適応的個体間距離に基づく複数解探索型 Bat Algorithm

発表者: 情報学専攻 メディア情報学 プログラム 学籍番号 1730022 岩瀬 拓哉
指導教員: 髙玉 圭樹 教授, 佐藤 寛之 准教授

1 はじめに

実問題を多峰性最適化問題として捉えた時，一つの最適解だけでなく複数の局所解を探索することは，有力な解候補を複数の選択肢として持つという意味で重要である．このような複数解探索手法として，局所解への収束を防ぐ機構である Niching scheme と進化計算アルゴリズムを組み合わせた Niching method が探究されているが，いずれの手法においても密接した解を除き，探索空間内にランダムに解生成するため，乱数に強く依存するという問題がある．この問題を解決するために，本研究では多点探索アルゴリズムの中でも大域探索と局所探索のバランスを調整可能な Bat Algortihm を採用し，個体間距離に基づく動的変化を考慮した Niching scheme を用いることで，最適解だけでなく局所解も同時に探索可能な複数解探索手法を提案する．具体的には，探索領域を動的に変更することで，個体同士を同じ局所解に留まらせない Bat Algorithm with Dynamic Niche Radius (DNRBA) を考案し，最適解と局所解の数が異なる多峰性関数を用いて他の複数解探索手法との比較を通して提案手法の有効性を検証する．

2 Bat Algorithm

Bat Algorithm(BA) [1] は群知能アルゴリズムの一つで，対象物までの方向や距離を知るコウモリの特性を利用して周囲の状況を認知し，大域探索と局所探索が進むにつれて探索速度を徐々に落とし，探索性能を自動調節することが可能なアルゴリズムである．各個体の周波数 f_i，速度 v_i，位置 x_i は世代数毎に，更新される．ラウドネス A は，個体が対象物に近づくと値が減少し，移動距離も比例して短くなる．探索は以下 3 つで構成される．

- 最良解方向へ探索: 各個体は位置 x_i において，自身が発する周波数 f_i の反響によって対象物との距離を測り，対象物に向かって速度 v_i で移動する．
- 局所探索: 対象物近辺に個体を移動させる．
- ランダム探索: 探索領域内に個体をランダムで移動させる．

3 提案手法

3.1 Dynamic Niche Sharing

探索空間の大きさと局所解数 (あるいは個体数) に基づいて算出される Niche Radius は $\sigma = \frac{\sqrt{(x_{ub}-x_{lb})^2}}{2\sqrt[D]{q}}$ で表される．この時，x_{ub}, x_{lb} は探索空間の上限と下限を表し，D は次元数を表す．q は局所解の数 (あるいは個体数) が適用される．ここでは同じ類似度を持つ個体同士の評価値を比較するため．類似度を次式で求める．

$$sh(d_{ij}) = \begin{cases} 1 - (\frac{d_{ij}}{\sigma})^{\alpha} & (\text{if } d_{ij} < \sigma) \\ 0 & (\text{otherwise}) \end{cases} \tag{1}$$

ここで d_{ij} は個体 i, j の間の距離を表し，α は係数で σ はある恣意的な閾値 (あるいは Niche radius) を表す．個体間距離が近いほど $sh(d_{ij})$ の値は大きくなり，この数値を基に Niche count $m_i = \sum_{j=1}^{N} sh(d_{ij})$ を算出する．m_i は i 番目の個体に対する全個体の密度を表し，この式を用いて Dynamic niche sharing [3] は次式で表される．

$$m_i^{dyn} = \begin{cases} n_j & (\text{if individual } i \text{ is within the dynamic niche } j) \\ m_i & (\text{otherwise}) \end{cases} \tag{2}$$

3.2 Dynamic Niche Radius based Bat Algorithm

局所解に収束しなかった個体を最適解や局所解へ移動させることで，収束性能を高めることを目的とした Dynamic Niche Radius を BA に適用させた DNRBA を提案する．

$$m_i^{dyn} = \begin{cases} \sigma & (\text{if } m_i < \sigma) \\ m_i & (\text{otherwise}) \end{cases} \tag{3}$$

$$\boldsymbol{v_i^{t+1}} = \boldsymbol{v_i^t} + (\boldsymbol{x_i^t} - \boldsymbol{x_{NR*}}) * f_i \tag{4}$$

v_i, x_i は個体の速度と位置を表し，x_{NR*} は x_i が属する Niche Radius 内の最良個体を示す．(3) 式に基づいて，個体の更新式は以下で表される．

$$\boldsymbol{x_i^{t+1}} = \begin{cases} \boldsymbol{x_i^t} + \boldsymbol{v_i^{t+1}} & (\text{if } d_{ij} < \sigma) \\ \boldsymbol{x_i^t} & (\text{otherwise}) \end{cases} \tag{5}$$

