

複数解探索を考慮した分散型 Bat Algorithm

○岩瀬拓哉 高野諒 上野史 佐藤寛之 高玉圭樹 (電気通信大学)

概要 多峰性最適化問題における、従来の多点探索アルゴリズムは一つの最適解に収束する傾向にあるが、実問題への適用を考慮した時に複数の最適解及び局所解を探索する必要がある。多点探索アルゴリズムの中でも大域と局所の探索バランスに優れた Bat Algorithm は探索する上で全個体の最良解を参照して移動するため、最終的に最適解に収束することから複数解を同時に探索することは困難である。本研究では各個体の探索領域を分割させる Niche Radius 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 Bat Algorithm の構築をする。従来手法と提案手法の性能を比較するため、いくつかの多峰性関数を用いてシミュレーション実験を行った結果、全ての関数に対して従来手法は一つの最適解に収束していたが、提案手法では全最適解及び局所解を探索したと見られ、従来に対する変更点が有効であることを示した。

キーワード: 多点探索アルゴリズム, 多峰性最適化問題, 複数解探索, Bat Algorithm

1 はじめに

近年、多点探索アルゴリズムは最適化問題において、一般的なメタヒューリスティック手法として用いられるようになった。多点探索アルゴリズムは特に非線形な問題に対しても適用することが可能であり、魚や鳥の群れをモデルにした Particle Swarm Optimization(PSO)¹⁾ や、ホタルの光強度により互いのホタルが引き寄せられる Firefly Algorithm(FA)²⁾ は高次元な最適化問題に対して有効であることを示している。中でも Bat Algorithm(BA) は大域探索と局所探索の性能を自動で切り替えるという点で優れたアルゴリズムである³⁾。しかし多峰性最適化問題における、従来の多点探索アルゴリズムは全個体の中の最良解を参照して一点へ移動するため、探索終了時に一つの最適解に収束する傾向にあるが、実環境への適用を考慮した時に最適解だけでなく局所解を探索し、保持しておくことは非常に重要な意味を持つ。応用先の一例として、災害時における被災者を解、救助ロボットを個体と見立てた時に、不特定多数の被災者を探索することは人間にとって困難であり、複数の解を同時に探索しなければならない。

そこで本研究では、探索範囲の自動調整可能な BA を用い、各個体の探索領域を分割させることで個体の分散化を図る。探索空間のスケールと解の数から算出される Niche Radius を用いることで、予め各個体の探索領域を決定し、その探索領域内の最良個体から遠ざかる方向へ移動することで、個体同士が同じ解に留まらず、散らばるように改良する。従来手法の探索アルゴリズムに対して3つの変更をした。(i) 大域探索: 分割探索領域内の最良解を参照;(ii) 局所探索: 分割探索領域内の最良解付近に解候補を生成;(iii) ランダム探索: 選択した個体の分割領域内にランダムで解候補を生成。これらの変更により、従来手法と提案手法の探索性能を比較するため、発見した最適解及び局所解の数を評価尺度として、複数の異なる峰を持つ多峰性関数を用いてシミュレーション上で実験を行う。

本論文の構成はまず、2章で従来手法である BA の概要とそのアルゴリズムについて説明し、3章にて提案手法の詳細と従来手法との変更点について述べる。4章では扱うベンチマーク関数とシミュレーション実験の内容及び、実験結果について記述し、5章で得られた結果に対して考察する。最後に6章にて、本論文の結論を述べる。

2 Bat Algorithm

2.1 全体の概要

Bat Algorithm(BA) 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特性(エコーケーション)を利用して周囲の状況を認知し、大域的な探索が進むにつれて探索速度を徐々に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 は以下の式で定義し、更新される。ラウドネス A は、コウモリが対象物に近づくとき値が減少し、移動距離も比例して短くなる。コウモリの行動は以下3つで構成される。

- 大域探索: 各コウモリは位置 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: 対象物近辺にコウモリを移動させる。
- ランダム探索: 探索領域内にコウモリをランダムで移動させる。

2.2 アルゴリズム

BA で扱う各個体の周波数 f_i 、速度 v_i 、位置 x_i は以下の式で定義される。

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (1)$$

$$v_i^{t+1} = v_i^t + (x_* - x_i^t) * f_i \quad (2)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (3)$$

