

遺伝的アルゴリズムと蟻協調行動モデルによる 巡回セールスマン問題の解法

高橋 良英*

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

Ryouei TAKAHASHI*

Abstract

In order to improve performance of solving the traveling salesman problem (TSP), we evaluate functionality and performance of both of ACO (Ant Colony Optimization) and CXO (Changing Crossover Operators). Our CXO substitutes the improved EX for SXX and it selects the local optimum sub-paths randomly on a pair of selected parents with higher fitness to create a global optimum solution. In ACO, ants probabilistically select next visiting cities based both on the quantity of pheromone trail deposited on all the edges between two cities and on the lengths of them. Our C experiments using the famous Eilon's 75 city problem show that CXO as well as ACO can provide us with the same optimum solution whose cyclic path's length is 542.31 and that CXO is superior to ACO in point of computer execution time on condition that an optimum seed_id with which the initial population is generated is selected.

Keywords: TSP, ACO, GAs

1. はじめに

巡回セールスマン問題 [4], [13] 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 法) が有効であることが知られている [1] [2]。ACO 法は, 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, その巡回路上の各都市間の経路上に記憶される。論文[2]によれば, ACO 法における Eilon の 75 都市問題 [5] の最適な実数解は 542.31 であること, GA 手法での最適な整数解は 545 である示されているが, 遺伝的アルゴリズム (GA) 手法との詳細な性能分析結果

については報告されていない。

一方, 我々は巡回セールスマン問題の最適解の近似を効率的に得る遺伝的アルゴリズム [3], [4], [12] の新しい手法として任意の時点で遺伝子交叉オペレータを交代可能な遺伝子交叉オペレータ交代法 (CXO 法) を提案し, その手法の有効性を, 200 都市問題 [15], Eilon の 75 都市問題 [14], Oliver の 30 都市問題 [6], [16] のデータを用いた C プログラム実験で検証した。CXO 法では比較的初期の世代では, 改良 EX 法 [5] 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選択して局所的に最適解を生成し, 後期の世代では改良 SXX 法 [7] 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解を生成する。特に, 論文 [14] の実験では, Eilon の 75 問題を対象として検証

平成 19 年 1 月 5 日受理

* システム情報工学科・教授

し、乱数種=1の初期経路条件もと、最適解544.81の解を得ている。CXO法もACO法の解は初期条件(乱数種)に依存することがわかっている。

このため、本論文では、巡回セールスマン問題(TSP)を効率的に解く遺伝子交叉オペレータ交代法(CXO法)と蟻協調行動モデル(ACO法)の機能と性能について実験的に比較評価することとした。Eilonの75都市問題に対するCプログラム実験の結果、乱数種=176のもと、(1)CXO法ではACO法ともいずれも同一の最適解(経路長=542.31)を探索可能なこと、(2)最適解を探索するのに、ACO法では64世代、292秒かかること、最適世代交代番号探索のもとではCXO法では68世代、33秒かかること(1世代に1,000ツアー)、(3)ACOとGAsの結合による更なる性能改善施策の基本検討結果等その詳細を報告する。

2. 巡回セールスマン問題

あるセールスマンがN個の都市を1回ずつ巡回して、出発都市に戻ってくる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という。本検討では対称型巡回セールスマン問題を研究対象とする。ここでは、セールスマンが巡回する都市が2次元ユークリッド空間に位置し、どの2都市間も上りと下りが同一距離のたった1つの道で結ばれている。逆順を同一経路と見なして全経路数は $(N-1)!/2$ 通りと計算できる。 $N!$ はStirlingの公式により、 $\sqrt{2\pi} \exp(-N) * N^{(N+1/2)}$ と近似されるが、この式は経路の選択に指数オーダー以上の時間がかかることを示している。TSPは、都市数Nが大きくなると、解くアルゴリズムはあるものの、実効上計算機でもその解を解くことが困難なNP完全クラスに属する問題であることが知られている [8]。

3. CXO法とACO法

3.1 CXO法 (CXO: Changing Crossover Operators)

(1) 着眼点

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。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叉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る。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。

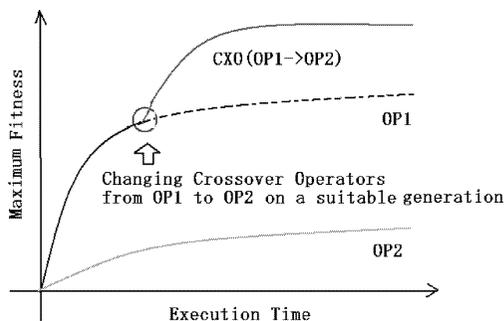


図1. 遺伝子交叉オペレータ交代法の概念図
Figure 1. The concept of Changing Crossover Operators (CXO).

遺伝子交叉オペレータ交代法の概念を図1に示す。

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。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 オードかかっていた検索時間を、都市数 n^1 のオードに削減している。図2はSXX法概念を示している。図中、a, b, c, dは都市を示している。

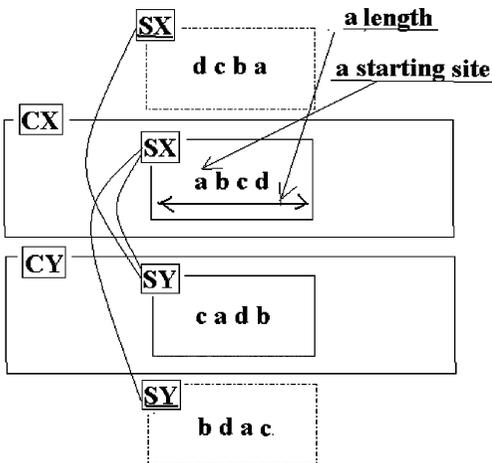


図2. SXX法概念図
Figure 2. In SXX sub-tour SX in the parent tour CX is exchanged with SY or its reverse order sub-tour \overline{SY} in the parent CY .

