

# 自己組織化マップを用いた遺伝的アルゴリズムの検討

沈 侃<sup>†</sup> 翟 菲<sup>†</sup> 北 栄輔<sup>†</sup>

本研究では、自己組織化マップを用いた進化的計算法である Self-Organizing Maps for Genetic Algorithm (SOM-GA) という手法を提案する。3つのテスト関数によって、提案手法の探索性能を実数値遺伝的アルゴリズム (RCGA) と比較した結果、SOM-GA が RCGA よりも良い探索結果を示すことが分かった。

## Investigation of Genetic Algorithm Using Self-organizing Map

KAN SHEN,<sup>†</sup> FEI ZHAI<sup>†</sup> and EISUKE KITA<sup>†</sup>

This paper describes Self-Organizing Maps for Genetic Algorithm (SOM-GA) which is the evolutionary algorithm by using self-organizing map. The search performance of the SOM-GA is compared with the real-coded genetic algorithm (RCGA) on three test functions. The results show that the present algorithm has better search performance than the RCGA.

### 1. はじめに

連続関数の最適化問題に対する進化的計算法として、従来の遺伝的アルゴリズムのようにビットストリングを用いる方法と異なり、設計変数値を実数値ベクトルで表現する実数値遺伝アルゴリズム (Real-coded Genetic Algorithms: RCGA)<sup>1)</sup> が熱心に研究されている。設計変数を実数値のまま扱うほうが親個体の性質を効率良く引き継ぐことができるために、効率的に解を探索できるとされている。また、様々な最適化問題では設計変数間で互いに従属性がある場合が多く、一般的な進化的計算法はそのような問題に対して収束性が悪化するなどの問題も指摘されている。

ところで、実数値設計変数をとる多目的最適化問題に対して Büche ら<sup>2)</sup> は、自己組織化マップ (Self-organizing Map: SOM)<sup>3)~5)</sup> を組み込んだ進化的計算法である Self-Organizing Maps for Multi-Objective Evolutionary Algorithms (SOM-MOEA) を提案している。SOM-MOEA は、その名称から分かるように多目的最適化問題に適用するために設計された手法である。多目的最適化問題ではパレート解を探索しなければならない。SOM-MOEA では自己組織化マップを用いてパレート解を格納することと、SOM の特長を生かして、探索パスを学習することで探索効率を改

善することなどを目的としている。SOM-MOEA では、最初に集団を構成する個体で自己組織化マップを学習する。そして、ある個体の最整合ノードを SOM マップ上で探し、その近傍ノードとの間で交叉を行って新しい個体を生成する。この方法では、交叉によって解の探索性能が向上する可能性は必ずしも高くなく、かえって性能を下げる可能性がある。著者らが、以前の研究<sup>6)</sup> において複数のテスト問題で SOM-MOEA を NSGA-II<sup>7)</sup>、SPEA2<sup>8)</sup> と比較したところ、SOM-MOEA には得意とする問題と不得意とする問題があり、従来の方法に比べて特別に優秀な探索性能を示すことはなかった。そこで、本研究では、自己組織化マップを用いるが、SOM-MOEA とは異なる概念に基づいた進化的計算法である Self-Organizing Maps for Genetic Algorithms (SOM-GA) を提案し、これを単一目的関数の最適化問題に適用する。なお、本論文では単一目的の最適化問題を解析対象としているので、多目的最適化問題のために開発された SOM-MOEA との比較は行っていない。今後、提案手法を多目的最適化問題に拡張したときに SOM-MOEA との比較を行いたいと考えている。

提案手法では、個体の設計変数値と目的関数値を入力ベクトルとして、自己組織化マップを学習させる。そして、各個体について自己組織化マップ上で最整合ノードを求め、その近傍にある複数の個体間で部分集団を構成し、そこで実数値遺伝的アルゴリズム (Real-Coded Genetic Algorithm: RCGA) を一定回数行い、

<sup>†</sup> 名古屋大学大学院情報科学研究科複雑系科学専攻  
Graduate School of Information Science, Nagoya University

得られた最良個体を次世代の個体とする。こうして構成された次世代の集団を用いて再び自己組織化マップを学習し、同様の操作を繰り返す。この操作を繰り返して、最良個体を解とする。提案手法は、自己組織化マップを用いて部分個体を作成し、部分個体の中で通常の RCGA を行うことで似通った解の付近からより良い解を探索している。本論文では、複数の試験関数において、提案手法をその基本である RCGA と比較する。

