

集合差分進化における設定変数に関する考察 Considerations on Setting valuable for Set-based Differential Evolution

國永 優人[†] 前田 道治[†]
Yuto Kuninaga Michiharu Maeda

1. はじめに

本研究では、集合差分進化の交叉方法として Binominal Crossover (BX) を採用する。BX の交叉率を実験的に検討し、他のアルゴリズムとの比較実験を行った。結果として、BX を用いた集合差分進化アルゴリズムは巡回セールスマン問題において有効であることが示された。

2. 集合差分進化

集合差分進化(SBDE)[1]は、連続最適化問題の最適手法である差分進化を離散空間上で動作するように拡張した手法である。SBDE のアルゴリズムについて記述する。

[SBDE アルゴリズム]

<Step 1>初期パラメータ設定

最大繰り返し回数 T_{max} 、個体数 N_p 、スケール因子 F 、交叉率 C_r 、現在の繰り返し回数 k を設定($k \leftarrow 0$)。また各都市間の経路の全体集合を E とする。

<Step 2>突然変異

集合差分進化における突然変異集合演算子を次式に示す。

$$V_i^j = \begin{cases} X_{r_2}^j - X_{r_3}^j & \text{if } (\text{rand}_j[0.0, 1.0] < F) \\ X_{r_1}^j & \text{otherwise} \end{cases} \quad (1)$$

V_i^j は個体 i の j 次元目における変位集合、 $X_{r_1}^j, X_{r_2}^j, X_{r_3}^j$ は個体群からランダムに選ばれた 3 つの異なる個体集合である。

<Step 3>交叉

集合差分進化では、交叉方法として Exponential Crossover (EX) が採用されている。交叉方法を次式に示す。

$$U_i^j = \begin{cases} V_i^j & \text{if } (j \in \{(n)_D, (n+1)_D, \dots, (n+L-1)_D\}) \\ X_i^j & \text{otherwise} \end{cases} \quad (2)$$

U_i^j は個体 i の j 次元目における試行集合である。 $\langle * \rangle_D$ は係数 D のモジュロ関数である。 n は交叉の開始地点であり、 $[1, D]$ の範囲でランダムに設定される。 L は交叉を行う範囲を指し、 $[1, D]$ の範囲で設定される。 L は 1 で初期化した後、 $[1, D]$ の範囲で交叉率 C_r 以上の乱数が生成されるまで L を加算する。ここで得た L と乱数 n によって交叉範囲を決定し、交叉を行う。

<Step 4>選択

選択演算子を次式に示す。

$$X_i^{j,k+1} = \begin{cases} U_i^j & \text{if } f(U_i^j) \leq f(X_i^j) \\ X_i^{j,k} & \text{otherwise} \end{cases} \quad (3)$$

$f(*)$ は適応度関数である。選択ステップでは、現在の個

[†]福岡工業大学大学院工学研究科 Graduate School of Engineering, Fukuoka Institute of Technology

体集合と試行集合を比較し、よりよい解を次の個体 i の個体集合とする。

<Step 5>最良解の更新

各個体の適応度を評価し、各個体と全体の最良解を更新。

<Step 6>終了条件

$k = T_{max}$ ならば終了、そうでなければ $k \leftarrow k + 1$ とし、<Step 2>へ戻る。

3. 設定変数

本研究では集合差分進化の交叉方法として、Binominal Crossover (BX) を採用する。BX による交叉方法を次式に示す。

$$U_i^j = \begin{cases} V_i^j & \text{if } (\text{rand}_j[0.0, 1.0] \leq C_r \text{ or } j = j_{\text{rand}}) \\ X_i^j & \text{otherwise} \end{cases} \quad (4)$$

j_{rand} は $[1, D]$ の範囲で生成された乱数である。各次元で乱数 rand_j を生成し、その値が交叉率以下であれば突然変異集合から学習する。BX では、少なくとも 1 次元(j_{rand} 次元目)は突然変異集合から学習する。

