

# 遺伝的アルゴリズムの効率的交叉法について

片山 謙吾・成久 洋之\*

岡山理科大学大学院工学研究科博士課程システム科学専攻

\*岡山理科大学工学部情報工学科

(1997年10月6日 受理)

## あらまし

巡回セールスマン問題 (TSP) に対する遺伝的アルゴリズムの交叉法として、形質の遺伝を重視した、山村らのサブツアー交換交叉がある。この交叉法は、子孫生成以前に二つの順列に対して共通するサブツアー (共有サブツアー) をすべて列挙する必要がある、山村らはその列挙に  $O(n^5)$  の時間を費やしている。本論文では、サブツアー交換交叉で対象となる共有サブツアーの選び方を一部簡略化した、完全サブツアー交換交叉 (CSE-X) について記述する。CSE-X ではその列挙が  $O(n)$  の時間で実現されることを示す。更に数値実験では、要素数50万までの一様分布で発生したランダムな順列に対する共有サブツアー数、その長さ等について検討する。また TSP のベンチマーク問題集 TSPLIB を用いて実際の動作傾向について検討を加える。

## 1. まえがき

遺伝的アルゴリズム (genetic algorithm, GA)<sup>1,2)</sup> は、生物の進化過程にヒントを得た探索アルゴリズムで、さまざまな組合せ最適化問題や応用問題<sup>3)</sup> に適用され、極めて優れたロバスト性を持っている。一般に GA は、選択、交叉、突然変異とよばれる三つの遺伝的操作によって探索が進められ、解空間に対して大域的な探索が非常に優れ、良好な解を算出できる解法である。

近年の GA においては、順序づけ問題に対して二つの順列に共通して含まれる区間を用いる交叉法が数多く提案されている<sup>7,10,12,13,18,28)</sup>。つまり、その区間を利用することで、親の形質をできる限り破壊しない試みがなされている。その他にもさまざまな異なる交叉法が文献<sup>21,22,23,24,25,26,27)</sup> で紹介されている。

山村ら<sup>7)</sup> は、順序づけ問題の一つである巡回セールスマン問題 (traveling salesman problem, TSP) に対して形質の遺伝を重視したサブツアー交換交叉を提案した (文献<sup>12,13)</sup> ではジョブショップスケジューリング問題に適用されている)。この交叉法は、二つの親に含まれるサブツアーを形質と見なし、後世に継承することを重視した方法で、子孫生成以前にその二つの親 (順列) に共通して含まれる区間 (共有サブツアー) をすべて列挙する必要がある。

る (Brady<sup>10)</sup> の交叉も同様の列挙を必要とする)。なお、山村らはその列挙に  $O(n^5)$  時間を費やしている。

一方、柳浦ら<sup>8)</sup> は、共有区間列挙問題および共有相補区間列挙問題の両方を解くことにより  $O(n^2)$  時間の共有サブツアー列挙アルゴリズムを提案し、一様分布により生成されたランダムな順列に対しては、その共有サブツアー数の期待値が通常の順列の場合に  $2 + O(n^{-1})(n \geq 5)$  以下 (円環状の順列では  $4 + O(n^{-1})$ ) であると報告している。

本論文では、山村らのサブツアー交換交叉や Brady の交叉で対象とした部分集合のサブツアーとは一部異なり、サブツアーの方向性を重視した部分集合を扱う、完全サブツアー交換交叉<sup>18),20)</sup> について紹介する。二つの順列に含まれるすべての共有サブツアーの列挙が  $O(n)$  の時間で可能であることを示す。更にその共有サブツアーの数が、実験的側面からどの程度に上るかを要素数50万までの一様分布で発生したランダムな順列について検討する。そこでは、二つの順列に共有サブツアーが全く含まれない割合が13~14(%)あることを示す。また、TSP のベンチマーク問題集 TSPLIB を取り扱い、完全サブツアー交換交叉の実際の動作傾向についても検討する。そこでは、比較的良好の解を算出し、候補解に対する多様性の点でも優れた交叉法であることを示す。

## 2. 完全サブツアー交換交叉について

完全サブツアー交換交叉<sup>18),20)</sup> は、サブツアー交換交叉や Brady の交叉で対象としたサブツアーの選び方を一部簡略化している。以下、その違いを明確に記述する。

### 2.1 サブツアー交換交叉

親の形質を子孫に継承する交叉法のひとつにサブツアー交換交叉<sup>7)</sup> がある。これは、GA の成功を助ける building blocks<sup>1,2,7)</sup> の蓄積のために部分的に成功しているサブツアーを破壊しないように配慮されている。更に実行不可能解を生成せず、GA 本来の振る舞いに適合した性質を持つ交叉法である。この交叉法は、二つの親に共通する部分集合を共有サブツアーと考慮し、それを列挙した上で子孫が生成される。なお、山村らはその列挙に  $O(n^5)$  の時間 (但し、高速化のためサブツアーの縮約に  $O(n^2)$  を要する) を必要とした。また、Brady<sup>10)</sup> の交叉でも同様な共有サブツアーが扱われている (なお、文献<sup>11)</sup> にも Brady の交叉の紹介がある)。

