Searching multiple local optimal solutions in Multimodal Function by Bat Algorithm based on Novelty Search

Takuya Iwase[†], Ryo Takano[†], Fumito Uwano[†], Yuta Umenai[†], Hiroyuki Sato[†], Keiki Takadama[†]

Department of Informatics, The University of Electro-Communications[†],

1 Introduction

Metaheuristic algorithms became major method for solving optimization problem recently. Generally, they are based on biological evolution in nature-inspired system. These various methods are adaptable for a specific situation using non-linear objective functions. Particle Swarm Optimization (PSO) modeled fish swarm if just a fish find a global optimum, the other fishes move to the fish [1)] . One of the other algorithm is Firefly Algorithm (FA), which is based on flashing light of fireflies. In two fireflies, a firefly is attracted by the other one which is brighter light. They are widely used for optimization problem, bat algorithm is one of these algorithms for searching global optimum with characteristic of echolocation [3)] We need to adapt for global searching because bat algorithm is also fallen local minima easily. For this reason, we propose BA for distributed solutions and validate performance of the algorithm.

2 Bat Algorithm

BA based on echolocation behaviour of microbat uses frequency and loudness for adaptive global search on a multimodal function. In this algorithm, loudness A^0 is used as a parameter to adjust frequency. When microbat moves toward target, loudness A^0 is also gradually decreased in proportion to travel distance of microbat decreases. Behaviour of microbat is consists of following three rules:

- Each bat measures the distance between own location and target using frequency f_i.
- On the location x_i, bat with velocity v_i moves to another bat closed target randomly.
- Loudness A^0 varies to sense how far approaching toward the target.

Each bat with velosity v_i, location x_i, and frequency f_i is updated as follows:

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

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

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

Velocity v_i varies 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. Firstly in global search step, BA calculates the distance from all bats position to current global best solution x_*, when population is generated. And then, each bat moves to location x_i with velocity v_i toward the solution. Secondly in local search step, generates a new solution x_{new} around global best solution. The equation as below

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

where ϵ is uniform random distribution between [0 1]. A^t is the average loudness of all bats. 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:

$$A_i^{t+1} = \alpha A_i^t \tag{5}$$

$$r_i^{t+1} = r_i^0[1 - exp(-\gamma t)] \tag{6}$$

Both of them are also updated in case of updated new solution by equation (4) for each iteration. Loudness gradually decreases as approaching to 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, they are set $\alpha = \gamma = 0.9$.

The pseudo code of basic BA is presented as below.

Algorithm 1 basic Bat Algorithm

Require: *Objective Function* $f(x)$
Ensure: Initialize Population $x_i (i = 1, 2, ..., n)$ and v_i
Define frequency f_i at location x_i
Initialize pulse rates r_i, and the loudness A_i

```
while (t < Max number of iterations) do
    for i=1 to n do
        Generate new solutions x_i by tuning frequency f_i
        Update location x_i and velocity v_i [eqs.(1) to (3)]
        if (rand > r_i) then
            Generate a new solution x_new around global best solution x_i [eq.(4)]
        else
            Continue
        end if
        Generate a new solution x_rnd randomly
        if (rand < A_i & f(x_i) < f(x_*)) then
            Accept the new solution,
            and update pulse rate r_i
            & the loudness A_i [eqs. (5)(6)]
        end if
        Evaluate the all bats and select a best solution x_* in the current solutions
    end for
end while
```

3 Proposed Bat Algorithm

3.1 Novelty Search

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

$$\rho(x) = \frac{1}{k} \sum_{i=0}^{k} dist(x, \mu_i) \tag{7}$$

, 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.

3.2 Bat Algorithm for Novelty Search

In order to adapt multimodal optimization not only single objective optimization, BA based on Novelty Search (NSBA) enables all population to reach local optimal solutions. This paper proposes a method of keeping over a certain distance between each location of bat, and letting population remain around local optimal solution. To use the characteristic, all population are concentrated at a point when initializing. Based on novelty search, updated equation of the distance between each other of population, is written as

$$d_i^t = \frac{1}{N} \sum_{j=1}^{N} (x_i^t - x_j^t) * \delta^{\frac{5}{|x_i^t - x_j^t|}}, \tag{8}$$

where N is population size, and δ is used as a parameter. Here is $\delta = 1.2$ in this simulation. In addition, bats with velocity v_i^t is updated as follows:

$$v_i^t = v_i^{t-1} + d_i^t * f_i \tag{9}$$

, and location x_i^t is updated same as equation (3). Here is the Algorithm flow on global minimum optimization.

- STEP1: Initialize population of bats
  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
  Generate new solutions x_i^t based on equation (3)(8)(9).

- STEP3: In local search phase, Generate a new solution around solutions x_i
  In case of a random distribution higher than parse rate r_i, generate a new solution x_{local} around x_i.

- STEP4: Generate a new solution randomly
  Generate a new solution x_{rnd} by random walk of bat.

- STEP5: Rank and update solutions
  In case of $rand < A_i$, choose the best from all solutions which are x_i, x_{local}, and x_{rnd}, and cross over as personal best solution unless it is higher than the value of former iteration.

