

メモリと予測を用いた種分化を行う遺伝的アルゴリズムによる動的に変化する多峰性問題の最適化

Memory and Prediction Based Genetic Algorithm Using Speciation in Dynamic Multimodal Function Optimization

市村匠¹⁾, 井上寛²⁾, 原章²⁾, 高濱徹行²⁾

Takumi Ichimura¹⁾, Hiroshi Inoue²⁾, Akira Hara²⁾, Tetsuyuki Takahama²⁾

¹⁾ 県立広島大学経営情報学部

¹⁾ Faculty of Management and Information Systems, Prefectural University of Hiroshima

²⁾ 広島市立大学院情報科学研究科

²⁾ Graduate School of Information Sciences, Hiroshima City University

Abstract: It is difficult problems for Evolutionary Algorithms to search an optimal solution in multimodal functions with dynamic environments, where individuals search more than one optima and their fitness value changes under such environments. In this paper we propose a method of Memory and Prediction Based Genetic Algorithm Using speciation. This method is extended with a case-based memory and a meta-learner for precise prediction of environmental change. Especially, the individuals in a memory consist of 4 kinds of predictors and they can adjust to the change of dynamic environment adaptively. To verify the effectiveness, the method is examined to search optimal solutions in multimodal functions.

1 はじめに

動的で多峰性のある関数最適化問題 [1] において、過去の最良解を保存する事例ベースメモリを用いた遺伝的アルゴリズム (GA) による探索手法 [2] がある。また、多峰性の問題に対しては種分化を GA に適用する手法 [3] がある。本研究では、メモリに保存された個体を用いて将来の環境における最良解を種ごとに予測する機能により、動的に環境が変化する問題においても、比較的高い適応度をもつ個体を維持したまま適応的にすべてのピークを発見する手法を提案する。

2 事例ベースメモリ

事例ベースメモリは、各世代での最良個体を保存し、現在の最適個体が進化過程により破壊されないようにエリートを蓄え、図1のように適切なタイミングでそれらを個体集団へ再挿入する手法である。周期 dt で環境が変化し、世代 i 及び $(i + dt)$ で環境の変化が起こったとすると、変化する環境の最良個体をメモリに保存する。個体集団へ再挿入するメモリ内の個体には、個体集団内からランダムに1個以上選択しメモリ内の個体と入れ替える「ランダム」、メモリ内の個体を現在の環境で評価し、集団内の悪い個体とメモリ内の最良個体を入れ替える「最良個体」、線形予測アルゴリズムを用いて、次世代で良い個体と予測される個体と集団の悪い個体を入れ替える「予測」などの方法がある。

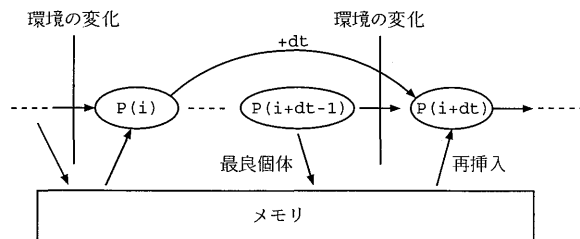


図1: 事例ベースメモリを用いた進化の過程

3 種の支配者と縄張り

複数の峰が存在する多峰性問題の最適化に GA を用いると、1つの峰に個体が集中してしまい、全ての峰を見つけにくい状態がある。これに対し、種分化の手法 [3] を用いることにより、それぞれの峰ごとに種を形成し、種ごとに遺伝的操作を行うことにより、複数の峰を発見することが可能となる。種の中で最も適応度の高い個体のことを種の支配者と呼び、種は図2のように、種の支配者を中心に縄張り半径 r_s の円の中にある個体によって形成される。また個体間の距離は式 (1) によって求められる。

$$dist(\bar{x}^{(i)}, \bar{x}^{(j)}) = \sqrt{\sum_{k=1}^D (x_k^{(i)} - x_k^{(j)})^2} \quad (1)$$

ここで、 $\bar{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_D^{(i)})$, $\bar{x}^{(j)} = (x_1^{(j)}, x_2^{(j)}, \dots, x_D^{(j)})$ であり、 D は次元数である。

