Niche Radius Adaptation in Bat Algorithm for Locating Multiple Optima in Multimodal Functions

Takuya Iwase
The Univ. of Electro-Communications
Tokyo, Japan
tanu_iwa@cas.lab.uec.ac.jp

Ryo Takano
The Univ. of Electro-Communications
Tokyo, Japan
takano@cas.lab.uec.ac.jp

Fumito Uwano
The Univ. of Electro-Communications
Tokyo, Japan
uwano@cas.lab.uec.ac.jp

Hiroyuki Sato
The Univ. of Electro-Communications
Tokyo, Japan
h.sato@uec.ac.jp

Keiki Takadama
The Univ. of Electro-Communications
Tokyo, Japan
keiki@hc.uec.ac.jp

***Abstract*—Evolutionary algorithms (EAs) are often used for multimodal optimization which is modeled as real-world problem. However, most EAs still not enough to find multiple local optima because of the concept of the solution movement between nearest neighbor solutions. This paper proposes the niche radius-based bat algorithm (NRBA), which is designed to find multiple local optima in multimodal optimization. We focus on bat algorithm (BA) which deals with the trade-off between exploration and exploitation in the evolutionary process and extend it with niche radius which can control and modify the search space of solutions to avoid overlapping the found optima. In detail, the proposed BA consists of three search phases: (i) the movement from neighbors for avoiding overlapping the same found optima; (ii) the exploitation for searching nearby the best solution of its domain with Niche Radius; (iii) the exploration for searching randomly in all domain of the radius. In order to evaluate the performance of NRBA, this paper employs some test-bed multimodal functions and compare NRBA with BA and NSBA. The experimental results suggest that NRBA is able to provide the better search performance than BA and NSBA to find multiple global optima in most of best-bed multimodal functions.**

***Index Terms*—Bat Algorithm, Multimodal Optimization, Swarm Intelligence**

I. Introduction

There are many studies on evolutionary algorithms (EAs) solving the real-world optimization problems which are mostly multimodal and complex optimization problems. These problems has not only single global optimum but also many local optima, hence EAs are required to find the both multiple optima which might be changed their own location as the environment changes.

To tackle multimodal optimization problems, numerous techniques which are commonly known as *niching methods* have proposed in [1]- [4], [6] [7]. Thomsen proposed the DE [5] extends with a crowding scheme (CDE) [6] to replace the high-quality solution by the most similar candidate solutions. Li proposed the DE with Speciation (SDE) [7] to keep a solution away from the nearest neighbor solution when the distance of both these solutions is less than the threshold. However, these niching methods are still not enough to find multiple local optima because both of them do not consider searching globally though they consider the solution movement according to the euclidean distance between the nearest neighbor solutions. For solving this problem, this paper focuses on Bat Algorithm (BA) that has the characteristic of echolocation which can predict the distance between bats and the target (*i.e., object/food source*). This algorithm enables bats to estimate the distance between their location and the target even in the situations such as the target surrounded by obstacles and the absence of light. Bats can adjust their velocity which is controlled by their loudness, pulse emission rate and frequency, toward the target. While the iteration search step continues until bats reaches the target in the evolutionary process, bats will stop searching the target within its perceptible distance. This research employs BA which copes with exploitation and exploration search, extending with Niche Radius for multimodal optimization. Niche Radius is the threshold distance calculated by the fitness landscape and the number of its peaks.

This paper is organized as follows. After this section, the mechanism of BA and the proposed algorithm NRBA are explained in Sections 2 and 3. Section 4 describes the multimodal functions as the test-bed problem in the experiment. Section 5 shows the results while Section 6 discusses them. Finally, this paper concludes in Section 7.

II. Bat Algorithm

As mentioned in Section 1, BA is a metaheuristic algorithm based on the bat behavior according to its loudness and pulse emission rate of the reflect wave, which control the balance between a local and global search. When a bat finds the better solution than the current one, the loudness A_i and the pulse rate r_i gradually decreases and increases, respectively. To find better solution, the bat has the following three solution search phases: (i) the bat i flies to the target (*i.e.*, the bat which finds the best solution) with the velocity controlled by frequency f_i; (ii) the bat i flies around the target as a local search; and (iii) the bat i flies randomly in search space as a global search. Let us explain these search phases. First, in the search phase (i), all bats change their locations x_i with their velocities v_i toward the global best solution. For this calculation, the frequency f_i, velocity v_i, and location x_i, of the bat i are calculated as follows:

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

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

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{3}$$

In detail, the new solution x_i is updated by adding the new the velocity v_i which is derived from the previous velocity v_i^t, the distance between the global best location and the previous location $(x_* - x_i^t)$, and frequency f_i which range is [f_{min}, f_{max}] where $f_{min} = 0$ and $f_{max} = 1$. β is uniform random value from 0 to 1. Next, in the solution search phase (ii), the new solution x_{loc} is generated around the global best solution as follows:

$$x_{loc} = x_* + \epsilon A^t \ , \tag{4}$$

where ϵ is uniform random value between $[0, \ 1]$. In Eq.(6), A^t is the averaged loudness of all bats. Finally, in the search phase (iii), x_{rnd} is generated randomly in search space as follows:

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

where x_{ub} and x_{lb} describe the upper and lower bounds of the search space, and $rand(1, D)$ is the D dimensional uniform random value between $[0, \ 1]$.

