

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

A Correlation-Based Group-Wise Crossover Operation Based on
Gene Grouping using Correlation Coefficients
for an Adaptive Differential Evolution Algorithm

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分布として、細い楕円形状の分布がある。このような場合に優れた子個体を生成するには、楕円形状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。本研究では、探索点の分布から相関行列を求め、相関の強い遺伝子をグループ化し、グループ単位で遺伝子を同時に交叉する(あるいは、交叉しない)交叉であるCGX (Correlation-based Group-wise Crossover)を提案する。CGXは相関に基づく交叉であるCBX (Correlation-based Binominal Crossover), GBX (Grouping-based Binomial Crossover)を発展させたものである。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関係の強い問題がある。このような問題を最適化する場合に、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、楕円形状の長軸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。

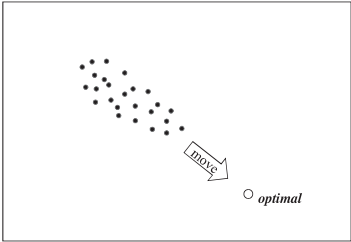


図 1: 探索点集団の最適解への移動

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。

本研究では、探索点の分布から相関行列を求め、相関の強い遺伝子をグループ化し、グループ単位で遺伝子を同時に交叉する（あるいは、交叉しない）交叉である CGX (Correlation-based Group-wise Crossover) を提案する。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、CBX (Correlation-based Binomial Crossover)[3] と GBX (Grouping-based Binomial Crossover)[4] を提案してきた。CBX と GBX では簡易的なグループ化を提案したが、CGX ではより詳細なグループ化とグループに基づく交叉を提案する。なお、本研究は Cai らの研究 [5] に基づいており、グループ化には Adaptive Grouping を拡張して相関行列に適応したものを、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[6] が提案されている。本研究では、JADE に CGX を導入する。幾つかのベンチマーク問題を最適化し、BX, CBX, GBX 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. で関連研究を紹介する。3. で DE および JADE について簡潔に説明する。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 関連研究

2.1 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の研究として、遺伝的アルゴリズムを対象に LINC (Linkage Identification by Nonlinearity Check)[7] が、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 [5] 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測るために、 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。

$$\Delta f_i = f(\cdots, x'_i, \cdots, x_j, \cdots) - f(\cdots, x_i, \cdots, x_j, \cdots) \quad (1)$$

$$\Delta f_j = f(\cdots, x_i, \cdots, x'_j, \cdots) - f(\cdots, x_i, \cdots, x_j, \cdots) \quad (2)$$

$$\Delta f_{ij} = f(\cdots, x'_i, \cdots, x'_j, \cdots) - f(\cdots, x_i, \cdots, x_j, \cdots) \quad (3)$$

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。

$$\Delta f_{ij} = \Delta f_i + \Delta f_j \quad (4)$$

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。

$$e_{ij} = |\Delta f_{ij} - (\Delta f_i + \Delta f_j)| \quad (5)$$

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 LINC を用いた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。

これに対して、相関行列を用いる方法は、余分な関数評価を必要としないため、このような問題にも適用することができる。

2.2 グループ化

Cai ら [5] 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の擬似コードを図 3 に示す。この交叉では、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、 $\mathbf{x}^{\text{child}}$ は生成される子個体、 $u(0, 1)$ は区間 $[0, 1]$ の一様乱数を生成する関数である。

```

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

```

図 2: Adaptive Grouping の擬似コード

```

for( $i = 1$ ;  $i \leq \text{numG}$ ;  $i++$ )
  if( $u(0, 1) < CR$ )
    for( $j \in \text{Group}_i$ )  $x_j^{\text{child}} = m_j$ ;
  else
    for( $j \in \text{Group}_i$ )  $x_j^{\text{child}} = x_j$ ;

