HOKKAIDO UNIVERSITY

Title	局所組み合わせ分割統治法による大規模TSPの解法
Author(s)	石川, 和典; 鈴木, 育男; 山本, 雅人 他
Citation	情報処理北海道シンポジウム講演論文集, 2010, 43-48
Issue Date	2010-10-02
Doc URL	https://hdl.handle.net/2115/51037
Rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。
Type	journal article
File Information	Hokkaidosympo20104348.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

情報処理北海道シンポジウム 2010

局所組み合わせ分割統治法による大規模TSPの解法

石川和典* 鈴木育男 山本雅人 古川正志
(北大情報科学)†

1 はじめに

巡回セールスマン問題（Traveling Salesman Problem, TSP）は，都市の集合と各2都市間の移動コストが与えられたとき，もっとも移動コストの総和が小さいハミルトン閉路を求める問題である．TSPはオペレーションズ・リサーチ分野においてもっとも研究されている問題のひとつである．TSPは基盤配線や穿孔，X線結晶構造解析など様々な工学的応用分野が存在し，基盤の大規模化など，近年TSPの応用分野が求める最適化規模は大きくなってきている．しかしながら，TSPは計算複雑性理論においてNP困難と呼ばれる問題クラスに属しており，このクラスに属する問題を決定性チューリングマシンによって多項式時間内で解くことは不可能であると予想されている．実際，TSPを問題サイズnに対し$O(n)$のオーダーで厳密解を求める解法は今のところ存在しない．このため，大規模TSPに対して実用的な時間内で厳密解を求めることはほぼ不可能であり，むしろ実用的な時間内で許容できる解精度を得るためにこれまでに多くの近似解法が研究されてきた．

本研究も大規模TSPに対して実用的な解精度を高速に得ることを目的とし，TSPから複数の小さな問題を生成する新たな分割統治法を提案する．提案手法は，元の問題の複数の部分的な経路情報から一つの小問題を生成する．任意の手法によりすべての小問題を繰り返し最適化することで，最終的に元の大規模問題は大域的に最適化される．数値計算実験では，各小問題の最適化に局所クラスタリング組織化法（Local Clustering Organization, LCO）を採用する．

2 関連研究

2.1 近似解法

これまで，遺伝的アルゴリズム（Genetec Algorithm,GA），タブーサーチ（Tabu Search, TS），自己組織化マップ（Self Organizing Map, SOM），Lin-Kernighan法など既存の近似解法の応用手法が大規模TSPに対し適用されてきた．これらの既存手法はもともと中小規模TSPに対して開発，適用されてきたものであり，大規模TSPに適用するにあたり，さまざまな仕様変更，チューニング，並列化の研究がなされている [1, 2, 3].

2.2 分割統治法

分割統治法（Divide and Conquer, D&C）は，大規模な問題を複数の小さな問題に分割し，小問題それぞれを解決することで最終的に元の問題を解決するという考え方である．分割統治法を大規模TSPに適用する従来の研究では，分割統治法の導入により高速に解を得ることができたと報告されている [4, 5]．しかしながら従来の分割統治手法は，ほとんどがTSPの都市座標の情報を利用して小問題を生成している．よって，これらの手法は都市の座標が与えられるユークリッドTSPには適用できるが，そうではない一般的なTSPには適用できないという欠点がある．

3 局所クラスタリング組織化法

局所クラスタリング組織化法（Local Clustering Organization, LCO）はTSPの近似解を高速に求めるために考案された最適化手法である [6]．この手法では解の局所的な最適化の繰り返しが最終的に大域的な最適化をもたらす．LCOの最適化はSOMのシナプス学習則をもとに考案されたが，SOMとは違い，LCOは一般的なTSPにも適用可能である．LCOではSOMのようにニューロンが定義され，TSPに適用する際，ニューロンは各々の都市と見なされ円環状に結合される．このニューロン結合の順序関係は，任意のクラスタリング手法により，繰り返し局所最適化されていく．

LCOはTSPだけでなく，n-TSPやVehicle Routing Problem，Jobshop Scheduling ProblemやQuadratic Assignment Problemなど，様々な組み合わせ最適化問題に適用され，その有効性が報告されている [7].