When a bat finds the better solution than the current one, the loudness A_i and pulse emission rate r_i are updated as follows:

$$A_i^{t+1} = \alpha A_i^t \tag{6}$$

$$r_i^{t+1} = r_i^0[1 - exp(-\gamma t)] \tag{7}$$

Note that the loudness A_i^0 is initialized as $A_i^0 = 1$ and the pulse rate is initialized as a uniform random value r^0 between $[0, \ 1]$ or a number closed around zero. The parameters α and γ are the symbolized damping coefficient. The pseudo code of BA is given in the Algorithm 1 and its brief summary is described below.

- STEP1: Population initialization of bats (line 1 to 3)
  The population of bats $x_i(i = 1, 2, ..., N)$, the loudness A_i^0, the pulse rate r_i^0 are initialized as the initial values. The frequency f_i is initialized by Eq.(1).
- STEP2: New solution updates (line 6)
  The new solutions x_i is calculated by Eqs. (2)(3).
- STEP3: New solution generation around global best solution x_* (line 7 to 9)
  A new solution x_{loc} is generated around x_* by Eq. (4) when the pulse emission rate r_i is lower than a random value.
- STEP4: Random new solution generation (line 10)
  A new solution x_{rnd} is generated randomly by Eq. (5).
- STEP5: Solutions update(line 11 to 14)
  When $rand < A_i$, the best solution is selected from x_i, x_{loc}, and x_{rnd} by Eqs.(6),(7)
- STEP6: Return to STEP2

Algorithm 1 Bat Algorithm

Require: Objective Function $F(x)$
1: Initialize Population $x_i(i = 1, 2, ..., N)$ and v_i
2: Define frequency f_i at location x_i [eq.(1)]
3: Initialize pulse rates r_i, and loudness A_i
4: **while** ($t <$ Max number of iterations) **do**
5:   **for** i=1 to N **do**
6:     Generate a new solution x_i and velocity v_i [eqs.(2) to (3)]
7:     **if** ($rand > r_i$) **then**
8:       Generate a new solution x_{loc} around a global best solution x_i [eq.(4)]
9:     **end if**
10:     Generate a new solution x_{rnd} randomly
11:     **if** ($rand < A_i \& \min(F(x_i), F(x_{loc}), F(x_{rnd}) < F(x_{i*})$) **then**
12:       Accept the new solution, and update pulse rate r_i & loudness A_i [eqs. (6)(7)]
13:     **end if**
14:     Evaluate all bats and select a best solution x_* in the current solutions
15:   **end for**
16:   t=t+1
17: **end while**

III. Proposed Algorithm

A. Niche Radius

Niche Radius (NR) [9] [10] is one of niching techniques to determine the radius which is calculated by the number of local optima and the scale of the fitness landscape as follows:

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

$$NR = \frac{\lambda}{\sqrt[D]{q}}, \tag{9}$$

where the lower and upper bound values are x_{ub} and x_{lb} of the D-th dimensional search space, and the number of local optima is q. Each domain of the radius as NR is wrapping the local optimum to avoid converging same optima.

B. Niche Radius-based Bat Algorithm

As stated earlier, proposed algorithm is extended BA with the concept of Niche Radius which is able to locate many solutions without converging the found global best solutions within the domain of niche radius. This process is given by

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

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

where x_{NR*} is the best solution in the domain of niche radius and d_i^t indicates the euclidean distance between the nearest neighbor solutions. Fig. 1 provides an example to illustrate the solution movement to keep the solution x_i from the best solution x_{NR*} in the domain of the radius, where the red circle and the yellow star indicates solution x_i and the best solution x_{NR*}. This figure shows the situation that the solutions x_1, x_2 and x_3 are located in the area of x_{NR*} with the radius. In this case, these solutions are moved out from the area of the best solution x_{NR*}.

