

遺伝的アルゴリズムの並列化モデルの提案と 並列計算機アーキテクチャへの適用

中條 拓伯[†] 小谷 学[†] 上原 邦昭^{††}

遺伝的アルゴリズムでは、各個体間の独立性が高いので、並列化による処理の高速化が期待されている。並列化モデルとして今までに離島モデルと近傍モデルが提案されてきた。しかし、離島モデルでは遺伝子の伝播を工夫しないと初期収束による局所解に陥りやすい。また、近傍モデルでは高い適応度を持つ遺伝子の伝播がゆるやかに進み、最適に近い解が得られやすいが、共有メモリを必要とするため大規模なプロセッサ台数による並列化が困難である。そこで、我々は近傍モデルにおいて、遺伝子の伝播を一次元の隣接間のみ限定した形態の並列 GA モデルを地中海モデルとして提案する。本稿では、地中海モデルの概要について述べ、リング結合型の並列計算機とワークステーションクラスターへの適用とモデルの有効性について検証を行なう。

A Model of a Parallel Genetic Algorithm Applied to a Parallel Computer Architecture

HIRONORI NAKAJO,[†] MANABU KOTANI[†] and KUNIAKI UEHARA^{††}

Since individual in genetic algorithms is highly independent with each other, an enhancement of performance by parallel processing has been expected. There exists two major models, an island model and a neighborhood model, in parallel GA. The island model tends to fall into local maximum because of an initial convergence without contriving propagation of genes. On the other hand, since genes with high fitness slowly propagate in the neighborhood model, semi-optimal results might be able to be obtained. This model, however, needs shared memory among processing nodes and it is difficult to apply to a large-scale parallel processing system. We, here, propose a new parallel GA model called the Mediterranean Sea model which limits propagation into a single dimension in a neighborhood model. We describe a concept of the Mediterranean Sea model and evaluate its effectiveness by applying to a ring-connected parallel computer system and a workstation cluster as well.

1. はじめに

遺伝的アルゴリズム (Genetic Algorithm: GA) は J.H. Holland¹⁾ が開拓した計算理論であり、巡回セールスマン問題やナップザック問題などの組合せ最適化問題に適しているとされている。

また、プロダクションシステムなどの人工知能や、生産システムにおけるスケジュール問題、パターン認識、その他種々の工学システムなどに応用され、成果

をあげてきた。²⁾³⁾

しかしながら、より効率的な解を探索するためには、初期収束という現象が大きな問題となってきた。初期収束とは探索の初期段階で個体群の多様性が失われる現象であり、これを回避する手法がいくつか提案されている。⁴⁾⁵⁾⁶⁾

また、遺伝的アルゴリズムは多数のデータに対し、独立して並列に処理できることから、SPMD (Single Program Multiple Data) モデルに適しており、今までにもさまざまな並列化 (Parallel GA: PGA)⁷⁾⁸⁾⁹⁾¹⁰⁾ の試みが進められてきた。

その中で、個体群を分割して並列に処理を行なうアプローチとして、離島モデルと近傍モデルの二つの

[†] 神戸大学工学部情報知能工学科
Department of Computer and Systems Engineering,
Faculty of Engineering, Kobe University

^{††} 神戸大学都市安全研究センター
Research Center for Urban Safety and Security, Kobe
University

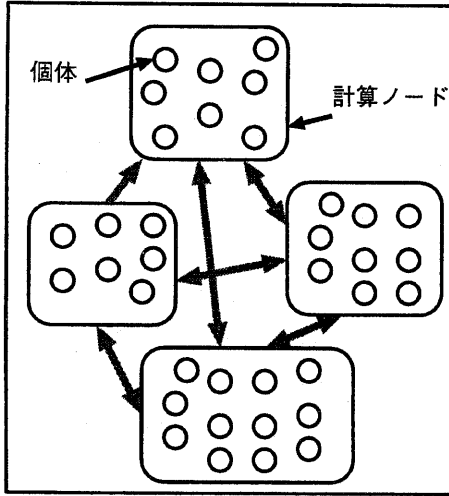


図1 離島モデル
Fig. 1 Island model of Parallel GA.