本論文の構成は以下のようになっている。2章では、研究の背景について述べる。3章では、自己組織化マップと本研究で提案する SOM-GA のアルゴリズムについて述べる。4章では、本研究が提案した SOM-GA と RCGA を 3 つの評価関数において評価比較する。5章はまとめである。

## 2. 研究の背景

本研究では、自己組織化マップの進化的計算手法への適用として、SOM-GA という方法を提案する。自己組織化マップを進化的計算と組み合わせて用いる研究としては、すでに Büche らが Self-Organizing Maps for Multi-Objective Evolutionary Algorithms (SOM-MOEA) を提案している<sup>2)</sup>。その名称から分かるように、SOM-MOEA は多目的最適化問題を対象として提案されたものである。本研究で提案する方法と簡単には比較できないが、最初に SOM-MOEA について述べてから、本研究で提案する方法について紹介し、両者の違いについて述べる。

### 2.1 Büche らの提案

Büche ら<sup>2)</sup> が提案した Self-Organizing Maps for Multi-Objective Evolutionary Algorithms (SOM-MOEA) は、多目的最適化問題においてパレート解集合を学習するために自己組織化マップを用いている。この方法では、各世代で集団を構成する個体によって自己組織化マップを学習した後、SOM 近傍進化と SOM 突然変異を用いている。なお、SOM 近傍進化と SOM 突然変異という用語は他と区別するために本論文で用いている。

SOM-MOEA のおおよそのアルゴリズムは以下のようになる。

- (1) 集団の個体について適応度を評価し、SOM マップを学習する。
- (2) 以下のようにして SOM 近傍進化を行う。
  - (a) ランダムに個体を選択する。
  - (b) 選択された個体に対して SOM マップ上での最整合ノードを選択する。

- (c) 最整合ノードの近接ノードを選択する。
  - (d) 近接ノードに最整合する個体と元の個体の内挿点として新たな個体を生成する。
- (3) 以下のようにして SOM 突然変異を行う。
    - (a) 正規乱数を加えることで、個体に突然変異を与える。

SOM 近傍進化では、SOM マップ上で、選択された個体の最整合ノードを選択し、それに近接するノードに属する個体との間で 1 回だけ交叉を行う。SOM マップで近接するノードどうしは一般に類似していると想像されるので、SOM 近傍進化は各個体の近傍を探索することになる。そこで、近傍以外のところの探索を行うために SOM 突然変異が用いられる。SOM 突然変異では、個体に乱数を加えることで、SOM 近傍進化では得られない個体を生成することができ、解の多様性を改善して、近傍以外のところでの探索を進めることになる。

SOM を用いれば多数の個体を類似の個体ごとに分類することができるので、適切に利用すれば類似個体間での探索から局所探索を速め、最適解への収束を速める可能性がある。しかし、Büche らの方法では、選択された個体の近傍から選択される個体が解の性能を改善するように適切に選択される保証がない。その結果、従来の方法に比べて有効な探索性能を示すということにはなかったと考えられる。

### 2.2 提案手法の概要

Büche らの提案を複数の試験関数でテストした結果から、著者らは本研究において自己組織化マップを用いるが異なるアルゴリズムに基づく Self-Organizing Maps for Genetic Algorithms (SOM-GA) を提案する。

提案手法の概念図を図 1 に示す。また、処理を簡潔書きにすると以下ようになる。

- (1) 元の集団に実数値遺伝的アルゴリズム (Real-Coded Genetic Algorithm: RCGA) を適用して別の集団を作る (図中の ①)。
- (2) 個体の設計変数値と目的関数値を入力ベクトルとして、SOM マップを学習させる (②)。これにより、類似の個体は SOM マップ上で近くに写像されることになる。
- (3) 各個体の SOM マップ上での最整合ノードを探索する (③)。
- (4) 最整合ノードの近傍にマップされている複数の個体で部分集団を構成する (④)。
- (5) 各部分集団に RCGA を適用する (⑤)。
- (6) 各部分集団で得られた最良個体を次世代の個体

