

REAL: Differential Evolution における関数評価回数の削減の提案

REAL: Generation Alternation Model for Reducing the Number of Function Evaluations in Differential Evolution

串田 淳一
Jun-ichi KUSHIDA

立命館大学情報理工学部
College of Information Science and Engineering, Ritsumeikan University
kushida@spice.ci.ritsumeai.ac.jp, <http://www.spice.ci.ritsumeai.ac.jp/top/index-j.html>

大場 和久
Kazuhiro OBA

日本福祉大学健康科学部
Faculty of Health Sciences, Nihon Fukushi University
oba@n-fukushi.ac.jp, <http://syslab.n-fukushi.ac.jp/>

亀井 且有
Katsuari KAMEI

立命館大学情報理工学部
College of Information Science and Engineering, Ritsumeikan University
kamei@ci.ritsumeai.ac.jp, <http://www.spice.ci.ritsumeai.ac.jp/top/index-j.html>

keywords: differential evolution, genetic algorithms, generation alternation models

Summary

Differential evolution (DE) was introduced by Stone and Price in 1995 as a population-based stochastic search technique for solving optimization problems over continuous space. DE is an effective solving method for many problems even though it has few control parameters to be set. DE belongs to a group of evolutionary algorithms and is similar to Genetic Algorithm (GA). The generation alternation model of DE is a discrete generation model. In a differential operation, a target vector, a base vector and differential vectors are chosen as parents and generate a child called trial vector. The children are evaluated and replaces its parent if its fitness is better than that of its parent.

In this paper, we propose a new DE generation alternation model called Roulette Selection Based on Evolutionary Advance Level (REAL) for reducing the number of function evaluations. Firstly, we design the DE/MGG model which based on a policy of Minimal Generation Gap (MGG). In the differential operation, a number of children are generated and the target vector is alternated to the best individual from the family. Secondly, we define an evolutionary advance level as a degree of progress of the alternation of generations and modify DE/MGG as a propose model which make a difference in the evolutionary advance level of the individuals. In the proposed model, parents are selected based on its evolutionary advance level and the number of children is also decided depending on the evolutionary advance level of the target vector. Finally, we compare the proposed model to conventional DE and DE/MGG through experiments on several test functions and show a searching performance of proposed model.

1. は じ め に

Differential Evolution (DE)[Storn 95, Price 05] は実数値最適化問題を対象とした探索手法であり, 多点探索法であること, 直接探索法であることから遺伝的アルゴリズム (Genetic Algorithm: GA) と同様に進化計算に分類できる. DE の優れた特徴の一つに, 制御パラメータの数が少なく設定が容易であることがあげられる. また, いくつかのテスト関数を対象とした数値実験によれば, 典型的な実数値 GA や進化戦略と比較し, DE は最適解への収束が速く頑強であることが分かっている [Price 05, 伊藤 05]. そのため DE は多くの分野で利用されているが, 近年の最適化問題の大規模化と複雑化に伴い性能改善が望まれている.

これまでの DE におけるアルゴリズムの改良として, 進

化の過程で得られたデータにもとづく制御変数の自動調整 [山口 08] や, 低精度の近似モデルを利用した解の比較推定法の提案 [高濱 08] などが報告されている. さらに DE の世代交代モデルに関する研究として, 連続世代モデルに基づく DE の提案 [田川 06, 田川 08] や genDE[Noman 06] などがある. 通常の DE は離散世代モデルであり, 現世代と次世代の二つの集団が存在し, 現世代の集団は次世代の集団によって一斉に置き換えられる. 一方, [田川 06] で提案された連続世代モデルは, 新たに得られた優れた個体を即座に親として子の生成に利用できるといった利点がある. また genDE は, 生存選択において親集団と子集団を合わせた集団の中から適応度の高い順に集団サイズ分の個体を選択する. これにより, 親よりも適応度が低いがある程度優秀な個体を生存させることができる.

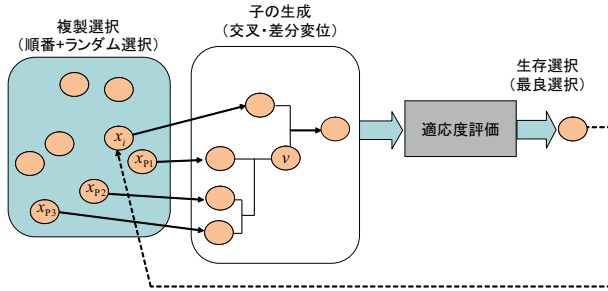


図3 DEの世代交代モデル

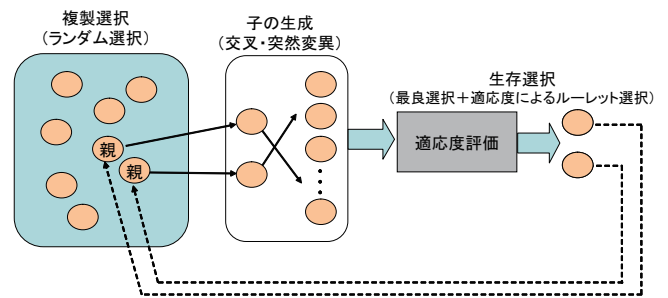


図4 MGGの世代交代モデル

3. 世代交代モデルの比較

3.1 DEの世代交代モデル

DEの世代交代モデルを図3に示す。DEの複製選択は第2章で示したように、対象親個体は個体集団から順番に選ばれ、他の親個体は集団からランダムで選ばれる。この際の各親の選択は重複を許さない非復元抽出となる。生存選択の範囲は親子(対象親個体と子個体)に局所化されており、親子の関数値を比較し、何れか勝る方を次世代の個体集団に選ぶ。DEは分散世代モデルであり、対象親個体と子個体の比較(親子間競争)を行って次世代の集団を形成した後に、現世代と次世代の集団を一斉に置き換える。そのため世代あたりの対象親選択回数は N_P となる。

