

2 個体分散遺伝的アルゴリズム

廣安 知之[†], 三木 光範[†], 佐野 正樹[†], 谷村 勇輔^{††}

[†] 同志社大学工学部 ^{††} 同志社大学大学院

本研究では分散遺伝的アルゴリズム (分散 GA) の新しいモデルである 2 個体分散遺伝的アルゴリズム (Dual Individual Distributed Genetic Algorithm : DuDGA) を提案する。提案するモデルは、分散 GA におけるサブ母集団内の個体数を 2 としたものであり、多様性を維持し、かつ解探索能力を高める遺伝的オペレータを採用している。本モデルの利点は、分散 GA と比較して特定すべきパラメータが少なくなり、遺伝的操作や移住が簡単になることである。いくつかのテスト関数に本アルゴリズムを適用したところ、他の分散遺伝的アルゴリズムのモデルと比較して、高い探索性能を有することが明らかとなった。また、島内に 2 個体しか存在しないが、その探索方法は、fine-grained モデルとは異なっている。

Dual Individual Distributed Genetic Algorithms

Tomoyuki HIROYASU[†], Mitsunori MIKI[†], Masaki SANO[†], and Yusuke TANIMURA^{††}

[†] Graduate School of Knowledge Engineering, Doshisha University

^{††} Knowledge Engineering Dept., Doshisha University

This paper describes a new model of distributed genetic algorithm, "Dual Individual genetic algorithms : DuDGA". In this algorithm, subpopulation size is two. Specialized genetic operators which keep the diversity of the solutions and contribute the high searching ability are performed in each subpopulation (island). The advantage of this model is that this algorithm has fewer parameters that need to be specified than traditional distributed genetic algorithm (DGA) has. Through the numerical example, it became cleared that DuDGA has a higher searching ability compared to the traditional DGA. It is also said that even there are two individuals in each island, searching method of DuDGA is different from that of fine-grained model.

1 序論

遺伝的アルゴリズム (Genetic Algorithm : GA) は、生物の進化を模倣した確率的な最適化手法である¹⁾。GA は、目的関数の勾配情報を使用せず、離散問題にも適用できためその適用範囲は広い。その反面、多点探索であるため膨大な反復計算を必要とし、計算コストが高い。

このため、GA の並列処理に関しては、多くの研究がなされてきた²⁾。その代表的なものとして、近傍モデルと島モデルが挙げられる。近傍モデルでは、個体の近傍を定義して局所的な操作を繰り返す。また島モデルでは、個体の母集団を複数のサブ母集団に分割して GA を適用する。島モデルは分散遺伝的アルゴリズム (Distributed Genetic Algorithm : DGA) とも呼ばれる。

本研究では、DGA の新しいモデルである 2 個体分散遺伝的アルゴリズム (Dual Individual Distributed Genetic Algorithm) を提案する。DuDGA は、DGA においてサブ母集団の個体数を 2 とし、その個体数に適した遺伝的オペレータを適用したものである。DuDGA は DGA と比較してアルゴリズムが単純化しており、かつその解探索能力が優れている。また、個体の母集団が非常に細かく分割されているが、近傍モデルとは異なる探索を行うモデルである。これらの DuDGA の特性は、数値計算例を通じて検討している。

2 遺伝的アルゴリズムと並列モデル

2.1 遺伝的アルゴリズムの概要

GA の概要は、以下の通りである。まず、個体の母集団 (population) を生成する。各個体は、ビット列として表現され、環境に対する適合度 (fitness) が設定される。そして、この初期母集団に対し、(1) 適合度の高い個体が増殖して生き残るようにする選択 (selection)、(2) ある個体の一部を別の個体の一部と入れ替えて新しい個体を生成する交叉 (crossover)、(3) 個体の一部を変化させる突然変異 (mutation)、という操作を繰り返し適用する。これにより、解の候補としての個体が成長し、より適合度の高い個体すなわち最適解に近い個体が増えていくことが期待される。また、上記の選択、交叉、突然変異を総称して遺伝的操作 (genetic operator) といい、遺伝的操作の繰り返し単位を、世代 (generation) という。