PGA モデルが提案されている¹¹⁾。これらのモデルは、並列処理による計算時間の短縮のみならず、進化の局所性を利用することにより、初期収束を回避し、個体群全体としての多様性を維持するといった点でも利点がある。以下にその概要を示す。

- 離島モデル (Island model)

いくつかの個体群に分割し、それぞれの部分個体群に対して独立に遺伝的操作を適用する方式である。

複数の計算機ノード上に、個体群の分割により生成された部分個体群を割りつける。そして、各部分個体ごとに通常の遺伝的操作を並列に適用しながら、計算機ノード間においていくつかの個体の交換を行なう。

離島モデルでは、各計算機ノードにおいて独立に遺伝的操作が行なわれるため、各々の個体群において独立性の高い進化が進められ、個体群全体としての多様性が保たれることが期待される。図1に離島モデルによる並列化の例を示す。

- 近傍モデル (Neighborhood Model)

個体群内の各個体が近傍内の限られた個体としか相互に遺伝的操作を行なわない方式である。

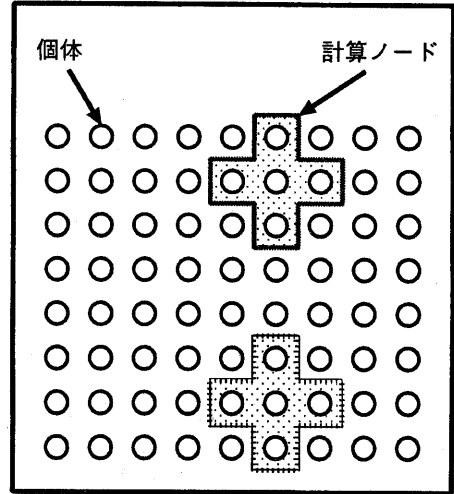


図2 近傍モデル
Fig. 2 Neighborhood model of Parallel GA.

このモデルでは、1つ、もしくは比較的少数の個体を1つの計算機ノードに割り当てる。そして、各個体についてその近傍を定義し、近傍ごとに遺伝的操作を局所的に行なうことになる。

近傍モデルでは、図2に示すように、遺伝的操作の局所性によって、個体群中のある個体の適応度が極端に高い場合でも、その影響は近傍間において緩やかに全体に波及していく。そのため、適応度の高い個体の急速な増加(初期収束現象)が抑制され、個体群の多様性が、比較的高く保たれることとなる。

離島モデルは地理的隔離により、独自の閉鎖的な生態系で進化の進んだ複数の個体群を、一定の世代毎に同期したり、完全な非同期の条件で個体を交換する処理として直観的に理解できる。しかしながら、単純に適応度の高い個体の交換や、任意の計算機ノード間での個体の交換では、初期収束による局所解に陥りやすい。したがって、移住させる個体の選択や交換するノードの選択についてはさまざまな工夫を施す必要がある。

また、近傍モデルでは少数の個体を計算機ノード間で頻繁に交換を行なうことから、ノード間の共有メモ

りを必要とする。したがって、共有バスによる密結合並列マシンを想定した場合、プロセッサ台数が制限されてしまう。

そこで、我々は離島モデルをベースとして、遺伝子の伝播を一次元の隣接間のみ限定した形態の並列GAモデルを地中海モデルとして提案する。次章では、地中海モデルの概念について述べる。3章では、地中海モデルをリング結合型の並列計算機およびワークステーションクラスタへ適用した場合について述べる。4章では、モデルの有効性を示すための実験について説明し、その検証を行なう。そして、結論と今後の予定については5章にまとめる。

