

外向きベクトル率と平行体交叉による 実数値遺伝的アルゴリズムの探索効率向上

Improving a Real-Coded Genetic Algorithm Using an Outward Vector Rate and Parallelotope-Shaped Crossover

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

実数値遺伝的アルゴリズムにおける代表的な交叉であるブレンド交叉 $BLX-\alpha$ は、2親を含む超直方体の拡張領域に子を生成する2親交叉であり、実装が容易で多様性に優れているため、多数のアプリケーションで使用されている。本研究では、 $BLX-\alpha$ を拡張し、2親を平行体の対角頂点とする平行体の拡張領域に子個体を一様に生成する平行体交叉 psBLX を提案する。さらに、集団が収束しているか移動しているかを判定するために外向きベクトル率を提案する。収束しているときはより狭い範囲に子個体を生成し、移動しているときは優良な親の周辺に子個体を生成するという方法を採用することにより探索効率の向上を目指す。本研究の有効性を示すために、様々なベンチマーク関数を用いて提案手法の性質を調べ、 $BLX-\alpha$ と比較する。

1 はじめに

進化的アルゴリズム (Evolutionary Algorithm, EA) は、生物進化の過程をモデル化した最適化アルゴリズムの総称であり、遺伝的アルゴリズム (Genetic Algorithm, GA)[1], 進化戦略 (Evolution Strategy, ES), 差分進化 (Differential Evolution, DE)[2] など多くのアルゴリズムが提案されている。EA は、最適化の対象である目的関数の値だけを利用して解を求めることができる直接探索法であり、アルゴリズムの実装が容易であることから、様々な最適化問題を解くために利用されている。

EA における重要な遺伝的操作として交叉 (crossover) がある。交叉は複数の親個体から子個体を生成する操作であり、様々な交叉が提案されている。実数値遺伝的アルゴリズムにおける代表的な交叉はブレンド交叉 $BLX-\alpha$ (Blend crossover)[3] である。ブレンド交叉は2親交叉であり、2親を対角頂点とする超直方体を拡大した領域に子個体を一様に生成するため、実装が容易で多様性に優れている。

本研究では、 $BLX-\alpha$ を拡張した新しい2親交叉として、2親を平行体の対角頂点とする平行体の拡張領域に子個体を一様に生成する平行体交叉 psBLX (Parallelotope-shaped blend crossover) を提案する。図1に示すように、2次元の場合、ブレンド交叉は拡張された長方形内に子個体を生成する。これに対して、平行体交叉は平行体内に子個体を生成するため、ブレンド交叉より細い子個体分布を実現でき、子個体の生成範囲の調整がしやすいという特徴がある。

さらに、本研究では、候補解集団の状態として、収束している (Converging) か移動している (Moving) かを検出する方法を提案する。状態検出を行うためには、集団がどのような方向に向かおうとしているかを推定する必要がある。このために、元となる親個体と生成された子個体に一对一の対応関係が存在することを想定する。親個体と子個体を比較し、子個体が良い場合に親個体から子個体への移動ベクトルを求める。このベクトルが集団の重心に近づく場合を内向きベクトル (inward vector)、重心から離れる場合を外向きベクトル (outward vector) と呼ぶことにする。集団が収束しているときは集団は重心方向に収束することが想定されるため、内向きベクトルの割合が多くなる。これに対して、集団が大きく離れた最適解に向かって移動するときは、重心よりも最適解から離れた半数程度が重心に近づき、残りが重心から離れると想定できる。この様子を図2に示す。

