

遺伝的アルゴリズムとアアントコロニー最適化手法による巡回セールスマン問題の解法

Solving the Traveling Salesman Problem through both of Genetic Algorithms and Ant Colony Optimization

高橋 良英

Ryouei Takahashi

八戸工業大学

Hachinohe Institute of Technology

キーワード: 遺伝的アルゴリズム (Genetic Algorithms), アアントコロニー最適化手法 (Ant Colony Optimization), シミュレーテッドアニーリング (Simulated Annealing), 巡回セールスマン問題(Traveling Salesman Problem)

連絡先: 〒031-8501 青森県八戸市妙字大開 88-1 八戸工業大学 システム情報工学科 高橋研究室
高橋良英, tel: 0178-25-3111, fax: 0178-25-1691, email: ryoei@hi-tech.ac.jp

1. はじめに

あるセールスマンが n 箇所の都市を 1 回ずつ訪問して出発都市に戻ってくる巡回経路のうち所要距離が最小となる経路を求める問題を巡回セールスマン問題 (TSP: Traveling Salesman Problem) という。TSP の最適解の近似を効率的に得る手法として蟻協調行動モデル (ACO: Ant Colony Optimization) ⁽¹⁾ が有効であることが知られている。ACO は, 都市間の距離の短さと都市間のフェロモン量の多さから次に訪れる都市を確率的に選択する手法である。フェロモン量は, 蟻が都市一周旅行で学習した「巡回路長」の逆数相当の情報で, その巡回路を構成する各都市間の辺上に記憶される。ACO の提案者である M. Dorigo によれば, Oliver の 30 都市問題 (Oliver30) の最適な実数解は ACO では 423.74 であること, 遺伝的アルゴリズム (GAs: Genetic Algorithms) ⁽²⁾ での最適な整数解は 421 である示されているが, ACO と GAs の詳細な性能分析

結果については報告されていない⁽³⁾。このため本論文では, TSP を解く GAs と ACO の機能と性能について, Oliver30 を用いて実験的に評価し, 性能改善策を提案する。

2. 遺伝的アルゴリズム (GAs)

本検討での GAs は, 比較的初期の世代では, 改良 EX (Edge Recombination Crossover) ⁽⁴⁾ で両親の隣接都市リストの中から距離の短い方を次に訪問する都市として確率的に選択して局所的に最適な解を生成し, 後期の世代では SXX (Sub-tour Exchange Crossover) ⁽⁵⁾ を適用して局所的に最適になっている親の部分巡回路と部分巡回路を結合して大域的に最適な解を確率的に生成する。この GAs を特に遺伝子交叉オペレータ交代法 (CXO: Changing Crossover operators) ⁽⁶⁾ と呼ぶ。CXO の概念を図 1 に示す。CXO は任意の時点で任意の遺伝子交叉オペレータ OP1

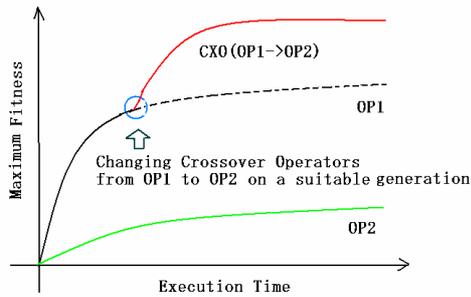


Fig. 1. The concept of Changing Crossover Operators.

図1. 遺伝子交叉オペレータ交代法

から他のオペレータ OP2 に交代可能である。CXO の有効性は 200 都市問題, Eilon の 75 都市問題による C 実験で検証している。

SXX 法⁽⁴⁾

含まれる都市が同じ都市から成る巡廻路の部分集合 S を親 CX と親 CY から探しそれぞれの部分巡廻路を SX と SY とする。選ばれた部分巡廻路から、以下の手続きで、4 人の子供を作る。(a) 親 CX の部分巡廻路 SX を SY または SY の逆巡廻路 (SY) に交換、(b) 親 CY の部分巡廻路 SY を SX または SX の逆巡廻路 (SX) に交換。4 人の子供の中から適応度の高い 2 人の子供を選択して残す。SXX 法では部分巡廻路を選択する効率を向上させるため、どの部分巡廻路を基準の部分巡廻路として選択するかは、その長さと起点に関して一様乱数を発生させて確率的に決定する方法とした。本方法では、最適な部分巡廻路を決定するのに、都市数 n として、網羅的検索では n³ オーダかかっていた検索時間を、都市数 n¹ のオーダに削減している。