```

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

2.3 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化と交叉について説明する。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. 各世代において第 i 遺伝子と第 j 遺伝子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。

$$\rho^{\text{strong}} = \bar{\rho} + Sr\sigma_\rho \quad (6)$$

3. $\rho_{ij} > \rho^{\text{strong}}$ ならば、第 i 遺伝子と第 j 遺伝子間に強い依存関係があると判定する。

CBX のアルゴリズムは以下になり、擬似コードを図 4 に示す。ここで、 $\text{randint}(1, D)$ は区間 $[1, D]$ 上の整数乱数を生成する関数である。

- 必ず交叉させる遺伝子座 j_{rand} を全ての次元 $[1, D]$ からランダムに選択し、子ベクトル x_i^{child} の第 j_{rand} 遺伝子は変異ベクトル m_i から必ず継承する。一巡するまで、各遺伝子に対して次の操作を実行する。その遺伝子が、直前に交叉した遺伝子と強い依存関係を持っていれば、同じグループとして交叉させる。その遺伝子が、直前に交叉しなかった遺伝子と強い依存関係にあれば、同じグループとして交叉させない。それ以外の場合は、交叉率 CR で交叉させるかどうかを決定する。

```

jrand=randint(1,D);
k0 = -1;    // index of the non-crossed gene
k1 = jrand; // index of the crossed gene
j = jrand;
for( $l = 1$ ;  $l \leq D$ ;  $l++$ ) {
  if( $j == jrand$ ) cross = 1;
  else if( $\rho_{k_1, j} > \rho^{\text{strong}}$ ) cross = 1;
  else if( $k_0 \neq -1$  &&  $\rho_{k_0, j} > \rho^{\text{strong}}$ ) cross = 0;
  else if( $u(0, 1) < CR$ ) cross = 1; else cross = 0;
  if(cross) {  $x_j^{\text{child}} = m_j$ ;  $k_1 = j$ ; }
  else     {  $x_j^{\text{child}} = x_j$ ;  $k_0 = j$ ; }
  j = j%D + 1;
}

```

図 4: CBX (Correlation-based Binomial Crossover) の擬似コード

GBX のアルゴリズムは以下のようになり、擬似コードを図 5 に示す。

- 必ず交叉させる遺伝子座 j_{rand} を全ての次元 $[1, D]$ からランダムに選択し、子ベクトル $\mathbf{x}_i^{\text{child}}$ の第 j_{rand} 遺伝子座は変異ベクトル $\mathbf{m}_i$ から必ず継承する。一巡するまで、各遺伝子に対して次の操作を実行する。 j_{rand} 遺伝子と強い依存関係にある遺伝子と同じグループとし、交叉させる。これ以外の場合は、交叉率 CR により交叉するかどうかを決める。さらに、その遺伝子と最も相関が強い遺伝子がまだ他のグループに属していなければ同じグループとし、交叉させるかどうかを決める。ある遺伝子座 k (変数 x_k) に対して最も相関が強い遺伝子座 m_{ρ_k} (変数 $x_{m_{\rho_k}}$) は以下のように決定する。

$$m_{\rho_k} = \arg \max_{j \neq k} \rho_{kj} \quad (7)$$

```

for(j=1; j ≤ D; j++) cross[j]=1; // flags for crossover
jrand=randint(1,D);
cross[jrand]=1; // always cross at jrand
j=jrand%D+1;
for(l=1; l ≤ D-1; l++) {
  if(cross[j] >= 0); // already in a group
  else if(ρjrand,j > ρstrong) cross[j]=1; // jrand's group
  else if(u(0,1) < CR) cross[j]=1;
  else cross[j]=0;
  if(cross[mρj]==-1 && ρj,mρj > ρstrong)
    cross[mρj]=cross[j];
  j=j%D+1;
}
for(j=1; j ≤ D; j++) {
  if(cross[j]==1) xjchild = mj;
  else xjchild=xj;
}

```

図 5: GBX (Grouping-based Binomial Crossover) の擬似コード

本研究では、CBX や GBX より明確なグループ化を行う。

3 差分進化と JADE

3.1 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) を使い、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。

3.2 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 [9]。

- 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。

$$\mathbf{m} = \mathbf{x}_{r1} + F(\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (8)$$

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. 図 6 に BX のアルゴリズムを示す.