3.2 MGG

佐藤らにより提案されたMGGは、初期収束や解探索の停滞を防ぐのに効果的なGAにおける世代交代モデルである。MGGの世代交代モデルを図4に示す。MGGの複製選択では集団から2個体をランダムに非復元抽出する。子の生成では、交叉および突然変異により複数の子個体を生成する。この親と子の集団を家族と呼ぶ。家族内から最良1個体とルーレット選択により1個体を生存選択し、母集団に戻す。そのため、世代あたりの対象親選択回数(入れ替え対象となる親の数)は2となる。

MGGによる世代交代の手順を以下に示す。

- (S1) 個体群から交叉の対象となる親個体を1組選択する。
- (S2) 1組の親個体に対して交叉を行い、 N_C 個の子個体を生成する。さらに生成した個体に対して突然変異を行う。
- (S3) 子個体の関数値を評価する。世代あたりの評価回数は N_C となる。
- (S4) 全ての子個体と親個体から、次世代に残す個体の組を選択して個体群に戻す。

3.3 DE/MGGモデル

MGGの設計方針に基づいたDE世代交代モデル(DE/MGG)の設計方法について説明する。DE/MGGの世代交代モデルを図5に示す。複製選択ではMGGと同様に全ての親

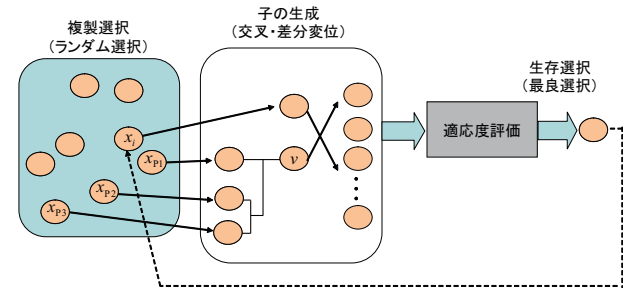


図5 DE/MGGの世代交代モデル

個体はランダムで非復元抽出する。DEでは入れ替え対象となる親は1個体のみであるため、MGGの生存選択をそのまま採用できない。また、差分操作による解探索では、個体間の差分を徐々に小さくさせる必要がある。そこでDE/MGGの生存選択では家族の中から最良1個体を選択する。子の生成では、1つの対象親個体から N_C 個の子を生成する。DE/MGGは連続世代モデルであり、世代あたりの対象親選択回数は1となる。

DE/MGGによる世代交代の手順を以下に示す。なお、 num_pair は1とする。

- (S1) 個体群から対象親個体、基本個体をそれぞれランダムで選択する。
- (S2) 差分生成個体を2体ランダムで選び、差分操作により子個体を生成し、関数値を評価する。
- (S3) 子個体数が N_C となるまで(S2)を繰り返す。世代あたりの評価回数は N_C となる。
- (S4) 対象親個体と生成された N_C 個の子を家族とし、家族の中から最良1個体を個体群の対象親個体と入れ替える。

各世代交代モデルの特徴を表1に示す。DE/MGGは、MGGと同様に連続交代モデルであるため、新たに得られた優れた個体を即座に親として子の生成に利用できる。また、子の生成では毎回差分生成個体を選びなおすため、多様な子の生成が期待できる。

表 1 各世代交代モデルの比較

	世代交代	複製選択	生存選択	子の数	入れ替え対象となる親の数
DE	離散	順番+ランダムにより非復元抽出	親子から最良 1 個体を選択	1	1
MGG	連続	ランダムにより非復元抽出	家族から最良 1 個体を選択 ルーレット選択で 1 個体を選択	N_C	2
DE/MGG	連続	ランダムにより非復元抽出	家族から最良 1 個体を選択	N_C	1

4. 提案世代交代モデル

4.1 提案モデルの概要

DE では対象親個体と基本個体の特性が子個体に引き継がれ、関数値の良い対象親個体、基本個体を選択されると、良い子個体が生成される可能性が高くなる。また、対象親個体の関数値が良く基本個体の関数値が悪い場合、基本個体の影響で子個体の関数値が対象親個体よりも悪くなり、進化できない可能性が高くなる。つまり、良い関数値を持つ個体の進化のためには個体集団全体の進化が必要となる。

DE/MGG では子を複数生成するため、対象親個体の関数値が他の親個体よりも良い場合でも進化できる可能性が高くなり、初期収束や探索の停滞に対して効果的である。しかし、その一方で関数の評価回数が多くなることが予想される。DE/MGG での関数評価回数の削減のために、対象親個体の関数値が良いときには進化の確率を高くするために多くの子を生成し、関数値が悪いときには生成する子を少なくすることが有効だと考えられる。

本論文では、個体の進化レベルに意図的に差を持たせることで、効率的に探索を行うための世代交代モデル Roulette Selection Based on Evolutionary Advance Level (REAL) を提案する。ここで、進化レベルとは子個体が親個体よりも良い解となった回数 (世代交代した回数) とし、親個体よりも良い解を生成できた割合を進化率とする。