2.2 遺伝的アルゴリズムの並列モデル

GA の代表的な並列モデルとしては、近傍モデルと分散 GA (DGA) が挙げられる。以下では、それぞれのモデルの概要を説明する。

近傍モデル (neighborhood model) は、細粒度の並列 GA (fine-grained parallel GA) とも呼ばれる (図 1)。このモデルでは、可動範囲 (range または mobility) によって各個体の近傍 (neighborhood) が定義され、その近傍内で選択や交叉を行う^{3, 4)}。各個体の近傍の一部は、他の個体の近傍の一部と重なり合っており、それぞれの個体の及ぼす影響は次第に個体集団内に波及していく。本研究では、このモデルを FG と呼ぶ。

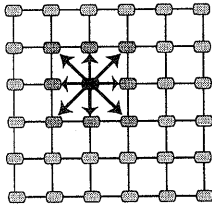


図 1: 近傍モデル

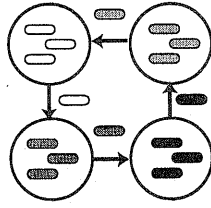


図 2: 分散 GA

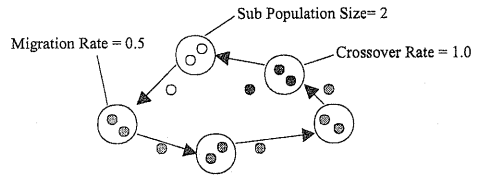


図 3: 2 個体分散遺伝的アルゴリズム

DGA は、島モデル (island model) または粗粒度の並列 GA (coarse-grained parallel GA) とも呼ばれる (図 2)。このモデルでは、個体の母集団を複数のサブ母集団 (島) に分割し、分散処理を行う^{5, 6, 7)}。より具体的には、サブ母集団ごとに遺伝的操作を適用し、ある間隔で他のサブ母集団と個体を交換する。この個体の交換を移住 (migration) といい、移住を行う間隔と移住を行う個体数の割合を、それぞれ移住間隔 (migration interval)、移住率 (migration rate) と呼ぶ。

なお、上記のような並列モデルに対し、母集団を分割せずに遺伝的操作を実行するモデルを、本論文では単一母集団 GA (single population GA : SPGA) と呼ぶことにする。

3 2 個体分散遺伝的アルゴリズム

DuDGA の手順は以下の通りである。まず、個体の母集団を、2 個体ずつの島に分割する。そして各島において、次の操作を世代ごとに繰り返す。

1. 一つ前の移住から一定世代経過していた場合に、移住を行う。まず、2 つの個体のうち、ランダムに一方を選択し、そのコピーを他の島に送る。そして適合度の低い方の個体は、他の島から送られてきた個体に置き換えられる。
2. 2 つの個体を交叉させ、新しい 2 つの子個体を生成する。本論文では、一点交叉を用いている。この段階では、親個体も存在している。
3. 突然変異を行う。交叉で生成された 2 つの子個体をそれぞれ 1 ビット反転させるが、反転する点は、2 つの個体で 1 ビットずらす。これは、島内の 2 つの個体が同一になるのを防ぐためである。
4. 適合度の評価を行う。
5. 2 つの親個体と、2 つの子個体から、それぞれ適合度の高い方の個体を選び、次世代の 2 個体とする。

DuDGA は、従来の分散 GA と比べてアルゴリズムが簡単化している (図 3)。サブ母集団内の個体数を 2 とすることにより、総個体数を決定すれば島数も一意に決まる。また、交叉を行うペアは一通りしかなく、移住率も 0.5 と決まる。

