Novelty Search-based Bat Algorithm: Adjusting 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,takano@cas.lab.uec.ac.jp,uwano@cas.lab.uec.ac.jp,h.sato@uec.ac.jp, keiki@hc.uec.ac.jp

Abstract: This paper proposes the novelty search-based bat algorithm (NSBA), which aims to search new solutions which have not yet searched to find as many local optima as possible in multimodal optimization. In detail, this paper focuses on bat algorithm (BA) which copes with the trade-off between exploration and exploitation in the process of the solution search and extends it by introducing novelty search for keeping the distance among solutions. Through simulations of the comparisons between NSBA and BA in the test-bed multimodal functions, the following implications have been revealed: (1) NSBA finds more number of local optima than BA in both Griewank and Rastrigin Functions; (2) the number of local optima in NSBA increases as the number of populations increases, while that in BA does not change even though the number of populations increases in both functions.

Keywords: swarm intelligence, bat algorithm, novelty search, multimodal function

1. Introduction

Most of metaheuristic algorithms for optimization problems are based on biological evolution as nature-inspired system. For example, Particle Swarm Optimization (PSO) [1] modeled as fish swarm searches solutions by considering both the local best solution of their own fishes and the global best solution among all fishes. Firefly Algorithm (FA) [2] searches solutions by moving to a brighter firefly (*i.e.*, better solution). As other algorithm, Bat Algorithm (BA) [3] searches solutions according to the characteristic of echolocation which promotes bats to start to find food or prey (*i.e.*, solutions) widely and narrows down the target food or prey by changing their loudness and pulse emission rate. In detail, all bats continuously search solutions by selecting the better solution than the current one, and reduces the number of times of selecting the better solution, which can be found by one of the following three searches: (i) a movement toward the target *i.e.,* the bat which finds the best solution; (ii) a local movement around the target as a local search; and (iii) a random movement as a *global* search. However, all of these algorithms (*i.e.,* PSO, FA and BA) are not appropriate for dealing with real-world problems which need to find many local optima in search space. This is because the above algorithms are designed to find one single global optimum.

To overcome this problem, this paper focuses on BA and extends it to propose the novelty search-based bat algorithm (NSBA), which aims to search new solutions which have not yet searched to find as many local optima as possible. In this research, BA is employed because of the following reasons: (1) BA based on the three search mechanisms from (i) to (iii) as described above, while PSO and FA is mainly based on one search mechanism (which is respectively based on the local and best solutions in PSO and the movement to better solutions in FA); and (2) this difference suggests that one of search mechanisms in BA can be modified for finding multiple local optima with keeping the two search mechanisms as original, while the search mechanism in PSO and FA are hard to be modified for such a purpose due to a lack of the original search mechanism. What should be noted here is that BA still has the functions of exploitation and exploration in the solution search process as a local search by the mechanism (ii) and a global search by the mechanism (iii).

This paper is organized as follows. After this section, the mechanisms of BA and the proposed BA 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 and Section 6 discusses them. Finally, our conclusion is given in Section 7.

2. Bat Algorithm

As described 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 mechanisms: (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 mechanisms. First, in the search mechanism (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:

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

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

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

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-1}, the distance between the global best location and the previous location $x_* - x_i^{t-1}$, 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 mechanism (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 mechanism (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: end while
```

3. Proposed Algorithm

3.1 Novelty Search

Novelty search [4] developed in the context of computation aims to search new solutions which have not yet searched. For this purpose, novelty search measures the distance among the candidate solutions to evaluate the dense of them and then generates new solutions into sparse area by considering the distance among candidate solutions. The sparseness of the solutions is calculated as follows:

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

where $\rho(x)$ is the sparseness at the location x, k is the number of solutions around x in the k-nearest neighbors, and $dist$ (x and μ_i) is the distance between x and μ_i which is the i-th nearest neighbor of x. The $\rho(x)$ value indicates the averaged distance between x and solutions around x. Fig. 1 shows an example of the case where $k = 3$ (*i.e.*, the number of solutions around the target solution). where the yellow star indicates the target solution and the red circles indicates the other solutions. In this figure, the yellow solution moves to the sparse area where is far from other three red solutions.

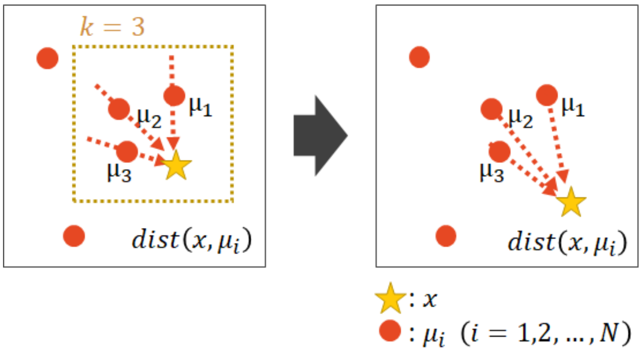


Fig. 1: distributed a solution to sparse area

3.2 Novelty Search-based Bat Algorithm

Novelty Search-based Bat Algorithm (NSBA) is extended from BA by introducing the concept of novelty search, however, we change the value of sparseness equation mentioned above eq.(8) to vector equation. The essential difference between NSBA and BA is related to the search mechanism (i). Concretely, all solutions are updated by adding their velocity calculated by the following Eqs. (9) and (10), which corresponds to Eq. (2) in BA.

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

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

where N is population size, x_{i*} is the current personal best solution, and x_i^{t-1} is previous solution. All bats with velocity v_i^t and location x_i^t are updated same as (10) and (3) of BA. Used distance function in Novelty search describes scalar equation. However in this proposes, we alter scalar to vector equation for determining search direction. Next, in the search mechanism (ii), x_{loc} is generated around the personal best solution x_{i*} as follows:

$$x_{loc} = x_{i*} + \epsilon A^t, \tag{11}$$

where ϵ is uniform random value between [0, 1].

3.3 Algorithm of NSBA

The pseudo code of NSBA is given in the Algorithm 2, and its brief summary is described below. Except for the step 2, the other steps are the same as BA.

- STEP1: Population initialization of bats (line 1 to 3)
  This step is the same as BA.
- STEP2: New solution update (line 6)
  The new solutions x_i^t are calculated by Eqs. (10)(3) with (9).
- STEP3: New solution generation around solutions x_i (line 7 to 9)
  A new solution x_{loc} is generated around x_{i*} by Eq. (11)
- STEP4: Generate a new solution randomly (line 10)
  This step is the same as BA.
- STEP5: Solutions update (line 11 to 15)
  This step is the same as BA.
- STEP6: Return to STEP2

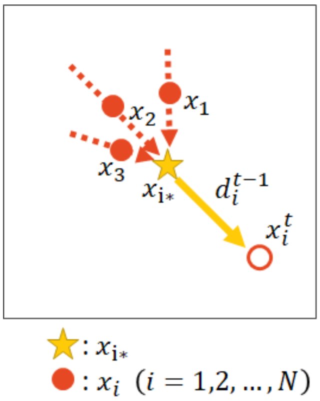


Fig. 2: Bat motion of NSBA

Algorithm 2 Novelty Search-based Bat Algorithm

Require: Objective Function $F(x)$

