

多項式回帰を用いた変数間依存関係の推定に基づく グループ交叉を導入した適応的差分進化 Adaptive Differential Evolution with Group-Wise Crossover Based on Dependency among Variables Estimated by Polynomial Regression

広島修道大学商学部

阪井 節子 (Setsuko Sakai)

Faculty of Commercial Sciences, Hiroshima Shudo University

広島市立大学大学院 情報科学研究科 高濱 徹行 (Tetsuyuki Takahama)

Graduate School of Information Sciences, Hiroshima City University

概要

最適化が困難な変数間依存性の強い問題に出現する特徴的な個体の分布として、直線に沿った細い楕円形分布や曲線に沿った狭い分布がある。このような場合に優れた子個体を生成するには、直線や曲線に沿って変数を同時に変更する必要がある。また、同様の分布は、探索点集合が最適解から離れており、最適解に近づくために特定の方向に移動する必要がある場合にも出現する。差分進化における2項交叉は、各変数(遺伝子)について同じ確率で交叉を行うかどうかを決定しているため、特定の遺伝子を同時に交叉することは困難である。我々は探索点の分布から相関係数を求め、相関の高い遺伝子をグループ化し、グループ単位で遺伝子を交叉するかどうかを決定するCGX(Correlation-based Group-wise Crossover)を提案した。しかし、相関係数は線形的な関係を示すものであり、より高次の多項式的な関係があるかどうかを判定することは困難であった。本研究では、探索点の分布に対して多項式回帰モデルを適用して各遺伝子間の決定係数を求め、決定係数が大きい遺伝子をグループ化し、グループ単位で遺伝子を交叉するPRGX(Polynomial Regression-based Group-wise Crossover)を提案する。PRGXは相関係数に基づくCGXを拡張し、直線的な分布だけではなく、曲線的な分布に対応可能な交叉である。本手法を差分進化の代表的手法であるJADEに導入し、幾つかのベンチマーク問題を最適化し、性能を比較することにより、本手法の効果を調べる。

1 はじめに

進化的計算などの集団的最適化手法では、良好な探索点の分布から様々な情報を得ることができる。最適化が困難な問題に変数間依存性の強い問題がある。このような問題を最適化する過程で、特徴的な分布として、図1のように傾いた細い楕円形状の分布や曲線に沿った狭い分布が出現することがある。

このような場合に優れた子個体を生成するには、直線や曲線に沿って変数を同時に最適解に近づくように変更する必要がある。また、同様の分布は、変数分離型の問題において探索点集合が最適解から離れている場合にも出現する。この場合にも、変数を同時に変更することにより効率的に最適解の方向へ移動できると考えられる。

進化的計算の一つとして、差分進化(Differential Evolution, DE)が提案されている[1, 2]。DEにおける代表的な交叉は、2項交叉(binomial crossover, BX)である。2項交叉では、ランダムに選択された遺伝子を必ず交叉する遺伝子とし、それ以外の遺伝子は交叉率 CR で交叉させる。各遺伝子について同じ確率で交叉を行うかどうかを決定しているため、特定の遺伝子を同時に交叉することは困難である。

我々は、相関係数を用いたグループ化に基づく交叉として、CGX(Correlation-based Group-wise Crossover)[3]を提案してきた。しかし、相関係数は線形的な関係を示すものであり、より高次の多項式的な関係があるかどうかを判定することは困難であった。本研究では、探索点の分布を解析するために、各変数(遺伝子)間に多項式回帰モデルを適用して決定係数を求め、決定係数の大きい遺伝子をグループ化し、グループ単位で交叉するかどうかを決めるPRGX(Polynomial Regression-based Group-wise Crossover)を提案する。これにより、CGXでは把握が困難であった線形的分布より高次の多項式的分布

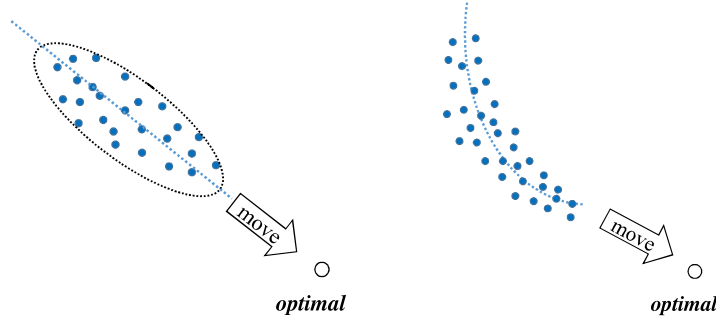


図 1: 探索点集団の最適解への移動

を把握しやすくなると考えられる。なお、本研究は Cai らの研究 [4] に基づいており、グループ化には Adaptive Grouping を拡張して決定係数に適応したものを、交叉には Group-wise Binomial Crossover を拡張したものを採用した。

DE の代表的手法の一つにアルゴリズムパラメータを適応的に調整する JADE[5] が提案されている。本研究では、JADE に PRGX を導入する。幾つかのベンチマーク問題を最適化し、CGX と性能を比較することにより、本手法の効果を調べる。

本論文の構成は次の通りである。2. で DE および JADE について簡潔に説明する。3. で関連研究を紹介する。4. で、本手法のアルゴリズムを説明する。5. で実験結果を述べる。6. はまとめである。