改良 EX 法⁽⁵⁾

改良 EX 法は、隣接都市の中で距離の最も短い都市を次々に辿る「最近近傍アルゴリズム」の応用である。各々の都市について、親 CX と親 CY の閉路上で隣接する都市の和集合を考え、それを各都市の隣接リストと呼ぶ。第 1 番目の子の最初の訪問都市は親 CX の最初の訪問都市とし、その隣接リストの中からまだ訪問してい

ない都市の中で最も距離の短い都市を二番目に訪れる都市として選択する。こうして隣接都市リストの中から次に訪れる都市を次々に選択する。隣接リストに訪れる都市がなく、まだ訪問先が残っている場合は未訪問先の中で最も距離の短い都市を次の訪問先として選ぶ。次に訪問する都市がなくなるまでこの処理を続けて、子供 1 の巡回路を決定する。第 2 番目の子の最初の訪問都市を親 CY の最初の訪問都市とすることから始めて、同様な手順で第 2 番目の子の巡回路を決定する。

3. アントコロニー最適化手法 (ACO)

3.1 基本数理モデル

(1) 次に訪れる都市の確率的選択

時刻 t+1 に蟻 k が都市 i に居て、次に訪問する都市として j を選択する確率 $p_{ij}^k(t+1)$ を以下の式で定義する。

$$p_{ij}^k(t+1) = \frac{\tau_{ij}(t)/d_{ij}^\beta}{\sum_{j=1}^{q-1} \tau_{ij}(t)/d_{ij}^\beta}$$

上式にて $\tau_{ij}(t)$ は都市 i から都市 j への t 時におけるフェロモン量、 d_{ij} は都市 i と都市 j 間の距離である。β は定数である。

(2) フェロモンの更新

我々の ACO はエリート・アントシステム Elitist Ant System(EAS)⁽¹⁾とアントコロニーシステム ACS⁽¹⁾の改良版であり、以下のルールでフェロモンを更新している。

① ローカル更新

フェロモン量 $\tau_{ij}(t+1)$ は、m 匹の蟻が旅を完

了する度に、以下の式で更新する。時間 t は m 匹の蟻が旅を完了する度に更新される。

$$\tau_{ij}(t+1) = (1 - \rho) * \tau_{ij}(t) + \rho * \Delta\tau_{ij}(t+1)$$

上式にて、 ρ はフェロモンの蒸発度である。

$\Delta\tau_{ij}(t+1)$ は時間 $t+1$ で旅をした m 匹の蟻が都市 i と都市 j 間の辺上に残すフェロモンの総和である。

②グローバル更新

グローバル更新では、時間 $t+1$ までに旅を終えた $m(t+1)$ 匹の蟻うち最短距離を実現した蟻のフェロモン量を、巡回路の各辺に加える。

$$\tau_{ij}(t+1) = (1 - \rho) * \tau_{ij}(t) + \rho * \alpha * \max \tau_{ij}(t+1)$$

上記にて α はグローバルフェロモン更新量調整パラメータである。 $\max \tau_{ij}(t+1)$ は時間 $t+1$ まで旅をした蟻のうち最も短い経路を探索した蟻（エリート蟻）が都市 i と都市 j 間の辺上に残すフェロモン量である。

3.2 本検討での ACO の工夫点

局所最適解に探索が陥らないように、本検討の ACO は以下の広義と狭義のシミュレーテッドアニーリング機能（SA: Simulated Annealing）を 3.1（2）に述べた基本機能に付加している。

（i）フェロモンによるシミュレーテッドアニーリング機能(SA)を有する ACO

本検討では 0 から $0.5 * m$ の値をとるように時間 t により周期的にグローバルフェロモン更新量調整パラメータ α を変動させる。この時、 α は経路選択を確率的動作から決定的動作に変更させるように働く。この機能を有す

る ACO をフェロモンによるシミュレーテッドアニーリング機能(SA)を有する ACO と呼び、広義の SA である。 α が固定的に 1 の場合は SA 手法を有しない ACO 法と呼び、EAS と同じである。

（ii）距離によるシミュレーテッドアニーリング機能(SA)を有する ACO

蟻が旅を終える度に 2-opt 法と 3-opt 法を適用して旅行ルート of 確率的更新を行う。この時、距離の短縮の程度を測定し、それが大きければルート更新を行う。世代数の逆数を温度パラメータとしたボルツマンマシンの受理関数により確率的な旅行ルートの更新を行う。これを距離によるシミュレーテッドアニーリング機能と呼び、狭義の SA である。

