Novelty Search-based Bat Algorithm Adjusting Direction and Distance among Solutions for Multimodal Optimization

Takuya Iwase † Ryo Takano † Fumito Uwano † Hiroyuki Sato † Keiki Takadama †

† The University of Electro-Communications, Tokyo, Japan

tanu_iwa@cas.lab.uec.ac.jp

Abstract: This paper proposes a new algorithm for finding multiple optima in multimodal optimization, which novelty search-based bat algorithm (NSBA). Conventional algorithms of swarm intelligence tend to straightforward converge a single global optimum or local optimum of high-fitness value. However finding local optima is important because local optima might become better solutions or a new global optimum due to make search domain changed in real world problems. In this paper, bat algorithm (BA) which is available to change exploration and exploitation performance automatically using the characteristic of echolocation, is extended with novelty search for keeping the distance between each solution, and we validate the performance used multimodal functions.

Keywords: Multimodal Optimization, Swarm Intelligence, Bat Algorithm

1. Introduction

Most of metaheuristic algorithms for optimization problem are based on biological evolution in nature-inspired system. These algorithms are adaptable for metaheuristic optimization using non-linear objective functions. For example, Particle Swarm Optimization (PSO) is modeled fish swarm if one fish find a single global optimum, the other fishes converge to the fish [1]. Meanwhile, another algorithm called Firefly Algorithm (FA), which is particularly well for exploitation with flashing light of fireflies [2]. In two fireflies, a brighter firefly attracted the other one. Bat Algorithm (BA) also one of metaheuristic algorithms is available to switch exploration and exploitation automatically with characteristic of echolocation [3]. All bats move to single bat which found food or prey, with loudness and pulse rate to sense the distance each other. Simultaneously, some of them fly randomly for searching the other prey globally. After finding a prey, they will drop loudness down and raise pulse rate up automatically, for adjusting to search domain. However considering for dealing with real-world problems which has many multiple optima in search domain, these algorithms are not adaptable because they tend to converge a single global optimum without keeping local optima.

However, exploitation is still higher performance than exploration, bat algorithm is easily fallen a single global optimum or high-fitness value on multimodal problems. For this reason, we propose distributed BA for migrating solutions away and keeping distance each other. Besides for the performance measurement, we set different changes: (a) for guiding local search using personal best solution or previous position of solution instead of a global best solution; (b) existence or nonexistence of a new solution generated by flying randomly.

This paper is composed of 7 sections. After introduction, we demonstrate mechanism of conventional BA and proposed BA in 2nd and 3rd section. In 4th section, we describe about the experiment using the multimodal functions and parameters. Followed by results in 5th section, we discuss about the results in 6th section, conclusion is 7th section finally.

2. Bat Algorithm

BA based on microbat behavior uses frequency and loudness for adaptive global search on a multimodal function. When microbat moves toward target, loudness A^0 is gradually decreased in proportion to travel distance of microbat decreases. Behavior of microbat is consists of following three steps:

- 1st Search Phase: Each bat flies to a target controlled by frequency f_i.
- 2nd Search Phase: Each bat flies around a target.
- 3rd Search Phase: Each bat flies randomly in search area.

Each bat with velocity v_i, location x_i, and frequency f_i is defined as follows:

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

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

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

Velocity v_i controlled by tuning frequency f_i from [f_{max}, f_{min}] as $f_{max} = 1$ and $f_{min} = 0$. β is uniform random distribution from 0 to 1. In 1st search phase, each bat moves

to location x_i with velocity v_i toward a global best solution . Secondly in 2nd search phase, a new solution x_{loc} is generated around a global best solution . The equation as below

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

where ϵ is uniform random distribution between [0, 1]. A^t is the average loudness of all bats. In 3rd search, a new solution x_{rnd} is generated randomly in search domain as follows:

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

x_{ub} and x_{lb} describe upper and lower limit of search domain. Initialized all bats start searching target using loudness A_i and the reflect wave as pulse emission rate r_i. Loudness and pulse rate are updated as follows:

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

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

These parameters are updated when a new solution is updated by equation (4) for each iteration. Loudness gradually decreases as approaching to a target, pulse rate increases in contrast. BA initializes pulse rate as a uniform random distribution r_i^0 between [0, 1] or a number closed around zero. α and γ are symbolized damping coefficient. In simulated experiment, these coefficient parameters are set $\alpha = \gamma = 0.9$. The pseudo code and the process of BA presented as below.