2 差分進化と JADE

2.1 差分進化

差分進化 (DE) は Storn and Price[1, 2] によって提案された進化的アルゴリズムである。DE は確率的な直接探索法であり、解集団を用いた多点探索を行う。DE には幾つかの形式が提案されており、DE/best/1/bin や DE/rand/1/exp などがよく知られている。これらは、DE/base/num/cross という記法で表現される。“base” は基本ベクトルとなる親の選択方法を指定する。例えば、DE/rand/num/cross は基本ベクトルのための親を集団からランダムに選択し、DE/best/num/cross は集団の最良個体を選択する。“num” は基本ベクトルを変異させるための差分ベクトルの個数を指定する。“cross” は子を生成するために使用する交叉方法を指定する。例えば、DE/base/num/bin は 2 項交叉 (BX) を用い、DE/base/num/exp は、指数関数的に減少する確率で遺伝子を交換する指数交叉 (exponential crossover) を用いる。

2.2 アルゴリズム

DE/rand/1/bin のアルゴリズムは以下のように記述できる [6]。

Step0 初期化。 N 個の初期個体 $\mathbf{x}_i$ を探索空間内に生成し、初期集団 $\{\mathbf{x}_i \mid i = 1, 2, \dots, N\}$ を構成する。

Step1 終了判定。終了条件を満足すれば、アルゴリズムは終了する。終了条件としては、最大の繰り返し回数や関数評価回数を用いることが多い。

Step2 突然変異。各個体 (target vector) $\mathbf{x}_i$ に対して、3 個体 $\mathbf{x}_{r1}, \mathbf{x}_{r2}, \mathbf{x}_{r3}$ を $\mathbf{x}_i$ および互いに重複しないようにランダムに選択する。基本ベクトル $\mathbf{x}_{r1}$ および差分ベクトル $\mathbf{x}_{r2} - \mathbf{x}_{r3}$ から変異ベクトル (mutant vector) $\mathbf{m}$ を以下のように生成する。

$$\mathbf{m} = \mathbf{x}_{r1} + F(\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (1)$$

ここで、 F はスケーリングパラメータである。

Step3 交叉。変異ベクトル $\mathbf{m}$ と親 $\mathbf{x}_i$ を交叉し、子ベクトル (trial vector) $\mathbf{x}_i^{\text{child}}$ を生成する。必ず交叉する点 j_{rand} を全ての次元 $[1, D]$ からランダムに選択する。子ベクトル $\mathbf{x}_i^{\text{child}}$ の j_{rand} 番目の要素は $\mathbf{m}$ の j_{rand} 番目の要素から継承し、それ以外の次元は、交叉パラメータ CR の確率で、 $\mathbf{m}$ の要素から継承する。残りの部分は、親 $\mathbf{x}_i$ から継承する。

実際の処理では、Step2 と Step3 は一まとまりの処理で実現される。

Step4 生存者選択。子ベクトルを評価する。子 $\mathbf{x}_i^{\text{child}}$ が親よりも良ければ子が生存者となり、親を子で置換する。

Step5 Step1 に戻る。

2.3 JADE

JADE では、スケーリングファクターの平均値 μ_F と交叉率の平均値 μ_{CR} によって良好なパラメータ値の確率分布を表現し、成功した時のパラメータ値を用いて適応的に平均値を更新する。初期値は、 $\mu_F = \mu_{CR} = 0.5$ である。各個体 $\mathbf{x}_i$ のために、異なるスケーリングファクター F_i と交叉率 CR_i が次式にしたがって独立に生成される。

$$F_i \sim C(\mu_F, \sigma_F) \quad (2)$$

$$CR_i \sim N(\mu_{CR}, \sigma_{CR}^2) \quad (3)$$

ここで、 $C(\mu_F, \sigma_F)$ は位置パラメータ μ_F 、尺度パラメータ $\sigma_F = 0.1$ の Cauchy 分布に基づく確率分布である。 $N(\mu_{CR}, \sigma_{CR}^2)$ は平均 μ_{CR} 、標準偏差 $\sigma_{CR} = 0.1$ の正規分布に基づく確率分布である。 CR_i は区間 $[0, 1]$ となるように切り捨てられる。 F_i は負の値あるいは 0 の場合は再生成され、それ以外の場合は 1 以下となるように切り捨てられる。 μ_F と μ_{CR} は、安定した学習を実現するために、指数移動平均を用いて更新される。

$$\mu_F = (1 - c)\mu_F + cS_{F2}/S_F \quad (4)$$

$$\mu_{CR} = (1 - c)\mu_{CR} + cS_{CR}/S_N \quad (5)$$

ここで、 S_N は親より良い子が生成された（成功した）回数、 S_F, S_{F2}, S_{CR} はそれぞれ成功時の F_i, F_i^2, CR_i の和である。すなわち、 μ_{CR} は成功時の単純な算術平均により更新される。これに対して、 μ_F は多様性を保持するために、大きな値を重視した重み付き平均によって更新される。定数 c は値を更新する際に使用される区間 $(0, 1]$ の重みであり、推奨値は 0.1 である。

JADE では“current-to-pbest”と呼ばれる突然変異戦略が提案され、親と上位個体の内分点が基本ベクトルとなる。変異ベクトルは次式で生成される。

$$\mathbf{m}_i = \mathbf{x}_i + F_i(\mathbf{x}_{pbest} - \mathbf{x}_i) + F_i(\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (6)$$

