

## Java 言語による枝組立交叉 (EAX) の並列処理実現方式の検討

～巡回セールスマン問題 (TSP) を解く GA の並列処理～

元田 剛<sup>††</sup> 高橋 良英<sup>††</sup>

Java 言語のマルチスレッド機能による枝組立交叉(EAX)の並列処理実現方式を研究する。EAX は両親の枝情報を交換して巡回セールスマン問題 (TSP) を解く近代的な遺伝的アルゴリズムの一手法である。遺伝的アルゴリズムの並列処理で集団を複数のプロセッサに分割・分散して処理時間の短縮を図るときの問題点は、各プロセッサで扱う集団サイズが小さくなるため、集団の多様性を確保できず全局最適解に陥る場合があることである。本検討では、集団の多様性を 2 段階で確保する方式を検討した。第 1 段では分割した小集団（島と呼ぶ）の多様性を確保する。その方法として<方式 1>島内の個体を自分の島の個体または他の島の個体と遺伝子交叉させ次世代の個体を生成する島間結婚方式、<方式 2>各島で生成した個体のうち経路長の短い優れた個体群を島間で移動する方式(島モデル)を検討した。第 2 段では、初期データが異なる条件の下、<方式 1>または<方式 2>で生成した暫定解群 (Family と呼ぶ) から優れた個体群を選択し併合する Family 併合方式と、優れた個体群を Family 間で移民する Family 間移民方式を検討した。方式の有効性を KroA150 並びに rat783 で検証した。

## Parallel Processing of Edge Assembly Crossover with Java Language

Gou Motoda<sup>††</sup> and Ryouei Takahashi<sup>††</sup>

Methodologies how to realize parallel processing of Edge Assembly Crossover to solve the Traveling Salesman Problem with Java multithreads are investigated. In parallel genetic algorithms it is important to invent methods to maintain diversity of the divided populations (islands). For maintaining the diversity of the islands, three hierarchical immigration methods are studied. The first method is to perform crossover operation on individuals between islands. The second method is to send superior emigrants between islands. The third method is to merge superior emigrants between islands. The validity of above methods is verified by using the data of KroA150 and rat783 in TSPLIB.

### 1. はじめに

巡回セールスマン問題 (TSP: Traveling Salesman Problem) とは、セールスマンが  $n$  箇所の都市を全て一回ずつ訪問し出発都市に戻る経路のうち最も短くなる経路（最適解）を求める問題である。TSP は代表的な組合せ最適化問題である。TSP の最適解の近似を効率的に得る手法として遺伝的アルゴリズム (GA: Genetic Algorithm) が有効であることが知られている。GA の探索性能は、遺伝的操作である交叉機能に依存するため、TSP に対して多くの交叉が提案されている<sup>[1]</sup>。その中でも枝組立交叉 (EAX: Edge Assembly Crossover) は、探索率、探索時間の両面において優秀である。EAX では親 A と親 B の同数の枝を交換し新たな巡回路を生成する。これまでの我々の実験では、集団サイズを大きく設定すれば、最適解を 100% 探索できることがわかっている。しかし集団サイズを増加させると計算量が増加してしまう。このため、集団を複数のコンピュータ（スレッドのこと）に分割し、分割した単位で並列処理する手法が解の探索時間短縮を図るために必要と考える。

本研究では、Java 言語のマルチスレッド機能を利用して、マルチプロセッサ上で遺伝子交叉処理 (EAX) を並列化させる方法を検討する。Java 言語では論理的な並列処理単位をスレッドと呼ぶ。マルチプロセッサを構成する任意のプロセッサを各スレッド実行時にスレッドに割り当て EAX を並列処理させる。マルチプロセッサではメモリはプロセッサ間で共用する。このため、各スレッドで管理する遺伝子データや遺伝子交叉のためにメソッド間で引き渡される作業用データはメモリ上スレッド毎に分割管理している。本資料では TSP のベンチマークテストデータ (TSPLIB) を用いた性能評価、性能改善結果を報告する。