図1は、要素数が10の親A ( $a b c d e f g h i j$ ) と親B ( $e j h g f a i c b d$ ) に対するサブツアー交換交叉の交叉例を示す。対象となる共有サブツアーは、親Aの  $\boxed{b, c, d}$  と  $\boxed{f, g, h}$  であり、親Bの  $\boxed{h, g, f}$  と  $\boxed{c, b, d}$  である。サブツアー交換交叉では、親Aの  $\boxed{b, c, d}$  と親Bの  $\boxed{c, b, d}$  のように点の順番が互いに異なっても、サブツアーとしての部分集合が一致している場合には、共有サブツアーとして列挙され子孫を生成する。例えば、子1では親Bの  $\boxed{c, b, d}$  を親Aの  $\boxed{b, c, d}$  の部分へ組み込む操作を施している。また、子4では親B

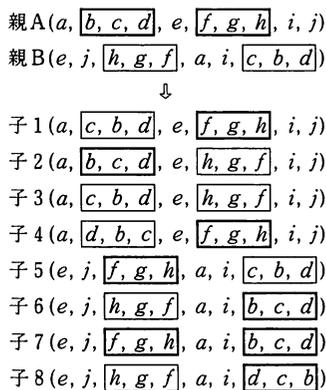


図1 サブツアー交換交叉の例



図2 完全サブツアー交換交叉の例

の $\overline{c, b, d}$ の逆方向 $\overline{d, b, c}$ を組み込んでいる。残りの子孫も、同様な操作により生成する。

## 2.2 完全サブツアー交換交叉

文献<sup>20)</sup>で報告された交叉法は、二つの親に共通する全く同じ方向性を有するサブツアー（逆方向のサブツアーも含む）を共有サブツアーとして列挙し子孫を生成する方法である。すなわち、サブツアー交換交叉で対象にした共有サブツアーの選び方を一部簡略化した共有サブツアーを扱う。従って、完全に一致した方向性を持つ共有サブツアーを扱う観点から完全サブツアー交換交叉（complete subtour exchange crossover, CSE-X）とよばれる。

図2は、対象となる共有サブツアーが両親に2つ存在する場合のCSE-Xの交叉例を示す。なお、図1で使用した親A、Bと同じ両親を扱い、両交叉法の列挙されるべき共有サブツアーの相違を明確に示す。図2で対象となる共有サブツアーは、親Aの $\overline{b, c}$ と $\overline{f, g, h}$ 、親Bの $\overline{h, g, f}$ と $\overline{c, b}$ である。例えば、図1のサブツアー交換交叉では親Aの $\overline{b, c, d}$ と親Bの $\overline{c, b, d}$ が共有サブツアーとして考慮されるが、CSE-Xでは、それを共有サブツアーとして考慮に含めない。よって図2のCSE-Xでは、親Aの $\overline{b, c}$ と親Bの $\overline{c, b}$ が共有サブツアーの対象となり、 $d$ はその対象から外す。つまり、CSE-Xで対象となる共有サブツアーは、その部分集合の方向性が互いに同一、または対称になる場合のみが考慮される。そのような共有サブツアーを対象とすることにより、次のbfで示す高速な共有サブツアー列挙アルゴリズムが可能となる。

子孫の生成は、図2に示すように、列挙された共有サブツアーの逆方向も含めて生成される。もし、両親に対象となる共有サブツアーが $Sub$ 個ある場合は、 $2 \times 2^{Sub} - 2$ 個の子孫が生成される。

### 3. 共有サブツアー列挙アルゴリズム

#### 3.1 定義

本節では、順列(本論文は通常の順列を扱う)、共有サブツアーの区間およびその区間長の定義を与える。

順列  $N = \{1, \dots, n\}$  を  $\sigma: N \rightarrow N$  で表し、 $i$  番目の要素が  $j$  であることを  $\sigma(i)$  または、 $\sigma^{-1}(j) = i$  とし、二つの順列は  $\sigma_A, \sigma_B$  とする。 $\sigma_A, \sigma_B$  のそれぞれの区間は  $[l_A, r_A], [l_B, r_B]$  (但し、区間長は、 $2 \leq |r_A - l_A| + 1 \leq n - 2, 2 \leq |r_B - l_B| + 1 \leq n - 2$  である。なお、 $|r_A - l_A|$  および  $|r_B - l_B|$  は絶対値である。また、 $+1$  は区間長を2以上とするためである) で、条件

$$\{\sigma_A(i) | i \in [l_A, r_A]\} = \{\sigma_B(i) | i \in [l_B, r_B]\} \quad (1)$$