4. 実験結果

4.1 実験データ

I. M. Oliver, D. J. Smith, and J. R. C. Holland が ICGA' 87 で発表した論文⁽⁸⁾の Appendix を参照のこと。二次元 Euclid 空間データであり、以下がその (x, y) 座標である。

{ {82, 7}, {91, 38}, {83, 46}, {71, 44}, {64, 60},
{68, 58}, {83, 69}, {87, 76}, {74, 78}, {71, 71},
{58, 69}, {54, 62}, {54, 67}, {37, 84}, {41, 94},
{2, 99}, {7, 64}, {22, 60}, {25, 62}, {18, 54},
{4, 50}, {13, 40}, {18, 40}, {24, 42}, {25, 38},
{41, 26}, {45, 21}, {44, 35}, {58, 35}, {62, 32} };

4.2 最適解

CXO は巡回路長 = 423.74 を持つ最適（最小）巡回路を乱数種 1 のもとで発見した。ACO は同一の最小巡回路を同じ乱数種 1 のもとで探索した。探索した最適巡回路を図 2 に示す。図中の番号は、都市の巡回順を示す。図中の各都市の (x, y) 座標は 4.1 の座標に一致している。

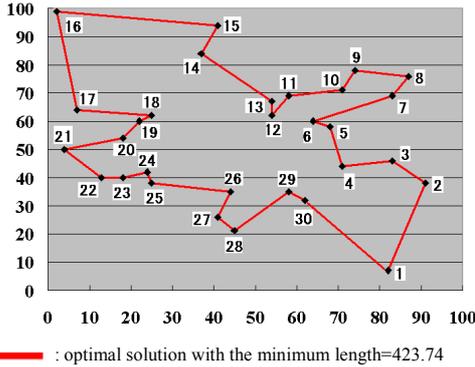


Fig. 2. The optimum solution found by GAs as well as ACO for Oliver's 30 city problem

図2. Oliver の 30 都市問題の最適解

4.3 各手法の評価

(1) 遺伝的アルゴリズム

(A) 起動パラメータ

- ・ 世代交代方式 ・ ・ 一括世代交代方式。
子供が人口数に達したら、世代交代する。
- ・ 親の選択法 ・ ・ ルーレット方式
- ・ 間引き ・ ・ 有り。
生成した 2 人の子供のうち適応度の高い子供のみ残す。
- ・ 乱数種 = 1
- ・ 人口数 = 300
- ・ 遺伝子交叉確率 = 0.8
- ・ 2-opt 法 ・ ・ 有り。
生成した子供に 2-opt 法を適用して距離の改善を図る。これにて突然変異を暗に実現している。

(B) 実験結果

改良 EX から SXX に 7 世代目に交代する CXO では 23 世代目、441 ミリ秒後に経路長 = 423.74 の最適解を探索した。表 1 に示すように、この CXO は改良 EX 単独や SXX 単独よりも、より短い経路をより短い時間で探索する。CXO が巡回経路長 = 423.74 の最適解を探索した 23 世代目に改良 EX は 432.62 の局所最適解しか探索

表 1 CXO、improved EX、SXX の最適解への収束過程

Table1 Process to converge to the optimum solution regards as CXO, improved EX and SXX

generation	CXO	improvedEX	SXX
0	1037.85	1037.85	1037.85
5	449.61	449.61	934.16
6	437.95	437.95	893.59
7*	437.95	437.95	886.83
8	437.74	437.95	858.89
9	434.77	437.03	850.76
10	430.71	437.03	839.28
11	430.71	432.62	819.60
20	424.57	432.62	699.19
23	423.74	432.62	670.75
40	423.74	432.62	547.35
50	423.74	432.62	510.40
60	423.74	432.62	474.90
74	423.74	432.62	466.56
100	423.74	432.62	466.56

on 7th generation CXO changes EX to SXX

表 2 改良 EX から SXX への最適な交代世代の探索

Table 2. Searching the best generation to exchange improved EX to SXX

generation for exchanging EX for SXX	minimum length found	time to find the minimum length (millisecond)	generation to find the minimum length
1	429.38	590	48
2	445.97	731	57
3	425.27	441	30
4	423.74	481	31
5	424.69	311	16
6	424.69	321	15
7	423.74	441	23
8	424.69	371	15
9	424.69	410	17
10	424.69	420	16
11	424.69	511	23