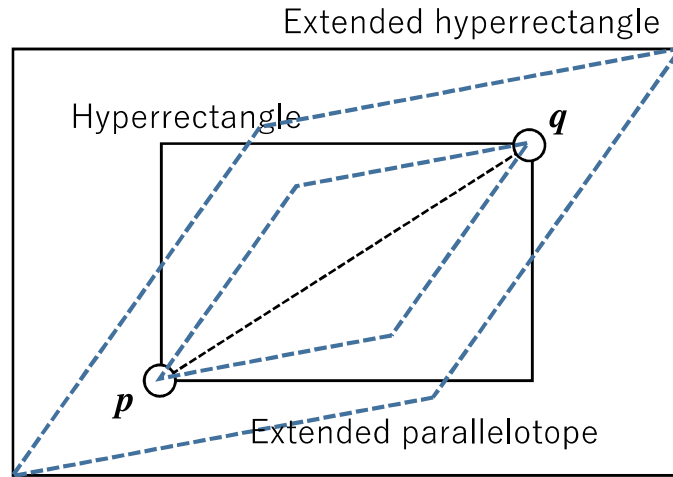


図 1: Area of children generated by BLX and psBLX

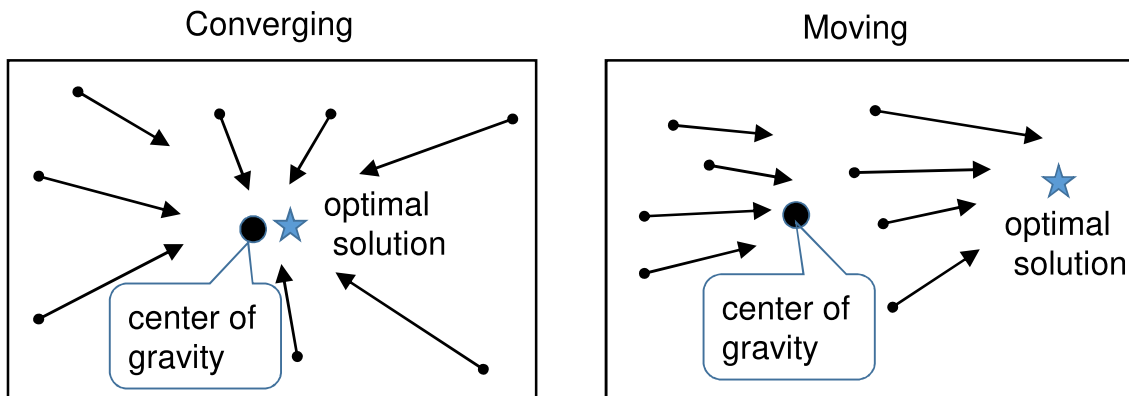


図 2: Inward and outward vectors in Converging and Moving states.

したがって、外向きベクトルの割合が半数を超えた場合は Moving に分類し、外向きベクトルの割合が低いときは Converging に分類できる。分類のための指標として、外向きベクトル率 OVR(Outward Vector Rate) を提案する。OVR は、全移動ベクトルに占める外向きベクトルの割合である。

本研究では、OVR を用いて Converging か Moving かを検出し、Converging の場合は収束速度を速めるように子個体の生成範囲を縮小する。Moving の場合は子個体の生成範囲を拡大して多様性を確保するという手法を提案する。提案手法の有効性を示すために、様々なベンチマーク関数を用いて提案手法の性質を調べる。

本論文の構成は次の通りである。2. で代表的な交叉について簡単に説明する。3. で状態検出の関連研究を説明する。4. で提案手法について説明する。5. にベンチマーク関数に関する実験結果を示す。6. はまとめである。

2 交叉

2.1 代表的な交叉

実数値進化的アルゴリズムにおける代表的な交叉の概略を表 1 に示す。計算量は、次元数が D のとき、子個体の一つ生成するために必要な時間計算量である。

算術交叉は 2 親を結ぶ直線上に子個体をランダムに生成する交叉である。2 親の間に子個体を生成するため、多様性が急速に失われやすいという問題がある。1 点交叉、2 点交叉、一様交叉は主に 2 値の進化的アルゴリズムで採用されている交叉である。2 親のいずれかの遺伝子を継承して子個体を生成するため、2 親を対角頂点とする超直

表 1: Crossover operations

交 叉	親個体	形状	子個体	計算量
算術交叉 (arithmetic crossover)	2 親	直線	2 親を結ぶ直線上に一様乱数で生成	$O(D)$
1 点交叉, 2 点交叉, 一様交叉 (one-point, two-point, uniform crossover)	2 親	超直方体	2 親を対角頂点とする超直方体の頂点の一つを選択	$O(D)$
BLX- α (blend crossover)	2 親	超直方体	2 親を対角頂点とする超直方体 (の拡張領域) 内に一様乱数で生成	$O(D)$
SBX (simulated binary crossover)	2 親	2 峰性確率分布	2 親をピークとする多項式分布に従う乱数で生成	$O(D)$
OBX (oblique crossover)	2 親	平行体	ランダムに選択した差分ベクトルを辺とする平行体内に一様乱数で生成	$O(D^2)$
UNDX- m (unimodal normal distribution crossover)	多親	釣鐘形	$m + 1$ 親の重心から m 個の親へのベクトルと, 残りの次元数の正規直交基底に沿った正規乱数で生成	$O(D^2)$
SPX (simplex crossover)	多親	単体	親を頂点とする単体内に一様分布で生成	$O(D^2)$
REX (real-coded ensemble crossover)	多親	平行体, 釣鐘形確率分布など	親の重心から親へのベクトルに沿った一様乱数や正規乱数で生成	$O(D^2)$

方体の頂点の一つを選択することになる。BLX- α は 2 親を対角頂点とする超直方体の拡張領域内に一様乱数で子個体を生成する交叉である。拡張領域は 2 親を含み, 元の超直方体の $1 + 2\alpha$ 倍の大きさの辺を持つため, 多様性を維持しやすい交叉である。SBX (Simulated Binary Crossover)[4] は 2 親を中心として多項式分布に基づいて子個体を生成する。OBX (Oblique crossover)[5] は, ランダムに選択した個体間の差分ベクトルによって 2 親を対角頂点とする平行体を構成し, 2 親を含む拡張領域に子個体を生成する。3 個以上の親を用いる多親交叉も提案されている。UNDX- m (Unimodal Normal Distribution Crossover)[6], SPX (Simplex Crossover)[7], REX (Real-Coded Ensemble Crossover)[8] では, 親個体集合を選択し, その重心から各親個体に向かうベクトルを軸として一様分布や正規分布に基づいて子個体を生成する。

本研究では, BLX- α を拡張し, 平行体形状の領域に一様乱数で子個体を生成するため, BLX- α についてより詳しく説明する。

2.2 ブレンド交叉

BLX- α は, 2 つの親 p, q が構成する超直方体を拡張した領域に子を生成する交叉であり, 子 x' は以下のように生成される。

$$x'_j = p_j + r_j(q_j - p_j) \quad (1)$$

r_j は区間 $[-\alpha, 1 + \alpha]$ の一様乱数であり, 次元毎に独立に生成される。拡張率 α ($\alpha \geq 0$) は, 2 親が対角位置にある超直方体をどれだけ拡張するかを指定するパラメータである。もし α が 0 ならば, 2 親が構成する超直方体の内部に生成される。 α の推奨値は 0.5 である。2 次元における BLX- α の様子を図 3 に示す。次元数を D とすると, 次元毎に式 (1) の計算を行うため, 時間計算量のオーダーは $O(D)$ である。