を満たすものとする<sup>9)</sup>。

#### 3.2 $O(n)$ 時間の列挙アルゴリズム

文献<sup>9)</sup>では、共有区間列挙問題および共有相補区間列挙問題の両方を解くことにより、共有サブツアーの列挙が  $O(n^2)$  時間で可能であることが報告された。そこでは、二つの順列における共有サブツアーの最大位置と最小位置の差がどちらの順列とも等しい事実に基づいているのに対して、CSE-X で用いられる列挙アルゴリズムは、常に二つの順列における共有サブツアーの各要素間の距離が互いに等しい事実に基づいている。つまり、暫定的に順列  $\sigma_A$  の  $i$  番目 ( $i = 1, \dots, n-1$ ) の要素が決定されたとき、条件

$$|\sigma_B^{-1}(\sigma_A(i)) - \sigma_B^{-1}(\sigma_A(i+1))| = 1 \quad (2)$$

となる事実を利用することで共有サブツアーの列挙を  $O(n)$  時間で可能にしている。その列挙アルゴリズムを図3に示す。但し、 $\sigma_B^{-1}$  ( $O(n)$  で実現可能) は、列挙アルゴリズムの処理以前に準備しておく。

図3は共有サブツアーの列挙およびその数、長さも算出できるように記述されている。変数  $num$  は二つの順列に存在する共有サブツアーの数、 $len$  は一つの共有サブツアーの長さ、 $flag$  は条件文で使用され、付属的な変数である。このアルゴリズムは、どちらか一つの順列に基準が置かれ (図3では  $\sigma_A$  である)、主に一つの **for** 文と三つの **if** 文で構成される。

簡単に列挙アルゴリズムを説明する。まず step07 では、一時的に共有サブツアーになりうる候補要素が置かれている順列の位置番号を保持しておく。(なお、 $i$  のインクリメントは step11 と step18 で行われる。) 次に  $\sigma_A$  の候補要素からの  $l_B$  と、保持されていた  $right_B$  との距離が1もしくは-1 (step14は条件(2)と同等) の時は、共有サブツアーになりうる部分集合が一時的に決定される。もしその距離が step14 の条件を満たさないならば、共有サブツアーになりうる範囲を超えたものと考えることができる。また共有サブツ

```

[共有サブツアー列挙アルゴリズム]
01: begin
02:  flag := 0 ; num := 0 ; len := 1 ;
03:  for i := 1 to n do
04:    begin
05:      lA := i ;
06:      lB :=  $\sigma_B^{-1}(\sigma_A(l_A))$  ;
07:      if flag = 0 then
08:        begin
09:          leftA := lA ;
10:          leftB := lB ; rightB := lB ;
11:          flag := 1 ; i := i + 1 ;
12:        end
13:      else
14:        if  $|right_B - l_B| = 1$  then
15:          begin
16:            rightA := lA ;
17:            rightB := lB ;
18:            i := i + 1 ; len := len + 1 ;
19:          end
20:        else
21:          if len > 1 then
22:            begin
23:              lA := leftA ; rA := rightA ;
24:              lB := leftB ; rB := rightB ;
25:              save [lA, rA], [lB, rB] and len of the subtour ;
26:              num := num + 1 ;
27:            end
28:            flag := 0 ; len := 1 ;
29:          end for
30:  output the all saved data and num of the subtours;
31: end.

```

図3 2つの順列に対する  $O(n)$  時間共有サブツアー列挙アルゴリズム

アーとなるためにはその長さが2以上であるので、それを step21 で判定し、共有サブツアーを構成した区間、その長さを保持し、*num* に1を加算する。そしてこの処理を繰り返す、最後に二つの順列に存在した共有サブツアーの数、その区間ならびにその長さを出力することにより、 $O(n)$  時間列挙アルゴリズムを終了する。

#### 4. 共有サブツアーに関する実験と考察

一様分布で発生した二つの順列  $\sigma_A, \sigma_B$  に対して、文献<sup>9)</sup>では共有サブツアー数の期待値は、要素数  $n$  が5以上の時、 $2 + O(n^{-1})$  以下であることが示された。本実験では、二つの順列の要素数  $n$  が  $5 \sim 5 \times 10^5$  の場合について実験的側面から検討する。実験には、32MBのメモリーを有した Sun 4/5 (microSPARCII 110MHz) 上にC言語を使用した。

表1は、各要素数に対して一様分布で生成された二つの順列に存在する共有サブツアー

の数, その長さならびに処理に要した平均計算時間 (図3の列挙アルゴリズムと  $\sigma_B^{-1}$  に必要な計算時間が含まれる) を各要素数につき試行回数1000回の結果を示す。共有サブツアー数および共有サブツアー長での max, avg, min は, 試行回数1000回で算出された最大値, 平均値, 最小値をそれぞれ示す。なお,  $10^2$  の要素数までの平均計算時間は, 処理が極めて高速であるため省略する。また, 表2は各要素数に対して算出された共有サブツアーの数別の割合(%)を示す。