```
Initialize Population x_i (i = 1, 2, ..., N) and v_i
Define frequency f_i at location x_i [eq.(1)]
Initialize pulse rates r_i, and loudness A_i
while (t < Max number of iterations) do
    for i=1 to N do
        Generate a new solution x_i and update velocity v_i [eqs.(3)(9)(10)]
        if (rand > r_i) then
            Generate a new solution x_loc around the solution x_i [eq.(11)]
        end if
        Generate a new solution x_rnd randomly (or without x_rnd)
        if (rand < A_i & min(F(x_i), F(x_new), F(x_rnd)) < F(x_i*)) then
            Accept the new solution, and update pulse rate r_i & loudness A_i [eqs. (6)(7)]
        end if
    end for
    Evaluate the all bats and select a best solution x_i* in the current solutions
end while
```

4. Experiment

To validate the effectiveness of NSBA, this paper employs the following well-known multimodal functions which have similar fitness landscape: Griewank function [5] and Rastrigin function [6].

4.1 Benchmark Test Functions

Table. 1 summarizes the features of the benchmark multimodal functions. In detail, the first, second, third, and fourth lines indicate the search space, the fitness value of the global optimum, the number of global and local optima, respectively. Figs. 3(a) & 4(a) show the fitness landscape of both functions: 3(b) & 4(b) show the contour plot of both functions from the two-dimensional viewpoint. These figures indicate a change of the fitness value with the color density, where the horizontal and vertical axes indicate x_1 and x_2, respectively. As a color becomes darker, the fitness value becomes smaller. This paper employs Griewank and Rastrigin functions because both functions are similar fitness landscape but they are some differences of the number of the local optima, search space and fitness value.

F_1**: Griewank Function**

This function is described as follows as shown in Fig. 3(a).

$$F(x) = \sum_{i=1}^{D} \frac{x_i}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1, \qquad (12)$$

Table. 1: Measurement of Benchmark Test Functions

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

where D is the number of dimension and the global optimum is $f(x_*) = 0$, at $x_* = [0,\ 0]$. In the case of $D = 2$ this function has 17 local optima $f(x_{i*}) \approx 0$ 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 between $-10 \le x_i \le 10$ with $i = 1, 2$.

