

多様な確率モデルを持つ実数値分布推定アルゴリズムの提案

中尾 昌広

1 はじめに

進化計算の新しいモデルとして、良好な評価値を持つ探索点群 (集団) の統計情報から確率モデルを構築し、新たな探索点のサンプリングを行う分布推定アルゴリズム (EDA: Estimation of Distribution Algorithm)^{1, 2)} と呼ばれる研究が提案されている。EDA は遺伝的アルゴリズム (GA: Genetic Algorithm)³⁾ における「交叉と突然変異」という操作を、「良好な探索点の分布推定とサンプリング」という操作に拡張したものとも考えることもできる。

本発表では、多様な複数の確率モデルを持つ新しい実数値 EDA (RMM: Real-coded EDA using Multiple Probabilistic Models) を提案する。RMM は既存の EDA である PBILc (Continuous Population-Based Incremental Learning)⁴⁾ と HIS (Hierarchical Importance Sampling)⁵⁾ から着想を得たアルゴリズムである。

2 分布推定アルゴリズム

2.1 一般的な分布推定アルゴリズムの枠組み

一般的な EDA の手順は下記の通りである。

1. 初期集団をランダムに生成する
2. 集団の中から評価値が良好な探索点 (例えば集団中の上位 $K\%$ の探索点) を選択する
3. 選択した良好な探索点から統計情報を獲得し、確率モデルを構築する
4. 確率モデルから新しい探索点のサンプリングを行い、現在の集団と置換する
5. 終了条件を満たすまで (2) に戻る

一般的な EDA では、構築する確率モデルと集団はそれぞれ 1 種類のみを用いる。そのため、解探索中に集団がある範囲に収束し、かつその範囲の外側に最適解がある場合 (初期収束) は、解探索をそれ以上に進めても、最適解を発見することは困難である。

2.2 Continuous Population-Based Incremental Learning

PBILc は設計変数毎に正規分布にもとづく確率モデル $N(\mu, \sigma^2)$ を構築し、解探索を行う。

各設計変数で平均 μ は式 (1) によって線形結合し、毎世代更新する。

$$\mu^{t+1} = (1 - \alpha)\mu^t + \alpha(X^{best1} + X^{best2} - X^{worst}) \quad (1)$$

式 (1) は集団の最良探索点と 2 番目に良い探索点の設計変数を足し、さらに一番悪い探索点の設計変数を引

いた値をもとに、次世代の平均 μ を計算することを表している。式 (1) の意図は、最良探索点と 2 番目に良い探索点を足すことで集団が 1 力所に留まることを防ぎ、かつ最悪探索点から離れる方向に平均 μ を移動させることで、良いと考えられる方向に確率モデルを移動させることである。 α ($0 < \alpha \leq 1$) は学習率であり、この値が小さいほど過去の確率モデルの影響を大きく受ける。

分散 σ^2 については、式 (2) を用いた更新方法が推奨されている。

$$\sigma^{t+1} = (1 - \alpha)\sigma^t + \alpha \sqrt{\frac{\sum_{j=1}^k (X^j - \bar{X})^2}{k}} \quad (2)$$

PBILc には集団数 P 、学習率 α 、切捨て率 K ($0 < K \leq 1$) という 3 種のパラメータがある。PBILc は切捨て率 K により、集団中の評価値の高い上位 $P \times K$ 個の探索点を選択する。この探索点数が式 (2) で用いている k である。 X は評価値順にソート済みであるとする。 \bar{X} は選択された探索点の設計変数の平均値である。この式の意図は、集団中の上位 k 個の探索点の分散 σ^2 をもとに次世代の分散 σ^2 を決定することである。

PBILc では学習率 α の設定が特に重要である。 α が大きすぎると収束が早いので十分な探索が行えず、局所解に留まることが多くなる。しかしながら、 α が小さすぎる場合、収束が遅いため、最適解に到達するまでに必要な計算時間は増大する。

2.3 Hierarchical Importance Sampling

HIS は多様性が異なる複数の集団と確率モデルを用いて解探索を行う EDA である。HIS は L 個の集団 (X_1, \dots, X_L) と L 個の確率モデル ($p_1(x), \dots, p_L(x)$) を解探索に用いる。各集団と各確率モデルは 1 対 1 対応している。各確率モデルが生成する探索点は、それぞれ多様性が異なるように目標分布 ($q_1(x), \dots, q_L(x)$) が各確率モデルで設定されている。

