

複数解探索を考慮した分散型 Bat Algorithm

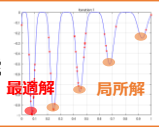
研究奨励賞対象

SS04-10

○岩瀬拓哉 高野諒 上野史 佐藤寛之 高玉圭樹 (電気通信大学)

背景 多峰性最適化における複数解探索

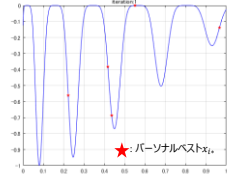
複数の解を保持することで環境変化に適用可能
例) 複数ロボットによる協調的探索



従来 Bat Algorithm(BA) [X.S. Yang, 2010]

STEP1: 最良個体方向へ探索

$$v_i^{t+1} = v_i^t + (x_i^t - x_{i*}^t) * rand$$
$$x_i^{t+1} = x_i^t + v_i^{t+1}$$



STEP2: グローバルベスト近辺を局所探索

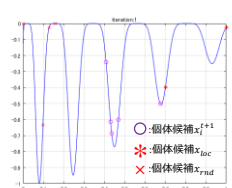
if rand > r_i

$$x_{loc} = x_* + \epsilon A_i^t \quad \epsilon \text{は} [-1, 1] \text{の乱数}$$

endif

STEP3: ランダム探索

$$x_{rnd} = x_{lb} + (x_{ub} - x_{lb}) * rand$$



STEP4: 評価と更新

if rand < A_i^t &

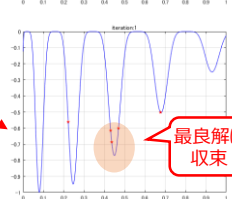
$$\min(F(x_i), F(x_{loc}), F(x_{rnd})) < F(x_{i*})$$

x_{i*}を更新

$$A_i^{t+1} = \alpha A_i^t$$

$$r_i^{t+1} = r_i^t [1 - \exp(-\gamma t)]$$

endif



提案

Niche Radius-based Bat Algorithm(NRBA)

Niche Radius [D. Beasley, et. al., 1993] の導入

$$\lambda = \frac{1}{2} \sqrt{(x_{ub} - x_{lb})^2}$$

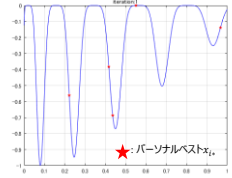
$$NR = \frac{\lambda}{\sqrt{q}}$$

探索範囲の上限と下限: x_{ub}, x_{lb}

次元数: D 解の数: q

STEP1: 最良個体から離れる方向へ探索

$$v_i^{t+1} = v_i^t + (x_i^t - x_{NR*}^t) * rand$$
$$x_i^{t+1} = \begin{cases} x_i^t + v_i^{t+1} & (\text{if } d_i < NR) \\ x_i^t & (\text{otherwise}) \end{cases}$$



STEP2: NR内で局所探索

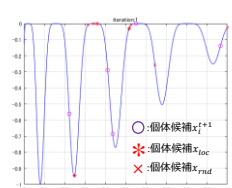
if rand > r_i

$$x_{loc} = x_{NR*} + \epsilon A_i^t$$

endif

STEP3: NR内でランダム探索

$$x_{rnd} = x_{NR*} + *rand(-NR, NR)$$



STEP4: 評価と更新

if rand < A_i^t &

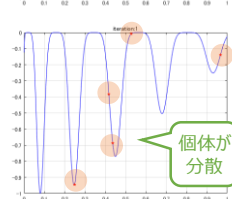
$$\min(F(x_i), F(x_{loc}), F(x_{rnd})) < F(x_{i*})$$

x_{i*}を更新

$$A_i^{t+1} = \alpha A_i^t$$

$$r_i^{t+1} = r_i^t [1 - \exp(-\gamma t)]$$

endif



実験内容

複数解探索性能の比較
BA vs NRBA

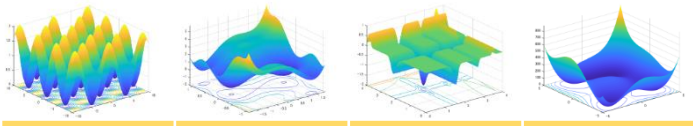
評価指標

$$\text{解発見率} = \frac{\sum_{run=1}^{MR} \text{発見した解の数}}{\text{全最適解及び最適解数} * MR}$$

