

集中多段交叉 (CMX) の定量的評価手法の研究

Verification of Centralized Multiple Crossover through Diversity Measurement to Solve the TSP

久我 元士†
Motoshi Kuga

高橋 良英‡
Ryoei Takahashi

1. はじめに

本研究の目的は、遺伝的アルゴリズムの並列処理[1]において最適解探索率と集団の多様性の関係を実験により明らかにすることである。実験では、島間で移民を行わない集中多段交叉 (Centralized Multiple Crossover: CMX) [2]と島間で移民を行う並列処理方式 (Family 併合法) [3]を集団の多様性維持機能の観点から定量的に比較評価した。その際、集団の多様性を熱力学的遺伝アルゴリズム (Thermo Dynamical Genetic Algorithm: TDGA) [4]のエントロピーで定量的に測定した。本報告では特に、巡回セールスマン問題 (Traveling Salesman Problem: TSP) を解く枝組み立て交叉 (Edge Assembly Crossover: EAX) [5]を利用して、両方式の有効性を実験により定量的に検証した。

島モデルによる遺伝的アルゴリズムの並列化では、集団 (個体群) を島と呼ばれる小集団に分割し、分割した単位で個体に対する「選択、交叉、突然変異」という一連の遺伝子操作を並列化する。しかし、分割すると島を構成する個体数が少なくなるので、遺伝子交叉によって生成される個体群の遺伝子構造パターンに限られ、また世代が経ると突然変異操作の効果も顕れず、局所最適解に陥る場合がある。局所最適解に陥らせないためには、島を構成する個体群の多様性を確保する必要がある。多様性とは、様々な遺伝子構造を持った個体が集団を構成していることである。このため島モデルによる並列処理では、島を構成する個体群の多様性を確保するため、島をリング状に繋ぎ、世代終了毎に隣接する島間で経路長の短い優秀な個体群を移民する方式を採用するのが一般的である。

しかし移民回数が増えて、各島で解探索の収斂が始まると、島間で移民する優秀な個体群が似通った構造を持ち、結果的に島間の個体群の多様性が確保できず、ほとんどの島で同じ局所最適解に陥る場合がある。我々もこの現象を実験で経験している。このため CMX では、島間で個体群の多様性を確保するために島間で移民をしない並列処理方式としている。そこではまず、島間で移民をしない遺伝的アルゴリズムの並列処理を行う。次に、各島で生成した優れた個体を併合し、併合した個体群 A の中から個体選択して遺伝子交叉を行い、更に優れた個体群 B を生成する。そして、生成した優れた個体群 B を、「島間移民をしない遺伝的アルゴリズムの並列処理」を行う島に戻す。論文[2]では、島間で移民をする島モデルによる並列処理方式に比較して CMX が優れていることを、最適解探索率を比較することにより検証している。しかし、集団の多様性を定量的に測定して、CMX の有効性を検証していない。本論文では集

団の多様性を TDGA のエントロピーで定量的に測定して並列処理方式を評価する。

2. TSP と EAX

2.1 TSP

TSP は、あるセールスマンが 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短な経路 (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の1つである。上りと下りで都市間の距離が同じ対称型 TSP の時、都市数 n とすると巡回路順が異なる巡回路の組み合わせの数は $(n-1)!/2$ 通りとなる。従って、都市数 n が多くなった場合、網羅的に全ての巡回路長を横並びに比較し最適解を求める手法では、解くのに都市数に関する指数オーダー以上の時間がかかり、計算機でも実効上最適解を探索することが困難となる。このため、計算の複雑さの観点から、TSP はその問題を解くことが困難な NP 困難なクラスに属する問題と理解されている [6]。TSP は運搬経路計画、基盤配線敷設計画、VLSI 設計等に応用されている [7]。

2.2 EAX

TSP の最適解の近似を効率的に得る確率的な手法として遺伝的アルゴリズム (Genetic Algorithm: GA) [8] が有効な手法であることはよく知られている。GA は、生物の進化を模した最適化モデルである。GA では最適化問題の解候補を生物進化の対象である個体と考え、個体を染色体 (遺伝子列) で表現する。ランダムに生成された複数の個体の中から適応度等によって確率的に二つの個体を選択し、選択した個体への遺伝子交叉、突然変異と呼ばれる遺伝子操作を施して次世代の個体を生成する。この遺伝子操作を何回も繰り返すことにより、最適解を探索する方法である。TSP を解く GA としてこれまで多くの交叉手法が提案されてきた。その中でも EAX が、最適解探索率、最適解探索時間の両面において最も優れた交叉手法であることが知られている。EAX は EXX (Edge Exchange Crossover) [9] を改良した手法であり、親 A と親 B の同数の枝を交換して幾つかの部分巡回路群を構成し、構成した部分巡回路を巡回路長が最小になるように逐次繋ぎ合わせることで巡回路を再構成する方法である。