個体番号を i とし、各個体の周波数 f_i は個体の速度を制限するパラメータであり、 $[0, 1]$ の区間で表される。ここでは $f_{min} = 0$ 、 $f_{max} = 1$ として設定し、 β は0から1の乱数が割り当てられる。局所探索では、全個体の最良解(グローバルベスト) x_* の周辺に新しい解 x_{loc} を生成する。生成式は次の通りである。

$$x_{loc} = x_* + \epsilon A_i^t \quad (4)$$

パラメータ ϵ は $1 \times D$ 次元の配列で $[-1, 1]$ 区間のランダムな値が割り当てられる。ランダム探索では解探索空間にランダムで新たに解を生成する。生成式は以下の通りである。

$$x_{rnd} = x_{lb} + (x_{ub} - x_{lb}) * rand(1, D) \quad (5)$$

解探索空間の上限と下限をそれぞれ x_{ub}, x_{lb} とし, $rand$ は 0 から 1 までの乱数が入る. 以上より各個体の解候補 x_i^{t+1} , x_{loc} , あるいは x_{rnd} の評価値が各個体の最良解 (パーソナルベスト) x_{i*} より良ければ更新され, 同時にラウドネス A とその反射波であるパルスレート r も以下の式に基づいて更新される.

$$A_i^{t+1} = \alpha A_i^t \quad (6)$$

$$r_i^{t+1} = r_i^t [1 - \exp(-\gamma t)] \quad (7)$$

アルゴリズムは以下の Algorithm 1 に記す.

Algorithm 1 Bat Algorithm

Require: 評価関数 $F(x)$ の設定

- 1: 各個体 $x_i (i = 1, 2, \dots, N)$ と速度 v_i の初期化
 - 2: 周波数 f_i の定義 [eq.(1)]
 - 3: パルスレート r_i とラウドネス A_i の初期化
 - 4: **while** ($t < \text{Max Iteration}$) **do**
 - 5: **for** $i=1$ to N **do**
 - 6: 大域探索: 新しい解候補 x_i^{t+1} の生成と速度 v_i の更新 [eqs.(2),(3)]
 - 7: **if** ($rand > r_i$) **then**
 - 8: 局所探索: グローバルベスト x_* 近辺に新しい解候補 x_{loc} を生成 [eq.(4)]
 - 9: **end if**
 - 10: ランダム探索: 解空間に解候補の生成 [eq.(5)]
 - 11: **if** ($rand < A_i$ & $F(x_i), F(x_{loc}), F(x_{rnd}) < F(x_{i*})$) **then**
 - 12: 新しい解の評価と更新
 - 13: パルスレート r_i の増加とラウドネス A_i の減少 [eqs.(6),(7)]
 - 14: **end if**
 - 15: **end for**
 - 16: $t=t+1$
 - 17: **end while**
-

3 Niche Radius-based Bat Algorithm

探索空間の分割方法の一つとして Niche Radius が挙げられる. Niche Radius⁴⁾ は探索空間のスケールと探索する解の数を元に個体の探索範囲を決定することのできる手法である. これにより, 各個体が同じ解に留まることなく分散させ, 従来の BA に以下 3 つの変更点を加えることで, 複数解を探索可能な Niche Radius-based Bat Algorithm(NRBA) を提案する.

3.1 Niche Radius の導入

個体間同士の距離が近い場合に遠ざかる方向へ移動させる機構を持つ Niche Radius を用い, 本研究では同じ解に個体が収束しないよう分散させ, Niche Radius は解空間のスケールと最適解数から算出した距離 (NR) であり, 式 (8),(9) で表される.

$$\lambda = \frac{1}{2} \sqrt{(x_{ub} - x_{lb})^2} \quad (8)$$

$$NR = \frac{\lambda}{\sqrt{q}} \quad (9)$$

x_{ub} , x_{lb} は解空間の上限と下限を示し, D は次元数, q は最適解の数を表す.