まず, 計算時間について検討する。表1では, 順列の要素数が各2倍, 5倍ずつ増加する場合の実験を行っている。よって  $O(n)$  時間の列挙アルゴリズムでは処理される計算時間が各2倍, 5倍ずつ増加することが期待される。しかしながら, 実際にはやや不規則な計算時間の増加が示された。これは計算時間を計測する時間関数の精度の問題が多分にあると考えられる。

表1から  $5 \times 10^1$  以上では, 共有サブツアー数は平均的に2であり, 最大値では多いもので9あるが, 最小値では全要素数を通して共有サブツアーの存在しない場合があることを確認できる。また長さでは平均的に2である。表2から, その共有サブツアーが存在しない割合が, 13~14(%)前後であることがわかる。また, 柳浦<sup>9)</sup>によれば, 表2に示した結果は, 理論値に非常に近いものであった。

次に表1, 2の実験結果から以下のことをまとめる。

表1 各要素数に対する共有サブツアー数, 長さ, 平均計算時間の実験結果 (試行回数: 各1000回)

要素数 $n$	5	$10^1$	$5 \times 10^1$	$10^2$	$5 \times 10^2$	$10^3$	$5 \times 10^3$	$10^4$	$5 \times 10^4$	$10^5$	$5 \times 10^5$
サブツアー数 max	2	4	6	6	8	8	7	9	9	7	8
avg	0.911	1.400	1.948	1.971	1.968	1.955	2.030	2.013	2.056	2.004	1.955
min	0	0	0	0	0	0	0	0	0	0	0
サブツアー長 max	4	4	3	3	3	3	2	2	2	2	2
avg	2.166	2.120	2.014	2.009	2.004	2.002	2.000	2.000	2.000	2.000	2.000
min	2	2	2	2	2	2	2	2	2	2	2
計算時間 avg (sec)	-	-	-	-	0.001	0.001	0.010	0.021	0.115	0.251	1.390

表2 共有サブツアーの数別の割合(%)

要素数 $n$	5	$10^1$	$5 \times 10^1$	$10^2$	$5 \times 10^2$	$10^3$	$5 \times 10^3$	$10^4$	$5 \times 10^4$	$10^5$	$5 \times 10^5$
0	20.2	16.0	13.0	13.5	11.9	14.7	12.9	12.3	14.0	14.4	14.1
1	68.5	39.6	27.6	28.6	29.0	27.1	26.5	28.6	25.5	25.7	27.6
2	11.3	33.5	28.8	25.6	28.7	26.5	27.5	27.3	26.6	27.2	26.0
3	-	10.2	17.7	17.4	18.0	18.4	18.3	18.0	18.0	17.9	18.2
4	-	0.7	9.0	10.4	7.4	8.2	9.3	8.0	9.6	9.3	10.4
5	-	-	2.9	3.5	3.6	3.7	3.8	3.8	4.7	4.1	2.9
6	-	-	1.0	1.0	1.1	1.1	1.5	1.4	0.7	0.9	0.5
7	-	-	-	-	0.2	0.1	0.2	0.2	0.5	0.5	0.2
8	-	-	-	-	0.1	0.2	-	0.3	0.3	-	0.1
9	-	-	-	-	-	-	-	0.1	0.1	-	-

- 50以上の要素数においては共有サブツアーの数は平均的に2前後で抑えられる。
- $10^3$ までの要素数に対しては共有サブツアーの長さは最長で3, 4であるが、それ以上の要素数では、3以上になることは極めて少ない。
- 50以上の要素数において共有サブツアーが存在しない割合は、13~14(%)前後である。

本実験から、一様分布で生成された二つの順列に対する共有サブツアーは存在しない場合が13~14(%)前後の割合で発生することを確認した。これは二つの順列の共有区間を利用する交叉法を用いる場合において、少ない割合であるとは考えにくい。更に共有サブツアーの長さも要素数によるが平均的に3以上になることは望めない。よって、サブツアー交換交叉に類似した交叉法を利用する場合には、初期世代からランダムで生成された解よりは、共有区間の数がある程度見込まれる、比較的質の良い解を用いた方が効率化の面また利用面から賢明であると考えられる。

## 5. TSPLIB による動作実験と考察

完全サブツアー交換交叉 (CSE-X) はさまざまな順序づけ問題に適用可能であるが、ここでは巡回セールスマン問題を具体例として取り上げ、CSE-X の実際の動作について検討を加える。

### 5.1 巡回セールスマン問題とコード化

巡回セールスマン問題 (TSP) <sup>4,6,16)</sup> は、 $n$  個の点から構成される無向完全グラフ  $G = (V, E)$ 、枝上の距離関数  $d: E \rightarrow Z^+$  が与えられたとき、すべての点をちょうど1度ずつ経由する巡回路 (Hamilton 閉路) で、枝上の総距離を最短とする巡回路を求める最小化問題である。本実験で扱う TSP は、2点  $i, j$  間の双方向の距離が等しい ( $d_{ij} = d_{ji}$ ) 対称巡回セールスマン問題 (symmetric TSP) である。なお、実行可能解の数は  $(n-1)!/2$  に上る。

