で終了することがわかる．なお，４０都市以上の問題に関しても実行してみたが，実行は数分程度では終らなかった．

以上の結果から，本論文でこれまでに提案したＧＡでは，最適解収束率，実行時間ともに，実用上満足のゆく結果とはいえない．これらのことから次節以降では，ＧＡ以外の方法と組み合わせたハイブリッドな解法について提案を行い，評価・考察を行う．

５．山上り法の導入

５．１　山上り法[1)]

ＧＡが大域的な探索に適しているのに対して，山上り法（以下ＨＣと略す．）は，可能性のある探索点の中で最も有望な点を選んで探索を進めてゆく，局所的な探索に適した方法である．ＨＣでは，部分順路を選び，それらを逆転するという操作を繰り返す．逆転により距離が短縮されない場合には逆転は行わない．

［例４］山上り法の例を図４に示す．この場合は逆転により距離が短縮される．

(0, [4, 1, 5], 3, 7, 2, 6)
逆転
⇓
(0, [5, 1, 4], 3, 7, 2, 6)

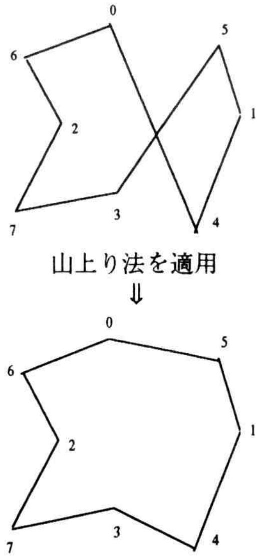

図４　山上り法

□

５．２　ＧＡとＨＣの組み合わせによる評価

前項で述べたＨＣを前節で提案したＧＡの手法と組み合わせて評価を行った．評価は最初にＨＣにより前処理を行い，ＧＡを行い，さらにＨＣにより後処理を行った．評価には４.７項で用いたのと同様な２０都市と４０都市のデータを用いた．また交叉確率，親の数も同様のものを用いた．また評価の終了は，世代の中で最短の距離を示す染色体の示す距離が１０世代同じであった場合に終了するものとする．結果を表３に示す．

表３　ＨＣ＋ＧＡ＋ＨＣの実行結果（1000回試行）

都市数	最適解収束率[%]	実行時間[秒]
２０	１００	７
４０	９７．９	１４

２０都市の場合，ＧＡのみの場合と比較すると最適解収束率が５９．４％から１００％へ，また実行時間も１４秒から７秒に短縮された．また４０都市の場合でもＧＡのみの場合には数分程度以上の時間を要していたものが１４秒程度で求められるようになった．しかしながら，表３には示してないが，５０都市以上の規模の問題に対しては，一つも最適解が得られず，最適解収束率は０％であった．これらのことから，本手法で求められる問題の規模は４０都市程度であると考えられる．

６．分割統治法の導入と平滑法の提案

５節の結果からＧＡとＨＣの組み合わせによる解法では４０都市程度が実用上限界であることがわかった．本節では，さらに大規模な問題に対して，厳密解ではなくとも実用上有効な程度の近似解を求める手法を考察する．本節では，最初に与えられた問題を，小規模な複数の問

題に分割するために分割統治法（以下，ＤＵと略す）を導入する．しかしながら分割統治法では，問題の規模は小さくなるが，分割された問題の一つ一つで最適化を行うため，最後に統合された解は，局所解の集合でしかなく，大域的に見ると，必ずしも最適解に近くない場合がある．この問題を解決するために，分割された領域間のデータの交流をする手法 - **平滑法**（**Smoothing Method**）- を提案し導入する．

6．1　分割統治法の導入

本項では，本研究で用いたＤＵについて述べる．

6．1．1　分割処理

図５に分割法を示す．この図の０～４は都市を表し，都市の分布する領域が縦長なら横に分割し，横長なら縦に分割する．図５の場合，横長なので領域の中央に位置する都市２を中心に領域を半分に分割する．