- STEP6: Loop to STEP2

The NSBA pseudo code is described in Algorithm 2.

Algorithm 2 Bat Algorithm for Novelty

Require: *Objective Function* $f(x)$
Ensure: Initialize Population $x_i (i = 1, 2, ..., n)$ and v_i
Define frequency f_i at location x_i
Initialize pulse rates r_i, and the loudness A_i

```
while (t < Max number of iterations) do
  for i=1 to n do
    Generate new solutions x_i by tuning frequency f_i
    Update location x_i and velocity v_i [eqs.(1)(3)(8)(9)]
    if (rand > r_i) then
      Generate a new solution x_local around the solution x_i [eq.(4)]
    else
      Continue
    end if
    Generate a new solution x_rnd randomly
    if (rand < A_i & f(x_i) < f(x_i)) then
      Accept the new solution,
      and update pulse rate r_i
      & the loudness A_i [eqs. (5)(6)]
    end if
  end for
  Evaluate the all bats and select a best solution x_i* in the current solutions
end while
```

In this paper, the algorithm is implemented on Matlab for a benchmark function.

3.3 Comparison with Nearest Neighbor Bat Algorithm

To validate proposed NSBA, we have tested another modified Bat Algorithm for Nearest Neighbor search (NNBA). The updated equation as below

$$d_i^t = x_i^t - x_j^t \tag{10}$$

, which d_i^t is the distance of nearest neighbor between x_i^t and x_j^t. NNBA also uses Algorithm 2, where equation (8) is replaced as equation (10).

4 Multi-Objective Function

As an example to demonstrate the bat motion of this algorithm, we use Griewank function as below (shows Fig. 1)

$$f(x) = \sum_{i=1}^{d} \frac{x_i}{4000} - \prod_{i=1}^{d} \cos(\frac{x_i}{\sqrt{i}}) + 1 \tag{11}$$

, where global optimum is $f(x_*) = 0$, at $x_* = [0\ 0]$. Local minima 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,...,d. The function $f(x)$ has global minimum $f(x)_{gbest} = 0$ and also the other local minima $f(x)_{pbest} = 0$ for $d = 2$.

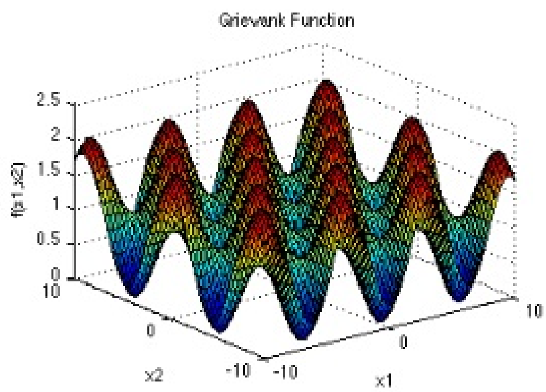

Fig. 1 Griewank function

5 Experiment

5.1 Evaluation Criteria

In this experiment, we focus on how many found local minima, and $\triangle - dist$ which total amount of the distance between local minima and solutions of nearest neighbor. We compare with the performance of these algorithms in term of the population size and the bat behavior by iteration.

5.2 Experimental Parameter

All experiments use same parameters, where population size of 20, frequency $f_{max} = 2, f_{min} = 0$, the loudness $A^0 = 1$, parse rate $r^0 \in [0\ 1]$ with $\alpha = \gamma = 0.9$, and $\delta = 1.2$.

5.3 Comparison with the other Algorithms

$\triangle - dist$ is total amount of the distance between local minima and nearest neighbor population, in case of initializing population randomly each algorithm. We compared proposed NSBA with the other algorithms, which NNBA and Original BA. Each algorithm was run for 10 seeds to validate the performance of NSBA.

6 Result

From Table 1, NSBA performed better than the other algorithms. It means the performance of distribution is very powerful. NNBA was a bit larger distance than NSBA. On the other hand, original BA is highest numerical value of three algorithms due to most of population tend to congregate to global best solution, as wee can see Fig. 2. Besides, the number of reached local minima of NSBA with 9.6 from 17 local minima (53.53%), is nearly the same numrical value as NNBA (56.47%).

Table 1 $\triangle - dist$ of basic BA, NNBA, and NSBA (n=20)

Seed	$basicBA$	$NNBA$	$NSBA$
1	117.4624	40.80115	34.73127
2	157.3706	34.486130	36.39555
3	145.3405	36.45273	39.05099
4	87.46077	54.79600	23.77053
5	157.0779	52.76253	34.5350
6	214.1643	58.38884	39.75885
7	157.3706	48.75871	43.40263
8	99.27593	46.05835	39.37115
9	156.868	33.65436	39.14996
10	124.6715	33.78423	29.0620
Average	141.706	43.9943	35.9228
SD	36.27235	9.406605	5.783791

Table 2 the number of reached local minima

	$basicBA$	$NNBA$	$NSBA$
average	1.7	9.6	9.1
search ratio	10.0 %	56.47 %	53.53 %
SD	1.05935	1.429841	0.875595

