異なるコード化を用いた環境分散遺伝的アルゴリズムの検討

細江 則彰

1　はじめに

遺伝的アルゴリズム（Genetic Algorithm: GA）[1, 2] とは，生物が環境に適応し進化していく過程を工学的に模倣した確率的な最適化アルゴリズムである．GA は優れた最適化手法として知られているが，「計算負荷が高い，早熟収束によって局所解に収束する，パラメータ設定が複雑である」などの問題が存在する．

このような問題を解決する方法として並列化モデルの分散遺伝的アルゴリズム（Distributed GA: DGA）[3] が提案されている．

また，パラメータ設定が複雑であるといった問題を解決する方法として，DGA において複数のサブ母集団に異なる遺伝的パラメータを与える環境分散遺伝的アルゴリズム（Distributed Environment GA: DEGA）[4] が提案されている．

DGA や DEGA などの分散モデルにおいて，個体群は各サブ母集団毎にある特徴を持っている．それらの個体群を交換する移住という操作について調査することは重要であると考えられる．本研究では移住をより効果的にするために，どのサブ母集団からどのサブ母集団に移住するかや，どの個体を移住するかを動的に調整することを目的としている．そこで共進化アルゴリズムの概念を取り入れ，移住の効果を協力，非協力に分類し，それが解探索性能にどのような効果をもたらすかを調査する必要がある．

本報告では，与えられた解の周りの探索分布を変化させ，新しい領域を発見すると報告されている，複数のコード化を用いた DEGA[5] の検証を行い，移住によってどのような効果がもたらされたのかを考察する．

2　分散遺伝的アルゴリズム

DGA は母集団を複数のサブ母集団に分割する GA の並列化モデルの 1 つである．各サブ母集団は独立に遺伝的操作を行うため，各サブ母集団において独自の探索が進み，母集団全体として多様性が維持される．

そして DGA では，一定期間ごとにサブ母集団間で移住と呼ばれる個体の交換操作を行う．移住を行うことにより，各サブ母集団の特徴を持った個体が移住先のサブ母集団に影響を与えるため，サブ母集団での早熟収束を防ぐことができる．また，DGA では各サブ母集団において異なる探索を行うため，最適解となる染色体の一部である部分解を各サブ母集団で形成しやすい．各サブ母集団で見つかった部分解を組み合わせるためにも，移住は重要な操作である．

移住に関するパラメータとして，移住を行う世代間隔を移住間隔，移住する個体のサブ母集団に対する割合を移住率と呼ぶ．Fig. 1 に移住の概念図を示す．

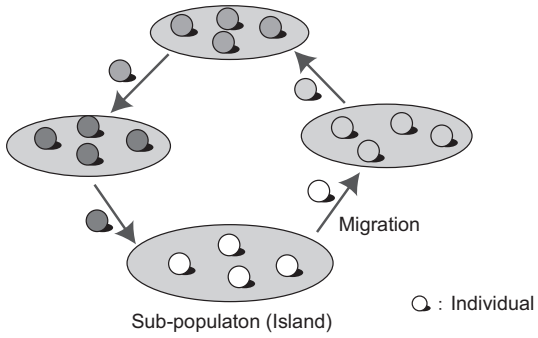


Fig. 1　DGA の移住の概念図

3　環境分散遺伝的アルゴリズム

GA を実行する際にはさまざまなパラメータを設定する必要があり，それらのパラメータの違いは解探索性能に大きく影響する．したがって，良好な解を得るためにはパラメータを適切な値で設定することが重要である．しかし，それらのパラメータの適切な値を知るためには膨大な予備実験を行うか，もしくは適応的にパラメータ調整する手法を用いる必要がある．この問題を解決するために，DGA においてサブ母集団ごとに異なる遺伝的パラメータを設定する DEGA が提案されている．DEGA では，様々な遺伝的パラメータを各サブ母集団に設定することによって，パラメータ調整の負担を軽減させると報告されている．