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.

```
binomial crossover DE/./bin
j_rand=randint(1,D);
for(j=1; j ≤ D; j++) {
  if(j == j_rand || u(0,1) < CR) x_j^child = m_j;
  else x_j^child = x_ij;
}
```

図 6: 二項交叉.

3.3 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 が次式に示されているように独立に生成される.

$$F_i \sim C(\mu_F, \sigma_F) \quad (9)$$

$$CR_i \sim N(\mu_{CR}, \sigma_{CR}^2) \quad (10)$$

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.

$$\mu_F = (1 - c)\mu_F + cS_{F2}/S_F \quad (11)$$

$$\mu_{CR} = (1 - c)\mu_{CR} + cS_{CR}/S_N \quad (12)$$

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.

$$\mathbf{m}_i = \mathbf{x}_i + F_i(\mathbf{x}_{pbest} - \mathbf{x}_i) + F_i(\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (13)$$

ここで, $\mathbf{x}_{pbest}$ は上位 100% 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.

4 提案手法

4.1 相関係数を用いたグループ化

BX では, 変異ベクトルの要素が子個体に継承される確率を交叉率 CR で指定するため, 全ての要素について同じ確率で子個体に継承されることになる. しかし, 変数間の依存関係が強い問題においては, 依存関係が強い要素を同時に継承しなければ優良な子個体を生成することは困難であると考えられる. このため本研究では, 変数間の相関関係に基づき, 相関の強い要素を同時に継承できる交叉を提案する.

2 変数間の相関の強さを測る指標として相関係数がある. これを D 変数間に拡張したものが相関行列であり, その要素は相関係数である. 個体群 $\{\mathbf{x}_i | \mathbf{x}_i = (x_{ij}), j = 1, 2, \dots, D, i = 1, 2, \dots, N\}$ とし, 相

関行列 $R = (r_{kj})$ を定義する。第 k 次元の変数 x_k と第 j 次元の変数 x_j の相関係数 r_{kj} は、以下のよう
に求めることができる。

$$r_{kj} = \frac{\frac{1}{N} \sum_{i=1}^N (x_{ik} - \bar{x}_k)(x_{ij} - \bar{x}_j)}{\sigma_k \sigma_j} \quad (14)$$

$$\bar{x}_j = \frac{1}{N} \sum_{i=1}^N x_{ij}, \quad \sigma_j = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_{ij} - \bar{x}_j)^2} \quad (15)$$

変数間の依存関係としては、相関係数が 1 の時は正の相関があり、-1 の時は負の相関があるが、要素の
継承という観点からはいずれも強い相関があると判断できる。このため、相関係数の絶対値 ρ_{kj} を相関の
強さとする。

$$\rho_{kj} = |r_{kj}| \quad (16)$$

ρ_{kj} の平均値 $\bar{\rho}$ と標準偏差 σ_ρ を用いて変数間の依存関係が強いかどうかを判定する。

$$\bar{\rho} = \frac{2}{D(D-1)} \sum_{k=1}^D \sum_{j < k} \rho_{kj}, \quad \sigma_\rho = \sqrt{\frac{2}{D(D-1)} \sum_{k=1}^D \sum_{j < k} (\rho_{kj} - \bar{\rho})^2} \quad (17)$$

このために、アルゴリズムパラメータ S_r を導入し、以下の条件式が満足されれば依存関係が強いと判断
する。

$$\rho_{kj} > \rho^{strong}, \quad \rho^{strong} = \max\{\bar{\rho} + S_r \sigma_\rho, 0.15\} \quad (18)$$

本研究では相関係数が非常に小さい場合に依存関係があると判定することを避けるため、閾値として 0.15
を採用した。 $S_r=0$ ならば、平均値 $\bar{\rho}$ より大きく、0.15 よりも大きければ依存関係が強いと判断されるこ
とになる。