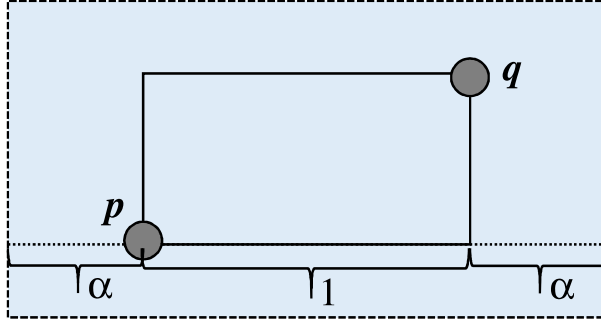


図 3: Area of children generated by BLX- α

3 探索状態の検出

探索状態あるいは進化状態の推定 (Evolutionary State Estimation, ESE) は、古くから研究されてきている探索における exploitation と exploration のバランス制御と強い関係がある [9]. これは、集団の多様性を観測し、初期集団のように多様性が大きいときは探索モードを exploitation に設定して有望な解の近傍を探索し、集団の多様性が失われて来たら探索モードを exploration に設定して新しい領域を探索するというような制御である. 例えば, [10] では、集団の重心からの正規化された距離を多様性尺度として採用し、その値が閾値 d_{high} を超えていれば探索モードを exploitation に設定し選択と交叉を適用する、閾値 d_{low} 未満ならば探索モードを exploration に設定し突然変異を適用するという方法が提案されている.

多様性尺度としては様々な尺度が提案されている. 距離に基づく多様性尺度としては、以下が提案されている [11]. ただし、 $\|\mathbf{x} - \mathbf{y}\|$ は 2 つのベクトル \mathbf{x} と \mathbf{y} の距離でありユークリッド距離を使用することが多い.

- Diameter:
集団中の最大距離 $\max_{i \neq j \in [1, N]} \|\mathbf{x}_i - \mathbf{x}_j\|$. ただし、 N は集団サイズである.
- Radius:
集団の重心との最大距離 $\max_{i \in [1, N]} \|\mathbf{x}_i - \mathbf{g}\|$. \mathbf{g} は重心 $\mathbf{g} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$ である.
- Normalized average distance:
探索領域の広さで正規化した集団の重心との平均距離 $\frac{1}{LN} \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{g}\|$. L は探索空間の広さ $L = \|\mathbf{u} - \mathbf{l}\|$ である. ただし、 \mathbf{l} は探索空間の下限、 \mathbf{u} は探索空間の上限である.

正規化しても閾値は集団サイズの影響を受けるため、閾値の設定が難しいという課題がある.

ESE は多様性を観測するのではなく、集団の探索状態を推定することを目的とする. 文献 [12] では、探索を 4 つの状態 Convergence, Exploitation, Exploration, Jumping-out に分類する. このために、最良個体と他の個体との位置関係に着目した進化因子 (evolutionary factor) を提案した. 集団が収束しているときには最良個体は他の個体に最も近く、集団が移動しているときには最良個体は他の個体から最も離れた位置にあると想定し、以下のように定義した.

$$d_i = \frac{1}{N-1} \sum_{j=1, j \neq i}^N \|\mathbf{x}_i - \mathbf{x}_j\| \quad (2)$$

$$EF = \frac{d_{best} - d_{\min}}{d_{\max} - d_{\min}} \quad (3)$$

ここで、 d_i は \mathbf{x}_i と他の個体との平均距離、 d_{best} は最良個体と他の個体との平均距離、 d_{\min} は d_i の最小値、 d_{\max} は d_i の最大値である. EF が 0 に近ければ Convergence, EF が 1 に近ければ Jumping-out に分類する. 全ての個体間の距離を計算するため、時間計算量は $O(N^2D)$ である. N は通常 D の倍数に設定するため、 $O(N^2D) = O(N^3)$ である. このため、計算量の小さい指標が必要であると考えられる.

文献 [13] では、最良個体との距離と関数値に基づき確率的に探索を 2 つの状態 exploitation と exploration に分類するために以下の最適化状態尺度 (Indicator of the Optimization State) を提案した。

$$IOS = \sum_{i=1}^N \|f_i - d_i\| \quad (4)$$

$$\overline{IOS} = \frac{IOS - IOS_{\min}}{IOS_{\max} - IOS_{\min}} \quad (5)$$

d_i は個体 x_i と最良個体との距離のランクで、距離が近いほどランクが小さい。 f_i は関数値のランクで、関数値が良いほどランクが小さい。 IOS_{\min} は IOS の最小値で、距離と関数値のランクが同一の場合 ($IOS=0$) である。 IOS_{\max} は IOS の最大値で、距離と関数値のランクが逆の場合 ($f_i + d_i = N + 1$) である。 IOS の確率で exploration, $1 - IOS$ の確率で exploitation に分類する。最良個体が最適解付近にあり距離が離れると関数値が悪くなる場合に IOS は小さくなるが、集団が最適解から離れており最良個体先頭になって最適解に移動している場合にも距離が離れると関数値が悪くなると考えられるため、Convergence と Jumping-out の分類が難しいと考えられる。

