

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

沈 侃[†] 北 栄 輔[†]

進化的多目的最適化手法について幾つかの方法が提案されている。本研究では、自己組織化マップ (Self-Organizing Map, SOM) を組み込んだ進化的多目的最適化手法である Self-Organizing Maps Multi-Objective Evolutionary Algorithms (SOM-MOEA) に着目する。SOM-MOEA は比較的最近提案された方法で、その特性について検討はなされていない。そこで、本研究では NSGA-II や SPEA2 と比較することで、SOM-MOEA の特性を調べる。また、SOM-MOEA と SPEA2 を同時に同一の集団に適用する方法についても述べる。解析結果より、SOM-MOEA の探索性能は扱う問題に比較的依存することがわかった。また、SOM-SPEA2 は SOM-MOEA の問題点を改善できることを確認した。

Performance Estimation of Evolutionary Algorithm Using Self-Organizing Map

SHEN KAN[†] and EISUKE KITA[†]

Several multi-objective evolutionary algorithms have been presented by many researchers. This paper focuses on Self-Organizing Maps Multi-Objective Evolutionary Algorithms (SOM-MOEA), which employs Self-Organizing Maps (SOM). Since SOM-MOEA has been presented recently, its performance is not discussed. In this paper, the SOM-MOEA is compared with the SPEA2 and NSGA-II. Besides, this paper also focuses on SOM-SPEA2 at which SOM-MOEA and SPEA2 are applied simultaneously to the same population. Numerical results indicate that the performance of SOM-MOEA depends on the problems to be solved and besides, that SOM-SPEA2 can overcome the disadvantage of the SOM-MOEA.

1. はじめに

多目的最適化問題は「互いに競合する複数の目的関数を与えられた制約条件の中で最小化する問題」と定義される。目的関数が互いに競合しあっているため、与えられた全ての目的関数に対して完全最適解を求めることは一般に難しい。そこで、多目的最適化問題では「ある目的関数の値を改善するためには、少なくとも他の1つ目的関数の値を改悪せざるをえないような解」であるパレート最適解 (Pareto optimal solution) を求めることになる。

進化的アルゴリズム (Evolutionary Algorithms, EAs) [1] とは、生き物の進化の過程に着想を得た計算手法の総称である。進化的アルゴリズムでは問題の解を同時並列に探索するので、パレート解を求める多目的最適化問題に適した手法と考えられている [1]。特に最近では、NSGA を改良した NSGA-II [2] と

SPEA の改良アルゴリズムである SPEA2 [3] が注目されている。これに対して、Buche ら [4] は、進化的最適化手法に自己組織化マップ (Self-Organizing Maps, SOM) を組み込んだ Self-Organizing Maps Multi-Objective Evolutionary Algorithms (SOM-MOEA) を提案している。この方法では、SOM の特性を生かして、突然変異ステップサイズや探索パスを学習しながら探索を行う。しかし、他の手法との比較はまだ十分には行われていない。そこで、本論文では、最初に複数のテスト関数を用いて SOM-MOEA を SPEA2 や NSGA-II と比較し、その特徴を検討する。また、SOM-MOEA と SPEA2 を組み合わせて利用する手法についても述べる。この方法は、集団からランダムに選択された一部の個体に SPEA2 を、残りに SOM-MOEA を適用しながら、進化を進めていくという方法である。これを複数のテスト関数において他の手法と比較する。

2. アルゴリズム

ここでは、最初に SOM について説明した後、SOM-

[†] 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

MOEA について紹介する。

2.1 自己組織化マップ (SOM) [5-7]

SOM は、 m 個のベクトル (ニューロン) が h 次元に結合されたマップに対する、入力データ区間 R^n の写像を定義する。SOM マップの i 番目のノードにおかれた参照ベクトルを

$$\mathbf{m}^i = \{m_1^i, m_2^i, \dots, m_n^i\}^T, \quad (1)$$

j 番目の入力ベクトルを

$$\mathbf{x}^j = \{x_1^j, x_2^j, \dots, x_n^j\}^T \in R^n \quad (2)$$

と定義する。

入力 $\mathbf{x}^j$ に対する最整合ノードは、ノルムとしてユークリッド距離 $\|\mathbf{x}^j - \mathbf{m}^i\|$ をとり、これを最小にするノードとする。つまり、

$$c(\mathbf{x}^j) = \arg \min_i \{\|\mathbf{x}^j - \mathbf{m}^i\|\} \quad (3)$$

$c(\mathbf{x}^j)$ 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。

$$\mathbf{m}^i(t+1) = \mathbf{m}^i(t) + h_{ci}(t) \{\mathbf{x}^j(t) - \mathbf{m}^i(t)\} \quad (4)$$