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ことになる。図 7 に擬似コードを示す。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。

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

図 7: 本論文で用いる Grouping の擬似コード

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 8 に CGX の擬似コードを示す。

```
 $i_{rand} = \text{randint}(1, \text{numG})$ ;
for(i=1; i ≤ numG; i++) {
  if( $i == i_{rand}$  ||  $u(0, 1) < CR$ ) cross[i]=1;
  else cross[i]=0;

  for(j=1; j ≤ D; j++) {
    if(cross[G_j]==1)  $x_j^{child} = m_j$ ;
    else  $x_j^{child} = x_j$ ;
  }
}
```

図 8: CGX の擬似コード

4.2 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
[3]。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。

$$R_{CBX} = \begin{cases} R_{CBX} + \Delta R, & \text{if CBX の成功率} > \text{BX の成功率} \\ R_{CBX} - \Delta R, & \text{if CBX の成功率} < \text{BX の成功率} \\ R_{CBX}, & \text{otherwise} \end{cases} \quad (19)$$

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。

4.3 アルゴリズム

BX と CGX 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、CGX 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。CGX の適用確率 $R_{CGX} = 0.5$ 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 から相関係数 r_{kj} 、相関の強さ ρ_{kj} 、その平均値 $\bar{\rho}$ と標準偏差 σ_{ρ} , ρ^{strong} を求める。ただし、 $k, j \in \{1, 2, \dots, D\}$ である。

Step4 DE 操作

R_{CGX} の確率で CGX か BX かを選択し、CGX 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 CGX 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_{CGX} 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 9 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 [6]。 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。

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 [6] に基づいて決定した。ただし、 f_5 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 $\frac{1}{2}$ にしている。CGX を CBX, GBX 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。

```

JADE/current-to-pbest/1/BX+CGX()
{
+  $\mu_F^K = \mu_{CR}^K = 0.5$ ,  $K=0,1$ ;
+  $\sigma_F = \sigma_{CR} = 0.1$ ;
+  $R_{CGX} = 0.5$ ;
// 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 CBX
+  $(r_{kj}) = \text{Correlation matrix of } P \text{ is obtained by Eq.(14)}$ ;
+  $(\rho_{kj})$  is obtained by Eq.(16);
+  $\rho^{strong} = \max\{\bar{\rho} + S_r \sigma_\rho, 0.15\}$ ;
for( $i=1$ ;  $i \leq N$ ;  $i++$ ) {
+ if( $u(0,1) < R_{CGX}$ )  $K=1$ ; // CGX
+ 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( $F_i > 1$ )  $F_i = 1$ ;
+  $x_{pbest} = \text{Randomly selected from top 100p\% in } P$ ;
+  $x_{r1} = \text{Randomly selected from } P(r1 \notin \{i\})$ ;
+  $x_{r2} = \text{Randomly selected from } P(r2 \notin \{i, r1\})$ ;
+  $m_i = x_i + F_i(x_{pbest} - x_i) + F_i(x_{r1} - x_{r2})$ ;
+ if( $K==1$ )
+  $x_i^{child} = \text{CGX between } x_i \text{ and } m_i$ ;
+ else
+  $x_i^{child} = \text{binomial crossover between } x_i \text{ and } m_i$ ;
+  $FE = FE + 1$ ;
// 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( $s^1 > s^0$ )
+  $R_{CGX} = R_{CGX} + \Delta R$ ;
+ else if( $s^0 > s^1$ )
+  $R_{CGX} = R_{CGX} - \Delta R$ ;
+ if( $R_{CGX} > 0.95$ )  $R_{CGX} = 0.95$ ;
+ else if( $R_{CGX} < 0.05$ )  $R_{CGX} = 0.05$ ;
+ }
+ }
}