F_2**: Rastrigin Function**

This function is described as follows as shown in Fig. 4(a).

$$F(x) = 10D + \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i)] \qquad (13)$$

where D is the number of dimension and the global optimum is $f(x_*) = 0$ at $x = [0,\ 0]$. In the case of $D = 2$, this function has 121 local optima in the search space at $\pm x_i = [0, ..., 11\ \ 0, ..., 11]$ in the range between $-5 \le x_i \le 5$ with $i = 1, 2$.

4.2 Evaluation Criteria

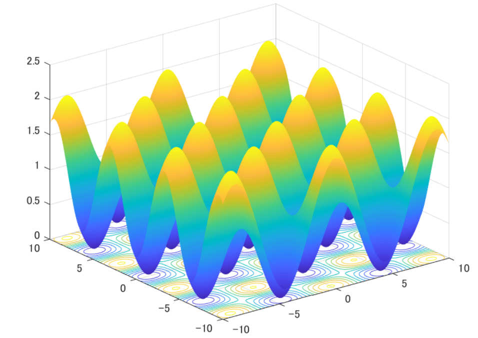
This experiment employs Peak Ratio(PR) [7] as the evaluation criterion in the CEC (IEEE Congress on Evolutionary Computation) 2013 competition [8]. The PR value measures the ratio of the found global and local optima in the total number of true peaks and it is calculated as follows:

$$PR = \frac{\sum_{run=1}^{MR} FPs}{TP * MR} \qquad (14)$$

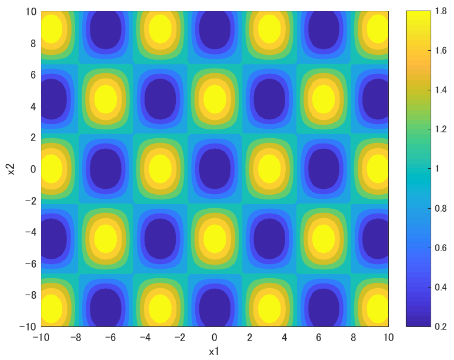
where MR indicates the maximum run, FPs indicates indicates the number of peaks found by the optimization algorithm. TP indicates the true number of peaks of the function. We define that the peak is found when the Euclid distance between the true peak and the solution calculated by the optimization algorithm is less than the threshold $\varepsilon = 0.1$.

4.3 Experimental Parameters

All experiments employs 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 $N = 50, 100$ for Griewank function and $N = 100, 150$ for Rastrigin function. In both functions, the optimization algorithm runs 30 time, each of which maximum iteration is 10000.

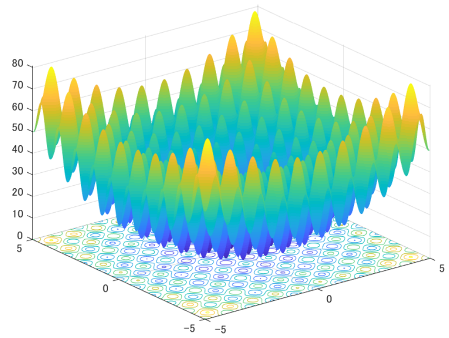


(a) Fitness landscape

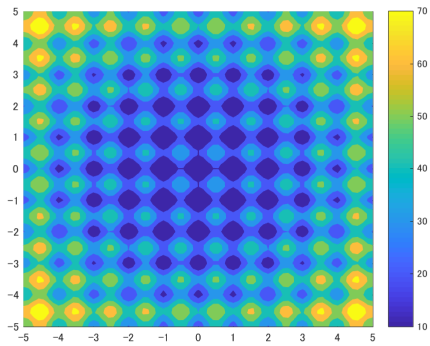


(b) Contour plot

Fig. 3: F_1: Griewank Function



(a) Fitness landscape



(b) Contour plot

Fig. 4: F_2: Rastrigin Function

5. Result and Discussion

To evaluate NSBA, this paper compares the performance between NSBA and BA in terms of (a) the number and ratio of the found peaks and (b) the convergence speed.

5.1 Number and Ratio of Found Peaks