3.2 変更点 1: 大域探索

本研究ではこの Niche Radius を使用し, 従来手法の式 (2),(3) を次式のように変更を加えた.

$$v_i^{t+1} = v_i^t + (x_i^t - x_{NR*}) * f_i \quad (10)$$

$$x_i^{t+1} = \begin{cases} x_i^t + v_i^{t+1} & (if \ d_i < NR) \\ x_i^t & (otherwise) \end{cases} \quad (11)$$

個体移動時のイメージ図を Fig. 1 に記す. 個体の持つ NR より個体間距離 d_i が小さい場合において, 式 (10) にて NR 内の最良解 x_{NR*} から離れる方向へ個体 x_i^t が速度 v_i で移動する. この変更により, 個体が同じ探索領域内に留まらず分散化をはかる.

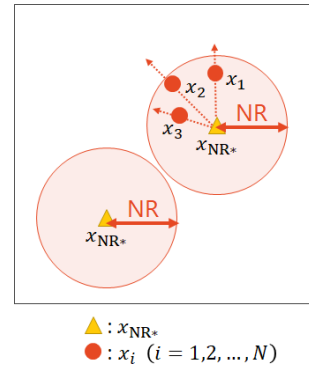


Fig. 1: Generating candidates of new solution

3.3 変更点 2: 局所探索

次に局所探索性能を上げるため, 各個体が持つ Niche Radius 内の最良解 x_{NR*} の周辺に新しい解候補 x_{loc} を生成するよう変更した. 生成式は次の通りである.

$$x_{loc} = x_{NR*} + \epsilon A_i^t \quad (12)$$

ϵ は $1 \times d$ 次元の配列で $[-NR, NR]$ 区間のランダムな値が割り当てられる. この変更により, 個体が局所解への収束を促す.

3.4 変更点 3: ランダム探索

ランダム探索では各個体の NR 内にランダムで解候補を以下の式のように生成する.

$$x_{rnd} = x_i^t + rand(1, D, [-NR, NR]) \quad (13)$$

$[-NR, NR]$ 区間内の $1 \times d$ 次元の配列により現在位置 x_i^t 周辺に解候補を生成する. この変更では各個体が評価値の低い解に留まらないよう, 最適解あるいは局所解近辺へ移動させる. アルゴリズムの疑似コードを Algorithm 2 に記す.

Algorithm 2 Niche Radius-based Bat Algorithm

Require: 評価関数 $F(x)$ の設定

各個体 $x_i (i = 1, 2, \dots, N)$ と速度 v_i の初期化
2: 周波数 f_i の定義 [eq.(1)]
パルスレート r_i とラウドネス A_i の初期化
4: Niche Radius の算出 [eqs.(8), (9)]
while ($t < \text{Max Iteration}$) **do**
6: **for** $i=1$ to N **do**
 if ($d_i < NR$) **then**
8: 大域探索: 新しい解候補 x_i^{t+1} の生成と速度 v_i の更新 [eqs.(10),(11)]
 end if
10: **if** ($\text{rand} > r_i$) **then**
 局所探索: 生成した解候補 x_i^{t+1} 近辺に新しい解 x_{loc} を生成 [eq.(12)]
12: **end if**
 ランダム探索: パーソナルベストの持つ NR 内に解候補の生成 [eq.(13)]
14: **if** ($\text{rand} < A_i$ & $F(x_i), F(x_{loc}), F(x_{rnd}) < F(x_{i*})$) **then**
 新しい解の評価と更新
16: パルスレート r_i の増加とラウドネス A_i の減少 [eqs.(6),(7)]
 end if
18: **end for**
 $t=t+1$
20: **end while**

4 実験

本研究では、最適解と局所解の数が異なる次の4つのパターンの評価関数を用意し、従来手法と比較することで提案手法の探索性能を検証する。一つの最適解に対して複数の局所解を持つ; 複数の最適解に対して同じ数の局所解を持つ; 一つの最適解の数に対して一つの局所解を持つ; 最適解のみ複数持つ。これらの多峰性関数を用いて実験を行う。

4.1 問題設定

実験で使用するベンチマーク関数の4つを Fig. 2 に示す。Fig. a から Fig. d は各評価関数の概形を表す。Fig. e から Fig. h の各グラフの平面は2変数 x_1, x_2 空間を表しており、カラーバーは評価値を色濃度で表現したものとなる。本実験では最小化問題として、色濃度の濃い（評価値の低い）部分となる最適解と局所解の探索を目的とした。Table 1 は各関数の探索領域、評価値、解の個数を示している。