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」[9], [13]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。

図3は改良EX法概念を示している。

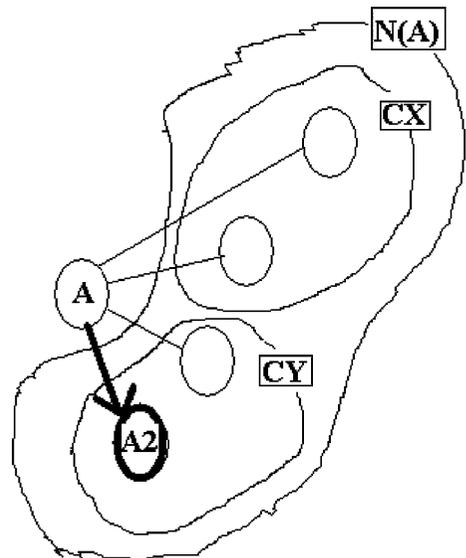


図3. 改良EX法概念図
Figure 3. Improved EX selects the next city $A2$ that the first child $C1$ visits out of $N(A)$ such that the distance between A and $A2$ is shortest.

(2) CXO 法のアルゴリズム

(A) 遺伝子型と表現型と適応率

巡回セールスマン問題における、「個体」の表現法と「適応度」の計算方法を以下に整理する。

(i) 個体の遺伝子型と表現型

個体は巡回セールスマン問題の解候補となる巡回路である。

●**遺伝子型**：各遺伝子は 32 バイトで構成され、訪れる都市の番号 (1 以上の整数) をその値として持つ。訪れる順番に都市番号が遺伝子に付与されている。

●**表現型**：個体は都市を巡回する経路を示す。巡回路は訪問する順番で並べられた都市列で表現される。

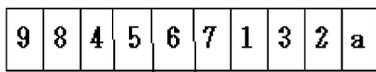
図 4 の上部で示される遺伝子型は“1”から“a”で表現される 10 個の都市の場合の遺伝子型を示し [4]、その表現型は図 4 の下部に示される各都市を巡回する巡回路となっている。

(ii) 個体の適応度：

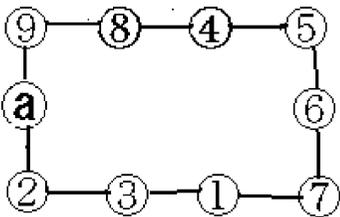
個体の適応度は、巡回路 α の相対的経路長の逆数を示す式であり、以下で定義される。

巡回路 α の適応度

= 各都市間の距離の総和 γ / 経路 α の長さ β



A genotype of a solution of a TSP



A phenotype of the above solution of a TSP

図 4. 巡回セールスマン問題の解の遺伝子型と表現型

Figure 4. Genotype and phenotype of a solution of a TSP.

分子は都市と都市を結ぶ全ての辺の長さの総和 γ であり、巡回する都市の位置関係のみで決まる選択する経路に依存しない固定パラメータである。また、分母は選択した経路の経路長 β である。経路長 β は経路 α を構成する各都市 X_i と X_j 間の距離の総和である。適応度は、全距離 γ が経路長 β の何倍なのかを示しており、経路長が短い程、適応度が大きくなる。

(B) CXO 法のアルゴリズムの概要

(a) 初期化処理

(i) 遺伝子交叉オペレータの最適な交代時期と最適な乱数種の決定

1 から 1,000 の乱数種の各乱数種について以下の手続きを実行し、最も短い経路を探索する最適な乱数種と最適な世代交代時期を探索する。

- 遺伝子交叉確率、人口数、観測世代数の設定
- 7 つの遺伝子交叉オペレータの中から遺伝子交叉オペレータ 1 の選択 [14] (デフォルト：改良 EX)
- 遺伝子交叉オペレータ 2 の選択 (デフォルト：改良 SXX)
- 1 世代目から遺伝子交叉オペレータ 1 が最短経路を探索した最適世代までの間の全ての世代を遺伝子交叉オペレータ 1 から遺伝子交叉オペレータ 2 への交代世代候補と考え、その中から最も短い経路を探索した世代を最適な交代世代時期と決定する。

(ii) 上記 CXO により初期人口を生成する。

- 遺伝子交叉オペレータ 1 を遺伝子交叉オペレータとして選択する。
- 上記最適乱数種で人口を初期化し各個体の適応度を計算する。
- 世代番号 = 0 とする。