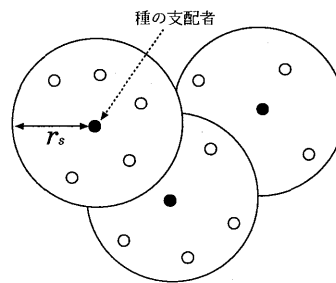


図2: 種の形成

図3は、種の支配者を定めるアルゴリズムである。

input : L_{sorted} (適応度でソートされた個体のリスト)
output : S (種の支配者のリスト)

begin

```

S を空にする;
while  $L_{sorted}$  が空になるまで do
     $L_{sorted}$  の先頭の個体  $p$  を取り出す;
    found ← FALSE;
    for  $\forall s \in S$  do
        if  $d(s, p) \leq r_s$  then
            found ← TRUE;
            break;
        end
    end
    if not found then
         $S \leftarrow S \cup \{p\}$ ;
    end
end
end
    
```

図 3: 種の支配者を決めるアルゴリズム

4 メモリと予測を用いた GA

周期的な環境の変化に対応するために、GA にメモリと予測器を適用する手法 (Memory and Prediction based GA: MPGA) を提案する。これは環境が変化する度に前の世代の最良個体をメモリに保存し、連続的にメモリに蓄えられている個体から予測を行うという手法である。ここでは、環境を特徴に応じて分類し、用いる予測器を適応的に選択できるようにした。リスト構造のメモリには遺伝子表現の個体を保存するが、その数の上限のために、新しい個体を追加するとき、メモリ内の個体を削除する必要がある場合には、保存した時系列で古いものから削除する。なお本研究では、環境における個体間の距離とその成す角度から 3 個の予測器と 1 個の適応度に関する予測器を用意した。

- (i) M_i から M_{i+1} ($i = 1, \dots, n-1$) へのそれぞれのベクトル \mathbf{v}_i を作成する。

$$\mathbf{v}_i = M_{i+1} - M_i \quad (2)$$

- (ii) \mathbf{v}_i のそれぞれの長さ l_i を計算する。

$$l_i = \|\mathbf{v}_i\| \quad (3)$$

- (iii) $X_d X_{d+1}$ -平面 ($d = 1, \dots, D-1$) において、 \mathbf{v}_i から \mathbf{v}_{i+1} ($i = 1, \dots, n-2$) へのそれぞれの角度の変化 $\Delta\theta_{i,d}$ を計算する。

$$\Delta\theta_{i,d} = \arccos \frac{v_{(i+1)d}}{\sqrt{v_{(i+1)d}^2 + v_{(i+1)d+1}^2}} - \arccos \frac{v_{id}}{\sqrt{v_{id}^2 + v_{id+1}^2}} \quad (4)$$

- (iv) l_i の変化の平均 $\overline{\Delta l}$ と $\Delta\theta_{i,d}$ の平均 $\overline{\Delta\theta_d}$ を計算する。

$$\overline{\Delta l} = \frac{1}{n-2} \sum_{i=1}^{n-2} (l_{(i+1)} - l_i),$$

$$\overline{\Delta\theta_d} = \frac{1}{n-2} \sum_{i=1}^{n-2} \Delta\theta_{i,d} \quad (5)$$

- (v) 次世代への変化を $l_{new} = l_{n-1} + \overline{\Delta l}$, $\Delta\theta_{new_d} = \Delta\theta_{(n-2),d} + \overline{\Delta\theta_d}$ とする。

- (vi) $l_{new}, \Delta\theta_{new_d}$ から、次世代への変化ベクトル

\mathbf{v}_{new} を作成する。
(vii) \mathbf{v}_{new} と M_n, M_{n-1} から、個体を予測して生成する。

図 4: 予測器 1 のアルゴリズム

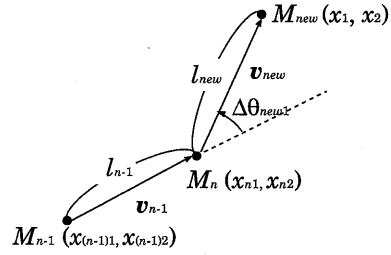


図 5: 2次元での予測ベクトルの作成

メモリに保存されている個体を M_i ($i = 1, 2, \dots, n$) とした。ここで n はメモリに保存されている個体数で、 D は次元数である。図 4 は予測アルゴリズムである。図 5 は、 $D = 2$ のときの変化ベクトルと予測の例で、 l_{new} は変化ベクトル \mathbf{v}_{new} の長さ、 $\Delta\theta_{new_1}$ は角度、 (x_{n1}, x_{n2}) , $(x_{(n-1)1}, x_{(n-1)2})$ はそれぞれ個体 M_n, M_{n-1} の座標である。予測により生成される個体 M_{new} の座標 (x_1, x_2) は式 (6) で表わされる。

