

# 不確実性を含む最適化問題に対する 差分進化の適用

末永 大樹<sup>1</sup> 田川 聖治<sup>2</sup>

概要：本稿では、不確実性を含む最適化問題である目的関数にノイズを含むノイズ問題を対象として進化計算の一種である差分進化（DE:Differential Evolution）を拡張した新たな DE を提案する。提案する DE では、予測区間による統計的検定とカットオフ点を用いて、目的関数値を 1 回サンプリングすることによって個体の最適性を評価し、有望でない個体に対しては、以降のサンプリングの繰り返しを省略する。

## 1. はじめに

進化計算（EA : Evolutionary Algorithm）は、不確実性を含む最適化問題に適応できることが報告されている [1][2][3][4][5]。このような不確実性を含む最適化問題に対して従来の進化計算を適用する際、各個体の目的関数値を複数回のサンプリングにより評価する。つまり、サンプリング回数によって得られる解の精度は左右される。しかし、サンプリングによる解の評価は計算コストが伴う。そのため、不確実性を含む最適化問題に対する解を効率的に探索する進化計算では、解の精度を損なわずサンプリング回数を減らす工夫が重要となる。

差分進化（DE : Differential Evolution）[6] は決定変数が実数値を取る関数最適化問題に対する進化計算の一種である。DE はアルゴリズムが単純であるため実装が容易である。また、代表的なテスト関数を対象とした数値実験によれば、典型的な遺伝的アルゴリズムや進化戦略と比較し、DE は解の収束に優れ、得られる解も頑健である。DE の遺伝子的演算子である戦略では、ターゲットベクトルと呼ばれる 1 つの親個体に対してトライアルベクトルと呼ばれる 1 つの子個体が生成される。次に、DE の生存選択では、ターゲットベクトルとトライアルベクトルを比較することで、親子間でのトーナメント選択を行う。

本稿では、不確実性を含む最適化問題の一種である、測定誤差など目的関数にノイズを含む問題を対象として、新たな DE のアルゴリズムを提案する。提案法では、DE の

生存選択に着目し、トライアルベクトルに対する目的関数値を予測区間とカットオフ点により評価し、トライアルベクトルが不良解と予想される場合は、そのサンプリングの繰り返しを省略する。また、目的関数値に付加されるノイズとしては、多くの現実的な問題でみられる正規分布に従うノイズを考える。

最後に、従来の DE のアルゴリズム、DE に予測区間を導入したアルゴリズム、DE にカットオフ点を導入したアルゴリズム、提案法である DE に予測区間とカットオフ点の両方を導入したアルゴリズムを比較し、サンプリング回数と同じ場合、多くのテスト問題において提案法により最良の解が得られることを確認する。

## 2. ノイズ問題の定式化

ノイズを含んだ目的関数  $F(\vec{x})$  は、平均  $\mu(\vec{x})$  で分散  $\sigma(\vec{x})^2$  の式 (1) のような正規分布に従う。

$$F(\vec{x}) \sim N(\mu(\vec{x}), \sigma(\vec{x})^2) \quad (1)$$

ある解  $\vec{x} \in \mathcal{R}^D$  に対して目的関数値を  $N$  回評価して得られた標本を  $\{F_1, \dots, F_n, \dots, F_N\}$  とすると、それらの標本平均は式 (2)、標本不変分散は式 (3) となる。

$$F(\vec{x}, N) = \frac{1}{N} \sum_{n=1}^N F_n \quad (2)$$

$$s(\vec{x}, N)^2 = \frac{1}{N-1} \sum_{n=1}^N (F_n - F(\vec{x}, N))^2 \quad (3)$$

次に、ある解  $\vec{x}$  に対して  $N$  個の目的関数値  $F_n$  が標本として得られているとき、その解  $\vec{x}$  に対する目的関数値の  $N+1$  番目の標本の値  $F_{n+1}$  は、有意水準を  $\alpha$  として式 (4) の範囲にあると予測される。

<sup>1</sup> 近畿大学総合理工学研究科  
Graduate School of Science and Engineering  
Research, Kinki University