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、 $t \rightarrow 0$ 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。

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

ここで、 r_i はノード i の位置ベクトルである。 $\alpha(t)$ は学習率係数と呼ばれ、 $0 < \alpha(t) < 1$ にとられる。 $\alpha(t)$ 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。

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

ここで、 T_S は学習ステップ、 R は初期半径である。また、 $\sigma(0) = R, \sigma(T_S) = 0$ となる。

SOM アルゴリズムの流れは以下になる。

- (1) ネットワークの初期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) を満たすノードを選択する。
- (3) 式 (4) により重みを学習する。
- (4) 式 (6) により重み付け半径を更新する
- (5) ステップ (2) へ戻る。

2.2 SOM-MOEA [4, 5]

各タイムステップの最良解を用いて SOM をトレーニングすることで、有望な進化の方向性をトラッキン

グすることができる。また、SOM がステップサイズを適応的に変化させるので、有望な解の領域に探索範囲を狭めていくことになり、収束速度の改善が期待できる。SOM-MOEA では、SGA における交叉、突然変異の代わりに、SOM 近傍進化法と SOM 突然変異を用いる。また、それ以外のアルゴリズムについては、他の GA アルゴリズムを用いることができるとされている。そこで、本研究では、Buche の研究 [4] に従い、SPEA2 を基本として、それに SOM アルゴリズムを組み合わせていることとする。

2.2.1 SOM-MOEA アルゴリズム

SOM-MOEA のアルゴリズムを以下に示す。

- (1) $t \leftarrow 0$
- (2) 初期化：初期個体集団 P_0 を生成し、アーカイブ Q_0 を空集合とする。
- (3) SOM 学習：設計変数空間で SOM を学習させる。
- (4) 適応度計算： P_t と Q_t の各個体の適応度を評価する。
- (5) 環境選択： P_t と Q_t にある非優位な個体を P_{t+1} へコピーする。そして、 Q_{t+1} がアーカイブサイズを超えた場合は Q_{t+1} のサイズを小さくする。
- (6) SOM 近傍進化：トーナメント戦略により親を選択し、SOM 近傍進化を行う。
- (7) SOM 突然変異： P_{t+1} に SOM 突然変異を適用する。

2.2.2 適合度

適合度を次式で定義する。

$$Fitness(i) = \sum_{j \in P_t + Q_t, j \succ i} s(j) \quad (7)$$

$$s(i) = |\{j \in P_t + Q_t \wedge i \succ j\}| \quad (8)$$

すべての個体 i に対して、優越している個体数を調べ、これを $s(i)$ とする。個体 i の適合度は、個体 i に優越するすべての個体の $s(j)$ の合計とする。従って、どの個体からも優越されない個体の適合度は 0 となる。

2.2.3 SOM 近傍進化法

SOM 近傍進化法では、まず、SOM アルゴリズムを用いて親個体の集団について SOM を学習させる。続いて、SOM マップ上の格子点とその近傍からランダムに選ばれた点から、新たな個体を生成する。具体的には以下になる。

- (1) SOM マップ上の格子点 $\mathbf{m}^i$ を選択する。
- (2) SOM マップ上で $\mathbf{m}^i$ を中心として半径 $\frac{R}{T_S}$ の中からランダムに 1 点を選択し、これを $\mathbf{m}'$ とする。
- (3) 新たな個体 $\mathbf{x}^i$ を次式により生成する。