- STEP1: Initialize population of bats (line 1 to 3)
  Initialize population of bats $x_i(i = 1, 2, ..., N)$, loudness A_0, parse rate r_i and frequency f_i as initial value.
- STEP2: Generate new solutions (line 6)
  Generate new solutions x_i^t based on equation (3).
- STEP3: In 2nd search phase, Generate a new solution around global best solution x_* (line 7 to 9)
  In case of a random distribution higher than parse rate r_i, generate a new solution x_{loc} around x_*.
- STEP4: Generate a new solution randomly (line 10)
  Generate a new solution x_{rnd} by random generation of bat.
- STEP5: Rank and update solutions (line 11 to 14)
  In case of $rand < A_i$, choose the best from all solutions which are x_i, x_{loc}, and x_{rnd}, and cross over as personal best solution unless it is higher than the value of former iteration.
- STEP6: Loop 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: **end while**

3. Proposed Algorithm

3.1 Using Novelty Search

Novelty search[4] is used as evolutionary search approach to expand dense solutions into sparse area and to measure the distance between current solutions to reward or delete it. The sparseness of solutions is calculated as below,

$$\rho(x) = \frac{1}{k}\sum_{i=0}^{k} dist(x, \mu_i), \tag{8}$$

where the sparseness $\rho(x)$ at a point x shows the scatter of solutions. The dist in k-nearest neighbors is the average distance between the point x and μ_i, which is the ith nearest neighbor of x. This is an example in case of k neighbor = 3 (shown in Fig. 1). It describes that a solution is migrated away from three neighbors.

Fig. 1: distributed a solution to sparse area

3.2 Novelty Search-based Bat Algorithm

In order to adapt multimodal optimization not only single objective optimization, Novelty Search-based Bat Algorithm (NSBA) enables all population to reach local optima. This paper proposes a method of keeping over a certain distance between each location of bat, and letting population remain around local optima. All population are updated by the equations as bellow,

$$d_i^{t-1} = \frac{1}{N}\sum_{j=1}^{N}\frac{(x_{i*} - x_j^{t-1})}{|x_{i*} - x_j^{t-1}|^2} \tag{9}$$

$$v_i^t = v_i^{t-1} + d_i^{t-1} * f_i \tag{10}$$

where N is population size, and x_{i*} indicates current personal best solution. x_i^{t-1} is previous position of solution. In addition, bats with velocity v_i^t and location x_i^t are updated same as (10) and (3) of conventional method. Used distance function in Novelty search describes scalar equation. However in this proposes, we alter scalar to vector equation for determining search direction.

3.3 Distance of each bat

Above-mentioned Novelty search, as distance of each bat is closer, they hardly move to sparse area. Conversely, as they located far away each other, they move greatly up to a boundary of search area. To control this movement, we introduce the denominator as equation (9). Here is the Algorithm flow on global minimum optimization. The NSBA pseudo code is described in Algorithm 2.

- STEP1: Initialize population of bats (line 1 to 3)
  Initialize location $x_i(i = 1, 2, ..., N)$ with velocity $v_i(i = 1, 2, ..., N)$ randomly. Each bat has loudness A_0, parse rate r_i and frequency f_i as initial value.
- STEP2: Generate new solutions (line 6)
  Generate new solutions x_i^t based on equation (10)(3) with (9).
- STEP3: In 2nd search phase, Generate a new solution around solutions x_i (line 7 to 9)
  In case of a random distribution higher than parse rate r_i, generate a new solution x_{loc} around x_i.
- STEP4: Generate a new solution randomly (line 10)
  Generate a new solution x_{rnd} by random walk of bat.
- STEP5: Rank and update solutions (line 11 to 15)
  If $rand < A_i$, choose the best from all solutions which are x_i, x_{loc}, and x_{rnd}. After that, cross over as personal best solution unless it is higher fitness value than previous iteration.
- STEP6: Loop to STEP2