### 2. 動機づけ実験

#### 2. 1 EAX の機能<sup>[1]</sup>

EAX では親 A と親 B の同数の枝を交換して新たな巡回路を生成する。

本検討の EAX は巡回路に適当な向きを決め有向グラフ（非対称型 TSP）として問題を解く。本検討では上りも下りも同じ長さである。

以下の考え方で両親 A、B から交換する枝を選び、枝交換後修正操作を施し子供を生成する。

<sup>††</sup> 八戸工業大学大学院 電子電気・情報工学専攻  
Hachinohe Institute of Technology

#### (1) AB-cycle 形成 :

EAX では両親の持つ枝を AB-cycle と呼ばれる複数の部分巡回路に分割する。AB-cycle は両親の枝を交互に繋げることによって得られる閉路のことである。そこでは、親 A の枝の終点を親 B の枝の終点、親 B の枝の始点を親 A の始点となるように交互に親 A と親 B の枝をつなぎ閉路を構成する。親 A と親 B の枝集合は互いに排他的な AB-cycle に一意に分割できることが定理として証明されている。

#### (2) 緩和個体形成 :

親 A の枝のうち AB-サイクルを構成する A の枝を B の枝に交換して緩和個体を生成する。AB-cycle の数を  $k$  とすると、AB-cycle の選択方法によって、 $2^k$  個の緩和個体が生成される。この組み合わせが多いことにより EAX は集団の多様性を確保している。緩和個体とは、AB-cycle によって枝交換された部分巡回路群のことを言う。

#### (3) 緩和個体の修正 :

緩和個体を Greedy-Repair 手法で修正操作することにより巡回路を再構成する。

### 2. 2 本検討での EAX の特徴 :

本検討では、AB-cycle ごとに緩和個体を生成する方式としている。それぞれの親から  $k$  個、合わせて  $2k$  個の緩和個体を生成する方式としている。この  $2k$  個の個体と両親を合わせトーナメントし、適応度の高い 2 個体を次世代の子とし生成する。

### 2. 3 並列処理の性能向上調査実験

EAX の並列処理の実現可能性を調査するために、TSPLIB にある 150 都市問題 (KroA150) で Java 並列処理プログラム開発実験を行った。EAX を並列処理する場合に実行時間に影響を与えるのはスレッド数である。実験では、並列処理の主パラメータであるスレッド数を 1, 2, 3, 4, 7, 8, 10, 12 と変化させ最適解と最適解の探索時間を測定した。最適解は TSPLIB で登録された経路長 (TSPLIB 最短経路長) と同じ経路長を持つ巡回路のことである。スレッド毎に 15 回の独立な最適解探索実験を行った。独立な試験とは初期巡回路群を変えて試験を行うことである。このため、一様乱数の初期値 (乱数種 (seed-id)) を変えて異なる初期巡回路群を生成した。

### 2. 4 実験環境

Windows7、マウスコンピュータ、Intel(R) Core (TM) i7 CPU 920 2.67GHz 2.67GHz, 6.0 GB RAM で行った。当コンピュータは 4 コア 8 スレッドのマルチプロセッサであり、最大 8 スレッドの並列処理が可能である。

### 2. 5 EAX の主な実行パラメータ

- (1) スレッド数・・1, 2, 3, 4, 7, 8, 10, 12
- (2) 集団サイズ=300、遺伝子交叉確率=0.8 (各スレッド共通)
- (3) スレッド毎に分割管理する個体数・・300/スレッド数

### 2. 6 実験結果

#### (1) TSPLIB 最短経路長平均探索時間

実験結果を表 1 と図 1 に示す。図 1 の X 軸はスレッド数 (1,2,3,4,7,8,10,12)、Y 軸は TSPLIB 最短経路長平均探索時間である。図 1 の詳細データを表 1 に示す。表 1 にはさらに、最適化探索時間の標本標準偏差を示している。スレッドサイズこの表と図から以下のことがわかる。