(b) 本体の実行

(i) 遺伝子交叉オペレータを変更するか否かの判断

(ii) ルーレット法で選択した両親の遺伝子を交叉させ個体を生成する。

(iii) 各世代の終了時、各個体の適応度を計算しそれまでの最大適応度モデルを求める。

(C) 終了処理

(i) 世代が最終観測世代数を越えたら処理を完了する。そうでなければ (b) に戻る。

3.2 ACO 法 (Ant Colony Optimization)

(1) 着眼点

巡回セールスマン問題に適用される ACO 法においては、各蟻が都市一周旅行で学習した「巡回路長」情報を巡回した回路上の各都市間の経路にフェロモンとして残す。後続する蟻は、都市間の「距離の短さ」と「フェロモン量」から次に訪れる都市を確率的に選択する。過去のフェロモン量は試行の度に少なくなるが、新しく一周して学んだ「巡回路長」情報は更新「フェロモン量」として後続の蟻に伝達する。初期状態でフェロモン量が等しく、従って等確率で選択されていた複数の選択経路は、巡回が積み重ねられる度に、巡回路長が短くなる経路が選択され、最終的には大域的に最適な経路上のフェロモン量が最も多くなるように増えていく。

(2) ACO を実現するアルゴリズム

ACO における個々の蟻と、GAs の個体を同一視することができるので、ACO と GAs のソフトウェア構造は酷似している。このため、ACO を実現する C プログラムは GAs を実現する我々が開発した C プログラム [10], [11] を修正して実現している。

本研究では ACO を以下のアルゴリズムで実現している。

<ACO の起動パラメータ>

- ① 蟻が最初に訪れる都市を決定する乱数種: seed_id (デフォルト=1)
- ② フェロモン量の忘却度 ρ : 群単位で旅をする m 匹の蟻が、旅を完了する度に更新するフェロモン量のうち過去に蓄積し

たフェロモン量の忘却度が ρ である。これによって長い経路長の探索という過去の経路選択について忘れることができる。 $0 \leq \rho \leq 1$ である (デフォルト=0.2)。

- ③ ある世代における蟻の数 n : 各都市に配置される蟻の数の総数。
- ④ フェロモン量の更新タイミング m : m 匹の蟻が群単位で旅を完了する度にフェロモン量を更新する時、群を構成する蟻の数 m のこと。蟻は、直前の蟻群の旅が完了するまでに残したフェロモン量の関数から成る確率に従って次に訪れる都市を選択する。一般に $m \leq n$ であり、ある世代の n 匹の蟻は、 $\lceil n/m \rceil$ 回の群数で旅を完了することができる。ここで $\lceil \alpha \rceil$ は実数 α を越える最小の整数のことで、例えば $\lceil 3 \rceil = 3$, $\lceil 3.5 \rceil = 4$ である。(デフォルト $m=10$)
- ⑤ 観測世代数 T : T 世代後に処理を停止する。
- ⑥ 蟻が時間 $t=t+1$ の時に都市 i にいる時、次に都市 j を訪れる確率 $P_{ij}^{k,i}(t+1)$ を決定する際に考慮する距離 d_{ij} の重み: β (デフォルト値=2)

<ACO 法を実現のアルゴリズム>

- (A) 世代 gen=0 における動作 ($t=0$ における)
 - (A-1) ACO の起動パラメータの決定
 - (A-2) 最初に訪れる都市の初期化と初期のフェロモン量の決定
- ① 世代を構成する n 匹の蟻 A_k のそれぞれについて、初期の巡回路 C_k ($k=1, 2, \dots, n-1, n$) を決定する。この時、最初に訪れる都市 T_k を記憶する。初期化処理では、次に訪れる都市はランダムに決定する。
- ② 都市 i から都市 j へのフェロモン量 τ_{ij} の初期値 $\tau_{ij}(0)$ は、① で求めた初期の巡回路 C_k の平均経路長とする。すなわ

ち、以下の式で求める。

$$\tau_{ij}(0) = \frac{\sum_{k=1}^n Q(k)}{n}$$

ここで、 $Q(k)$ は、(巡回路 C_k の長さの逆数) * (二都市間の距離の総和) であり、遺伝的アルゴリズムにおける適応度に一致する。

(B) ある世代 $\text{gen} = \alpha$ における $\text{time} = t + 1$ 時の蟻の動作

(B-1) n 匹の蟻 A_k は都市 T_k から始めて、時間 $\text{time} = t + 1$ に都市 i にいるとしよう。蟻 A_k は次に都市 j を、都市 i から都市 j への 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} から決定される確率 $p_{ij}^k(t+1)$ で選択する。ここで蟻 A_k はこれまで q 都市のうち1個の都市について訪問が終わっているものとし、 $q-1$ 個の未訪問の都市から次に訪れる都市を決定するものとする。

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

(B-2) フェロモン量の更新

以下のローカルタイミングとグローバルなタイミングで、フェロモン量を更新する。

(B-2-1) ローカルフェロモン更新 $\Delta\tau_{ij}(t+1)$

時間 $t = t + 2$ で旅をする蟻が参照するフェロモン量 $\tau_{ij}(t+1)$ は、現在 $t = t + 1$ で参照しているフェロモン量 $\tau_{ij}(t)$ と $t = t + 1$ の旅で更新したローカルフェロモン更新量 $\Delta\tau_{ij}(t+1)$ から、 m 匹の蟻が旅を完了する度に、以下の式で更新する。時間 t は m 匹の蟻が旅を完了する度に更新される。

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

ここで、 ρ はフェロモンの蒸発度である。 $Q_{ij}(k)$ 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、巡回路長の逆数 * 二都市間の総距離 (適応度) であり、都市 i から j を経由しない場合は 0」である。