ここで、 $\mathbf{x}_{pbest}$ は上位 100 p % 個体からランダムに選択された個体であり、 p の推奨値は 0.05 である。

JADE ではアーカイブを使用する方法も提案されているが、本研究ではアーカイブを使用しないため、説明は省略する。

3 関連研究

3.1 リンケージ同定

変数間の依存関係を同定することは、探索プロセスにおいて非常に重要な問題であり、リンケージ同定と呼ばれている。リンケージ同定のため研究として、遺伝的アルゴリズムを対象に LINC (Linkage Identification by Nonlinearity Check) [7]、粒子群最適化を対象にリンケージの強さを示すリンケージ行列の学習 [8] が、差分進化を対象にリンケージ行列の学習 [4] が提案されている。これらの研究では同様の考え方が採用されているため、LINC について説明する。 i 番目の変数と j 番目の変数間のリンケージの強さを得るために、 i 番目の変数のみを摂動したときの関数値の変化 Δf_i 、 j 番目の変数のみを摂動したときの変化 Δf_j 、両方の変数を摂動したときの変化 $\Delta f_{ij} (i < j)$ を以下のように求める。

$$\Delta f_i = f(\dots, x'_i, \dots, x_j, \dots) - f(\dots, x_i, \dots, x_j, \dots) \quad (7)$$

$$\Delta f_j = f(\dots, x_i, \dots, x'_j, \dots) - f(\dots, x_i, \dots, x_j, \dots) \quad (8)$$

$$\Delta f_{ij} = f(\dots, x'_i, \dots, x'_j, \dots) - f(\dots, x_i, \dots, x_j, \dots) \quad (9)$$

もし i 番目の変数と j 番目の変数が独立ならば、以下の式が成立する。

$$\Delta f_{ij} = \Delta f_i + \Delta f_j \quad (10)$$

逆に、この条件が成立しなければ、 i 番目の変数と j 番目の変数間にリンケージがあると考えられる。リンケージの強さ e_{ij} は以下のように定義できる。

$$e_{ij} = |\Delta f_{ij} - (\Delta f_i + \Delta f_j)| \quad (11)$$

LINC では $O(D^2)$ 回の関数評価が必要となる。ただし、 D は変数の次元数である。したがって、目的関数の計算コストが高い場合、何度もリンケージ同定を行うことは非常に困難である。しかし、巨視的な関数景観と微視的な関数景観が大きく異なる問題も多い。このような問題ではリンケージ同定を繰り返す必要があるため、関数評価回数の観点からもリンケージ同定に LINC を用いることは適当ではないと考えられる。

3.2 グループ化

Cai ら [4] は、リンケージの強さを示すリンケージ行列 $LM[i, j]$ を用いてグループ化を行う Adaptive Grouping を提案している。擬似コードを図 2 に示す。 LM の成分の平均値を ϵ とし、リンケージの強さがそれ以上である遺伝子の対を同じグループに所属させるという方法を採用している。 Z_i は次元 i がグループに未所属 (0) か所属済み (1) かを示しており、いずれかのグループに一度所属すると他のグループには所属できなくなる。したがって、強さを調べる順序によって結果が変化する可能性がある。このため、調べる順序をランダムに入れ替えた π を用いて調べる順序を決めている。また、Cai らが提案したグループ毎の交叉を図 3 に示す。 $u(0, 1)$ は区間 $[0, 1]$ の一様乱数を生成する関数である。グループ毎に交叉率 CR で交叉するかどうかを決めている。 $\mathbf{x}$ と $\mathbf{m}$ は交叉する親個体と変異ベクトル、 $\mathbf{x}^{child}$ は生成される子個体である。

```

 $\epsilon$ =the mean value of  $LM[i, j]$ ;
Initialize the state vector  $Z=\{0, 0, \dots, 0\}$ ;
Initizlize the order vector  $\pi=\{1, 2, \dots, D\}$ ;
Shuffle  $\pi$  randomly;
 $numG=1$ ;
for( $i=1$ ;  $i \leq D$ ;  $i++$ )
    if( $Z_{\pi_i}==0$ ) {
         $Z_{\pi_i}=1$ ;
         $Group_{numG}=\{\pi_i\}$ ;
        for( $j=1$ ;  $j \leq D$ ;  $j++$ )
            if( $Z_{\pi_j}==0$  &&  $LM[\pi_i, \pi_j] \geq \epsilon$ ) {
                 $Z_{\pi_j}=1$ ;
                 $Group_{numG}=Group_{numG} \cup \{\pi_j\}$ ;
            }
         $numG=numG + 1$ ;
    }
}

```

図 2: Adaptive Grouping の擬似コード

```

for( $i=1$ ;  $i \leq numG$ ;  $i++$ )
    if( $u(0, 1) < CR$ )
        for( $j \in Group_i$ )  $x_j^{child}=\mathbf{m}_j$ ;
    else
        for( $j \in Group_i$ )  $x_j^{child}=\mathbf{x}_j$ ;

```

図 3: Group-wise Binomial Crossover の擬似コード

3.3 相関係数を用いたグループ化

相関係数を用いた遺伝子のグループ化に基づく交叉として提案したグループ交叉 CGX について説明する。グループ化のための判断基準を以下のように決定する。

1. 各世代において次元 i と次元 j の相関係数の絶対値 ρ_{ij} を求める。
2. ρ_{ij} の平均値 ($\bar{\rho}$) と標準偏差 (σ_ρ) を求め、パラメータ Sr に対して ρ^{strong} を以下のように定義する。

$$\rho^{strong} = \max\{\bar{\rho} + Sr\sigma_\rho, 0.15\} \quad (12)$$

Sr の推奨値は 1 である。

3. $\rho_{ij} > \rho^{strong}$ ならば、次元 i と次元 j 間に強い依存関係があると判定する。

CGX のアルゴリズムは、遺伝子のグループ化については、Adaptive Grouping と同様のアルゴリズムを採用する。ただし、 ρ_{ij} が ρ^{strong} より大きければ、同じグループとする。また、実装を簡単にするために、集合を用いずに、 G_i にグループ番号を記憶させた。図 4 に擬似コードを示す。