HIS がある確率モデル (p_i) 構築のために複製選択を行う際は、その確率モデルと対応関係にある集団 (X_i) とその上下にある集団 (X_{i+1}, X_{i-1}) をマージした集団から良好な探索点を選択する (Fig. 1)。この工夫により、HIS はある集団の良好な探索点をその周りに伝搬することができるため、すべての確率モデルを反復して良くすることができる。

3 Real-coded EDA using Multiple Probabilistic Models

3.1 設計指針

RMM は多様な複数の集団と確率モデルを用いて解探索を行う実数値 EDA である。複数の集団と確率モデル

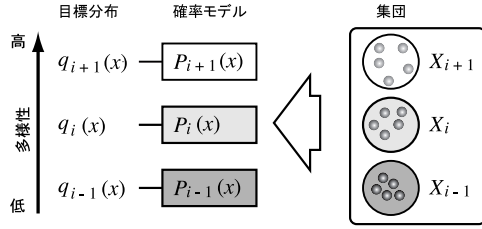


Fig.1 Hierarchical Importance Sampling

を用いる仕組みについては HIS を，集団から確率モデルを構築し，探索点を生成する方法については PBILc を参考にして RMM を設計した．

RMM の工夫点として，PBILc における集団の多様性は学習率 α に依存するため，確率モデルの更新に用いる学習率を各確率モデルで変化させることで，多様な集団を形成している点を挙げることができる．

3.2 パラメータとアルゴリズム

RMM のパラメータは，確率モデルと集団の組の数 L ，1 つの確率モデルが生成する探索点数 C ，切捨て率 K ，最大学習率 β_1 ，最小学習率 β_L の 5 つである．RMM の疑似コードを Fig. 2 に示す．

まず，RMM は HIS と同様に 1 対 1 対応した L 個の集団 (X_1, \dots, X_L) と L 個の確率モデル ($p_1(x), \dots, p_L(x)$) を用意し，それぞれ初期化を行う．初期集団として，一様分布で生成した探索点を代入する．初期確率モデルとして，平均 μ は各初期集団の最良個体の設計変数の値とし，分散 σ^2 は設計空間の絶対値の $1/4$ の 2 乗とする．

次に，各集団で多様性を変化させるため，多様性を最も低くしたい確率モデル p_1 には最大学習率 β_1 を，多様性を最も高くしたい確率モデル p_L には最小学習率 β_L を設定する．その他の確率モデル p_2, \dots, p_{L-1} には最大学習率 β_1 と最小学習率 β_L を等比的に変化させた値 β_i (式 (3)) をその学習率に設定した．

$$\beta_i = \beta_1 \cdot \left(\frac{\beta_L}{\beta_1} \right)^{\frac{i-1}{L-1}} \quad (3)$$

確率モデル p_i を更新するために，集団 X_{i-1}, X_i, X_{i+1} から評価値の高い順に $3 \times C \times K$ 個を選択し，式 (1)，式 (2) と学習率 β_i をもとに確率モデルを更新する．ただし， $i = 1$ の場合は X_1 と X_2 から $2 \times C \times K$ 個を選択し， $i = L$ の場合は X_{L-1} と X_L から $2 \times C \times K$ 個を選択する．次に，更新された確率モデルから C 個の新しい探索点を生成し，すべての探索点の評価値を計算する．そして，新しい集団と現在の集団との置換を行う．この動作をすべての確率モデルで行う．

4 実験

4.1 設定

RMM の性能を評価するための実験を行う．比較のために PBILc と実数値 EDA の 1 つである分散確率モデル遺伝的アルゴリズム (DPMBGA: Distributed Probabilistic Model-Building Genetic Algorithm) ⁶⁾，さら