REAL の世代交代の手順は DE/MGG と同様とし、複製選択と家族数の決定は以下で述べる方法で行う。

進化レベルによるルーレット選択

複製選択では、全ての親個体を進化レベルに比例したルーレット選択により非復元抽出する。関数値に比例したルーレット選択では選択圧が関数値の景観に依存するため、選択圧を調整し効率的な探索を行うためにはスケーリングなどの手法と組み合わせる必要がある。進化レベル比例のルーレット選択では関数値の景観による選択圧の差異はないため、さまざまな問題に適用できる。

家族数の決定方法

関数値の悪い対象親個体と関数値の良い基本個体を選択された場合、子個体は基本個体の良い形質を受け継ぐため、対象親個体と比べて関数値が良い可能性が高い。したがって、関数値の悪い対象親個体を選択された場合、家族数が少なくても世代交代が期待できる。また、進化レベルが高い個体ほど関数値が良くなる傾向があることから、対象親個体 x_i の

表 2 親個体の組合せによる解探索の特徴

	進化レベル			解探索の特徴
	対象親	基本	差分生成	
パターン 1	高	高	高 高	局所的な探索
パターン 2	高	高	高 低	大域的な探索
パターン 3	高	低	**	大域的な探索
パターン 4	低	高	**	関数値は大きく改善

子の生成数 $N_{C,i}$ は、以下の式により決定する。

$$N_{C,i} = N_C(LV_i/LV_{\max}) \quad (2)$$

ここで、 LV_i は対象親個体 x_i の進化レベル、 $LV_{\max}$ はその世代での最大の進化レベルとなる個体の進化レベルとする。

4.2 提案モデルの解析

提案手法では、進化レベルの高い個体については、対象親個体としての選択回数と子個体生成数を増やし、より良い解を探索するために用いる。子個体は対象親個体だけでなく基本個体の形質も受け継ぐため、選択された対象親よりも関数値の良い個体が多いと進化し易い。進化レベルの低い個体は対象親個体としての選択回数は少なく集団の外側に位置するが、一度の進化で大きな関数値改善が期待できるため少ない評価回数で関数値を向上させることができる。また、対象親以外の親に選ばれた際には進化レベルの高い個体の進化を助けることとなる。つまり、REAL では進化レベルにより個体の役割を区別し、進化レベルの高い個体にはより良い解の探索と進化レベルの低い個体の引き上げ、進化レベルの低い個体には進化レベルの高い個体の進化を助ける役割を担わせるという考えに基づいている。

表 2 に親個体の組合せによる解探索の特徴例を示す。ここでは 2 つの差分生成個体の進化レベルが両方とも高い場合は「高 高」、一方の進化レベルが低い場合は「高 低」とする。また「**」は両者の進化レベルを問わないものとする。選択された差分生成個体において、両者の進化レベルが高い場合は基本個体の周辺を細かく探索し、進化レベルに差がある場合は基本個体の周辺を広く探索することとなる。また、両者ともに進化レベルが低い場合でも関数の形状によっては差分は大きくなる。パターン 1 のように全ての個体の進化レベルが高い場合、その世代での優良な解付近を中心とした局所探索を行う。パターン 2 のような場合では、パターン 1 より広く優良解の周囲を探索する。また、パターン 3 のように基本個体

の進化レベルが低い場合では、優良解よりもある程度離れた範囲を探索できる．このように、進化レベルの低い個体は、その存在により集団の多様性を保ち広い範囲の探索を可能とする．パターン 4 では、前節の家族数の決定方法で述べたように、関数値が大きく改善されることが期待される．

以上より、REAL は単に DE/MGG おいて関数値によって生成する子個体数を調整するだけではなく、進化レベルに差を持たせることで関数の評価回数を少なくしようとするものである．これによって個体集団の関数値の分散は大きくなるものの、少ない探索でより良い関数値を得ることができる．

5. 数 値 実 験

5.1 実 験 設 定

本章では数値実験として、通常 DE、DE/MGG、REAL の比較を行う．各モデルの DE の設定は、文献 [高濱 08] と同様に $S=0.7$ 、 $C_R=0.95$ 、個体数 $N_P=50$ 、 num_pair は 1、 $cross$ は exp とした．子の生成数 N_C の設定基準は MGG と同様に、適応度景観の複雑さに依存すると考えられる．値が小さすぎると初期収束を起こすことが予想されるため、DE/MGG と REAL では予備実験より $N_C=20$ とした．また、生成した子個体の決定変数が定義域を超える場合は、全ての決定変数が定義域を満たすまで子の生成を繰り返す．

テスト関数は、変数間依存性、悪スケール性、大谷構造を有する問題として f_1, f_2, f_3, f_4 を用いた．表 3 に各テスト関数の特徴を示す．以下に、関数とその初期化領域を示す．また、各テスト関数では次元数 $N_D = 30$ とする．