2. 地中海モデル (Mediterranean Sea Model) の概念

離島モデルと近傍モデルの利点欠点をまとめると以下のようになる。

- 離島モデル

利点

- 大規模なシステムでの実現が可能
- アーキテクチャに依存しない実装が容易

欠点

- 初期収束による局所解陥りやすい。

- 近傍モデル

利点

- 個体の伝搬が緩やかで、局所解に陥りにくい。

欠点

- 隣接する計算ノード間において個体を共有するために共有メモリが必要
- 大規模な台数への対応が困難

そこで、上記の離島モデルと近傍モデルのそれぞれの欠点を克服し、利点を探り入れた新たなモデルを考案する必要がある。

共有メモリアーキテクチャは、共有バス結合をベースとして比較的小規模な台数においては、成功をおさめてきた。しかしながら、大規模な問題に対して並列

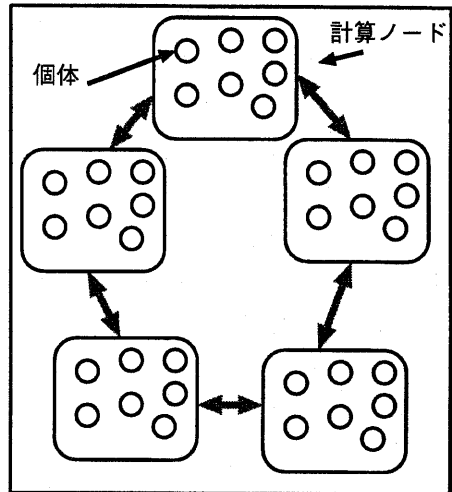


図3 地中海モデル

Fig. 3 Mediterranean Sea model of Parallel GA.

度を上げるためには共有バスのバンド幅の限界から、接続する台数が限定されてしまう。

近年、分散したメモリを仮想的に一つの共有メモリとしてユーザに提供する、分散共有メモリアーキテクチャの研究が進められ、商品化がされるようになってきた¹³⁾。しかしながら、分散共有メモリではキャッシュミスのペナルティが大きく、隣接ノード間において少数の個体を頻繁にやりとりする近傍モデルにおいてはキャッシュのヒット率は期待できず、並列化による性能向上は見込めないであろう。

一方、離島モデルは共有メモリを必要とせず計算ノード間においてメッセージ通信をおこなうことにより容易に実現できるが、前述の初期収束の問題を克服するためには、個体の移動に関して何らかの方策を取らなければならない。

そこで、離島モデルにおいて計算ノードを仮想的に一次元に接続された形態を想定し、個体の移動を隣接間において双方向に行なうものだけに限定した形態を考え、これを地中海モデル (Mediterranean Sea Model) として定義する。その概念を図3に示す。

地中海モデルの利点をまとめると以下のようになる。

- アーキテクチャに依存しない実装が容易

- リング結合のマシンへの適用が可能
- 初期収束による局所解陥りにくい。
- 共有メモリは必ずしも必要とせず、メッセージ通信による個体の移動が可能

3. 並列計算機への適用

ここでは、地中海モデルを実装するにあたっての、並列計算機アーキテクチャに関して述べる。地中海モデルの特徴としては、離島モデルにおける柔軟性をもとに、初期収束の回避をアーキテクチャの点からサポートしている点にある。

3.1 アーキテクチャ上の検討

地中海モデルにおいては、仮想的に一次元に接続されたノード間においてのみ個体の移動を行なう。したがって適用される並列計算機アーキテクチャとしては、リング結合アーキテクチャだけでなく、ネットワークスイッチにより接続されたワークステーションクラスタにおいても実現が可能である。すべてのノードからアクセスされる共有メモリは必要としないが、隣接間のみ限定することにより、既存のデュアルポートメモリを用いることで容易に実現できる。

3.2 リング結合型並列計算機への適用

我々は、4つのDSPを隣接間において共有メモリを設けたリング結合型の二次元信号処理システムを構築した¹⁴⁾。

その構成を図4に示す。

マルチDSPシステムは、当初は二次元信号処理のアルゴリズム検証のために構築したが、並列GAの評価にも適用可能である。この環境上で、ナップザック問題の並列処理を行なった。その処理過程を以下に示す。

- (1) 初期化において、いくつかのグループに個体を分割、配置する。
- (2) 以下の遺伝アルゴリズムの処理を各DSP上で行なう。
 - (a) 各個体の環境への適応度の評価
評価関数に基づいて各個体の持つ評価値

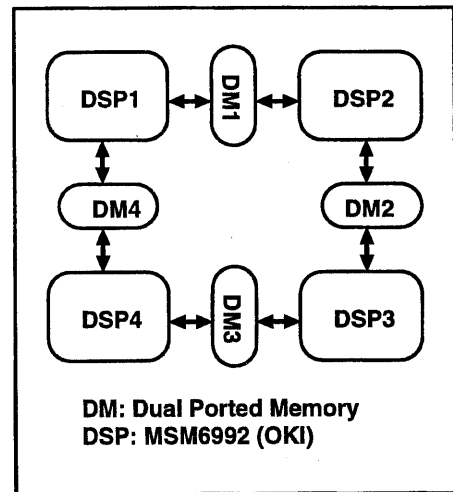


図4 マルチDSPシステム
Fig. 4 Configuration of multi-DSP system.