交叉についても、Group-wise Binomial Crossover と同様のアルゴリズムを採用する。ただし、必ず交叉させるグループをランダムに決めた。図 5 に CGX の擬似コードを示す。

```

Initialize the group numbers  $G=\{-1,-1,\dots,-1\}$ ;
Initizlize the order vector  $\pi=\{1,2,\dots,D\}$ ;
Shuffle  $\pi$  randomly;
numG=1;
for( $i=1$ ;  $i \leq D$ ;  $i++$ )
    if( $G_{\pi_i} < 0$ ) {
         $G_{\pi_i} = \text{numG}$ ;
        for( $j=1$ ;  $j \leq D$ ;  $j++$ )
            if( $G_{\pi_j} < 0 \ \&\& \ \rho_{\pi_i, \pi_j} > \rho^{strong}$ )  $G_{\pi_j} = \text{numG}$ ;
        numG=numG + 1;
    }

```

図 4: CGX における Adaptive Grouping の擬似コード

```

 $i_{rand} = \text{randint}(1, \text{numG})$ ;
for( $i=1$ ;  $i \leq \text{numG}$ ;  $i++$ ) {
    if( $i == i_{rand} \ || \ u(0,1) < CR$ ) cross[i]=1;
    else cross[i]=0;
}
for( $j=1$ ;  $j \leq D$ ;  $j++$ ) {
    if(cross[ $G_j$ ]==1)  $x_j^{\text{child}} = m_j$ ;
    else  $x_j^{\text{child}} = x_j$ ;
}

```

図 5: CGX の擬似コード

4 提案手法

本研究では、探索点の分布から次元間に多項式回帰を適用し、決定係数の値が大きい次元間に依存関係があると推定することにより、変数のグループ化を行う。これによって、直線的な関係だけでなく、多項式的な曲線的な関係も把握できると考えられる。

4.1 多項式回帰と決定係数

回帰分析では、説明の対象となる目的変数 (従属変数) と説明のために使用する説明変数 (独立変数) の関係を表現する式を推定する。例えば、多項式回帰モデルでは、目的変数 y と説明変数 x の間に多項式関係を想定し、推定誤差を最小化するパラメータ $\mathbf{c} = (c_0, c_1, \dots, c_p)^T$ を求める。

$$y = c_0 + c_1x + c_2x^2 + \dots + c_px^p \quad (13)$$

ここで、 p は多項式の最大次数である。変数 x に関するデータ $\{x_1, x_2, \dots, x_N\}$ と対応する変数 y に関するデータ $\{y_1, y_2, \dots, y_N\}$ が得られたと仮定すると、以下が成立する。

$$y_i = c_0 + c_1x_i + c_2x_i^2 + \dots + c_px_i^p + \epsilon_i \quad (i = 1, 2, \dots, N) \quad (14)$$

ここで、 N はデータ数、 ϵ_i は誤差である。行列ベクトル表現で記述すると、以下のようになる。

$$\mathbf{y} = \mathbf{X}\mathbf{c} + \boldsymbol{\epsilon} \quad (15)$$

$$\mathbf{y} = (y_1, y_2, \dots, y_N)^T, \mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^p \\ 1 & x_2 & x_2^2 & \dots & x_2^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^p \end{bmatrix}, \boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \dots, \epsilon_N)^T \quad (16)$$

相異なる $p+1$ 個以上のデータが得られている、すなわち、 $N \geq p+1$ を仮定すると、最小二乗法により以下のように $\mathbf{c}$ の推定値が得られる。

$$\hat{\mathbf{c}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (17)$$

決定係数 R^2 (coefficient of determination) は、データから求めた回帰モデルのあてはまりの良さ (誤差の少なさ) を示す尺度であるが、様々な定義が存在する。本研究では、以下の決定係数を用いる。

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (18)$$

ここで、 $\hat{y}_i$ は回帰モデルによる y_i の推定値、 $\bar{y}$ は $\{y_i\}$ の平均値である。分数の部分は平均値（最も単純なモデル）とデータの 2 乗和誤差に対する回帰モデルとデータの 2 乗和誤差の割合である。 R^2 の値は、0 から 1 の範囲である。 R^2 が 1 に近い場合は、回帰モデルによって目的変数の変化をほとんど説明できており、データに非常によく適合していることを示す。 R^2 が 0 に近い場合は、回帰モデルが目的変数の変化をほとんど説明できていないことを示す。本研究では、決定係数を用いて変数間依存関係を検出する。