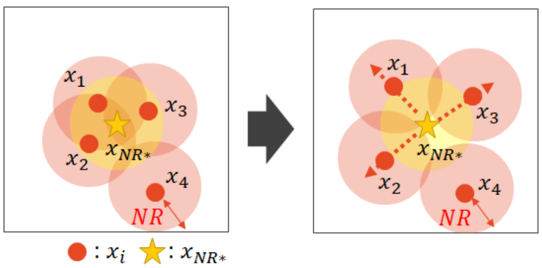


Fig. 1. Example of Solution Movement based on Niche Radius

In the local search phase, the new solution x_{loc} is generated around the best solution x_{NR*} in the domain of the radius as follows:

$$x_{loc} = x_{NR*} + \epsilon A_i^t \tag{12}$$

where A_i and ϵ is the same values as BA. In the global search phase, the new solution x_{rnd} is generated randomly in all domains of the range between $[-NR, NR]$ as follows:

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

C. Algorithm Description

The pseudo code of NRBA is given in Algorithm 2 and its brief summary is described below.

Algorithm 2 Niche Radius-based Bat Algorithm

Require: Objective Function $F(x)$
Initialize Population $x_i (i = 1, 2, ..., N)$ and v_i
2: Define frequency f_i at location x_i [eq.(1)]
Initialize pulse rates r_i, and loudness A_i
4: **while** ($t <$ Max number of iterations) **do**
for i=1 to N **do**
6: **if** $d_i < NR$ & $x_i \neq x_{NR*}$ **then**
Generate a new solution x_i and velocity v_i [eqs.(10) to (11)]
8: **end if**
if $(rand > r_i)$ **then**
10: Generate a new solution x_{loc} around a global best solution x_i [eq.(12)]
end if
12: Generate a new solution x_{rnd} randomly [eq.(13)]
if $(rand < A_i \& \min(F(x_i), F(x_{loc}), F(x_{rnd}) < F(x_{i*}))$ **then**
14: Accept the new solution, and update pulse rate r_i & loudness A_i [eqs. (6)(7)]
end if
16: Evaluate all bats and select a best solution x_* in the current solutions
end for
18: t=t+1
end while

- STEP1: Population initialization of bats (line 1 to 3)
  The population of bats $x_i (i = 1, 2, ..., N)$, the loudness A_i^0, the pulse rate r_i^0 are initialized as the initial values. The frequency f_i is initialized by Eq.(1).
- STEP2: New solution updates (line 6 to 8)
  The new solutions x_i is calculated by Eqs. (2)(3).
- STEP3: New solution generation around global best solution x_* (line 9 to 11)
  A new solution x_{loc} is generated around x_* by Eq. (4) when the pulse emission rate r_i is lower than a random value.
- STEP4: Random new solution generation (line 12)
  A new solution x_{rnd} is generated randomly by Eq. (5).
- STEP5: Solutions update(line 13 to 15)
  When $rand < A_i$, the best solution is selected from x_i, x_{loc}, and x_{rnd} by Eqs.(6),(7)
- STEP6: Return to STEP2

IV. Experimental Setup

To measure the number of local optima and the convergence speed, we compare with the performance of all algorithms. In this section, four multimodal functions which are considered for maximization, are employed in *Congress on Evolutionary Computation (CEC) 2013* competition [12].

A. Benchmark Test Functions

In order to verify the effectiveness of our proposed algorithm compared with conventional BA and another *Niching method* which is Novelty Search-based BA (NSBA) [11], four benchmark functions that are popular in multimodal optimization are employed in this experiments. The detail of these functions which are the search space, the fitness value of known global optima and the number of known global and local optima, are summarized in Table I.

F_1: Himmelblau Function

$$F_1(x_1,x_2) = 200 - (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \tag{14}$$

The fitness value of the global optima is $F(x_*) = 200$. This function has 4 global optima in the range between $x_1, x_2 \in [-6,\ 6]$.

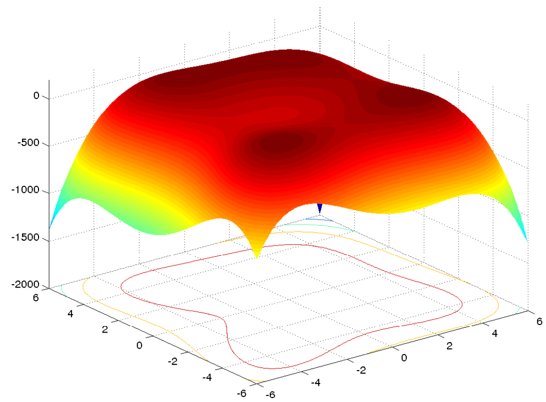


Fig. 2. Himmelblau

F_2: Shubert Function

This function is described as follows as shown in Fig. 3.

$$F_2(x) = -\prod_{i=1}^{D}\sum_{j=1}^{5} j\cos[(j+1)x_i + j], \tag{15}$$

where D is the number of dimension and the fitness value of the global optima is $F(x_*) = 187.731$. This function has $D \cdot 3^D$ global optima and numerous local optima in the range of search space between $x_i \in [-10, 10]^D$ with $i = 1, 2, ..., D$. Fig. 3 shows an example of the Shubert 2D function which has 18 global optima.