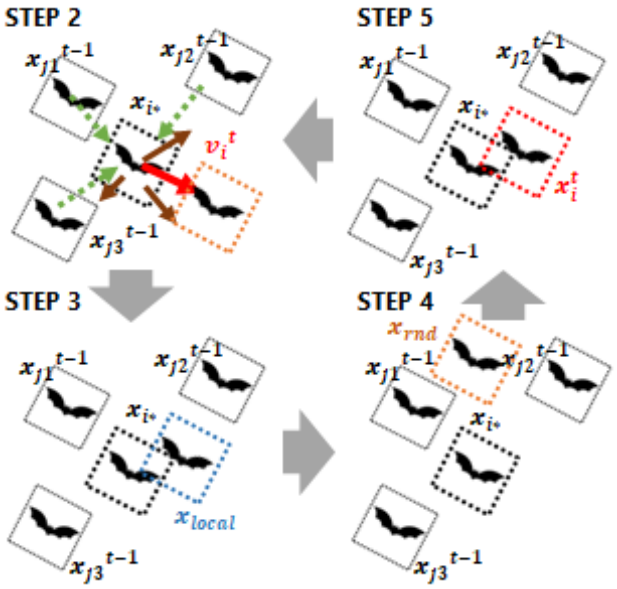


Fig. 2: Bat motion of NSBA

Algorithm 2 Novelty Search-based 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 update velocity v_i [eqs.(10)(3)(9)]
7: **if** $(rand > r_i)$ **then**
8: Generate a new solution x_{loc} around the solution x_i [eq.(4)]
9: **end if**
10: Generate a new solution x_{rnd} randomly (or without x_{rnd})
11: **if** $(rand < A_i \ \& \min(F(x_i), F(x_{new}), F(x_{rnd})) < F(x_{i*}))$ **then**
12: Accept the new solution, and update pulse rate r_i & loudness A_i [eqs. (6)(7)]
13: **end if**
14: **end for**
15: Evaluate the all bats and select a best solution x_{i*} in the current solutions
16: **end while**

4. Experiment

We compared proposed NSBA with BA to validate the performance. In this paper, the algorithm is implemented on MATLAB for the benchmark test functions in [5].

4.1 Benchmark Test Functions

Table. 1 shows the benchmark test functions, the number of optima and the fitness value. Fig. 3(b) & 4(b) describe the search domain of function as horizontal axis x1 and vertical axis x2. The color density describes how fitness value changes. As color becomes darker area, fitness value gets lower. For validating NSBA to distribute spread widely, there are 2 multimodal functions. Focused on depth of fitness value, scale of multimodal domain and number of local optima, we used these functions as following section.

Table. 1: Measurement of Benchmark Test Functions

Function	F_1	F_2
Search Domain	$-10 \leq x_i \leq 10$	$-5 \leq x_i \leq 5$
$F(x_*)$	0	0
Num of global optima	1	1
Num of local optima	16	120

F_1: Griewank Function

As an example to demonstrate the bat motion of proposed algorithm, we use Griewank function as below (shown in Fig. 3(a))

$$F(x) = \sum_{i=1}^{D} \frac{x_i}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1, \qquad (11)$$

where global optimum is $f(x_*) = 0$, at $x_* = [0, 0]$. There are 17 local optima at $\pm x \approx [6.2800, 8.8769]$, $[3.1400, 4.4385]$, $[0, 8.8769]$, $[6.2800\ 0]$, $[9.4200, 4.4385]$ in the range of this function is between $-10 \leq x_i \leq 10$ with $i = 1, 2$. The function $f(x)$ has global minimum $f(x_*) = 0$ and also the other local optima $f(x_{i*}) \approx 0$ for $D = 2$.