From the viewpoint of the (a) criteria, Table. 2 compares both algorithms in terms of the Mean and SD of the number of the found peaks and the PR value of averaged over 30 runs. From this table, NSBA performs better than BA because both the number of the found peaks and the PR value of NSBA is larger than those of BA in both functions with the different population size. Figs. 5 and 6 show the location of the solutions (marked with the white circle) of both algorithms at the final iteration. BA converged to the single global optimum on any run, while NSBA distributes the solutions to not only the single global optimum but also the other local optima.

5.2 Convergence Speed

From the viewpoint of the (b) criteria, Fig. 7 measures the convergence speed of the peak ratio (PR) of NSBA and BA. In this figure, the vertical axis indicates the peak ratio while horizontal axis indicates the iteration. The lines with the black and while circle indicate the PR value of NSBA and BA, respectively. As shown in Fig. from 5(a) to 5(d), the PR value in BA sharply decreases to almost 0 % until 1000 iterations and keeps the same value in both functions. In comparison with the BA, the PR value of NSBA in F_1 increases around 70 % until 1000 iterations and then gradually decreases to around 30-50 % after 1000 iterations. However, the PR value of NSBA in F_2 does increase but gradually decreases from 20 % to 10 % after 1000 iterations. This results suggests that the search mechanism (ii) works to spread solutions away toward sparse area, however, NSBA has strong convergence to the global best solution and the personal best solutions around it.

5.3 Analysis of Results

5.3.1 Population Size

In order to investigate the influence of the population size, this subsection analyzes the results of BA and NSBA in the different population size. From Table 2, the number of the found peaks and the PR value of BA do not change by the different population size in both functions, while those of NSBA increase as the population size increases. Concretely, the number of the found peaks and the PR value of NSBA increase from 6.8 and 40 % (with 50 population size) to 7.267

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 %