<sup>2</sup> 近畿大学理工学部  
School of Science and Engineering, Kinki University

$$L(\vec{x}, N) \leq F_{N+1} \leq U(\vec{x}, N) \quad (4)$$

$$\begin{cases} L(\vec{x}, N) = F(\vec{x}, N) - \beta s(\vec{x}, N) \\ U(\vec{x}, N) = F(\vec{x}, N) + \beta s(\vec{x}, N) \end{cases} \quad (5)$$

$$\beta = t(N-1, \alpha/2) \sqrt{1 + \frac{1}{N}} \quad (6)$$

ここで、 $t(N-1, \alpha/2)$  は有意水準を  $\alpha$  とする t 分布の上側確率である。また、本稿では  $\alpha = 0.05$  とする。

本稿では、式 (4) の予測区間の上限値  $U(\vec{x}, N)$  を目的関数とする以下のノイズ問題を対象とする。

$$\begin{cases} \text{minimize} & U(\vec{x}, N) \\ \text{subject to} & x_j \leq \bar{x}_j, j = 1, \dots, D \end{cases} \quad (7)$$

上記のノイズ問題は、予測区間の上限値  $U(\vec{x}, N)$  を最小とする問題であり、決定変数は  $D$  個の実数値  $x_j \in \mathbb{R}$  であり、その上下限値  $\underline{x}_j$  と  $\bar{x}_j$  が与えられている。

### 3. DE の拡張法

ノイズ問題に DE を適用し解の精度を高めるには、式 (2) と式 (3) のサンプリング回数  $N$  を多くする必要がある。しかし、従来の DE では、各個体の評価において、式 (1) の関数値を  $N$  回計算する必要があり、サンプリング回数  $N$  を増やすと DE の実行時間が膨大となる。

#### 3.1 予測区間を用いた DE (DE-P)

DE-P の擬似コードを図 1 に示す。まず、従来の DE と同様、初期集団の個体を生成し、各個体の目的関数値  $U(\vec{x}_i, N)$  を計算する。次にトライアルベクトル  $\vec{u}$  を生成し、式 (2) で  $N = 1$  として、1 回のサンプルで作られた目的関数値である  $F(\vec{u}, 1)$  を計算後、 $F(\vec{u}, 1)$  とターゲットベクトル  $\vec{x}_i$  の目的関数値  $U(\vec{x}_i, N)$  を比較する。 $F(\vec{u}, 1)$  より  $U(\vec{x}_i, N)$  が劣れば  $U(\vec{u}, N)$  の計算を省略し、 $F(\vec{u}, 1)$  が勝れば  $U(\vec{u}, N)$  を計算して再び  $U(\vec{x}_i, N)$  と比較する。最後に、事前に定められた終了条件が満たされると、集団内で目的関数値  $U(\vec{x}_b, N)$  が最小の個体  $\vec{x}_b \in \mathbb{R}^D$  を出力して終了する。

#### 3.2 カットオフ点を用いた DE (DE-C)

DE-C の擬似コードを図 2 に示す。カットオフ点は任意の値  $\gamma$  ( $\gamma > 0$ ) であり、 $F(\vec{u}, 1)$  と  $\gamma$  を比較する。 $F(\vec{u}, 1)$  がカットオフ点  $\gamma$  より大きければ、現実的に解として不良であるような目的関数値が得られたとして、 $U(\vec{u}, N)$  の計算を省略する。事前に定められた終了条件が満たされると、集団内で目的関数値が最小の個体を出力して終了する。

#### 3.3 予測区間とカットオフ点を用いた DE (DE-PC)

提案する DE-PC は DE-P と DE-C を組み合わせたものである。DE-PC では DE-C と同様、全ての個体を 1 回のみサンプリングし、それがカットオフ点よりも小さい場合の

```

for (i = 1; i ≤ NP; i++) {
  Randomly generate  $\vec{x}_i \in \mathbf{P}$ ;
  Evaluate  $U(\vec{x}_i, N)$ ;
}
i = 1; /* index of the target vector */
while (termination condition == false) {
  Generate the trial vector  $\vec{u}$ ;
  Evaluate  $F(\vec{u}, 1)$ ;
  if ( $F(\vec{u}, 1) \leq U(\vec{x}_i, N)$ ) {
    Evaluate  $U(\vec{u}, N)$ ;
    if ( $U(\vec{u}, N) \leq U(\vec{x}_i, N)$ ) {
       $\vec{x}_i = \vec{u}$ ;
       $U(\vec{x}_i, N) = U(\vec{u}, N)$ ;
    }
  }
  i = (i mod NP) + 1;
}
Output  $\vec{x}_b \in \mathbf{P}$  with the minimum  $U(\vec{x}_b, N)$ ;

