

拡張遺伝子交叉オペレータ交代法の有効性の検証

高橋良英*・吉川克哉**・小林史和***

A validation of Extended Changing Crossover Operators

Ryouei TAKAHASHI*, Katsuya YOSHIKAWA** and Fumikazu KOBAYASHI***

Abstract

In order to efficiently obtain an approximate solution of the traveling salesman problem (TSP), extended changing crossover operators (ECXO) which can substitute ant colony optimization (ACO) for genetic algorithms (GAs) at any time is proposed. In our study ECXO uses ACO in early generations to create local optimum sub-paths, and it uses EAX (Edge Assembly Crossover) to create a global optimum solution after generations. With ACO any individual or any ant determines the next city he visits from lengths of edges or tours' lengths deposited on edges as pheromone, and he generates local optimum paths. With EAX a parent exchanges his edges with another parent's ones reciprocally to create sub-cyclic paths, before restructuring a cyclic path by combining the sub-cyclic paths with making distances between them minimum. In this paper validity of ECXO is verified by using detailed analysis of four hundred forty two city problem pcb442 in TSPLIB. From C experiments, we can see that the above ECXO(ACO \rightarrow EAX), where ACO delivers chromosomes they generated to EAX, can find the best solution earlier than EAX alone.

Keywords : ECXO, GA, ACO, EAX, TSP

1. はじめに

巡回セールスマン問題 (TSP: Traveling Salesman Problem) ^{(4) (21)} の解法として, (i) 生物進化のメカニズムを遺伝子交叉や突然変異で擬似する遺伝的アルゴリズム (GAs: Genetic Algorithms) ^{(3) (4)} (ii) 蟻の採集行動における群知能の学習メカニズムを擬似するア

ントコロニー最適化手法 (ACO: Ant Colony Optimization) ^{(1) (16)} (iii) 物質を高温から低温に徐々に下げていき安定した素材を得る焼きなまし手法を擬似するシミュレーテッドアニーリング手法 (SA: Simulated Annealing) ⁽¹⁵⁾ (iv) 巡回路を λ 個の部分経路に分割し, 分割した λ 個の部分経路間をつないでいる連結子 (links) を他の連結子と交換しながら更に短い巡回路の探索を行なう Lin-Kernighan 法 ^{(13) (14)} (その連結子交換法は, 連結子数 λ を 2 から始めて 3, 4 というように変化させながら最適解探索を行なう variable λ -opt アルゴリズムである。) が提案されている。これまでの検討結果によれ

平成 21 年 12 月 14 日受理

* 大学院工学研究科電子電気・情報工学専攻・教授

** 大学院工学研究科電子電気・情報工学専攻・博士前期課程・1 年

*** 大学院工学研究科電子電気・情報工学専攻・博士前期課程・1 年

ば、各手法単独では局所最適解に陥り大域的に最適な解を探索することが困難なため、この問題を解決する手法として上記のオペレータを組み合わせて最適解を探索するハイブリッド手法が着目されるようになった。

TSP を解くハイブリッド手法として以下が知られている。

(i) Edge Assembly Crossover(EAX)^{(11) (12)} と Chained Lin-Kernighan(CLK) 法^{(18) (19)} を組み合わせた Heterogeneous Selection Evolutionary Algorithm(HeSEA)⁽¹⁷⁾ (ii) Distance Preserving Crossover(DPX)⁽²⁰⁾ (iii) Compact Genetic Algorithms(CGA)⁽²²⁾ と CLK を組み合わせた方法、(iv) ACO⁽¹⁾ と 2-opt や 3-opt を組み合わせた方法⁽¹⁶⁾。

本検討では、TSP の最適解の近似を効率的に得る新しいハイブリッド手法として GAs や ACO 等の遺伝子交叉オペレータを任意の時点で交代可能な拡張遺伝子交叉オペレータ交代法 (ECXO: Extended Changing Crossover Operators)^{(29) (30) (31)} を研究する。ECXO では、ある遺伝子交叉オペレータが探索した巡回路情報を、他の遺伝子交叉オペレータが任意の時点で引継ぎ更新可能である。これにより、巡回路の多様性を確保しつつ効率的に最適な解の探索を可能とする。特に、巡回路の多様性を確保しつつ効率的に最適な解の探索をする手法として近年着目を浴びている枝組み立て交叉 (EAX: Edge Assembly Crossover) を主制御対象の遺伝子交叉オペレータとする ECXO を検討した。本 ECXO では比較的早期の世代では遺伝子交叉オペレータ EX または ACO ならびに EXX により局所的に最適で多様な解を生成し、これを入力として後期の世代では遺伝子交叉オペレータ EAX により大域的に最適な解を効率的に生成する。その際、EAX は 2-opt や 3-opt と同様なローカルサーチ⁽¹⁵⁾ として働く。EAX がローカルサーチとしても有効に働くことは論文⁽¹¹⁾ に述べられている。当 ECXO の有効性を、中規模 TSPLIB データ⁽²⁶⁾ を用いた C プログ

ラミング実験で検証したので報告する。

論文⁽³³⁾ では lin318 の詳細データで ECXO の有効性を示した。本論文では pcb442 の詳細データで ECXO の有効性を検証する。

2. 巡回セールスマン問題

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 (または所要費用や所要時間) 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) という。 n が 20 以上と大きくなるとコンピュータでも実効上計算が困難となる。このような意味で TSP は n に関する多項式時間で解くことが困難な NP クラスに属する問題であると理解されている^{(15) (23)}。

3. 拡張遺伝子交叉オペレータ交代法

〈3・1〉拡張遺伝子交叉オペレータ交代法 ECXO (EX (or ACO) → EAX)

主遺伝子交叉オペレータを EAX、副遺伝子交叉オペレータを EX または ACO とする拡張遺伝子交叉オペレータ ECXO (EX (or ACO) → EAX) の概念を図 1 に示す。両親が部分的に巡回路の長さが最短ならばそれを結合して大域的に最短な解を生成する能力に関しては、EAX は EX または ACO より優れている。しかし、枝の長さに関する情報から子供を生成する機能がないため、EAX 単独では局所的に最適な解を生成するのに時間がかかる。一方副遺伝子交叉オペレータ EX または ACO は両親の枝の長さに関する情報から子供の経路を決定するため局所的に最適な解を生成する能力は EAX より優れているが、大域的な枝と枝のつながりに関する情報から子供を生成できないため局所最適解に陥りやすい。このため、本検討の ECXO では比較的早期の世代では遺伝子交叉オペレータ EX または ACO により局所的に最適な経路を生成し、後期の世代で

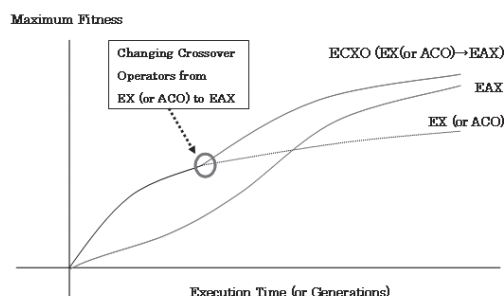


Fig. 1. Concept of performance improvement to solve TSP by ECXO (EX(or ACO) → EAX)

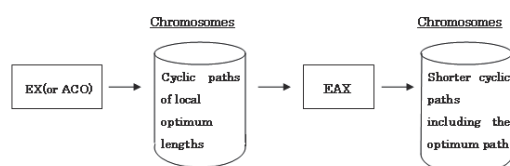


Fig. 2. In ECXO(EX (or ACO) → EAX), EX(or ACO) delivers chromosomes generated by himself to EAX.