and 42.75 % (with 100 population size) in F_1 and from 7.9333 and 6.56 % (with 100 population size) to 8.0667 and 6.67 % (with 150 population size) in F_2. This results suggests that the large size of the population contributes to increasing the performance of NSBA while does not contributes to BA because the distributed solutions by NSBA can cover many local optima as the population size increases, while the converged solution by BA mainly cover the only global optimum even though the population size increases. Fig. 7 also supports the above good/bad influence of the population size in NSBA, *i.e.*, the PR value of NSBA in F_1 decreases from 70 % to 30 % in the case of the small (50) population size, while that in F_1 keeps around 50 % after 6000 iterations in the case of the large (100) population size.

5.4 Distribution of Solutions

In order to investigate the reason why the PR values of NSBA in both function decreases in Fig. 7 (*i.e.* why NSBA cannot keep the maximum PR value), this subsection analyzes how the solutions are distributed in Fig.8 which shows the location of the solutions in 2D dimension at the 1000 iterations. From Fig. 8, NSBA can cover almost all peaks at the 1000 iterations on F_1 but the number of the found peaks decreases after the 1000 iterations. This because NSBA has the strong convergence to the best solution in the evolutionary process, even though NSBA has a potential of finding the global optimum and many local optima. The same tendency of the decrease of the found peaks can be found in F_2 even though the shape of the PR value is different.

6. Conclusions

This paper focused on bat algorithm (BA) and extended it to propose the novelty search-based bat algorithm (NSBA), which aims to search new solutions which have not yet searched to find as many local optima as possible in multimodal optimization. Through the comparisons between NSBA and BA in the test-bed multimodal functions, this paper validated the effectiveness of NSBA. Concretely, the following implications have been revealed: (1) although NSBA and BA succeeded to find a global optimum, NSBA found more number of local optima than BA in both Griewank and Rastrigin functions; and (2) the number of local optima in NSBA increased as the number of populations increased, while that in BA did not change even though the number of populations increased in both functions. Our future prospects are summarized as follows: (1) applying NSBA into other benchmark functions, and (2) increasing the performance to find the large number of local optima.

