

遺伝子交叉オペレータ交代法による 巡回セールスマン問題の解法

高 橋 良 英*・出 貝 賢 一**

Solving the Traveling Salesman Problem through Changing Crossover Operators

Ryouei TAKAHASHI* and Kenichi DEGAI**

Abstract

In order to solve the Traveling Salesman Problem (TSP) through Genetic Algorithms (GAs), a method of changing crossover operators (CXO), which can flexibly substitute the current crossover operator for another suitable crossover operator at any time, is proposed. We examined seven crossover operators. Our experiments using data of the famous Eilon's 75-city problem show that CXO which uses an improved EX in early generation and uses SXX after several generations can efficiently provide with a best approximate solution of TSP, where improved EX, whose idea comes from greedy algorithm, selects the nearest town from the current visiting city selected out of adjacent cities in parents to create local optimum sub-paths and our SXX efficiently generates cyclic paths in the next generation by performing a crossover operation on a pair of the selected parents that have the local optimum sub-paths. In our case study, a CXO can find an approximate solution of the optimum cyclic paths on the 77-th generation after exchanging improved EX for SXX on the 40-th generation, and it takes about 21.9 seconds to select the best solution with the length (≈ 544.81). Our experiments have shown that CXO can improve path's length as well as computer execution time which the improved EX and SXX have. This experimental result suggests that changing crossover operators at arbitrary time according to city data is available to improve both the functionality and the performance of GAs.

Keywords: GAs, TSP, Changing crossover operators, Improved EX, SXX, C

1. はじめに

巡回セールスマン問題(TSP) [2], [9], [13]を遺伝的アルゴリズム [1] で解く時の致死遺伝子対策 (遺伝子交叉オペレータ) としてこれまで, (1) 訪れる順番に並べた n 個の都市を表現する遺伝子列 (染色体) を n 個の文字の順列と考え, 順列に対する互換や表現方法に対して

工夫をこらすことで致死遺伝子が生じないような遺伝子交叉 (置換) を実現した Grefenstette 法 [3], PMX 法 [2], CX 法 [4], OX 法 [5], (2) 隣接都市間の所要距離等枝のつながりに着目した距離 EX 法 [6], SXX 法 [7], EXX 法 [8] が発見されている。これまでの我々の実験結果によれば, 比較的初期の世代では, 所要距離の最も短い都市を親の隣接都市リストの中から次々に選択していく改良 EX 法が適応度の高い個体を生成する傾向があること, 世代が経るにつれ, ある都市からある都市まで部分的に最適な経路となっている親の部分枝と部分枝を交

平成 17 年 12 月 16 日

* システム情報工学科・教授

** 大学院工学研究科電気電子工学専攻博士前期課程・2 年

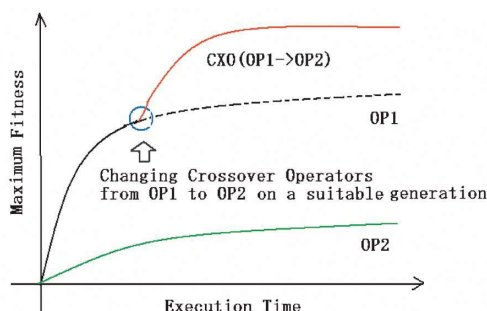
又させて次世代の個体を生成する SXX 法が適応度の高い個体を生成する傾向のあることがわかっていて、このため、より所要距離の短い経路をより効率的に選択可能とするため、遺伝子交叉法と交替時期を柔軟に選択可能な、遺伝子交叉オペレータ交代法 [14] について研究することとした。