本章では、BX の交叉率 C_r を細かく変化させた時の解精度を比較する。実験は複数のインスタンスに対して行った。ここでは kroB150 と kroB200 の結果を示す。 $F = 0.9$ 、各パラメータの実行回数は 30 回とし、平均値を算出した。図 1 および図 2 は交叉率を 0.1 刻みで変化させたときの解精度、図 3 および図 4 は 0.01 刻みで変化させ、0.5 ごとにプロットしたものである。各図の横軸は交叉率、縦軸は平均距離を示している。

実験結果において、図 1 および図 2 より、交叉率 0.1 を境に、値の増加とともに解の精度が低下する傾向が見られる。一方図 3 および図 4 では、0.05 と 0.1 の交叉率で比較的良好な値が得られている。他のインスタンスでも同様の実験を行った結果、交叉率 0.1 のときに最良ではないものの、最悪解を回避しつつ、平均的に安定した精度を示した。これにより、評価実験において交叉率 0.1 を採用する。

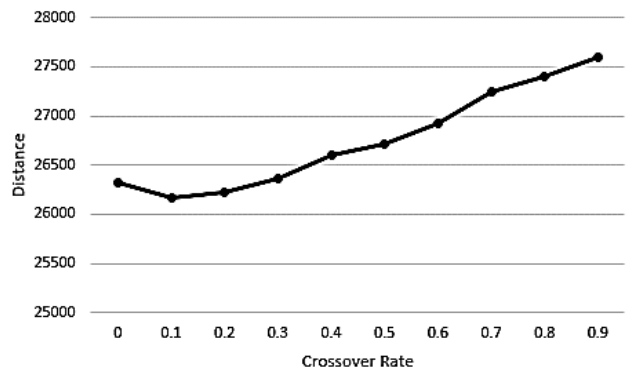


図 1 kroB150 の交叉率

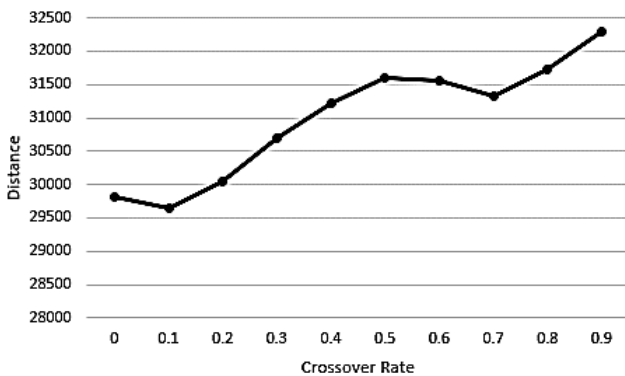


図 2 kroB200 の交叉率

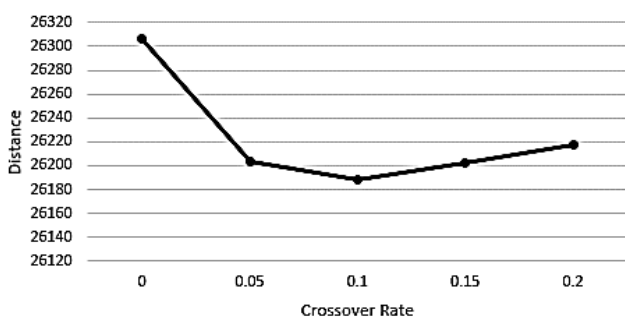


図 3 kroB150 における交叉率

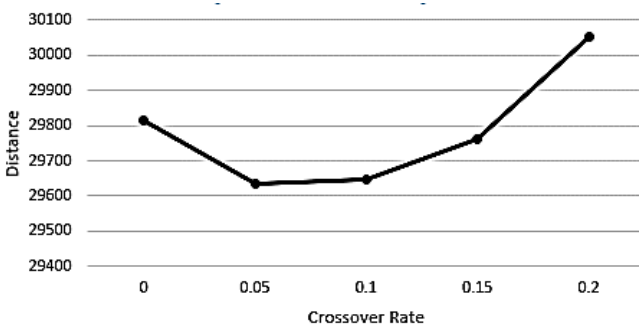


図 4 kroB200 における交叉率

4. 評価実験

本実験は、最大最小アントシステム(MMAS)[2], メモリー二値粒子群最適化(MBPSO)[3], 包括的学習集合粒子群最適化(SCLPSO)[4], 記憶包括的学習集合粒子群最適化(SCLPSOM)[5]と Exponential Crossover を用いた SBDE-EX[1], Binominal Crossover を用いた SBDE-BX の 6 つを実装した。テストデータとして、TSPLIB[6]の st70(最適解: 675), kroB150(最適解: 26130), kroB200(最適解: 29437)を用いた。共通の設定は、都市数を次元数 D , 個体数を $N_p = 20$, 最大繰り返し回数 $T_{max} = 10000D/N_p$ としている。また各アルゴリズムの試行回数は 30 回である。各アルゴリズムのパラメータは次のように設定。MMAS: $\alpha = 1, \beta = 2$, 蒸発係数 $\rho = 0.98$. MBPSO: 慣性荷重 $w = 1.4 - 1.3 * k/T_{max}$. SCLPSO, SCLPSOM: 慣性荷重 $w = 0.9 - 0.5 * k/T_{max}$, 係数 $c = 2.0$. 本手法のパラメータは $rand_j \in [0.0, 1.0]$, スケ

ール因子 $F = 0.9$, 交叉率は SBDE-EX で $C_r = 0.9$, SBDE-BX で $C_r = 0.1$ としている。

各アルゴリズムの実行結果を表 1, 表 2, 表 3 に示す。太字の数値は各インスタンスにおける最も優れた最良値と平均値を表す。結果より, SBDE-BX は各インスタンスに対して他のアルゴリズムよりも平均値, 最良値ともに最良値を出力したことが分かる。