GA を TSP に適用する場合 (コード化) は、候補解 (または個体とよぶ) が Hamilton 閉路に対応し、Hamilton 閉路に含まれる1点が遺伝子に対応する。すなわち、染色体の長さは都市サイズに対応し、それをパス表現とよぶ。また、適応度は Hamilton 閉路の総距離で計算する。

### 5.2 完全サブツアー交換交叉を施した GA

完全サブツアー交換交叉 (CSE-X) を施した GA について記述する。CSE-X では、二つの親に共通する全く同じ方向性を有するサブツアー (逆方向のサブツアーも含む) を共有サブツアーと考慮して子孫を生成する方法であり、サブツアー交換交叉や Brady の交叉で対象とする共有サブツアーの選び方を一部簡略化している。そのようなサブツアーを扱うことで、子孫の数が膨大になろうともその生成は容易になる。つまり、子孫生成に際しては、そのサブツアーの部分を逆順にすることが可能であり、その操作は TSP の 2-Opt 近傍<sup>4,6,16)</sup>を適用することに等しい (なお、一つの順列にサブツアーが複数ある場合は、2

-Opt 近傍をその都度施す)。よって、共有サブツアーの数分の組合せの子孫が生成されるとしても、適応度の計算はその子孫すべてに対して容易、また高速に行うことができる。そのような操作を施すことで、生成された膨大な数の子孫およびその親から、優れた個体を必要な数だけ次世代に残すことで良質の解を算出できる。

なお、実験における突然変異およびその他の詳細については次節で記述する。

### 5.3 動作実験と結果

本節では、TSP のベンチマーク問題集 TSPLIB<sup>17)</sup>から、問題サイズ51~1000都市の最適解既知の問題を4問選び、CSE-X を施した GA の動作について、単純な枠組みの GA の下で CSE-X の収束性、二つの順列に存在したサブツアー数、その長さ等に対して検討、考察する。

本実験の詳細事項として、個体数(候補解数)を32、交叉確率を1.0、突然変異確率は0.05で、TSP に対して一般的に採用される一回の2-Opt をランダムに使用する。なお、これらのパラメータは処理中一定である。初期解は一律にランダムで生成し、処理の打ち切り世代は50000世代までとする。選択淘汰には、最も適応度の高い個体を次世代に残すエリート保存戦略<sup>12)</sup>を採用する。また、交叉のペア決定に際しては、候補解からランダムにペアを選ぶ方法を採用する。このような単純な枠組みにすることで CSE-X の動作傾向を見えやすくした。なお、本実験は Sun 4 / 5 上に C 言語を使用した。

表3に、eil51問題、kroA100問題、att532問題、dsj1000問題に対して(問題名にある数値は都市数を示す)、TSP 専用アルゴリズムとして有名な、貪欲法の一つである Nearest Neighbor (NN) 法<sup>6,16)</sup>、および2-Opt 近傍を施した局所探索法 (Local Search, LS) で得られた結果と、CSE-X を有した GA を適用した場合の50000世代目で算出された最適解から離れた解の質 (Min., Max., Avg.)、および平均計算時間(CT)を示す。なお、解の質は  $\frac{\text{得られた解} - \text{最適解}}{\text{最適解}} \times 100(\%)$  で算出した。

表3に示した結果について考察する。NN で算出される解の質は、最適解から約25(%)程度であることが報告されており<sup>6)</sup>、我々の実験からもほぼ同程度の解が算出された。また2-Opt 近傍を施した LS では、NN よりも良質の解が算出可能である。CSE-X の結果から、膨大な計算時間を必要としているが、LS より良質の解が算出された。局所探索法を組み込んだ GA を考慮する場合は、更に高精度な解を期待できる。従って、ここで算出された解は局所探索法を含まず CSE-X の探索による GA のみの場合における結果であり、局所探索法との組み合わせを必要とすることが推察できる。しかしながら、簡単な GA の枠組みであるにも関わらず、比較的良質の解を得られる交叉法であることを示した。なお、本実験で設定したパラメータ等は経験的に選んだものであり、パラメータ値の最適化および選択淘汰の高性能化により、更に良質の解が期待できるものと考えられる。

次に、図4の(a)eil51問題、(b)kroA100問題、(c)att532問題、(d)dsj1000問題には、それぞれの問題例に対して CSE-X を有した GA によって算出された、試行回数5回の平均の共

表3 TSPLIB に対する Nearest Neighbor, 局所探索法および CSE-X の算出結果 (試行回数: 各5回)