一方で，DEGA における環境を分散する方法として，コード化を用いる手法が考えられる．各々のサブ母集団で異なる表現を用いたモデルは，移住の間に個体を 1 つの表現からもう 1 つの表現に変化させるものである．表現を切り替えることで，与えられた解の周りの探索分布を変化させ，新しい領域を発見することが期待できる．

本報告では，1 つのサブ母集団がバイナリの符号化を使用し，もう 1 つのサブ母集団がグレイコードを使用する 2 つの母集団の並列モデルで検証する．

4　共進化アルゴリズム

共進化アルゴリズムは，従来の進化的計算のように静的に与えられた適応度関数を用いるのではなく，個体同士が影響を及ぼし合うことによって適応度が決定する．ある個体の適応度はそれを評価するための集団によって違うため，適応度地形は世代に応じて動的に変化する．

また，その相互作用の定義によって競合型（両者が害を与え合う：非協力），協調型（両者共に利益を受ける：協力）の 2 種類に分けられる．

本来 DGA などの分散モデルと共進化アルゴリズムは直接的に関係のない概念であるが，分散モデルにおいて行われる移住という操作によって各サブ母集団が互いに影響し合っていることから，その影響を協力と非協力に分類して移住という操作を調査する必要があると考えることができる．

5　異なるコード化を用いた DEGA

5.1　コード化方法

GA は設計変数をコード化することで最適化を行うことができる手法である．連続最適化問題の場合，遺伝子に設計変数をビットにコード化したビット列を入れることにより状態を表す．設計変数をコード化する方法は，主にバイナリコーディング (binary-coding) とグレイコーディング (gray-coding) の 2 つがある．

- バイナリコーディング

  整数を 2 進法によって表現したものである．隣り合う値に対応するコードのハミング距離が一定でないため，最適解に収束しにくいという問題が現れる．

- グレイコーディング

  連続する 2 つの符号間のハミング距離が常に 1 であるという特徴がある．これにより，グレイコーディングはバイナリコーディングと比べて，特に終盤での解探索の効率が良くなるという報告がある [6]．

10 進数の 0〜7 をバイナリ，グレイコードで 2 進数にコード化するときの対応表を Table 1 に示す．

Table 1　バイナリコードとグレイコードの対応

DN	Binary	distance	Gray	distance
0	000	-	000	-
1	001	1	001	1
2	010	2	011	1
3	011	1	010	1
4	100	3	110	1
5	101	1	111	1
6	110	2	101	1
7	111	1	100	1

5.2　母集団間の表現方法

本報告では，それぞれのサブ母集団に 2 つのコード化を適用した DEGA を用いて実験をする．そこでは，各サブ母集団間で移住する際にコード化の方法が変化する．その状況を Fig. 2 に示す．例えば設計変数が (5，3) である個体が移住すると，染色体の情報を変化させることで整合性をとる．

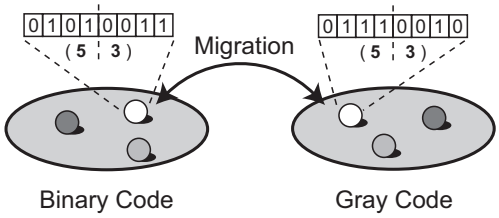


Fig. 2　表現方法の変化

6　数値実験

6.1　実験概要

移住によって協力，または非協力に働くことで全体の性能が良好になることを確かめるために，複数のコード化を用いた DEGA が，単一のコード化を用いる DGA よりも良好な解探索性能をとることを示す．そこで，あるコード化にとって難しい領域において，他のコード化にとっては簡単であるような F 関数を用いる．比較手法は，バイナリコードとグレイコードを適用するサブ母集団が 1 つずつ存在する DEGA と，バイナリコードを適用するサブ母集団が 2 つの DGA と，グレイコードを適用するサブ母集団が 2 つの DGA である．