F_3: Vincent Function

This function is described as follows as shown in Fig. 4.

$$F_3(x) = \frac{1}{D}\sum_{i=1}^{D} \sin(10log(x_i)) \tag{16}$$

where D is the number of the dimension and the fitness value of the global optima is $F(x_*) = 1.0$. This function has 6^D global optima in the range of search space between $x_i \in [0.25, 10]^D$ with $i = 1, 2, ..., D$. This figure provides the 2D function in case of 36 global optima with $D = 2$.

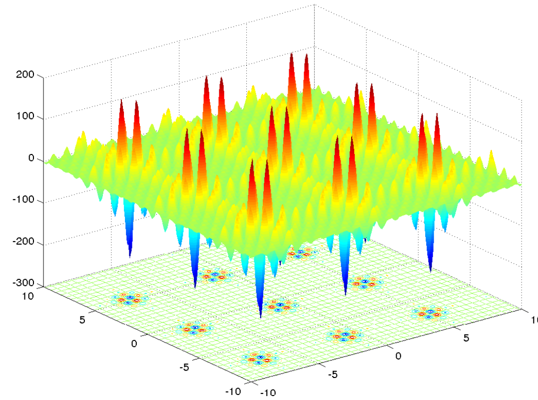


Fig. 3. Shubert

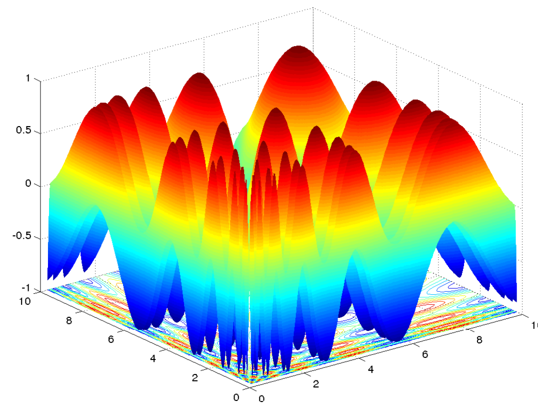


Fig. 4. Vincent

F_4: Modified Rastrigin Function

This function is described as follows as shown in Fig. IV-A.

$$F_4(x) = -\sum_{j=1}^{D}(10 + 9\cos(2\pi k_i x_i)) \tag{17}$$

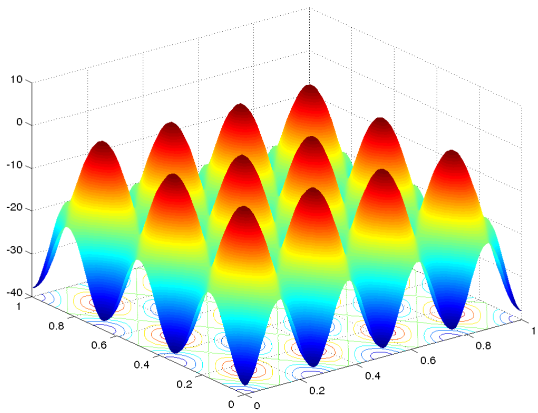
where D is the number of dimension and the fitness value of the global optima is $f(x_*) = -2.0$. In the case of $D = 2$, this function has $\prod_{i=1}^{D} k_i$ *(12 global optima)* with following setting: $k_1 = k_3 = 1, k_2 = 4, k_= 2$ global optima in the range of search space between $x_i \in [0, 1]^D$ with $i = 1, 2, ..., D$.

B. *Performance Measurements*

To determine how many global optima the algorithm found, accuracy distance ρ is used to calculate the distance between nearest neighbor individuals as shown in Algorithm 3. At the first step, the individuals at the final iteration are sorted by the fitness value in descending order. The next step, if the distance between the fitness of the global optimum ph and the nearest neighbor individual p is less than ε, similar nearest neighbor individuals are compared within accuracy distance ρ. The better fitness of individual is added to S.