(B-2-2) グローバルフェロモン更新 $\Delta_1\tau_{ij}(t+1)$

グローバル更新では、時間 $t = t + 1$ 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、その巡回路の各辺に加える。 $\Delta_1\tau_{ij}(0) = 0$ であり、世代を更新する度にリセットする。

$$\Delta_1\tau_{ij}(t+1) = \max_{1 \leq k \leq (t+1), m \leq n} Q_{ij}(k),$$

$$\tau_{ij}(t+1) = (1-\rho)^* \tau_{ij}(t) + \rho^* \Delta_1\tau_{ij}(t+1)$$

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り、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。その機能を有する 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 by pheromone) を有する ACO と呼ぶ。 α が固定的に 1 の場合は SA 手法を有しない ACO と呼ぶ。

(B-2-3) ある世代でまだ m 匹未満の蟻が旅を完了していないなら、(B-2-1) のフェロモンローカル更新を行う。

(C) 最終世代でなければ、(B)に戻る。この時フェロモン量は世代を越えて引き継ぐ。

留意事項: 蟻は旅を終る度に 2-OPT 法と 3-OPT 法を適用して旅行ルートの仮更新を行い距離の短縮度を測定し、その値が大きければ旅行ルートの確率的本更新を行う。フェロモン更新回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 (SA by distance) と呼ぶ。

4. 実験結果

4.1 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。図中の線は探索した最小経路長=542.31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。

<実験データ>

最適解を求める性能は実験データの並び順にも依存する。データの並びはCコードで以下のように定義されている。

```
static struct {double x, y;} cityxy
[CityNo]=
{{48,21}, {52,26}, {50,30}, {55,34}, {54,38}, {50,
40}, {51,42}, {55,45}, {55,50}, {50,50}, {41,46},
{45,42}, {45,35}, {40,37}, {38,33}, {33,34}, {29,
39}, {33,44}, {35,51}, {30,50}, {22,53}, {21,48},
{21,45}, {21,36}, {20,30}, {26,29}, {22,22}, {27,
24}, {30,20}, {35,16}, {36,6}, {26,13}, {15,5}, {15,
14}, {16,19}, {12,17}, {6,25}, {11,28}, {12,38},
{7,43}, {9,56}, {15,56}, {10,70}, {17,64}, {26,59},
{30,60}, {31,76}, {40,66}, {35,60}, {40,60}, {47,
66}, {50,70}, {55,65}, {57,72}, {70,64}, {62,57},
{55,57}, {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}, {40,20}, {36,
26}, {43,26}};
```

4.2 最適解

CXOは最小巡回路長=542.31を乱数種176のもとで発見した。ACOは最小巡回路長=542.31の同一の解を同じ乱数種176のもとで探索できた。探索した最適巡回路を図5に示す。図中の都市番号は、4.1で定義された都市番号を示す。例えば都市番号=11はその座標が(41,46)である。最適巡回路の都市番号列は以下の通りである。

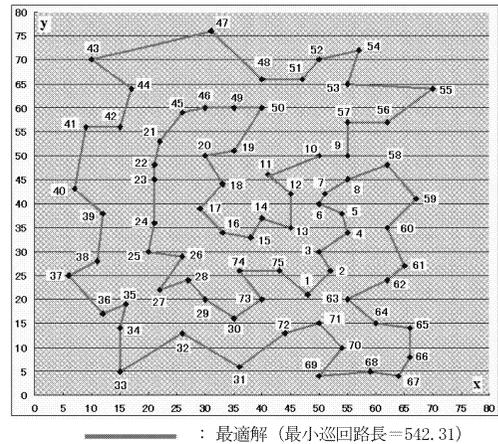


図5. CXO法、ACO法が見つけた最適解
Figure 5. The optimum solution found by CXO and ACO.

(11, 12, 13, 14, 15, 16, 17, 18, 20, 19, 50, 49, 46, 45, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 73, 74, 75, 1, 2, 3, 4, 5, 6, 7, 8, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 44, 43, 47, 48, 51, 52, 54, 53, 55, 56, 57, 9, 10).

4.3 ACO法の起動パラメータと走行結果

(1) ACOの起動パラメータ

- 蟻が最初に訪れる都市を決定する乱数種： seed_id=176。Seed-idはCXO法に合わせ176とする。
- フェロモン量の忘却度 $\rho=0.3$
- 蟻の数（ある世代を構成する蟻の数：最初に訪れる都市を決定するパラメータ） $n=75$
- local フェロモン量の更新のタイミング $m=10$ ：
- 確率 $p_{ij}^k(t+1)$ を決定する際に考慮する距離 d_{ij} の重み： $\beta=2$

(2) 走行結果

我々のACOはエリート蟻システムの改良版である。乱数種=176の下、全ての都市を開始都市としたツアー探索により平均巡回路長（各蟻の初期フェロモン量）を決定する。フェロモン