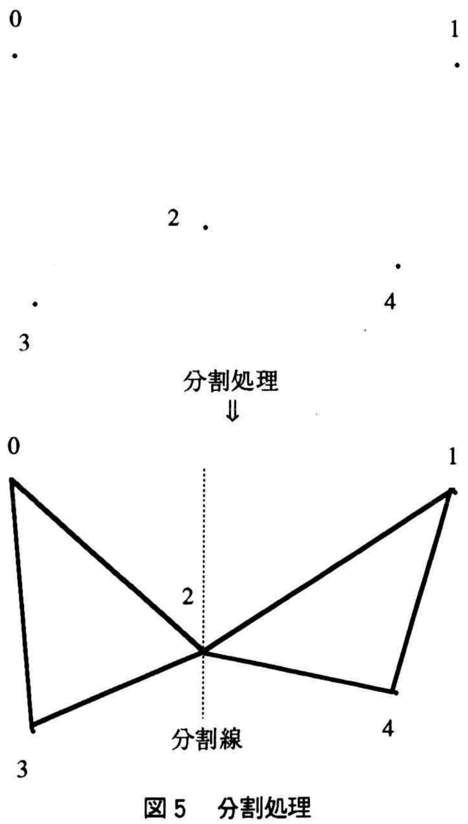

図５　分割処理

6．1．2　統治処理

図６に統治法を示す．統治するときには，経路（0-1）か経路（3-4）のどちらかが選ばれ，再び順回路となる．
(1)（経路（0-2-1）の長さ）－（経路（0-1）の長さ）
(2)（経路（3-2-4）の長さ）－（経路（3-4）の長さ）
図６の場合では二つの場合を比較し，差が大きい方が選択されるので，(1)の場合が選択され統治処理が行われる．

6．2　ＤＵ，ＧＡ，ＨＣの組み合わせによる解法

ＤＵ，ＧＡ，ＨＣの組み合わせによる解法では，都市の規模をＤＵにより分割（具体的には，5.2項の結果から４０都市以下の規模に分割）して，それぞれの分割された領域内でＧＡとＨＣの組み合わせによって最適化を行い，すべての領域内での最適化が終了したのち，それらを統治する．しかしながら，この手法では，分割された各々の領域内で局所的な最適解は求められるが，領域間の交流が無いため，大域的な最適解には近づかない場合が多いのが問題点として残る．

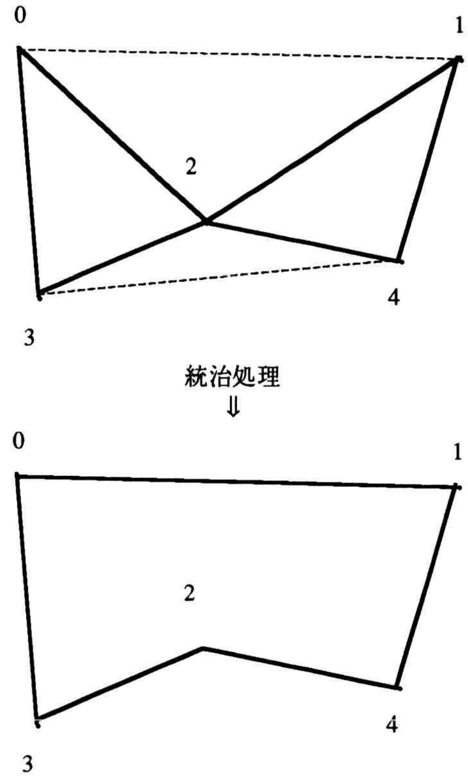

図６　統治処理

6．3　平滑法の提案

6.2項で挙げられた問題を緩和する手法として，本項では**平滑法**（以下，ＳＭと略す）を提案する．平滑法とは，図７のようにＤＵによって分割された分割線を中心にして，ある大きさ（都市数が４０以下になる大きさ）の領域を設定し，その領域内のすべての都市に対してＧＡとＨＣを組み合わせた手法を再度行う方法である．この手法により，ＤＵを適用することによって生じた問題点が改善されることが図７からも容易にわかる．