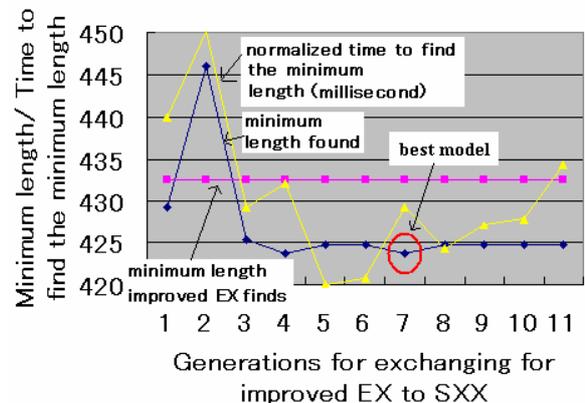


Fig.3. Searching the optimum generation to exchange improved EX to SXX

できなかった。SXX も 670.75 の局所解しか探索できなかった。オペレータ交代後改良 EX 単独では 11 世代目に探索した局所最適解が安定状態になりそれ以降経路長は改良されなかった。SXX は巡回路長=466.56 の局所最適解を得るのに 74 世代要した。1 世代は 300 回の旅からなるので、CXO では最適解を探索するのに、9,300 回の旅を要している。CXO、改良 EX、SXX のそれぞれについて各世代までに探索できた経路長を表 1 にまとめる。

CXO においてオペレータの交代時期によって局所最適解の経路長は変化する。その詳細を表 2 に示す。CXO は 1 世代目から改良 EX が最適解を探索した 11 世代目の間に最適な遺伝子交叉オペレータの交代時期を探索する。表 2 からわかるように、CXO で最適解を探索した交代世代番号は 4 と 7 でありそれ以外の交代世代では最適解は見つからない。また、オペレータ交代時期が 4 世代目以降の場合 CXO が探索した局所最適解の経路長は改良 EX より短い。尚、CXO が各世代交代方式で 100 世代分の旅 (30,000 回=100 世代×300 回の旅) を観察したとすると、最適交代世代を決定するのに 1300msec×6 回=7800 ミリ秒かかる。この時間を考慮すると CXO の最適解探索時間の総計は 8,241 ミリ秒となる。尚、3-opt 法を、2-opt 法に加えると更に最適解への収束効率が良くなることを確認しているが詳細は略す。

(2) アントコロニー最適化手法

(A) 起動パラメータ

- ・忘却率 $\rho = 0.2$
- ・距離の重み $\beta = 2$
- ・ローカルフェロモン更新タイミング $m = 10$:
m 匹の旅毎にフェロモンを更新
- ・人口数・初期化する蟻の数 300
- ・乱数種・1
- ・「フェロモンによる SA 機能」・「有り」
グローバルフェロモン更新定数 $\alpha = 0.5$ 。

表 3 シミュレーテッドアニーリング機能による

ACO の性能向上

Table 3 Performance improvement of ACO endowed with

Simulated Annealing

methods		best-length	required number of generations to find the best solution	computer execution time to find the best solution (msec)
ACO with SA	distance_all + pheromone cyclically	423.74	102	630
	pheromone cyclically updated	423.74	449	2,723
	distance_all	423.74	512	3,924
	distance_with_2-opt	423.91	29	200
	distance_with_3-opt	423.91	37	230
ACO without SA		423.91	411	2,553

0 から $(m \cdot \alpha - 1) = (10 \cdot 0.5 - 1) = 4$ 倍の間でエリート蟻が探索した経路上にフェロモンを単調増加に残す。これが 4 倍以上になればフェロモンを一端少なくして再び同様にフェロモンを落とす。

- ・「距離による SA 機能」・「有り」

蟻が旅を終える毎に、2 opt 法と 3opt 法で旅行ルートを変更し、「距離の短縮度」をエネルギー減少量、「フェロモン更新回数」を時間としたボルツマン受理関数で確率的にその変更を有効とするか否かを決める。