- f_1 : Sphere 関数

$$f(x) = \sum_{i=1}^{N_D} x_i^2, \quad -5.12 \leq x_i \leq 5.12 \quad (3)$$

単峰性の関数で、点 $(0, 0, \dots, 0)$ で最小値 0 をとる．

- f_2 : Rosenbrock 関数

$$f(x) = \sum_{i=2}^{N_D} \{100(x_1 - x_i^2)^2 + (x_i - 1)^2\}, \quad (4)$$

$$-2.048 \leq x_i \leq 2.048$$

単峰性の稜構造を有する関数で、点 $(1, 1, \dots, 1)$ で最小値 0 をとる．

- f_3 : ill-scaled Rosenbrock 関数

$$f(x) = \sum_{i=2}^{N_D} \{100(x_1 - ix_i^2)^2 + (ix_i - 1)^2\}, \quad (5)$$

$$-2.048/i \leq x_i \leq 2.048/i$$

単峰性の稜構造を有する関数で、点 $(1, 1, \dots, 1)$ で最小値 0 をとる．

表 3 テスト関数の特徴

関数	変数間依存性	悪スケール性	大谷構造
f_1	なし	なし	なし
f_2	強い	なし	なし
f_3	強い	あり	なし
f_4	なし	なし	あり

- f_4 : Rastrigin 関数

$$f(x) = 10n + \sum_{i=1}^{N_D} x_i^2 - 10\cos(2\pi x_i), \quad (6)$$

$$-5.12 \leq x_i \leq 5.12$$

多峰性の大谷構造を有する関数で、点 $(0, 0, \dots, 0)$ で最小値 0 をとる．

以上の設定で、各テスト関数に対して、通常 DE、DE/MGG、REAL を各 20 回ずつ適用する．試行打ち切り条件は、評価値が 10^{-7} 以下になった場合、あるいは、評価回数が f_1 は 15×10^4 、 f_2 および f_3 は 5×10^5 、 f_4 は 4×10^5 に達した場合とする．

5.2 結 果 と 考 察

§ 1 性能比較

各関数における最適解発見までの評価回数、解発見時の個体集団の平均進化率および進化レベルの合計（総進化レベル）の平均値を表 4 から表 7 に示す．また、20 回の試行中、最適解を発見できた回数を括弧内に示す．

表 4 f_1 における性能比較

モデル	評価回数	進化率	総進化レベル
通常 DE	75903 (20)	0.142	10804
DE/MGG	130136 (20)	0.879	6023
REAL	58927 (20)	0.638	4291

表 5 f_2 における性能比較

モデル	評価回数	進化率	総進化レベル
通常 DE	381843 (20)	0.047	17872
DE/MGG	474659 (20)	0.510	12723
REAL	289486 (20)	0.371	10426

表 4 から表 7 より、各モデルは全ての試行で最適解の発見に成功している．また、全てのテスト関数において REAL の関数評価回数が最も低くなっている．DE/MGG および REAL では 1 回の差分操作で子を複数生成するため、通常 DE と比較して、より優れた子を生成する可能性が高い．そのため、総進化レベルは通常 DE よりも少なくなっていると考えられる．また、REAL では進化レベルの低い個体の家族数を減らすことで、評価回数を削減できている．

表 6 f_3 における性能比較

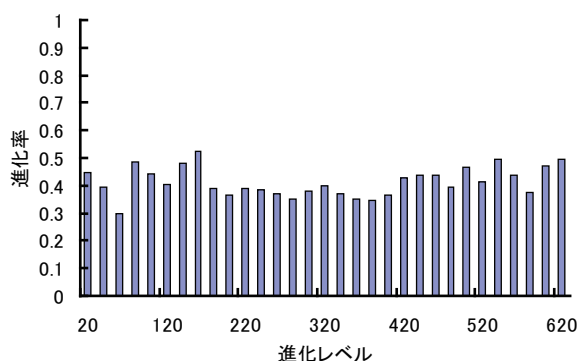
モデル	評価回数	進化率	総進化レベル
通常 DE	382628 (20)	0.047	17864
DE/MGG	468625 (20)	0.514	12659
REAL	289464 (20)	0.362	10234

表 7 f_4 における性能比較

モデル	評価回数	進化率	総進化レベル
通常 DE	263793 (20)	0.054	14193
DE/MGG	339881 (20)	0.521	9315
REAL	211612 (20)	0.366	7819

次に, REAL の f_2 における探索終盤(評価回数=200000 以降)での対象親個体の進化レベルと進化率を図 6 示す. REAL では 4.1 節で述べたように対象親個体の進化レベルが低い場合は生成する子の数は少ないが他の親個体の影響により世代交代が期待できるため, 進化率は進化レベルが高い場合とほぼ同程度となっている.