次に，複数のコード化を用いた DEGA の有効性を示すために，代表的なテスト関数である，Rastrigin 関数，Schwefel 関数，Griewank 関数，および Rosenbrock 関数の 4 つの関数を用いて，上記の 3 つの手法を比較する．なお，全て 50 回試行の中央値と平均値で比較を行う．

6.2　実験 (F 関数)

6.2.1　対象問題とパラメータ

本実験の対象問題は，最小化問題の F 関数である．F 関数を構成する関数を以下に説明する．

$$easy(a) \quad = \quad hamming(a, 0^n)/4 \tag{1}$$

式 (1) に $easy$ 関数を示す．$hamming$ 関数は，2 つの引数の異なるビット数を返す．また 0^n は 0 からなる長さ n の文字列を意味する．この関数は，0^n である 1 つの大域的最適解を持つ単純な単峰性関数である．

$$deceptive(a) \quad = \quad \begin{cases} 3 * hamming(a, 0^n) \\ \qquad\qquad hamming(a, 0^n) < n/4 \\ n - hamming(a, 0^n) \\ \qquad\qquad otherwise \end{cases} \tag{2}$$

式 (2) に *deceptive* 関数を示す．この関数はだまし問題であり，2 つの最適解を持つ．1 つは大域的最適解である 0^n であり，もう 1 つは局所解である 1^n である．

$$F_1(x) \quad = \quad deceptive(gray(x)) + easy(binary(x)) \quad (3)$$

式 (3) に F_1 関数を示す．グレイコードを使用するとき，*easy* 関数の要素は小さく，影響も少ない．しかし，バイナリコードを使用するとき，性能に影響を及ぼす．F_1 はグレイコードには難しく，バイナリコードには比較的簡単である．

$$F_2(x) \quad = \quad deceptive(binary(x)) + easy(gray(x)) \quad (4)$$

式 (4) に F_2 関数を示す．同じように，F_2 はバイナリコードには難しく，グレイコードには比較的簡単である．

$$F(x) \quad = \quad min(F_1(x), F_2(x)) \quad (5)$$

式 (5) に F 関数を示す．この関数は F_1 と F_2 の局所解を継承し，評価値が 0 である 0^n において 1 つの大域的最適解を持つ．

Table 2 に本実験で用いたパラメータを示す．

Table 2　パラメータ

Number of islands	2
Number of individuals	100
Chromosome length	100
Selection method	Tournament selection
Number of tournament size	4
Number of elite individuals	1
Crossover method	Two point crossover
Crossover rate	1.0
Mutation rate	0.01
Migration rate	0.1
Migration interval	5，50
Migration policy	copy best - replace random

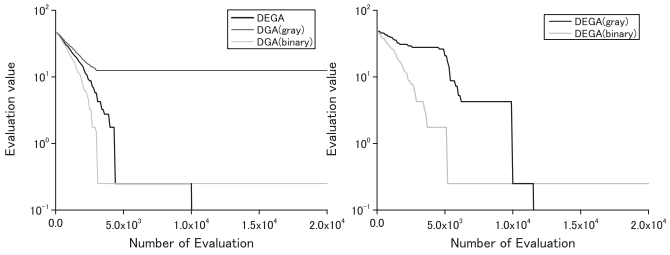
6.2.2　実験結果

Fig. 3(a) は，F 関数における DEGA と DGA の解探索履歴である．横軸は評価計算回数，縦軸は評価値を示しており，小さい評価値を示す手法ほど優れている．

Fig. 3(a) より，DGA はそれぞれ探索序盤で局所解に陥っているのに対して，DEGA のみが最適解に達しているのがわかる．

次に，移住の影響をより明らかにするために，移住を 50 世代ごとに行った 1 回試行のそれぞれのサブ母集団の個体の値の結果を Fig. 3(b) に示す．

Fig. 3(b) より，グレイコードを適用した母集団が局所解から抜け出せないでいるが，バイナリコードのサブ母集団の個体が移住したことよって，グレイコードのサブ母集団の個体が最適解に達しているのがわかる．