6．4　評価

ＤＵの導入，ＳＭの導入による評価をベンチマークデータ集ＴＳＰＬＩＢ[7)8)]のデータを用いて行ったのが表４である．使用したデータは対称ＴＳＰのユークリッド距離によってコストが与えてあるものを用いた．

表４の結果から，ＤＵの導入により最適解に対して，精度は都市数が増加するにしたがい悪化する傾向にあることがわかる．これは領域の分割数が都市数の増加に伴い多くなることからも当然の結果である．これに対して，ＳＭを導入しない場合の最悪な精度の解は，ｐｒ２３９２の場合の１．１５倍であるが，ＳＭ導入後は，１．０６倍となり，ＳＭにより９％程度の改善が見られる．このことからＤＵによって生じた問題点である局所解による精度の悪化がＳＭの導入により大きく改善されたことがわかる．また，実行時間に関してもｐｒ２３９２の場合で約４１分であり，適用するアプリケーションによっても異なるが，実用許容範囲内の時間であると考えられる．

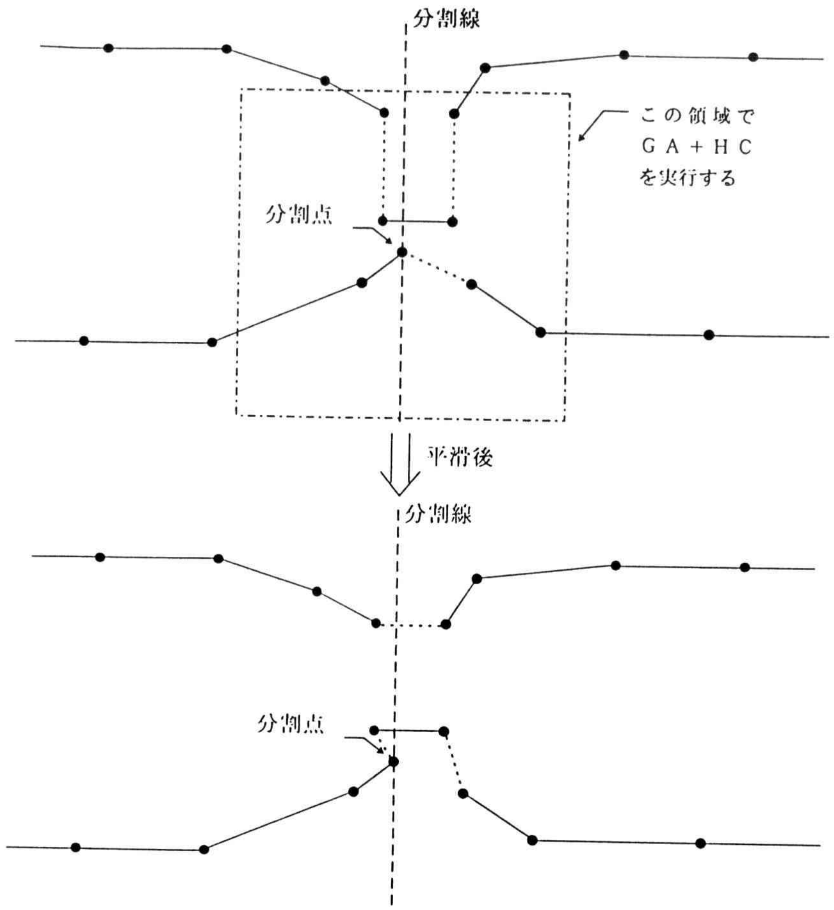

図７　平滑法

表４　評価結果

都市数	最短距離	平滑法なし	平滑法あり	平滑法ありの実行時間［秒］
eil５１	４２６	４３４ （１．０２）	４２８ （１．００）	２３
eil７６	５３８	５６５ （１．０５）	５５１ （１．０２）	５２
kroA１００	２１２８２	２３６５３ （１．１１）	２１３０８ （１．００）	５３
kroD１００	２１２９４	２３６４６ （１．１１）	２２１１９ （１．０４）	６０
eil１０１	６２９	６８２ （１．０８）	６４３ （１．０２）	９１
lin１０５	１４３７９	１６３７９ （１．１４）	１４５０４ （１．０１）	５０
pcb４４２	５０７７９	５７７０７ （１．１４）	５２８８３ （１．０４）	４１７
pr２３９２	３７８０３２	４３３７７３ （１．１５）	４０１３９５ （１．０６）	２４６３

（　）内は総距離／最短距離

７．むすび

本論文では，ＴＳＰの解法についての考察を行った．ここでの考察は，最初にＧＡとＨＣの組み合わせによる単純な解法の高速化を行い，評価し，その限界となる問題規模の同定を行った．次に大規模な問題をＤＵを導入することにより，先に高速化を行ったＧＡとＨＣの組み合わせによる手法の大規模な問題への適用を行った．その際，ＤＵによる弊害として生じる領域間の交流の遮断による解の局所化が明らかにされた．これを解決する手法としてＳＭを導入し，領域間の交流をある程度可能にし，ＤＵによる弊害の改善を行い，その有効性を実験的に確認した．