```

図 1 DE-P の擬似コード

```

for (i = 1; i ≤ NP; i++) {
  Randomly generate  $\vec{x}_i \in \mathbf{P}$ ;
  Evaluate  $F(\vec{x}_i, 1)$ ;
  if ( $F(\vec{x}_i, 1) \leq \gamma$ ) {
    Evaluate  $U(\vec{x}_i, N)$ ;
  } else {
     $U(\vec{x}_i, N) = \infty$ ;
  }
}
i = 1; /* index of the target vector */
while (termination condition == false) {
  Generate the trial vector  $\vec{u}$ ;
  Evaluate  $F(\vec{u}, 1)$ ;
  if ( $F(\vec{u}, 1) \leq \gamma$ ) {
    Evaluate  $U(\vec{u}, N)$ ;
    if ( $U(\vec{u}, N) \leq U(\vec{x}_i, N)$ ) {
       $\vec{x}_i = \vec{u}$ ;
       $F(\vec{x}_i, 1) = F(\vec{u}, 1)$ ;
       $U(\vec{x}_i, N) = U(\vec{u}, N)$ ;
    }
  } else { /*  $F(\vec{u}, 1) > \gamma$  */
    if ( $F(\vec{u}, 1) \leq F(\vec{x}_i, 1)$ ) {
       $\vec{x}_i = \vec{u}$ ;
       $F(\vec{x}_i, 1) = F(\vec{u}, 1)$ ;
    }
  }
  i = (i mod NP) + 1;
}
Output  $\vec{x}_b \in \mathbf{P}$  with the minimum  $U(\vec{x}_b, N)$ ;

```

図 2 DE-C の擬似コード

み  $N$  回サンプリングする。さらに、トライアルベクトル  $\vec{u}$  については、1 回のサンプリング値がカットオフ点とターゲットベクトル  $\vec{x}_i$  の目的関数値  $U(\vec{x}_i, N)$  よりも小さい場合のみ、 $\vec{u}$  を  $N$  回サンプリングして  $U(\vec{u}, N)$  を求める。

表 1 最良解の目的関数値  $U$

$f_p$	DE	DE-P	DE-C	DE-PC
$f_1$	371.718	3.709	3.198	<b>2.780</b>
$f_2$	54.291	3.281	2.994	<b>2.661</b>
$f_3$	75.996	19.289	18.971	<b>18.273</b>
$f_4$	3500.078	4.203	6.556	<b>3.221</b>
$f_5$	7.404	3.847	3.986	<b>3.815</b>

#### 4. 数値実験

DE, DE-P, DE-C と DE-PC のプログラムを Java 言語で実装した. また, ノイズ問題の目的関数  $U(\vec{x}, N)$  において  $F(\vec{x})$  を定義するため, 5 種類のテスト関数  $f_p$  を用意した. すべての DE において集団の個体数  $N_p = 100$ , 終了条件をサンプルの総数  $3 \times 10^5$ , カットオフ点  $\gamma = 50$  として, 5 種類のノイズ問題に対して, 30 回ずつ適用した.

各 DE で得られた目的関数値を表 1 に示す. 表 1 では各テスト関数ごとに最良値を太字で示している. 表 1 から提案法である DE-PC の有効性が確認できる.

図 3 に DE, DE-P, DE-C と DE-PC による式 (7) の最適化問題の目的関数値のサンプル数に対する変化を示す. 図 3 から予測区間とカットオフ点は DE の収束性を高めることがわかる. また, カットオフ点は探索の初期段階で極めて有効である.