2. 遺伝子交叉オペレータ交代法

遺伝子交叉オペレータ交代法 (CXO: Changing Crossover Operators) は、任意の遺伝子交叉オペレータ OP1 で生成される個体群 (集団) の最大適応度 (適応度 = 2 都市間の距離の総和 / 経路長) を監視しそれが平衡状態 (値が一定の状態) になったタイミング等で、他の遺伝子交叉オペレータ OP2 に遺伝子交叉主体を交代させる。これにより、適応度のより高いモデルをより短時間で探索可能とする。遺伝子交叉オペレータ交代法の概念を図 1 に示す。

実験では、改良 EX 法から SXX 法に交代させる遺伝子交叉オペレータ交代法について評価することとした。遺伝子交叉オペレータは、その最適な交代世代を、1 世代目から改良 EX 法が探索した最適世代数の間に探索する。

SXX 法, 改良 EX 法の機能とその特徴を以下に示す。



(○: OP1 から OP への最適交代時期)

図 1. 遺伝子交叉オペレータ法の概念図

Figure 1. The concept of Changing Crossover Operators (CXO).

SXX 法

含まれる都市が同じ都市から成る巡回路の部分集合 S を親 CX と親 CY から探しそれぞれの部分巡回路を SX と SY とする。選ばれた部分巡回路から、以下の手続きで、4 人の子供を作る。(a) 親 CX の部分巡回路 SX を SY または SY の逆巡回路 (\overline{SY}) に交換, (b) 親 CY の部分巡回路 SY を SX または SX の逆巡回路 (\overline{SX}) に交換。SXX 法では部分巡回路を選択する効率を向上させるため、どの部分巡回路を基準の部分巡回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、最適な部分巡回路を決定するのに、都市数 n として、網羅的検索では n^3 オーダかかっていた検索時間を、都市数 n^1 のオーダに削減している。本実験では、他の遺伝子交叉法との整合をとるため、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している。

改良 EX 法

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「最近傍アルゴリズム」[10] と「けちの原理」[2] 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. 実験結果

3.1 実験データ

図2の二次元ユークリッド空間に位置する有名な Eilon の 75 都市問題について最短巡回路探索実験を行った。都市間の距離はピタゴラスの定理で求めている。図中の朱線は探索した最小経路長 ≈ 544.81 の最適巡回路である。巡回路上の各都市の座標は以下の通りである。その座標は図中の 1 番目の都市から始めて 2, 3..., 75 と最短巡回路の順番に並んでいる。

〈Eilon の 75 都市問題の都市座標〉

{6,25}, {11,28}, {12,38}, {7,43}, {9,56}, {15,56}, {17,64}, {10,70}, {31,76}, {40,66}, {47,66}, {50,70}, {57,72}, {55,65}, {70,64}, {62,57}, {55,57}, {55,50}, {50,50}, {41,46}, {45,42}, {45,35}, {40,37}, {38,33}, {33,34}, {29,39}, {33,44}, {30,50}, {35,51}, {40,60}, {35,60}, {30,60}, {26,59}, {22,53}, {21,48}, {21,45}, {21,36}, {20,30}, {26,29}, {27,24}, {30,20}, {35,16}, {40,20}, {36,26}, {43,26}, {48,21}, {52,26}, {50,30}, {55,34}, {54,38}, {50,40}, {51,42}, {55,45}, {62,48}, {67,41}, {62,

35}, {65,27}, {62,24}, {55,20}, {60,15}, {66,14}, {66,8}, {64,4}, {59,5}, {50,4}, {54,10}, {50,15}, {44,13}, {36,6}, {26,13}, {22,22}, {16,19}, {15,14}, {15,5}, {12,17}}

注意：最適解の順番に都市は並んでいる。

3.2 遺伝子の遺伝子型と表現型

遺伝子列（染色体）は訪れる順番に並べた n 個の都市として表現する [2]。遺伝子型の一遺伝子は unsigned int で実現している。コンパイラの処理系に依存するが、我々の実験では一遺伝子は 32 ビット（4 バイト）で表現されている。