(a) 50 回試行の中央値　　(b) DEGA における各サブ母集団の値

Fig. 3　解探索履歴

6.3　実験 (4 つの関数)

6.3.1　対象問題とパラメータ

本実験の対象問題は，最小化問題である 4 つのテスト関数，Rastrigin 関数，Schwefel 関数，Griewank 関数，および Rosenbrock 関数を用いる．それぞれの関数の特徴を以下に述べる．

- Rastrigin 関数

  最適解の周辺に格子状に複数の準最適解を持つ多峰性の関数であり，変数間に依存関係はない．

- Schwefel 関数

  多峰性の関数であり，四隅のいずれか 1 つで最小値をとる．また，変数間に依存関係はない．

- Griewank 関数

  大域的には単峰性であるが最小値付近では多くの準最適解をもち，変数間に弱い依存関係がある．

- Rosenbrock 関数

  単峰性の関数であり，変数間に強い依存関係がある．

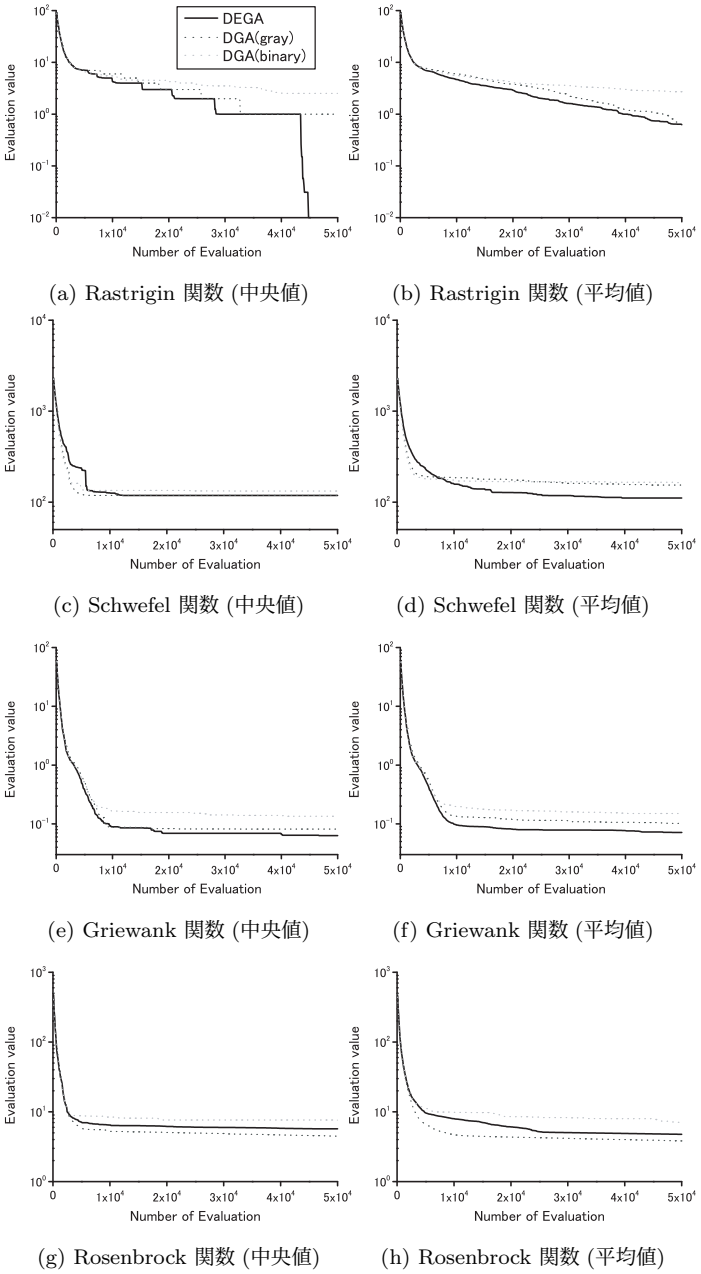
Table 3 に本実験で用いたパラメータを示す．

Table 3　パラメータ

Number of islands	2
Number of individuals	100
Chromosome length	120
Selection method	Tournament selection
Number of tournament size	4
Number of elite individuals	1
Crossover method	Two point crossover
Crossover rate	1.0
Mutation rate	0.008333
Migration rate	0.1
Migration interval	5
Migration policy	copy random - replace random