これまでの我々の小規模実験でも、集団を構成する集団サイズを大きくすれば、EAX が最適解を探索できる確率は高くなることがわかっている [10]。規模が大きく複雑な TSP を解く場合は、集団サイズを更に大きくし探索空間を広げる必要がある。しかし集団サイズを増加させると計算量やメモリ量が増加し問題を解くことが困難となる。このため、集団を複数の島に分割し、分割した単位で遺伝処理を並列処理する島モデルによる並列 GA を研究することとした。

† 八戸工業大学大学院 電子電気・情報工学専攻 (修士課程), HIT

‡ 八戸工業大学 教授, HIT

3. 並列化方式における集団多様性確保方式の検討

3.1 集中多段交叉 CMX 方式

島間移民方式による GA の並列化においては、何回かの移民を施すと移民する個体群が似通った構造を持ち、結果として遺伝子交叉し生成される個体群の構造が似通い、ほとんどの島で同一の局所最適解に陥るという問題が生じる。このため、CMX による並列処理方式では、各島で生成される個体群の遺伝子構造が異なったものになるよう、島間で移民をしない並列処理を行なっている。そして、各島で生成される個体群が局所最適解に陥らぬよう、次の段階では、各島で生成した優れた個体群を併合して集団の多様性を確保する。それから、その併合ファイル（交叉島）内の個体間で遺伝子交叉を行ない、交叉で生成した個体群を、並列 GA 処理を行う各島に戻す方式としている。

CMX の並列処理方式を Fig. 1 に、そのアルゴリズムを Fig. 2 に示す。Fig. 2 にて、①～②を第 1 ステージ、③～⑤を第 2 ステージと呼ぶ。

(1) CMX のアルゴリズム

Fig. 2 に示したアルゴリズムを以下に説明する。以下の①～⑤は Fig. 2 の①～⑤に対応している。

CMX による並列 GA 制御アルゴリズム

- ② 期個体群を生成する。
- ② Family 系列毎に島間移民を行わない並列 GA (isolated distributed GA: i-DGA) を行う。
- ③ i-DGA で生成した個体群を新たな島（交叉島）に集める。
- ④ 交叉島において i-DGA を行う。
- ⑤④で最適解が見つからなければ、生成された個体群のうち適応度の高い優秀な個体群 α を、②の各々の Family 系列に戻す。その際、②の Family 系列で生成された個体群のうち適応度の低い個体群を上記優秀な個体群 α と入れ替える。

<留意点>

本論文での CMX は論文[2]の CMX に以下の機能改良や機能の明確化を図っている。

① Family 系列の概念の追加

Family 系列とは、一様乱数系列に対応した並列 GA の処理プロセスとそれが使用する資源のことである。Family 系列については、論文[2]の CMX では、明記していない。

一様乱数の初期値 (seed-id) によって一様乱数系列は定

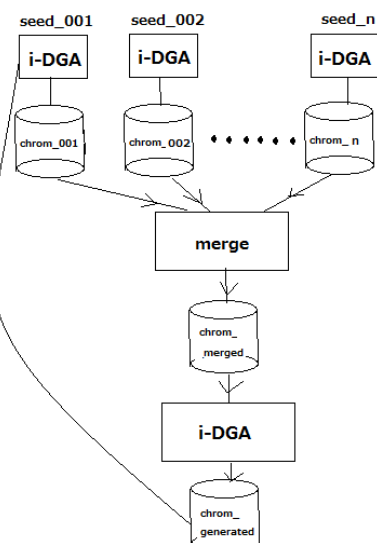


Fig. 1. A schematic of parallel GA with CMX.

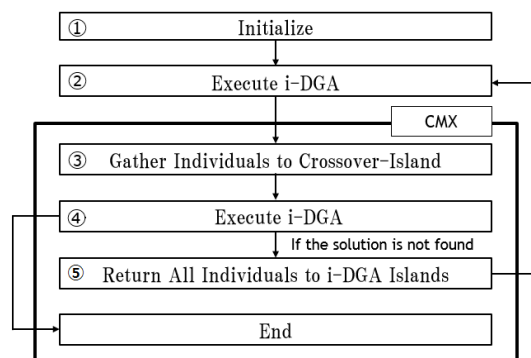


Fig. 2. Control flow of parallel GA through CMX.