はEAXにより大域的に最適解を生成する。ECXOはEXまたはACOが局所的に最適解を生成した後で、かつそれによる解の収束が始まり多様性が確保できなくなる前にEXまたはACOからEAXに交代させる。図1の○印の時点でEXまたはACOからEAXに遺伝子交叉オペレータを交代させる。図2に、ECXO (EX (or ACO) → EAX) の実現方式を示す。図2に示すように、EAXはEXまたはACOが生成した解をChromosomesというファイルで引き継ぐ。ECXOは、ACOまたはEXが生成した遺伝子情報群Chromosomesの多様性 (Diversity) と収束度 (Convergence) を監視し各々が α_0 かつ β_0 以上の値をとった時に遺伝子交叉オペレータをACOまたはEXからEAXに交代させる。その際、 α_0 、 β_0 はECXOの走行パラメータとして与える。個体が様々な経路を持つことが多様性であり、その多様性を「(現世代を構成する個体の最大適応度 - 現世代を構成する個体の平均適応度) / (現世代を構成する個体の最大適応度)」で測定する。解が収束するとは現世代までに探索した個

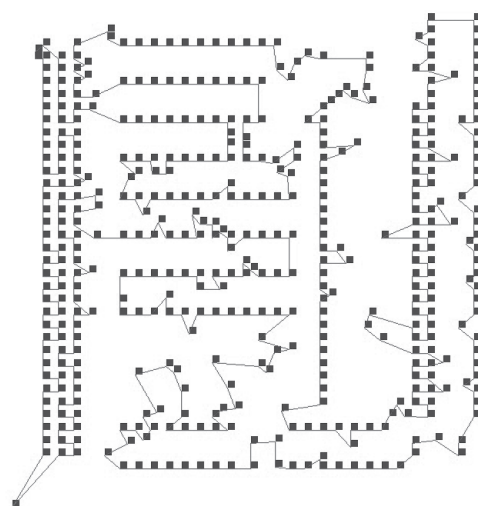


Figure 5. The optimum solution of pcb442 found by ECXOs. It has the minimum length of 50,778.

体の最大適応度が最適適応度に近づくことであり、その収束度を「(これまでに探索した個体で最大適応度を持つ個体 (最良解) の最大適応度) / (最適適応度)」で測定する。最適適応度とは TSPLIB⁽³³⁾ で管理されている最短経路を持つ個体 (最適解) の適応度である。個体の適応度は「定数 / 巡回経路長」で測定する。詳細は5節に述べている。

〈3・2〉拡張遺伝子交叉オペレータ交代法 ECXO (EX (or ACO) → EXX → EAX)

拡張遺伝子交叉オペレータ ECXO (EX (or ACO) → EXX → EAX) は初期世代, 中期世代, 後期世代の三段階で遺伝子交叉オペレータを交代させる。その主遺伝子交叉オペレータは遺伝子交叉オペレータ EAX であり、個体の多様性を確保し最適解への収束効率を向上させるために、遺伝子交叉オペレータ EX または ACO の遺伝子操作結果と EXX の遺伝子操作結果を二段階に分けて引き継ぐ。まず、副遺伝子交叉オペレータ EX または ACO は部分的に最適だが全体として経路長にばらつきのある多様な個体群 X を生成する。次に、この遺伝子群 X を初期値として遺伝子交叉オペレータ EXX に

Table 1. Comparison of ECXOs with EAX, EX, EXX and ACO on best length, average length, the number of optimal trials, relative error, and computer time to find the best length. Results are obtained from fifteen independent trials for pcb442.

No.	Method	Best length	Average length	No. optimum trials	Relative error*	Minimum computational time (sec.)	Average computational time (sec.)
1	EAX	50,778	50,780	13	0.00004	10,446	11,186
2	EX	53,490	54,122	0	0.06585	3,198	3,198
3	EXX	55,397	56,589	0	0.11444	1,049	1,049
4	ACO	52,243	54,060	0	0.06464	50,738	50,738
5	ECXO(EX → EAX)	50,778	50,778	15	0.00000	8,375	8,794
6	ECXO(ACO → EAX)	50,778	50,778	15	0.00000	9,146	9,932
7	ECXO(EX → EXX → EAX)	50,778	50,780	14	0.00004	8,431	9,449
8	ECXO(ACO → EXX → EAX)	50,778	50,778	15	0.00000	9,780	10,511

Relative error is defined as ((Average length)/(Optimal length) - 1)

より解 w への仮収束を図る。遺伝子交叉オペレータ EXX は局所最適解には陥りやすいものの EAX に比べて解の収束効率が良い。そして遺伝子交叉オペレータ EX または ACO で生成した遺伝子群 X に遺伝子交叉オペレータ EXX の仮収束解 w を併合させ遺伝子群 $X' = X \cup w$ を生成した後 EAX の初期値として X' を使用し広域的に最適な解をより効率的に探索する。こうして、主遺伝子交叉オペレータ EAX の解の多様性と最適解への収束効率の向上を、経路長にばらつきのある遺伝子群 X と X より巡回路長の短い個体 w を混在させることで実現している。EXX 経由の場合は EXX の収束度があまり高くない段階で EAX に遺伝子情報を引き継ぐことが有用と判断される。尚、ACO (または EX) から EXX へのオペレータ交代時期、ならびに EXX から EAX に引き渡す遺伝子 w を生成する時期は先と同様に「多様性」と「収束度」で制御する。

4. 実験結果

〈4・1〉実験データ

論文^{(16) (12)}で参照されている TSPLIB95⁽³²⁾の 442 都市問題 pcb442 を用いて、EAX を主遺伝子交叉オペレータとする EAX の有効性を

検証した。TSPLIB によれば、これまでに見つかっている pcb442 の最短経路長は 50,778 であり、本実験においても ECXO と EAX がこれと同じ最短経路を探索した。図 5 に探索した pcb442 の最短経路を示す。二都市間の距離は小数点以下四捨五入した整数である⁽³²⁾。

〈4・2〉実験環境

数値実験は、NEC VersaPro Mobile Intel (R) CPU, 2.19GHz, 240MB RAM 上の Windows XP 上で行なった。

〈4・3〉評価対象の遺伝子交叉オペレータ

以下の 8 つの遺伝子交叉オペレータについて機能と性能を評価した。

- ① EAX ② EX ③ EXX ④ ACO
⑤ ECXO(EX → EAX) ⑥ EXO(ACO → EAX)
⑦ ECXO(EX → EXX → EAX) ⑧ EXO(ACO → EXX → EAX)

〈4・4〉評価項目

4.3 の遺伝子交叉オペレータを以下の観点で評価した。

- ① 「探索した最短経路長 L 」
② 「最短経路長を探索するまでに要した個体数 N 」

Table 2 Typical comparison of ECXOs with other operators EAX, EX, EXX and ACO on performance to find best lengths of pcb442. Results are obtained from a run of the seed_id of one. The symbol * indicates the best model.