本研究では、親個体から子個体への移動ベクトルに着目し、外向きベクトル率に基づいて Convergence, Jumping-out, その他に分類することになる。

4 提案手法

4.1 平行体形状を持つブレンド交叉

本研究で提案する平行体交叉 psBLX(parallelootope-shaped BLX) を定義する。個体集団 $P = \{x_i \mid i = 1, 2, \dots, N\}$ を用いて親ベクトル x_p と x_q から子 x' を生成する。 N は個体数である。ベクトルの次元を D , d をベクトル x_p から x_q への差分ベクトル, d_j を第 j 成分のみからなるベクトルとすると以下が成立する。

$$d = x_q - x_p = (d_1, d_2, \dots, d_D) \quad (6)$$

$$d_j = (0, \dots, 0, d_j, 0, \dots, 0) \quad (7)$$

d_j は BLX における基底となるベクトルである。

平行体の一辺に対応するベクトル e_j を以下のように定義する。

$$e_j(\beta) = \frac{1}{2}(d_j + d_{j+1}) + \frac{1}{2}\beta(d_j - d_{j+1}) \quad (8)$$

$$d = \sum_j e_j(\beta) \quad (9)$$

ただし, $d_{D+1} = d_1$, $\beta(\geq 0)$ はパラメータである。 $\beta = 1$ ならば $e_j(1) = d_j$ となるため、BLX と完全に一致する。 $\beta = 0$ ならば $e_j(0) = \frac{1}{2}(d_j + d_{j+1})$ となるため、子個体の分布は接続された線上に限定されるため、算術交叉に近い交叉となる。BLX の超直方体の内部に平行体を生成する場合には、 $0 \leq \beta \leq 1$ であるが、外部にも生成するときは $\beta > 1$ も可能である。 $D = 2$ の時の様子を図 5 に示す。

得られた $e_j(\beta)$ に基づき子 x' を生成する。

$$x' = x_p + \sum_{k=1}^D r_k e_k(\beta) \quad (10)$$

r_k は軸ベクトル毎に生成される一様乱数であり、BLX- α と同様に $[-\alpha, 1 + \alpha]$ の範囲で生成する。 $e_j(\beta)$ は以下のように求めることができる。

$$e_j(\beta) = \frac{1}{2}(1 + \beta)d_j + \frac{1}{2}(1 - \beta)d_{j+1} \quad (11)$$

ベクトルの計算を行っているように見えるが、実際には $j, j + 1$ の 2 次元の計算を D 回繰り返せば良いため、時間計算量のオーダーは $O(D)$ である。

図 5 に疑似コードを示す。

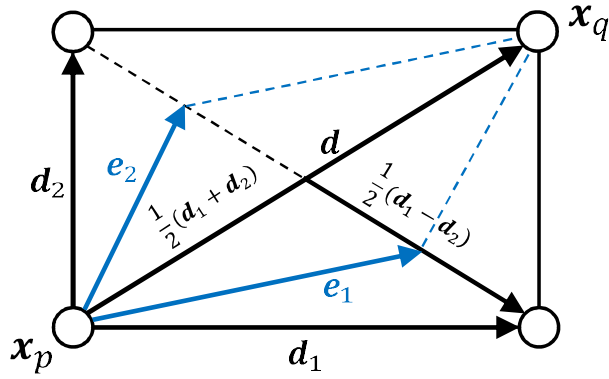


図 4: Axis vectors and a parallelogram in 2D

```

void psBLX(x', alpha, beta, x_p, x_q)
{
  for(j=1; j<=D; j++) {
    d_j=x_qj - x_pj;
    x'_j=x_pj;
  }
  for(j=1; j<=D; j++) {
    j1=j + 1;
    if(j1 > D) j1=1;
    r=-alpha + (1 + 2alpha) * u(0,1);
    x'_j=x'_j + r * 0.5 * (1 + beta) * d_j;
    x'_{j1}=x'_{j1} + r * 0.5 * (1 - beta) * d_{j1};
  }
}

```

図 5: The pseudo-code of psBLX, $u(0,1)$ is a random number in $[0,1]$.

4.2 外向きベクトル率

外向きベクトル率 (OVR) を定義する. 親個体から子個体への移動ベクトルが重心に近づいていけば内向きベクトル, 遠ざかっていけば外向きベクトルに分類し, 外向きベクトルの割合を求めればよい. このためのアルゴリズムを以下に示す.

- 1) 集団の重心 g を求める. 内向き, 外向きベクトルのカウンター n_{in}, n_{out} を 0 にする.
- 2) 各親個体を x_i , 子個体を x'_i とし, 子個体が親個体より良くなければ, 5) に進む.
- 3) x_i, x'_i と重心の距離 d_{pg}, d_{cg} を求める. 親個体から子個体への移動ベクトルは, $x'_i - x_i$ である.
- 4) $d_{cg} < d_{pg}$ ならば移動ベクトルは内向きベクトルであり, n_{in} を 1 増加させる. $d_{cg} > d_{pg}$ ならば移動ベクトルは外向きベクトルであり, n_{out} を 1 増加させる.
- 5) 全ての個体の処理が終了するまで 2) に戻る.
- 6) 以下の式で OVR を求める.

$$OVR = \frac{n_{out}}{n_{in} + n_{out}} \quad (12)$$

なお, 急激な OVR の変化を避けるために, 指数平滑移動平均を用いて OVR の値を平滑化して用いる. 平滑化定数は 0.5 である. 各個体と重心の距離を計算するため, 時間計算量は $O(ND) = O(N^2)$ である.