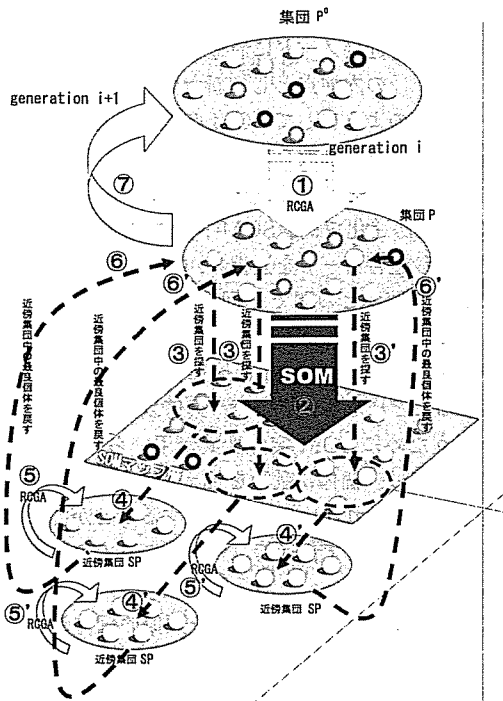


図1 SOM-GA 概念図

Fig.1 SOM-GA conceptual diagram.

とする (6)。

- (7) 新たに構成された集団を用いて同様の操作を繰り返す (7)。
- (8) 最終的に得られた最良個体を解とする。

SOM-MOEA で行われる SOM 近傍進化では、選択された個体間で交叉は 1 回しか行われぬ。交叉される個体はランダムに選択されるので適切に選択されないと、新たな個体は親個体に比べて改悪される可能性があり、近傍探索が進まないことになる。これに対して、提案手法では、自己組織化マップを用いて集団全体から複数の部分集団を作成し、部分集団ごとに RCGA を行う。複数の個体で部分集団を構成し、そこで一定回数の探索を行うことで、近傍探索をある程度収束させることができるので、新たに生成された個体が改悪に進むことは起こらず、近傍探索を十分進めることができる。この結果、SOM-MOEA よりも精度良い探索が可能となる。

提案手法では、部分集団に対して適用される RCGA には BLX- $\alpha$  交叉と実数値突然変異を用いている。そこで、性能比較には BLX- $\alpha$  交叉と実数値突然変異を用いた RCGA を採用する。しかし、アルゴリズムは他の交叉法や突然変異を用いた RCGA に容易に拡張できる。

### 3. SOM-GA アルゴリズム

ここでは、自己組織化マップと SOM-GA のアルゴリズムについて述べる。

#### 3.1 自己組織化マップ (SOM)<sup>3)~5)</sup>

自己組織化マップは動物や人間の視覚に関する細胞で行われる自己組織化過程の研究に基づいており、教師なし学習を経てパターン間の関係を表出させる 2 層ネットワークである。第 1 層は入力層であり、第 2 層は競合層 (マップ層) と呼ばれる。これを用いることで、多次元データを 2 次元のマップに写像し、拡散した情報に含まれる潜在的な知識が獲得できる。本研究では、ノードを六角形に配置した 2 次元の SOM マップを用いている。

SOM マップの  $i$  番目のノードにおかれた参照ベクトルを

$$w^i = \{w_1^i, w_2^i, \dots, w_n^i\}^T, \quad (1)$$

$j$  番目の入力ベクトルを

$$v^j = \{v_1^j, v_2^j, \dots, v_n^j\}^T \in \mathcal{R}^n \quad (2)$$

と定義する。

ノルムとしてユークリッド距離  $\|v^j - w^i\|$  をとり、入力  $v^j$  に対する最整合ノードをノルムを最小にするノードとして選択する。これを  $c(v^j)$  とすると、次式で定義できる。

$$c(v^j) = \arg \min \{\|v^j - w^i\|\} \quad (3)$$

ひとたび  $c(v^j)$  が決定されると、入力ベクトルに近づくように参照ベクトルは次式で更新される。

$$w^i(t+1) = w^i(t) + h_{ci}(t) \{v^j(t) - w^i(t)\} \quad (4)$$

ここで、 $t = 0, 1, 2, \dots$  は離散時間である。また、 $h_{ci}(t)$  は近傍関数と呼ばれ、格子点上に定義される平滑効果カーネルとして作用する。収束するためには、 $t \rightarrow \infty$  において  $h_{ci}(t) \rightarrow 0$  であることが必要であるので、次式のような関数として定義される。

$$h_{ci}(t) = \alpha_s \exp \left( -\frac{\|r_c - r_i\|^2}{2\sigma^2(t)} \right) \quad (5)$$