- ①適解平均探索時間が最も少ない場合はスレッド数が 8 の場合であること。
- ②スレッド数が 1 から 8 と増えると、最適解を見つけるまでに必要な平均最適解探索時間は段々減少していき最終的に 243 秒から 67 秒と 70% 減っている。この結果は、EAX が開始処理、世代交代処理という処理を並列化できない部分の割合  $\alpha$  が遺伝子交叉処理という並列化できる部分の割合  $(1 - \alpha)$  に比較して小さい為、スレッド数しが多くなると EAX の処理が  $\alpha + (1 - \alpha) / L$  に短縮できるというアムダール・グスタフソンの法則 ( $L=8$ ) に従っている。 $1 / (\alpha + (1 - \alpha) / 8) = 243 / 67$  を解き  $\alpha$  は本実験から約 0.17 であることがわかった。 $\alpha$  は集団サイズが大きければ大きいほど小さくなる。

表 1 KroA150 による並列 EAX の実験結果

スレッドサイズ	平均最適解探索時間	最適解探索時間の標本標準偏差	最適解探索率
1	243	31.6	93%
2	133	27.1	100%
3	109	39.6	100%
4	90	22.4	93%
7	71	8.7	100%
8	67	3.5	100%
10	70	7.7	100%
12	70	6.7	100%

③スレッド数が8から10,12と増えると、最適解を見つけるまでに必要な平均最適解探索時間は67秒から70秒へと若干増えていること。

(2)最適解探索率(15回中何回TSPLIB最短経路長を探索したか)(表1)

スレッド数=1,4を除いて、100%(=15/15)であった。スレッド数=1,4の場合、最適解探索率は93%=(14/15)であった。

## 2.7 まとめ

以下をマシン実験で確認した。

(1) EAXのTSPLIB最短経路長探索率はスレッド数に依存せず、ほぼ100%であった。これはEAXのA-Bサイクルによる集団多様性維持機能による。

(2)スレッド数=8の時にTSPLIB最短経路長を探索する時間が最も少ないと。スレッド数がそれよりも多くても少なくとも最適解探索時間は増えること。尚、この結果は、アムダール・グスタフソンの法則に従っていること。

## 2.8 検討課題

最適なスレッド数や最適な人口サイズがTSPで対象となる都市数の規模や複雑さ、またはコンピュータの性能(最大走行可能スレッド数)に依存するかを確認すること。

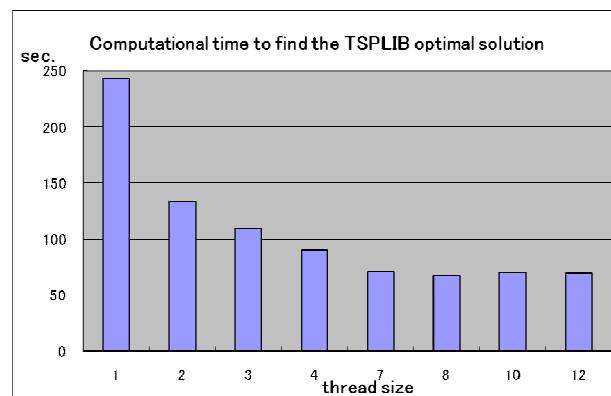


図1.並列EAXの平均最適解探索時間のマルチスレッド数による変化  
(KroA150の場合)

## 3. EAX並列処理方式

EAXの並列処理では集団を複数のプロセッサに分散し処理時間の短縮を図る。TSPデータの規模や複雑さが大きくなった場合、並列処理により解探索効率は向上するものの、局所最適解に陥る場合がある。局所最適解に陥らないためには集団の多様性を確保しつつ解を収束させる必要がある。本検討では、移民方式ならびにその階層化によりそれを実現することとした。

### 3.1 移民方式

解探索効率化のため、個体群を部分集合に分割し分割した単位で個体群を並列に処理する。分割した個体群の集合と処理を島と呼ぶ。各島で生成管理する個体群の多様性を確保するため、生成した個体群を島間で移動させる以下の二方式を検討した。

#### 〈方式1〉島間結婚方式(図2)

島内の個体を自分の島の個体または他の島の個体と遺伝子交叉させ次世代の個体を生成する。当移民方式は直接的な移民ではなく間接的な移民である。点線は個体群の結婚関係を示す。丸は島(Island)を示す。