表 1 パラメータ設定

Table 1 Parameters for algorithms

個体数	$N = 100$
アーカイブサイズ	100
最大世代数	3000(SPH2, ZDT) 500(KUR)
コーディング	20 ビットの 2 進数 (NSGA-II, SPEA2) 実数 (SOM-MOEA, SOM-SPEA2)
交叉率	1.0
突然変異率	0.005(NSGA-II, SPEA2) 0.01(SOM-MOEA, SOM-SPEA2)
SOM マップ	20×20 の六角型
初期 SOM 近隣半径	$R = 15$
SOM 学習率	$\alpha = 0.04$
SOM 学習回数	$T_g = 100$
SOM 近傍進化法	$\beta = 0.6$
SOM-SPEA2	$\omega = 0.2$

$$\mathbf{x}^i = \beta \mathbf{m}^i + (1 - \beta) \mathbf{m}^i \quad (9)$$

ここで、パラメータ β は $0 < \beta < 1$ にとられる。

2.2.4 SOM 突然変異

SOM-MOEA では、SOM マップでの外に新しい個体を生じるために突然変異を用いる。この操作は、一様乱数を新しい点に加えることにより行う。個体 i の設計変数ベクトルを $\mathbf{x}^i$ とし、SOM 突然変異により更新された後の設計変数ベクトルを $\hat{\mathbf{x}}^i$ とすると、SOM 突然変異は次式で定義される。

$$\hat{\mathbf{x}}^i = \mathbf{x}^i + \sigma (\mathbf{m}^i - \mathbf{x}^i) \quad (10)$$

ここで、 σ は SOM 学習の近傍半径 $\frac{R}{T_g}$ から学習ステップによってランダムで選択されたノードのエッジのユークリッド長さである。

2.3 SOM-SPEA2

本研究では、SOM-MOEA の性能を改善するための単純な方法についても述べる。この方法は、同一の集団に SOM-MOEA と SPEA2 を平行して適用するという単純で、適用しやすい方法である。以下では仮に SOM-SPEA2 と呼ぶことにする。

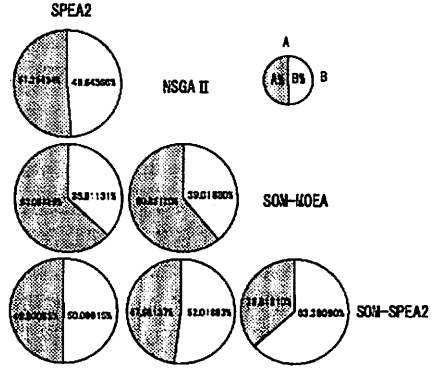
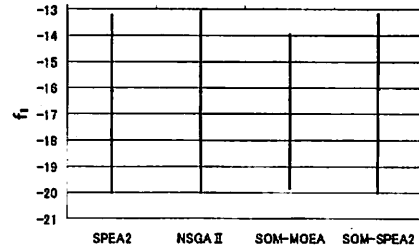
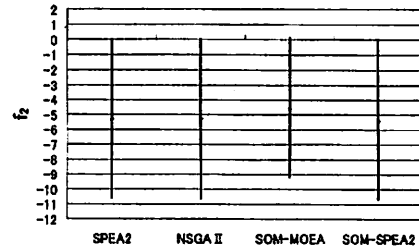
集団の総個体数を N とすると、この方法では、各世代において ωN に SOM-MOEA を $(1 - \omega)N$ に SPEA2 を適用する。ここで $0 < \omega < 1$ にとられる。また、それぞれのアルゴリズムを適用される個体はランダムに選択する。

3. 数値実験

例題として KUR 問題 [8] を採り上げる。この問題の Pareto フロントは不連続性を有している。設計変数 $\mathbf{x} = \{x_1, x_2, x_3\}^T$ に対して、目的関数 $f_1(\mathbf{x}), f_2(\mathbf{x})$ が次式で定義される。