1. $t = 1$
2. 対象関数の設計空間領域に一様分布で探索点を生成し，初期集団 X_1^t, \dots, X_L^t とする
3. 初期集団 X_1^t, \dots, X_L^t の評価値を計算
4. 確率モデル $p_1^t(x), \dots, p_L^t(x)$ の初期化． μ には各集団の最良個体の設計変数を代入する． σ^2 には設計空間の絶対値の $1/4$ の 2 乗を代入する．
5. 各確率モデルの学習率 $\beta_i (i = 1, \dots, L)$ を計算
6. 終了条件を満たすまで以下の手順を繰り返す
 - (1) $i = 1$
 - (2) 確率モデル p_i 毎に以下の手順を行う．
 - i. 集団 $X_{i-1}^t, X_i^t, X_{i+1}^t$ から良好な個体を選択
 - ii. 式 (1)，式 (2)， β_i に従い，確率モデル p_i^t を p_i^{t+1} に更新
 - iii. 確率モデル p_i^{t+1} から次世代の集団 X_i^{t+1} を生成
 - iv. 集団 X_i^{t+1} の評価値を計算
 - v. 集団 X_i^{t+1} と集団 X_i^t とを置換
 - vi. $i = i + 1$
 - (3) $t = t + 1$

Fig.2 RMM の疑似コード

Table1 実験結果 (－ は最適解を発見できなかったことを示す)

関数	RMM	PBILc	DPMBGA	REX ^{star}
Sphe.	3.4×10^3	2.8×10^3	1.3×10^5	6.9×10^3
Ellli.	1.0×10^4	3.3×10^4	1.7×10^5	8.5×10^3
k-tab.	8.4×10^3	3.0×10^4	1.6×10^5	1.1×10^4
Ackl.	6.9×10^3	7.7×10^3	2.7×10^5	1.4×10^4
Rast.	1.2×10^5	1.4×10^5	6.0×10^5	1.2×10^5
Schw.	1.1×10^5	1.4×10^6	8.0×10^5	(No Data)
Boha.	6.4×10^3	6.2×10^3	2.7×10^5	1.5×10^4
Grie.	5.8×10^3	8.6×10^3	3.0×10^5	(No Data)
Ridg.	5.4×10^4	9.1×10^4	3.5×10^5	(No Data)
R _{Chain}	1.3×10^5	－	9.5×10^5	4.7×10^4
R _{Star}	6.3×10^4	－	4.5×10^5	5.5×10^4
Scha.	5.4×10^4	9.4×10^4	－	7.7×10^4

に性能評価値の参考として，性能の高い実数値 GA である REX^{star7)} との比較も行う．

実験では数学的テスト関数として，様々な性質を持つ関数 (悪スケール性，多峰性，設計変数間の依存関係など) を用いて性能評価を行った．次元数は $n = 20$ である．評価計算回数が 2.0×10^6 回に達する前に，最良探索点の評価値が 1.0×10^{-7} 以下の値になったとき，最適解を発見したと判定する．

4.2 結果

実験結果を Table 1 に示す．なお，REX^{star} の結果は ⁷⁾ からの引用である．⁷⁾ で用いられていないテスト関数の結果については，(No Data) と記述した．また，REX^{star} の結果は 30 試行の平均評価計算回数であるが，RMM，PBILc，DPMBGA の結果は 20 試行の平均評価計算回数である．

まず，RMM と PBILc と比較する．Sphere 関数と Bohachevsky 関数以外では，RMM は PBILc よりも少ない平均評価計算回数で最適解を発見できることがわかる．ただし，Sphere 関数と Bohachevsky 関数との平均評価計算回数の差は約 20% と約 3% であり，その差は比較的少ないと言える．また，PBILc と RMM は同じ確率モデルの生成方法を用いているのにも関わらず，

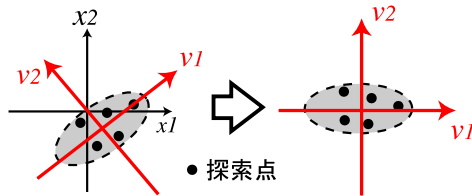


Fig.3 設計変数の無相関化

Table2 独立成分分析機能を組み込んだ RMM の結果

関数	RMM with PCA	RMM with no PCA	削減率
Ridg.	7.0×10^3	5.4×10^4	0.87
R_{Chain}	6.5×10^4	1.3×10^5	0.50
R_{Star}	2.5×10^4	6.3×10^4	0.60

PBILc は最適解を発見できない Rosenbrock-Chain 関数と Rosenbrock-Star 関数に対して、RMM では最適解を発見できる点は大きいと考える。

RMM と DPMBGA を比較した場合、すべての関数において、RMM は DPMBGA よりも少ない平均評価計算回数で最適解を発見できることがわかる。RMM と DPMBGA の評価計算回数の差は約 5～52 倍であり、その差は比較的大きいと言える。