```

図 9: 提案手法の擬似コード

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 $	$[-10, 10]^D$
$f_3(\mathbf{x}) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^D$
$f_4(\mathbf{x}) = \max_i \{ x_i \}$	$[-100, 100]^D$
$f_5(\mathbf{x}) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^D$
$f_6(\mathbf{x}) = \sum_{i=1}^D [x_i + 0.5]^2$	$[-100, 100]^D$
$f_7(\mathbf{x}) = \sum_{i=1}^D i x_i^2 + \text{rand}[0, 1)$	$[-1.28, 1.28]^D$
$f_8(\mathbf{x}) = \sum_{i=1}^D -x_i \sin \sqrt{ x_i }$ $+ D \cdot 418.98288727243369$	$[-500, 500]^D$
$f_9(\mathbf{x}) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$
$f_{10}(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right)$ $- \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$	$[-32, 32]^D$
$f_{11}(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{D}} \right) + 1$	$[-600, 600]^D$
$f_{12}(\mathbf{x}) = \frac{1}{25} [10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2$ $\{1 + 10 \sin^2(\pi y_{i+1})\} + (y_D - 1)^2]$ $+ \sum_{i=1}^D u(x_i, 10, 100, 4)$ where $y_i = 1 + \frac{1}{4}(x_i + 1)$ and $u(x_i, a, k, m) =$ $\begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-50, 50]^D$
$f_{13}(\mathbf{x}) = 0.1[\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2$ $\{1 + \sin^2(3\pi x_{i+1})\} + (x_D - 1)^2$ $\{1 + \sin^2(2\pi x_D)\}] + \sum_{i=1}^D u(x_i, 5, 100, 4)$	$[-50, 50]^D$

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、CGX が最良の結果となっており、 $f_1, f_2, f_5, f_6, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}$ の 10 関数で JADE よりも有意に優れている。GBX では、 $f_1, f_2, f_3, f_5, f_6, f_8, f_9, f_{10}, f_{12}, f_{13}$ の 10 関数で JADE よりも有意に優れているが、 f_4 で有意に劣っている。CBX では、 $f_1, f_2, f_5, f_6, f_8, f_9, f_{10}, f_{12}, f_{13}$ の 9 関数で有意に優れている。

中央値については、 f_{11} の結果はすべて同じであるため、それ以外の関数について比較する。CGX では $f_1, f_2, f_6, f_9, f_{10}, f_{12}, f_{13}$ の 7 関数で最良の結果となった。GBX では f_3, f_4, f_6, f_8 の 4 関数で最良の結果となり、CBX では f_6 のみで最良の結果となった。JADE は f_4, f_7 の 2 関数で最良の結果となっている。

安定性と中央値を考慮すると CGX が最も優れていると考えられる。

6 おわりに

本研究では、相関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 CGX を提案した。CGX 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、CGX と BX を併用し、CGX 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 CGX 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、GBX と比較し、CGX がより性能が高く、安定した方法であることも示した。

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 S_T 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。相関は直線的な分布を捉えることができるが、2 次関数などの非線形分布は把握できないため、これに対処する方法について考察する予定である。CGX 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。