表 1 st70 の実行結果

	最良値	平均値
MMAS	675	684
MBPSO	675	681.33
SCLPSO	675	676.03
SCLPSOM	675	675.83
SBDE-EX	675	675.07
SBDE-BX	675	675

表 2 kroB150 の実行結果

	最良値	平均値
MMAS	26130	26414.63
MBPSO	26228	26556.93
SCLPSO	26130	26301.73
SCLPSOM	26130	26272.57
SBDE-EX	26130	26239.33
SBDE-BX	26130	26180.97

表 3 kroB200 の実行結果

	最良値	平均値
MMAS	29676	29904
MBPSO	30383	31293.5
SCLPSO	29617	29847.73
SCLPSOM	29545	29836.83
SBDE-EX	29529	29722.37
SBDE-BX	29454	29642.47

5. まとめ

本稿では, BX の導入と, パラメータ設定の検討および他のアルゴリズムとの評価実験を行った。これにより SBDE-EX の有用性が示された。

参考文献

- [1] M. Maeda, Y. Chikuba, "An Algorithm of Set-Based Differential Evolution for Discrete Optimization Problem", DMBD 2022, CCIS 1745, Part 2, pp.105-117(2022)
- [2] Stützle T., Hoos H., "Max-min ant system", Future Generation Computer Systems, Vol.16, No.9 (2004).
- [3] Ji Z., Tian T., He S., "A memory binary particle swarm optimization", Proceedings of the IEEE International Conference on Evolutionary Computation, pp.1-5 (2012).
- [4] Chen W.-N., Zhang J., Chung H.S.H., "A novel set-based particle swarm optimization method for discrete optimization problems", IEEE Transactions on Evolutionary Computation, Vol.14, No.2 (2010).
- [5] Y. Chikuba., M. Maeda., "Set-based comprehensive learning and particle swarm optimization with memory for discrete optimization problem", NOLTA, IEICE, Vol.13, No.2 (2022).
- [6] TSPLIB, <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/tsp/>