4.3 OVR に基づくパラメータ制御

OVR に基づいて集団の状況を Converging, Examining, Moving に分類する。これは, Convergence, Exploitation/Exploration, Jumping-out に対応する。本研究では Converging と Moving を判定してパラメータ制御を行い, その他の場合は BLX ($\beta=1$ の psBLX) をそのまま適用することにした。OVR の値と状態の関係およびパラメータ制御を以下に示す。

- $OVR \in [0, 0.1)$: 状態を strongly Converging と判断し, 収束速度を速めるために, $\beta=0.7$ とする。
- $OVR \in [0.1, 0.2)$: 状態を Converging と判断し, 収束速度を少し速めるために, $\beta=0.8$ とする。
- $OVR \in [0.5, 0.6)$: 状態を Moving と判断し, 多様性を確保するために, $\beta=1.05$ とする。
- $OVR \in [0.6, 1]$: 状態を strongly Moving と判断し, 多様性を強化するために, $\beta=1.1$ とする。

第 1 世代では OVR の値が求まっていないため, この制御は第 2 世代から適用する。

4.4 アルゴリズム

本研究における実数値 GA の擬似コードを図 6 に示す。各個体を必ず親の一つとして選択し, 子が良ければ親と置換するという差分進化と同様の方法を採用している。

```
GA with psBLX using OVR()
{
// Initialize a population
P=N individuals generated randomly in S;
FE=FE+N;
OVR=0;
for(t=1; FE < FEmax; t++) {
  nin=nout=0;
  g=center of gravity of P;
  for(i=1; i ≤ N; i++) {
    if(t>1) set β according to OVR;
    xr=randomly selected from P (r ≠ i).
    psBLX(xi, α, β, xi, xr);
    FE=FE+1;
// Survivor selection
    if(f(xi') < f(xi)) {
      zi = xi';
      dpg=||xi - g||; dcg=||xi' - g||;
      if(dcg<dpg) nin++;
      else if(dcg>dpg) nout++;
    }
    else zi = xi;
  }
  P = {zi};
  if(nin+nout>0) {
    if(t==1)
      OVR=nout/(nin + nout);
    else
      OVR=0.5 OVR+0.5 nout/(nin + nout);
  }
}
}
```

図 6: The pseudo-code of real-coded GA with psBLX using OVR, where \mathcal{S} is the search space.

5 実験

5.1 テスト問題

表 2 に、テスト問題の関数定義とその初期化領域を示す。なお、 D は次元数を表している。

表 2: Test functions of dimension D . These are sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rosenbrock, step, noisy quartic, Schwefel 2.26, Rastrigin, Ackley, Griewank, and two penalized functions, respectively[14]

Test functions	Boundary
$f_1(\mathbf{x}) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$
$f_2(\mathbf{x}) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10, 10]^D$
$f_3(\mathbf{x}) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^D$
$f_4(\mathbf{x}) = \max_i \{ x_i \}$	$[-100, 100]^D$
$f_5(\mathbf{x}) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^D$
$f_6(\mathbf{x}) = \sum_{i=1}^D [x_i + 0.5]^2$	$[-100, 100]^D$
$f_7(\mathbf{x}) = \sum_{i=1}^D ix_i^4 + \text{rand}[0, 1)$	$[-1.28, 1.28]^D$
$f_8(\mathbf{x}) = \sum_{i=1}^D -x_i \sin \sqrt{ x_i } + D \cdot 418.98288727243369$	$[-500, 500]^D$
$f_9(\mathbf{x}) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$
$f_{10}(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$	$[-32, 32]^D$
$f_{11}(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]^D$
$f_{12}(\mathbf{x}) = \frac{\pi}{D} [10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 \{1 + 10 \sin^2(\pi y_{i+1})\}] + (y_D - 1)^2 + \sum_{i=1}^D u(x_i, 10, 100, 4)$ 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 $N = 100$ 、 $\alpha = 0.5$ を用いる。各関数について 50 回の試行を行い、結果を比較する。

実験結果を表 3 に示す。各関数に対して、最大関数評価回数、各試行における最良値の平均値と標準偏差を上段に、中央値を下段に示した。関数によっては、各試行における結果の変動が大きい場合、平均値ではなく中央値による比較を重視し、最良の中央値を示した結果を太字で表示した。さらに、Wilcoxon signed rank test を行い、BLX- α 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、中央値の右の括弧内に示した。

中央値については、psBLX($\beta = 0.8$) が 7 関数 $f_1, f_2, f_7, f_{10}-f_{13}$ で最良の結果となった。OVR による制御は 3 関数 f_3, f_4, f_5 で最良の結果となった。psBLX($\beta = 0.7$) は 3 関数 f_6, f_8, f_9 で、psBLX($\beta = 1.1$) は 1 関数 f_5 で最良の結果となった。BLX は最良の結果を得ることができなかった。

平均値については、psBLX($\beta = 0.8$) が 6 関数 $f_1, f_2, f_7, f_{10}, f_{12}, f_{13}$ で、OVR による制御が 3 関数 f_3, f_4, f_5 で、psBLX($\beta = 0.7$) が 3 関数 f_6, f_8, f_9 で最良の結果となった。BLX が 1 関数 f_{11} で最良の結果となった。psBLX($\beta = 1.1$) は最良の結果を得ることができなかった。

