



# Bat Algorithm with Adaptive Speed

Siqing You<sup>1(✉)</sup>, Dongjie Zhao<sup>1</sup>, Hongjie Liu<sup>2</sup>, and Fei Xue<sup>1</sup>

<sup>1</sup> School of Information, Beijing Wuzi University, Beijing 101149, China  
93028603@qq.com

<sup>2</sup> Beijing Advanced Innovation Center for Future Internet Technology, Beijing  
100124, China

**Abstract.** As a famous heuristic algorithm, bat algorithm (BA) simulates the behavior of bat echolocation, which has simple model, fast convergence and distributed characteristics. But it also has some defects like slow convergence and low optimizing accuracy. Facing the shortages above, an optimization bat algorithm based on adaptive speed strategy is proposed. This improved algorithm can simulate the bat in the process of search based on adaptive value size and adaptive speed adjustment. His approach can improve the optimization efficiency and accuracy. Experimental results on CEC2013 test benchmarks show that our proposal has better global searchability and a faster convergence speed, and can effectively overcome the problem convergence.

**Keywords:** Heuristic optimization algorithm · Bat algorithm · Convergence · Adaptive speed

## 1 Introduction

In order to solve complex computational problems in engineering optimization, computational science and other fields, many heuristic optimization algorithms [1–3] are proposed. According to the theory of the no free lunch theorem (NFL), it is shown that all optimization algorithms can not deal with all optimization problems [4, 5]. Therefore, researchers continue to propose various new heuristic intelligent algorithms [6–8].

In 2010, Yang proposed BAT algorithm based on bat echolocation behavior, which has good parallelism and robustness [9, 10]. Bat algorithm has excellent performance, but there are still some problems such as falling into local optimum and low accuracy. A new bat algorithm with adaptive speed strategy is proposed in this paper. The strategy can effectively prevent the algorithm from falling into local optimum and improve the searchability. The CEC2013 test function [11] is used to test the proposed algorithm, which shows that the proposed algorithm is more efficient than the other three comparison algorithms.

Other parts of this paper: Part 2 outlines the basic bat algorithm. Part 3 introduces the adaptive speed strategy bat optimization algorithm proposed in this paper. Section 4 gives the experimental conclusion. The last is the conclusion of the paper.

## 2 Standard Bat Algorithm

In order to make use of the bionic principle of bat echolocation, three idealized rules need to be assumed in the basic bat algorithm [10]:

- (1) All bats use echolocation to sense distances and distinguish between prey and background obstacles.
- (2) Each bat flies randomly at position  $\vec{x}_i(t)$  at velocity  $\vec{v}_i(t)$ , searching for prey at fixed frequency  $f_i$ , loudness  $A_i(t)$ , and rate of emission pulse  $r_i(t)$ .
- (3) The loudness changes from the maximum to the fixed minimum as the number of searches increases (from  $A_{\max}$  to  $A_{\min}$ ).

For bat  $i$  at time  $t + 1$ , the velocity  $\vec{v}_i(t + 1)$  and position  $\vec{x}_i(t + 1)$  show as follow:

$$\vec{x}_i(t + 1) = \vec{x}_i(t) + \vec{v}_i(t + 1) \tag{1}$$

$$\vec{v}_i(t + 1) = \vec{v}_i(t) + (\vec{x}_i(t) - \vec{p}_i(t)) * f_i \tag{2}$$

$$f_i = f_{\min} + (f_{\max} - f_{\min}) * \beta \tag{3}$$

where  $\vec{P}(t)$  is the global optimum position of time  $t$ , and  $\beta \in [0, 1]$  is a random following the normal distribution.

The new location update formula of bat algorithm is as follows:

$$\vec{x}_i(t) = \vec{p}_i(t) + \varepsilon A_{\text{mean}}(t) \tag{4}$$

where  $\varepsilon \in [-1, 1]$  is a random number and  $A_{\text{mean}}(t)$  is equal to the average of all loudness at time  $t$ .

With the bats approach their target, the loudness  $A$  decreases and  $r$  increases. They are following the formulas:

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

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

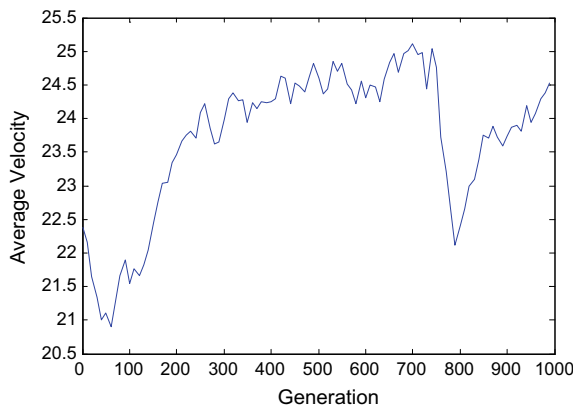
where  $r_i(0)$  is the initial value of pulse rate,  $\gamma$  and  $\alpha$  are constant.

### 3 Bat Algorithm with Adaptive Speed

Bat individual will inherit the previous generation of speed, and the best individual flight departing. When the algorithm progresses to the later stage, most bat individuals fall into local optimization, and the bat's speed cannot make it escape local optimization. To solve this problem, we introduced the concept of average speed.

$$\bar{v}^t = \sum_{i=1}^n v_i^t \tag{7}$$

which  $\bar{v}^t$  represents the first generation of the average speed of all the bat,  $n = 100$  as the number of the population, the standard algorithm bat sphere function as an example of the change in average speed will be shown in Fig. 1.