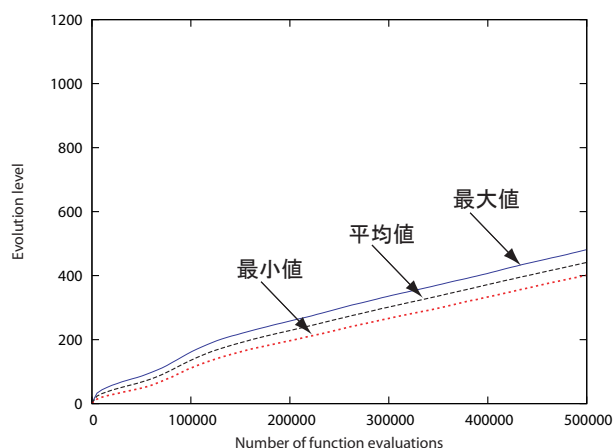
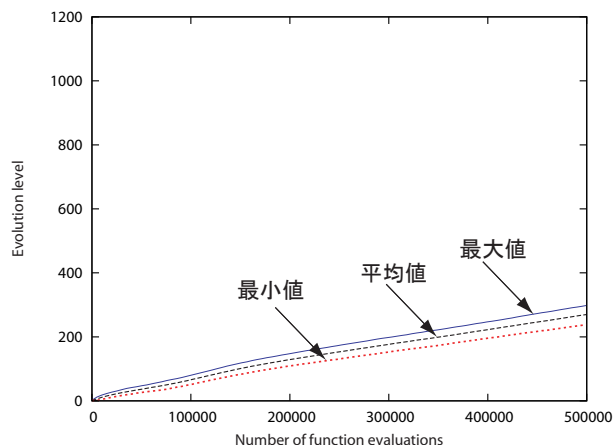
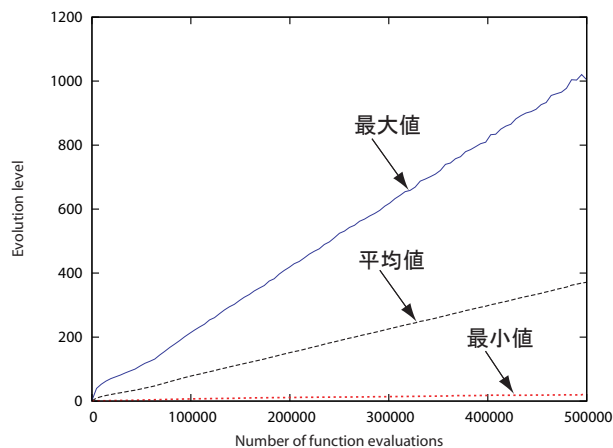
以上の結果より, REAL は通常 DE よりも探索効率の高い世代交代モデルであるといえる.

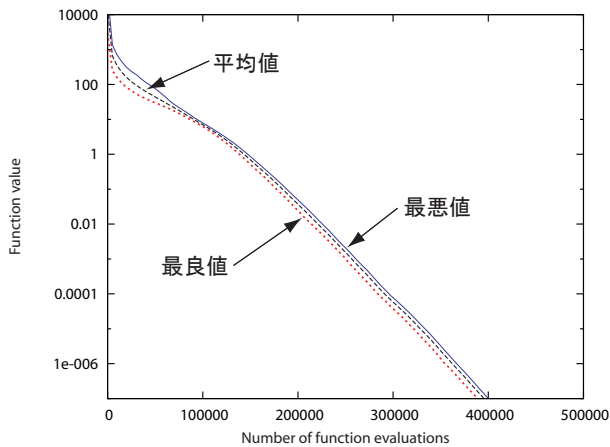
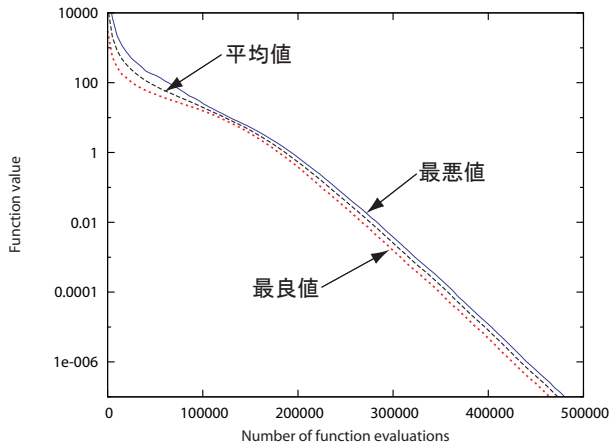
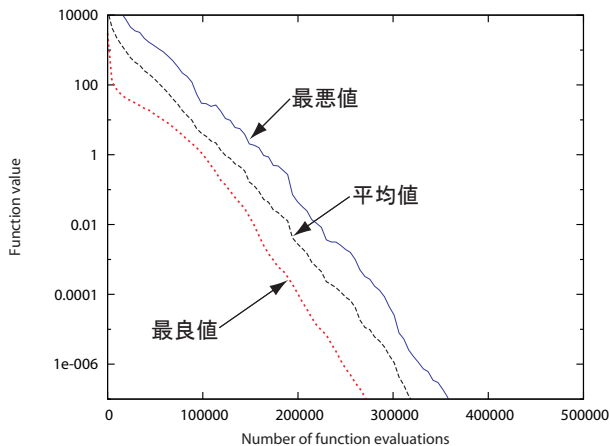
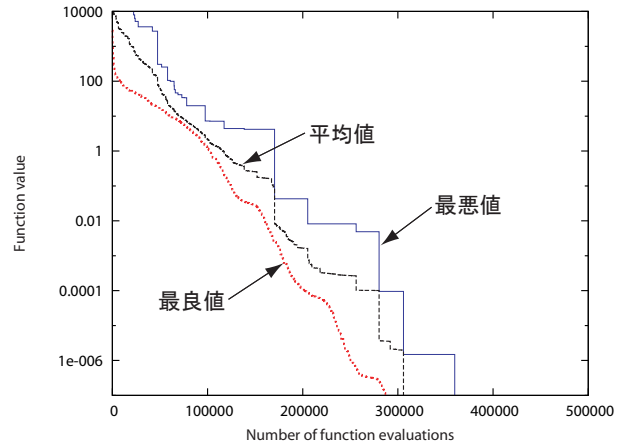
図 6 REAL の f_2 における対象親個体の進化レベルと進化率

§ 2 進化レベルと関数値

各モデルの f_2 における個体集団内の最小進化レベル, 平均進化レベル, 最大進化レベルの時間的変化を図 7 から図 9 に示す. また, 各モデルの f_2 における関数値の最良値, 平均値, 最悪値の時間的変化を図 10 から図 13 示す. なお, 時間は個体の評価回数で表す. 図 7 から図 12 は 20 試行の平均値とし, 図 13 は 1 試行のみの結果とする.