ここで、 $r_i$  はノード  $i$  の位置ベクトル、 $r_c$  は最整合ノードの位置ベクトルである。 $\alpha_s$  は学習率と呼ばれ、 $0 < \alpha_s < 1$  にとられる。 $\alpha_s$  を時間の単調減少関数とする場合もあるが、本研究では定数としている。また、 $\sigma(t)$  は時間の単調減少関数として定義され、次式のように与える。

$$\sigma(t) = \sigma(t-1) - \frac{R}{TS} \quad (6)$$

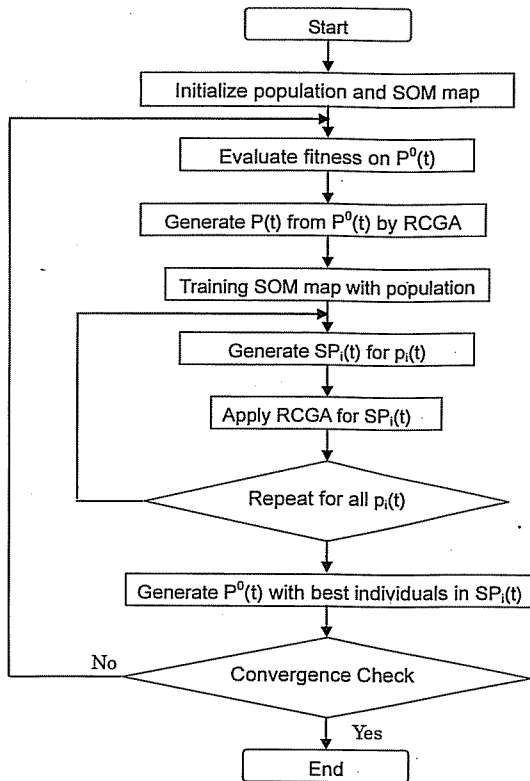


図2 SOM-GA アルゴリズムフローチャート  
Fig.2 Flow chart of SOM-GA algorithm.

ここで、 $TS$  は学習ステップ回数、 $R$  は初期半径である。また、 $\sigma(0) = R$ 、 $\sigma(TS) = 0$  である。

SOM アルゴリズムの流れは以下ようになる。

- (1) SOM の入力層とマップ層間の初期重みをランダムに設定する。
- (2) 入力サンプルからランダムに選択して、マップ層で入力ベクトルとの距離を計算し、式 (3) を満たすノード  $c(v^j)$  を選択する。
- (3) 式 (4) により参照ベクトルを更新する。
- (4) 式 (5) により近傍関数を更新する。
- (5) ステップ (2) へ戻る。

### 3.2 提案手法のアルゴリズム

提案手法のアルゴリズムは以下ようになる。また、フローチャートを図2に示す。以下において、世代  $t$  における集団を  $P(t)$ 、そこに属する個体を  $p_i(t)$  ( $i = 1, 2, \dots, M$ )、探索の過程で作成する部分集団を  $SP(t)$  とする。また、自己組織化マップを  $Map$  とする。

- (1)  $t = 0$  とする。
- (2) 自己組織化マップ  $Map$  を初期化する。
- (3) ランダムに生成した  $M$  個の個体から初期集団