NO.	method	Best length	time (sec.)	no. generated tours	G.C.C.	No. generations
1	EAX	50,778	10,487	609,960		23
2	EX	53,590	9,843	967,096		1,094
3	EXX	55,532	1,089	4,294,472		4,858
4	ACO	55,767	22,303	65,416		148
5*	ECXO(EX->EAX)	50,778	8,923	543,660	15	35
6	ECXO(ACO->EAX)	50,778	9,460	530,842	1	21
7	ECXO(EX->EXX->EAX)	50,778	8,948	2,144,584	10, 1846	1,875
8	ECXO(ACO->EXX->EAX)	50,778	9,837	2,148,562	3, 1829	1,852

G.C.C. denotes the number of generations to change crossover operators

最適解を探索するのに必要な生成個体（経路）数 N は EX, EXX では「集団サイズ×最短経路長を探索するまでに要した世代数×2」, EAX では「集団サイズ×最短経路長を探索するまでに要した世代数×2×Ncross」, ACO では「集団サイズ×最短経路長を探索するまでに要した世代数」で計算する。GAsにおいては、両親が子供を2人生成し、適応度の高い子供を残す方式としているため生成個体は集団サイズの2倍とした。Ncross は両親から生成される子孫の最大世代数であり収束が開始すると実効上1に近づく。

③「最短経路を探索したコンピュータ時間（秒）T」

〈4・5〉プログラム起動パラメータ

本実験での ACO の起動パラメータは Dorigo と Stutzle の教科書⁽¹⁾に掲載されている実験結果を参考にして、フェロモン更新タイミング $m = 442$, フェロモン揮発率 $\rho = 0.3$, 距離の重み $\beta = 2$ とした。GAs(EAX, EXX, EX) の起動パラメータは ACO と性能を横並びに比較可能とするため、集団サイズ = 442 とした。また、都市数が中規模なので Ncross = 「30」と論文⁽¹¹⁾に掲載されている値より小さな値にした。起動パラメータの詳細は以下の通りである。

（1）GAs の起動パラメータ

（A）EAX, EX, EXX 共通の GAs 起動パラメータ

- ①乱数種：初期集団を決定する。：1 から 15 の整数。
- ②集団サイズ：442
- ③遺伝子交叉確率：0.8
- ④最大観測世代数：30（EAX）, 500（EX）, 5000（EXX）
- ⑤親の選択方式：ルーレット方式（EX, EXX のみ）
- ⑥世代交代方式：次世代の子が「集団サイズ」分生成されたタイミングで生成した子を次世代の親として一括世代交代する。
- ⑦親と子供を併せた個体群の間でトーナメントを行ない適応度の高い個体を残す。：YES
- ⑧生成した2人の子供のうち適応度の高い子供のみ残す。：YES

（B）EAX 固有の GAs 起動パラメータ

- ①親 A として現世代の個体全てを対象として選ぶ。もう一方の親 B を、現世代の個体から任意に確率的に1個選択して遺伝子交叉を行い2個の個体を生成する。
- ②3.2⁽¹⁾③で述べた「エリート戦略」における Ncross 回数：30

Table 3 Selection of the best generation to change crossover operators from EX to EAX.

NO.	method	Best length	time (sec.)	no. generated tours	G.C.C.	No. generations
5-1	ECXO(EX->EAX)		9,885	587,860		27
5-1-1	EX	85,807	49	4,420	5	5
5-1-2	EAX	50,778	9,836	583,440		22
5-2	ECXO(EX->EAX)		9,392	592,280		32
5-2-1	EX	61,801	93	8,840	10	10
5-2-2	EAX	50,778	9,299	583,440		22
5-3*	ECXO(EX->EAX)		8,923	543,660		35
5-3-1	EX	59,785	133	13,260	15	15
5-3-2	EAX	50,778	8,790	530,400		20

Results are obtained from a run of the seed_id of one. The best model is referred in Table 2.

(2) ACO の実行パラメータ ^{(1) (25)}

(a) 数理モデル

以下は ACO の数理モデルの概略である。

①都市 i に居る蟻 A_k が都市 j を訪れる確率 $p_{ij}^k(t+1)$

時間 $t+1$ に都市 i にいる蟻 A_k は次に訪れる都市 j を、都市 i と都市 j を結ぶ枝上に残された t 時におけるフェロモン量 $\tau_{ij}(t)$ と、都市 i と都市 j 間の距離 d_{ij} から決定される以下の確率 $p_{ij}^k(t+1)$ により選択する。ここで蟻 A_k はこれまで q 都市のうち r 個の都市について訪問が終わっているものとし、 $q-r$ 個の未訪問の都市から次に訪れる都市を決定する。

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

②ローカルフェロモン量更新

フェロモン量 $\tau_{ij}(t+1)$ は、現在のフェロモン量 $\tau_{ij}(t)$ と $t+1$ の旅で更新したフェロモン更新量 $\Delta \tau_{ij}(t+1)$ から、 m 匹の蟻が旅を完了する度に、以下の式で更新される。時間 t は m 匹の蟻が旅を完了する度に更新される。

$$\begin{aligned} \Delta \tau_{ij}(t+1) &= \sum_{k=1}^m Q_{ij}(k), \\ \tau_{ij}(t+1) &= (1-\rho) * \tau_{ij}(t) + \rho * \Delta \tau_{ij}(t+1) \end{aligned}$$

式 2

ここで、 ρ はフェロモンの忘却度である。は蟻 k が i から j の経路を選択する大きさを測る尺度であり、「蟻 k が、都市 i から j を経由する巡回路を旅した場合は、「定数／巡回路長」、すなわち「GAs での適合度」であり、都市 i から j を経由しない場合は 0」である。