TABLE I
MEASUREMENT OF BENCHMARK TEST FUNCTIONS

Function	F_1	F_2	F_3	F_4
Search Space	$x_1, x_2 \in [-6,\ 6]$	$x_i \in [-10,\ 10]$	$x_i \in [0.25,\ 10]$	$x_i \in [0,\ 1]$
Fitness Value	200.0	186.731	1.0	-2.0
Number of global optima	4	18	36	12
Number of local optima	0	many	0	0
accuracy distance ρ	0.01	0.5	0.5	0.01



Algorithm 3 Calculate how many global optima the algorithm found

---

Require: L_{sorted} a list of individuals in descending order of the fitness value

$S = \emptyset$

while ($not\ reaching\ the\ end\ of\ L_{sorted}$) **do**

Get best unprocessed $p \in L_{sorted}$;

$found \leftarrow$ FALSE;

if $d(ph, fit(p)) \leq \varepsilon$ **then**

for $all\ s \in S$ **do**

if $d(s, p) \leq \rho$ **then**

$found \leftarrow$ TRUE;

break;

end if

end for

if not found **then**

let $s \leftarrow S \cup \{p\}$

end if

end if

end while

---

1) Peak Ratio: This experiment employs Peak Ratio(PR) [6] as the evaluation criterion in the CEC (IEEE Congress on Evolutionary Computation) 2013 competition [12]. The PR value measures the ratio of the found global and local optima in the total number of true peaks and it is calculated as follows:

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

where MR indicates the maximum run, FPs indicates indicates the number of peaks found by the optimization algorithm. TP indicates the number of all known peaks of the function. We define that the peak is found when the Euclid distance between the all known peaks and the nearest solution calculated by the optimization algorithm is less than the thresholds $\varepsilon = \{1.0, 1.0E-1, 1.0E-2\}$.

2) Peak Accuracy: To measure how far solutions are close to the peaks, we employ Peak Accuracy (PA) [6] calculated as follows:

$$PA = \sum_{j=1}^{TP} |F(s_j) - F(x_{NN_j})|, \tag{19}$$

where s_j and x_{NN_j} denote the each known peak and the nearest neighbor solution. As the closest distance between both of them is short, the value of PA is close to 0.

C. Experimental Parameters

All experiments employ the parameters as follows: frequency $f_{max} = 1, f_{min} = 0$, loudness $A^0 = 1$, parse rate $r^0 \in [0,\ 1]$ with $\alpha = \gamma = 0.9$. The population size $N = 100$. This experiments are simulated 30 runs with different random seeds and 10000 evaluations as the termination condition for each run.

V. RESULTS AND ANALYSIS

To test the effectiveness of NRBA mechanism, this section investigates the peak ratio (PR) and the peak accuracy (PA) of each benchmark test function. Table II, III, and IV show the results that the PR and the PA values of two algorithms based on the settings of averaged over 30 individual runs at the final iteration. Fig. 5 and 7 show that the solutions relocating and exploiting at the final iteration for all functions.

A. Peak Ratio

From Table II in the case of $\varepsilon = 1.0$, NRBA has outperformed than the other algorithms for almost all test functions. For F_1, the PR value of NRBA was the same as the other algorithms. It can see from this table, all algorithms are able to locate all global optima for all runs (as shown in Fig. 5(a) 6(a) and 7(a). The case of $\varepsilon = 1.0E-1$ from Table III, the PR values of NRBA were also higher than BA and NSBA. However, the values of both algorithms went down greatly from Table II to III because F_2 is extremely sharp and complex landscape compared with the other functions. Table IV shows that NRBA has outperformed than BA and NSBA in F_3 and F_4 in the case of $\varepsilon = 1.0E-2$. The performance of NRBA gradually decreased from Table II to IV. As it can be seen in this table, NRBA was eventually worse than BA and NSBA in F_1 and F_2.

TABLE II

PEAK RATIO AND PEAK ACCURACY OF BA AND NRBA (AVERAGED OVER 30 RUNS)

accuracy level: $\varepsilon = 1.0$

	BA		NSBA		NRBA	
Function	PR (Mean and SD)	PA (Mean and SD)	PR (Mean and SD)	PA (Mean and SD)	PR (Mean and SD)	PA (Mean and SD)
F_1	**1** $\pm$ 0	**0.0060** $\pm$ 0.0028	**1** $\pm$ 0	0.0071 $\pm$ 0.0031	**1** $\pm$ 0	0.0326 $\pm$ 0.017
F_2	0.5870 $\pm$ 0.0991	**3.9272** $\pm$ 1.199	**0.8148** $\pm$ 0.0763	5.0605 $\pm$ 1.1713	0.7111 $\pm$ 0.1077	4.765 $\pm$ 1.3987
F_3	0.4407 $\pm$ 0.0839	**0.0044** $\pm$ 0.0021	0.5963 $\pm$ 0.0530	0.0066 $\pm$ 0.0019	**0.6685** $\pm$ 0.0699	0.0080 $\pm$ 0.0027
F_4	0.9833 $\pm$ 0.0333	**0.0287** $\pm$ 0.0102	0.0199 $\pm$ 0.0022	2.4686 $\pm$ 0.4510	**1** $\pm$ 0	0.029 $\pm$ 0.0097