有意差については、OVR による制御が f_8 を除く 12 関数で BLX より有意に優れており、有意に劣った関数はなかった。psBLX($\beta = 0.8$) は 10 関数で BLX より有意に優れており、3 関数 f_3, f_4, f_5 で有意に劣っていた。psBLX($\beta = 0.7$) は 6 関数で BLX より有意に優れており、6 関数で有意に劣っていた。psBLX($\beta = 1.1$) は 11 関

表 3: Results of BLX, psBLX and psBLX with controlling β using *OVR*

func	FEmax	BLX-0.5	psBLX($\beta=0.7$)	psBLX($\beta=0.8$)	psBLX($\beta=1.1$)	Controlling β using <i>OVR</i>
f_1	150000	9.79e-42 \pm 1.0e-41 6.69e-42	2.57e-12 \pm 1.2e-11 5.15e-26 (--)	2.53e-53 \pm 1.4e-52 1.19e-54 (++)	6.85e-32 \pm 4.4e-32 5.91e-32 (--)	2.47e-52 \pm 3.7e-52 1.01e-52 (++)
f_2	200000	6.17e-35 \pm 3.1e-35 5.36e-35	1.36e-44 \pm 2.0e-44 4.63e-45 (++)	4.04e-48 \pm 3.5e-48 3.23e-48 (++)	1.26e-27 \pm 4.7e-28 1.22e-27 (--)	6.52e-42 \pm 4.3e-42 5.33e-42 (++)
f_3	500000	2.01e-02 \pm 1.7e-02 1.39e-02	3.06e+01 \pm 1.8e+01 2.70e+01 (--)	1.28e-01 \pm 2.1e-01 6.08e-02 (--)	1.02e+00 \pm 6.1e-01 9.05e-01 (--)	2.36e-03 \pm 2.0e-03 1.59e-03 (++)
f_4	500000	5.37e-20 \pm 6.7e-20 3.09e-20	2.87e+00 \pm 1.1e+00 2.81e+00 (--)	1.04e+00 \pm 7.0e-01 8.26e-01 (--)	4.96e-14 \pm 3.6e-14 4.04e-14 (--)	1.72e-20 \pm 2.8e-20 6.60e-21 (++)
f_5	300000	2.74e+01 \pm 1.3e+01 2.40e+01	5.52e+01 \pm 3.5e+01 3.02e+01 (--)	3.06e+01 \pm 1.7e+01 2.55e+01 (--)	3.16e+01 \pm 2.0e+01 2.39e+01 (=)	2.71e+01 \pm 1.3e+01 2.39e+01 (+)
f_6	10000	6.29e+01 \pm 1.7e+01 6.35e+01	1.78e+00 \pm 1.9e+00 1.00e+00 (++)	3.60e+00 \pm 1.7e+00 3.50e+00 (++)	2.83e+02 \pm 5.3e+01 2.78e+02 (--)	5.98e+00 \pm 2.3e+00 6.00e+00 (++)
f_7	300000	1.81e-03 \pm 4.4e-04 1.75e-03	1.30e-03 \pm 4.2e-04 1.26e-03 (++)	1.13e-03 \pm 2.7e-04 1.10e-03 (++)	2.80e-03 \pm 6.1e-04 2.84e-03 (--)	1.21e-03 \pm 4.0e-04 1.14e-03 (++)
f_8	100000	5.29e+03 \pm 1.3e+03 5.69e+03	3.72e+03 \pm 1.5e+03 3.64e+03 (++)	4.26e+03 \pm 1.5e+03 4.51e+03 (++)	5.63e+03 \pm 1.1e+03 5.88e+03 (=)	5.00e+03 \pm 1.6e+03 5.44e+03 (=)
f_9	100000	6.07e+01 \pm 3.1e+01 6.24e+01	4.42e+00 \pm 2.3e+00 3.98e+00 (++)	7.63e+00 \pm 8.3e+00 5.97e+00 (++)	1.01e+02 \pm 2.2e+01 1.05e+02 (--)	1.77e+01 \pm 1.5e+01 1.09e+01 (++)
f_{10}	50000	1.37e-06 \pm 2.7e-07 1.34e-06	3.65e-06 \pm 4.8e-06 1.99e-06 (--)	4.37e-09 \pm 2.1e-09 3.99e-09 (++)	6.33e-05 \pm 1.3e-05 6.21e-05 (--)	1.52e-08 \pm 5.9e-09 1.42e-08 (++)
f_{11}	50000	1.42e-10 \pm 1.0e-10 1.02e-10	1.19e-03 \pm 3.4e-03 6.10e-10 (--)	3.01e-04 \pm 1.5e-03 2.22e-15 (++)	2.97e-07 \pm 2.1e-07 2.42e-07 (--)	1.48e-04 \pm 1.0e-03 7.36e-14 (++)
f_{12}	50000	3.13e-11 \pm 3.0e-11 2.05e-11	8.11e-08 \pm 5.7e-07 1.87e-12 (+)	7.48e-17 \pm 8.6e-17 3.75e-17 (++)	1.89e-07 \pm 1.1e-07 1.60e-07 (--)	2.09e-15 \pm 2.5e-15 1.30e-15 (++)
f_{13}	50000	3.25e-10 \pm 2.1e-10 2.60e-10	1.10e-03 \pm 3.3e-03 7.92e-11 (=)	2.43e-15 \pm 3.8e-15 1.17e-15 (++)	2.34e-06 \pm 3.2e-06 1.64e-06 (--)	3.78e-14 \pm 4.6e-14 1.70e-14 (++)
+		—	6	10	0	12
=		—	1	0	2	1
-		—	6	3	11	0