TSPLIB	Nearest Neighbor (%) CT (sec)				Local Search (%) CT (sec)				CSE-X (%) CT (sec)			
	Min.	Max.	Avg.	Avg.	Min.	Max.	Avg.	Avg.	Min.	Max.	Avg.	Avg.
eil51	13.8	20.9	18.0	0.00	1.2	13.3	6.1	0.11	2.1	5.4	3.8	5356
kroA100	20.5	26.7	25.0	0.01	5.8	14.1	10.3	0.61	3.2	7.0	5.5	11838
att532	24.2	34.6	28.7	0.41	9.3	14.1	11.7	29.2	9.2	13.5	11.2	65932
dsj1000	27.3	35.1	29.7	0.68	11.9	14.5	13.0	92.2	10.0	20.9	12.9	135879

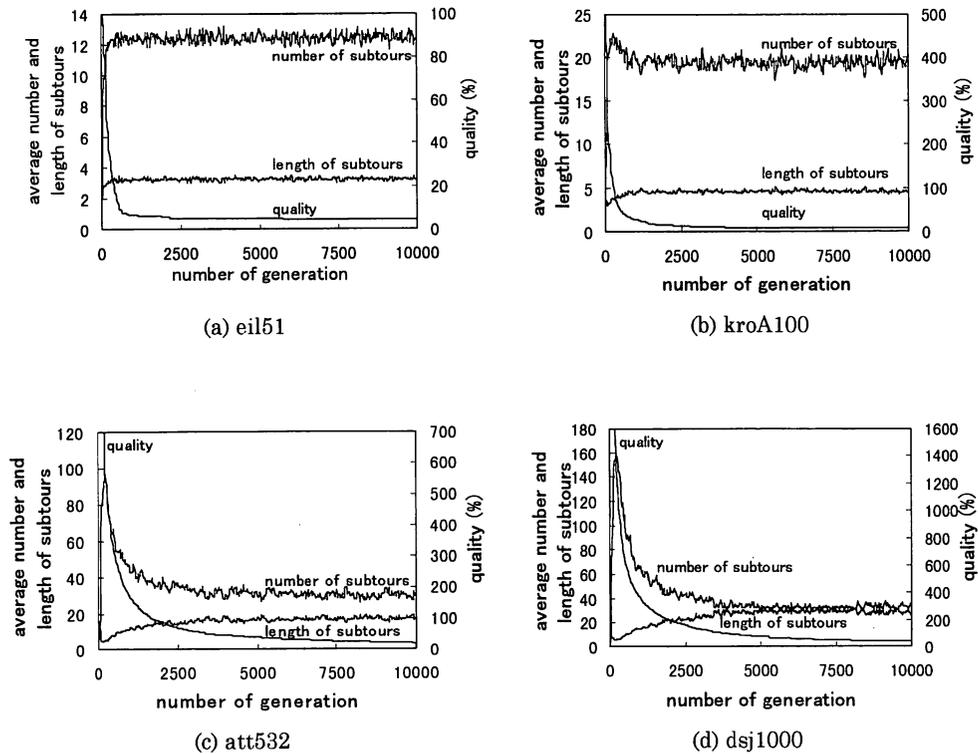


図4 TSPLIB による完全サブツアー交換交叉の共有サブツアー数, その長さと収束傾向の例

有サブツアー数およびその長さと収束傾向を, 各25世代ごとに算出された数値に対して10000世代まで示す。図4の(a)以外では, 共有サブツアー数は比較的短い世代数で急激にその数を増加し, 更には減少し, ある世代からは横ばいになる傾向を示した。この傾向から考えられることは, 初期の世代においては, 比較的短い長さを持つ良質のサブツアーや質の悪いサブツアーが含まれることにより, その数を増加させることが予測される。また, 探索が進むに従い良質のサブツアーを building blocks として蓄積することで, 共有サブツアーの長さを伸ばすとともにその数を減少させるものと考えられ, ある世代からはほとんど大きな増減はなくなり, 横ばい状態になる。共有サブツアーの長さについては, 初期世代から徐々に長くなり, 共有サブツアー数とほぼ同世代からその増減はなくなることが確認で

きる。また、収束傾向を示す適応度の曲線においては、比較的長い世代をかけて横ばい状態に達し、その横ばいになる世代あたりから共有サブツアー数および長さはその増減が少なくなる傾向を示した。なお、局所探索法を併用した場合においては、その曲線が更に急勾配になるものと考えられ、そこで算出される共有サブツアー数、その長さにおいても、本実験で得られた傾向とは異なるものと推測できる。局所探索法を併用した場合の傾向については、文献<sup>19)</sup>参照。

#### 5.4 解の多様性について

候補解の多様性の点では、図4の結果からある程度の推測が可能である。もし、多様性を失う場合には、その結果が共有サブツアー、その長さに顕著に現れると考えられる。つまり、多様性が失われることは、個体の集団に類似した解が多数含まれることを意味している。共有サブツアーの数が極めて減少し(零付近まで減少すると考えられる)、その共有サブツアーの長さは長くなるものと考えられる。しかしながら、図4でその減少が観測されたのは、収束が大幅に進んでいる時期であり、これは多様性を喪失しているのではなく、その時期と同時期に共有サブツアー長が伸びていることから悪質のサブツアーの排除によるものと考えられる。ある程度収束した後には、共有サブツアー数や、長さに大きな増減が見られないことから、候補解に対して多様化を保ちながら探索が進められる交叉法であることが確認できる。