$P^0(t)$  を構成する。

- (4) 個体の適応度を評価する。
- (5) 集団  $P(t)$  を生成する。
  - (a) 集団  $P^0(t)$  からルーレット選択によって2個体を選択し、BLX- $\alpha$  交叉により新たな個体を生成する。
  - (b) (5a) の操作を繰り返して、 $M$  個の個体からなる集団  $P(t)$  を生成する。
- (6) 集団  $P(t)$  に属する全個体によって  $Map$  を学習する。このとき、入力ベクトルとして各個体の設計変数値と目的関数値をとる。
- (7) 以下の操作により新たな個体を生成する。
  - (a)  $Map$  上において、個体  $p_i(t)$  を中心としてユークリッド距離が  $Rn$  以内に含まれるすべての個体で部分集団  $SP_i(t)$  を構成する。
  - (b)  $SP_i(t)$  において RCGA を  $N_g$  回行い、得られた最良個体を  $p_i(t+1)$  とする。
  - (c)  $p_i(t)$  ( $i = 1, 2, \dots, M$ ) について上記の操作を行い、 $p_i(t+1)$  ( $i = 1, 2, \dots, M$ ) を得る。
- (8) 最良個体が必要な精度を有していれば出力して、処理を終える。そうでなければ、以下の処理を行う。
- (9) 個体  $p_i(t+1)$  ( $i = 1, 2, \dots, M$ ) により次世代の集団  $P^0(t+1)$  を構成する。
- (10)  $t \leftarrow t+1$  として、ステップ (4) へ戻る。

## 4. 数値実験

ここでは、Rastrigin 関数、Rosenbrock 関数、Griewank 関数の3つの評価関数を用いて、本研究で提案した SOM-GA の探索性能を RCGA と比較する。提案手法である SOM-GA の中で用いる RCGA、さらに、比較に用いる RCGA においても、そのアルゴリズムの中で交叉法に BLX- $\alpha$  を、突然変異に実数値突然変異を用いる。

### 4.1 評価関数

#### 4.1.1 Rastrigin 関数

Rastrigin 関数を式 (7) で定義する。設計変数が2個の場合のグラフを図3に示す。

$$F(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$$

$$(-5.12 < x_i < 5.12) \quad (7)$$

$$\min(F(x)) = F(0, 0, \dots, 0) = 0$$

Rastrigin 関数は、最適解の周辺に格子状に準最適

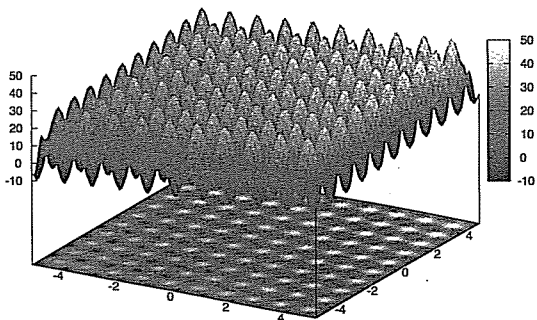


図3 2次元のRastrigin関数  
Fig.3 Rastrigin Function in two-dimension.

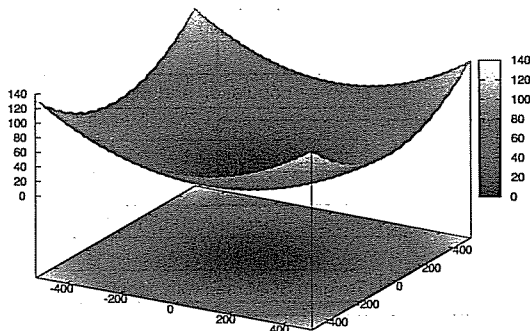


図5 2次元のGriewank関数  
Fig.5 Griewank Function in two-dimension (large).

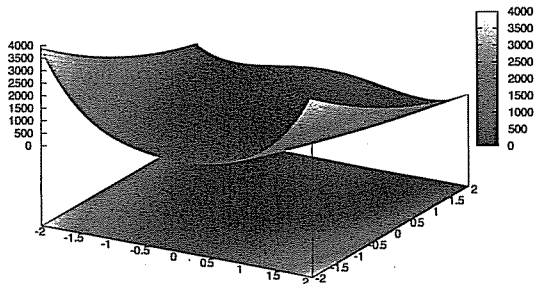


図4 2次元のRosenbrock関数  
Fig.4 Rosenbrock Function in two-dimension.

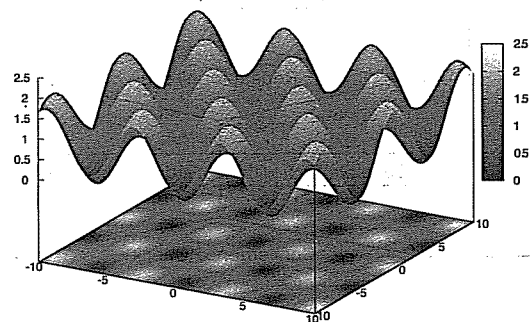


図6 2次元のGriewank関数(最適解近傍)  
Fig.6 Griewank Function in two-dimension (Near optimum).