まる。一様乱数系列が異なると、GA で生成する初期巡回路群は異なる。初期巡回路群が異なると並列 GA が生成する暫定解（局所解）群の遺伝子構造は異なる傾向がある。この現象を本実験でも定量的に確認している。

② 交叉島における交叉個体の選択法

論文[2]の CMX では「交叉島では交叉のみを連続して行い、最後に島間移民を行なう並列 GA (DGA) を行う」とあるが、親 A と交叉するもう一方の親 B の選択法については明記していない。本検討における i-DGA では 3.1 節 (2) に述べる「島間交叉方式」により交叉する二個体を選択している。これにより、島内の個体群の多様性を確保している。尚、本検討では、第 1 ステージと第 2 ステージの並列処理の整合性を図るため、交叉島においても i-DGA を行う方式とした。

③ 2-opt 法の適用について

論文[2]では初期個体群生成時に 2-opt 法を適用して初期値を改善し、その後 i-DGA を適用している。本実験では初期個体群生成時、2-opt 法を適用していない。本検討では 2-opt 法を、3.1 節 (4) で述べる通り、遺伝子交叉で生成した各個体への「突然変異」を実現する手段として使用している。

④ 本論文では論文[2]の「分散 GA」を「並列 GA」という用語で一本化した。「分散 GA」は「ネットワークを介した複数のコンピュータで GA 処理を分散化すること」であり、並列 GA を実現する一手法と理解できるためである。

(2) 島内個体群の多様性を確保する方式

論文[2]の CMX では、i-DGA において各島で生成管理する個体群の多様性を確保する方式については論じていない。

本検討では、各島で生成管理する個体群の多様性を確保するため、以下の機能を論文[2]の CMX に追加した。

島間交叉方式 (Fig. 3)

島内の個体を自分の島の個体または他の島の個体と任意に遺伝子交叉させ次世代の個体を生成する。

本方式は、島外の個体を交叉対象として選択可能とすることで遺伝子の交叉パターンを増やし、生成する個体の多様性の維持を複数の島全体で図ることをねらいとしている。Fig. 3 にて点線は可能な個体間の交叉関係を示す。丸は島 (Island) を示す。親選択では親 A は自島の全ての個体、親 B は任意の島の任意の個体を対象として一様乱数によりランダムに選択する。当選択法は「Heterogeneous Selection Evolutionary Algorithm: HeSEA [11]」の個体選択法を簡略化した手法である。

<留意点>

HeSEA の個体選択法では、二つの巡回路について、巡回路を構成する枝（任意の都市に関する隣接都市情報）が

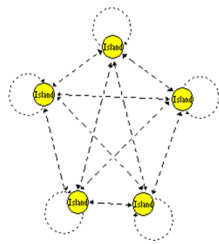


Fig. 3. Crossover operation between Islands.

一致している割合を類似度として定義する。現世代の任意の親 A に対して集団を構成する全ての個体との類似度を測定し平均類似度を求める。そして、平均類似度より低い類似度を持つ個体を、親 A の交叉対象となる親 B としてランダムに選択する方式としている。

(3) 島間の多様性を確保する方式

CMX では島間で個体を移民しない遺伝的アルゴリズムの並列処理方式を実現することにより、島間の多様性を確保している。更に (1) の留意点①で述べたように、複数の Family 系列により、適応度の高い遺伝子構造の異なる個体群を生成することで、Family 系列で生成される個体群間の多様性を確保している。

(4) 島間交叉方式による並列遺伝的アルゴリズム

島間交叉方式による並列遺伝的アルゴリズムの手続きを Fig. 4 に示す。Fig. 4 にて太枠の「選択」と「同期をとる」と「島間で同期をとって終了」の三つは並列化のために追加した遺伝的アルゴリズムの手続きである。「選択」では親 A と交叉する親 B を他島からランダムに選択し、島間で親 B の情報を授受する。「同期をとる」では、世代交代 (各島で生成した次世代の個体群を現世代の個体群と入れ替える) のために、島間で同期をとる。「島間で同期をとって終了」では、ある島で最適解が見つかった場合や、島の多様性がなくなった場合に、その情報を他島に伝え、同期をとって全ての島の処理を終了する。上記にて「選択」は C 言語 MPI では MPI_Allgather 命令と MPI_Send 命令と MPI_Receive 命令で実現する。Java 言語マルチスレッドでは島間でメモリが共用のため、他島の情報も自島から参照可能であり、「選択」は親 B を検索する範囲を全島に広げることで実現する。また、「同期をとる」と「島間で同期をとって終了」は C 言語 MPI では MPI_Allgather 命令、Java 言語マルチスレッドでは join 命令で実現する。