## 6. むすび

本論文は、サブツアー交換交叉で対象とした部分集合のサブツアーとは一部異なり、サブツアーの方向性を重視した部分集合を扱う、完全サブツアー交換交叉(CSE-X)について記述した。ここでは、二つの順列に含まれるすべての共有サブツアーを  $O(n)$  時間で列挙するアルゴリズムを示した。更に要素数が50万までの数値実験によって、一様分布で発生したランダムな順列に存在する共有サブツアーの個数、その長さならびに提案した列挙アルゴリズムの高速性を確認した。ここでは、二つの順列に対して13~14(%)前後の割合で共有サブツアーが存在しないことなどを示した。

更に、TSPのベンチマーク問題集 TSPLIB の問題例に対する完全サブツアー交換交叉の動作実験では、完全サブツアー交換交叉で算出される各世代の共有サブツアー数、その長さについてその傾向を示すとともに、単純なGAの枠組みの下で、比較的良質の解を得られることを示した。また、多様性維持の点でも優れた交叉法であることを確認した。なお、文献<sup>20)</sup>では、Whitleyによって提案された *Edge Recombination Crossover* (ER-X)<sup>24)</sup>の改良版 Improved ER-X (IER-X)<sup>25)</sup>より、CSE-Xは優れた性能を示している(Starkweatherら<sup>25)</sup>は、六つの有名な交叉法の性能を分析し、IER-Xが最も優れていたことを報告している)。更にCSE-Xは、2-Opt近傍を有した局所探索法をハイブリッドしたGAの枠組みの下で、TSPで現在有望視されているMühlenbeinらの交叉(Müh-X)<sup>11)</sup>より、優

れた性能を示している<sup>19)</sup>。形質の遺伝を重視した交叉を施した GA は、局所最適解からの脱出能力が、Simulated Annealing 法より優れていると報告された文献がある<sup>14,15)</sup>。今後、多方面からの交叉法の性能に関する研究<sup>19)</sup>が重要であると考えられる。また最近の研究から、Freisleben ら<sup>28)</sup>は、*Distance Preserving Crossover* (DP-X) とよばれる新しい交叉法を提案した。これは、CSE-X と同様な共有サブツアーを用いており、その共有サブツアーなどを貪欲的手法によってつなぎ合わせ、実行可能な子孫を生成する交叉法である。この DP-X を交叉法として施した GA は、TSP に対して最も強力とされる Lin と Kernighan の手法 (LK 法)<sup>5)</sup>をハイブリッドしている。このハイブリッド GA は、今までに提案された数々の GA のアプローチ (ハイブリッド GA 等も含む) を凌ぐ性能が文献<sup>28)</sup>で報告されている。よって、我々の今後の課題として、CSE-X を有した GA に LK 法をハイブリッドする予定である。また、どの程度の性能が示されるか興味深い。

## 謝 辞

共有サブツアー数などに関して有益なコメント、御指導等を頂きました、京都大学工学部の柳浦陸憲先生に感謝の意を表します。

## 参考文献

- 1) D. E. Goldberg: "Genetic algorithms in search Optimization and machine learning," Addison-Wesley Publishing Company Inc., 1989.
- 2) 北野宏明編: "遺伝的アルゴリズム," 産業図書, pp. 3-60, 1993.
- 3) L. Chambers: "Practical handbook of genetic algorithms applications," CRC Press Inc., Vol. 1, pp.143-172, 1995.
- 4) M. O. Bell, T. L. Magnanti, G. L. Nemhauser, and C. L. Monma: "Handbooks in operations research and management science network models," North Holland, Vol. 7, pp.225-330, 1995.
- 5) S. Lin and B. W. Kernighan: "An effective heuristic algorithm for the traveling salesman problem," Operations Research, Vol.21, pp.498-516, 1973.
- 6) D. S. Johnson and L. A. McGeoch: "The traveling salesman problem: a case study," Local Search in Combinatorial Optimization, Wiley-Interscience Series in Discrete Mathematics and Optimization, pp.215-310, 1997.
- 7) 山村雅幸, 小野貴久, 小林重信: "形質の遺伝を重視した遺伝的アルゴリズムに基づく巡回セールスマン問題の解法," 人工知能誌, Vol. 7, No. 6, pp.1049-1059, 1992.
- 8) 柳浦陸憲, 永持 仁, 茨木俊秀: "サブツアー交換交叉に対する二つのコメント," 人工知能誌, pp.464-467, 1995.
- 9) 柳浦陸憲: ディスカッション, 1997.
- 10) R. M. Brady, "Optimization strategies gleaned from biological evolution," Nature, vol. 317, pp.804-806, 1985.
- 11) H. Mühlenbein, M. Gorges-Schleuter, and O. Krämer: "Evolution algorithms in combinatorial optimization," Parallel Computing 7, North-Holland, pp.65-85, 1988.
- 12) 小林重信, 小野 功, 山村雅幸: "遺伝的アルゴリズムによるジョブショップスケジューリングの性能評価," 日本機械学会シンポジウム講演論文集, No.930-80, pp.27-32, 1993.