解発見の定義: (解座標) - (最近傍個体座標) < 0.1

問題設定

解の数/形状が異なる
評価関数を使用



関数	F ₁ : Griewank	F ₂ : Six-Hump Camel	F ₃ : Michalewicz	F ₄ : Himmelblau
探索範囲	$x_i \in [-10, 10]$	$x_1 \in [-2, 2]$ $x_2 \in [-1, 1]$	$x_i \in [0, 4]$	$x_i \in [-5, 5]$
最適解の座標	$x_* = [0, 0]$	$x_* = [\pm 0.0898, \mp 0.7126]$	$x_* = [2.20, 1.57]$	$x_* = [3.2, -2.8051, 3.2832, -3.7793, -3.2832, 3.5845, -1.8481]$
最適解数 / 局所解数	1 / 16	2 / 2	1 / 1	4 / 0

パラメータ設定

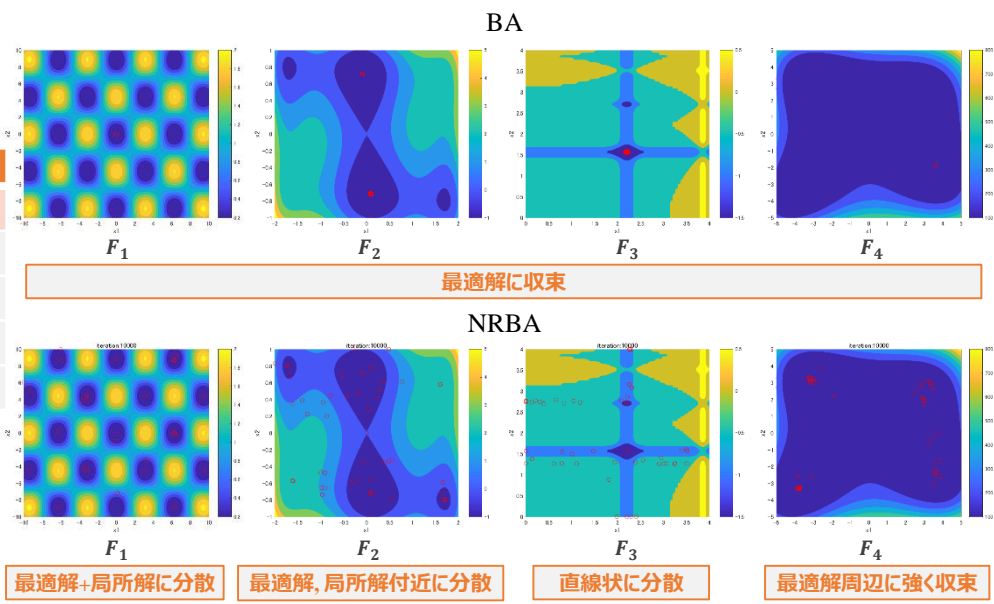
個体数: N	50	実験回数: MR	30
世代数: Iteration	10000	ラウドネス: A ⁰	1
次元数: D	2	パルスレート: r ⁰	rand [0, 1]
		α, γ	0.9

実験結果

発見した解の数 (30シートの平均値)

	BA		NRBA	
関数	Mean ± SD	解発見率	Mean ± SD	解発見率
F ₁	1.0 ± 0	5.88 %	11.77 ± 1.67	69.22 %
F ₂	2.0 ± 0.18	49.17 %	3.97 ± 0.18	99.17 %
F ₃	1.0 ± 0	50.00 %	1.4 ± 0.49	70.00 %
F ₄	0.97 ± 0.55	24.17 %	3.43 ± 0.50	85.83 %

全ての関数においてNRBAの方が
複数解探索性能が高い



最適解+局所解に分散 最適解, 局所解付近に分散 直線状に分散 最適解周辺に強く収束

結論

NRBAは従来手法より複数解探索性能が大きく向上し、最適解だけでなく局所解を保持することが可能