4 数値実験

4.1 テスト関数

本論文で対象とするテスト関数は、表 1 に示す 5 つの関数である。同表に示す関数のうち、Rastrigin 関数 (F1)、Rosenbrock (F2)、Griewank 関数 (F3)、Ridge 関数 (F4) における大域的最適解は 0 である。また、Schwefel 関数 (F5) については、定数を足しこむことで、最適解が 0 となるようにしている。

表 1: テスト関数

$$\begin{aligned}
 F1 &= 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \\
 &\quad (x_i \in [-5.12, 5.12]) \\
 F2 &= \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2) \\
 &\quad (x_i \in [-2.048, 2.048]) \\
 F3 &= 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \left(\cos\left(\frac{x_i}{\sqrt{i}}\right) \right) \\
 &\quad (x_i \in [-512, 512]) \\
 F4 &= \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2 \\
 &\quad (x_i \in [-64, 64]) \\
 F5 &= \sum_{i=1}^n -x_i^2 \sin(\sqrt{|x_i|}) \\
 &\quad (x_i \in [-512, 512])
 \end{aligned}$$

4.2 分散遺伝的アルゴリズムとの比較

本章では、DuDGA と DGA の比較を行う。対象とする関数は表 1 の、F1, F2, F3, F4 である。このうち、F1 は 20 次元、F2 は 5 次元、F3 と F4 は 10 次元のものを使用した。特に断りが無い限り、実験結果は全て 20 試行平均である。また、本章における数値実験では、表 2 に示すパラメータを用いた。

4.2.1 島数と性能

DuDGA は、DGA において島数を最大にしたものである。そこで本節では、DGA における島数の及ぼす影響を調べ、DuDGA との比較を行う。

まず、一定世代探索を実行した時、どの程度の確率で最適解を得ることができるか、という点について検討する。本節ではこの指標を、解探索の信頼性と称す。図 4 は、20 試行において、5000 世代後に最適解を発見する割合を示したものである。同図より、島数が多いほど探索の信頼性が向上する傾向がある。そして DuDGA は、どの島数における DGA よりも優れている。

次に、最適解への収束の速さの比較を行う。図 5 は、最適解を発見するまでの関数評価回数を示したもので

表 2: 使用したパラメータ

項目	説明
個体数	240
エリート保存	1 個体
染色体長 (L)	設計変数の数 × 10
選択法	ルーレット選択
交叉率	1.0
交叉法	1 点交叉
突然変異率	1/L
移住間隔	5
移住率	0.3