3.1 クラスタリング手法

クラスタリング手法は，局所的なニューロンの結合関係を最適化する手法である．本節ではLCOをTSPに適用する際よく用いられるクラスタリング手法である単純交換法，逆位交換法，平滑法について簡単に説明する．

クラスタリング手法を適用するために，LCOは次のようなパラメータを用いる．cはクラスタリング手法が適用される範囲の基点となるニューロンである．rはクラスタリング半径と呼ばれ，クラスタリング手法が適用される範囲の大きさである．N_cはcの近傍ニューロンの集合であり，ニューロンの円環状トポロジーの上で，cから前後r個のニューロンを含む．

LCOをTSPに適用する際，以下の三つのクラスタリング手法がよく用いられる．

k-ishikawa@complex.eng.hokudai.ac.jp
札幌市北区北14条西9丁目北海道大学大学院情報科学研究科

- 単純交換法（Simple Exchange Method, SEM）
  c を基点とし，N_c に含まれるニューロンに対し解コストが下がる場合のみ結合順序を交換する．
- 逆位交換法（Inverse Exchange Method, IEM）
  c を基点とし，c と $c' \in N_c$ に対し 2opt を行う．すなわち，コストが下がる場合のみ c から c' までの結合順序を逆順にする．
- 平滑法（Smoothing Method, SM）
  N_c 内のニューロンすべてに対し，2opt を総当たりで適用する．

これらのクラスタリング手法は，各ステップにおいて任意の割合でランダムに選択される．

3.2 アルゴリズム

LCO を TSP に適用する際の一般的な手順を以下に示す．

1. 都市数と同じ数のニューロンを定義し，ランダムな順序で円環状に結合する．
2. クラスタリング半径 r の値を更新する．
3. クラスタリング基点ニューロン c をランダムに選択する．
4. c とその近傍 N_c にクラスタリング手法を適用する．
5. 終了条件を満たせば終了し，そうでなければ 2. へ戻る．

LCO においては，2. から 5. までを 1 ステップとする．

一般的に，r の値は最初は $r = 1$ などの小さな値に設定され，都市数の半分 $n/2$ などの大きな値へ各ステップ毎に増加させる．

4 提案手法

分割統治法の考えに基づき，提案手法は現在の巡回経路から複数の小さな巡回経路を生成する．問題分割に都市の座標情報は利用しないため，提案手法は一般的な TSP に適用化可能である．

提案手法において，“problem” とは与えられた TSP の都市の集合を表す．“tour” は TSP のひとつの解である．“sub-problem” は分割により得られる，problem の部分集合である．“sub-tour” は tour の一部分であり，tour のなかのいくつかの経路を削除することにより得られる．“child-problem, CP” は複数の sub-problem に含まれる都市の集合であり，仮想的に TSP とみなす．CP の解を “CP-tour” と呼ぶ．

CP は後述する 2 つの例外を除き，一般的な TSP とみなすことができる．生成された CP-tour を繰り返し最適化することで，最終的に tour の大域的な最適化が実現される．提案手法は，tour の分割により複数の CP-tour を生成する．CP-tour は各 CP の初期解となるため，各 CP を最適化する手法は局所探索法であることが想定される．図 1 に提案手法の概要を示す．

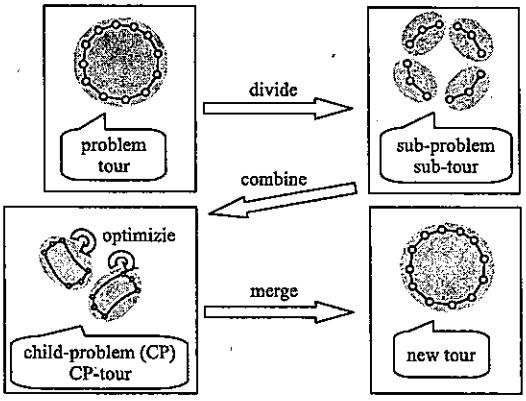


Fig. 1 Overview of the proposed D&C method

次節以降では，CP および CP-tour の生成，最適化および統合の具体的な手順を述べる．

4.1 CP および CP-tour の生成

CP は複数の sub-problem から構成される．CP は初期解としての CP-tour をもち，この初期解は複数の sub-tour を組み合わせることで構成する．sub-tour は，元の tour の連続した一部分である．