3.3 GA の起動パラメータ

遺伝子交叉オペレータを横並びに比較評価するため、GA 起動パラメータは、以下の通り固定した。

- 世代交替方式…一括型世代交代方式
- 2 オプト法 [9] …有り。2 オプト法は 75 個の遺伝子列の中から確率的に二都市 A と B を選びその間の都市の訪問順番を交換する。
- 子供の生成方法…「間引き」あり
- 親の選択…ルーレット方式
- 遺伝子交叉確率…0.8
- 乱数種（初期集団の規定パラメータ）…1

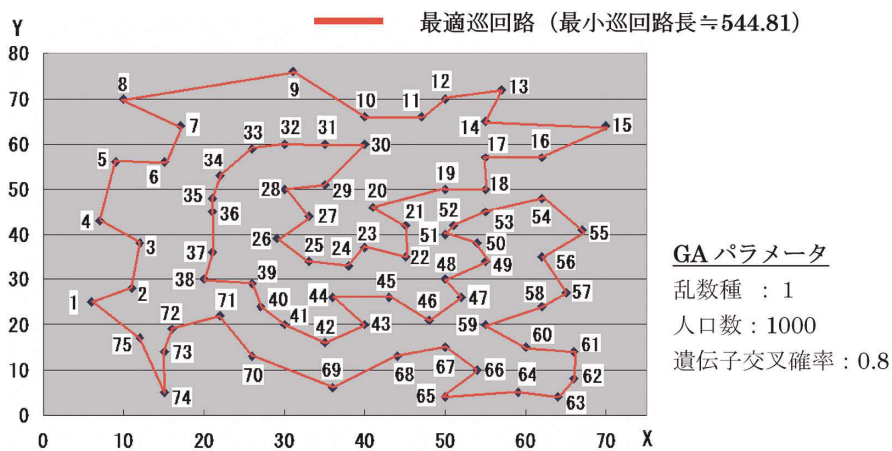


図2. 遺伝子交叉オペレータ法（改良 EX 法→SXX 法）が探した 75 都市問題の最短経路
 Figure 2. An optimum solution of the Eilon's 75-city traveling salesman problem.

表 1. 各遺伝子交叉オペレータの性能比較結果

Table 1. Performance evaluation of crossover operators

遺伝子交叉オペレータ	最大適応率	最小巡回路長	最適解（最小巡回路長を持った巡回路）を選択するまでにかけた世代数	最適解を選択するまでにかけたコンピュータ実行時間（秒）
CXO (improved EX → SXX) *	341.0	544.8	77	21.9
Improved EX	332.6	558.6	48	18.5
EXX	329.3	564.1	144	19.7
CX	324.9	571.8	399	8.4
Grefenstette	318.5	583.4	692	285.5
SXX	313.8	592.0	203	28.8
OX	152.1	1221.4	714	21.0
PMX	135.2	1373.8	1000	28.9

〈注意〉 (improved EX → SXX) * : 40 世代目に 遺伝子交叉法を改良 EX 法 (improved EX) から SXX に換える。乱数種=1, 遺伝子交叉確率=0.8, 人口数=1,000, 最大観測世代数=1,000。

●人口数…1,000

●観測最大世代数…1,000

2 オプト法と突然変異

「突然変異」はある都市 A を別の都市 B に強制的に変化させる。そのままでは、致死遺伝子が生じるので、A と B を交換することで、致死遺伝子を生じさせないようにできる。2 オプト法では、親の訪問都市順番の履歴を残すために、A と B を単に交換するのみでなく、その間の訪問順番を交換する。この意味で、TSP における突然変異の一実現法が 2 オプト法であるといえる。

3.4 遺伝子交叉オペレータ交代法の評価

改良 EX から SXX に交代させるオペレータ交代法 (CXO 法) について評価することとした。評価結果を以下に整理する。

3.4.1 最適解

40 世代目で改良 EX から SXX に交代させるオペレータ交代法を評価した。乱数種は 1 で探索した。そして、世代数 77 の時、実行時間約 21.9 秒で経路長最小 (≒544.81) の巡回路 (最適解) を探索した。CXO 法が発見した最適巡回路を図 2 に示す。