F_2: Rastrigin Function

This function has 121 local optima in the search domain, at $\pm x = [0, ..., 11\ \ 0, ..., 11]$. And global minimum is $f(x_*) = 0$ at $x = [0, \ 0]$. The function equation is

$$F(x) = 10D + \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i)] \qquad (12)$$

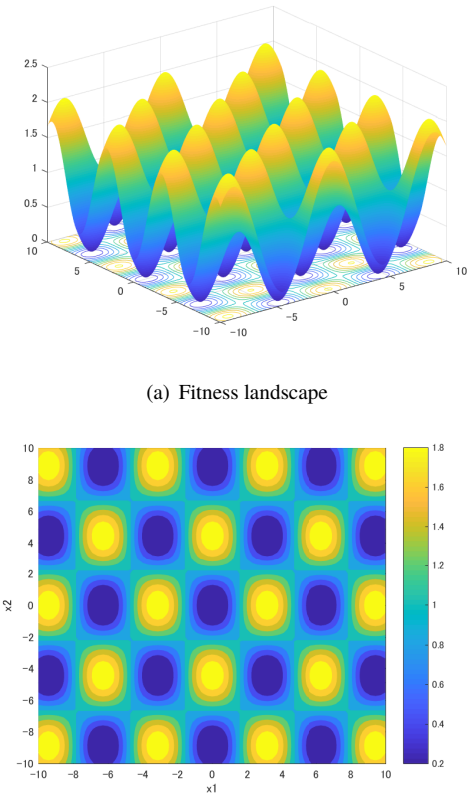
The search domain is $-5 \leq x_i \leq 5$ with $i = 1, 2$. 3D model and contour of this function are showed in Fig. 4(a) & 4(b).

4.2 Evaluation Criteria

In this experiment, Peak Ratio(PR)[6] which used by on CEC 2013 competition [7] measures how many found global and local optima. The equation as below,

$$PR = \frac{\sum_{run=1}^{MR} FPs}{TP * MR} \qquad (13)$$

where MR is maximum run, and FPs is found peaks. TP means all known peaks of the function.



(a) Fitness landscape

(b) Contour plot

Fig. 3: F_1: Griewank Function

4.3 Experimental Parameters

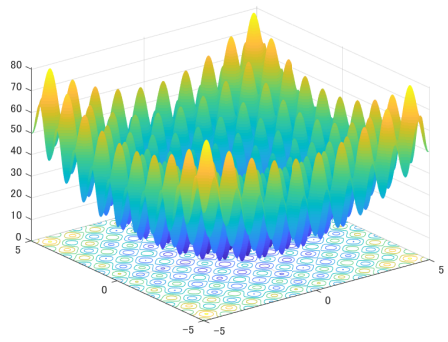
All experiments use same parameters as follows, frequency $f_{max} = 1$, $f_{min} = 0$, loudness $A^0 = 1$, parse rate $r^0 \in [0, 1]$ with $\alpha = \gamma = 0.9$. population size $N = 50, 100$ for Griewank function and $N = 100, 150$ for Rastrigin function. Maximum iteration set 10000. and these algorithms are run 50 times on each function.

4.4 Result

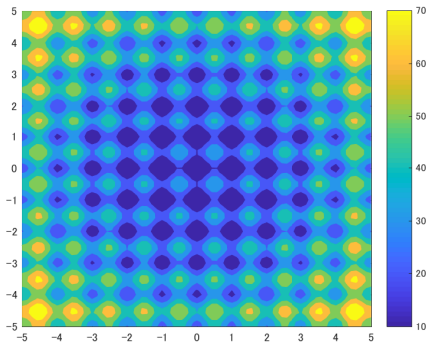
We evaluated each algorithm and compare these performance of convergence speed and the number of found peaks (as shown in Table. 2). From this table, NSBA performed better than BA, reaching some peaks on each function. Fig. 5 & 6 show population (marked with white circle) of each algorithm at final iteration. BA converged straightforward to a single global optima on any run. Meanwhile, NSBA distributed population to a single global optima and around it.