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_{n1} \\ x_{n2} \end{pmatrix} + \begin{pmatrix} \frac{l_{new}}{l_{n-1}} & 0 \\ 0 & \frac{l_{new}}{l_{n-1}} \end{pmatrix} \times \begin{pmatrix} \cos \Delta\theta_{new_1} & -\sin \Delta\theta_{new_1} \\ \sin \Delta\theta_{new_1} & \cos \Delta\theta_{new_1} \end{pmatrix} \mathbf{v}_{n-1} \quad (6)$$

さらに、 D 次元では式 (7) のようになる。

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix} = \begin{pmatrix} x_{n1} \\ x_{n2} \\ \vdots \\ x_{nD} \end{pmatrix} + L \cdot R_{D-1} \cdot R_{D-2} \cdot \dots \cdot R_1 \cdot \mathbf{v}_{n-1} \quad (7)$$

ここで L と R_i は式 (8), (9) で与えられる。

$$L = \begin{pmatrix} \frac{l_{new}}{l_{n-1}} & & & 0 \\ & \frac{l_{new}}{l_{n-1}} & & \\ & & \ddots & \\ 0 & & & \frac{l_{new}}{l_{n-1}} \end{pmatrix} \quad (8)$$

$$R_i = \begin{pmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & & \\ & & & \cos \Delta\theta_{new_i} & -\sin \Delta\theta_{new_i} \\ & & & \sin \Delta\theta_{new_i} & \cos \Delta\theta_{new_i} \\ & & & & & 1 & \\ & & & & & & \ddots & \\ & & & & & & & & 1 \end{pmatrix} \quad (9)$$

予測器 2 は $\Delta\theta_{new_d}$ を $-\Delta\theta_{new_d}$ として、予測器 1 とは対称的な位置に個体を生成し、予測器 3 は $\Delta\theta_{new_d} = 0$ とし、 \mathbf{v}_{n-1} の直線上に個体を生成する。

個体を予測する際は、これらの予測器のうちどれか1つを用いる。予測器はそれぞれスコア値を持ち、この値に基づいてルーレット選択により予測器を選択する。このスコア値は、予測後の最良個体の適応度と環境が変化する前の最良個体の適応度を比較し、予測の効果に基づき、予測が有効であればスコア値を1増加し、無効であれば1減少させるような強化学習により更新される。

環境の変化が予測しにくい場合でも適応して動作するために、予測によって生成された個体の半径 R 以内で個体をランダムに生成し、それらと個体集団の一部を適応度の低い個体から順に置き換える。これにより、予測による最良解の周辺にいくつかの個体を配置することができ、たとえ予測が外れたとしても解への収束を早めることができる。環境 T における R を R_T とすると、次式になる。

$$R_T = \frac{1}{n} \sum_{t=T-(n-1)}^T \sqrt{\sum_{i=1}^D (e_{ti} - e_{ti}^*)^2} \quad (10)$$

ここで e_t^* は環境 t での予測解、 e_t は環境 t で得られた最良解である。

5 種分化を取り入れた MPGA

さらに、多峰性問題に適用するため、MPGA のメモリを種ごとに用意し、予測や個体の再挿入も種ごとにを行う。この手法を Memory and Prediction based GA using Speciation: SMPGA とする。種分化の手法のみでは、種の数に対する制限が加えられていなければ種は増加する一方であり、結果としてメモリ数も増加する。このため、図6に示すような、種の足切り処理を行う。種の支配者を適応度の高い順にソートし、低い方の適応度が高い方の適応度の50%以下ならば、そこで足切りを行う。

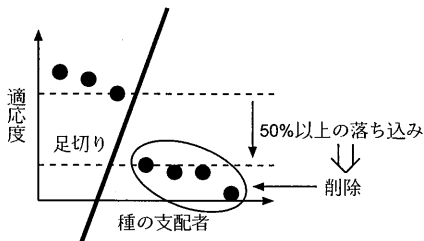


図 6: 種の足切り

1 回目の種の足切りの際に残った種数を K とし、2 回目以降の足切りの際に上位 K 個の種を残して足切りを行う。また環境の変化が起こった際には、変化する前の環境の種の支配者を保存する。どの支配者をどのメモリに保存するかは、メモリ内の最も新しい個体と支配者の位置を比べて、最も距離に近いものから順に各メモリに1個体ずつ保存する。さらに種に属する個体の数を一定にするために、1つの種に属する個体の数を(個体集団)/(種の数)とし、その数より多い種に対しては適応度の高い個体のみを残し、それ以外は削除する。そうでない種に対しては種の縄張り内にランダムに生成した個体を追加する。またMPGAと同様に、予測によって生成された個体の半径 R 以内で個体をランダムに生成し、それらと個体集団の一部を適