Griewank Function

多峰性関数の一つである、16 個の局所解と1つの最適解を持つ Griewank Function を用いる (Fig. a 参照)。評価関数の式は以下の通りである。

$$F_1(x_i) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (14)$$

解空間の探索領域は $x_i \in [-10, 10]$ である ($i = 1, 2$)。最適解の座標は $x_* = [0, 0]$ で、その評価

値は $F(x_*) = 0$ である。局所解の座標は $\pm x \approx [6.2800, 8.8769], [3.1400, 4.4385], [0, 8.8769], [6.2800, 0], [9.4200, 4.4385]$ となる。

Six-Hump Camel Function

最適解と局所解を2つ持つ関数で以下の式で表される。

$$F_2(x_1, x_2) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \quad (15)$$

探索領域は $x_1 \in [-2, 2], x_2 \in [-1, 1]$ である。最適解の座標は $x_* = [0.0898, -0.7126], [-0.0898, 0.7126]$ であり、その評価値は $F_3(x_*) = -1.0316$ である。また局所解は $\pm x \approx [1.704, -0.7965]$ となる。

Michalewicz Function

関数式を以下に示す。

$$F_3(x_i) = - \sum_{i=1}^D \sin(x_i) \sin^{2m}\left(\frac{ix_i^2}{\pi}\right) \quad (16)$$

最適解 $x_* = [2.20, 1.57]$ の評価値 $F_3(x_*) = -1.8013$ であり、局所解は $x \approx [2.203, 2.7115]$ である。探索領域は $x_i \in [0, 4]$ である ($i = 1, 2$)。

Himmelblau Function

評価関数式は次の通りとなる。

$$F_4(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \quad (17)$$

評価関数の最適解は4つあり、それぞれ $x_* = [3, 2], [-2.805118, 3.283186], [-3.779310, -3.283186], [3.584458, -1.848126]$ に位置し、その評価値は $F_4(x_*) = 0$ である。探索領域は $x_i \in [-5, 5]$ となる ($i = 1, 2$)。

4.2 評価尺度

本実験において、Congress on Evolutionary Computation(CEC2013)⁵⁾ のコンペティションで用いられた評価尺度である Peak Ratio(PR)⁶⁾ により評価する。評価式は以下のように設定した。

$$PR = \frac{\sum_{run=1}^{MR} FPs}{TP * MR} \quad (18)$$

Max Run(MR) は実験回数を表し、Found Peaks(FPs) は発見した解の数を、Total Peak(TP) は全ての最適解及び局所解数を表す。また最適解及び局所解の位置座標と最近傍個体とのユークリッド距離が0.1未満であれば、その解を発見したと定義する。

4.3 パラメータの設定

個体数 $N = 50$ とし、各個体のパラメータ $A_i^0 = 1, r_i^0 \in [0, 1], f_{max} = 1, f_{min} = 0, \alpha = \gamma = 0.9$ と設定した。また Table 1 より、探索領域の上限 x_{ub} と下限 x_{lb} 、各評価関数の解の総数 q として使用した。また次元数 $D = 2$ 、世代数を10000、実験回数 $MR = 30$ とした。

Table 1: Measurement of Benchmark Functions

Function	F_1	F_2	F_3	F_4
Search Domain	$-10 \leq x_i \leq 10$	$-2 \leq x_1 \leq 2, -1 \leq x_2 \leq 1$	$0 \leq x_i \leq 4$	$-5 \leq x_i \leq 5$
Fitness of global optimum	0	-1.0316	-1.8013	0
Num of global optima	1	2	1	4
Num of local optima	16	2	1	0

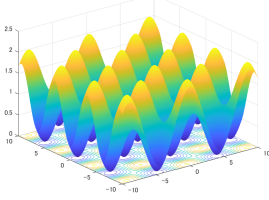


Fig. a:

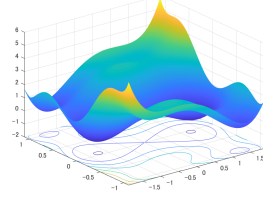


Fig. b:

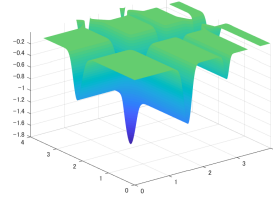


Fig. c:

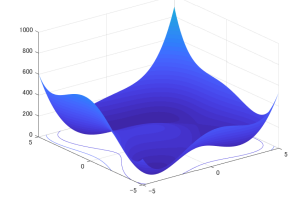


Fig. d:

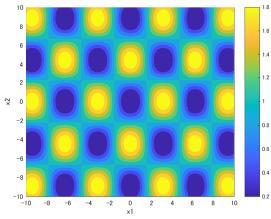


Fig. e:

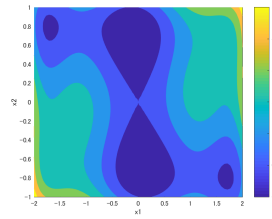


Fig. f:

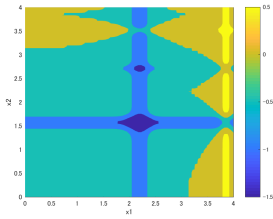


Fig. g:

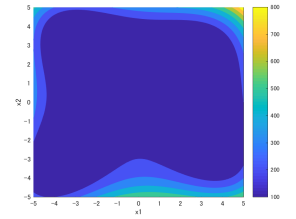


Fig. h:

Fig. 2: Fitness landscape and contour of functions

4.4 実験結果

各評価関数について従来のBAと提案手法のNRBAにおける発見した解の数をTable 2に示す。表中のMean(平均値)とSD(標準偏差)は実験回数30回分での最終世代における発見した最適解及び局所解数の実験結果である。各手法の最終世代での個体の分布をFig. 3, 4で表す。またグラフ中の赤い丸は個体の分布を示す。従来手法であるBAは全個体の最良解へ向かって進んでしまうため、全ての評価関数において、最適解あるいは評価値の高い局所解に収束した。しかしFig. 3(d)については局所解は存在しないが、最終世代では一つの最適解へ収束する結果となった。一方で提案したNRBAではFig. 4から全評価関数において、全ての解に個体が到達しているように分布しているが、Table 2から実際に最適解や局所解の位置まで到達していないケースが多く見られた。また最適解や局所解に到達していない個体については用いた評価関数によって分布に偏りがあった。

5 考察

全ての評価関数において、従来のBAより提案のNRBAのほうが探索した解の数および発見率が高かったことから最適解への収束を防ぎ、複数の局所解へ分散させることができた。また提案手法の探索性能の有効性を確かめるため、評価尺度による結果と最終世代の解の分布という観点から分析を行った。

5.1 BA vs NRBA

5.1.1 解の発見数

Table 2から従来手法のBAは全ての評価関数において最適解へ収束する傾向が見られ、特に F_2 関数では2

つの最適解を探索することができたが局所解へ収束することはなかった。これは従来手法のアルゴリズムの解候補の生成において、全個体の最良解へ向かって探索をしていることが原因であると考えられる。提案手法のNRBAは全ての評価関数に対して、従来手法よりも発見した解の数が多く、最適解と複数の局所解を探索することができた。評価関数の中でも F_2 、次いで F_4 のPRの値が高いことから最適解のみ存在する場合と、最適解と局所解の評価値の差が小さい場合において提案手法の探索が有効に働いたと考えられる。また局所解を多く含む F_1 関数や局所解の範囲が非常に狭い F_3 関数では探索性能が落ちたことから今後の課題として手法の改良をする必要がある。

5.1.2 最終世代における解の分布

Fig. 3から従来のBAは、最適解への収束が非常に強かった。しかし、Fig. 3(b)では2つの最適解に収束していたものの、Fig. 3(d)では複数の最適解へ収束することなく一つの最適解へ収束した。これは、 F_4 関数の方が探索領域が広く、評価値の数値の範囲も広いことから各個体の持つ評価値にバラつきが出やすく、一つの最適解へ収束したと考えられる。提案手法のNRBAは、 F_1 関数では色濃度の濃い領域に各個体を分散させることができたが、 F_2, F_3, F_4 関数では最適解や局所解でない場所に分布する個体も多く見られた。このことから従来手法から変更した、Niche Radius内の最良解より個体が遠ざかる機構が働いたと考えられる。以上より探索領域を分割し、各個体の探索領域内での大域探索性能を保ちながらも徐々に局所解へ収束させることが、解の発見率の増加に繋がったと考えられる。