解を持つ多峰性関数である。また設計変数間に依存関係がない問題である。

4.1.2 Rosenbrock 関数

Rosenbrock 関数を式 (8) で定義する。設計変数が 2 個の場合のグラフを図 4 に示す。

$$F(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2) \quad (8)$$

(-2.048 < x<sub>i</sub> < 2.048)

min(F(x)) = F(1, 1, ..., 1) = 0

Rosenbrock 関数は設計変数間に依存関係を持つ単峰性関数である。また、この関数は一般的に GA にとって解き難い問題といわれる。

4.1.3 Griewank 関数

Griewank 関数を式 (9) で定義する。設計変数が 2 個の場合のグラフを図 5 に、最適解近傍の様子を図 6 に示す。

$$F(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \left( \cos\left(\frac{x_i}{\sqrt{i}}\right) \right) \quad (9)$$

(-512 < x<sub>i</sub> < 512)

min(F(x)) = F(0, 0, ..., 0) = 0

Griewank 関数は設計変数間に依存関係を有する多峰性関数である。大域的には単峰性関数のような性質

表 1 共通パラメータ

Table 1 Common parameters.

|       |                                |
|-------|--------------------------------|
| 設計変数  | 20                             |
| 個体数   | 50                             |
| 選択方法  | ルーレット選択                        |
| 交叉方法  | BLX- $\alpha$ , $\alpha = 0.5$ |
| 交叉率   | 1.0                            |
| 突然変異率 | 0.002                          |
| 評価回数  | 10 回                           |

を持つため、準最適解は比較的容易に求めることができる。しかし、局所的には多数の局所最適解が存在するため、最適解を探索するのは困難である。

4.2 パラメータ設定

SOM-GA と RCGA に用いる共通パラメータの設定値を表 1 に示す。各最適化問題において設計変数は 20 個である。個体集団は 50 個体からなり、交叉率 1.0、突然変異率 0.002 としている。選択法としてルーレット選択、交叉法として BLX- $\alpha$  を用いる。また、本研究で用いる突然変異では、設計変数の定義域内でランダムに値を変更する方法を用いる。

SOM-GA のパラメータ設定を表 2 に示す。ここで、部分集団の近傍半径  $R_n = 2$  は、各個体を含む部分集団を定義するための、自己組織化マップ上でのユーク

表 2 SOM-GA パラメータ  
Table 2 SOM-GA parameters.

|              |                  |
|--------------|------------------|
| SOM マップ      | 20 × 20 の六角型     |
| 初期近隣半径       | $R = 10$         |
| 学習率          | $\alpha_s = 0.8$ |
| 学習回数         | $TS = 1000$      |
| 部分集団のための近傍半径 | $Rn = 2$         |
| 部分集団進化世代数    | $N_g = 1000$     |

表 3 最大世代数  
Table 3 Maximum generation step.

|        | Rastrigin | Rosenbrock | Griewank  |
|--------|-----------|------------|-----------|
| RCGA   | 1,000,000 | 20,000,000 | 1,000,000 |
| SOM-GA | 20        | 200        | 20        |

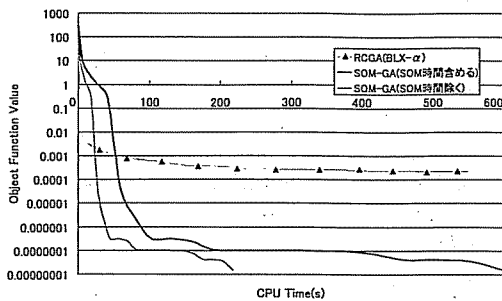


図 7 Rastrigin 関数の収束結果  
Fig. 7 The result of Rastrigin function.