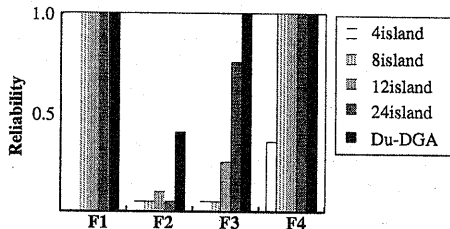


図 4: 解探索の信頼性と島数

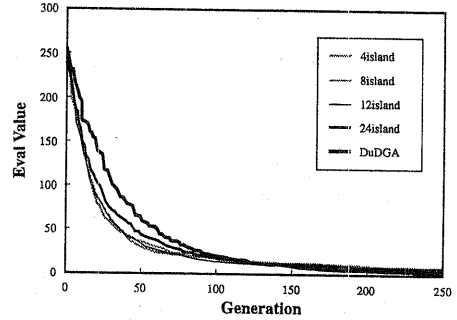


図 6: 関数評価値

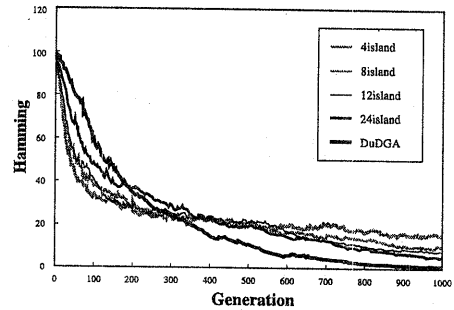


図 7: ハミング距離

ある。同図より、島数が多い DGA ほど少ない評価回数で最適解を発見できる、という傾向がある。そして、DuDGA はさらに収束が速い。

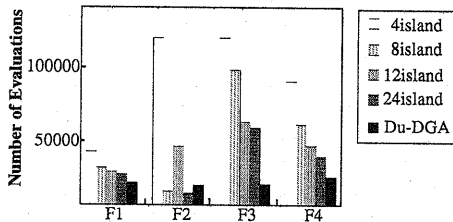


図 5: 関数評価回数と島数

最後に、図 6 と図 7 に F1 に対する解探索の遷移の様子を示す。図 6 では、世代数を横軸にとり、関数評価値の履歴を示している。また図 7 は、最適解とのハミング距離の平均を示したものである。同図より、DuDGA では、探索の初期では DGA よりも多様性が維持され、解の収束が遅い。そして探索が進むにつれて、DuDGA における多様性は急速に失われると共に解の収束が速くなる。このことは、DuDGA は初期の段階では広域的な探索を行い、探索が進むにつれて探索範囲を限定していくことを示している。

4.2.2 個体数の影響

GA では、個体数が少なすぎると、得られる解の精度が悪くなる。このため、最適な個体数が未知である対象問題に対しては、ある程度多めの個体数を設定しなければならない。しかし、個体数が増えると計算コストが高くなる。よって、個体数の増加に対して評価計算回数の増加ができるだけ小さいモデルが望ましい。

図 8 は、F1 における最適解発見までの評価計算回数と個体数の関係を示したものである。DGA においては、島数が多い方が評価回数の増加が少ない。そして、DuDGA はさらに個体数の影響を受けにくい。よって、この観点においては DuDGA が優れているといえる。

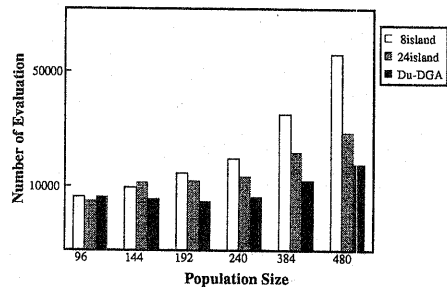


図 8: 関数評価回数と個体数

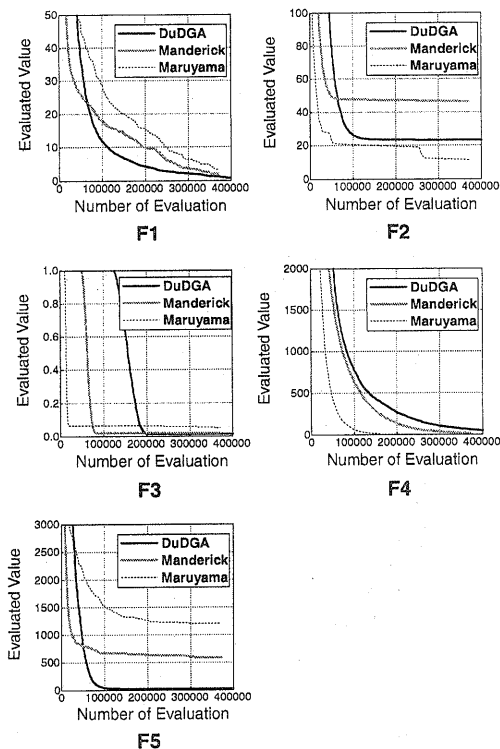


図 9: DuDGA と FG との比較

4.3 FG との比較

DuDGA では、母集団が非常に細かく分割されている。この点においては FG と類似している。そこで本章では、DuDGA と、以下に示す 2 種類の FG との比較を行う。テスト関数は表 1 の 5 つの関数であり、いずれも 30 次元である。