Table 2: Found Peaks and Peak Ratio

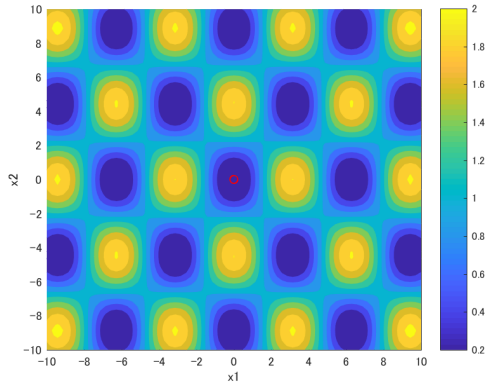
Function	BA			NRBA		
	Mean	SD	PR	Mean	SD	PR
F_1	1.0	0	5.88 %	11.767	1.667	69.22 %
F_2	1.967	0.180	49.17 %	3.967	0.180	99.17 %
F_3	1.0	0	50.00 %	1.4	0.490	70.00 %
F_4	0.967	0.547	24.17 %	3.433	0.496	85.83 %

6 おわりに

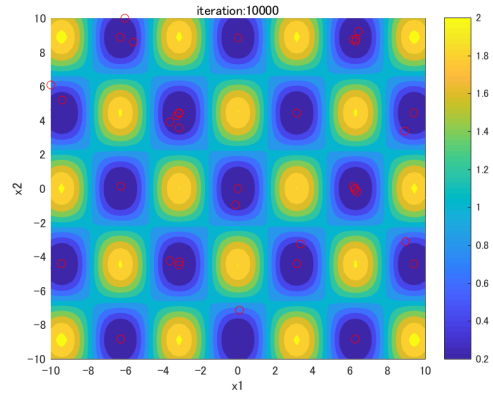
本論文では従来手法のBAについて、全個体の最良解を参照して一点へ集まってしまうことによる最適解への収束を防ぎ、同時に複数の局所解を探索可能なNRBAを提案した。提案手法では次の3点を変更した。(i) 大域探索: 分割探索領域内の最良解を参照;(ii) 局所探索: 分割探索領域内の最良解付近に解候補を生成;(iii) ランダム探索: 選択した個体の分割領域内にランダムで解候補を生成。提案手法の探索性能の検証及び従来手法と比較するため、複数の最適解と局所解を持つベンチマーク関数を用いてシミュレーション上で実験した。結果、全てのベンチマーク関数において、従来手法より提案手法のほうが解の発見数及び発見率が高かった。このことから(i)の修正により最適解への収束を避け、(ii)より分割した探索領域内の探索性能を向上させ、(iii)より各最適解と局所解への収束を促すことにより、個体の分散化が有効に働いたといえる。今後の課題としては全ての解を探索可能な性能の向上と、実問題への適用を考慮した動的環境においても探索性能を発揮できるアルゴリズムを構築する。

参考文献

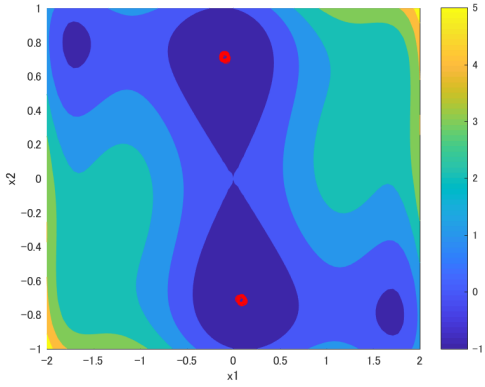
- 1) Eberhart, R. C., and Kennedy, J. : “A New Optimizer Using Particle Swarm Theory”, Proc. Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan), IEEE Service center, Piscataway, NJ, 39-43 (1995)
- 2) Yang, X. S. ” Firefly Algorithms for Multimodal Optimization”, in: Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences, Vol. 5792, pp. 169-178 (2009)
- 3) Yang, X. S. “A Metaheuristic Bat-Inspired Algorithm”, in: *Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)* (Eds J.R. Gonzalez et al.), *Studies in Computational Intelligence*, Springer Berlin, 284, Springer, 65-74 (2010).
- 4) D.Beasley, D.R. Bull, and R.R. Martin, ”A sequential niche technique for multimodal function optimization,” *Evolutionary Computation*, vol. 1, no.2, pp. 101-125, 1993.
- 5) X. Li, A. Engelbrecht, and M. G. Epitropakis, ”Benchmark Functions for CEC’2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization”, *Evol. Comput. Mach. Learn. Group*, RMIT University, Melbourne, VIC, Australia, Tech. Rep., 2013.
- 6) R. Thomsen, ”Multimodal Optimization using Crowding-based Differential Evolution,” *In the IEEE Congress on Evolutionary Computation*, 2004. CEC2004, vol.2, pp. 1382-1389, 19-23 June, 2004.