3.4.2 既存の遺伝子交叉オペレータとの機能と性能の比較結果

CXO 法が交代対象とする遺伝子交叉オペレータは Grefenstette 法, PMX 法, OX 法, CX 法, 改良 EX 法, SXX 法, EXX 法である。各遺伝子交叉オペレータの性能評価結果を表 1 に示す。表 1 では、遺伝子交叉オペレータ毎に、(1) 最大適応度、(2) 最小巡回路長、(3) 最適解 (最小巡回路長を持った巡回路) を選択するまでにかけた世代数、(4) 最適解を選択する

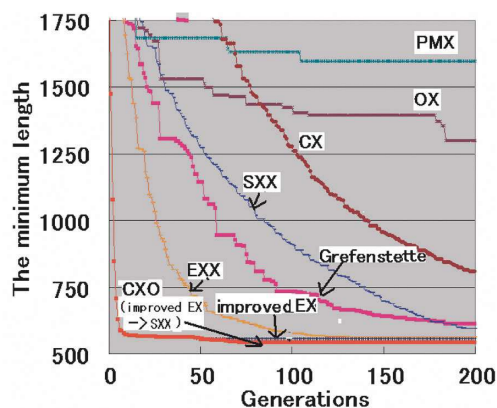


図 3. 最小巡回路長の世代による推移状況

Figure 3. The off-line performance evaluation of crossover operators.

までにかかったコンピュータ実行時間を示している。初期世代の集団を決定するために使用した乱数種は1である。

致死遺伝子対策毎にその世代までに探せた最小巡回路長 (オフラインパフォーマンス [2]) の世代に対する推移状況を図3に示す。

表1と図3から以下のことが読み取れる。

- ① 改良EX法単独では、経路長最小(=558.6)の巡回路を48世代目に探索しているが、実行時間18.5秒かかっており、機能(経路長の長さ)の観点からCXO法より劣る。
- ② SXX法単独では経路長=592.0の次善解を探索するのに203世代、実行時間28.8秒かかっており、機能と探索効率の両方の観点からCXO法より劣る。EXXが探した最小巡回路長は564.1であり、機能の観点からCXO法より劣る。
- ③ PMX法やOX法は枝のつながり(距離の長さ)に関する情報が遺伝子交叉で欠落するため、最適巡回路に収束しなかつ

た。探索した最小経路長は1,000以上であった。

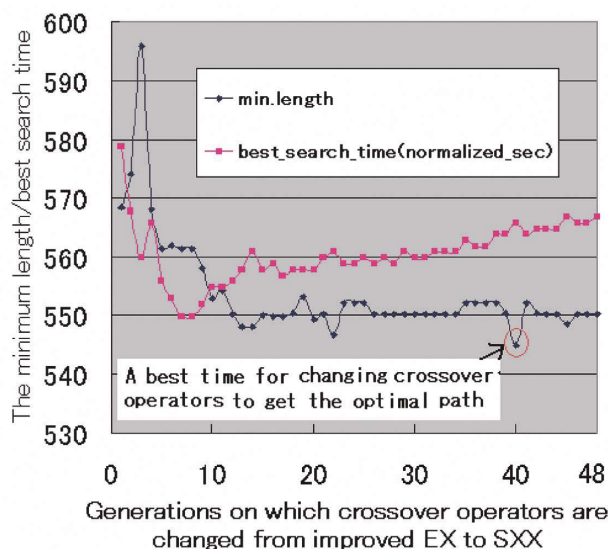
- ④ Grefenstette法, CX法が探索した最小巡回路経路長はそれぞれ571~584であり, PMX法OX法より短いものの改良EX法やEXX法より長く, 中間解である。最適巡回路には収束しておらず, 機能の観点からCXO法より劣る。