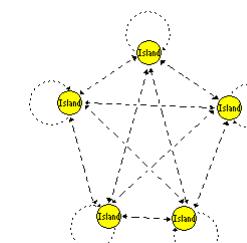


図2.島間結婚方式

#### 〈方式2〉島間での優れた子孫の入れ替え方式(島モデル)(図3)

島内の個体間で遺伝子交叉を行い次世代の個体を生成する。各島で生成した個体のうち経路長の短い優れた個体群を島間で移動する。実線は優れた個体群の移動を示す。点線は個体群の結婚関係を示す。

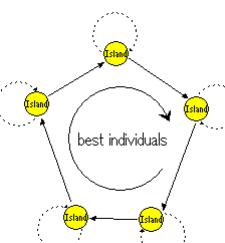


図3.島間での優れた子孫の入れ替え方式

### 3.2 Family 構成方式

各 Family 系列の結果生成した個体群の中から優れた個体を併合したり移民したりすることにより集団の多様性を確保する。

#### Family 構成方式

(1) 第一階層・・・初期データが異なる最適解探索プロセスとそれに使用した資源を Family と呼ぶ。Family 每に最適解の近似解(暫定解)を求める。本検討では、各 Family を疎結合マルチプロセッサ上の Java マルチスレッド SPMD 並列処理プログラムで実現した。当並列処理プログラムは EAX を各スレッドで並列処理させる。

(2) 第  $i$  階層 ( $i \geq 2$ ) ・・・第  $i-1$  階層の各 Family 系列の結果生成した個体群を併合することにより集団の多様性を確保する。その際、解探索効率を向上させるため優秀な個体群を選択し、併合したり移民したりする。これ等のファイルを入力として EAX 並列処理プログラムを再走行させる。優れた個体群を併合する方式を Family 併合方式(図 4)、優れた個体群を移民する方式を Family 間移民方式(図 5)と呼ぶ。

## 4. 実験

### 4.1 テストデータ

783 都市問題、rat783。TSPLIB に登録されている最短経路長(TSPLIB 最短経路長)は 8806。

### 4.2 テスト環境

Windows7、Dell PrecisionT5500、Intel(R) Xeon (R) CPU E5645、2.40GHz 2.39GHz, 8.0 GB RAM、6 コア 12 スレッドマシン

### 4.3 多様性を図る尺度

多様性(diversity)は、集団を構成する個体の多様性を図る尺度であり、本検討では次のように定義した。当尺度は TDGA における枝情報の多様性を測る尺度(Entropy)を簡易化した尺度である。

1-最短経路長/平均経路長

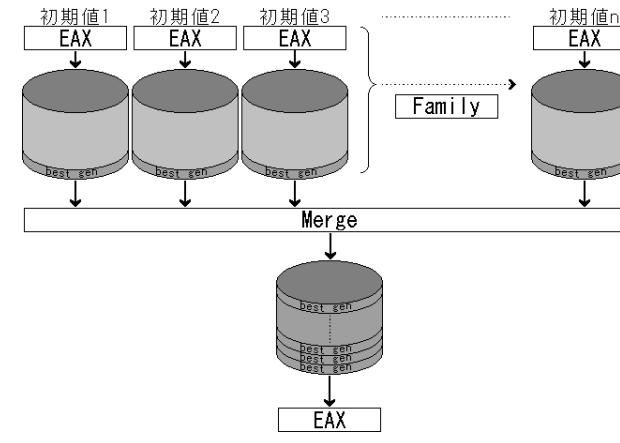


図 4. Family 併合方式

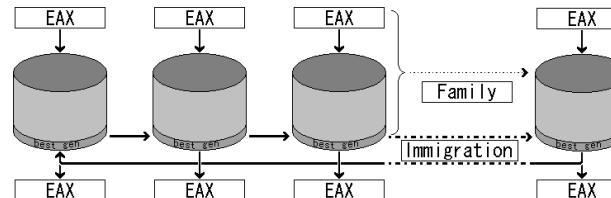


図 5. Family 間移民方式

### 4.4 島間結婚方式 × Family 併合方式の測定結果

#### (1) 測定条件

##### (a) 第 1 段の測定パラメータ

seed-id を 1 から 15 まで変化させた 15 回の Family 構成で島間での結婚方式の実験を行った。集団サイズ=300。スレッド数=12。分割集団サイズ=25。その他の EAX の主要な起動パラメータは NCROSS=0、2-opt,3-opt 法なし。親の選択方法は family 每に「親 A は全ての現世代の親、親 B は family をまたがってランダムに選択する」である。