(b) 実行パラメータ

①乱数種：主に蟻の初期旅行経路と経路長を決定する。：1

②次に訪問する都市を隣接する cl 個の都市の中から優先的に次に訪れる都市を選択する際の cl を指定する。：20

③集団サイズ p 。ACO は p 匹の蟻の旅行経路と経路長に関するデータを GAs に引き継ぐ。 p は蟻が初めて訪れる異なる都市の数である。：442

④観測最大世代数：200

⑤都市 i と都市 j 間の辺上に蓄積されるフェロ

モン量の辺の長さによる重みづけ β (式1): 2

⑥フェロモン忘却率 ρ (式2): 0.3

⑦フェロモン更新タイミング m ; 何匹の蟻の旅度にフェロモンを更新するか: 442

⑧蟻の探索経路に 2-opt 法と 3-opt 法を 1 回ずつ適用して距離の改良を図る SA (シミュレートッド・アニーリング) を利用するか否か。本 SA では、距離の短縮度合いとこれまでの経過時間を考慮しながら、2-opt 法と 3-opt 法を適用して蟻の探索経路を確率的に変更・改善する。本 SA では経路長が短くなった時に次状態に確率的に遷移する。そうでない場合は遷移を保留する。: YES

〈4・6〉実験結果

(1) ECXOs による性能改善効果の統計的評価 EAX, EX, EXX, ACO ならびに 4 つの ECXO モデル (EX \rightarrow EAX, ACO \rightarrow EXX \rightarrow EAX, EX \rightarrow EXX \rightarrow EAX, EX \rightarrow EXX \rightarrow EAX, ACO \rightarrow EAX) を併せた合計 8 つの交叉モデルについて、乱数種を 1 から 15 まで変化させて 15 回の独立な試行を行ない、(i) 探索した最短経路の中で最も短い経路長 L_{min} , (ii) 探索した最短経路長の平均値 $L_{average}$, (iii) 最適経路探索に成功した回数, (iv) TSPLIB に登録されている最適経路長 $L_{optimal}$ と平均経路長 $L_{average}$ 間の相対誤差 e ($= (L_{average} / L_{optimal} - 1)$, 一般に $L_{min} \geq L_{optimal}$)⁽¹⁶⁾, 最短経路を探索するのに要した最短探索時間 (単位: 秒) と平均探索時間 (単位: 秒) を評価した。その結果を表 1 にまとめる。表 1 から以下のことがわかった。① EAX と 4 つの ECXOs が最適経路を探索できていること, ②最適経路発見回数は EX \rightarrow EAX, ACO \rightarrow EAX, ACO \rightarrow EXX \rightarrow EAX が各々 15 回, EX \rightarrow EXX \rightarrow EAX が 14 回と高いこと, ③相対誤差 e に関して上記の ECXO 群が最も小さいこと, EXX が $e=0.114$ と最も e が大きいこと, ④最適解探索時間の平均値に関し 4 つの ECXO は全て EAX より早く最適解

を探索できていること, ECXO (EX \rightarrow EAX) 8,794 秒, ECXO (EX \rightarrow EXX \rightarrow EAX) 9,449 秒, ECXO (ACO \rightarrow EAX) 9,932 秒, ECXO (ACO \rightarrow EXX \rightarrow EAX) 10,511 秒, EAX 11,186 秒の順番で最適解に到達していること, CPU 時間の短縮率は従って 6% から 21% であること, 最適解探索の最小時間も同様な傾向を示していること。尚, 各 ECXO モデルは乱数種のみならずオペレータの交代時期によって探索できる最短経路や探索時間が異なる。表 1 では ECXO (EX \rightarrow (EXX \rightarrow) EAX) の場合は EX から EAX または EXX への交代時期を 5, 10, 15 世代目とした 3 つの ECXOs の中で最短経路を探索した ECXO を, ECXO (ACO \rightarrow (EXX \rightarrow) EAX) の場合は ACO から EAX または EXX への交代時期を 1, 2, 3 世代目とした 3 つの ECXOs の中で最短経路を探索した ECXO を各 ECXO 種別代表の ECXO として選択し統計処理している。尚, 表 1 は $N_{cross}=30$ のもとでの評価結果である。

(2) 「探索した最短経路長 L 」, 「最短経路を探索したコンピュータ時間 (秒) T 」, 「最短経路長を探索するまでに要した個体数 N 」の評価 4.3 で述べた 8 つの遺伝子交叉オペレータについて pcb442 の例題データを用いて L , T , N を評価した結果を表 2 に示す。表 2 で* のついたモデルは最も少ないコンピュータ時間で最短経路長 =50,778 を探索したモデルである。尚、試験は乱数種 =1 の下で行った。

(i) EAX は 23 世代目 10,487 秒で最短経路長 50,778 を探索していること,

(ii) EX 単独では経路長 53,590 の解しか探索できないこと, EXX 単独では経路長 55,532 の解しか探索できないこと, ACO 単独では経路長 55,767 の解しか探索できないこと,

(iii) EX から 15 世代目に EAX に交代する ECXO (これを ECXO (EX \rightarrow EAX) と記す) は 35 世代目 8,923 秒で最短経路長 50,778 を探

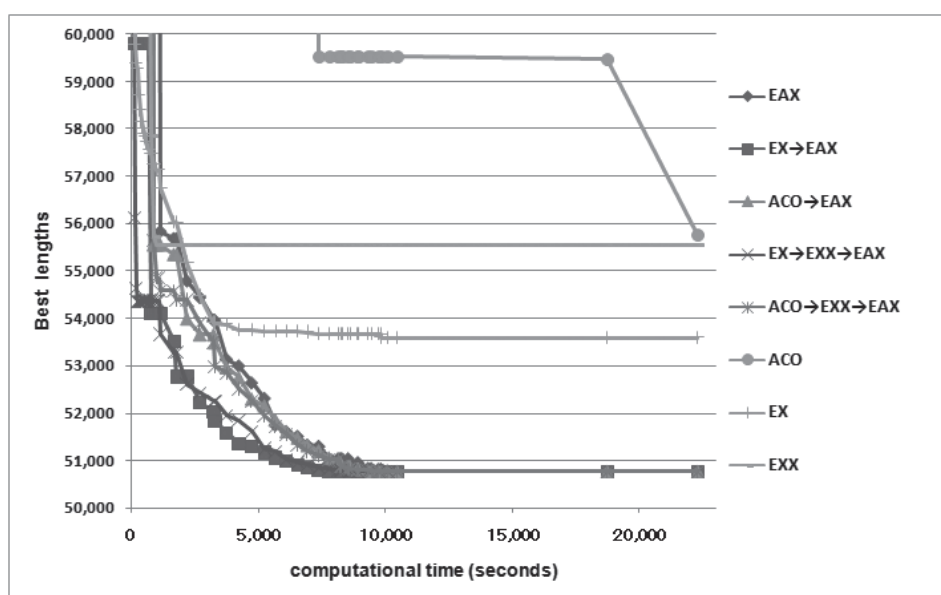


Figure 6. Evaluation of both of ECXOs and other crossover operators on computer time to find their best lengths of pcb442.