本論文での報告範囲では，単に大規模な問題が近似的に実用時間内に実用可能な精度で求められることが実験的に示されるに留まっている．しかしながらＤＵによって生じる弊害がＳＭの導入により緩和される事実から，並列計算を導入することにより，大幅な計算時間の短縮が図れる可能性が見い出せた．ＤＵによって分割された複数の問題やＳＭの対象となる複数の問題は，互いに独立な問題であり，同時平行に計算することができる．したがって，分割された問題数のプロセッサーが用意できれば，全体の計算時間は，荒削りな見積りでは，４０都市のＴＳＰを解く時間の数倍程度の時間で終らせることができると考えられる．このプロセッサ数を具体的に求めてみると，本論文で扱ったｐｒ２３９２で約６０プロセッサであり，現在の世界記録である７３９７都市の問題で約１８５プロセッサとなる．この数は，現在の一般的な超並列計算機の規模からすると現実離れした数ではない．今後は神奈川工科大学のＳＲ２２０１を用いて大規模な問題に，本論文の手法を適用する予定である．

最後に，ＴＳＰＬＩＢの最新情報等，貴重な情報を提供していただいた東京商船大学の久保幹雄先生に感謝致します．また，本研究は情報工学科における一連の卒業研究の指導を通して進められたものであり，実験を進めるにあたり卒業生の倉沢司君，深谷大吉君の多大な努力があったことを付記し，深謝致します．

参考文献

1) 北野宏明編：遺伝的アルゴリズム，産業図書(1993).
2) 北野宏明編：遺伝的アルゴリズム2，産業図書(1995).
3) 久保幹雄：巡回セールスマン問題への招待I，日本オペレーションズリサーチ学会誌，Vol. 39，No.1，pp.25-31(1994).
4) 久保幹雄：巡回セールスマン問題への招待II，日本オペレーションズリサーチ学会誌，Vol. 39，No.2，pp.91-96(1994).
5) 久保幹雄：巡回セールスマン問題への招待III，日本オペレーションズリサーチ学会誌，Vol. 39，No.3，pp.156-157(1994).
6) 山本芳嗣，久保幹雄：巡回セールスマン問題への招待，朝倉書店(1997).
7) http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html
8) http://www.ing.unlp.edu.ar/cetad/mos/TSPBIB_home.html