数で BLX より有意に劣っており、有意に優れた関数はなかった。このように、psBLX($\beta = 0.8$) は、子個体の生成領域を狭くすることにより、収束速度を向上することが可能であるが、 $f_3 \sim f_5$ のように多様性を必要とする関数では BLX より劣った結果となった。これに対して、*OVR* に基づく制御を行うことにより、 $f_3 \sim f_5$ において β を大きくして多様性を確保し、その他の関数で β を小さくして収束速度を向上することが可能となったため、ほとんどの関数で BLX より結果が得られたと考えられる。

図 7 に幾つかの関数について BLX と *OVR* による制御を用いた場合の関数値の変化 (左の縦軸) と、*OVR* と β の変化 (右の縦軸) を示す。横軸は関数評価回数であり、最大関数評価回数を 100,000 回とした。関数としては、単峰性関数 f_1 、稜構造を持つ f_5 、乱数を伴う単峰性関数 f_7 、最適解が探索空間の端にある多峰性関数 f_8 、強い多峰性関数 f_9 を選択した。

単峰性関数 f_1 では、*OVR* の値は 0.2 程度で多くの場合 Converging と判定できている。しかし、0.2 を超える場合もあるため、 β の値は 0.8 ではなく 0.9 程度となっている。このため、psBLX($\beta=0.8$) より収束速度が低下していると考えられる。稜構造関数 f_5 における典型的な探索過程は、原点に向かって少し収束し、その後細い谷を通して最適解に移動するというものである。最初 *OVR* は 0.2 以下で Converging と判定し、その後 *OVR* が次第に大きくなり 0.5 程度になり Moving と判定している。したがって、典型的な探索過程に近い判定を行っていると考えられる。 β の値は、最初は 1 以下であるが、次第に 1 を超え、細い谷で収束してしまわないように多様性を重視した探索に切り替わっていると考えられる。ノイズ付き単峰性関数 f_7 では、本来 Converging と判定することが正しいのであるが、ノイズの影響で正しい判定が困難となっており、*OVR* の値の変化が大きい。Converging と判定されることは少なく、Moving と判定されている場合もある。 β の値は 0.9 程度であるため、BLX よりは収束速度を重視した探索を実現できていると考えられる。最適解が端にある多峰性関数 f_8 では、端に向かって大きく移動する必要がある。*OVR* は 0.5 以上であることが多いため、Moving と判定できている。 β の値は 1 近辺であり、BLX と同程度の多様性を保ちながら探索を行っているため、BLX と有意差が生じなかったと考えられる。強い多峰性関数 f_9 では、最初は広域に探索し、最適解が含まれる谷に到達すれば収束するという探索過程となる。*OVR* の値は、0.5 程度で大きく振動しているため、まだ広域探索の状態であり、収束状態に到達していないと考えられる。 β の値も 1 よりやや小さい値から 1 に向かって増加している。関数値からみてもまだ最適解の近くに達していないと考えられる。初期にやや収束的な探索を行ったために、BLX よりも良い局所解を早く見つけたと考えられる。