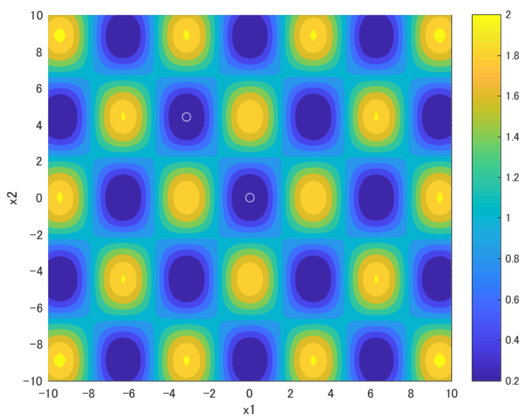

Fig. 2 Distribution of Original BA

7 Discussion

7.1 Different Population Size

We can see $\triangle - dist$ was directly proportional to increase population size from Table. 3, 4. Likewise SD gradually grew between $n = 10$ to 40. $\triangle - dist$ was a significant rise from $n = 20$ to 40.

7.2 Bat Behavior by Iteration

After initialized population, two algorithms of NNBA and NSBA remains from 20 to 40 $\triangle - dist$ by iterations. Especially, original BA is a sharply rise until 100 iterations from Fig. 5. From 800 iteration, $\triangle - dist$ of all three algorithms became stable.

8 Conclusion

We validated the performance of proposed Bat Algorithm based on Novelty Search, in comparison to original Bat Algorithm. As a result, NSBA is nearly same performance as NNBA, better than original BA for searching multiple local minima. As population size of bat increases, the number of searched local minima also increased. Our future prospects are adapting this algorithm for the other benchmark functions, and blushing up the perfor-

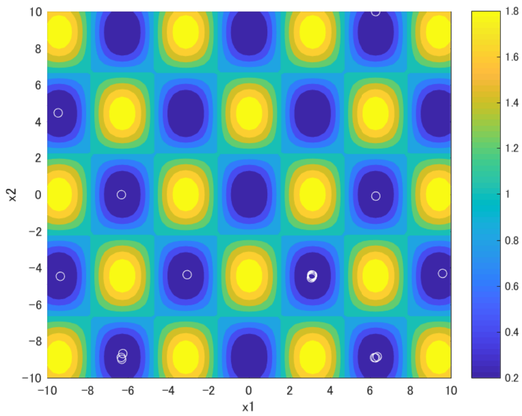

Fig. 3　Distribution of NNBA

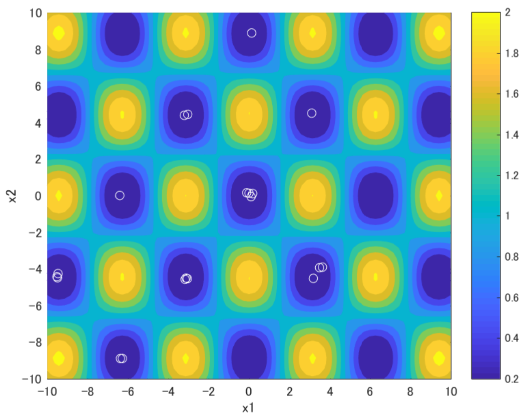

Fig. 4　Distribution of NSBA

mance to cover unspecified large number of local minima.

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, Pis-cataway, 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. Gonzales et al.), Studies in Computational Intelligence, Springer Berlin, 284, Springer, 65-74 (2010)

Table 3　$\triangle - dist$ of NSBA

Seed	$n = 10$	$n = 20$	$n = 40$
1	57.89055	34.73127	36.01941
2	74.33342	36.39555	17.5624
3	58.77926	39.05099	28.25238
4	55.87579	23.77053	17.093489
5	62.14158	34.53509	17.76876
6	49.07995	39.75885	22.30472
7	53.441226	43.40263	12.50444
8	65.24583	39.37115	28.42639
9	65.17686	39.149967	22.16795
10	64.97366	29.06209	22.04256
Average	60.6938	35.9228	22.4143
SD	7.22257	5.783791	6.889419

Table 4　the number of reached local minima

	$n = 10$	$n = 20$	$n = 40$
average	6	9.1	12.7
search ratio	35.30 %	53.53 %	74.71 %
SD	1.247219	0.875595	1.766981

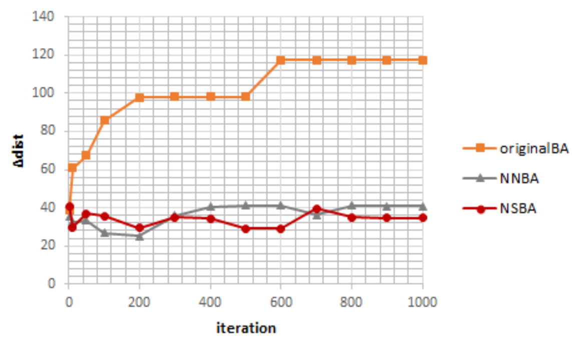

Fig. 5　$\triangle - dist$ of bats motion by iteration