##### (b) 第 2 段の測定パラメータ (Family 併合方式)

第一段の結果の統合ファイルを入力とし、乱数種の異なる独立な試験を 15 回行なった。乱数種が異なる場合結婚相手が異なってくる。集団サイズ、スレッド数以外は第 1 段と同じ走行パラメータ(島間結婚方式)である。

## (2) 測定結果 (表 2)

表 2 に実行結果を示す。表 2 の (第 1 段 phase1) では全 family 内における最短経路長(minimum best length) (必ずしも TSPLIB 最短経路長ではない)、family 毎に見つけた最短経路長の平均値(mean best length)、最短経路探索時間の平均値(comp. time(sec.))、最短時間探索世代番号の平均値(best gen.)、全 family 内のジョブ開始(div. (start))終了(div. of all ind. (end))時の集団多様性の平均値、第二段階に引き渡される最良個体群のみの多様性(div. of sup. ind. (end))、TSPLIB 最短経路長探索数(cumulative number of optimal)を示している。表 2 の (第 2 段 phase2) では乱数種をかえた 15 個の全試験に對して同様の情報を示している。なお、第二段階に引き渡される最良個体群は同じ経路長で異なる巡回路を持つ個体群である。

併合後の集団サイズは 2131、スレッド数は 86 となった。

この表から、Family 系列毎の最終世代で EAX が生成した次善解の div. はそれ等を混在させることにより、0.0000 から 0.0218 に向上したことが読み取れる。これにより最適解探索率を 100%(=15/15)に向上できた。

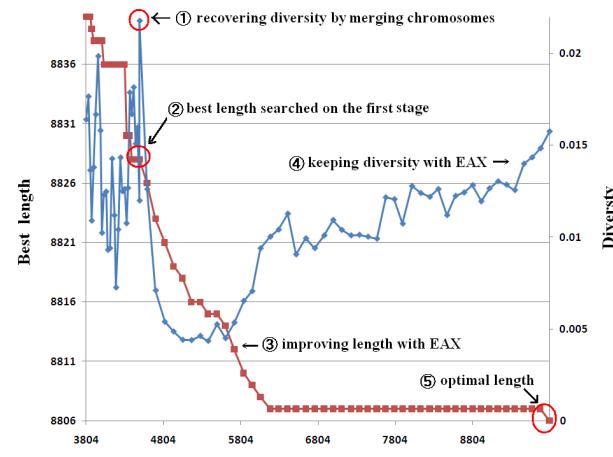


図 6. 最適解探索プロセス

## 4. 5 島間結婚方式 × Family 間移民方式

### (1) 測定パラメータ

(a) 第 1 段・・「島間結婚方式」

表 2 rat783 による並列 EAX  
(島間結婚方式 × Family 併合方式) の実験結果

phase	minimum best length	mean best length	comp. time (sec.)	best gen	div. (start)	div. of all ind. (end)	div. of sup. ind. (end)	cumulative number of optimal
1	8828	8864	4399	196	0.0466	0.0120	0.0000	0/15
2	8806	8806	5525	49	0.0218	0.0126	15/15	

seed-id を 1 から 15 まで変化させた 15 個の Family 系列で実験を行った。各系列での測定パラメータは、島間結婚方式 × Family 併合方式の第 1 段の測定パラメータと同じである。

(b) 第 2 段・・「Family 間移民方式」

Family 系列毎に生成した個体群のうち優秀な個体群を各系列間で移民する。その後、第 1 段と同じパラメータ (島間結婚方式) で EAX を再走行させた。最適解を見つけるまで、第 2 段の処理を第 n 段まで繰り返す。