応度の低い個体から順に置き換える。環境 T における R を R_T とすると、

$$R_T = \frac{1}{nK} \sum_{k=1}^K \sum_{t=T-(n-1)}^T \sqrt{\sum_{i=1}^D (e_{tki} - e_{tki}^*)^2} \quad (11)$$

となる。ここで D は次元数であり、 n はメモリに保存されている個体の数、 K はメモリの数、 e_{tk}^* は環境 t でのメモリ k による予測解、 e_{tk} は環境 t で得られたメモリ k に対応する局所的な最良解である。

5.1 SMPGA のアルゴリズム

図7は、SMPGA のアルゴリズムである。

- (i) 初期個体集団をランダムに生成する。
- (ii) 集団内の全ての個体を評価する。
- (iii) 種分化を行う。
- (iv) 種の足切り処理を行う。
- (v) 種の均衡を保つ処理を行う。
- (vi) 環境が変化していた場合、以下の操作を行う。
 - (a) 前の世代の種の支配者を種ごとにメモリに保存する。
 - (b) 種ごとに予測を用い、種の最悪個体と置き換える。
 - (c) R を適応的に変化させる。
 - (d) 種ごとに種の一部を予測の個体を中心に半径 R の円の中に新しく生成した個体で再初期化する。
- (vii) 環境が変化しておらず、前の世代で予測をしているとき、以下の操作を行う。
 - (a) 予測が外れているとき、以下の操作を行う。
 - i. 前回使った予測器のスコアを更新(減少)する。
 - ii. スコアに基づいて予測器を切り替える。全ての予測器を試した場合は予測しない。
 - iii. 種ごとに予測を用い、種の最悪個体と置き換える。
 - iv. 個体集団の一部を予測の個体を中心に半径 R の円の中に新しく生成した個体で再初期化する。
 - (b) 予測が外れていないとき、前回使った予測器のスコアを更新(増加)する。
- (viii) それぞれの種に、遺伝的操作を行う。
 - (a) 個体が m 個以下なら、個体数が m になるまで種の縄張り内にランダムに生成した個体を追加する。
 - (b) 生成した子個体が種の支配者と同じなら、子個体をランダムに生成した個体と置き換える。
- (ix) 全ての個体の中から、次世代に残す N 個体を選ぶ。
- (x) 終了条件を満たしていなかったら、ii. に戻る。

図 7: SMPGA のアルゴリズム

6 実験

本研究では、環境の変化が起こると峰の位置、幅、高さが変わる動的な多峰性問題 [1] を取り扱う。テスト関数は以下の式である。

$$F(\vec{x}, t) = \max_{i=1 \dots 5} \frac{H_i(t)}{1 + W_i(t) \sum_{j=1}^D (x_j - X_{ij}(t))^2} \quad (12)$$

ここでピーク i の世代 t での高さ $H_i(t)$ と幅 $W_i(t)$ は Δe 世代毎に次のように変化する。

$$\sigma \in N(0, 1) \quad (13)$$

$$H_i(t) = H_i(t-1) + 7 \cdot \sigma \quad (14)$$

$$W_i(t) = W_i(t-1) + 0.01 \cdot \sigma \quad (15)$$

またノイズは $\{(x_1, x_2) | x_1^2 + x_2^2 \leq 1\}$ の乱数を変化後の位置に追加した。それぞれのピーク的位置と高さ H と幅 W は表 1 により初期化される。遺伝子は 20 ビットで 1 つの次元の座標を表しており、1 個体あたり $20 \times D$ ビットの遺伝子を持つ。また遺伝子型から表現型への変換にはグレイコードを用いており、表現型の変数 x_i の範囲は $0 \leq x_i \leq 100$ である。実験で用いたパラメータは、個体集団の大きさ: 400, 世代数: 300, 交叉率: 0.6, 突然変異率: 0.1, $\Delta e = 10$, メモリの大きさ: 15, $R = 5$ (初期値), 次元数 $D = 2, 3, 5$ とした。

表 1: 各ピークの初期値

	X_1	X_2	X_3	X_4	X_5	W	H
peak1	8.0	64.0	67.0	55.0	4.0	0.1	50.0
peak2	50.0	13.0	76.0	15.0	7.0	0.1	50.0
peak3	9.0	19.0	27.0	67.0	24.0	0.1	50.0
peak4	66.0	87.0	65.0	19.0	43.0	0.1	50.0
peak5	76.0	32.0	43.0	54.0	65.0	0.1	50.0