**Fig. 1.** Bat average speed change map algorithm

In view of this shortcoming, we propose an adaptive algorithm bat speed. This particular improvement is based on the fitness of each individual (optimal distance from the closer adaptation higher the value, the lower the farther away) its speed plus a weight in accordance with the right to a strategy. Obviously, if the individual is closer to the optimal solution than others, its adaptation value is higher. We should slow down the individual rapidly, so that the individual converges to the optimal solution. In this case, we give a smaller weight value. On the contrary, if the individual's adaptation value is very low, it means that the individual is far from the optimal solution. We need to give it a larger weight value.

According to the above analysis, weight calculation method is designed as follows:

$$w_i^t = \frac{e_i^{\text{score}_i^t}}{e_i^{\text{score}_1} + e_i^{\text{score}_2} + \dots + e_i^{\text{score}_n}} \tag{8}$$

where the symbol  $e_i^{\text{score}_i(t)}$  is defined as:

$$\text{score}_i(t) = \begin{cases} 1, & \text{if } (f_{\text{worst}} = f_{\text{best}}) \\ \frac{f_{\text{worst}} - f_i}{f_{\text{worst}} - f_{\text{best}}}, & \text{otherwise} \end{cases} \tag{9}$$

### 4 Simulation Results

To test the performance of ASBA, the CEC2013 test functions [11] are employed, witch is a set of 28 shifted and rotated benchmark problems. In the laboratory, the parameter  $\alpha = 0.95$ , and the parameter  $\gamma = 0.9$ . We set the population size of all algorithms equal to 100 and the dimensionality equal to 30. The algorithms run 30 times independently in the test. The stop condition for all algorithms is that the number of functions evaluated reaches a maximum of 50 dimensions.

Comparison results of the mean error at CEC2013 test functions with BA are presented in Table 1. In this table,  $w$  means that ASBA better than BA on  $w$  functions,  $l$  mean ASBA worse on  $l$  functions,  $t$  means that they are tied. From the results, ASBA performs better than BA on 16 functions, while BA outperforms ASBA on 12 functions.

**Table 1.** Comparison of the CEC 2013 benchmark problems

Fun	BA	ASBA	Fun	BA	ASBA
1	1.96E+00	1.37E+00	16	2.16E+00	1.98E+00
2	3.69E+06	1.07E+06	17	8.92E+02	4.11E+02
3	3.44E+08	3.70E+08	18	9.44E+02	3.50E+02
4	3.20E+04	1.43E+03	19	6.07E+01	2.54E+01
5	5.86E-01	3.03E-01	20	1.44E+01	1.42E+01
6	5.63E+01	5.00E+01	21	3.38E+02	3.13E+02
7	2.16E+02	1.77E+02	22	5.94E+03	5.88E+03
8	2.09E+01	2.09E+01	23	5.77E+03	5.99E+03
9	3.57E+01	3.52E+01	24	3.15E+02	3.22E+02
10	1.32E+00	1.12E+00	25	3.49E+02	3.53E+02
11	4.07E+02	5.13E+02	26	2.00E+02	2.30E+02
12	4.06E+02	5.35E+02	27	1.28E+03	1.28E+03
13	4.37E+02	4.92E+02	28	3.42E+03	4.22E+03
14	4.78E+03	4.82E+03	w/t/l	16\12\0	
15	4.89E+03	4.93E+03			

**Table 2.** Friedman test

Algorithm	Ranking
ASBA	1.43
BA	1.57

**Table 3.** Wilcoxon test

Algorithm	$p$ -values
ASBA versus BA	0.946

Table 2 presents the Friedman test results, as we can see, the ranks donate that the performance of ASBA is better than BA. Table 3 is the  $p$ -value achieved by Wilcoxon test; this can show that our proposal has better searchability.

## 5 Conclusion

This paper proposed an improved bat algorithm with adaptive speed strategy, which can simulate the bat in the process of search based on adaptive speed adjustment. This approach can improve the optimization efficiency and accuracy by providing a balance between exploitation and exploration. Experimental results on CEC2013 test benchmarks show that our proposal has better global searchability and a faster convergence speed, and can effectively overcome the problem convergence. Furthermore, research topics include some applications.

**Acknowledgments.** This work was supported by Beijing Key Laboratory (No: BZ0211) and Beijing Intelligent Logistics System Collaborative Innovation Center.

## References

1. Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: New York: IEEE; 1995. p. 39–43.
2. Holland JH. Genetic algorithms and the optimal allocation of trials. *SIAM J Comput.* 1973;2(2):88–105.
3. Yang X. Nature-inspired metaheuristic algorithms. Luniver press, 2010.
4. Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE Trans Evol Comput.* 1997;1(1):67–82.
5. Ho Y, Pepyne DL. Simple explanation of the no-free-lunch theorem and its implications. *J Optim Theory Appl.* 2002;115(3):549–70.
6. Dorigo M, Birattari M, Stutzle T. Ant colony optimization. *IEEE Comput Intell Mag.* 2006;1(4):28–39.
7. Karaboga D, Akay B. A survey: Algorithms simulating bee swarm intelligence. *Artif Intell Rev.* 2009;31(1–4):61–85.
8. Whitley D. A genetic algorithm tutorial. *Stat Comput.* 1994;4(2):65–85.

9. Yang X, Hossein Gandomi A. Bat algorithm: A novel approach for global engineering optimization. *Eng Comput.* 2012;29(5):464–83.
10. Yang X. A new metaheuristic bat-inspired algorithm. In: *Nature inspired cooperative strategies for optimization (NICSO 2010)*, 2010. p. 65–74.
11. Liang JJ, Qu BY, Suganthan PN, et al. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report.* 2013, 201212: 3–18.