3.4.3 最適なオペレータ交代時期の探索

最適なオペレータ交代時期は, 固定的ではなく都市がどこに位置するかデータの位置関係に依存する。実験では, 1世代目からEX法で探索した最適世代番号(=48)までの間で最適なオペレータ交代時期を探索することとした。1世代目から48世代目を遺伝子交叉オペレータの交代世代番号とした場合に探索できた最小経路長と最小経路を探索するまでにかかったコンピュータ実行時間を図4に示す。

特に交代世代番号5世代ごとのそれ等の値と最小経路を探索した時の世代番号を表2に示す。

表2と図4から以下が読み取れる。



(説明)

① min. length :
最小巡回路長

② best search time (normalized sec)
最小巡回路を探索するまでの(正規化)コンピュータ実行時間

③ (○印): 40世代目が改良EXからSXXへの最適交代世代であることを示している。

図4. 遺伝子交叉オペレータ交代法(改良EX法→SXX法)における交代世代番号と探索した最小巡回路長ならびに最小巡回路を選択するまでに要した時間の関係(Eilonの75都市問題の場合)

Figure 4. Selection of the optimum generation on which crossover operators are changed to get the best solution. (in a case study of the Eilon's 75 city problem).

表 2. 遺伝子交叉オペレータ交代法 (改良 EX 法→ SXX 法) の交代世代番号毎の最小巡回路長と最小巡回路長を探索するまでに要した実行時間ならびに世代数の関係

Table 2. Variation of the execution time ET for searching local optimum paths with the generation when crossover operators are changed from improved EX to SXX

改良 EX 法から SXX 法への 交代世代番号	最小巡回路長	最小巡回路長を探索するのに 要したコンピュータ実行時間 (秒)	最小巡回路長を探索する までに要した世代数
5	561.4	11.0	59
10	552.9	10.3	46
15	550.1	13.9	60
20	549.2	13.6	53
25	552.1	15.3	55
30	550.2	15.9	53
35	552.1	18.4	62
40*	544.8	21.9	77
45	548.6	22.0	72

〈注意〉 40 世代目に改良 EX 法 to SXX に交代して、最適解 (経路長≒544.8) を探索できた。図 4 の正規化時間は表 2 のコンピュータ時間に 544.8 を足した数値である。

- ① 40 世代目で改良 EX 法から SXX 法に交代した後、77 世代目、21.9 秒後に、最小巡回路長≒544.81 の最適解に到達できること。
- ② 48 個の交代時期候補のうち 40 世代目のみで最短経路を見つけられること
- ③ 5 世代目に交代する以外の 47 個の交代時期の全てについて探索した最小巡回路長は改良 EX が探索した 558.6 より短かった。この結果は改良 EX 法から SXX 法に任意の時期に交代するのみで次善解を得られる傾向が高いことを示している。
- ④ 交代世代番号が増えるにつれ最適解を探索する時間は劣化する (遅くなる) 傾向があること。

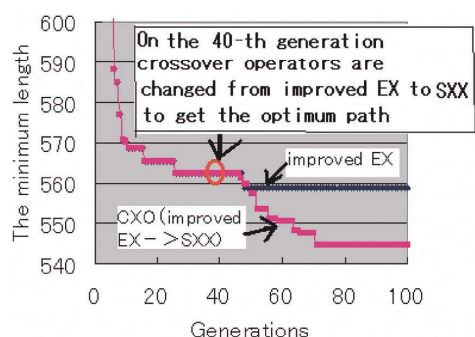
尚、性能と機能はトレードオフ関係にあり、最適交代世代候補となる全ての世代について最小巡回路長を調べその中から最終的な最小巡回路長を選択するのか、最適交代世代番号候補の中から確率的に何件かに交代世代番号をしぼって最小巡回路長を調べるのかは、調査の目的に