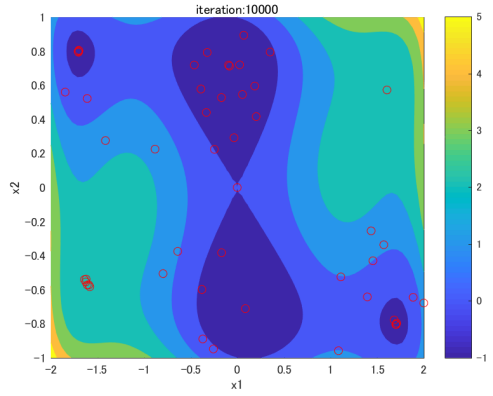
(a) F_1 : Griewank



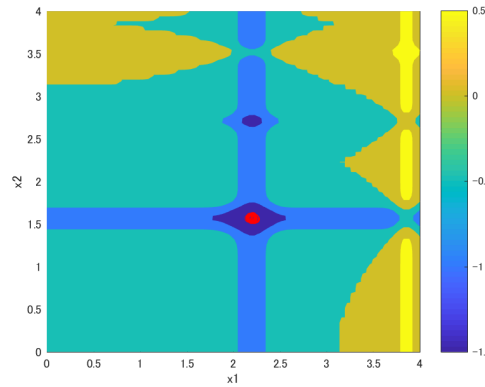
(a) F_1 : Griewank



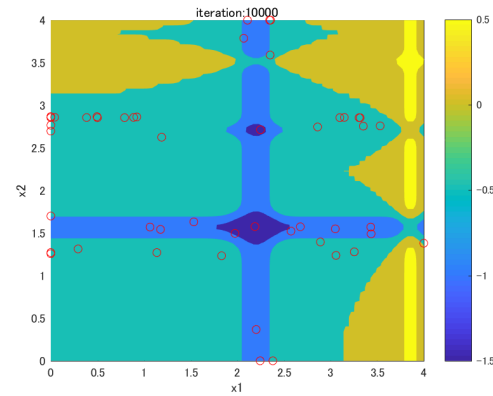
(b) F_2 : Six-Hump Camel



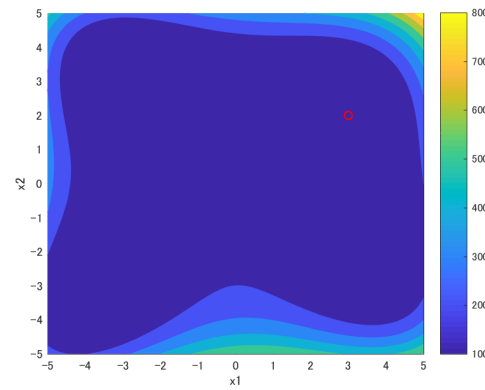
(b) F_2 : Six-Hump Camel



(c) F_3 : Michalewicz

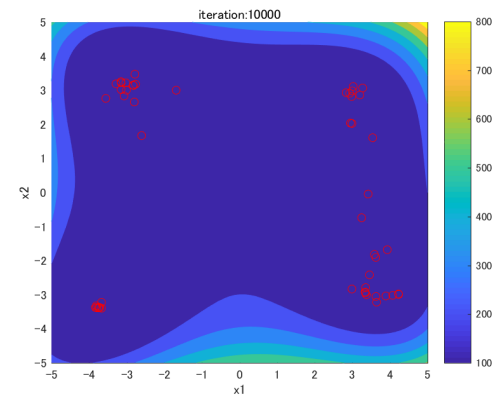


(c) F_3 : Michalewicz



(d) F_4 : Himmelblau

Fig. 3: BA



(d) F_4 : Himmelblau

Fig. 4: NRBA