- 13) 小野 功, 山村雅幸, 小林重信: “遺伝的アルゴリズムによるジョブショップスケジューリング,” 第19回知能システムシンポジウム予稿集, pp.23-28, 1994.
- 14) 仙石浩明, 吉原都男: “遺伝的アルゴリズムの最適解探索能力に関する評価 —GA とSAの比較—,” 第47回情報学全国大会論文集, pp.223-234, 1993.
- 15) 仙石浩明, 吉原都男: “GAによるヒューリスティック探索の最適化,” 情報学論, Vol.37, No.10, pp.1811-1820, 1996.
- 16) 久保幹雄: “メタヒューリスティックス”, “離散構造とアルゴリズムIV, 近代科学社, pp.171-230, 1995.
- 17) G. Reinelt: “TSPLIB,” ftp://softlib.rice.edu/pub/tsplib/tsplib.tar.
- 18) K. Katayama, H. Sakamoto, and H. Narihisa: “An efficiency of hybrid mutation genetic algorithm for traveling salesman problem,” Proc. 2nd Australia-Japan Workshop on Stochastic Models in Engineering, Technique & Management, Gold Coast, Australia, pp.294-301, 1996.
- 19) 片山謙吾, 平林永行, 成久洋之: “TSPに対する並列ハイブリッド遺伝的アルゴリズムの一検討,” 信学技報, COMP 97-31, Vol.97, No.157, pp.17-24, July, 1997.
- 20) 片山謙吾, 成久洋之: “完全サブツアー交換交叉の共有サブツアー高速列挙アルゴリズムと動作特性,” 信学論 (DI), 1998 (印刷中).
- 21) J. Grefenstette, R. Gopal, B. Rosmaita and D. Van Gucht: “Genetic algorithms for the traveling salesman problem,” Proc. 1st ICGA, pp.160-168, 1985.
- 22) I. M. Oliver, D. J. Smith, and J. R. C. Holland: “A study of permutation crossover operators on the traveling salesman problem,” Proc. 2nd ICGA, pp.224-230, 1987.
- 23) P. Jog, J.Y. Suh, and D. Van Gucht: “The effects of population size, heuristic crossover and local improvement on a genetic algorithm for the traveling salesman problem,” Proc. 3rd ICGA, pp.110-115, 1989.
- 24) D. Whitley, T. Starkweather, and D. Fuquay: “Scheduling problems and traveling salesman: the genetic edge recombination operator,” Proc. 3rd ICGA, pp.133-140, 1989.
- 25) T. Starkweather, S. McDaniel, K. Mathias, D. Whitley, and C. Whitley: “A comparison of genetic sequencing operators,” Proc. 4th ICGA, pp.69-76, 1991.
- 26) K. Mathias and D. Whitley: “Genetic operators, the fitness landscape and the traveling salesman problem,” Parallel Problem Solving for Nature - PPSM II, (R. Männer and B. Manderick, eds.), pp.219-228, 1992.
- 27) A. Y. C. Tang and K. S. Leung: “A modified edge recombination operator for the traveling salesman problem,” Parallel Problem Solving for Nature - PPSN III, (H. -P. Schwefel and R. Männer, eds.), pp.180-188, 1994.
- 28) B. Freisleben and P. Merz: “New genetic local search operators for the traveling salesman problem,” Parallel Problem Solving form Nature - PPSN IV, (H. -M. Voigt, W. Ebeling, I. Rechenbergm, and H. -P. Schwefel, eds.), pp.890-899, 1996.

# An Efficient Crossover Operator on Genetic Algorithms

Kengo KATAYAMA and Hiroyuki NARIHISA\*

*Graduate School of Engineering*

*\*Department of Information & Computer Engineering*

*Faculty of Engineering*

*Okayama University of Science,*

*Ridai-cho 1-1, Okayama 700-0005, Japan*

(Received October 6, 1997)

The subtour exchange crossover operator was originally proposed by Yamamura et al. in the genetic algorithm for solving the traveling salesman problem (TSP). In advance, the crossover required to enumerate all common subtours, which consist of the same set in a pair of subtours on the given two tours of  $n$  cities. The algorithm of Yamamura et al. consumed an  $O(n^5)$  time in this enumeration of all common subtours.

In this paper, we propose a new crossover operator, *complete subtour exchange crossover* (CSE-X), which consumes only  $O(n)$  time for the enumeration by simplifying to select the common subtours of the subtour exchange crossover. In our experiment, we show the experimental number and length, of common subtours for two randomly generated tours, and the practical behaviors of the CSE-X using the TSPLIB, a set of benchmark instances for the TSP.