通常 DE では各世代において全ての個体が順番に対象親個体として選ばれるため, 集団内の各個体の進化レベルはほぼ均等となる. そのため各個体の関数値もほぼ均等に下降し, 最適解に収束していく. また, ランダムで親個体を選択した DE/MGG も同様の結果となっている. 一方, REAL では探索序盤は各個体の進化レベルに差はなく, 親個体はほぼランダムで選ばれる. その後, ある

図 7 DE の f_2 における進化レベル図 8 DE/MGG の f_2 における進化レベル図 9 REAL の f_2 における進化レベル

図 10 DE の f_2 における関数値図 11 DE/MGG の f_2 における関数値図 12 REAL の f_2 における関数値図 13 REAL の f_2 における関数値 (1 試行のみ)

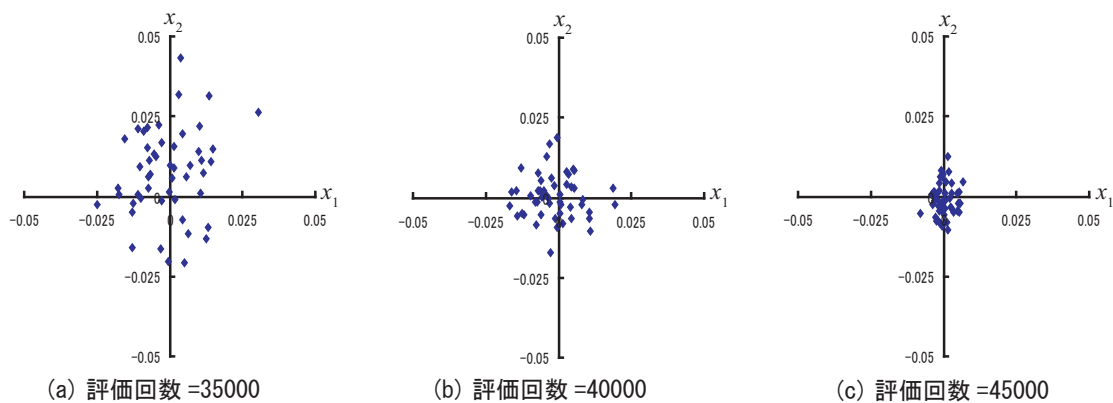
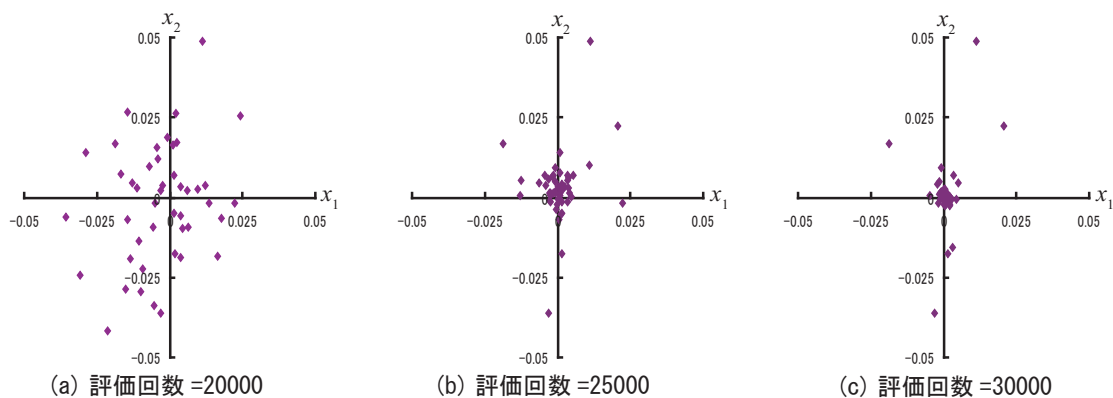
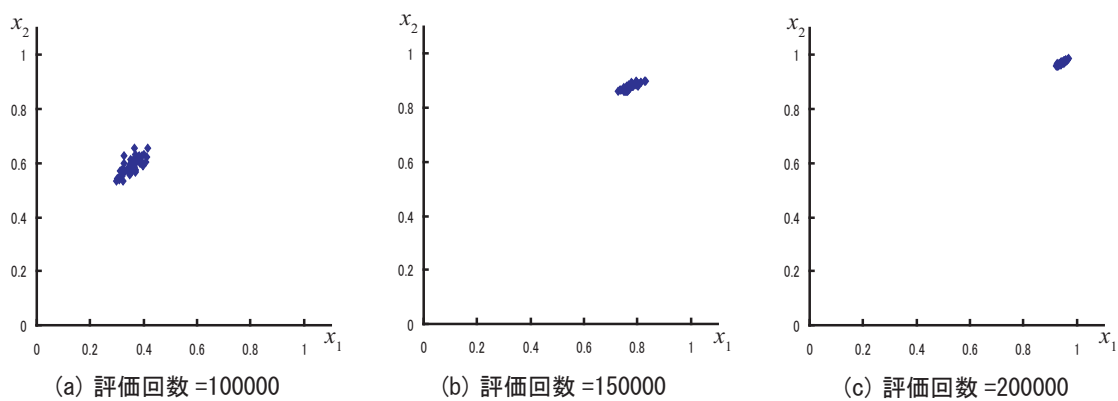
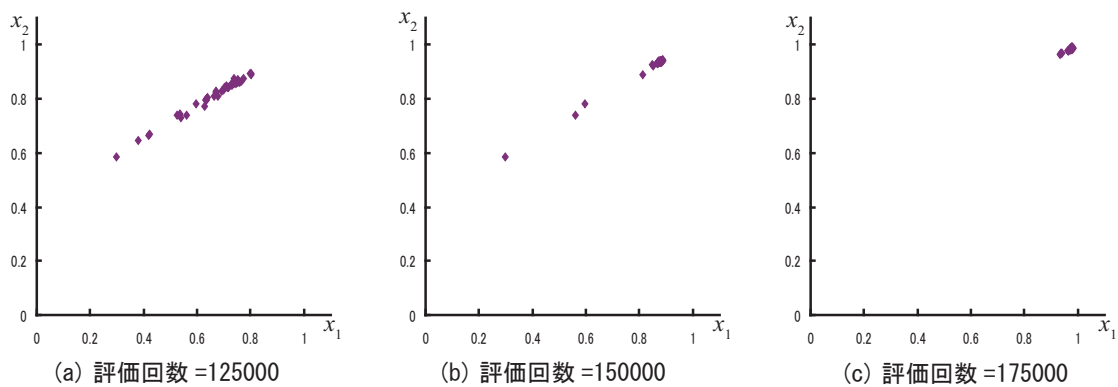
程度世代が経過すると進化レベルが大きい個体がより親個体として選ばれ易くなり、各個体の役割が区別される。図 13 において、関数値が集団内での最良値となる個体は進化レベルが高い個体であり、進化率は高く関数値の最良値は徐々に減少している。また、関数値が最悪値となる個体は進化レベルが低い個体であり対象親となる機会は低いが、1 回の世代交代での関数値が大きく減少していることが確認できる。