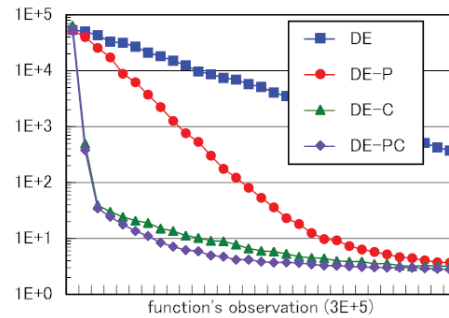
#### 5. おわりに

本稿では, ノイズ問題に適用する DE のサンプリング回数を短縮するため, 新たに DE を拡張した DE-PC を提案した. 数値実験の結果より, 提案した DE-PC は同じサンプル数で従来法より良い解が得られることを確認した.

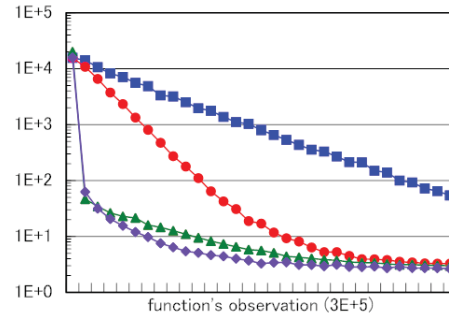
今後の課題は, カットオフ点  $\gamma$  の調整法の提案である.

#### 参考文献

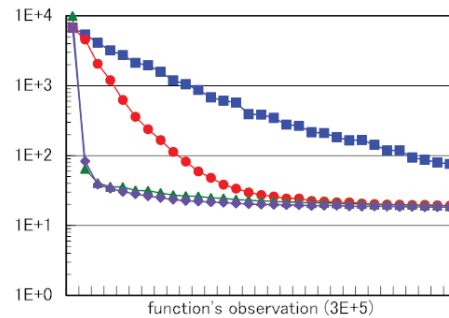
- [1] Y. Jin and J. Branke: Evolutionary optimization in uncertain environments – a survey, *IEEE Trans. on Evolutionary Computation*, Vol. 9, No. 3, 2005, pp. 303–317.
- [2] J. Branke and C. Schmidt: Sequential sampling in noisy environments: *Proc. of PPSN VIII*, LNCS 3242, Springer, 2004, pp. 202–211.
- [3] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama: Opposition-based differential evolution for optimization of noisy problems, *Proc. of IEEE Congress of Evolutionary Computation*, 2006, pp. 6756–6763.
- [4] B. Liu, X. Zhang, and H. Ma: Hybrid differential evolution for noisy optimization, *Proc. of IEEE Congress of Evolutionary Computation*, 2008, pp. 587–592.
- [5] 末永大樹, 田川聖治: ロバスト最適化問題に対する差分進化の適用, 情報処理学会関西支部 支部大会 講演論文集, 2013, B-101.
- [6] R. Storn and K. Price: Differential evolution - a simple and efficient heuristic for global optimization over continuous space, *Journal of Global Optimization*, Vol. 4, No. 11, 1997, pp. 341–359.



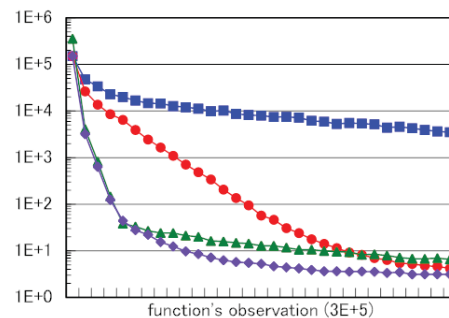
Sphere function:  $f_1$



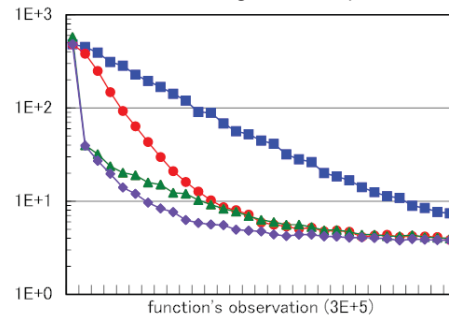
Hyper-Ellipsoid function:  $f_2$



Rosenbrock function:  $f_3$



Schwefel's Ridge function:  $f_4$



Griewank function:  $f_5$

図 3 目的関数値  $U$  の変化