量 $\tau_{ij}(t+1)$ を 10 匹 ($=m$) の蟻のフェロモン量の総和 $\Delta\tau_{ij}(t+1)$ で local に更新する。グローバルなフェロモン量 $\Delta_1\tau_{ij}(t+1)$ も観察する。その時点までに最適であった蟻のグローバルなフェロモン量 $\Delta_1\tau_{ij}(t+1)$ の α 倍がローカルな上記フェロモン量を更新する時点で参照されその値が $\Delta\tau_{ij}(t+1)$ に加算される。 α は、0 から $5=(m/2)$ の値をとるように時間 t により、周期的に変動させている。 α はシミュレーションでアニーリングにおける温度の役割を果たしており、周期的にフェロモン量の重みを変更するパラメータである。グローバルなフェロモン量 $\Delta_1\tau_{ij}(t+1)$ は 75 匹の蟻が旅程を完了する世代の完了時点でリセットされる。 $\Delta\tau_{ij}(t+1)$ は 10 匹の蟻が旅程を完了した度にリセットされる。忘却率=0.3 でフェロモン量を更新する。実験の結果、SA ありの ACO は、855 世代目の終わり (75×855 匹の蟻の旅の後)、292 秒のコンピュータ時間で最適解 (経路長=542.31) を探索できた。最適解に収束する過程についての詳細については 4.5 節で述べる。

4.4 CXO 法の起動パラメータと走行結果

(1) 起動パラメータ

以下のパラメータのうち最適解を探索する最適な乱数種の決定方法が大きな課題と考える。

- 乱数種 (初期集団の規定パラメータ) ・ ・ 176
- 人口数 ・ ・ 1,000
- 遺伝子交叉オペレータ交代時期 ・ ・ 5
- 世代交替方式 ・ ・ 一括型世代交代方式
- 2 オプト法 ・ ・ 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逆順にする。
- 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

(2) 走行結果

乱数種=176 の初期条件のもと、improved EX から SXX 法に 5 世代目に交代する CXO

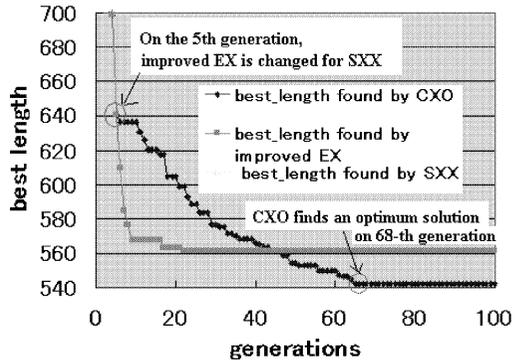


図6. CXO 法による巡回路長の改善
Figure 6. Improvement of best lengths through changing crossover operators.

法では 68 世代目、35 秒後に最適解 (経路長=542.31) に到達した。図6は5世代目に improved EX から SXX 法に交代する CXO 法が improved-EX や SXX 法単独よりも、より短な

表1 最適巡回路長への収束速度の観点からの CXO 法, 改良 EX 法, SXX 法の比較
Table 1 Comparison of CXO with both of improved EX and SXX from the point of view of best lengths' convergence.

generations	best length found by CXO	best length found by improved EX	best length found by SXX
0	2156.769	2156.769	2156.769
5*	640.8796	640.8796	2031.738
10	636.3507	567.4242	1871.399
20	604.477	562.9991	1676.694
30	576.4049	561.0988	1512.557
40	566.0646	561.0988	1371.964
50	554.1085	561.0988	1285.224
60	549.5626	561.0988	1180.07
65	542.3314	561.0988	1152.198
68**	542.3094	561.0988	1124.848
70	542.3094	561.0988	1100.916
80	542.3094	561.0988	1031.582
90	542.3094	561.0988	969.1414
100	542.3094	561.0988	912.4564

5*: 改良 EX 法から SXX 法への交代世代番号
68**: CXO 法により最適解を探索した世代番号

最短経路をより短い時間で探索できたことを示している。表1はその詳細であり、ある世代までに CXO 法, improved EX, SXX 法が探索できた局所最適巡回路長を示している。表1では, CXO が巡回路長=542.31 の最適解を探索した 68 世代目に improved-EX は 561.10 の局所最適解しか探索できていないこと, オペレータ交代後改良 EX 法単独では 30 世代目以降, 探索する局所最適解が安定状態になっていること, SXX 法も 1124.85 の局所解しか探索できてい

ないことを示している。1 世代は 1,000 回の旅からなるので, CXO 法では最適解を探索するのに, 68,000 回の旅を要している。

CXO 法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がどのように変化するかを図7と表2に示している。この図表は, CXO 法で最適交代世代番号は5でありそれ以外の交代世代では最適解は見つからないこと, オペレータ交代時期が5世代目以降の場合 CXO 法が探索した局所最適解の経路長は improved-EX より短いことを示している。表2から CXO 法では, 各世代交代方式で 100 世代 100,000 回の旅を観察したとすれば, 最適な交代世代を決定するのに約 178 秒かかることがわかる。図7は, 各局所最適解を探索するのに必要な正規化コンピュータ実行時間 (コンピュータ実行時間+最小巡回路長 (=543.21)) も示している。

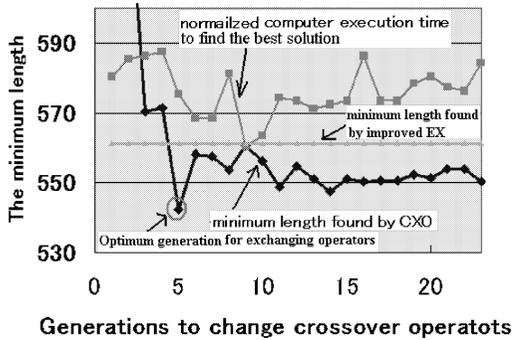


図7. 最適な遺伝子交叉オペレータ交代世代番号の探索

Figure 7. To find an optimum generation to change crossover operators.