ここでは個体の分布密度が高いほど，個体 x_i の持つ Niche Count m_i の範囲内にある最良個体 x_{NR*} から遠ざかる方向

へ新たに解候補を生成する．解候補生成の流れを図 1 に示す．図 1 は 5 つの個体のうち，4 つの個体が同じ Niche radius 内に位置する場合において，その領域内の最良解 x_{NR*} に近い ($m_i < \sigma$) ほど個体 x_i は遠ざかる方向へ移動する．

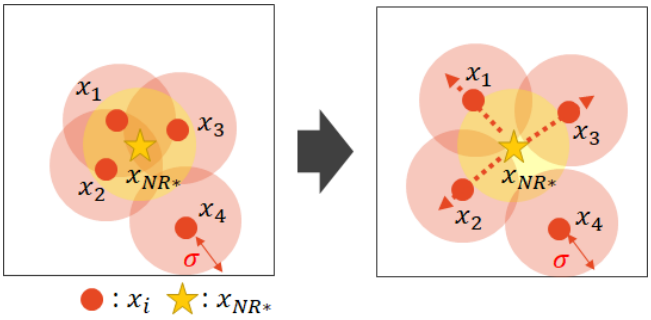

図 1　解候補の生成

局所探索では (3) 式で算出した最良個体 x_{NR*} が持つ m_i の範囲内で個体 x_i が新たに解候補 x_{loc} を生成する．

$$\boldsymbol{x_{loc}} = \boldsymbol{x_{NR*}} + A_i^t * rand(1, D, [-m_i, m_i]) \tag{6}$$

ランダム探索では，個体 x_i の持つ m_i の範囲内で新たに解候補 x_{rnd} を生成することで，局所解への収束性能を高める．生成式は次式で表される．

$$\boldsymbol{x_{rnd}} = \boldsymbol{x_i^t} + rand(1, D, [-m_i, m_i]) \tag{7}$$

4　実験

最適解と局所解を持つベンチマーク関数 F_1 : Griewank, F_2 : Six-Hump-Back-Camel, F_3 : Michalewicz と，局所解を持たず最適解のみ存在する F_4 : Himmelblau を使用し，従来手法と 3 つの提案手法を比較することで探索性能を調べる．評価尺度として発見した解探索率 Peak Ratio (PR) [2] を採用し，次式で表される．

$$PR = \frac{\sum_{run=1}^{NR} NPF_{run}}{NKP * NR} \tag{8}$$

NPF_{run} は，そのシードにおけるアルゴリズムが発見した最適解数を示し，NKP は評価関数が持つ全最適解数を示す．NR は実験の試行回数を示す．解発見の定義は最近傍個体との評価値の差分が閾値 ε によって設定した．本実験では $f_{max} = 1$, $f_{min} = 0$，ラウドネス $A^0 = 1$，パルスレート $r^0 \in [0, 1]$ と設定した．また $\alpha = \gamma = 0.9$ とし，個体数 $N = 50$，世代数を 10000 とし，ランダムシードを変えた実験を 30 回行った．

表 1　PR 値の平均及び標準偏差 (実験回数 50 試行)

Func	BA Mean ± St.D	NSBA Mean ± St.D	NRBA Mean ± St.D	DNRBA Mean ± St.D
F_1	0.351 ± 0.050	0.371 ± 0.041	0.692 ± 0.098	**0.745** ± 0.061
F_2	0.50 ± 0	0.50 ± 0	0.992 ± 0.045	**1** ± 0
F_3	0.50 ± 0	0.50 ± 0	0.70 ± 0.245	**1** ± 0
F_4	0.808 ± 0.108	**1** ± 0	0.858 ± 0.124	0.867 ± 0.127

5　結果と考察

表 1 は $\varepsilon = 0.1$ とした時の PR の平均値をヒートマップ化したものである．全体的に，提案した 3 つの手法の中では DNRBA が最も PR 値が高かったが，局所解が存在しない最適解のみの Himmelblau 関数では NSBA の方が探索性能が高かった．これは DNRBA の同じ局所解への収束を避ける機構が F_1 から F_3 までは有効であったが，最適解のみを持つ F_4 関数では性能が NSBA よりも劣ったと考えられる．

6　おわりに

本研究では最適解だけでなく，複数の局所解を同時に探索可能な DNRBA を提案した．提案手法の有効性を検証するため，評価関数を用いて，他の手法との比較実験を行った．結果，複雑な多峰性を持つ関数においては DNRBA の方が探索性能が高く，複雑な関数において有効であることを示した．

参考文献

[1] X. S. Yang, “A Metaheuristic Bat-Inspired Algorithm”, *in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds J.R. Gonzalez et al.), Studies in Com-putational Intelligence*, Springer Berlin, 284, Springer, 65-74 (2010).

[2] X. Li, A. et.al. , ”Benchmark Functions for CEC’2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization”, *Evol. Comput.* Mach. Learn. Group, RMIT University, Tech. Rep., 2013.

[3] B. Miller, ”Genetic algorithms with dynamic niche sharing for multimodal function optimization”, *IEEE International Conference on Evolutionary Computation*, 1996.