索していること,

(iv) ACO から 1 世代目に EAX に交代する ECXO (ACO → EAX) は 21 世代目 9,460 秒で最短経路長 50,778 を探索したこと,

(v) EX から 10 世代目に EXX に交代し, EXX から EAX に 1,846 世代目に交代する ECXO (EX → EXX → EAX) は 1,875 世代目 8,948 秒で最短経路長 50,778 を探索していること, ここで ECXO (EX → EXX → EAX) では解の多様性を確保しかつ効率的な最適解の探索を可能とするため EX 法で探索した 441 個の解に EXX 法で探索した最適解 1 個を混在させた 442 個の遺伝子群を EAX の入力としている。

(vi) ACO から 3 世代目に EXX に交代し, EXX から EAX に 1,829 世代目に交代する ECXO (ACO → EXX → EAX) は 1,852 世代目 9,837 秒で最短経路長 50,778 を探索していることを示している。ここで (v) と同様に ACO で探索した 441 個の解に EXX 法で探索した最適解 1 個を混在させた 442 個の遺伝子群を EAX の入力としている。

上記のことから, ECXO (EX → EAX), ECX

O(ACO → EAX), ECXO (EX → EXX → EAX), ECXO (EX → EXX → EAX) の各遺伝子交叉オペレータ交代法はいずれも最適解を探索できること, いずれも EAX より短い時間で最適解を探索できること, 本事例では ECXO (EX → EAX) が最も探索時間を削減できその割合は 15% であった。

(2) ECXO における遺伝子交叉オペレータ交代時期の選択

① ECXO (EX → EAX) における EX から EAX への交代世代 (表 3)

5 世代目, 10 世代目, 15 世代目に EX から EAX に遺伝子交叉オペレータを交代させる ECXO (EX → EAX) について調査した結果, 各々最適解 50,778 を 9,885 秒, 9,392 秒, 8,923 秒で探索した。いずれも EAX が最適解を探索するのに要した 10,487 秒より少ないコンピュータ時間で最適解を探索した。そのうち 15 世代に EX から EAX に交代させる ECXO (EX → EAX) が最も短い時間で最適解を探索できた。表 2 で示した ECXO (EX → EAX)

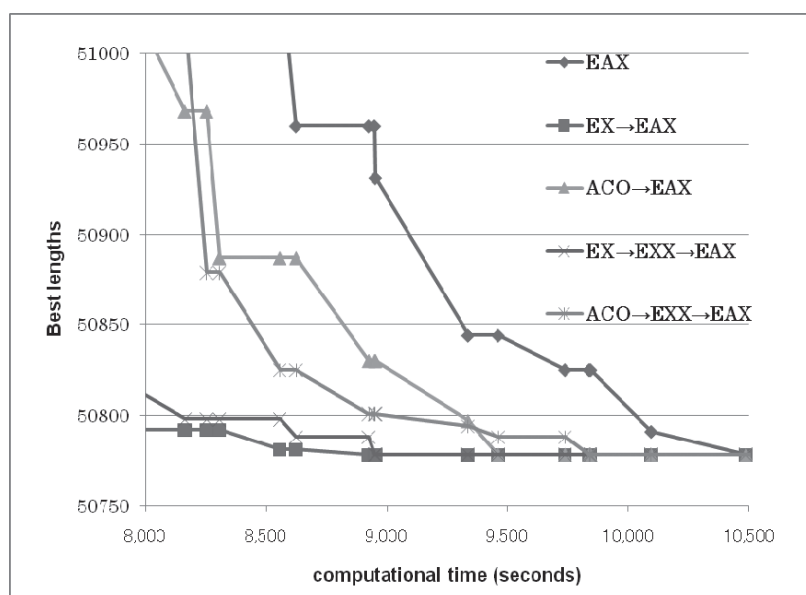


Figure 7. Some ECXOs converge to the optimum solution earlier than EAX on the problem pcb442.

は15世代にEXからEAXに交代させるモデルのECXO (EX → EAX) である。

② ECXO (ACO → EAX) における ACO から EAX への交代世代

1世代目, 2世代目, 3世代目にACOからEAXに交代させるECXO (ACO → EAX) について調査した結果, 各々最適解 50,778 を 9,460 秒, 9,797 秒, 10,158 秒で探索した。いずれもEAXが最適解を探索するのに要した10,487秒より少ないコンピュータ時間で最適解を探索した。表2で示したECXO (ACO → EAX) はその中で最も短い時間で最適解を探索した1世代にACOからEAXに交代させるモデルのECXO (ACO → EAX) である。

③ ECXO (EX → EXX → EAX) における EX から EXX への交代世代

5世代目, 10世代目, 15世代目にEXからEXXに交代させEXXで仮収束を図り, その後EAXで最終的な解の収束を図るECXO (EX → EXX → EAX) について調査した結果,

各モデルは最適経路長 50,778 を 11,757 秒, 8,948 秒, 12,263 秒で探索した。この結果, 10世代目でEXからEXXに交代させEXXで仮収束を図ったモデルが最も少ないコンピュータ時間で最適解を探索できることがわかった。表2に示したECXO (EX → EXX → EAX) は10世代にEXからEXXに交代しその後EXXからEAXに交代するモデルである。

④ ECXO (ACO → EXX → EAX) における ACO から EXX への交代世代

1世代目, 2世代目, 3世代目にACOからEXXに交代させEXXで仮収束を図り, その後EAXで最終的な解の収束を図るECXO (ACO → EXX → EAX) について実験した結果, 各モデルは最適経路長 50,778 を 11,667 秒, 10,267 秒, 9,837 秒で探索した。表2で示したECXO (ACO → EXX → EAX) は3世代目にACOからEXXに交代させEXXで仮収束を図るECXO (ACO → EXX → EAX) モデルである。

(3) 最適解探索のプロセス

図 6 のグラフは EAX の各世代の終了時刻で EAX ならびに他の 7 つの遺伝子交叉オペレータが探索した最短経路長がどのように変化しているかを示している。グラフの X 軸は実行時間、Y 軸はその時刻までに探索した最短経路長を示す。図 6 から、(i) 2 世代目完了時刻 1,089sec では EAX が探索した最短 67,597 より短い経路長 57,133 を EX は探索しているが、4 世代目完了時刻 9,843sec 以降、EX の解は局所最適解 53,590 に陥り解の改善が見られないこと、(ii) 1 世代目完了時刻 1,089sec 以降、EXX は局所最適解 55,532 に陥り解の改善が見られないこと、(iii) EAX が最適解 50,778 を探索した 23 世代目完了時刻 18,745sec で ACO は 59,460 の局所最適解した探索できていないことがわかる。図 7 は 16 世代目以降を詳細化したグラフであり、EAX ならびに 4 つの ECXO の最適解 50,778 への収束速度を比較している。表 2、図 7 は 8,923sec で最初に ECXO (EX → EAX) が最適解を探索し、次に ECXO (EX → EXX → EAX), EAX, ECXO (ACO → EAX), ECXO (ACO → EXX → EAX) の順番で最適解を探索し、最後に 10,487 秒で EAX が単独で最適解を探索していることを示している。