CP の初期解を構成するために，まず元の tour を分割し，複数の sub-problem と sub-tour を生成する．tour に含まれる複数の経路を除去して得られる複数の部分的な経路を sub-tour とする．各 sub-tour は同じ長さになるよう，tour 内でランダムに選ばれた経路を起点に除去する経路を選択する．sub-tour の長さを均等にすることが CP-tour を構成するために最適な戦略であるかはまだ未検証であり，このことは今後の課題のひとつでもある．

複数の sub-tour が生成されると，任意の数の sub-tour は仮想的に定義された経路で一つの円環状の経路となるよう繋げられる．この仮想的な経路を以降では virtual edge と呼ぶ．各 sub-tour は，virtual edge によりその両端が繋がり，一つの円環状の経路となり，これが一つの CP の初期解となる．一つの CP-tour を構成する sub-tour の組合せはランダムに選択する．

大域的な最適化を実現するためには，一つの CP-tour 内で複数の sub-tour を組み合わせる必要がある．なぜなら，各 sub-tour それぞれを単独に最適化するだけでは最適化の対象範囲は sub-tour の大きさを超えることができないからである．複数の sub-tour を組み合わせ同時に最適化を施し，組み合わせを変えながらこのような最適化を繰り返すことで，sub-tour のサイズを超える最適化が実現可能である．このことは 5 章での実験結果からも示される．

生成する CP の数 s，組み合わせる sub-tour の数 u，除去する元の tour 内の経路の数 d の間には式 1 のような関係がある．

$$d = s \cdot u \tag{1}$$

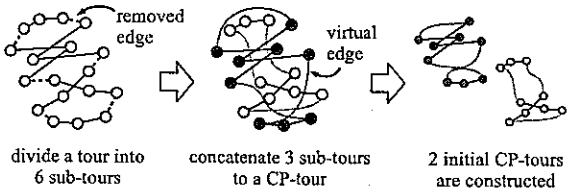


Fig. 2 An example of constructing initial CP-tours

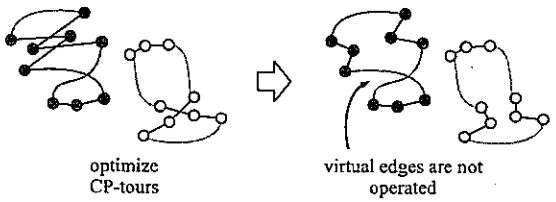


Fig. 3 An example of optimizing CP-tours

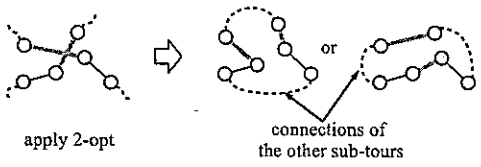


Fig. 4 Two variations of edges created by 2opt

Fig.2 は $s = 2, u = 3, d = 6$ としたときの，CP-tour の構成例である．

4.2 CP-tour の最適化

初期 CP-tour から開始し，CP-tour は任意の手法により最適化される．CP はつぎの二つの例外を除き一般的な TSP とみなすことができる．

一つ目は，CP-tour を構成する際に定義した virtual edge は，最適化の対象にすることができない（Fig.3）．なぜなら，virtual edge は便宜上定義したものであり元の tour に含まれる経路とは関係がないからである．さらに，virtual edge の位置を保存しておくことで CP-tour 最適化後の統合処理を単純に行うことが可能となる．

二つ目の例外は，最適化手法が sub-tour の順序関係に依存する操作を行う場合に生じる．この場合各 sub-tour の，元の tour 上での順序関係を管理する必要がある．例えば，“2-opt” と呼ばれる操作は，二つの経路を総コストが下がる場合にのみ交換する操作である．この操作は Fig.4 に示されるように，sub-tour の結合関係が分からなければ交換後の経路は一意に定まらない．