リッド距離である。部分集団進化世代数  $N_g = 1000$  は、部分集団において RCGA を実行する回数を示す。

各手法における最大世代数を表 3 に示す。SOM-GA では、1つの世代で1個体ごとに部分集団を構成し、そこで  $N_g = 1,000$  回の RCGA を実行する。今回は個体数 50 なので、1つの世代ごとに  $50 \times 1000 = 50,000$  回の RCGA を行うことになる。そこで、RCGA と SOM-GA の進化的処理の回数をそろえるために、

$$\begin{aligned} & \text{RCGA の最大世代数} \\ & \geq \text{SOM-GA の最大世代数} \times 50,000 \end{aligned}$$

となるように設定している。

### 4.3 実験結果

#### 4.3.1 Rastrigin 関数

解析結果を図 7 に示す。縦軸には各世代の最良個体の目的関数値を、横軸には計算に使用した CPU 時間を示している。グラフには、RCGA の収束状況、本研究で提案した SOM-GA の収束状況に加えて、SOM-GA の処理時間から SOM 学習時間を除いた場合の結果を示す。この図から、RCGA は最初の収束が速いが、目的関数値が 0.001 あたりをすぎてから収束速度が遅くなるのが分かる。SOM-GA では最初の収束速度は RCGA に劣るものの、目的関数値が 0.001 あたりか

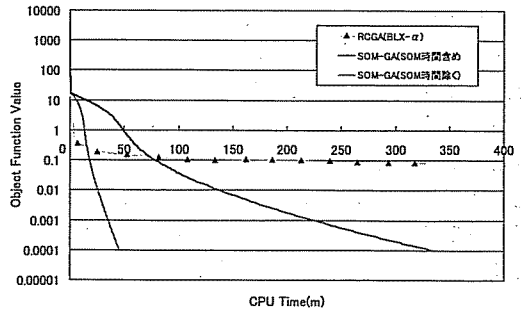


図 8 Rosenbrock 関数の収束結果  
Fig. 8 The result of Rosenbrock function.

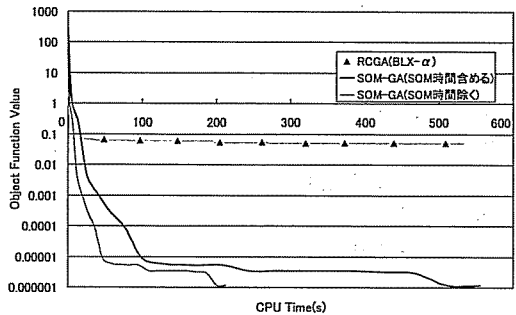


図 9 Griewank 関数の収束結果  
Fig. 9 The result of Griewank function.

ら 0.0000001 あたりまでの収束速度が速く、CPU 時間が 50 秒あたりをすぎると、RCGA よりかなり大域的最適解に近い値に収束することが分かる。

#### 4.3.2 Rosenbrock 関数

解析結果を図 8 に示す。縦軸には各世代の最良個体の目的関数値を、横軸には計算に使用した CPU 時間を示している。図 8 から、RCGA では 0.1 あたりまでしか収束できないのに対して、SOM-GA では 0.0001 ぐらいまで収束できることが分かる。

しかし、先の例題に比べると計算時間は長いことが分かる。これは Rosenbrock 関数が単峰性であり、提案した手法では近傍個体で部分集団を構成して探索を行うので、最適解に到達するまでに毎世代で少しずつしか進化できないため、収束速度が遅くなると考えられる。

#### 4.3.3 Griewank 関数

解析結果を図 9 に示す。縦軸には各世代の最良個体の目的関数値を、横軸には計算に使用した CPU 時間を示している。この問題では、提案した SOM-GA の収束速度が、RCGA に比べてかなり速いことが分かる。そして、RCGA では目的関数値が 0.1 あたりからあまり収束していないが、SOM-GA では目的関数値が 0.00001 あたりまで収束できることが分かる。

Griewank 関数は設計変数間に依存関係を有する多峰性関数である。大域的には単峰性関数のような性質を持つため準最適解は比較的容易に求めることができるが、局所的には多数の準最適解が存在し、最適解を発見するのは困難となる。SOM-GA では、自己組織化マップの学習により、多峰性関数の設計変数間の近傍関係を容易に発見することができ、そのために効率的に最適解を探索することができたと考えられる。

## 5. ま と め

本研究では、多峰性のある連続関数の最適化問題に対する進化的計算法として、自己組織化マップを用いた遺伝的アルゴリズムである Self-Organizing Maps for Genetic Algorithms (SOM-GA) を提案した。提案手法では、各世代において集団で自己組織化マップを学習し、集団から複数の部分集団を構成する。そして、部分集団に実数値遺伝的アルゴリズム (RCGA) を適用して新たな個体を生成する。この処理を繰り返すことで解を探索する。