本検討では突然変異を 2-opt 法と 3-opt 法 [12] で実現している。突然変異の結果、「探索経路長」が改良されな

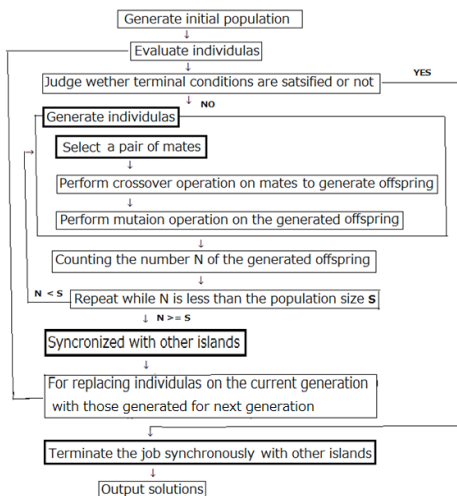


Fig. 4. Procedures of parallel GA with crossover operation between Islands

い場合は、突然変異で生成した個体を次世代の個体としていない。突然変異機能を使用するか否かを GA の起動パラメータで選択可能としている。

3.2 Family 併合方式 (Family Merging Crossover: FMX)

島内の多様性を確保するため、島間移民のある並列処理系列を検討した。Fig. 5 に FMX 処理方式, Fig. 6 に FMX のアルゴリズムを示す。Fig. 6 にて、①~②を第 1 ステージ、③~④を第 2 ステージと呼ぶ。

(1) FMX のアルゴリズム

Fig. 6 に示したアルゴリズムを以下に説明する。以下の①~④は Fig. 6 のそれに対応している。

FMX の並列 GA 制御アルゴリズム

- ① 期個体群を生成する。
- ② Family 系列毎に移民を行う並列 GA を行ない、暫定解を生成する。
- ③ Family 系列毎に生成した優れた個体を併合させ、新たな個体群を生成する。
- ④ 併合した個体群を入力として、移民を行う並列 GA を行う。最適解が探索できなければ、探索した暫定解を②で探索した各 Family 系列の暫定解群に混在させる。各 Family は混在したファイルを入力として並列 GA を続行する。

(2) 島内個体群の多様性を確保する方式

島間移民方式

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。本方式では、交叉対象の個体は全て自身の島内である

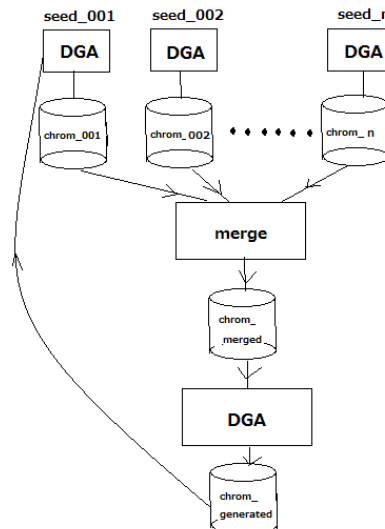


Fig. 5. A schematic of parallel GA with FMX.

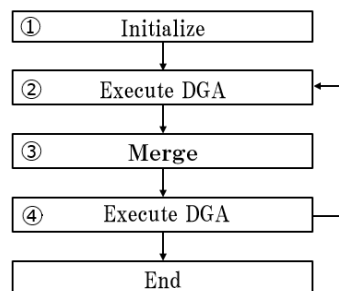


Fig. 6. Control flow of parallel GA through FMX.

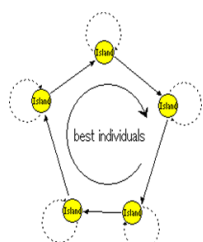


Fig. 7. Migration to other Islands connected with ring structure.

ため、島毎に独自の個体群を生み出すことができる。しかし、選択できる個体数は少ないので、島内の個体の多様性が確保できず局所最適解に陥る可能性が高い。この問題を解決するため、本方式は、世代交代の度、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民し、各島の多様性を維持する。この方式を Fig. 7 に示す。Fig. 7 にて実線は優れた個体群の移民を示す。点線は個体間の交叉関係を示す。

(3) 島間の多様性を確保する方式

島間移民方式による遺伝的アルゴリズムの並列処理では進化対象遺伝子データを複数の島に分割し、分割した単位で遺伝子操作を並列化する。この並列処理単位が Family である。3.1 節で述べたように各 Family 系列内では、島間で何回かの移民を施すと、移民する優秀な個体群の構造が似通い、結果としてほとんどの島で生成される個体群が同一となる傾向がある。この問題を解決するため、初期巡回回路群が異なる複数の Family 系列から生成した暫定解群の中から優秀な個体群を選択併合し、これを入力として解探索を続行させる並列処理方式とした。これにより Family 系列間の多様性を維持し、結果として解探索効率を向上させる並列処理方式を実現した。