TABLE III

PEAK RATIO AND PEAK ACCURACY OF BA AND NRBA (AVERAGED OVER 30 RUNS)

accuracy level: $\varepsilon = 1.0E-1$

	BA		NSBA		NRBA	
Function	PR (Mean and St. D.)	PA (Mean and St. D.)	PR (Mean and St. D.)	PA (Mean and St. D.)	PR (Mean and St. D.)	PA (Mean and St. D.)
F_1	**1** $\pm$ 0	**0.0060** $\pm$ 0.0028	**1** $\pm$ 0	0.0071 $\pm$ 0.0031	**1** $\pm$ 0	0.0326 $\pm$ 0.0170
F_2	0.1148 $\pm$ 0.0640	**0.0808** $\pm$ 0.0629	0.2 $\pm$ 0.0985	0.1664 $\pm$ 0.0955	**0.1185** $\pm$ 0.0821	0.1135 $\pm$ 0.0922
F_3	0.4407 $\pm$ 0.0839	**0.0044** $\pm$ 0.0021	0.5963 $\pm$ 0.0530	0.0066 $\pm$ 0.0016	**0.6685** $\pm$ 0.0699	0.0081 $\pm$ 0.0027
F_4	0.9833 $\pm$ 0.0333	**0.0287** $\pm$ 0.0102	0.0046 $\pm$ 0	0.0555 $\pm$ 0.0140	**1** $\pm$ 0	0.029 $\pm$ 0.0097

TABLE IV

PEAK RATIO AND PEAK ACCURACY OF BA AND NRBA (AVERAGED OVER 30 RUNS)

$\varepsilon = 1.0E-2$

	BA		NSBA		NRBA	
Function	PR (Mean and St. D.)	PA (Mean and St. D.)	PR (Mean and St. D.)	PA (Mean and St. D.)	PR (Mean and St. D.)	PA (Mean and St. D.)
F_1	**1** $\pm$ 0	**0.0060** $\pm$ 0.0028	**1** $\pm$ 0	0.0071 $\pm$ 0.0031	0.6917 $\pm$ 0.3006	0.0127 $\pm$ 0.0076
F_2	**0.0315** $\pm$ 0.04223	0.0029 $\pm$ 0.0040	0.0296 $\pm$ 0.0379	0.0026 $\pm$ 0.0044	0.0167 $\pm$ 0.0255	**0.0020** $\pm$ 0.0034
F_3	0.4407 $\pm$ 0.0839	**0.0044** $\pm$ 0.0021	0.5963 $\pm$ 0.0530	0.0066 $\pm$ 0.0016	**0.6685** $\pm$ 0.0699	0.0078 $\pm$ 0.0031
F_4	0.9444 $\pm$ 0.0583	**0.0215** $\pm$ 0.0069	NaN	NaN	**0.9806** $\pm$ 0.0352	0.0234 $\pm$ 0.0074

B. Peak Accuracy

NRBA is worse than BA and NSBA for all functions from all Table II, III and IV except for F_2 in $\varepsilon = 1.0E-2$. As it can be seen the results, the exploit phase of BA has worked well more than NRBA regardless of the difference value of threshold ε from Table II, III and IV. Furthermore, the random search phase of BA is assumed to cope with the exploitation and the exploration. However, the density distribution of the solutions is unbalanced (especially in Fig. 5(a) and 5(c)) due to promoting all solutions to exploit the global best solution. In contrast, NRBA is able to spread the solutions to multiple optima even if some global optima are located the same domain of Niche Radius.

VI. CONCLUSION

This paper proposes BA extending with Niche Radius which enables to avoid overlapping the solutions into the same peak and locate multiple global optima in several multimodal functions which are different from landscape and the number of peaks. For solving multimodal optimization, we improved the three search phases of BA: (i) the movement from neighbors for avoiding overlapping the same found optima; (ii) the exploitation for searching nearby the best solution of its domain with Niche Radius; (iii) the exploration for searching randomly in all domain of the radius. To evaluate the performance of NRBA, this algorithm were compared with BA and NSBA. The results show that NRBA performed better than the other algorithms because the spatial distribution mechanism in NRBA copes with locating many multiple global optima. In contrast, BA and NSBA are still better than NRBA regarding the peak accuracy which indicates how far the peaks are close to the solutions, due to the distribution mechanism of NRBA.

In the future, we will improve the algorithm performances: the exploration for searching all global optima completely, and the exploitation for locating global optima rapidly.

REFERENCES

[1] S. W. Mahfoud, A comparison of parallel and sequential niching methods, in *Proc. 6th Int. Conf. Genet. Algorithms*, pp.136-143, 1995.

[2] G. Singh and K. Deb, Comparison of Multi-modal Optimization Algorithms Based on Evolutionary algorithms., *In Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation.*, 1305-1312, 2006.

[3] D. E. Goldberg and J. Richardson, Genetic Algorithms with Sharing for Multimodal Function Optimization, in *Proc. Genetic Algorithms Their Applicat. Conf.*,, pp.41-49, 1987.