本研究では Rastrigin, Rosenbrock, Griewank の 3 つのテスト関数に対して、提案した SOM-GA とそのもととなった RCGA を比較した。計算結果より、Rastrigin と Griewank 関数において、SOM-GA が RCGA より良い最適解を探索できたことが分かった。また、Rosenbrock 関数に対して、SOM-GA は RCGA より良い解を探索できたが、他のテスト関数よりも長い計算時間が必要であることが分かった。SOM-GA では、同一グループにある個体間での探索は親個体を結ぶ軸の近傍にある最適個体を探索する。そのため、SOM-GA は多峰性関数に対して良い探索性能を示した。しかし、異なるグループにある個体間での探索を行わないので、単峰性関数では最適個体の探索速度が遅くなる。

今後は、Rosenbrock 関数のように単峰性関数の問題における収束速度を改善することと、自己組織化マップの学習時間を減少させるためにアルゴリズムを改良する必要があると考えている。さらに、実用的な問題に対する手法の応用についても研究を進めていく予定である。そのために、まず多目的な最適化問題に適用できるように提案アルゴリズムを改良することが必要と考えている。

## 参 考 文 献

- 1) Eshelman, L.J. and Schaffer, J.D.: Real-coded genetic algorithms and interval schemata, *Foundation of Genetic Algorithms 2*, Whitley,

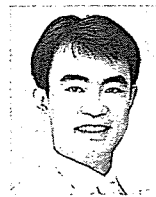
L.D. (Ed.), pp.187-202, Morgan Kaufmann Publications (1992).

- 2) Büche, D., Milano, M. and Koumoutsakos, P.: Self-organizing maps for multi-objective optimization, *GECCO 2002: Proc. Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference*, Barry, A.M. (Ed.), New York, AAAI, pp.152-155 (2002).
- 3) T. コホネン (著), 徳高平蔵, 岸田 悟, 藤村喜久郎 (訳): 自己組織化マップ, シュプリンガー・フェアラーク東京 (1996).
- 4) 徳高平蔵, 岸田 悟, 藤村喜久郎: 自己組織化マップの応用, 海文堂 (1999).
- 5) マーク M. ヴァン・フッレ (著), 徳高平蔵, 岸田 悟, 藤村喜久郎 (訳): 自己組織化マップ—理論・設計・応用, 海文堂 (2001).
- 6) 沈 侃, 北 栄輔: 自己組織化マップを用いた進化的最適化手法の性能評価について, 情報処理学会研究報告, No.56, pp.5-8 (2006).
- 7) Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, *Proc. Parallel Problem Solving from Nature VI*, Schoenauer, M., et al. (Eds), pp.849-858 (2000).
- 8) Zitzler, E., Laumanns, M. and Thiele, L.: Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization, *EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, Giannakoglou, K., et al. (Eds), Athens, Greece, September 12-21, pp.21-24 (2001).

(平成 18 年 11 月 22 日受付)

(平成 18 年 12 月 28 日再受付)

(平成 19 年 4 月 21 日採録)



沈 侃

1977 年生。名古屋大学大学院情報科学研究科博士課程後期課程在学中。自己組織化マップと進化的計算等基礎的研究、および金融・経済問題への応用研究に従事。

**翟 菲**

1980年生。名古屋大学大学院情報科学研究科博士課程前期課程在学中。マルチエージェントを用いた人工市場モデル，行動ファイナンス理論に関する研究に従事。

**北 栄輔 (正会員)**

1964年生。1991年名古屋大学大学院工学研究科博士課程後期課程修了。博士(工学)。1999年より名古屋大学助教授。2007年より准教授に配置換え，現在に至る。セル・オートマトン (Cellular Automata)，進化的計算手法，数値解析法 (BEM, Trefftz 法) 等の研究に従事。著書に，偏微分方程式の数値解法，計算のための線形代数，Trefftz 法入門等。IEEE, ISBE, 応用数理学会，日本機械学会，シミュレーション学会，日本計算工学会，知能情報ファジィ学会，情報文化学会，日本図学会各会員。