Table. 2: Found Peaks and Peak Ratio of BA and NSBA

	BA			NSBA		
Function	Mean	SD	PR	Mean	SD	PR
F_1 (N = 50)	1.0	0	5.89 %	6.8	0.7024	40.0 %
F_1 (N = 100)	1.0	0	5.89 %	7.267	0.5735	42.75 %
F_2 (N = 100)	1.0	0	0.87 %	7.9333	0.8929	6.56 %
F_2 (N = 150)	1.0	0	0.87 %	8.0667	0.7717	6.67 %



(a) Fitness landscape



(b) Contour plot

Fig. 4: F_2: Rastrigin Function

5. Discussion

5.1 Convergence Speed

To measure the convergence speed of proposed algorithm, we present the results of peak ratio in each iteration (shown in Fig. 7). The graph shows that vertical axis is iteration and horizontal axis is peak ratio. BA is sharply decreased to almost 0 % until 1000 iteration followed by remained on each function (as shown in Fig. 7-7(a) to 7-7(d)). Between beginning of iteration and 1000 iteration, NSBA increased strongly up to 70 % on F_1. After 1000 iteration, it gradually went down until final iteration. Beginning of the iteration on F_2, NSBA was the highest peak ratio 0.2 %. However, it decreased to 10 % at 1000 iteration followed by being stable.

5.2 Population Size

In this section,

5.3 Distribution of Population

6. Conclusion

We validated the performance of proposed bat algorithms for k-nearest neighbor and novelty search with changes of updating solutions and generating a new solution randomly. As a result, both algorithms performed for reaching local optima with global optimum. Especially the method using personal best without s_{rnd}, performed better than the other proposed methods. However, we have to adjust parameter k which is the number of neighbors for feasible multimodal functions. As population size of bat increases, the number of searched local optima also increased. Our future prospects are adapting this algorithm for the other benchmark functions, and blushing up the performance to cover unspecified large number of local optima. Future experiments on the other multimodal functions and investigation will be studied.

References

[1] Eberhart, R. C., and Kennedy: A New Optimizer Using Particle Swarm Theory, *Proc. Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan), IEEE Service center, Pis-cataway, NJ*, No. 1, pp. 39–43, 1995.

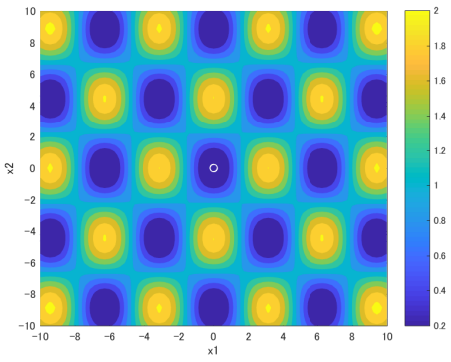
[2] Yang, X. S: Firefly Algorithms for Multimodal Optimization, *in:Stochastic Algorithms: Foundations and Applications, SAGA*, Vol. 5792, pp. 169–178, 2009.

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

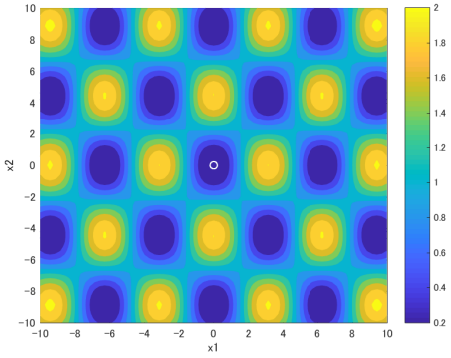
[4] Joel Lehman and Kenneth O Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pp. 329–336, 2008.

[5] Surjanovic, S. and Bingham, D. (2013). Virtual Library of Simulation Experiments: Test Functions and Datasets , Retrieved October 9, (2017)

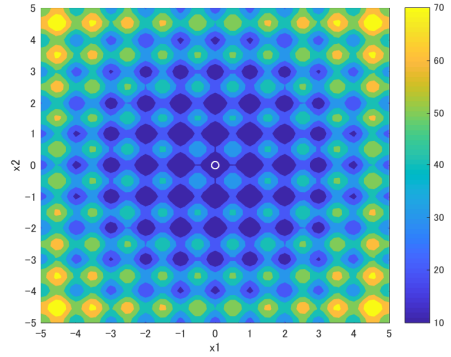
[6] R. Thomsen,"Multimodal Optimization using Crowding-based Differential Evolution,"*In the IEEE Congress on Evolutionary Computation,*2004. CEC2004, vol.2, pp. 1382-1389, 19-23 June, 2004.