References

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

[3] 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.

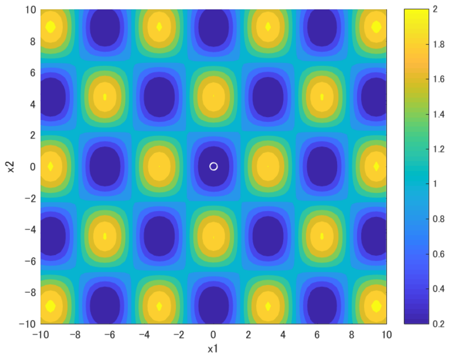
[4] J. Lehman and K. O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pp. 329–336, 2008.

[5] M. Molga, and C. Smutnicki, Test functions for optimization needs (2005). Retrieved June 2013, from `http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf`.

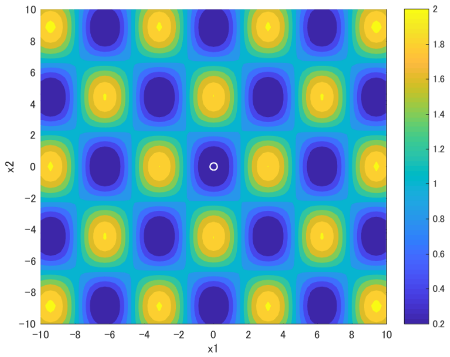
[6] H. Pohlheim, GEATbx Examples: Examples of Objective Functions (2005). Retrieved June 2013, from `http://www.geatbx.com/download/GEATbx_ObjFunExpl_v37.pdf`.

[7] 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.

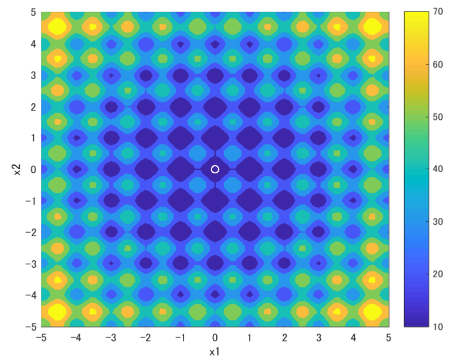
[8] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark Functions for CEC'2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization", *Evol. Comput.* Mach. Learn. Group, RMIT University, Melbourne, VIC, Australia, Tech. Rep., 2013.



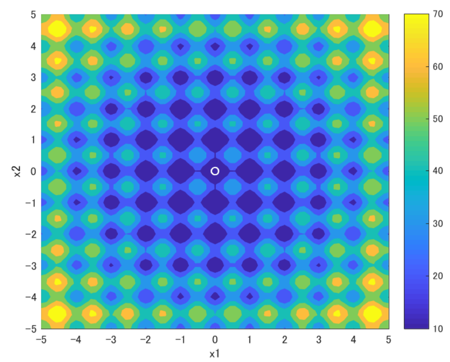
(a) $F_1 : (N = 50)$



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

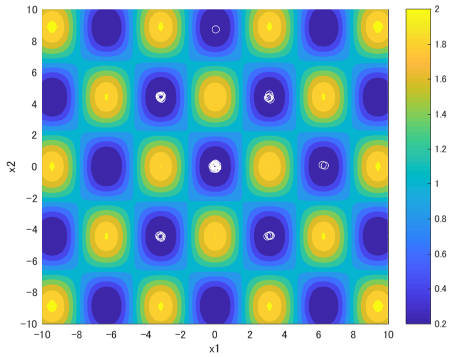


(c) $F_2 : (N = 100)$

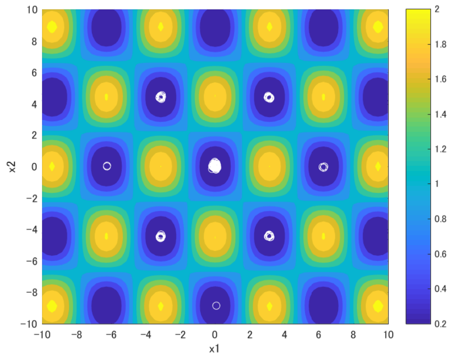


(d) $F_2 : (N = 150)$

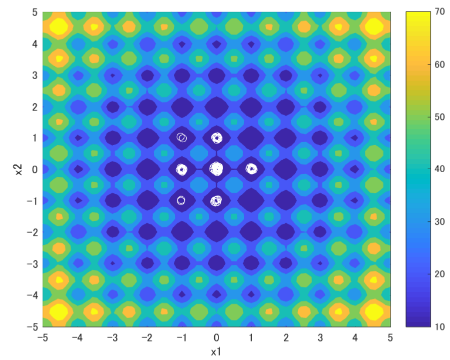
Fig. 5: BA



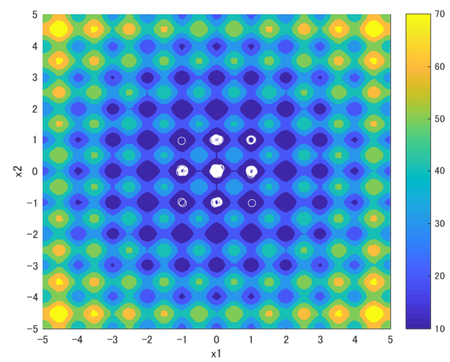
(a) $F_1 : (N = 50)$



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

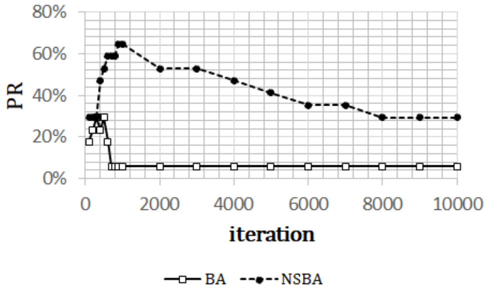


(c) $F_2 : (N = 100)$

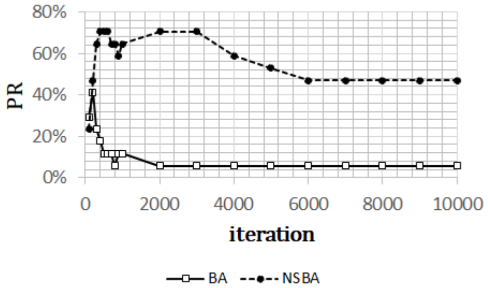


(d) $F_2 : (N = 150)$

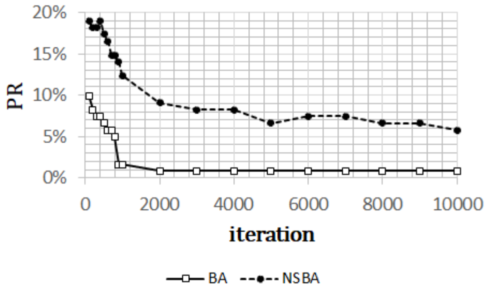
Fig. 6: NSBA



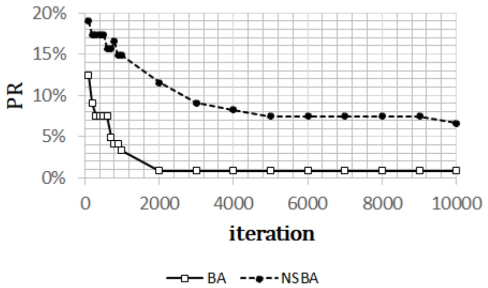
(a) $F_1 : (N = 50)$



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

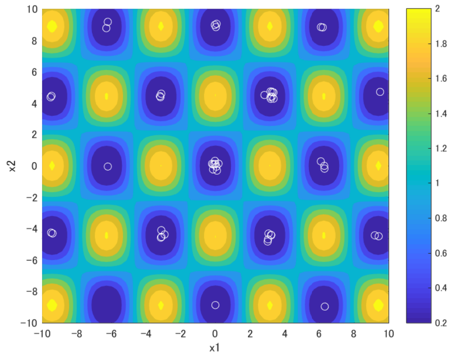


(c) $F_2 : (N = 100)$

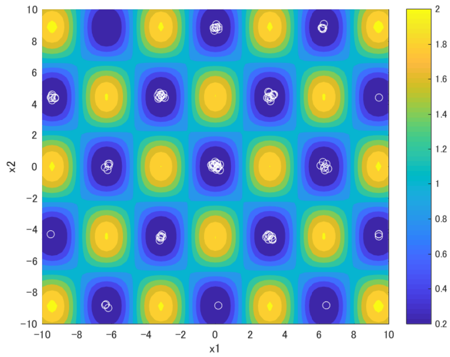


(d) $F_2 : (N = 150)$

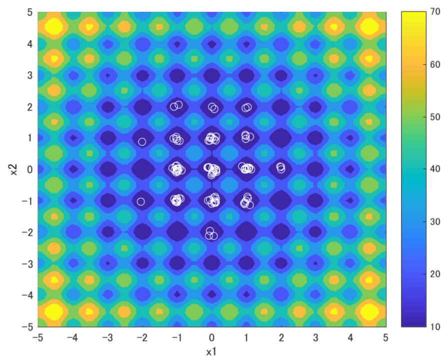
Fig. 7: Convergence Speed of Peak Ratio implemented by BA and NSBA



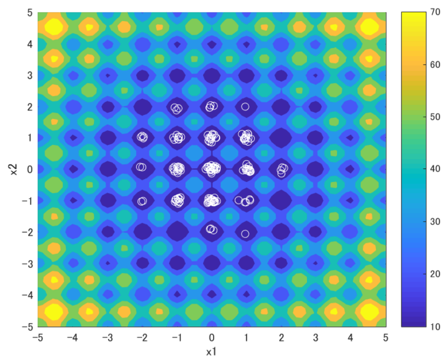
(a) $F_1 : (N = 50)$



(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