RMM と REX^{star} を比較した場合、Rosenbrock-Chain 関数を除いて、RMM と REX^{star} の平均評価計算回数の差は約 0.5～1.2 倍であり、その差は比較的小さいと言える。ただし、Rosenbrock-Chain 関数において、RMM が最適解を発見するために必要な評価計算回数は REX^{star} の約 2.8 倍であり、その差は大きいと言える。

5 独立成分分析機能の追加

5.1 概要

RMM は PBILc と同様に、設計変数毎に確率モデルを構築し、探索点をサンプリングする。しかし、設計変数に依存関係がある問題の場合は、その相関を無くすように探索点群を変換してから確率モデルを構築する方がよい結果が得られると考えられる (Fig. 3)。

そこで、高橋らの独立成分分析を用いた実数値交叉の知見⁸⁾にもとづき、主成分分析 (Principal Component Analysis: PCA) を用いた独立成分分析機能を RMM に組み込んだ。これは、DPMBGA で用いられている機能と同様のものである。

5.2 実験

4.2 節で用いた関数の中で、依存関係のある単峰性関数に PCA を用いた RMM を適応させた。その結果と、評価計算回数の削減率を Table 2 に示す。ここで削減率とは、PCA を用いた RMM が最適解に到達する評価計算回数を PCA を用いない RMM の評価計算回数で割った値である。

Table 2 に示す通り、PCA を組み込んだ RMM を用いる事で約 50%～87% ほど評価計算回数が削減できたことを確認できた。

6 まとめと今後の課題

本稿では、既存の EDA である PBILc と HIS を参考にして、多様性が異なる複数の確率モデルを用いた新しい実数値 EDA である RMM の提案を行った。RMM は各確率モデルに異なる学習率を設定することで多様な集団を形成し、解探索を行う特徴を持つ。

様々な性質を持つ 12 種の連続テスト関数を用いて RMM の性能評価を行った結果、RMM は良好な結果を得ることができた。さらに、独立成分分析機能を RMM に追加することにより、依存関係のある単峰性関数において、大幅な評価計算回数の削減を行えることを示した。

しかしながら、多峰性関数においては、局所解が多数存在するため、独立成分分析機能が想定通りに動作しないことが予備実験からわかった。この点については、環境分散スキーム⁹⁾の仕組みを組み込む事で、ある程度の改善が見込めるのではと考えている。また、RMM の実問題への適用として、タンパク質の立体構造予測問題に取り組む。

参考文献

- 1) H. Mühlenbein and G. Paass. From recombination of genes to the estimation of distributions I. Binary parameters. *Parallel Problem Solving from Nature*, pp. 178–187, 1996.
- 2) P. Larranaga and J. A. eds. Lozano. *Estimation of Distribution Algorithm*. Kluwer Academic Publishers, 2002.
- 3) D. E. Goldberg. *Optimization and Machine Learning*. Addison-Wesley Publishing Company, 1989.
- 4) M. Sebag and A. Ducoulombier. Extending population-based incremental learning to continuous search spaces. *Proc. of the Parallel Problem Solving from Nature - PPSN V*, pp. 418–427, 1998.
- 5) Takayuki Higo and Keiki Takadama. Maintaining multiple populations with different diversities for evolutionary optimization based on probability models. *Information and Media Technologies*, Vol. 3, No. 2, pp. 362–374, 2008.
- 6) 廣安知之, 三木光範, 下坂久司, 佐野正樹, 筒井茂義. 分散確率モデル遺伝的アルゴリズム. 情報処理学会論文誌「数理モデル化と応用」, Vol. 45, No. SIG2, pp. 56–65, 2004.
- 7) 小林重信. 実数値 GA のフロンティア. 人工知能学会論文誌 24 巻 1 号 SP-N, pp. 147–162, 2009.
- 8) 高橋仁人, 喜多一. 実数値遺伝的アルゴリズムのための独立成分分析を用いた交叉の提案. 第 13 回 自律分散システム・シンポジウム 資料, pp. 245–250, 2001.
- 9) 三木光範, 廣安知之, 金子美華, 畠中一幸. 環境分散型並列遺伝的アルゴリズム. 電子情報通信学会, Vol. 19990528, pp. 87–93, 1999.