これら二つの例外を除き，CP-tour は TSP の解と見なし最適化することができる．

4.3 CP および CP-tour の統合

全ての最適化された CP-tour は一つの tour に統合され，より最適化された新たな解を得る．CP-tour を統合するために，これらはまず virtual edge を除去することにより分割される．この分割により，d 個の最適化された sub-tour が得られる．これらの sub-tour を，最適化前に除去した tour の経路を復帰させることにより繋げる．この操作により一つの円環状の巡回経路が得られ，これを新しい tour とする．CP-tour を統合する例を Fig.5 に示す．

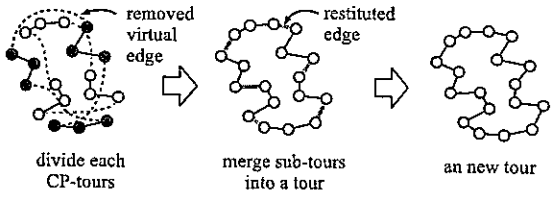


Fig. 5 An example of merging CP-tours

4.4 アルゴリズム

提案手法により TSP を最適化する一般的な流れを以下に示す．ここで，終了条件とパラメータ s, u, d は予め設定されているものとする．

1. 任意の方法により，初期解を生成する．
2. 現在の tour から，d 個の sub-tour を生成する．
3. sub-tour を u 個ずつランダムに組み合わせ，s 個の CP-tour を生成する．
4. 任意の最適化手法を各 CP-tour に適用する．
5. 全ての CP-tour を一つに統合し，これを新たな tour とする．
6. 終了条件を満たせば終了し，そうでなければへ戻る 2.

提案手法においては 2 から 6 を 1 ステップとする．

5 パラメータ特性評価実験

提案手法のパラメータのうち，CP の数 s，一つの CP-tour 内で結合される sub-tour の数 u と tour を分断する数 $d = s \cdot u$ は特に提案手法の性能に大きな影響を与えると予想できる．本節の実験では，提案手法を大規模 TSP に適用させた際にこれらのパラメータがどのような影響をもたらすかを評価する．実験結果で得られる結果は，提案手法のパラメータをチューニングする際の指標となる．

5.1 実験条件

本実験では二つの TSP のベンチマーク問題 “pla85900”, “lrb744710” を用いる．これらの問題は Web サイト “TSPLIB[1] ”, “The Traveling Salesman Problem[2] ” に掲載されているベンチマーク問題のうち，最も多い都市数をもつ問題である．pla85900 は 85900 都市，lrb744710 は 744710 都市の問題である．

[1] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/
[2] http://www.tsp.gatech.edu/

Table 1 Parameters for LCO to optimize CP-tours

clustering method (selection rate)	SEM, (40%)	IEM, (40%)	SM (20%)
r_i	min { 1.01^{i-1}, $n_s/2$ }		
termination conditions	running $n_s^{1/2}$ steps		

Table 2 Relation between n, s, and n_s

$n = 85900$ (pla85900)

	$s = n^{1/2}$	$s = n^{1/3}$	$s = n^{1/4}$
s	293	44	17
n_s	293	1952	5052

$n = 744710$ (lrb744710)

	$s = n^{1/2}$	$s = n^{1/3}$	$s = n^{1/4}$
s	862	90	29
n_s	863	8274	25679

全ての CP-tour は LCO により最適化される．LCO のパラメータは表 1 のように設定される．ここで n_s は一つの CP に含まれる都市数を表し，r_i は LCO における i ステップ目のクラスタリング半径を意味する．

提案手法において，s の値は都市数 n を用いて $s = n^{1/2}, n^{1/3}, n^{1/4}$ と変化させる．n の指数オーダーを用いたのは，予備実験により s の値を $s = 1000$ などと大きな値に設定すると良い結果が得られなかったためである．具体的な n の値に対応する s と n_s の値は，表 2 のとおりである．

u の値は，$u = 1, 2, 4, 6, 8$ の中で変化させる．u も予備実験により 10 以上の大きな値に設定しても良い結果が得られないことが示されている．