4.2 決定係数による変数間依存性の検出

D 次元実数空間における個体集合を $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, $\mathbf{x}_k = (x_{k1}, x_{k2}, \dots, x_{kD})$, $k = 1, 2, \dots, N$ とする。 i 次元目の変数を目的変数、 j 次元目の変数を説明変数とし、 $\{(x_{ki}, x_{kj}) | k = 1, 2, \dots, N\}$ のデータを用いた多項式回帰分析の結果として得られる決定係数を r_{ij}^2 とする。このとき、 j 次元目の変数を目的変数、 i 次元目の変数を説明変数とする決定係数 r_{ji}^2 は、一般に r_{ij}^2 とは異なる。ただし、線形回帰の場合は逆関数も線形となるため、 $r_{ji}^2 = r_{ij}^2$ となる。本研究では、次元 i と次元 j 間の依存関係の強さを r_{ij}^2 と r_{ji}^2 の大きい方の値で推定する。

$$\rho_{ij}^2 = \max\{r_{ij}^2, r_{ji}^2\} \quad (19)$$

$$\rho^{strong} = \max\{\rho^2 + Sr\sigma_{\rho^2}, 0.2^2\} \quad (20)$$

$\rho_{ij}^2 > \rho^{strong}$ ならば、遺伝子 i, j 間に強い関係があると判定する。

4.3 適応的な適用確率の制御

多様な問題に適用できるように、CBX と 2 項交叉 (BX) を確率的に併用する方法が提案されている [9]。適用確率 R_{CBX} を導入し、子個体が親個体より良くなった割合を示す成功率に基づき適応的に制御する。CBX の成功率と BX の成功率を比較し、以下のように、CBX の成功率が高ければ R_{CBX} を増加させ、BX の成功率が高ければ R_{CBX} を減少させる。

$$R_{CBX} = \begin{cases} R_{CBX} + \Delta R & \text{CBX の成功率} > \text{BX の成功率} \\ R_{CBX} - \Delta R & \text{CBX の成功率} < \text{BX の成功率} \\ R_{CBX} & \text{otherwise} \end{cases} \quad (21)$$

ただし、 R_{CBX} は $[0.05, 0.95]$ の範囲になるように、範囲外の値になれば、0.05 あるいは 0.95 に設定し直す。 ΔR の推奨値は 0.01 である。

本研究でも同じ方法を採用する。

4.4 アルゴリズム

BX と PRGX では適切な F, CR の平均値が異なると考えられるため、各交叉に対して異なる平均値 μ_F^K, μ_{CR}^K を用いる。BX では $K=0$ であり、PRGX では $K=1$ である。

提案手法のアルゴリズムは以下の通りである。

Step0 パラメータの初期化

スケーリングファクターの平均値 $\mu_F^K = 0.5$ 、交叉率の平均値 $\mu_{CR}^K = 0.5$ とする。ただし、 $K=0, 1$ である。パラメータ生成時の標準偏差を $\sigma_F=0.1$ 、 $\sigma_{CR}=0.1$ とする。PRGX の適用確率 $R_{PRGX} = 0.5$ とする。

Step1 個体の初期化

初期集団 $P = \{\mathbf{x}_i | i = 1, 2, \dots, N\}$ を探索空間内でランダムに生成する。 N は集団サイズである。

Step2 終了条件

関数評価回数が最大評価回数 $FE_{\max}$ を超えれば、アルゴリズムは終了する。

Step3 各世代における初期化

成功時のパラメータ値を保持するリスト S^K を空にする。集団 P から決定係数 r_{ij}^2 、関係の強さ ρ_{ij} 、その平均値 $\bar{\rho}^2$ と標準偏差 σ_{ρ^2} 、 ρ^{strong} を求める。ただし、 $i, j \in \{1, 2, \dots, D\}$ である。

Step4 DE 操作

R_{PRGX} の確率で PRGX か BX かを選択し、PRGX では $K=1$ 、BX では $K=0$ とする。交叉率 CR_i を正規分布 $N(\mu_{CR}^K, \sigma_{CR}^2)$ で生成する。スケーリングファクター F_i を Cauchy 分布 $C(\mu_F^K, \sigma_F)$ に基づき生成する。パラメータを F_i, CR_i として DE/current-to-pbest/1/bin (BX) あるいは PRGX を実行し、子 $\mathbf{x}_i^{\text{child}}$ を生成する。子が親より良ければ、成功と判断し、子を生存者として選択し、成功時のパラメータ値 (F_i, CR_i) をリスト S^K に追加する。成功でなければ、親 $\mathbf{x}_i$ を生存者とする。

Step5 パラメータの更新

μ_F^K と μ_{CR}^K を S^K に基づいて更新する。ただし、 $K=0, 1$ である。 R_{PRGX} を成功率の比較に基づいて更新する。

Step6 Step2 へ戻る。

提案手法の擬似コードを図 6 に示す。‘+’で始まる行は、JADE に対する変更点を示している。

5 実験

5.1 テスト問題と実験条件

表 1 に、テスト問題の関数定義とその初期化領域を示す [5]。 f_1 は sphere 関数、 $f_2 \sim f_4$ は Schwefel 2.22, Schwefel 1.2, Schwefel 2.22 関数、 f_5 は Rosenbrock 関数、 f_6 は Step 関数、 f_7 は雑音を含む 4 次関数、 f_8 は Schwefel 2.26 関数、 f_9 は Rastrigin 関数、 f_{10} は Ackley 関数、 f_{11} は Griewank 関数、 $f_{12} \sim f_{13}$ はペナルティ付き関数である。なお、 D は次元数を表している。

表 1: 13 個のテスト問題 [10]