表 2: 適用確率に適応制御を導入した実験結果

	$FE_{\max}$	JADE	CBX	GBX	CGX
f_1	150,000	9.38e-59 $\pm$ 6.5e-58 5.83e-72	2.49e-60 $\pm$ 1.7e-59 1.45e-72 (+)	1.62e-61 $\pm$ 1.1e-60 1.82e-71 (++)	5.39e-68 $\pm$ 1.5e-67 3.85e-75 (++)
f_2	200,000	4.19e-31 $\pm$ 2.4e-30 1.72e-47	9.34e-32 $\pm$ 6.5e-31 8.16e-49 (++)	2.62e-37 $\pm$ 1.7e-36 2.15e-49 (++)	1.02e-41 $\pm$ 6.9e-41 8.88e-51 (++)
f_3	500,000	8.17e-62 $\pm$ 3.0e-61 8.77e-69	1.15e-61 $\pm$ 5.0e-61 5.68e-68 (=)	5.22e-85 $\pm$ 3.6e-84 3.51e-93 (++)	1.26e-61 $\pm$ 4.9e-61 2.14e-68 (=)
f_4	500,000	2.01e-23 $\pm$ 9.8e-23 1.58e-30	5.15e-24 $\pm$ 1.5e-23 1.02e-28 (=)	2.28e-23 $\pm$ 8.9e-23 1.68e-29 (-)	2.68e-24 $\pm$ 8.7e-24 5.81e-30 (=)
f_5	150,000	5.83e-01 $\pm$ 3.6e+00 1.23e-12	3.19e-01 $\pm$ 1.1e+00 1.06e-22 (++)	2.39e-01 $\pm$ 9.5e-01 2.56e-30 (++)	4.78e-01 $\pm$ 1.3e+00 6.38e-19 (++)
f_6	10,000	3.02e+00 $\pm$ 1.3e+00 5.00e-01	2.26e+00 $\pm$ 1.3e+00 0.00e+00 (++)	2.38e+00 $\pm$ 1.4e+00 0.00e+00 (+)	8.80e-01 $\pm$ 9.1e-01 0.00e+00 (++)
f_7	300,000	6.04e-04 $\pm$ 2.4e-04 7.27e-05	6.90e-04 $\pm$ 2.6e-04 1.75e-04 (=)	6.71e-04 $\pm$ 2.6e-04 1.25e-04 (=)	6.30e-04 $\pm$ 2.2e-04 1.34e-04 (=)
f_8	100,000	2.37e+00 $\pm$ 1.7e+01 2.43e-06	2.37e+00 $\pm$ 1.7e+01 1.10e-07 (++)	2.37e+00 $\pm$ 1.7e+01 0.00e+00 (++)	1.46e-08 $\pm$ 5.9e-08 3.41e-13 (++)
f_9	100,000	1.01e-04 $\pm$ 3.9e-05 2.21e-05	6.77e-05 $\pm$ 4.4e-05 3.69e-06 (++)	8.62e-06 $\pm$ 1.1e-05 3.07e-10 (++)	5.53e-06 $\pm$ 8.4e-06 2.00e-10 (++)
f_{10}	50,000	9.20e-10 $\pm$ 6.4e-10 8.67e-11	4.21e-10 $\pm$ 2.9e-10 4.57e-11 (++)	5.11e-10 $\pm$ 3.7e-10 1.92e-11 (++)	8.63e-11 $\pm$ 7.3e-11 9.17e-12 (++)
f_{11}	50,000	1.15e-08 $\pm$ 6.9e-08 0.00e+00	7.21e-07 $\pm$ 5.0e-06 0.00e+00 (=)	1.48e-04 $\pm$ 1.0e-03 0.00e+00 (=)	1.97e-04 $\pm$ 1.4e-03 0.00e+00 (++)
f_{12}	50,000	2.40e-16 $\pm$ 1.6e-15 3.06e-21	4.23e-18 $\pm$ 1.5e-17 3.82e-21 (+)	1.77e-18 $\pm$ 3.0e-18 2.83e-21 (++)	4.91e-20 $\pm$ 1.0e-19 7.53e-23 (++)
f_{13}	50,000	1.15e-16 $\pm$ 2.2e-16 3.75e-19	5.01e-17 $\pm$ 1.7e-16 3.22e-20 (++)	1.50e-17 $\pm$ 2.8e-17 3.55e-20 (++)	2.15e-18 $\pm$ 8.0e-18 6.83e-22 (++)
+			9	10	10
=			4	2	3
-			0	1	0

謝辞

本研究は, JSPS 科研費 17K00311, 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] 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).
- [4] 高濱徹行, 阪井節子: 差分進化における相関係数に基づく遺伝子のグループ化とグループ単位の交叉の提案, 情報処理学会 第 120 回数理モデル化と問題解決研究会, 第 2018-MPS-120(8) 巻, pp. 1–6 (2018).
- [5] Cai, Y. and Wang, J.: Differential evolution with hybrid linkage crossover, *Information Sciences*, Vol. 320, pp. 244–287 (2015).
- [6] 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).
- [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., 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).
- [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).