(4) 島間移民方式による並列遺伝的アルゴリズム

島間移民方式による並列遺伝的アルゴリズムの手続きを Fig. 8 に示す。Fig. 8 にて太枠の「島間で同期をとって終了」と「同期をとる」と「島間移民処理」の三つは並列化のために遺伝的アルゴリズムに追加した処理である。「島間移民処理」における遺伝子情報の送受信を、C 言語 MPI では、MPI_Sendreceive 命令で実現する。Java マルチスレッドでは、遺伝子情報のメモリ更新エリアを自島から全島に拡げることで実現する。「島間で同期をとって終了」と「同期をとる」の実現法は、島間交叉方式による並列遺伝

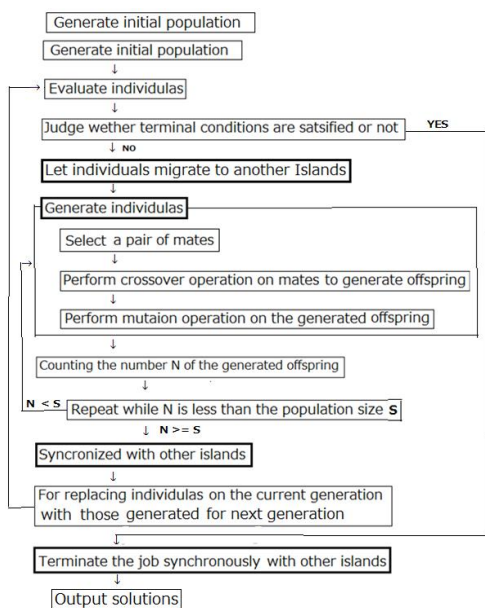


Fig.8. Procedures of parallel GA with migration between Islands.

的アルゴリズムの手続きにおける場合と同じである。

3.3. CMX と FMX の違い

CMX による並列処理方式では、島間の多様性を確保するため移民をしない並列処理方式とし、次のステップで各島で生成した優れた個体群を選択・併合しそれを入力として解を並列 GA で再探索する方式としている。一方、FMX による並列処理方式では、島内の多様性を確保するため島間で移民をする並列処理方式としている。そして、複数の Family 系列から生成した暫定解群の中から優秀な個体群を選択併合し Family 間の多様性を確保している。それ以外の並列 GA 制御機能は両方式で同じである。

4. 集団の多様性の定量的測定法

本検討では「TDGA における集団の多様性を測る尺度」で並列 GA 方式の有効性を定量的に検証した。TDGA は、集団の多様性を定量的に測定し、測定結果を次世代の個体生成に反映する方法である。TDGA では、集団の多様性をエントロピー H として明確に評価し、自由エネルギー F が最小となるように個体群を形成する。自由エネルギー F は、「平均エネルギー <E> - 温度パラメータ T × エントロピー H」で定義される。TSP を解く GA では、平均エネルギー <E> を巡回回路長相当情報、すなわち適応度の逆数相当情報と解釈する。自由エネルギー F を最小化させることは集団の平均エネルギー <E> を最小化しかつ、集団のエントロピー H を最大化させることである。TSP を解く GA では、エントロピー H を次式で解釈する。

$$H = \sum_{i=1}^N H_i \quad \dots \dots \dots (1)$$

$$H_i = - \sum_{j=1}^N P_{ij} \log(P_{ij}) \quad \dots \dots \dots (2)$$

$$P_{ij} = \frac{n_{ij}}{2N_p} \quad \dots \dots \dots (3)$$

ここで、H は集団のエントロピー、H_i は都市 i の隣接都市がどの都市であるか、その確率分布が示すエントロピー、n_{ij} は都市 i の隣接都市が j である個体の数、N は都市数、N_p は個体数である (i=1, ..., N)。

TDGA の尺度とその方式の有効性は、論文 [13] と [14] で論じている。

5. 遺伝的アルゴリズム並列処理の実装法の検討

5.1 Java 言語のマルチスレッド機能による遺伝的アルゴリズムの実装 (Fig. 9)

Java 言語のマルチスレッド機能 [15] を利用して、遺伝的アルゴリズムを並列化する方法を検討した。Java マルチスレッドでは並列処理を密結合マルチプロセッサで実現する。Java 言語では論理的な並列処理単位をスレッドと呼ぶ。スレッド生成時に、プロセッサを各スレッドに割り当て、

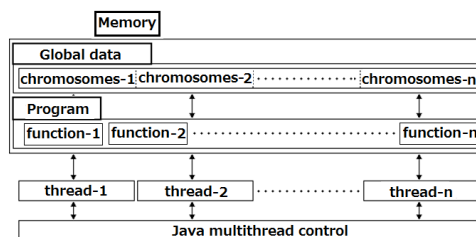


Fig. 9. Implementation of parallel GA with Java multi-thread.