テスト関数	初期化領域
$f_1(\mathbf{x}) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$
$f_2(\mathbf{x}) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10, 10]^D$
$f_3(\mathbf{x}) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^D$
$f_4(\mathbf{x}) = \max_i \{ x_i \}$	$[-100, 100]^D$
$f_5(\mathbf{x}) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^D$
$f_6(\mathbf{x}) = \sum_{i=1}^D [x_i + 0.5]^2$	$[-100, 100]^D$
$f_7(\mathbf{x}) = \sum_{i=1}^D ix_i^4 + rand[0, 1)$	$[-1.28, 1.28]^D$
$f_8(\mathbf{x}) = \sum_{i=1}^D -x_i \sin \sqrt{ x_i }$ + $D \cdot 418.98288727243369$	$[-500, 500]^D$
$f_9(\mathbf{x}) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$
$f_{10}(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right)$ - $\exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$	$[-32, 32]^D$
$f_{11}(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]^D$
$f_{12}(\mathbf{x}) = \frac{\pi}{D} [10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2$ $\{1 + 10 \sin^2(\pi y_{i+1})\} + (y_D - 1)^2]$ + $\sum_{i=1}^D u(x_i, 10, 100, 4)$ where $y_i = 1 + \frac{1}{4}(x_i + 1)$ and $u(x_i, a, k, m) =$ $\begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-50, 50]^D$
$f_{13}(\mathbf{x}) = 0.1[\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2$ $\{1 + \sin^2(3\pi x_{i+1})\} + (x_D - 1)^2$ $\{1 + \sin^2(2\pi x_D)\}] + \sum_{i=1}^D u(x_i, 5, 100, 4)$	$[-50, 50]^D$

次元数 $D = 30$ に設定し、個体数 $N = 100$ 、最大関数評価回数は文献 [5] に基づいて決定した。ただし、 f_5 と f_{11} については収束後に探索し続けている状態であったため、評価回数をそれぞれ $\frac{1}{2}, \frac{4}{5}$ にしている。 PRGX では多項式近似の次数 $p \in \{1, 2, 3\}$ を比較した。 PRGX を CGX と比較するために、全ての交叉において適応的な適用確率制御を導入し、パラメータ $Sr = 1$, $\Delta R = 0.01$ とした。各関数について 50 回の試行を行い、結果を比較した。

5.2 実験結果

実験結果を表 2 に示す。各関数に対して、各試行における最良値の平均値と標準偏差を調べた。少数回の失敗試行が発生することがあり、平均値が不安定になったため、中央値も調べ最良の中央値を太字で示した。さらに、Wilcoxon signed rank test を行い、JADE に対して有意に優れていた場合に +、有意に劣っていた場合に −、有意差がない場合に = を付与した。なお、有意水準 5% の場合は +, −、有意水準 1% の場合は ++, −− で表現している。

提案手法では、PRGX($p = 2$) が最良の結果となっており、 $f_1, f_2, f_3, f_4, f_5, f_6, f_8, f_{10}, f_{11}, f_{12}, f_{13}$ の 11 関数で JADE よりも有意に優れている。 PRGX($p = 3$) および CGX では、10 関数で JADE よりも優位に優れており、PRGX($p = 1$) では 9 関数で優位に優れている。

```

JADE/current-to-pbest/1/BX+PRGX()
{
+  $\mu_F^K = \mu_{CR}^K = 0.5$ ,  $K=0,1$ ;
+  $\sigma_F = \sigma_{CR} = 0.1$ ;
+  $R_{PRGX} = 0.5$ ;
// Initialize a population
 $P = N$  individuals generated randomly in search space;
 $FE = FE + N$ ;
for( $t=1$ ;  $FE < FE_{max}$ ;  $t++$ ) {
+  $S^K = \emptyset$ ,  $K=0,1$ ;
+  $m^K = 0$ ,  $K=0,1$ ; // number of trials for BX and PRBX
+  $(r_{ij}^2)$ =coefficients of determination of  $P$  is obtained;
+  $(\rho_{ij}^2) = \max\{r_{ij}^2, r_{ji}^2\}$ ;
+  $\rho^{strong} = \max\{\rho^2 + S_r \sigma_{\rho^2}, 0.15^2\}$ ;
+ for( $i=1$ ;  $i \leq N$ ;  $i++$ ) {
+   if( $u(0,1) < R_{PRGX}$ )  $K=1$ ; // PRGX
+   else  $K=0$ ; // BX
+    $CR_i = \mu_{CR}^K + N(0, \sigma_{CR}^2)$ ;
+   if( $CR_i < 0$ )  $CR_i = 0$ ;
+   else if( $CR_i > 1$ )  $CR_i = 1$ ;
+   do {
+      $F_i = \mu_F^K + C(0, \sigma_F)$ ;
+   } while( $F_i \leq 0$ );
+   if( $F_i > 1$ )  $F_i = 1$ ;
+    $x_{pbest}$  = Randomly selected from top 100p% in  $P$ ;
+    $x_{r1}$  = Randomly selected from  $P(r1 \notin \{i\})$ ;
+    $x_{r2}$  = Randomly selected from  $P(r2 \notin \{i, r1\})$ ;
+    $m_i = x_i + F_i(x_{pbest} - x_i) + F_i(x_{r1} - x_{r2})$ ;
+   if( $K==1$ )
+      $x_i^{child}$ =PRGX between  $x_i$  and  $m_i$ ;
+   else
+      $x_i^{child}$ =binomial crossover between  $x_i$  and  $m_i$ ;
+    $FE = FE + 1$ ;
// Survivor selection
+   if( $f(x_i^{child}) < f(x_i)$ ) {
+      $z_i = x_i^{child}$ ;
+      $S^K = S^K \cup \{(F_i, CR_i)\}$ ; // a success case is added
+      $m^K++$ ;
+   }
+   else  $z_i = x_i$ ;
+ }
+  $P = \{z_i\}$ ;
+ for( $K=0$ ;  $K < 2$ ;  $K++$ )
+   if( $|S^K| > 0$ ) {
+      $\mu_F^K = (1 - c)\mu_F^K + c \sum_{F_i \in S^K} F_i^2 / \sum_{F_i \in S^K} F_i$ ;
+      $\mu_{CR}^K = (1 - c)\mu_{CR}^K + c \sum_{CR_i \in S^K} CR_i / |S^K|$ ;
+   }
+   if( $m^0 > 0 \ \&\& \ m^1 > 0$ ) {
+      $s^0 = |S^0|/m^0$ ; // success rate of BX
+      $s^1 = |S^1|/m^1$ ; // success rate of CBX
+     if( $s^1 > s^0$ )
+        $R_{PRGX} = R_{PRGX} + \Delta R$ ;
+     else if( $s^0 > s^1$ )
+        $R_{PRGX} = R_{PRGX} - \Delta R$ ;
+     if( $R_{PRGX} > 0.95$ )  $R_{PRGX} = 0.95$ ;
+     else if( $R_{PRGX} < 0.05$ )  $R_{PRGX} = 0.05$ ;
+   }
+ }
}