§ 3 個体の推移

通常 DE と REAL の f_1 における個体の推移を図 14 と図 15 に示す。なお、両モデルでは解発見までの収束の速さが異なるため、それぞれの特徴がわかり易い評価回数で個体集団の様子を比較した。

通常 DE では全ての個体が均等の速さで進化するため、個体集団は徐々に最適解 $(0, \dots, 0)$ に収束していくことが確認できる。一方、REAL では、探索序盤で進化の進んだ個体ほど、その後の進化も速いため、集団の内部ほど個体密度は高くなっていく。進化レベルの低い個体は対象親に選ばれにくく、集団全体の存在領域は通常 DE よりも広くなる。しかしながら、進化レベルの高い集団の内側の個体が親個体に使われやすいため、集団の中央を重点的に探索することとなる。一方、進化レベルの低い集団の外側の個体は対象親に使われにくい、選ばれた場合は進化レベルの高い他の親の影響で集団の中央付近を探索する。そのため、探索終盤では全ての個体はある程度最適解付近に収束している。

次に、通常 DE と REAL の f_2 における個体の推移を図 16 と図 17 に示す。通常 DE では、個体集団はまず原点付近に集まり、そこから稜に沿って個体集団の外部にある最適解 $(1, \dots, 1)$ へと移動していく。REAL においても個体集団はまず原点付近に集まるが、その後は、個体間の進化レベルの差が広がることで通常 DE よりも集団が長くなり、最適解に近い集団の先頭部分の密度が高くなる。そのため、個体間の差分(最適解への移動距離)が大きくなり、通常 DE よりも早く最適解に到達できる。

図 14 f_1 における個体の推移 (DE)図 15 f_1 における個体の推移 (REAL)図 16 f_2 における個体の推移 (DE)図 17 f_2 における個体の推移 (REAL)

また, REAL の f_2 における探索終盤 (評価回数=200000 以降) での 2 体の差分生成個体の進化レベルの差と差分の大きさの平均値を図 18 示す. 進化レベルの差が開くほど, そのときの差分は大きくなっていることが確認できる. そのため, 4.2 節で述べたように進化レベルの差が小さい場合では局所的な探索となり, 差が大きくなるほど大域的な探索となる. 親個体の選択は進化レベル比例ルーレット選択で行われるため, 差分生成個体が 2 個体とも進化レベルが高い, つまりは局所探索となる確率が高くなる.

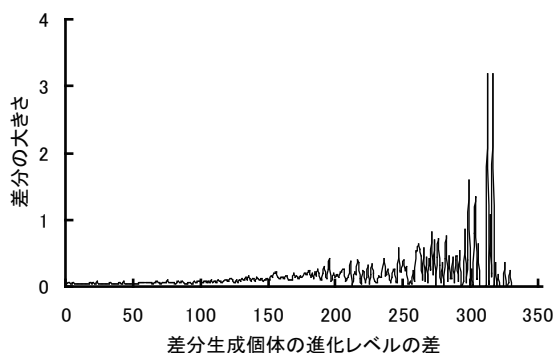


図 18 差分生成個体の進化レベルの差と差分の大きさ

6. おわりに

本稿では, GA における世代交代モデルである MGG の設計方針に基づいた DE 世代交代モデル (DE/MGG) を設計し, DE/MGG において個体の進化レベルに意図的に差を持たせた世代交代モデル REAL を提案した. REAL は連続世代モデルとし, 複製選択では, 全ての親個体を進化レベルに比例したルーレット選択により非復元抽出し, 対象親個体の進化レベルに応じて生成する子個体数を決定した.