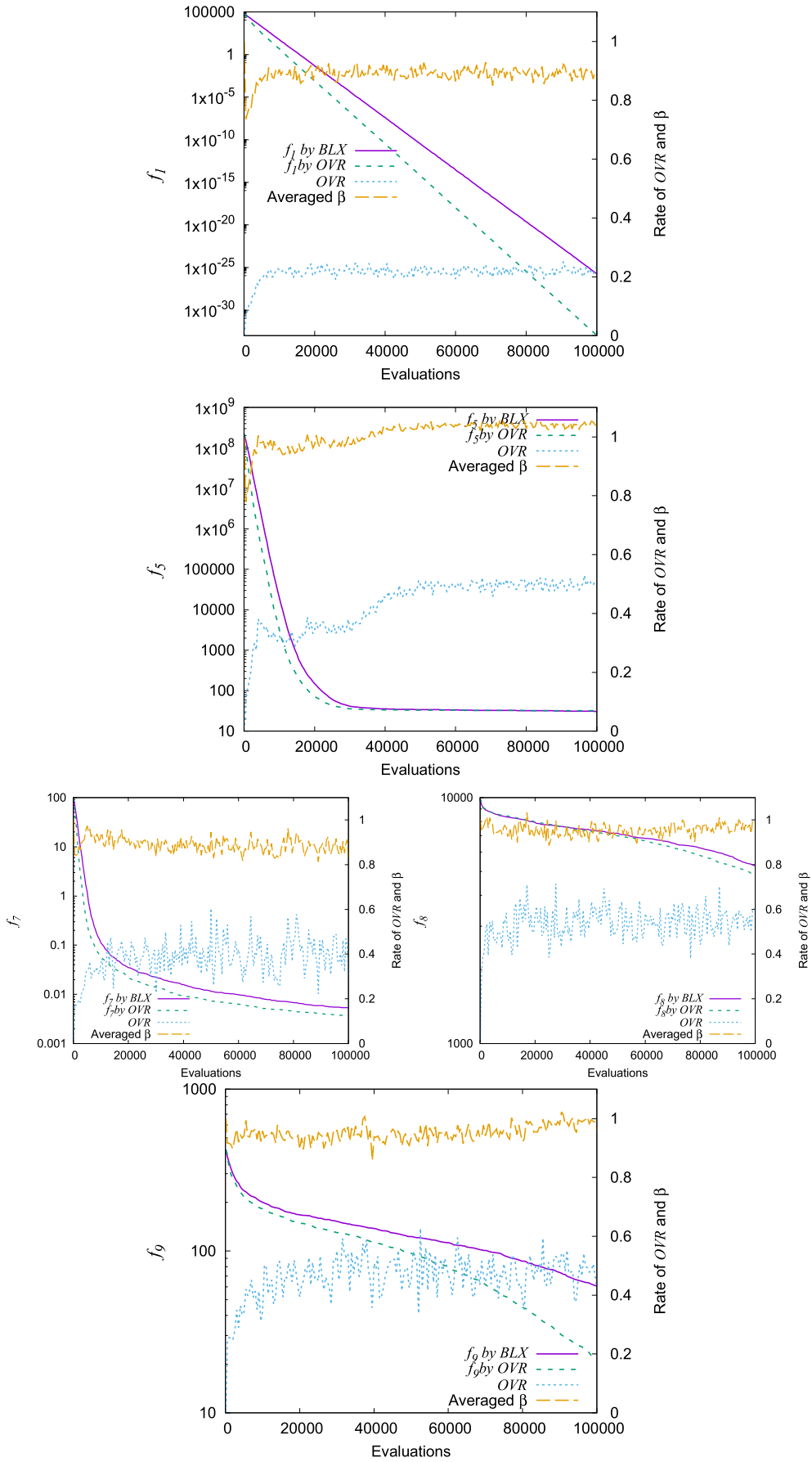


图 7: The graphs of objective values, OVR and β by BLX and OVR .

6 おわりに

本研究では、2親を対角頂点とする平行体内に一様に子個体を生成する2親交叉である psBLX を提案した。psBLX はパラメータ β によって形状を変化させることができる。また、探索状態を検出するために、子が親より良くなった時に親から子に向かうベクトルに着目し、重心から遠ざかる外向きベクトルの割合である OVR を提案した。OVR を用いて探索状態を推定し、探索状態に合わせて psBLX のパラメータである β を制御する方法により、収束を早めたり、移動を強化する手法を考案した。提案手法により、様々な関数を最適化し、BLX と比較して探索効率が大きく向上することを示した。今後は、提案手法を実数値 GA 以外のアルゴリズム、例えば DE や PSO など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, 20K11977 の助成を受けて行われた。

参考文献

- [1] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley (1989).
- [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] Eshelman, L. J. and Schaffer, J. D.: Real-Coded Genetic Algorithms and Interval Schemata, in Whitley, L. D. ed., *Foundations of Genetic Algorithms 2*, pp. 187–202, Morgan Kaufmann Publishers, San Mateo, CA (1993).
- [4] Deb, K. and Agrawal, R. B.: Simulated binary crossover for continuous search space, *Complex systems*, Vol. 9, No. 2, pp. 115–148 (1995).
- [5] 高濱徹行, 阪井節子: 斜交座標系に基づく回転不変なブレンド交叉の提案, 情報処理学会研究報告, 第 2017-MPS-113 巻, pp. 1–6 (2017).
- [6] Kita, H., Ono, I. and Kobayashi, S.: Multi-parental extension of the unimodal normal distribution crossover for real-coded genetic algorithms, *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Vol. 2, pp. 1581–1588 (1999).
- [7] 樋口隆英, 筒井茂義, 山村雅之: 実数値 GA におけるシンプレクス交叉の提案, 人工知能学会誌, Vol. 16, No. 3, pp. 147–155 (2001).
- [8] Akimoto, Y., Nagata, Y., Sakuma, J., Ono, I. and Kobayashi, S.: Proposal and Evaluation of Adaptive Real-coded Crossover AREX, *Trans. of the Japanese Society of Artificial Intelligence*, Vol. 24, No. 6, pp. 446–458 (2009), in Japanese.
- [9] Črepinšek, M., Liu, S.-H. and Mernik, M.: Exploration and exploitation in evolutionary algorithms: A survey, *ACM computing surveys (CSUR)*, Vol. 45, No. 3, pp. 1–33 (2013).
- [10] Ursem, R. K.: Diversity-guided evolutionary algorithms, in *International Conference on Parallel Problem Solving from Nature*, pp. 462–471 Springer (2002).
- [11] Olorunda, O. and Engelbrecht, A. P.: Measuring exploration/exploitation in particle swarms using swarm diversity, in *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*, pp. 1128–1134 IEEE (2008).
- [12] Zhan, Z.-H., Zhang, J., Li, Y. and Chung, H. S.-H.: Adaptive particle swarm optimization, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 39, No. 6, pp. 1362–1381 (2009).
- [13] Yu, W.-J., Shen, M., Chen, neng W., Zhan, hui Z., Gong, Y.-J., Lin, Y., Liu, O. and Zhang, J.: Differential Evolution With Two-Level Parameter Adaptation, *IEEE Transactions on Cybernetics*, Vol. 44, No. 7, pp. 1080–1099 (2014).
- [14] 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).