〈4・6〉他の TSP 問題による ECXO の有効性の検証

pcb442 に加え論文 (12), (16) で参照されている lin318, d198 に対してそれぞれ乱数種を変えた独立な 15 回の ECXO 方式の試験を行い pcb442 と同等の性能評価結果を得ている。

5. 多様性と収束性による遺伝子交叉オペレータ交代時期の最適制御

〈5・1〉多様性と収束性の測定法

ACO (または EX) が、必要があれば EXX 等を経由して、主遺伝子オペレータ EAX に最新の集団サイズ分の旅行に関する遺伝子情報を

Chromosomes というファイル経由で引き継ぐ場合、EAX の収束効率に影響を与えるのは引き継ぐ遺伝子情報の多様性 (diversity) と収束度 (convergence) である。個体が様々な経路を持つことが多様性であり、それは個体の適応度のばらつきの大きさとなって現れる。このため、多様性を $Div = \frac{(\text{現世代を構成する個体の最大適応度} - \text{現世代を構成する個体の平均適応度})}{(\text{現世代を構成する個体の最大適応度})}$ で測定する。多様性が大きければ Div は 1 に近づき、多様性が小さければ Div は 0 に近づく。解が収束するとは現世代までに探索した個体の最大適応度が最適適応度に近づくことであり、その収束度を $Conv = \frac{(\text{これまでに探索した個体で最大適応度を持つ個体 (最良解) の最大適応度})}{(\text{最適適応度})}$ で測定する。最適適応度とは TSPLIB⁽³³⁾ で管理されている最短経路を持つ個体 (最適解) の適応度と定義する。最適適応度に最良解が近づけば Conv は 1 に近づき、遠のけば Conv は 0 に近づく。ここで個体の適応度は「定数/巡回経路長」で測定する。遺伝子交叉を繰り返すと多様性を失い最良解 (大域的に最適な場合もある) への収束速度が早くなる現象は GAs における「積み木仮説」の教えるところである⁽²⁾。従って、「多様性を保ちながら、最適解への収束速度を向上させる」という、相反する条件を満たす最適解探索手法、すなわち「積み木仮説」を階層的に適用して最適解を探索する手法、を研究することが TSP を解く時の課題となる^{(11) (12)}。ECXO では上記 Div と Conv の二つのパラメータで遺伝子交叉オペレータの交代時期を決定することによりこの課題を解決する。

〈5・2〉多様性と収束性の世代に対する変化状況の調査ならびに仮説の設定

pcb442 を例にして多様性 (diversity) と収束度 (convergence) の世代に対する変化の様子を、図 8 と図 9 に示す。EX で個体を生成する場合の変化の様子を図 8 に、ACO で蟻が最

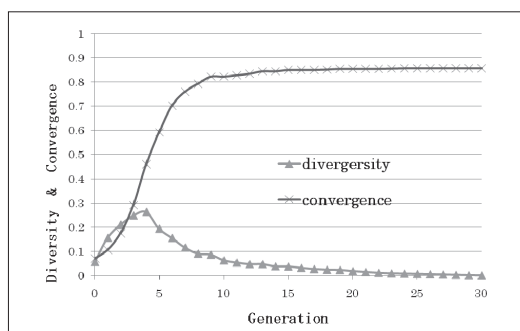


Fig. 8. How diversity and convergence change as generation passes on EX operation.

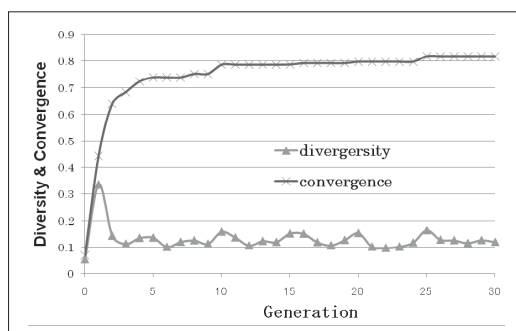


Fig. 9. How diversity and convergence change as generation passes on ACO operation.

短経路を学習する場合の変化の様子を図9に示す。図18と図29から(i)EXもACOの場合も、初期の世代においては局所最適解にばらつきがあるため、多様性と収束度はいずれも上昇すること、(ii)EXでは、ある世代G1を超えると(例では4世代目)、多様性は下降し始め0に近づくこと、これに伴い局所最適解への収束が始まり、収束度は急激に増加していくこと、(iii)ACOにおいても、ある世代G2を超えると(例では1世代目)、多様性の上昇は止まり一旦下降すること、そしてある範囲内の多様性(0.10から0.20)を上下昇しながら解の探索を続行していくこと、このため収束速度はEAXに比べてゆっくりしていることがわかる。このため、ACO(またはEX)からEAXへの交代時期は「初期世代=1」と「多様性の増加が最大となり局所最適解への収束が始まるG2(またはG1)」との間が最適解探索の確率が最も高くなると考えられる(仮説I)。この仮説に基づきECXOは多様性(diversity)と収束度(convergence)という二つの変数の変化状況を観察し、各々が閾値 α かつ β 以上の値をとった時に遺伝子交叉オペレータを交代させる。

〈5・3〉多様性と収束性による最適解探索領域の特徴づけ

実際、上記の仮説Iを検証するため、多様性(diversity)と収束度(convergence)の最適解探索効率への影響度を標本から調査し

た。①ACOからEAXへの交代時期T1、②ACOからEXXに交代する時期T2、③EXからEAXへの交代時期T3、④EXからEXXに交代する時期T4の各時期で集団の多様性と収束度を測定しそれを初期値としてEAXが最適解を探索した回数との関係について調査した。EAX、EX、ACO、EXXの各手法の個体生成効率を考慮して、T1、T2については1、2、3の各世代の終わり、T3、T4については5、10、15の各世代の終わりで調査した。

表4はpcb442を例題として、各手法、各交代世代候補毎に、①乱数種を1から15まで変化させた15回の独立試行中に、何回最適経路長(=50,778)を探索したか、②ACO(またはEX)から、(必要に応じてEXX経由で、)EAXが引き継いだときの遺伝子情報の多様性と収束度の平均値を示している。表4は、各EAXが最適解を探索する確率が、ACOからEAXに交代する方式が $39/45=0.87$ 、ACOからEXX経由でEAXに交代する方式が $39/45=0.87$ 、EXからEAXに交代する方式が $39/45=0.87$ 、EXからEXX経由でEAXに交代する方式が $27/45=0.60$ と、ACOからEAX(またはACOからEXX経由で)に引き継ぐ方式がEXからEAX(またはEXからEXX経由で)に引き継ぐ方式より最適解の探索確率が大きいことを示している。

Table 4. Characteristics of diversity and convergence of chromosomes which are delivered to EAX through ACO, EX or EXX (in a case of pcb442).

opeartions	G.C.C.	no. optimal trials	avg. diversity	avg. convergence
ACO → EAX	1	14	0.35	0.45
	2	13	0.13	0.63
	3	12	0.12	0.69
ACO → EXX → EAX	1	12	0.68	0.91
	2	14	0.41	0.92
	3	13	0.34	0.93
EX → EAX	5	15	0.21	0.62
	10	15	0.07	0.83
	15	9	0.04	0.85
EX → EXX → EAX	5	10	0.47	0.92
	10	9	0.18	0.94
	15	8	0.14	0.95

Avg. Div. and Avg. Conv. indicate the average divergence and average convergence of the populations at which crossover operators are changed into the EAX. G.C.C. is defined in Table 2.