表2 最適な遺伝子交叉オペレータ交代世代番号の探索

Table 2 To select an optimum generation for changing operators

generations for changing crossover operators	minimum length	computer execution time to find the best model (seconds)	computer execution time to observe ten thousand tours (seconds)
1	805.28	38	38
2	636.73	42	43
3	570.39	43	47
4	571.39	44	50
5*	542.31	33	48
6	558.12	31	71
7	557.44	25	63
8	553.47	27	55
9	560.55	18	55
10	556.25	31	68

5*: 最適なオペレータ交代世代番号

4.5 ACO 法と CXO 法の最適解への収束効率の比較

(1) 最適解を探索するために必要な旅の数の観点からの比較

表3は CXO と ACO が最適解を見つけるまでに何回の旅 (ACO 法では旅をした蟻の数, CXO 法では生成した個体の数) を必要としたかを示している。1,000 * n 回 (n=0,1,...,100) の旅毎にそれまでに探した最短経路長を示している。この表から 68 世代目 68,000 回の旅の後に CXO 法は最短経路長 542.31 を探索していること, シミュレーテッドアニーリング (SA) 機能を有する ACO 法では 64 世代目 64,125 (=75×855) 回の旅の後に同じ最短経路長 542.31 を探索していること, SA 手法を有しない ACO 法では, 47 世代目 47,175 (=75×629) 回の旅の後に経路長 556.8 の局所最適解を探索していることがわかる。

表3をグラフ化し図8で示す。

このことから最適解を探索するのに要する旅の数の観点からは, SA 機能を有する ACO 法は

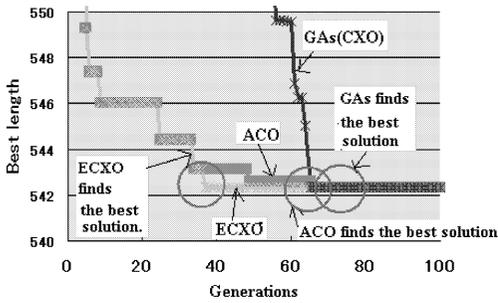


図8. GA手法とACO法の最適解を探索するまでに必要なツアー数の比較
 Figure 8. A number of tours required for both of GAs and ACO to obtain the optimum solution.

GA手法 (CXO法) を凌駕することがわかる。

(2) コンピュータ実行時間の評価

表4はCXO法とACO法 (SA機能あり), ACO法 (SA機能なし) が最適解を見つけるまでにどの位のコンピュータ実行時間を必要としたかを示している。この表からCXO法は5世代目5秒後に improved EX法からSXXに交代し68世代目33秒後に最短経路長542.31を探索していること、この時間にACO法 (SA機能あり) では547.38の経路長, ACO法 (SA機能なし) では567.85の経路長しか探索できていないことがわかる。292秒後にACO法 (SA機能あり) は同じ最短経路長542.31を探索しているが, ACO法 (SA機能なし) は211秒後局所最適解 (経路長=556.80) を探索したのみである。表4をグラフ化し図9で示す。表4, 図9から最適乱数種選択, 最適世代交代時期の選択のもとではACO法よりCXO法の方が収束時間がよいことがわかる。しかし, CXO法で最適交代世代番号=4の処理が完了するまでに178秒かかっており (表2参照のこと) 5世代目に improved EX法からSXXに交代させるCXO法で探索した33秒に加えると211秒となる。

表3 最適解を探索するまでに必要なツアー数の観点からのGAs (CXO) 手法とACO法の比較

Table 3 Comparison of GAs with ACO from a viewpoint of a number of tours required to find best solutions.

number of ant's tours (×1000)	ACO with SA	ACO without SA	CXO	ECXO	SXX
0	2259.99	2259.99	2156.77	2259.99	2156.77
5	549.28	569.22	640.88	549.28	2031.74
10	546.03	567.85	636.35	546.03	1871.40
20	546.03	560.19	604.48	546.03	1676.69
30	544.43	557.61	576.40	544.43	1512.56
39*	543.14	557.61	568.06	542.31	1374.96
50	542.61	556.80	554.11	542.31	1285.22
60	542.61	556.80	549.56	542.31	1180.07
64*	542.31	556.80	545.03	542.31	1152.20
68*	542.31	556.80	542.31	542.31	1124.85
80	542.31	556.80	542.31	542.31	1031.58
90	542.31	556.80	542.31	542.31	969.14
100	542.31	556.80	542.31	542.31	912.46

<注意>表は完了した旅行数で探索できた巡回経路長を示す。

39*: SA (シミュレーテッドアニーリング) 機能を有するACO法がGAsに451世代目に交代して最小回路長を探索するまでに要したツアー数

64*: SA (シミュレーテッドアニーリング) 機能を有するACO法が最小回路長を探索するまでに要したツアー数

68*: GAs (CXO) 法にて大域的な最小巡回経路長を探索するまでに要したツアー数

4.6 考察

(1) 機能比較

ACO法とCXO法は, 本実験では同一の最適解を探索したが, 最適解を探索する方法として, 以下の機能の特徴の違いがある。CXO法がある時点において空間上に複数存在している経路と経路長 (適応度) に関する情報から経路長の更に短い (適応率の高い) 巡回回路を確率的に生成する方法であるのに対して, ACO法は接する各二都市間にフェロモン情報として残された学習

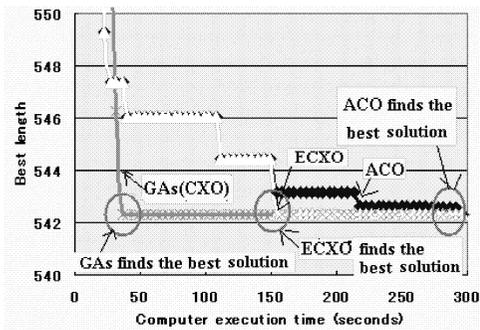


図9. GAsとACOのコンピュータ実行時間の評価

Figure 9. A computer execution time evaluation of both of GAs with CXO and ACO.