(B) 実験結果

SA 機能を有する ACO は経路長 423.74 の最適解を 3,060 (=30 匹×102 世代) 回の旅行、630 ミリ秒で探索した。SA 機能を有さない ACO では最適解を探索できず、経路長 423.91 の局所最適解を 13,230 (=30 匹×411 世代) 回の旅行、2,553 ミリ秒で見出すのみである。このように ACO (SA 機能あり) は ACO (SA 機能なし) に SA 機能を加えることで探索経路長 (機能) や最適解の探索時間やツアー数 (性能) を大幅に改善できた。詳細を表 3 に示す。表 3 は「距離による SA」、「フェロモンによる SA」も同じ最適解を探索しているが、「フェロモンによる SA」の方が探索時間について優れていること、両方の SA 機能の相乗効果で機能と性能が更に向上していることを示している。

表4 GAs, ACO と CXO, ECXO の機能と性能比較

Table 4. Performance evaluation of GAs, ACO, and their combination CXO and ECXO

NO	methods	best length found	required number of tours to find the best length	required computer execution time to find the best length (mseconds)	population size
1	ECXO (ACO with SA→SXX)	423.74	1,620	260	900
2	ACO with SA	423.74	3,060	630	30
3	CXO (improved EX→SXX)	423.74	6,900	441	300
4	ACO without SA	423.91	13,230	2,553	30
5	improved EX	432.62	3,300	380	300
6	SXX	466.56	22,200	851	300

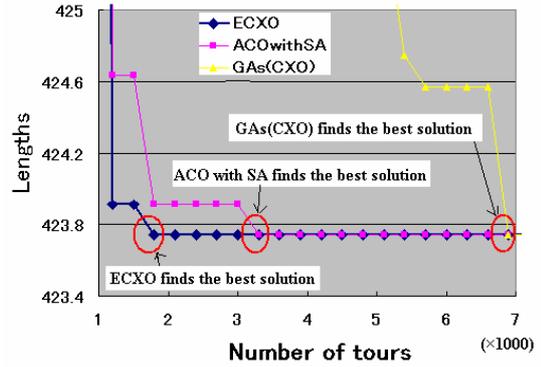


Fig. 4. Process to obtain the optimum solution for both of GAs and ACO from viewpoints of number of tours.

図 4. GAs と ACO の最適解への収束効率の比較

(3) ACO と GAs の最適解への収束効率の比較

最適交代世代探索のもと、CXO は 6,900 回の旅行、441 ミリ秒で最適解を探索している。一方、ACO (SA 機能あり) は 3,060 回の旅行、630 ミリ秒で最適解を探索している。このデータは、最適解探索のための所要コンピュータ実行時間の観点からは CXO は ACO (SA 機能あり) より優れていること、所要旅行回数の観点からは ACO (SA 機能あり) が CXO より優れていることを示している。

(4) 遺伝的アルゴリズムとアントコロニー最適化手法の結合による性能改善

(A) 考え方

ACO と GAs を結合することで更に経路長の短い経路を探索する拡張遺伝子オペレータ交

代法 ECXO (Extended Changing Crossover Operators)⁽⁷⁾を検討した。ECXO は CXO の OPI として ACO を可能とした方式である。本方式は初期の世代で ACO により局所的に最経路長の短い経路を探索し、安定状態に突入した後期の任意の世代に SXX に交代して更に広域的に最適な解を生成する。

(B) 起動パラメータ

●交代前の ACO の起動パラメータ：

「距離による SA」、「フェロモンによる SA」の両方の機能を有する ACO である。詳細パラメータは (2) (A) と同じである。

●交代後の SXX の起動パラメータ：

- ・人口数：900
- ・SXX が ACO から引き継ぐ蟻の旅行数：30
- ・世代交代方式・・・一括世代交代方式。

表 5. ACO から SXX に交代する最適世代の探索

Table 5. Finding out the optimum generation for exchanging crossover operators from ACO to SXX (GAs).

generations for exchanging ACO for SXX	best length ACO finds	best length improved by ECXO	number of generations to find the best length found by SXX	total number of generations to find the best length by ECXO	computer execution time to find the best solution (mseconds)		
					execution time for ACO (msec)	execution time for SXX (msec)	sum (msec)
1	605.09	423.74	463	464	10	761	771
2	465.48	423.74	412	414	10	701	711
6	463.80	425.27	169	175	40	280	320
10	460.41	423.74	311	321	60	530	590
21	459.64	423.74	206	227	140	340	480
22	459.48	424.69	60	82	140	100	240
23	441.82	436.15	254	277	150	450	600
31	425.82	423.74	23	54	230	30	260
36	424.64	423.74	23	59	260	30	290
60	423.91	423.74	34	94	430	60	490
102	423.74	-	0	102	630	-	630