提案手法の終了条件は，pla85900 問題に対しては計算開始から 1 時間の経過，lrb744710 問題に対しては計算開始から 5 時間の経過とする計算にはプロセッサ Intel(R) Core(TM)2 Quad CPU Q9550 2.83GHz の 1 コアのみ用いる．

5.2 実験結果

まず，CP の数を $s = n^{1/2}$ と固定し，結合数を $u = 1, 2, 4, 6, 8$ と変化させたときの結果を示す．図 6 は pla85900 問題を解いたときの，図 7 は lrb744710 を解いたときの，解コストの計算時間による変化をグラフにしたものである．どちらも x 軸は計算時間，y 軸は解コストを対数軸で表している．

どちらの問題を解いた場合でも，$u = 1$ としたとき，コストは高速に収束しているが最終的な収束値は $u \neq 1$ のときと比べ高い値となっている．$u \neq 1$ としたときは，解の収束する値はほぼ一致し，u の値が増加するにつれ解収束の速度は遅くなることが示された．これは，CP-tour を最適化する LCO が行う最適化の操作（2 都市の順序交換と 2-opt）が，最低でも二つの sub-tour を組み合わせることで大域的な最適化を行えることを示している．この結果から，CP-tour を LCO により最適化する場合は，$u = 2$ と設定することが適切であることが分かった．図 8 は $u = 1$ としたときの，lrb744710 問題において最終的に得られた解の一部を拡大したものである．図 9 は同様に $u = 2$ としたときの，図 8 と同じ部分を拡大したものである．$u = 1$ としたときは完全に解が収束しているにもかかわらず，解には多くの交差辺が残り，sub-tour の長さを超える範囲での最適化が実現されていないことが示された．一方 $u = 2$ としたときは，全てではないものの，多くの交差辺が解消され大域的な最適化がなされ

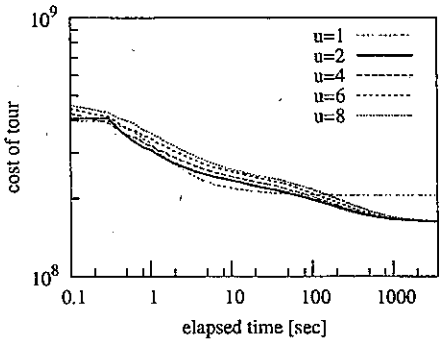


Fig. 6 Cost transition of pla85900 ($s = n^{1/2}$)

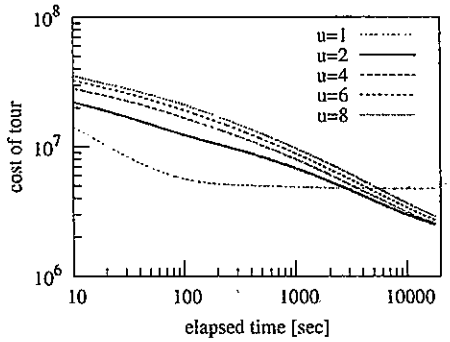


Fig. 7 Cost transition of lrb744710 ($s = n^{1/2}$)

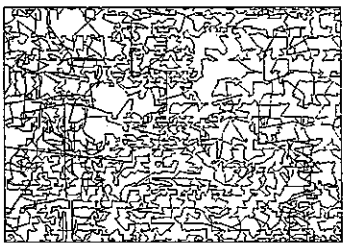


Fig. 8 A part of the lrb744710 tour ($s = n^{1/3}, u = 1$)

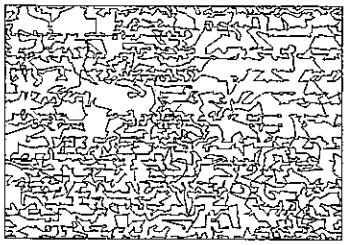


Fig. 9 A part of the lrb744710 tour ($s = n^{1/3}, u = 2$)