表4 コンピュータ実行時間の評価
Table 4 Evaluation of computer execution time

time (seconds)	ACO with SA	ACO without SA	CXO	ECXO	SXX
0	2259.99	2259.99	2156.77	2259.99	2156.77
5	277.67	569.22	640.88	577.67	2031.74
33*	547.38	567.85	542.31	547.38	940.69
67	546.03	560.19	542.31	546.03	619.74
99	546.03	557.61	542.31	546.03	575.25
131	544.43	557.61	542.31	544.43	575.25
155*	543.14	557.61	542.31	542.31	575.25
181	543.14	557.61	542.31	542.31	575.25
211	543.14	556.80	542.31	542.31	575.25
240	542.61	556.80	542.31	542.31	575.25
273	542.59	556.80	542.31	542.31	575.25
292*	542.31	556.80	542.31	542.31	575.25
301	542.31	556.80	542.31	542.31	575.25

33*: CXO法が最適解を見つけるのに要したコンピュータ実行時間

155*: ACO法 (SA機能あり) からGAsに交代して最適解を見つけるために要したコンピュータ実行時間

292*: ACO法 (SA機能あり) が最適解を見つけるために要したコンピュータ実行時間

情報、それはその都市間を過去に巡回した蟻が残した巡回路長データなのであるが、その過去の学習情報から次に訪れる都市を確率的に選択する方法であることである。

① ACO法が隣接する都市間の距離の短さに着目して次に訪れる都市を確率的に決定する点は、CXO法が初期世代において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選択することと考え方が同じである。

② ACO法が「この二都市間の経路を通れば、経路長の短い経路を選択できる」というフェロモン量として表現された過去の学習情報に基づき次に訪れる都市を確率的に選択する方法であるのに対して、CXO法が局所的に最適となっている部分巡回路を確率的に選択・結合することによってより短い巡回路長を探索する方法である。

(2) 性能比較

シミュレーテッドアニーリング機能を有さないACO法では最適解を探索できず、局所最適解(経路長556.8)を47,000ツアー後211秒後に見出すのみである(表5参照)。シミュレーテッドアニーリング機能を有するACO法は最適解(経路長542.31)を64,000ツアー後292秒後に探索している。178秒後に最適交代世代探索後、CXO法は68,000ツアー後、33秒後に最適解を探索している。最適交代世代探索時間を考慮してもCXO法はシミュレーテッドアニーリング機能を有するACO法より性能は良い。最適な交代世代選択のもとではCXO法が最も性能が良い。その理由は、CXO法では次に選択する都市は二都市以上の都市間のつながりを考慮し選択できるが、ACO法では二都市間のフェロモン量のみに着目して次の都市を選択しているので、最適経路上の都市間のフェロモン量を多く残すためには多くの蟻の学習が必要なためである。CXOの問題点は最適解に遭遇する確率が低いことである。

5. ECXO 法による最短経路長の探索効率向上

ACO 法と GAs を結合することで更に経路長の短い経路を探索する ECXO 法を検討した。

本方式は CXO 法において交代可能な遺伝子交叉オペレータの1つとして ACO 法を可能とする。この方法を拡張遺伝子オペレータ交代法 ECXO 法と呼ぶ。この方法では比較的初期の世代では ACO 法により局所的に最適な解を生成し、その解が経路長の短い経路を探索して安定状態に突入した後期の任意の世代に SXX 法に

交代して更に広域的に最適な解を生成する。ACO 法や CXO 法の解は一般に初期値に依存する。ある任意の初期値で生成された ACO の初期ツアー群について、拡張遺伝子交代法 ECXO は ACO が見つけた最適解を最小限保証する。

実際、seed-id=176 の時 ACO 法も CXO 法も単独で最適解(経路長=542.31)に到達できたが、ACO 法が早期の世代(451 世代=(75×451 匹の蟻の旅後)で SXX 法に交代することで ACO が見つけた時間より(292 秒)更に短い時間(155 秒)で最適解を見つけることができた。

表5 SA 機能による ACO の性能改善詳細
Table 5 Detailed performance-evaluation of ACO

NO	methods	最短経路長	最適解を探索するための旅の数(×75)	探索時間(秒)	
1	ACO (SA 機能あり)	distance_all+pheromone	542.31	855	292
2		pheromone	584.83	684	231
3		distance_all	561.34	894	302
4		distance_with_2-opt	558.37	872	293
5		distance_with_3-opt	558.08	160	54

表6 拡張遺伝子オペレータ交代方式の有効性の評価(乱数種=176 の場合)
Table 6 A Validation of extended changing crossover operators (seed_id=176)