6.3.2 実験結果

Fig. 4 は，4 つの関数における DEGA と DGA の 50 回試行の中央値と平均値の解探索履歴である．



(a) Rastrigin 関数 (中央値)　(b) Rastrigin 関数 (平均値)

(c) Schwefel 関数 (中央値)　(d) Schwefel 関数 (平均値)

(e) Griewank 関数 (中央値)　(f) Griewank 関数 (平均値)

(g) Rosenbrock 関数 (中央値)　(h) Rosenbrock 関数 (平均値)

Fig. 4　解探索履歴

Fig. 4 より，Rosenbrock 関数を除く 3 つの関数に対しては，2 つのコード化を用いた DEGA が最も良好な解探索性能を示した．これは移住によって表現が互いに助け合った効果だと考えられる．Rosenbrock 関数に対しては，グレイコードを用いた DGA が僅かに良い解探索性能を示した．また，グレイコードとバイナリコードを比較すると，全ての関数に対してグレイコードの方が良好な解探索性能を示した．

6.4 考察

Rosenbrock 関数において DEGA よりも，グレイコードを用いた DGA の方が解探索性能が良かったのは，Rosenbrock 関数が単峰性の関数であり，ハミング距離が常に 1 であるグレイコードが極めて適していたためであると考えられる．

本報告で示した結果は，異なったコード化や移住を変化させた島モデルが，だまし問題や多様な難しい問題に対して，DGA よりも良好な解探索性能であることを示した．DEGA が 2 つのコード化のいずれかを用いた DGA よりも良好な性能をしたのは興味深いことである．このことは，動的に近傍を変化させる表現は一般的に静的な近傍の表現よりも良好であると考えることができる．また共進化の観点で考えると，DEGA では移住によって協力的に探索が進んでいたと考えられる．

7 今後の課題

どのサブ母集団からどのサブ母集団に移住するかや，どの個体を移住させてどの個体と交換するかを，動的に調整するためには，移住によって解探索性能がどのように変化するのかを調査しなければならない．そこで，どのような状況において移住が協力，または非協力に働けば効果的かを調査する必要がある．

今後まず，DGA において作為的に移住の方法を変化させて，その影響を調査する．次に DEGA において，それぞれ異なった特徴を持つ個体群に対しての移住の方法についても調査することが課題である．

参考文献

1) 三宮信夫，喜多一，玉置久，岩本貴司. 遺伝的アルゴリズムと最適化. 朝倉書店,1998.

2) J.H.Holland. Adaptation In natural and Artificial Systems. University of Michigan Press,1975

3) Reiko Tanese. Distributed Genetic Algorithms, Proceedings of the Third International Conference on Genetic Algorithms,1989.

4) Mitsunori Miki, Tomoyuki Hiroyasu, Mika Kaneko. A Parallel Genetic Algorithm with Distributed Environment Scheme. Genetic and Evolutionary Computation Conference,pp376,2000.

5) Zbigniew Skolicki, Kenneth De Jong. Improving Evolutionary Algorithms with Multi-representation Island Models. Lect Notes Comput Sci,pp420,2004.

6) 三木 光範, 廣安 知之, 吉田 純一, 金子 美華. 分散遺伝的アルゴリズムの性能に及ぼす交叉法とコーディング法の影響. 情報処理学会第 59 回全国大会,pp.139,1999.