変数間依存性, 悪スケール性, 大谷構造を有するテスト関数を用いて REAL, 通常 DE, DE/MGG を比較した結果より, REAL では全てのテスト関数において解発見までの評価回数および総進化レベルが最も低くなっていることを確認した. また, 個体集団内で進化レベルが開いていくことで各個体の役割が区別され, 進化レベルの高い個体は良い解の探索と進化レベルの低い個体の引き上げを行い, 進化レベルの低い個体は集団の多様性を保ち進化レベルの高い個体の進化を助けることを確認した.

以上より, REAL は大域探索と局所探索のバランスを損なうことなく, 探索性能を向上できる世代交代モデルであるといえる. REAL では子の生成において差分変位親個体の貢献度が大きく, そのため, 通常 DE よりも C_R の値が進化率に大きく影響すると考えられる. また, 他の発見的手法と同様に REAL においても最適性や計算量

が問題次元に影響を受けると考えられる. REAL における制御パラメータの設定基準についての考察および高次元関数での性能評価を今後の課題としたい.

◇ 参考文献 ◇

- [Noman 06] Noman, N., Iba, H.: A New Generation Alternation Model for Differential Evolution, Proceedings of 2006 Genetic and Evolutionary Computation Conference (GECCO2006), pp. 1265-1272 (2006)
- [Price 05] Price, K.V., Storn, R. and Lampinen, J.: Differential Evolution: A Practical Approach to Global Optimization. Springer-Verlag, London, UK (2005)
- [Storn 95] Storn, R. and Price, K.: Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, ICSI (1995)
- [伊藤 05] 伊藤稔, 田中雅博: 関数値最適化のための Particle Swarm Optimization, Differential Evolution, 実数値遺伝的アルゴリズムの探索性能に関する検討, 甲南大学紀要理工学編, Vol. 52, No. 1, pp. 125-135 (2005)
- [小林 09] 小林重信: 実数値 GA のフロンティア, 人工知能学会誌, Vol. 24, No. 1, pp. 147-162 (2009)
- [佐藤 97] 佐藤浩, 小野功, 小林重信: 遺伝的アルゴリズムにおける世代交代モデルの提案と評価, 人工知能学会誌, Vol. 12, No. 5, pp. 734-744 (1997)
- [高濱 08] 高濱 徹行, 阪井 節子, 原 章: 低精度の近似モデルを用いた比較推定法による Differential Evolution における関数評価回数の削減, 電子情報通信学会論文誌 D, Vol. J91-D, No. 5, pp. 1275-1285 (2008)
- [田川 06] 田川 聖治: 微分進化法における連続世代モデルの提案と分散分析による評価, 情報処理学会研究報告, MPS-71, pp. 462-468 (2008)
- [田川 08] 田川 聖治: Differential Evolution における世代交代モデルの比較研究, 進化計算シンポジウム 2008 公演論文集, pp. 23-28 (2008)
- [田中 06] 田中 雅晴, 土谷 千加夫, 佐久間 淳, 小野 功, 小林 重信: Saving MGG: 実数値 GA/MGG における適応度評価回数の削減, 人工知能学会論文誌, Vol. 21, No. 6, pp. 547-555 (2006)
- [山口 08] 山口智: Differential Evolution における制御変数の自動調節, 電気学会論文誌 C 分冊, Vol. 128, No. 11, pp. 1696-1703 (2008)

〔担当委員: 小野 功〕

2010 年 8 月 4 日 受理

◇ 付 録 ◇

A. パラメータ N_C の検証に関わる予備実験

REAL の f_2 における子の生成数 N_C を変えたときの性能を表 A.1 に示す. なお, N_C 以外のパラメータは 5.2.1 節での実験と同様である. 全て試行で探索に成功しているが, $N_C = 5$ の場合は進化率が低く通常 DE とほぼ同程度の性能となっている. N_C の増加に伴い進化率は上昇していくが $N_C = 20$ のときが最も評価回数は低い.

表 A.1 REAL の f_2 における N_C を変えたときの性能

N_C	評価回数	進化率	総進化レベル
5	366034 (20)	0.070	15965
10	318132 (20)	0.159	12708
20	289486 (20)	0.371	10426
30	318063 (20)	0.502	9683

著 者 紹 介



串田 淳一(一般会員)

2006 年 3 月立命館大学理工学研究科フロンティア理工学専攻満期退学. 2009 年より立命館大学情報理工学部助手, 現在に至る. 博士 (工学). 進化計算, 対戦型ゲームに関する研究に従事. 人工知能学会, システム制御情報学会会員.



大場 和久(一般会員)

1992 年立命館大学大学院理工学研究科博士後期課程修了. 博士 (工学). 1993 年大阪工業大学工学部電子工学科講師. 同助教授, 日本福祉大学情報社会科学部助教授を経て, 現在, 日本福祉大学健康科学部福祉工学科准教授. Differential Evolution を用いた最適化問題, 環境や災害対策についてのシステムのアプローチ, e-Learning システムなどの研究に従事. 電子情報通信学会, 計測自動制御学会, 環境経済・政策学会などの会員.



亀井 且有

1983 年 3 月立命館大学大学院博士課程後期課程修了. 同年, 立命館大学理工学部助手, 助教授を経て, 1998 年教授, 現在に至る. ソフトコンピューティングによる知能システムに関する研究に従事. 工学博士. 日本自動制御協会 (現システム制御情報学会)1987 年度榎木記念賞奨励賞受賞. 感性工学会 2007 年学会技術賞受賞.