NO	methods	best length found	required number of tours to find the best length (×1000)	required computer execution time to find the best length (seconds)	population size
1	ECXO* (ACO with SA → SXX)	542.31	39	155	75
2	ACO with SA	542.31	64	292	75
3	CXO** (improvr EX-→SXX)	542.31	68	33	1.000
4	ACO without SA	556.8	47	211	75
5	improved EX	561.1	22	22	1.000
6	SXX	585.34	198	81	1.000

表7 ACO (SA 機能あり) から GAs (SXX) への最適な交代世代番号の探索
Table 7 Finding out an optimum generation for exchanging ACO for SXX

NO	generations for exchanging ACO for SXX	best length ACO finds	best length improved by ECXO	computer execution time to find the best solution (seconds)		
				execution time for ACO	execution time for SXX	sum
1	1	737.06	560.61	1	18	19
2	72	547.4	546.83	25	3	28
3	113	546.03	544.65	38	5	43
4	327	544.43	543.05	117	4	121
5*	451	543.14	542.31	153	5	155
6	633	542.04	542.31	215	2	217
7	791	542.59	542.31	270	2	272
8	855	542.31	—	292	—	292

表5に性能評価結果の詳細を示す。ACO法からSXX法への最適な交代時期は1世代目から次善解を探索した791世代目の中で見つける。探索した最適解の経路長が変化した21ヶの世代(1, 2, 3, 4, 5, 13, 25, 30, 31, 62, 63, 64, 65, 72, 75, 80, 113, 327, 451, 633, 791)が交代世代候補となるが、そのうち451, 633, 791世代が最適解を探索できた。このうち最適解探索時間が最も短い最適世代交代時期は451であった。表7にその一部の内訳を示す。図8, 表3にツアー数の観点からのECXOの評価, 図9, 表4にコンピュータ実行時間からの評価結果をまとめている。

6. 結論と今後の課題

本論文では、巡回セールスマン問題(TSP)を効率的に解く手法として遺伝子交叉オペレータ交代法(CXO法)と蟻協調行動モデル(ACO法)の機能と性能について実験的に比較評価した。Eilonの75都市問題に対するCプログラム実験の結果、(1)CXO法もACO法もいずれも同一の最適解(経路長=542.31)を探索可能なこと、(2)最適解を探索するのに、シミュレーテッドアニーリングを有するACO法では64, 125ツアー292秒かかること、最適なオペレー

タ交代時期選択のもとCXO法では68,000ツアー33秒であることがわかった。両方法に性能差が生じる理由は、次に選択する都市は現在の都市のみならず過去に訪れた都市を考慮してCXO法では選択できるが、二都市間のフェロモン量のみに着目して次の都市を選択するACO法では過去に訪れた都市によって次に訪れる都市を選択できないためである。(3)ACO法からSXXに交代するECXO法は、ACO法やCXO法の探索効率を向上できることを実験で検証した。実験空間を拡張してのECXO法の有効性の検証を今後の課題とする。

ACO法やCXO法において、4.3節と4.4節に述べたACOの起動パラメータやCXOの起動パラメータが機能(「探索した最短経路長」)や性能(=最短経路を探索するのに要したコンピュータ実行時間)に大きな影響を与えている。また同じ都市データでも並び順が違くと性能や機能に大きな影響を与えることがわかっている。これらの起動パラメータと機能や性能の関係の分散分析を今後の課題である。

参考文献

- [1] M. Dorigo and T. Stützle, Ant Colony

- Optimization, The MIT Press, 2004.
- [2] M. Dorigo and L.M. Gambardella, Ant Colonies for the traveling salesman problem, TR/IRIDIA/1996-3, Universite Libre de Bruzelles, Belgium, Biosystems, 1997.
- [3] J.H. Holland, Adaptation in Natural and Artificial Systems, MIT Press, 1992.
- [4] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Inc., 1989.
- [5] 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.
- [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, 1987, pp. 224-230.
- [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, 1996, pp. 535-540.
- [8] M. Garey and D. Johnson, Computers and Intractability. A Guide to the Theory of NP-Completeness, W. H Freeman and Company, 1979.
- [9] S.J. Russell and P. Norvig, Artificial Intelligence-A Modern Approach, Prentice Hall, Upper Saddle River, New Jersey, 1995.
- [10] B.W. Kernighan and D.M. Ritchie, The C Programming Language, Second Edition, Bell Telephone Laboratories, 1988.
- [11] H. Hirano, A Genetic Algorithm C programming, Personal Media Inc., 1995, pp. 232-238 (in Japanese).
- [12] R.L. Haupt and S.E. Haupt, Practical Genetic Algorithms, John Wiley & Sons, Inc., 2004.
- [13] E. Aarts and J.K. Lenstra, Local Search in Combinatorial Optimization, Princeton University Press, 2003.
- [14] Ryouei Takahashi, "Solving the Traveling Salesman Problem through Changing Crossover Operators," in Proceedings of the 10th Conference on Artificial Intelligence and Applications, p. 42, hosted by National University of Kaohsiung, 2005.
- [15] 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.
- [16] Ryouei Takahashi and Kenichi Degai, "A Performance Improvement of Genetic Algorithms through Changing Crossover Operators to Solve the Traveling Salesman Problem," in Proceedings of the 8th International Conference on Computer and Information Technology, pp. 40-45, organized by Islamic University of Technology, 2005.