表 4 を変形して、初期データの多様性 x と収束度 y と EAX の最適経路長探索回数 z の関係 $z = f(x, y)$ を図 10 のグラフで表現する。グラフの X 軸は多様性、Y 軸は収束度、数値は $z = f(x, y)$ を示す。グラフから、最適解に収束する確率が高い領域には 2 つあることが読み取れる。それは、(i) 多様性を確保しながら収束度を向上させ始める初期の世代で EAX に交代させる収束領域 A と、(ii) 収束度がある程度高くなった中間段階で EAX に引き継ぐ収束領域 B である。収束領域 A で走行する ECXO モデルには、ACO の 1 世代目の終わり（多様性 = 0.35, 収束度 = 0.45）に EAX に交代させて高い最適解探索確率（ $14/15 = 0.93$ ）を確保するモデルと、多様性を確保しながら収束度を上昇させている段階（多様性 = 0.4 ~ 0.7, 収束度 = 0.9 ~ 0.93）で ACO や EX から EXX 経由で EAX に交代させることで高い最適解探索確率（ $(10+12+13+14) / (15+15+15+15) = 0.81$ ）を確保するモデルがある。この収束領域 A が、仮説 I が成立する領域である。収束

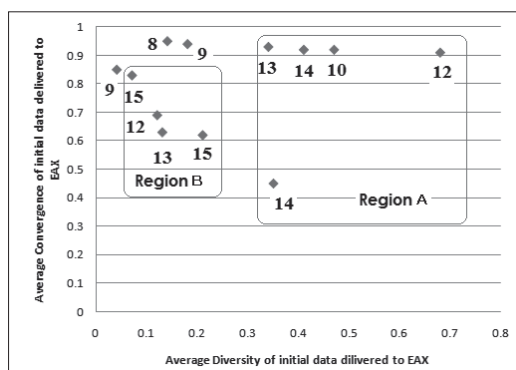


Fig. 10. Relationship between a number of optimal trials and diversity & convergence of populations at which crossover

領域 B は、多様性は 0.05 ~ 0.12 と低いものの最良解の収束方向が最適解の方向と一致したため、最適解を探索できた ECXO モデルの走行領域と解釈できる。その他の領域にある ECXO モデルは EX から必要があれば EXX 経由で EAX に引き継ぐ場合であり、調査事例ではその収束度が 0.85 以上と高すぎた遺伝子を EAX が引き継いだため局所最適解に陥り最適解の探索確率が低くなってしまった領域である。

〈5・4〉多様性と収束度の閾値を走行パラメータとする遺伝子交叉オペレータ交代方式

上記収束領域 A は EAX に遺伝子情報を引き渡せば最適解探索確率は高くなる多様性と収束度に関する領域であり、ACO や EX や EXX 等の各遺伝子操作オペレータに依存しない共通領域である。本実験では ACO から EAX に引き継ぐ方が EX から EAX に引き継ぐより最適解探索効率が高いことから、ACO と同一の最適解収束条件（多様性 = 0.14, 収束度 = 0.33）を満たす EX の世代を調べると、EX は 5 世代目ですでに多様性 = 0.20, 収束度 = 0.63 となっており、積み木仮説を考慮すると、5 世代目よりもっと早い段階で EX から EAX に交代することにより最適解を探索する確率と探索効率の向上が可能と考えられる。また EXX 経由の場合は EXX の収束度があまり高くない段階

でEAXに遺伝子情報を引き継ぐことが有効と考えられる。

以上のことからACOまたはEXについて、多様性と収束度が各々固定値 α 0かつ β 0以上の値をとった時に遺伝子操作オペレータを交代するECXO方式を検討することとした。 α 0, β 0の値は一般には集団サイズや扱うデータの複雑さに依存するので、ECXOの走行パラメータとして与えるようにした。

(5・5) 実験結果

(1) pcb442 最適解探索成功率：

本実験では集団サイズ = 442のもと、(i) α 0 = 0.10, β 0 = 0.15でACO (またはEX) からEAXまたはEXXに交代させ、(ii) EXXを経由して遺伝子情報をEAXに引き継ぐ場合は α 0 = 0.00, β 0 = 0.80でEXXを仮収束させた。上記パラメータで収束しない場合は多様性を確保しつつ収束度も向上し始める初期の世代のEXとACOの領域内(〈5・3〉で述べた領域Aのこと)でオペレータ交代が行なえるようにパラメータ値を調整した後にECXOを適用した。領域Aで収束しない場合は β 0 = 0.5 ~ 0.7と収束がやや進んだ段階でオペレータ交代させる領域Bで収束を図った。このような方法で4つのECXOに対して各々15回の独立な試験を行った結果、領域A内での最適解探索率は92% (=55/60)、領域B内での最適解探索率は8% (=5/60) 合計100%であった。実験結果は以下の通りである。

領域Aでのpcb442最適解探索成功率：

ECXO全体：55/60=0.92；内訳：ACO → EAX：14/15= 0.93,

ACO → EXX → EAX：15/15= 1, EX → EAX：14/15=0.93, EX → EXX → EAX：12/15= 0.80。

領域Bでのpcb442最適解探索成功率：

ECXO全体：5/60=0.08；内訳：ACO → EAX：1/15= 0.07, EX → EAX：1/15=0.07, EX → EXX → EAX：3/15= 0.20。

EAX: 13/15= 0.87。

尚、領域A内で最適解に収束したモデルのうちパラメータ修正で最適解を得たのは1回である。また領域B内で収束したモデルの全てについて多様性は α 0 \geq 0.2を満たしていた。