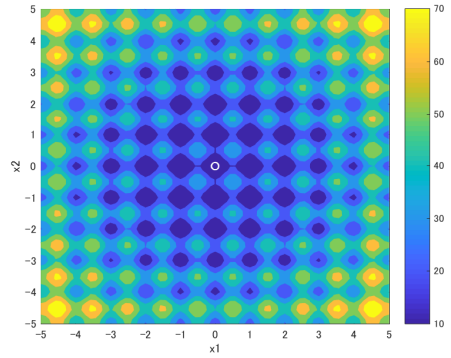
(a) $F_1 : (N = 50)$



(b) $F_1 : (N = 100)$

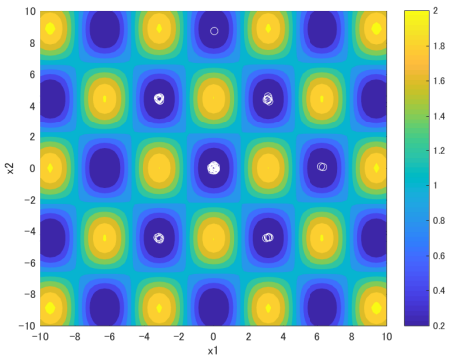


(c) $F_2 : (N = 100)$

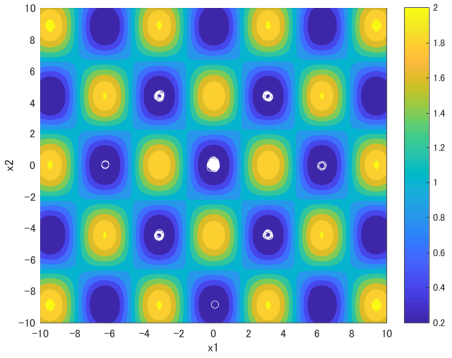


(d) $F_2 : (N = 150)$

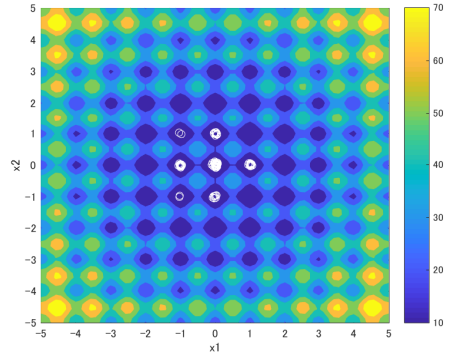
Fig. 5: BA



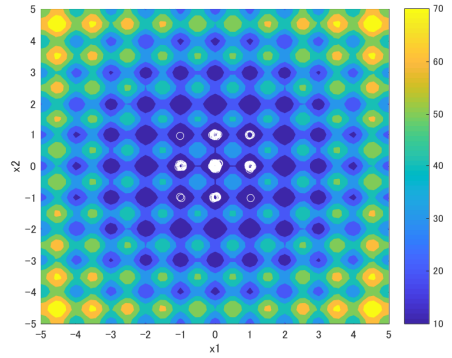
(a) $F_1 : (N = 50)$



(b) $F_1 : (N = 100)$

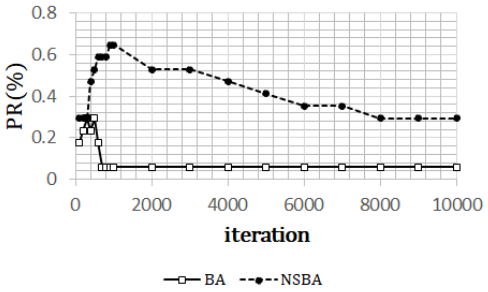


(c) $F_2 : (N = 100)$

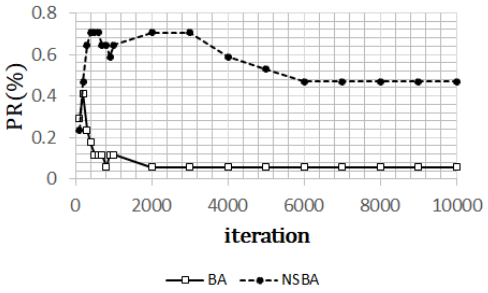


(d) $F_2 : (N = 150)$

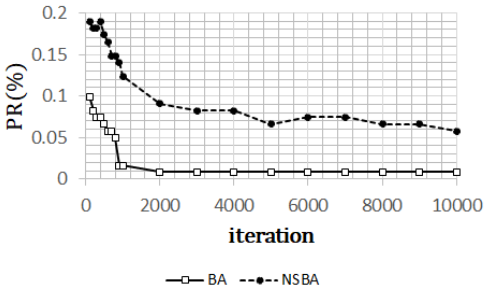
Fig. 6: NSBA



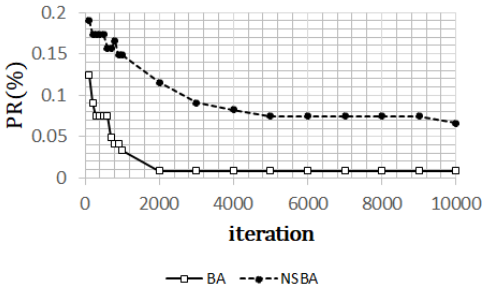
(a) $F_1 : (N = 50)$



(b) $F_1 : (N = 100)$