```

図 6: 提案手法の擬似コード

表 2: 適用確率に適応制御を導入した実験結果

	$FE_{\max}$	JADE	CGX	PRGX($p = 1$)	PRGX($p = 2$)	PRGX($p = 3$)
f_1	150,000	4.85e-61 $\pm$ 2.57e-60 7.45e-66	9.25e-63 $\pm$ 4.39e-62 5.56e-67 (++)	1.12e-62 $\pm$ 7.39e-62 6.42e-68 (+)	7.94e-62 $\pm$ 5.49e-61 5.58e-67 (++)	2.23e-64 $\pm$ 7.08e-64 3.12e-68 (++)
f_2	200,000	4.18e-27 $\pm$ 2.34e-26 7.10e-39	1.56e-28 $\pm$ 1.09e-27 2.50e-42 (++)	3.62e-30 $\pm$ 2.54e-29 3.81e-41 (++)	1.01e-33 $\pm$ 6.42e-33 2.50e-43 (++)	2.26e-36 $\pm$ 1.17e-35 3.33e-43 (++)
f_3	500,000	7.49e-62 $\pm$ 2.63e-61 1.23e-63	1.08e-61 $\pm$ 3.64e-61 8.18e-64 (=)	4.58e-62 $\pm$ 1.80e-61 2.73e-65 (+)	4.58e-62 $\pm$ 2.81e-61 2.63e-65 (++)	4.45e-60 $\pm$ 2.83e-59 1.26e-64 (++)
f_4	500,000	4.70e-24 $\pm$ 1.54e-23 1.70e-25	3.72e-24 $\pm$ 1.33e-23 2.82e-26 (+)	1.47e-24 $\pm$ 4.44e-24 5.32e-26 (=)	7.27e-25 $\pm$ 2.23e-24 3.48e-26 (++)	5.83e-25 $\pm$ 3.22e-24 2.78e-26 (++)
f_5	150,000	1.59e-01 $\pm$ 7.81e-01 2.92e-09	3.54e-13 $\pm$ 2.28e-12 9.66e-18 (++)	7.97e-02 $\pm$ 5.58e-01 5.36e-18 (++)	2.39e-01 $\pm$ 9.47e-01 2.05e-18 (++)	3.19e-01 $\pm$ 1.08e+00 4.24e-18 (++)
f_6	10,000	3.36e+00 $\pm$ 1.29e+00 4.00e+00	2.22e+00 $\pm$ 1.06e+00 2.00e+00 (++)	2.72e+00 $\pm$ 1.30e+00 2.50e+00 (++)	2.46e+00 $\pm$ 1.40e+00 2.00e+00 (++)	2.04e+00 $\pm$ 1.28e+00 2.00e+00 (++)
f_7	300,000	6.32e-04 $\pm$ 3.42e-04 5.55e-04	7.20e-04 $\pm$ 2.92e-04 6.63e-04 (=)	7.13e-04 $\pm$ 3.17e-04 6.32e-04 (=)	6.65e-04 $\pm$ 2.75e-04 6.12e-04 (=)	6.55e-04 $\pm$ 2.31e-04 6.21e-04 (=)
f_8	100,000	2.37e+00 $\pm$ 1.66e+01 2.97e-05	6.83e-06 $\pm$ 9.78e-06 3.67e-06 (++)	4.74e+00 $\pm$ 2.32e+01 1.27e-05 (++)	1.19e-05 $\pm$ 1.26e-05 7.81e-06 (++)	2.37e+00 $\pm$ 1.66e+01 5.77e-06 (++)
f_9	100,000	1.00e-04 $\pm$ 4.30e-05 8.62e-05	8.09e-05 $\pm$ 6.26e-05 7.08e-05 (++)	1.18e-04 $\pm$ 1.02e-04 8.28e-05 (=)	9.58e-05 $\pm$ 6.88e-05 8.11e-05 (=)	7.33e-04 $\pm$ 4.35e-03 8.15e-05 (=)
f_{10}	50,000	1.24e-09 $\pm$ 1.22e-09 8.18e-10	4.63e-10 $\pm$ 2.87e-10 4.41e-10 (++)	5.54e-10 $\pm$ 4.27e-10 4.16e-10 (++)	6.39e-10 $\pm$ 5.38e-10 4.92e-10 (++)	6.96e-10 $\pm$ 6.87e-10 4.78e-10 (++)
f_{11}	40,000	1.48e-04 $\pm$ 1.04e-03 5.25e-12	1.66e-04 $\pm$ 1.04e-03 7.47e-13 (=)	2.23e-05 $\pm$ 1.25e-04 1.10e-12 (+)	1.48e-04 $\pm$ 1.04e-03 5.97e-13 (++)	1.24e-11 $\pm$ 5.64e-11 3.80e-13 (++)
f_{12}	50,000	4.76e-17 $\pm$ 1.99e-16 2.92e-18	3.22e-18 $\pm$ 1.09e-17 3.29e-19 (++)	7.02e-18 $\pm$ 3.45e-17 4.41e-19 (++)	9.16e-18 $\pm$ 2.77e-17 3.88e-19 (++)	4.96e-18 $\pm$ 1.30e-17 3.86e-19 (++)
f_{13}	50,000	8.16e-17 $\pm$ 2.00e-16 1.40e-17	3.29e-17 $\pm$ 9.39e-17 7.89e-18 (++)	3.46e-17 $\pm$ 7.24e-17 7.44e-18 (=)	2.76e-17 $\pm$ 9.85e-17 4.82e-18 (++)	1.34e-16 $\pm$ 5.86e-16 5.18e-18 (=)
+			10	9	11	10
=			3	4	2	3
-			0	0	0	0