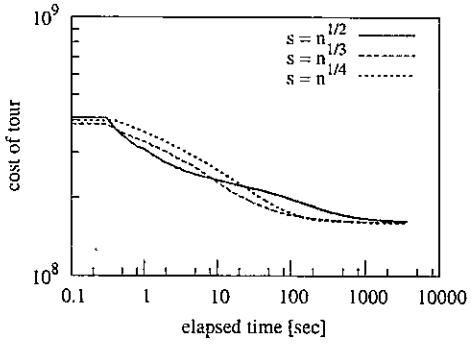


Fig. 10 Cost transition of pla85900 ($u = 2$)

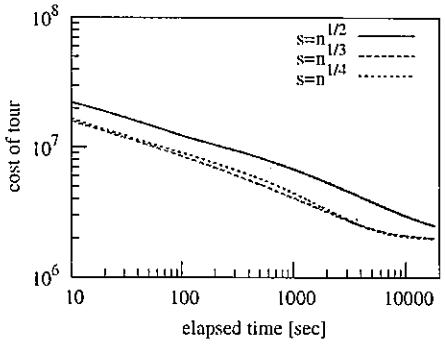


Fig. 11 Cost transition of lrb744710 ($u = 2$)

ていることが分かる．

つぎに，CP の数を $s = n^{1/2}, n^{1/3}, n^{1/4}$ と変化させ，結合数を $u = 2$ と固定させたときの結果を示す．図 10 は pla85900 問題を解いたときの，図 11 は lrb744710 を解いたときの解コストと計算時間の関係を表したものである．pla85900 問題では，計算開始から約 10 秒までは $s = n^{1/2}$ としたものがより高速に解を収束させ，それ以降は $s = n^{1/3}$ としたものが最も高速であった．最終的に収束する解コストの値はほぼ一致した．lrb744710 問題では，pla85900 問題のときとは異なり，時間帯によって優位性を見せるパラメータが変わるという結果は得られず，$s = n^{1/3}, n^{1/4}$ としたものが高速に解を収束させた．s の値を動的に変化させる方法も考えられるが，これについては今後詳細な調査が必要である．この結果から，s の値を固定させる場合は $s = n^{1/3}$ 程度の値に設定すべきであることが分かった．

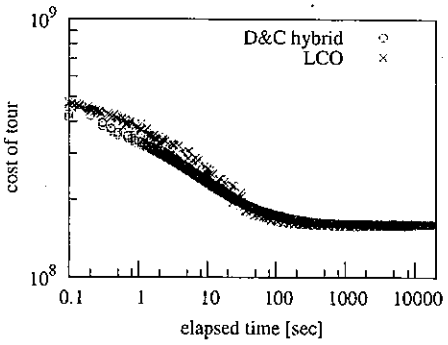


Fig. 12 Time variations of costs of pla85900 tours

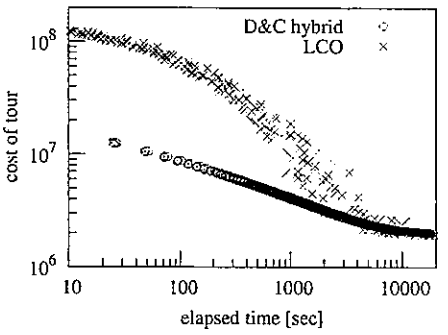


Fig. 13 Time variations of costs of lrb744710 tours

6 LCO との比較実験

この実験では，提案手法が TSP の解を求める速度と得られる解の精度にどのような影響を与えるか評価する．5 章で用いた提案手法により得られる CP-tour を LCO により最適化する手法（D&C hybrid）と，提案手法を用いない LCO をベンチマーク問題に適用する．

6.1 実験条件

TSP ベンチマーク問題は 5 章にて行われた実験と同じ pla85900 と lrb744710 を用いる．提案手法のパラメータは，5 章の実験により得られた結果を参考に，$s = n^{1/3}$，$u = 2$ と設定する．提案手法により生成される CP-tour を最適化する LCO は，表 1 のとおりにパラメータを設定する．比較手法としての LCO の設定は，終了条件を除いて表 1 のとおりである．どちらの手法も，終了条件は計算開始から 5 時間の経過とする．