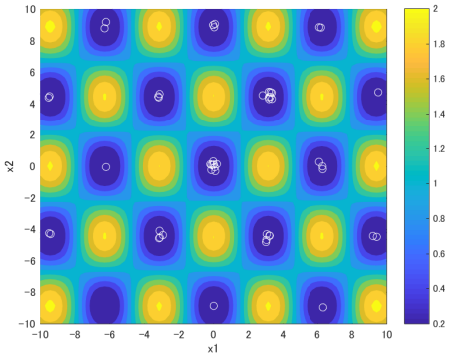
(c) $F_2 : (N = 100)$



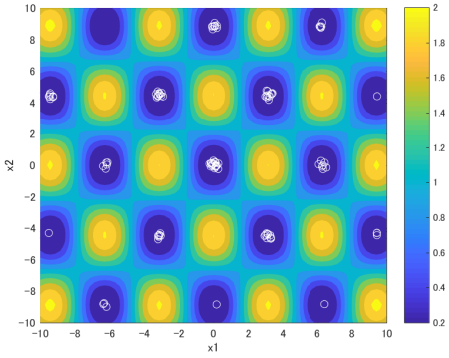
(d) $F_2 : (N = 150)$

Fig. 7: Convergence Speed of Peak Ratio implemented by BA and NSBA

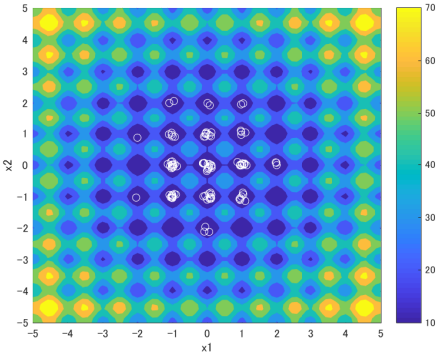
[7] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark Functions for CEC'2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization", *Evol. Comput.* Mach. Learn. Group, RMIT University, Melbourne, VIC, Australia, Tech. Rep., 2013.



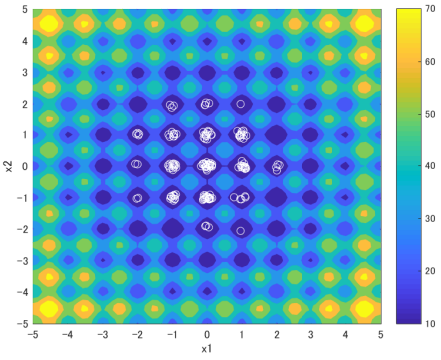
(a) $F_1 : (N = 50)$



(b) $F_1 : (N = 100)$



(c) $F_2 : (N = 100)$



(d) $F_2 : (N = 150)$

Fig. 8: Distribution of population of NSBA at 1000 iteration