中央値については、PRGX($p = 2$) では $f_2, f_3, f_5, f_6, f_7, f_{13}$ の 6 関数で最良の結果となった。PRGX($p = 3$) では f_1, f_4, f_6, f_{11} の 4 関数、CGX では f_6, f_8, f_9, f_{12} の 4 関数、PRGX($p = 1$) では f_{10} で最良の結果となった。JADE は最良の結果を示せなかった。

安定性と中央値を考慮すると PRGX($p = 2$) が最も優れていると考えられる。

6 おわりに

本研究では、多項式近似により得られた決定係数に基づいて遺伝子をグループ化し、グループ単位で遺伝子を交叉させる交叉 PRGX を提案した。PRGX は従来の交叉よりも収束速度を速める傾向があり、問題によっては早期収束を起こしやすくなる。この問題に対処するため、PRGX と BX を併用し、PRGX の適用確率を適応的に制御する方法を導入した。DE の代表的手法である JADE に PRGX を導入し、適用確率を適応的に制御することにより、様々な関数において JADE よりも優れた性能が得られることを示した。また、相関係数に基づくグループ交叉である CBX とも比較し、PRGX がより性能が高く、安定した方法であることも示した。

PRGX の問題点として、多項式近似に時間がかかることが挙げられる。時間を短縮するために、多項式近似を 30 世代に 1 回に制限しているが、それでも毎世代相関係数を求める CGX より 3~4 倍の時間がかかっている。また、関数近似や相関係数を用いる場合、依存関係が無い変数間に相関が現れることがある。このようなノイズに対応するために、パラメータ Sr や閾値を導入したが、さらにノイズを軽減するために、優良個体のみの相関を取る方法などについて調べる予定である。PRGX は 2 つのベクトルを組み合わせる仕組みを有するアルゴリズム。例えば Particle Swarm Optimization などに応用可能であると考えられるため、DE 以外のアルゴリズムへの導入を検討したいと考えている。

謝辞

本研究は、JSPS 科研費 19K04916 の助成を受けて行われた。

参考文献

- [1] Storn, R. and Price, K.: Minimizing the Real Functions of the ICEC'96 Contest by Differential Evolution, in *Proc. of the International Conference on Evolutionary Computation*, pp. 842–844 (1996).
- [2] Storn, R. and Price, K.: Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization*, Vol. 11, pp. 341–359 (1997).
- [3] 阪井節子, 高濱徹行: 相関係数を用いた遺伝子のグループ化に基づくグループ交叉を導入した適応的差分進化, 不確実性・不確定性の下における数理的意思決定の理論と応用, 数理解析研究所講究録 2158, pp. 110–119 (2020).
- [4] Cai, Y. and Wang, J.: Differential evolution with hybrid linkage crossover, *Information Sciences*, Vol. 320, pp. 244–287 (2015).
- [5] Zhang, J. and Sanderson, A. C.: JADE: Adaptive Differential Evolution With Optional External Archive, *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 5, pp. 945–958 (2009).
- [6] Takahama, T., Sakai, S. and Iwane, N.: Solving Nonlinear Constrained Optimization Problems by the ε Constrained Differential Evolution, in *Proc. of the 2006 IEEE Conference on Systems, Man, and Cybernetics*, pp. 2322–2327 (2006).
- [7] Munetomo, M. and Goldberg, D. E.: A genetic algorithm using linkage identification by non-linearity check, in *Proc. of the 1999 IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 1, pp. 595–600 (1999).
- [8] Devicharan, D. and Mohan, C. K.: Particle swarm optimization with adaptive linkage learning, in *Proceedings of the 2004 Congress on Evolutionary Computation*, Vol. 1, pp. 530–535 (2004).
- [9] Takahama, T. and Sakai, S.: A New Binomial Crossover Considering Correlation Among Decision Variables for Adaptive Differential Evolution, in *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems and 19th International Symposium on Advanced Intelligent Systems*, pp. 467–473 (2018).
- [10] Yao, X., Liu, Y., Liang, K.-H. and Lin, G.: Fast Evolutionary Algorithms, in Ghosh, A. and Tsutsui, S. eds., *Advances in Evolutionary Computing: Theory and Applications*, pp. 45–94, Springer-Verlag New York, Inc., New York, NY, USA (2003).