よって使い分ける必要がある。前者は機能を性能より重視する場合であり、後者はその逆である。

3.4.4 オペレータ交代法の有効性の検証

「40 世代目で改良 EX 法から SXX 法に交代する CXO 法」の有効性を図 5 に示す。図 5 の○印のタイミング(40 世代目、開始から 21.9 秒後)で CXO 法では改良 EX 法から SXX 法に交代している。表 3 は 10 世代毎に、CXO 法 (改良 EX → SXX) と、改良 EX 法と SXX 法が探索した最小巡回路長の具体的数値を示している。実際、CXO 法 (改良 EX → SXX) が最適モデル (巡回路長≒544.81) を探せた 77 世代目に、改良 EX 法と SXX 法は、最小巡回路長が 558.63 と 1050.36 であり、まだ最適モデル (最も最小巡回路長の小さなモデル) を探せていないことがわかる。尚、改良 EX 法単独では 77 世代目以降も最小巡回路長は 558.63 の値で安定化してしまい、経路長の更に短い巡回路を探索できなかった。

図 6 は最適解を探索するまでの時間が「40 世代目 (15.61 秒) 後で改良 EX 法から SXX 法に



(○の説明) 40 世代目が改良 EX から SXX に遺伝子交叉オペレータを交代させる最適交代世代であることを示している。

図 5. オペレータ交代法 (改良 EX → SXX) の交代時期と探索した最小巡回路長

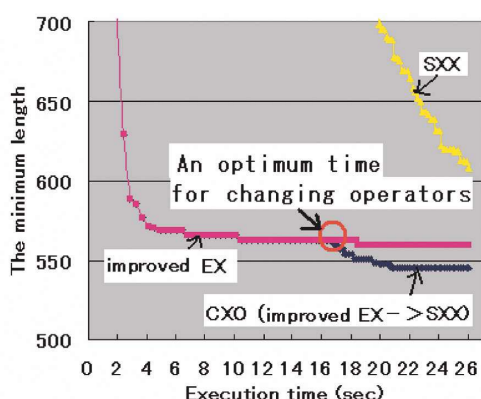
Figure 5. Minimum-length improvement through CXO (improved EX → SXX).

表 3. オペレータ交代法 (CXO) と改良 EX, SXX 間の最小巡回路長の世代による推移状況の比較

Table 3. Minimum lengths evaluation of CXO, improved EX and SXX

世代	最小巡回路長		
	CXO (improved EX → SXX)	improved EX	SXX
0	2,108.24	2,108.24	2,108.24
10	570.18	570.18	1,895.49
20	565.52	565.52	1,685.24
30	562.46	562.46	1,513.73
40*	562.46	562.46	1,389.09
50	557.15	558.63	1,272.57
60	550.42	558.63	1,191.69
70	547.45	558.63	1,099.49
75	544.84	558.63	1,074.98
77**	544.81	558.63	1,050.36
80	544.81	558.63	1017.92
90	544.81	558.63	966.57
100	544.81	558.63	909.26

(注意) 40*: 40 世代目に改良 EX から SXX に交代
77**: 77 世代目に CXO が最適巡回路を探索



(○印の説明) 改良 EX から SXX に交代させる最適なタイミングを示している。

図 6. CXO (改良 EX 法 → SXX) による実行時間改良

Figure 6. Execution-time improvement through CXO (improved EX → SXX).