を計算する。評価は個体の各ビットに対応した荷物の容積の総和を計算することで得られ、収納容積により近いものが優秀な個体であると判断する。

- (b) 各個体の環境内での淘汰と増殖
適応度の評価で得られた評価値をもとに、収納容積に近い値を持つ優秀な個体の生存確立を高くすることにより、最適値から離れた個体が淘汰されるようになる。そして、淘汰された分だけ、優秀な個体の増殖を行ない、より多くの優秀な個体を生み出す。
 - (c) 交叉
増殖された個体のすべてについて、交叉の処理を行なう。交叉の長さや個体の位置については乱数により決定する。ただし、交叉後の評価値が全体の収納容積を越えないようにする。
 - (d) 突然変異
初期収束を回避するため、ある確率で任意の個体の任意のビットを反転させる。
- (3) 1世代の処理の終了後、DSP間で個体のいくつ

かを交換する。

以上の (1)~(3) の操作を準最適解が得られるまで順次繰り返す。

3.3 ワークステーションクラスタへの適用

近年、複数のワークステーションを用いたワークステーションクラスタが並列処理システムとして利用されている¹²⁾。

ワークステーションクラスタ上においては、PVM や MPI などのメッセージ通信ライブラリが利用され、実際の並列アプリケーションに活用されている。アルゴンヌ国立研究所においても、MPI をベースとした並列 GA のライブラリパッケージが開発され、利用されるようになった¹⁵⁾。

しかしながら、ワークステーションクラスタ上においては、仮想的に共有メモリを実現する研究はすすめられているものの¹⁶⁾、前述のように近傍モデルを実行するためには、キャッシュメモリへのヒット率などのさまざまな問題を含んでいる。

そこで、地中海モデルを適用することにより、ワークステーションクラスタ上においても、遜色ない収束速度で、なおかつ並列処理による処理時間の短縮により、単時間で準最適解が得られるようになる。

4. 性能評価

前章で述べたマルチ DSP システムを用いて、ナップザック問題を解いた場合について述べる。ここで用いた遺伝的アルゴリズムの処理方法、パラメータについて以下にまとめる。

- 個体数：64 個 (各 DSP ごとに 16 個)
- 交換する個体数：4 個
- 荷物の数：15 個および 30 個
- 総容量：15 個の場合 1000, 30 個の場合 2000
- 荷物の容積：5 ~ 300 のランダム

なお、比較のために 1 台のワークステーションを用いて、逐次的に同じ問題を解いた場合との比較を行った。

4.1 評価結果

以下に、実験結果を示す。

(1) 荷物数が 15 個の場合

- 地中海モデル
 - 得られた準最適解：平均 999.4 (997 ~ 1000)
 - 収束世代数：平均 10.37 世代 (9 ~ 13)

- 逐次処理

- 得られた準最適解：平均 999.4 (996 ~ 1000)
- 収束世代数：平均 9.03 世代 (9 ~ 13)

(2) 荷物数が 30 個の場合

- 地中海モデル

- 得られた準最適解：平均 1999.4 (1997 ~ 2000)
- 収束世代数：平均 11.47 世代 (9 ~ 27)

- 逐次処理

- 得られた準最適解：平均 1999.8 (1997 ~ 2000)
- 収束世代数：平均 10.07 世代 (8 ~ 17)

以上の結果、得られた準最適解は、地中海モデルにおいては、ワークステーションを用いて逐次 GA により処理した場合とではほとんど差はなかった。しかしながら、逐次に比べると荷物数が 15 個の場合は、最適解に近いものが、より多く得られていることがわかる。

また、収束するまでの世代数は地中海モデルの方がやや長くなっている。これは、個体の移住が隣接間に限ったものであるため、全体としての収束に至るまでにやや世代を費やしているものと考えられる。