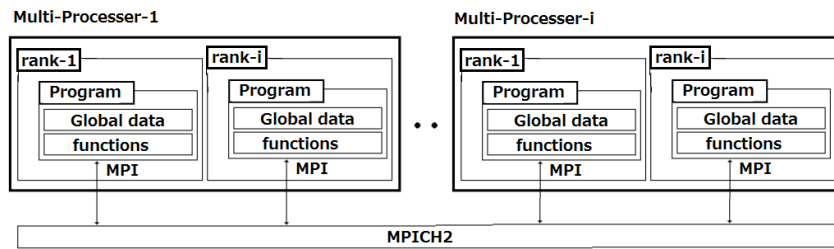


Fig. 10. Implementation of parallel GA with C MPI (MPICH2).

割り当てたプロセッサ上で並列処理を実現する。遺伝的アルゴリズムの並列処理を実現する方式として、集団を構成する個体群を幾つかの島に分割し、分割した島単位で個体選択、遺伝子交叉、突然変異を行う並列処理方式を検討した。各島の処理をJavaの各スレッドで実現した。複数のスレッドについて次世代の個体群の生成処理が終わったことを、子スレッド間で同期をとる命令 `join` で確認する等、親スレッドは集団全体の世代交代処理を行う方式とした。密結合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する。スレッドの実行順序に依存せず、実験の再現結果を同じとするため、各スレッド毎に異なる「一様乱数の発行系列」を割り当て、それに基づきGAが解探索を行う方式とした。

5.2 C 言語 MPI による遺伝的アルゴリズムの実装 (Fig. 10)

MPIは「ランク (並列処理単位) 間のデータ送受信機能」など並列処理支援機能を標準化したライブラリである[16]。MPIは疎結合マルチプロセッサを前提にしており、そこでは複数のランクがそれぞれ異なるプロセッサ上、異なるメモリ上で同一のプログラムを走行させ、情報を交換し合いながら同期をとって並列処理を進める。この並列処理プログラミング法はSPMDと呼ばれる。C関数がMPI命令を用いることでランク間での情報授受を実現する。「同期処理をとって終了」はMPI_Allgather命令で、「島間移民処理」等の遺伝子情報の送受信はMPI_Sendreceive命令で実現する。

6. 実験

6.1 実験データ 532都市問題 att532による実験を行った。TSPLIB[17]に登録されたatt532の最短経路長(TSPLIB最適解の経路長)は27686である。

6.2 マシン環境 Widows7, Dell Precision T5500, Intel(R) Xeon (R) CPU E5645, 2.4GHz 2.39GHz, 8.0 GB RAM, 6コア12スレッドマシン (1台)

6.3 実現方式

(1) CMXの実現方式

Java言語マルチスレッド機能によりCMX方式による遺伝的アルゴリズムの並列処理を実現した。

(2) FMXの実現方式

C言語MPIによりFMX方式による遺伝的アルゴリズムの並列処理を実現した。

6.4 起動パラメータ

(1) CMX方式

- ①集団サイズ=1200 (=50個体×24スレッド)
- ②遺伝子交叉法・・・島間交叉方式 (簡易 HeSEA 法)
- ③多様性重視パラメータ・・・2-opt なし, 曾孫生成なし, 親と子のトーナメントなし
- ④スレッド数・・・24 ⑤観測世代数・・・200

(2) FMX方式

- ①集団サイズ=1200 (=300個体×4プロセス)
- ②遺伝子交叉法・・・HeSEAによる個体選択
- ③収束性重視パラメータ・・・2-opt あり, 曾孫生成あり, 親と子のトーナメントあり。曾孫生成では, 親として選択した二個体を交叉して生成した子の適応度が親の適応度より高ければ, 親と子を入れ替える。その回数を本実験では30とした。この曾孫生成機能が論文[2]における多段交叉機能にあたる。
- ④プロセス数: 4 ⑤移民率: 0.1 ⑥観測世代数・・・100