- ・親の選択法・・・ルーレット方式
- ・間引き・・・有り。
- ・遺伝子交叉確率=0.8
- ・2-opt 法・・・有り。
- ・3-opt 法・・・「有り」
- ・突然変異確率：0.5

生成した2人の子供のうち一方の子供に対して、強制的に2-opt法で突然変異を起こす。明示的な「突然変異」を実現する。

(C) 起動結果

930 (=30匹×31世代)回, 230ミリ秒のACOの旅の後にSXXに交代させることでSXXは690 (=30匹×23世代)回, 30ミリ秒の旅で最適解を見つけた。すなわち総計1,620回の旅行回数260ミリ秒のコンピュータ実行時間でECXOは最適解を見つけている。この結果はACO(SA機能あり)が最適解を見つける為に要した旅行回数3,060回とコンピュータ実行時間630ミリ秒を大幅に削減したものとなっている。表4に性能評価結果の詳細を示す。表4は探索した最適経路長と最適解を探索に要した旅行数の観点から評価すると、ECXO, ACO(SA機能あり), CXO, ACO(SA機能なし), 改良EX, SXXの順番となること、コンピュータ実行時間の観点からはCXOが最も優れていることを示す。表4に各手法の人口数も示す。図4は旅行回数に対する各手法の最適探索経路長の変化過程を示す。

(D) 最適のACOからSXXへの交代時期

ACOからSXXへの最適な交代時期は1世代目から次善解を探索した60世代目の間に見つける。探索した最適解の経路長が平衡状態となった10個の世代(1, 2, 6, 10, 21, 22, 23, 31, 36, 60)を交代世代候補とした。そのうち1, 2, 10, 21, 31, 36, 60の各世代で交代するECXOが最適解を探索した。この5つのうち最適解探索時間の最も少ない最適交代時期は31と比較的若い世代であった。その詳細を表5に示す。表5では

ACOで探索した局所最短距離、SXXに交代後ECXOが探索した最短距離、最短距離を探索するのに要したACOとSXXの旅行回数とそのトータルであるECXOの旅行回数、最短距離を探索するのに要したACOとSXXのコンピュータ実行時間(単位:msec)とそのトータルであるECXOのコンピュータ実行時間を示している。

5. 結論

Oliverの30都市問題Cプログラム実験の結果、(i) GAs(CXO)とACOはいずれも同一最適解(経路長=423.74)を探索可能なこと、(ii) ACOにSA機能を加えることで探索経路長や探索時間を大幅に改善できること、(iii) 最適解を探索するのに必要な最適旅行回数の観点からはACOがGAsより優れていること、最適コンピュータ実行時間の観点からはGAsがACOより優れていること、(iv) ACOとSXXを結合するとACOの探索旅行回数や探索時間を更に短縮できることがわかった。

参考文献

- (1) M. Dorigo and T. Stutzle, *Ant Colony Optimization*, The MIT Press, 2004.
- (2) D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc., 1989.
- (3) M. Dorigo and L. M. Gambardella, "Ant Colonies for the Traveling Salesman Problem," *Biosystems*, 43(2), pp. 73 - 81, 1997.
- (4) D. Whitley, T. Starkweather and D'Ann Fuquary, "Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operation," *Proc. of 3rd Int. Conf. on Genetic Algorithms*, 1989, pp. 133-140.
- (5) M. Yamamura, I. Ono and S. Kobayashi,

“Emergent Search on Double Circle TSPs using Subtour Exchange Crossover,” Proc. of 1996 IEEE Int. Conf. on Evolutionary Computation, 1996, pp.535-540.

(6) Ryouei Takahashi, ”Solving the Traveling Salesman Problem through Genetic Algorithms with Changing Crossover Operators,” in Proceedings of Fourth International Conference on Machine Learning and Applications, pp. 319- 324, published by IEEE Computer Society, 2005.

(7) R. Takahashi, “A Performance Improvement of Solving the Traveling Salesman Problem through Uniting Changing Crossover Operators to Ant Colony Optimization,” Advance in Natural Computation and Data Mining, Proceedings the 2nd International Conference on Natural Computation and the 3rd International Conference on Fuzzy Systems and Knowledge Discovery, pp. 114 - 130, Xidian University, 2006.

(8) I. M. Oliver, D. J. Smith, and J. R. C. Holland, “A Study of Permutation Crossover Operations on the Traveling Salesman Problem,” Proc. of 2nd Int. Conf. on Genetic Algorithms, 1987, pp. 224- 230.