(2) d198最適解探索成功率、lin318最適解探索成功率はECXOで100%であった。

6. 結論

拡張遺伝子交叉オペレータ交代法ECXO (Extended Changing Crossover Operator) の有効性をTSPLIBの中規模TSPデータ(pcb442, d198, pcb442)を用いたCプログラム実験で検証した。当ECXOは比較的早期の世代では遺伝子交叉オペレータEXまたはACOならびにEXXにより局所的に最適で多様な解を生成し、これを入力として後期の世代では遺伝子交叉オペレータEAXにより大域的に最適な解を効率的に生成する。実験の結果、ACOやEX, EXX単独では最短経路を探索することが困難なこと、EAX単独で最短経路を探索できることがわかった。EAX単独では局所的に最適となっている経路を形成するまでに時間がかかる。本実験により、ECXOはEAXの最適解探索時間を10% ~ 30%向上できること、ACOやEXから、必要があればEXXを経由して、EAXに交代する最適な交代時期を引き継ぐ遺伝子群の「多様性」と最適解への「収束度」状況を観測してECXOは制御できること等を検証した。

尚、att532, rat575, u574, rat783等より複雑な問題に対しては繰り返しECXO方式が有効である。これについては別の機会に報告する。

文 献

- (1) M. Dorigo and T. Stutzle: "Ant Colony Optimization", The MIT Press (2004)
- (2) J. H. Holland: "Adaptation in Natural

- and Artificial Systems”, MIT Press (1992)
- (3) D. E. Goldberg: “Genetic Algorithms in Search, Optimization, and Machine Learning”, Addison-Wesley Publishing Company, Inc. (1989)
 - (4) N. Sannomiya, H. Kita, H. Tamaoki, T. Iwamoto: “Genetic Algorithms and Optimization”, Asakura-Shoten, 1998 (in Japanese) . 三宮信夫, 喜多 一, 玉置 久, 岩本貴司:「遺伝的アルゴリズムと最適化」, pp.37-51, 朝倉書店 (1998)
 - (5) J. Grefenstette, R. Gopal, B. Rosmaita, and 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)
 - (6) 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)
 - (7) L. Davis: “Applying Adaptive Algorithms to Epistatic Domains”, Proc. of 9th Int. Joint Conf. on Artificial Intelligence, pp. 162-164 (1985)
 - (8) 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)
 - (9) 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)
 - (10) 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)
 - (11) Y. Nagata and S. Kobayashi: “The proposal and evaluation of a crossover for Traveling Salesman Problem: Edge Assembly Crossover”, JSAI, Vol.14, No.5, pp.848-859, 1999 (in Japanese) . 永田裕一, 小林重信:「巡回セールスマン問題に対する交叉：枝組み立て交叉の提案と評価」, 人工知能学会誌, Vol.14, No.5, pp.848-859 (1999)
 - (12) M.-K.Tsai, J.-M.Yang, Y.-F.Tsai, C.-Y. Kao: “Some issues of designing genetic algorithms for traveling salesman problem”, Soft Computing, No.8, pp.689-697 (2004) .
 - (13) K. Helsgaun: “An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristics”, European Journal of Operational Research, 126 (1), pp. 106 - 130 (2000)
 - (14) S. Lin & B. W. Kernighan: “An Effective Heuristic Algorithm for the Traveling-Salesman Problem”, Oper. Res. 21, pp.498-516 (1973)
 - (15) E. Aarts and J. K. Lenstra: “Local Search in Combinatorial Optimization”, Princeton University Press (2003)
 - (16) M. Dorigo and L. M. Gambardella: “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”, IEEE Transactions on Evolutionary Computation, 1 (1), pp.

- 53 - 66 (1997)
- (17) M.-K.Tsai, J.-M.Yang, Y.-F.Tsai, C.-Y. Kao: "An Evolutionary Algorithm for Large Traveling Salesman Problems", IEEE Transactions on Systems, Man, and Cybernetics- Part B. Cybernetics., Vol. 34, No.4, pp. 1718 - 1729 (2004)
- (18) D. Applegate, R. Bixby, V. Chvatal, W. Cook: "Finding Tours in the TSP", Tech. Rep. TR99-05, Dept. Computat. Appl. Math., Rice Univ., Houston, TX 77005 (1999)
- (19) O. Martin, S. W. Otto, E. W. Felten: "Large-Step Markov Chains for the TSP Incorporating Local Search Heuristics", Operation Research Letters, vol. 11, pp.219-224 (1992) .
- (20) B. Freisleben, P. Merz: "New genetic local search operators for the traveling salesman problem", Proc. of Parallel Problem Solving from Nature IV (PPSN IV), pp. 890-899 (1996)
- (21) D.S. Johnson, L. A. McGeoch: "The Traveling Salesman Problem: A Case Study in Local Optimization", E. H. L.Aarts, J. K. Lenstra, eds., Local Search in Combinatorial Optimization, John Wiley and Sons, Ltd., pp. 215-310 (1997)
- (22) R. Baraglia, J. I. Hidalgo, R. Perego: "A Hybrid Heuristic for the Traveling Salesman Problem", IEEE Transactions on Evolutionary Computation, Vol.5, No. 6, pp. 613 - 622 (2001) .
- (23) M. Garey and D. Johnson: "Computers and Intractability, A Guide to the Theory of NP-Completeness", W. H Freeman and Company (1979)
- (24) S. J. Russell and P. Norvig: "Artificial Intelligence - A Modern Approach", Prentice Hall, Upper Saddle River, New Jersey (1995)
- (25) B. W. Kernighan and D.M. Richie: "The C Programming Language, Second Edition", Bell Telephone Laboratories (1988)
- (26) H. Hirano: "A Genetic Algorithm C programming", Personal Media Inc., pp. 232-238, 1995 (in Japanese) . 平野 広美:「遺伝的アルゴリズム C プログラミング」, pp.232-pp.238, パーソナルメディア社 (1995) .
- (27) R. L. Haupt and S. E. Haupt: "Practical Genetic Algorithms", John Wiley & Sons, Inc. (2004)
- (28) S. M. Sait, H. Youssef: "Iterative Computer Algorithms with Applications in Engineering", (translated into Japanese by Y. Shiraishi), Maruzen Co., Ltd. (2002) .
- (29) R. Takahashi: "Solving the Traveling Salesman Problem through Genetic Algorithms with Changing Crossover Operators", Proceedings of Fourth International Conference on Machine Learning and Applications, IEEE Computer Society, pp. 319 - 324 (2005)
- (30) R. Takahashi and K. Degai: "A Performance Improvement of Genetic Algorithms through Changing Crossover Operators to Solve the Traveling Salesman Problem", Proc. of the 8th International Conference on Computer and Information Technology, organized by Islamic University of Technology, pp. 40 - 45 (2005)
- (31) 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, Proc. of the 2nd International
Conference on Natural Computation
and the 3rd International Conference
on Fuzzy Systems and Knowledge
Discovery, Xidian University, pp. 114 -
130 (2006)
- (32) TSPLIB95: <http://www.informatik.uni-heidelberg.de/>
- (33) 高橋良英、「拡張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」、電気学会, Vol.128, No.12, Sec. C.,
pp.1820-1832, 2008.