表 2 I_{cover} の比較 (KUR 問題)Table 2 Comparison of I_{cover} on KUR problem

SOM-SPEA2	0.4874166
SPEA2	0.4870833
SOM-MOEA	0.4666667
NSGA2	0.4452833

図 1 I_{RNI} の比較 (KUR 問題)Fig. 1 Comparison of I_{RNI} on SPH2 problem(a) 関数 f_1 (b) 関数 f_2 図 2 I_{MMA} の比較 (KUR 問題)Fig. 2 Comparison of I_{MMA} on KUR problem

$$\min f_1(\mathbf{x}) = \sum_{i=1}^2 \left[-10e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}} \right] \quad (11)$$

$$\min f_2(\mathbf{x}) = \sum_{i=1}^3 (|x_i|^{0.8} + 5 \sin(x_i^3)) \quad (12)$$

解析に用いたパラメータ設定を表1に示す。

解析は異なる初期値から30回実行した。

I_{cover} の比較を表2に示す。被覆率(I_{cover})は、得られた非劣解を絶対的に評価する手法であり、パレート最適フロントに対する幅広さ、隙間のなさの評価を行う。数値が1に近いほど良い解であるといえる。これにより、SPEA2とSOM-SPEA2が同程度でもっとも良い値を示しており、わずかにSOM-SPEA2のほうがよいといえる。続いて、SOM-MOEA、NSGA2であるといえる。

I_{RNI} の比較を図1に示す。二つ以上の非劣解集合の比較に基づく評価手法であり、相手に対して非劣である解の数を比較する事が出来る。この評価方法は2つ以上の手法で得られた非劣解集合を足し合わせて、その中より非劣解を選び出す。この指標から判断すると、SOM-MOEAはSPEA2やNSGA-IIに劣るが、SOM-SPEA2は、SPEA2やNSGA-IIと同等以上の性能を示していることがわかる。

目的関数 f_1, f_2 の I_{MMA} を、それぞれ図2(a),(b)に示す。 I_{MMA} は、得られた非劣解における各目的関数軸の最大値、最小値、平均値をあわせてプロットすることにより、得られた非劣解の幅広さの評価を行うことができる。これより、SOM-MOEAが得ている解の範囲が最も狭く、それ以外の方法では同程度の収束号を得ているといえる。

4. ま と め

本研究では、自己組織化マップを組み入れた進化的最適化法であるSOM-MOEAと、それをSPEA2と組み合わせて利用する手法について述べた。SOM-MOEAではSOMの特長を生かしたSOM近傍進化法とSOM突然変異を用いる。しかし、それ以外のアルゴリズムには従来の手法が利用できる。そこで、本研究では、提案者[4]に従ってSPEA2を基本とし、それにSOM近傍進化法とSOM突然変異を組み合わせたSOM-MOEAを用いた。試験関数としてKUR問題を用いて、探索性能をNSGA-IIやSPEA2と比較し、KUR問題については良い探索性能を示していることがわかった。また、SOM-MOEAをSPEA2と組み合わせて利用する方法について以下のことがいえる。この方法は、SOM-MOEAとSPEA2を平行して実行するという方法で、非常に単純で実装しやすい手法である。SOM-MOEAが得意な問題ではSOM-MOEAアルゴリズムが有効に作用し、SPEA2が得意な問題ではSPEA2アルゴリズムが有効に作用することが予

想される。しかし、 I_{cover} 値に示されるようにSPEA2やSOM-MOEAよりもよい探索データを示す場合もあることがわかった。

参 考 文 献

- 1) K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.
- 2) K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, 2000.
- 3) Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.
- 4) D. Buche, M. Milano, and P. Koumoutsakos. Self-organizing maps for multi-objective optimization, 2002.
- 5) T. コホネン (著), 徳高平蔵, 岸田悟, 藤村喜久郎 (訳) (編). 自己組織化マップ. シュプリンガー・フェアラーク東京, 1996.
- 6) 徳高平蔵, 岸田悟, 藤村喜久郎 (編). 自己組織化マップの応用. 海文堂, 1999.
- 7) マーク M. ヴァン・フッレ (著), 徳高平蔵, 岸田悟, 藤村喜久郎 (訳) (編). 自己組織化マップ—理論・設計・応用. 海文堂, 2001.
- 8) Gregorio Toscano Pulido and Carlos A. Coello Coello. The micro genetic algorithm 2: Towards online adaptation in evolutionary multiobjective optimization. In C.M. Fonseca, P.J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Second International Conference on Evolutionary Multi-Criterion Optimization*, pp. 252–266, 2003.