(c) 最適解が見つかるまで (b) を繰り返す。

### (2) 測定結果 (表 3)

表 3 では、各段階 (フェーズ) で見つけた全 family 内における最短経路長(minimum best length)、family 毎に見つけた最短経路長の平均値 (mean best length)、最短経路探索時間の平均値 (comp. time(sec.))、最短経路探索世代番号の平均値(best gen.)、全 family 内におけるジョブ開始(start)終了(end)時の集団多様性の平均値 (div.)、ファミリー数 (the number of families)、集団サイズ数(pop size)、その世代までに発見された最適解の数(cumulative number of optimal)を示している。第 1 ~ 17 段(各 phase)の Family 移民により最終的に最適解探索率を 60% = 9/15、最適解探索平均計算時間はフェーズ毎に 32580 秒であった。

## 4. 6 島間での優れた子孫の入れ替え方式 × Family 併合方式の測定結果

### (1) 測定条件

EAX 内での移民方式を島間結婚方式から島間での優れた子孫の入れ替え方式に変更して測定する。その他のパラメータは島間結婚方式 × Family 併合方式のものと同じである。なお、移民率は 0.3 と固定して実験した。

(a) 第 1 段の測定パラメータ

seed-id を 1 から 15 まで変化させた 15 回の Family 構成で島間での優れた子孫の入れ替え方式の実験を行った。集団サイズ = 300、スレッド数 = 12、分割集団サイズ = 25。その他の EAX の主な起動パラメータは NCROSS = 0、2-opt, 3-opt 法なし。親の選択方法は

表3 rat783による並列EAX  
(島間結婚方式×Family間移民方式)の実験結果

phase	minimum best length	mean best length	comp time (sec)	best gen	div. (start)	div. (end)	the number of families	pop size	cumulative number of optimal
1	8831	8865	3112	148	0.0466	0.0196	15	300	0/15
2	8817	8841	4157	139	0.0098	0.0155	15	366	0/15
3	8809	8827	5254	186	0.0137	0.0117	15	337	0/15
4	8806	8819	6827	153	0.0144	0.0087	15	671	1/15
5	8806	8815	162	11	0.0119	0.0085	14	241	3/15
6	8806	8812	58	6	0.0134	0.0109	12	205	6/15
7	8806	8813	5	1	0.0022	0.0014	9	203	7/15
8	8812	8813	758	32	0.0077	0.0076	8	264	7/15
9	8810	8812	2762	82	0.0045	0.0155	8	480	7/15
10	8809	8812	1367	52	0.0163	0.016	8	280	7/15
11	8808	8811	5105	132	0.0144	0.0176	8	502	7/15
12	8807	8810	20	1	0.0104	0.0168	8	320	7/15
13	8807	8809	527	19	0.0105	0.0137	8	265	7/15
14	8807	8808	11	1	0.0114	0.0108	8	239	7/15
15	8806	8807	2444	86	0.0214	0.0184	8	326	8/15
16	8807	8807	2	0	0.0087	0.0105	7	164	8/15
17	8806	8807	9	1	0.0301	0.0138	7	133	9/15

family毎に「親Aは全ての現世代の親、親Bはfamilyをまたがってランダムに選択する」である。

#### (b) 第2段の測定パラメータ

第一段の結果の統合ファイルを入力とし、乱数種の異なる独立な試験を15回行なった。集団サイズ、スレッド数以外は第1段と同じ走行パラメータ(島間での優れた子孫の入れ替え方式)である。(Family併合方式)

#### (2) 測定結果(表4)

各フィールドの意味は表2と同様である。

併合後の集団サイズは2163、スレッド数は87となった。

この表から、Family系列毎の最終世代でEAXが生成した次善解のdiv.はそれ等を混在させることにより、0.0000から0.0090に向上了ることが読み取れる。これにより最適解探索率を60%(=9/15)に向上了。