[4] R. K. Ursem, Multinational evolutionary algorithms, in *Proc. Congr.Evol. Comput.(CEC)*, vol.3., pp.1633-1640, 1999.

[5] R. Storn, K. Price, Differential Evolution — A simple and efficient adaptive scheme for global optimization over continuous spaces, Journal of global optimization, Vol.11, No.4, 341-359, 1997.

[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.

[7] X. Li, Efficient Differential Evolution using Speciation for Multimodal Function Optimization, *In Proceedings of the Conference on genetic and evolutionary computation*, (GECCO'05), 873-880, 2005.

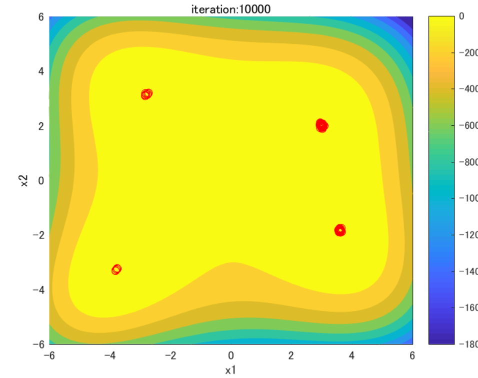
[8] X. S. Yang: A Metaheuristic Bat-Inspired Algorithm, *in: Nature Inspired Cooperative Strategies for Optimization(NICSO 2010), Springer, Berlin*, Vol. 284, pp. 65–74, 2010.

[9] D.Beasley, D.R. Bull, and R.R. Martin, "A sequantial niche technique for multimodal function optimization," *Evolutionary Computation*, vol. 1, no.2, pp. 101-125,1993.

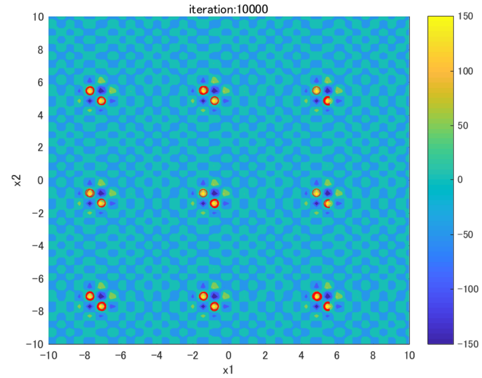
[10] O. M. Shir, T. Back: Dynamic niching in evolution strategies with covariance matrix adaption. *In: Proceedings of the 2005 Congress on Evolutionary Computation(CEC2005)*, Piscataway, NJ, USA, IEEE Press (2005)

[11] T. Iwase, R. Takano, F. Uwano, H. Sato and K. Takadama, "Novelty Search-based Bat Algorithm: Adjusting Distance among Solutions for Multimodal Optimization", *Proceedings of The 22nd Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES 2018)*, pp.29–34, 2018.

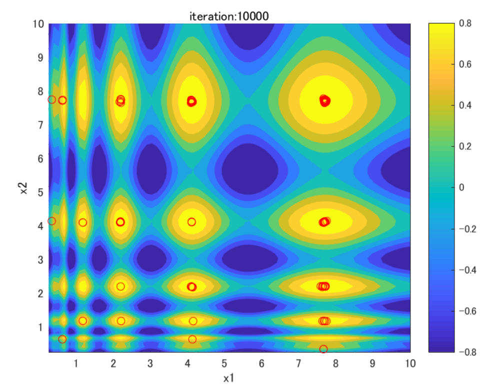
[12] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark Functions for CEC'2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization", *Evol. Comput.* Mach. Learn. Group, RMIT University, Melbourne, VIC, Australia, Tech. Rep., 2013.



