

Real-Coded Genetic Algorithms-UNDX and SPX

李 翠敏

1 はじめに

遺伝的アルゴリズム (Genetic Algorithm:GA) の最も一般的なコーディングはバイナリーコーディングである。バイナリーコーディングではすべての染色体は、0 または 1 のビットで表現される。しかし、バイナリーコーディングの精度は探索が制限されるなどの問題点があり、連続関数最適化問題の場合にバイナリーコーディングを使うのは非常に難しい。したがって、実数値コーディングを実行して行く。実数値コーディングでは染色体は実数ベクトルで表現される。

実数値 GA のアルゴリズムはたくさんある。そこで、本報告では実数値 GA の代表的な交叉手法に SPX(Simplex Crossover: SPX)¹⁾ と UNDX(Unimodal Normal Distribution Crossover: UNDX)²⁾ について探索性能を比較する。

2 SPX と UNDX の概要

本報告では、生成された子個体のうち次世代に残る個体を選択する手法は MGG(Minimal Generation Gap) を用いる。MGG とは 1 家族 (2 つの親個体と生成個体数分の子個体) の中から、最良 1 個体とランキンクルール選択により選ばれた 1 個体を次世代に残す。

2.1 シンプレックス交叉 (SPX) とは

シンプレックス交叉 (Simplex Crossover: SPX) は、母集団から複数の個体を抽出し、抽出した個体の分布から一様乱数を発生させ新しい個体を生成する交叉である。SPX のアルゴリズムは以下ようになる。

1. 母集団から (n+1) 個の親個体 \vec{P}_0, \vec{P}_n をランダムに選ぶ。
2. (1-1) 式にしたがって親個体の重心 \vec{G} を求める。

$$\vec{G} = \left(\sum_{i=1}^{n+1} \vec{P}_i \right) / (n+1) \quad (1-1)$$

3. 式 (1-2) と (1-3) と (1-4) から、 \vec{x}_k, \vec{C}_k を $k=0, \dots, n$ について求める。e は正のパラメータで拡張率 (Expansion Rate) とよぶ。e の理論値は $\sqrt{n+2}$ であり、その中の n は関数の引数である。 \vec{x}_k は区間 [0,1] の一様分布乱数 $u(0,1)$ を (1-3) 式で変換して得られる乱数である。

$$\vec{x}_k = \vec{G} + e(\vec{P}_k - \vec{G}), (k = 0, 1, \dots, n) \quad (1-2)$$

$$r_k = u(0, 1)^{\frac{1}{k+1}}, (k = 0, 1, \dots, n-1) \quad (1-3)$$

$$\vec{C}_k = \begin{cases} 0, (k = 0) \\ r_{k-1}(\vec{x}_{k-1} - \vec{x}_k + \vec{C}_{k-1}), (k = 1, \dots, n) \end{cases} \quad (1-4)$$

4. 式 (1-5) にしたがって子個体を生成する。

$$\vec{C} = \vec{x}_n + \vec{C}_n \quad (1-5)$$

また、Fig.1 に SPX による子個体生成範囲を示す。

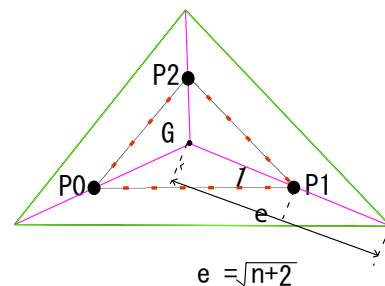


Fig. 1 SPX(出典：参考文献 1 より引用)

2.2 単峰性正規分布交叉-UNDX とは

UNDX は母集団から 3 つ親個体の個体をランダムに抽出し、Fig.2 に示すように 2 つ親個体を結んだ直線の中心近傍に正規分布にしたがって、ランダムに基本的 2 つの子体を生成する。子個体 \vec{C}_1, \vec{C}_2 は以下ようになる。

$$\begin{aligned} \vec{C}_1 &= \vec{m} + z_1 \vec{e}_1 + \sum_{k=2}^n z_k \vec{e}_k \\ \vec{C}_2 &= \vec{m} - z_1 \vec{e}_1 - \sum_{k=2}^n z_k \vec{e}_k \end{aligned} \quad (2-1)$$

$$\vec{m} = (\vec{P}_1 + \vec{P}_2) / 2 \quad (2-2)$$

$$z_1 : N(0, \sigma_1^2); z_2 : N(0, \sigma_2^2) \quad (2-3)$$

$$\sigma_1 = \alpha d_1; \sigma_2 = \beta d_2 / \sqrt{n} \quad (2-4)$$

$$\begin{aligned} \vec{e}_1 &= (\vec{P}_2 - \vec{P}_1) / |\vec{P}_2 - \vec{P}_1| \\ \vec{e}_i &\perp \vec{e}_j (i, j = 1, \dots, n) \end{aligned} \quad (2-5)$$

z_1 と z_2 は正規分布のランダム数字である。 d_1 は P_1 から P_2 までの距離、 d_2 は第 3 の親個体 P_3 から P_1 と P_2 両親を結ぶ主軸方向までの距離である。