### 4.7 島間での優れた子孫の入れ替え方式×Family間移民方式の測定結果

#### (1) 測定条件

EAX内での移民方式を島間結婚方式から島間での優れた子孫の入れ替え方式に変

更した。その他のパラメータは島間結婚方式×Family間移民方式のものと同じである。なお、移民率は0.3とする。

表4 rat783による並列EAX  
(島間での優れた子孫の入れ替え方式×Family併合方式)の実験結果

phase	minimum best length	mean best length	comp time (sec)	best gen	div. (start)	div. of all ind. (end)	div. of sup ind. (end)	cumulative number of optimal
1	8827	8854	3083	151	0.0429	0.0147	0.0000	0/15
2	8806	8807	7083	65	0.0090	0.0112	△	9/15

表5 rat783による並列EAX  
(島間での優れた子孫の入れ替え方式×Family間移民方式)の実験結果

phase	minimum best length	mean best length	comp time (sec)	best gen	div. (start)	div. (end)	the number of families	pop size	cumulative number of optimal
1	8827	8855	3753	179	0.0598	0.0108	15	300	0/15
2	8813	8829	3782	143	0.0128	0.011	15	319	0/15
3	8809	8820	175	8	0.025	0.0189	15	299	0/15
4	8808	8814	2713	103	0.0172	0.0151	15	305	0/15
5	8807	8812	125	9	0.0159	0.0053	15	210	0/15
6	8806	8810	3331	157	0.0143	0.0099	15	232	2/15
7	8807	8809	725	27	0.0145	0.0134	13	333	2/15
8	8807	8808	1865	66	0.0114	0.0132	13	280	2/15
9	8806	8808	294	12	0.0162	0.0116	13	334	3/15
10	8806	8808	140	9	0.0113	0.0078	12	194	4/15
11	8806	8808	183	12	0.0168	0.0109	11	193	5/15
12	8806	8808	1497	60	0.0144	0.0138	10	247	8/15
13	8807	8808	848	35	0.0123	0.0113	7	201	8/15
14	8807	8808	65	4	0.0081	0.0145	7	207	8/15
15	8807	8808	3	1	0.0143	0.0138	7	214	8/15
16	8806	8807	615	30	0.019	0.0112	7	217	9/15
17	8807	8808	1	0	0.0178	0.0116	6	168	9/15

#### (a) 第1段・・「島間での優れた子孫の入れ替え方式」

seed-idを1から15まで変化させた15個のFamily系列で実験を行った。各系列での測定パラメータは、島間での優れた子孫の入れ替え方式×Family併合方式の第1段の測定パラメータと同じである。

#### (b) 第2段・・「Family間移民方式」

Family 系列毎に生成した個体群のうち優秀な個体群を各系列間で移民する。その後、第 1 段と同じパラメータ（島間での優れた子孫の入れ替え方式）で EAX を再走行させた。最適解を見つけるまで、第 2 段の処理を第 n 段まで繰り返す。

#### (2) 測定結果（表 5）

各フィールドの意味は表 3 と同様である。第 1~17 段(各種 phase)の Family 移民では最適解探索率は 60% = 9/15、最適解探索平均計算時間は 20115 秒であった。

### 5. まとめ

rat783 の実験の結果、島間結婚方式×Family 併合方式で 100%(15/15)、島間結婚方式×Family 間移民方式で 60%(9/15)の最短経路長(=8806)を探索できた。また、島間での優れた子孫の入れ替え方式×Family 併合方式で 60%(9/15)、島間での優れた子孫の入れ替え方式×Family 間移民方式で 60%(9/15)探索できた。

今後の課題として Family 間移民方式の詳細検討、ならびに実験空間の拡張を行う。

### 参考文献

- [1] 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, Vol.15 No.5, 1998 年, pp.848~859.
- [2] R. Takahashi, A Hybrid Method of Genetic Algorithms and Ant Colony Optimization to Solve the Traveling Salesman Problem, Proceedings of ICMLA2009, IEEE computer society, pp.81~88.
- [3] E. Cantu-Paz, Efficient and Accurate Parallel Genetic Algorithm, Kluwer Academic Publishers, 2001.