比較対象とする FG の一つは、Manderick らのモデル⁴⁾である。このモデルでは、各個体を 2 次元の格子状に配置し、隣接する個体同士で遺伝的操作を行う。本実験では、可動範囲を 1 とし、近傍の最良個体が生き残る選択手法を用いた。

もう一つの FG は、Maruyama らのモデル⁸⁾である。このモデルの特徴は、通信の量が少ないことと、近傍を定義しないことである。またこのモデルでは、他ノードの個体の情報をバッファに格納する。本実験では、バッファサイズを 3 とし、ルーレット選択を用いた。

本章では、DuDGA・FG とともに個体数を 400 とした。DuDGA におけるその他の設定は 3 章に従う。FG に関しては、交叉率を 0.8 とした。その他の設定については表 2 に従うものとする。また、実験結果は 10 試行平均である。

図 9 は、横軸に関数評価回数をとり、関数評価値の履歴を示したものである。どの関数においても、探索の初期においては FG の方が DuDGA よりも解の収束が速い。また、解の探索能力の優劣は問題によって異なっている。GA での解探索に適した F1 では DuDGA の

方が FG よりも解の優れているといえ、GA での解探索に適していない F2 や F4 では、FG の方が優れている。SPGA が得意な F5 では、DuDGA は非常に良い解を発見している。これらの結果から、DuDGA と FG とは異なった探索を行っていると言える。

5 結論

本論文では、分散遺伝的アルゴリズム (DGA) の新しいモデルである 2 個体遺伝的アルゴリズム (DuDGA) を提案した。そして、モデルの有効性を検討するために、数値計算例を通じて他のモデルの GA との比較を行い、その基本性能について検討を行った。

DGA は個体数が一定の場合、島数が多いほど解探索の性能が向上する場合があり、島数を限界まで多くした DuDGA は高い探索能力を持っている。また DuDGA は DGA と比較して、個体数の増加に対して、最適解を求めるまでの関数評価回数の増加が少ない。本論文における実験では、DuDGA が通常の DGA よりも優れた結果を示した。

また、FG は DuDGA と比較して、探索の初期段階における解の収束が速いという傾向が見られた。解の探索能力は問題依存であり、目的関数によっては DuDGA によって得られる解の精度が勝る場合もある。

なお、本研究の遂行に当たり、平成 9 年度「学術フロンティア推進事業」(電磁環境とインテリジェントエレクトロニクス)における研究課題「PC クラスタによる数値解析の並列処理」に関わる研究費を用いた。ここに記して謝意を表す。

参考文献

- 1) D.E.Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley, 1989.
- 2) Erick Cantú-Paz. A survey of parallel genetic algorithms. *Calculateurs Paralleles*, Vol. 10, No. 2, 1998.
- 3) H. Mühlenbein. Parallel genetic algorithms, population genetics and combinatorial optimization. *Proc. 3rd International Conference on Genetic Algorithms*, pp. 416-421, 1989.
- 4) Bernard Manderick and Piet Spiessens. Fine-grained parallel genetic algorithms. *Proc. 3rd International Conference on Genetic Algorithms*, pp. 428-433, 1989.
- 5) Reiko Tanese. Distributed genetic algorithms. *Proc. 3rd International Conference on Genetic Algorithms*, pp. 434-439, 1989.
- 6) Chrisila C. Petty and Michael R. Leuze. A theoretical investigation of a parallel genetic algorithm. *Proc. 3rd International Conference on Genetic Algorithms*, pp. 398-399, 1989.
- 7) H. Mülenbein and J. Born M. Schomisch. The parallel genetic algorithm as function optimizer. *Proc. 4th International Conference on Genetic Algorithms*, pp. 271-278, 1991.
- 8) Tsutomu Maruyama, Tetuya Hirose, and Akihiko Konagaya. A fine-grained parallel genetic algorithm for distributed parallel systems. *Proc. 5th International Conference on Genetic Algorithms*, pp. P.184-190, 1993.