また、Fig.2 に UNDX による子個体生成範囲を示す。

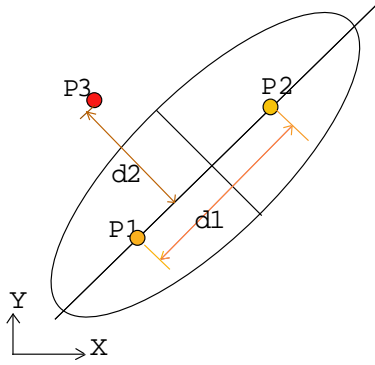


Fig. 2 UNDX(出典：参考文献2より引用)

×	UNDX	SPX
Function	Benchmark function	
Initial Population Size	300	300
Dimension of Function	10	10
Parents Size	3	11
α	0.5	×
β	0.35	×
Children Size	100	100
Threshold	10e-7	10e-7
Termination of evaluation	6e+6	6e+6
Trials	5	5

Table 1 パラメーター

3 SPX と UNDX の交叉手法の実験結果

3.1 パラメーター

実験パラメーターは Table 1 である.

以下の図にはグラフの横軸は評価計算回数, 縦軸は関数値を示している.

3.2 SPX の実験結果

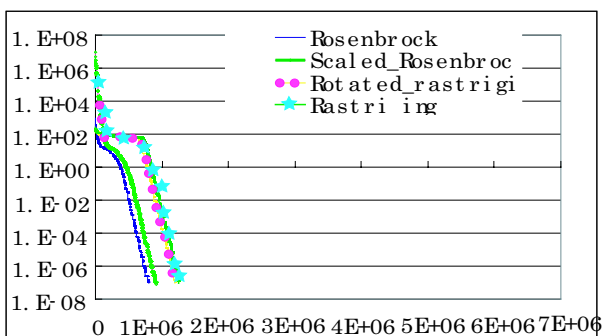


Fig. 3 SPX の実験結果 (出典：自作)

Fig.4にはそれぞれの関数に対する実験結果を示す. 実験の結果, どの関数でも, SPX は最も良好な解を探索できた. さらに, Scaled Rosenbrock と Rotated rastrigin を用いて, 座標系を変更しても最も良好な解を求めることができる. Rastrigin と Rotated rastrigin に比べて Rosenbrock と Scaled Rosenbrock は早く良好な解求めること

ができる.

3.3 UNDX の実験結果

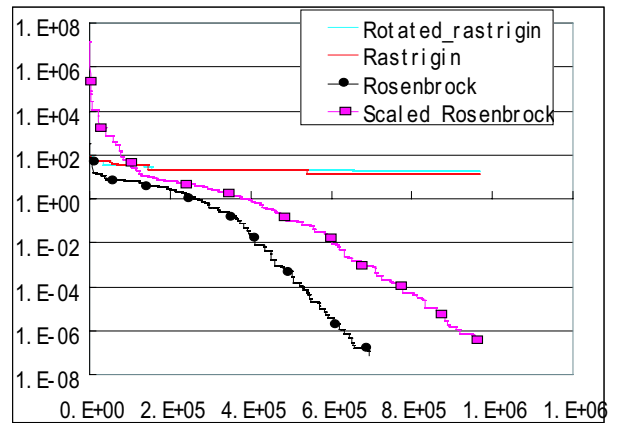


Fig. 4 UNDX の実験結果 (出典：自作)

実験結果によって, UNDX は関数の次元数が 10 の時, Rosenbrock 関数は良い解求めることができたことがわかる. しかし, Rastrigin 関数は多峰の関数のため, 最も良好な解を求めることができなかった.

3.4 UNDX と SPX の比較

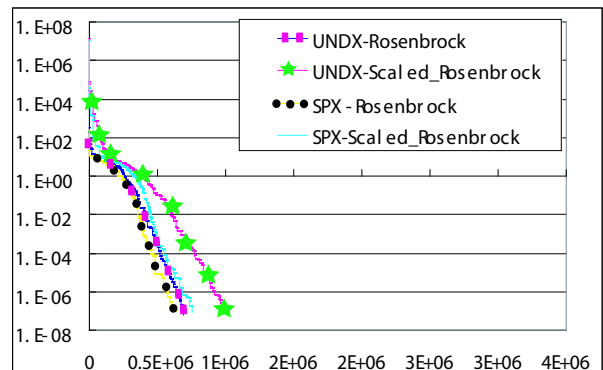


Fig. 5 比較結果 (出典：自作)

UNDX と SPX を Rosenbrock と Scaled-Rosenbrock に関して比較した. UNDX と SPX もどちらも最も良好な解を求めることができる.

参考文献

- 1) Takahide Higuchi, Shigeyoshi Tsutsui, Masayuki Yamamura. *Simplex Crossover for Real-Coded Genetic Algorithms*. AIC Vol.16, No.1, p146-155 (2001).
- 2) Isao Ono, Shigenobu Kobayashi. *A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover* ICGA'97 No.1249 (1997)