各問題に対して，それぞれ 6 回の試行が行われる．

6.2 実験結果

図 12, 13 は，pla85900 問題と lrb744710 問題を解いたときの，全 6 回の試行においての解コストと計算時間の関係を示したものである．x 軸は計算時間，y 軸は解コストを表し，どちらも対数軸である．

どちらの問題に対しても，提案手法を用いたものは計算の初期段階において高速に解コストを減少させている．

Table 3 Accuracies of obtained solutions

pla85900	aver.	best	worst
D&C hybrid	1.120	1.115	1.124
LCO	1.109	1.108	1.111

lrb744710	aver.	best	worst
D&C hybrid	1.219	1.215	1.226
LCO	1.211	1.196	1.232

この初期段階における速度差は，問題規模が大きくなるほど広がりを見せている．

最終的に得られた解精度の，6回の試行での平均値，最良値，最悪値を表3に示す．ここで，解精度の値は，得られた解が現在世界的に知られている pla85900，lrb744710 問題の最良の解コストと比較し何倍の大きさであるかを表す．この結果から，提案手法を用いることで解精度は1%から2%減少することが明かとなった．

7 おわりに

従来の分割統治法とは異なる，一般的な TSP にも適用可能な分割統治手法を提案した．提案手法は任意の TSP の解から，任意の数の小さな TSP を生成するものである．生成された小問題の最適化を繰り返すことで，元の問題の大域的な最適化が実現される．

実験では，適用する TSP のサイズが大きくなるほど近似解を求める速度において提案手法が優位性を示した．このことは問題を分割することの利点であると言える．提案手法は問題を分割することにより比較的小さな最適化範囲で大域的最適化を実現可能にする．一般的に TSP の解法の計算コストは問題サイズが増加するほど大きくなる．したがって，本稿では組み合わせる解法として LCO を採用したが，提案手法は他の解法に対しても計算スピードを増加させることができると予想される．他の解法を組み合わせた時の提案手法の影響を評価することは今後の課題の一つである．

しかしながら，提案手法を用いることで最終的に得られる解精度が 1 ～ 2% ほど低下することが実験により示された．組み合わせる最適化手法と得られる解精度との関係性や，より効率的な分割法の考案も今後の重要な課題である．

参考文献

[1] B. Fritzke and P. Wilke. FLEXMAP-a neural network for the traveling salesman problem with linear time and space complexity. *Proc. of IJCNN, Singapore*, pp. 929–934, 1991.

[2] D. Johnson and L. McGeoch. Experimental analysis of heuristics for the STSP. *The traveling salesman problem and its variations*, pp. 369–443, 2004.

[3] D. Applegate, W. Cook, and A. Rohe. Chained Lin-Kernighan for large traveling salesman problems. *INFORMS Journal on Computing*, Vol. 15, No. 1, pp. 82–92, 2003.

[4] S.A. Mulder and D.C. Wunsch. Million city traveling salesman problem solution by divide and conquer clustering with adaptive resonance neural networks. *Neural Networks*, Vol. 16, No. 5-6, pp. 827–832, 2003.

[5] C.L. Valenzuela and A.J. Jones. Evolutionary divide and conquer (I): A novel genetic approach to the TSP. *Evolutionary Computation*, Vol. 1, No. 4, pp. 313–333, 1993.

[6] Masashi Furukawa, Michiko Watanabe, and Yusuke Matsumura. Local clustering organization (lco) solving a large scale of tsp. *The 4th International Conference on Advanced Mechatronics -Toward Evolutionary Fusion of IT and Mechatroincs-(ICAM'04).*

[7] Nobuhiro Sakamoto, Ikuo Suzuki, Michiko Watanabe, Masahito Yamamoto, and Masashi Furukawa. A new solution for vehicle routing problem using local clustering organization. *Proceeings of the 20th National Conference of Australian Society for Operations Research and the 5th International Intelligent Logistics System Conference*, pp. 83.1–83.14, 2009.