しかしながら、並列処理の効果により各計算ノードにおける処理時間は短縮されているため、多少の世代数の増加は充分補えるものと考えられる。

5. おわりに

本稿では、並列 GA のモデルとして、地中海モデルの提案を行ない、その実現方式について述べた。

また、実際にナップザック問題に適用した場合の実

験を行なった結果、世代数においてはほとんど差はなく、得られる解についても準最適解により近いものが得られていることがわかった。

また、地中海モデルが、既存の並列計算機システムとの親和性が高いということを示した。並列処理により単時間で結果が得られることがわかった。

今回は、ハードウェアの制限により、4台の計算ノードによる実現にとどまったが、地中海モデルの有効性を示すためにはより多くの台数による実現が望まれる。

現在、ワークステーションを高速ネットワークで接続した並列処理環境の実現を行ない、その上で、より多くの台数により並列に処理を行なうことで、地中海モデルの有効性の検証を進める予定である。

また、分散共有メモリを用いて、キャッシュのヒット率を考慮した個体の移動を行なうことで、効率的な地中海モデルの実装を行なえるものと考えている。

謝 辞

マルチ DSP システム上において、並列 GA の実装に協力してくれました、NEC 株式会社の植村和容氏に感謝いたします。

参 考 文 献

- 1) H. Holland: "Adaptation in natural and artificial systems", Univ. of Michigan Press (1975).
- 2) 北野宏明: "遺伝的アルゴリズム 1,2,3", 産業図書 (1993~1997).
- 3) 玉置, 喜多, 岩本, 三宮: "遺伝アルゴリズム I~V", システム制御情報学会, (1995~1996).
- 4) D. E. Goldberg: "Genetic algorithms in search, optimization, and machine learning", Addison-Wesley (1989).
- 5) 市井, 石井: "遺伝子分布評価に基づく遺伝的アルゴリズムの多様性維持", 計測自動制御学会論文集, Vol.32, No.1, pp.1242-1250 (1994).
- 6) 森, 吉田, 喜多, 西川: "遺伝アルゴリズムにおける熱力学的選択ルールの提案", システム制御情報学会論文誌, Vol.9, No.2, pp.82-90 (1996).
- 7) T. C. Fogarty and R. Huang: "Implementing the genetic algorithm on transputer based parallel processing systems", Parallel problem solving from nature, Springer verlag, pp.145-149 (1990).
- 8) M. Schwehm and A. Ost: "Inference of

stochastic regular grammars by massively parallel genetic algorithms", Proc. of the 6-th international conference on genetic algorithms, pp.520-527 (1995).

- 9) T. J. Stanley and T. Mudge: "A parallel algorithm for multiobjective microprocessor design", Proc. of the 6-th international conference on genetic algorithms, pp.597-604 (1995).
- 10) E. Cantu-Paz and D. E. Goldberg: "Predicting speedups of idealized bounding cases of parallel genetic algorithms", Proc. of the 7-th international conference on genetic algorithms, pp.113-120 (1997).
- 11) P. Adamidis: "Review of parallel genetic algorithms bibliography", Technical Report, Automation and Robotics Lab., Aristotle univ. of Thessaloniki (1994).
- 12) C. Douglas, Timothy G. Mattson and Martin H. Shultz: "Parallel Programming Systems for Workstation Clusters," Technical Report TR-975, Yale University Department of Computer Science Research (1993).
- 13) 天野英晴, "並列コンピュータ," 株式会社 昭晃堂 (1996).
- 14) Hironori Nakajo, Kazuhiro Yoshikawa, Takao Hinamoto, Sadao Maekawa: "An Architecture of Multi-DSP System and Development Environment for Two-dimensional Digital Signal Processing", Proc. of Int. Conf. on Systems Engineering, pp.193-196 (1992).
- 15) David Levine: "Users Guide to the PGAPack Parallel Genetic ALgorithm Library", [http : //www.mcs.anl.gov/pgapack.html](http://www.mcs.anl.gov/pgapack.html) (1996).
- 16) 中條拓伯, 藏前健治, 金田悠紀夫, 前川禎男: "ソフトウェア DSM におけるコヒーレント・キャッシュシステムの実装と評価," 情報処理学会論文誌, 36 卷 7 号, pp.1719-1728, (1995).