図 8, 9, 10, 11 は, 5 次元における最良個体の適応度の変化, オフラインパフォーマンス, 各ピークに対する適応度の変化, 各種に属する個体数をそれぞれ示している。

7 おわりに

単峰性で動的に変化する問題に対しては, 環境の変化が起こる度に適応度が急激に落ち込んだが, 提案した MPGA では, 変化後の最良解を予測することによって適切に進化を行うことができた。複数の峰が存在する動的な多峰性問題で実験を行ったところ, 種分化を取り入れている手法では峰の変化には追従できなかったが, 提案した SMPGA の手法では, 種の足切り処理を行うことで各峰に対応した種を形成することができ, 種の均衡を保つ処理により各種の中でメモリと予測を用いて適切に進化を行うことができた。

参考文献

- [1] Jürgen Branke, "Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems", Proc. of CEC99, Vol.3, pp.1875-1882 (1999)
- [2] J. Eggermont, T. Lenaerts, S. Poyhonen, and A. Termier, "Raising the Dead; Extending Evolutionary Algorithms with a Case-based Memory", Proc. of EuroGP'01, vol. 2038 of LNCS. Springer-Verlag, pp.280-290 (2001)
- [3] Xiaodong Li, "Efficient Differential Evolution using Speciation for Multimodal Function Optimization", Proc. of GECCO05, pp.873-880 (2005)

問合せ先

〒734-8558 広島市南区宇品東一丁目 1-71
 県立広島大学経営情報学部
 市村 匠
 E-mail: ichimura@pu-hiroshima.ac.jp

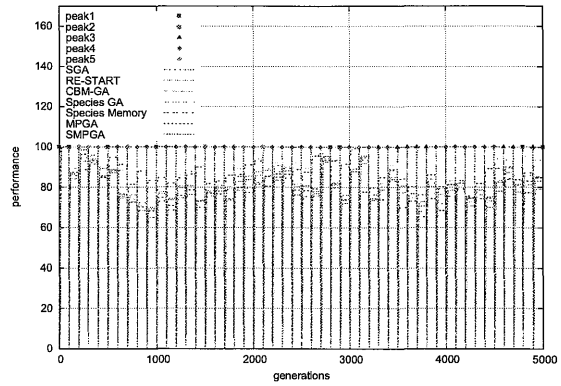


図 8: 最良個体の適応度

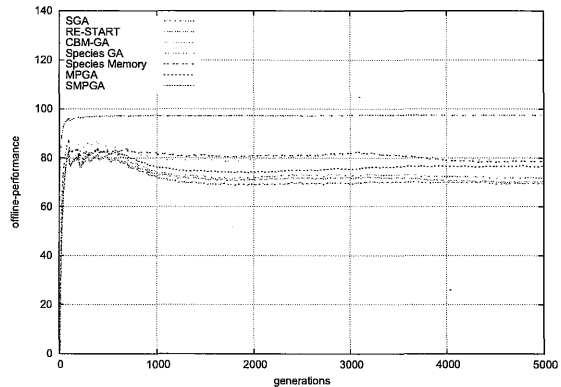


図 9: オフラインパフォーマンス

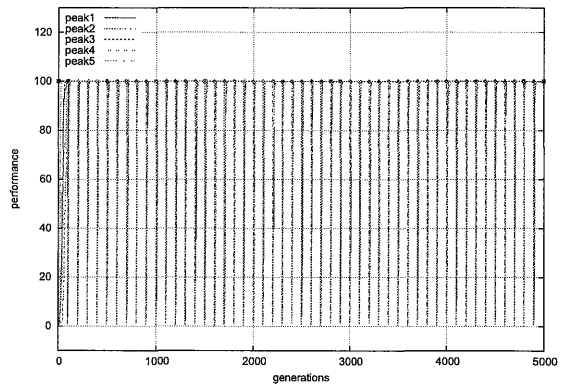


図 10: SMPGA による峰の適応度

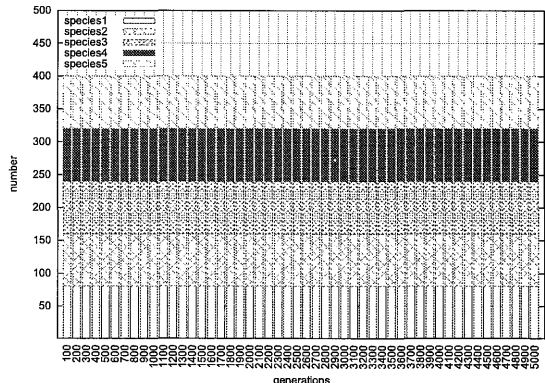


図 11: SMPGA による種の個体数