6.5 実験結果

6.5.1 CMX方式

(1) Family系列毎でのCMX実験結果 (Table 1) 乱数種を1から15に変えた独立なFamily系列による実験を行った。系列毎に, ①探索した最良経路長, ②③最良経路長を見つけるまでに要した世代番号と計算時間(秒), ④⑤⑥開始時と最良解探索時とジョブ終了時の多様性を測定した。Table 1は測定結果である。平均探索経路長=27706, 平均の開始時の多様性=4194.8, 最良経路を見つけた時の平均多様性=144.9, 終了時の平均多様性=129.3であった。

(2) Family併合島でのCMX交叉処理結果 (Table 2) 上記Family系列で生成した経路長が短い優秀な個体群の併合ファイルを入力として並列EAXに遺伝子交叉をさせた。併合ファイルの集団サイズは1997であった。スレッド数=80で実験を行った。収束性重視パラメータモード(conv. mode) (2-opt回数=10000または100)と多様性重視モード

Table 1. CMX evaluation for parallel GA families on the first stage.

family-id	best length	best gen.	time(s.)	initial diversity	diversity when best length found	final diversity
1	27703	153	24260	4194.5	136.8	108.5
2	27704	123	19631	4194.7	132	116.9
3	27704	183	29004	4194.6	160.4	155.1
4	27705	119	18834	4194.6	158.3	141.1
5	27706	131	20956	4195.5	154.4	130.4
6	27710	147	23162	4195.1	148	133.1
7	27707	109	17343	4194.4	147.5	118.6
8	27704	144	22938	4195	164.9	144.5
9	27703	194	31019	4195.1	97.3	114.8
10	27703	119	19003	4195.1	158.4	123.1
11	27716	139	22210	4194.9	160.6	146.5
12	27703	130	20576	4195.1	138.5	116.5
13	27706	151	24027	4194.9	142.5	130.6
14	27707	161	25613	4194.6	153.6	145.1
15	27709	143	7381	4194.5	120.4	114.5
average	27706	143.1	21730.5	4194.8	144.9	129.3

Table 2. CMX evaluation on the crossover-Island on the second stage.

GA parameters	no. of two-opt trials	no. of threads	ratio of optimum trials	best length	best gen.	time (s.)	initial diversity	diversity when best length found	final diversity
conv. mode	100	80	15/15	27686	1.7	1216.7	261.1	103.4	143.2
conv. mode	10000	80	15/15	27686	1.7	1253.6	261.1	109.2	109.2
div. mode	0	80	14/15	27686	9.1	3169.5	261.1	142.4	143.5

Table 3. FMX evaluation for parallel GA families on the first stage.

family-id	best length	best gen.	time(s.)	initial diversity	diversity when best length found	final diversity
1	27706	35	15372	3891.8	5.7	5.1
2	27703	33	14713	3892.5	44.19	4.7
3	27725	35	15368	3892.1	28.9	20.7
4	27704	34	14439	3893.8	49.6	13.1
5	27706	35	15361	3892.3	16.3	3.4
6	27707	33	14643	3893	26.4	26.4
7	27703	34	14525	3892	36.5	1.6
8	27706	33	15189	3892.2	27.5	19.8
9	27693	34	14629	3893.6	25.6	23.6
10	27699	35	15250	3892.3	12.4	0.2
11	27706	35	15281	3890.9	16.2	12.1
12	27705	32	13815	3890.8	66.8	9.3
13	27709	30	13519	3893	40.6	21.5
14	27703	34	14981	3893	7.3	3.6
15	27705	34	15560	3893.2	11.3	3.1
average	27705.3	33.7	14843.0	3898.4	27.7	11.2

Table 4. FMX evaluation on the crossover-Island on the second stage.

GA parameters	no. of two-opt trials	no. of ranks	ratio of optimum trials	best length	best gen.	time (s.)	initial diversity	diversity when best length found	final diversity
conv. mode	100	1	0/15	27693	0	0	98.8	98.8	23.8
	100	4	0/15	27693	0	0	80.2	80.2	22.9
div. mode	0	1	14/15	27686	11	683	101.9	102.1	102
	0	4	10/15	27686	19.3	329.9	80.2	73.1	75.2

(div. mode)で走行させた。乱数種が異なる15回の独立な実験の結果、全走行モードでCMXは経路長27686の最適解を最適解探索率=0.93~1(=14/15~15/15)で見つけた。

6.5.2 FMX方式

(1) Family系列毎でのFMX実験結果 (Table 3)

一様乱数種を1~15に変えた15回の独立な実験を行った。実験の結果、平均探索経路長=27705、平均の開始時の多様性=3898.4、最良経路長を見つけた時の平均多様性=27.7、終了時の平均多様性=11.2であった。

(2) Family併合島でのFMX交叉処理結果 (Table 4)