表 4. オペレータ交代法 (CXO) と改良 EX, SXX 間の最小巡回路長探索時間の推移状況の比較

Table 4. Execution time evaluation of CXO, improved EX and SXX

コンピュータ 実行時間 (秒)	最小巡回路長		
	CXO (improved EX → SXX)	improved EX	SXX
0.14	2,108.24	2,108.24	2,108.24
4.55	570.18	570.18	1,410.95
8.31	565.52	565.52	1,132.75
11.97	562.46	562.46	930.91
15.61*	562.46	562.46	812.03
17.23	557.15	562.46	774.34
18.84	550.42	558.63	730.5
20.61	547.45	558.63	688.71
21.5	544.84	558.63	668.82
21.86**	544.81	558.63	668.82
22.38	544.81	558.63	655.87
24.2	544.81	558.63	622.16
26.08	544.81	558.63	607.94

(注意) 15.61*: 40 世代目に改良 EX から SXX に交代
21.86**: 77 世代目に CXO が最適巡回路を探索

交代する CXO 法によってどのように改良したかを示している。この CXO 法と改良 EX 法、SXX 法について、コンピュータ実行時間とその時間までに発見できた最小巡回回路長を示している。表 4 はその詳細を示している。CXO 法が最小経路長 544.81 を発見した 21.86 秒後に、改良 EX 法は 558.63、SXX 法は 668.82 の最小巡回回路長しか探せていないことを示している。図 6 と表 4 は、改良 EX 法のまま遺伝子交叉を続ける場合に比べて、遺伝子交叉オペレータを交代して SXX 法で遺伝子交叉する方法の方が時間の経過と共に徐々に探索する最小巡回回路長が向上していることを示している。尚、SXX 単独の場合、最小巡回回路長は観測中 600 以上でありグラフにその全貌を現わせない。

この実験により、最適解を探索する機能と効率の両観点から、CXO 法が改良 EX 法と SXX 法のみならず他の 5 つの遺伝子交叉法より優れていることを検証した。

4. 結 論

本研究では、時間の経過と共に遺伝子交叉の方法を変化させるような遺伝的アルゴリズムの実現方法の有効性を、75 都市を対象とした C プログラミング実験により検証した。この結果、単独で改良 EX 法や SXX を適用するより、初期の段階では改良 EX 法を適用し後期の段階では SXX を適用する遺伝子交叉オペレータ法が有効であることを検証した。この実験結果は、各都市間の位置関係等、巡回セールスマン問題で対象としているデータの特徴を考慮して、遺伝子交叉法を交代させる方法が最適巡回回路を効率的に選択するのに有効であることを示唆している。

参 考 文 献

[1] J.H. Holland, *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press (1975), MIT Press (1992)

[2] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc., 1989.

[3] J. Grefenstette, R. Gopal, B. Rosmaita, D. Van Gucht, *Genetic Algorithms for the Traveling Salesman Problem*, Proc. of 1st Int. Conf. on Genetic Algorithms and Their Applications, pp. 160-168, 1985.

[4] 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, pp. 224-230, 1987.

[5] L. Davis, *Applying Adaptive Algorithms to Epistatic Domains*, Proc. Of 9th Int. Joint Conf. on Artificial Intelligence, pp. 162-164, 1985.

[6] 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, pp. 133-140 (1989).

[7] 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, pp. 535-540 (1996).

[8] K. Maekawa, N. Mori, H. Tamaki, H. Kita and Y. Nishikawa, *A Genetic Solution for the Traveling Salesman Problem by means of a Thermodynamical Selection Rule*, Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC '96), pp. 529-534, 1996.

[9] M. Garey, and D. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, W. H Freeman and Company, 1979.

[10] S.J. Russell and P. Norvig, *Artificial Intelligence—A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey, 1995.

[11] R.L. Haupt and S.E. Haupt, *Practical Genetic Algorithms*, John Wiley & Sons, Inc., 2004.

[12] E. Aarts and J.K. Lenstra, *Local Search in Combinatorial Optimization*, Princeton University Press, 2003.

[13] H. Hirano, *The Genetic Algorithm C programming*, pp. 17-28, pp. 232-238, Personal Media Inc., 1995 (in Japanese).

[14] R. Takahashi, *solving the Traveling Salesman Problem through Genetic Algorithms with changing cross over operators*, Proc. of ICMLA2005, pp. 319-324, IEEE Computer Society, 2005.