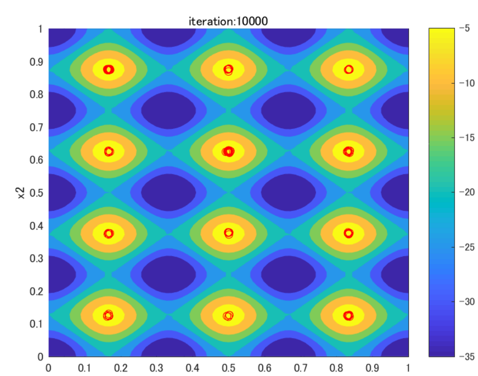
(a) F_1



(b) F_2

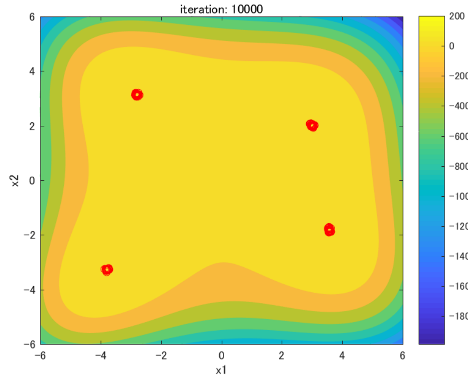


(c) F_3

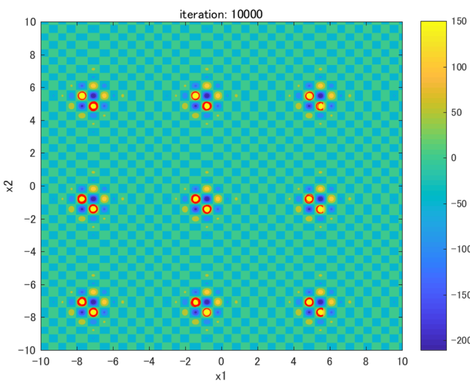


(d) F_4

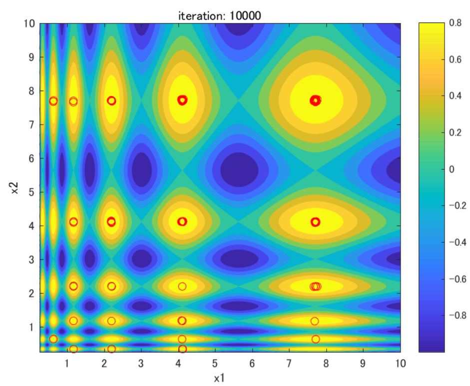
Fig. 5. BA



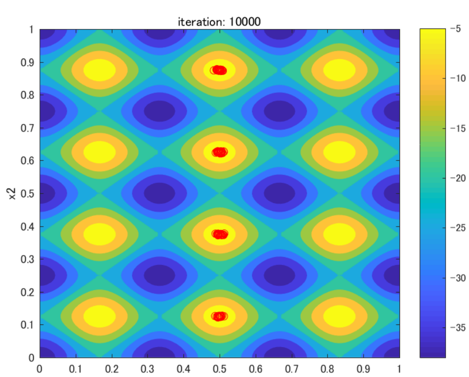
(a) F_1



(b) F_2

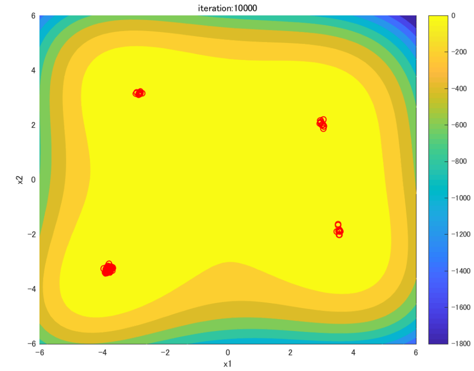


(c) F_3

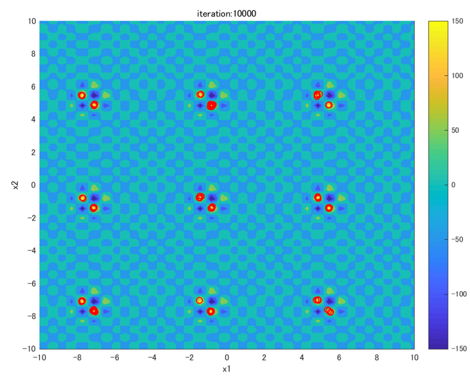


(d) F_4

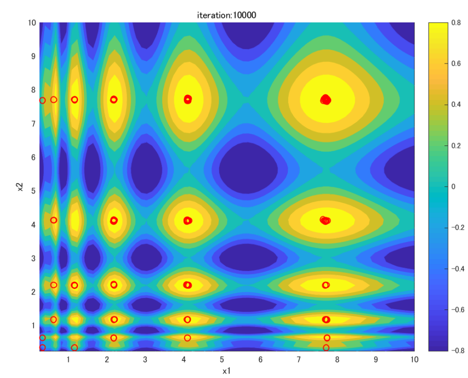
Fig. 6. NSBA



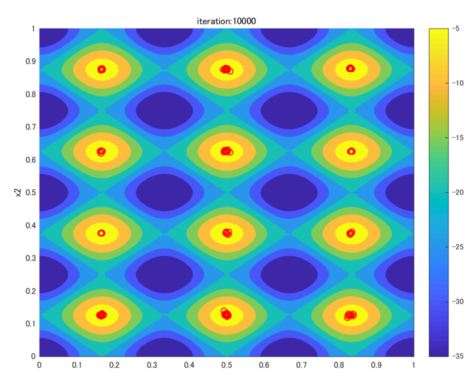
(a) F_1



(b) F_2



(c) F_3



(d) F_4

Fig. 7. NRBA