上記Family系列で生成された経路長が短い優秀な個体群を併合した。併合ファイルの集団サイズは107であった。プロセス数=4と1で、乱数種の異なる15回の独立な実験を行った。収束性重視モード(2-opt回数=100)と多様性重視のパラメータモードで並列EAXを走行させた。多様性重視モードで、経路長27686の最適解を、80%(=24/30)の探索率で見つけた。

6.5.3 考察

CMX方式はTSPLIBに登録されたatt532の最短経路長を0.98(=44/45)の確率で探索した。最適解探索に成功した理由は、複数の並列処理系列から生成された暫定解群の中から優れた個体群を選択・併合することによって、集団の多様性を拡大できたことである。実際、集団の多様性をTDGAのエントロピーで定量的に測定し、129.3から261.1に集団の多様性が拡大していることを確認した。

FMX方式についても同様に、att532の上記最適解を80%(=24/30)の確率で探索できた。最適解を探索できた理由は、複数の並列処理系列から生成された暫定解群の中から優れた個体群を選択・併合することによって、集団の多様性を拡大できたことである。実際、集団の多様性をTDGAのエントロピーで定量的に測定し、11.2から98.8または80.2に向上していることを確認した。本実験では、CMX方式の最適解探索率は98%であり、FMX方式の最適解探索率80%より高かった。その理由は前者の集団の多様性が261.1と、後者の集団の多様性98.8より大きかったためである。

7. 結論

本論文では、島間で移民をしない遺伝的アルゴリズムの並列処理方式CMXと島間で移民をする遺伝的アルゴリズムの並列処理方式FMXをプログラミング実験により定量的に比較評価した。集団の多様性を熱力学的遺伝アルゴリズムTDGAで定義されたエントロピーで測定し、「集団のエントロピーを大きくすることにより最適解探索率を向上できる」という仮説が成立することを、TSP小規模実験により確認した。今後、実験空間を拡張し、並列処理方式において上記仮説が成立することを明らかにしていく。

参考文献

- [1] E. Cantu-Paz, "Efficient and Accurate Parallel Genetic Algorithm", Kluwer Academic Publishers, 2001.
- [2] 三木 光範, 廣安 知之, 花田 良子, 水田 典伯, "エリート解の集中的な交叉メカニズムを持つ分散遺伝的アルゴリズムのTSPにおける解探索性能の検討", システム情報学会文誌, Vol16, No12, pp.607-615, 2003.
- [3] 元田 剛, 高橋良英, "Java マルチスレッドによる枝組立交叉 (EAX) の並列処理実現方式の検討", 情報処理学会第74回全国大会, 3M-7, 1-455~1-456, 2011
- [4] 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.
- [5] 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, pp.848 - 859, 1998.
- [6] M. Garey and D. Johnson, "Computers and Intractability", A Guide to the Theory of NP-Completeness, W. H Freeman and Company, 1979.
- [7] 山本芳嗣, 久保幹夫, "巡回セールスマン問題への招待", 朝倉書店, 1997.
- [8] D. E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley Publishing Company, Inc. 1989.
- [9] 前川景示, 玉置久, 喜多一, 西川禎一, "遺伝的アルゴリズムによる巡回セールスマン問題の一解法", 計測自動制御学会論文集, Vol.31, No.5, pp. 598-605, 1995.
- [10] 吉川 克哉, 高橋 良英, "ACO突然変異方式による枝組立交叉 (EAX) の性能改善~巡回セールスマン問題 (TSP) の解法~, 全国大会講演論文集 第72回全国大会, pp.2-417 - 2-418, 2010.
- [11] H.-K. Tsai, J.-M. Yang, Y.-F. Tsai, and 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.
- [12] K. Helsgaun, "An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristics", European Journal of Operational Research, 126(1), pp. 106 - 130 (2000)
- [13] R. Takahashi, "Solving the Travelling Salesman Problem through Iterative Extended Changing Crossover Operators", Proceedings of ICMLA2011, IEEE Computer Society, pp.253-258, 2011.
- [14] R. Takahashi, "Quantitative Evaluation of Iterative Extended Changing Crossover Operators to Solve the Traveling Salesman Problem", Proceedings of 2014 10th International Conference on Natural Computation (ICNC), pp. 235-244, IEEE, 2014
- [15] ジョセフ・オニール, "独習 Java ", pp.256 - 280, 翔泳社, 2001.
- [16] Peter S. Pacheco, "Parallel programming with MPI", Morgan Kaufmann Publishers, Inc., 1997.
- [17] TSPLIB95: <http://www.informatik.uni-heidelberg.de/>