ATLANTIS PRESS

Advances in Engineering Research, volume 141

5th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering (ICMMCCE 2017)

An improved self-adaptive bat algorithm

Shilei Lyu[1,a], Yonglin Huang[2,b], Zhen Li[*,3,c] and Yueju Xue[4,d]

[1] College of Electronic Engineering, South China Agricultural University & Guangdong Engineering Research Center for Monitoring Agricultural Information, Guangzhou 510642, China

[2] College of Electronic Engineering, South China Agricultural University, Guangzhou 510642, China

[3] College of Electronic Engineering, South China Agricultural University, Guangzhou 510642, China

[4] College of Electronic Engineering, South China Agricultural University, Guangzhou 510642, China

[a]lvshilei@scau.edu.cn, [b]1178509796@qq.com, [c]lizhen@scau.edu.cn, [d]xueyueju@163.com

Keywords: Bat algorithm; Self-adaptive; Step-controlled mechanism; Mutation mechanism

Abstract: Considering the problems of low solution precision and suffering from pre-mature convergence by the basic bat algorithm, an improved self-adaptive bat algorithm (SABA) is proposed after the introduction of the self-adaptive step-controlled mechanism and mutation mechanism in this paper. The results of the proposed algorithm based on 6 test functions indicate that compared with the particle swarm algorithm and the basic bat algorithm, the SABA algorithm effectively avoid falling into the local optimum in the early stage and had the higher solution accuracy.

1. Introduction

BA algorithm has strong robustness and fast convergence rate, but it also has the disadvantage of being easy to fall into local optimum and low convergence precision, which restrict the application of BA algorithm in engineering field. Xiao proposed an improved BA algorithm based on differential evolution in 2014 [1]. In order to solve the engineering planning problems, Wang presented a hybrid BA algorithm in 2015 [2]. Gai-Ge Wang solved the three-dimensional path planning problems by using an improved BA algorithm in 2016 [3]. For improving the accuracy of BA algorithm, an improved self-adaptive bat algorithm (SABA) is proposed after the introduction of the step-controlled mechanism and mutation mechanism.

2. Improved self-adaptive bat algorithm

2.1 Step-controlled mechanism. Based on the basic bat algorithm, the inertia weight coefficient is added into the velocity item:

$$\boldsymbol{v_i^t} = \boldsymbol{\omega v_i^{t-1}} + \boldsymbol{f_1 r_1}(\boldsymbol{h_*} - \boldsymbol{x_i^{t-1}}) + \boldsymbol{f_2 r_2}(\boldsymbol{x_*} - \boldsymbol{x_i^{t-1}}) \tag{1}$$

$$\boldsymbol{f_1} = \boldsymbol{\alpha}\left(\mathbf{1} - \boldsymbol{e}^{-|\boldsymbol{F_{avg}} - \boldsymbol{F_{best}}|}\right) + \boldsymbol{\gamma}\left(\mathbf{1} - \frac{\boldsymbol{t}}{\boldsymbol{t_{max}}}\right) + \boldsymbol{f_{min}} \tag{2}$$

$$\boldsymbol{C_w} = \boldsymbol{f_1} + \boldsymbol{f_2} \tag{3}$$

where h_* is the optimal solution of the current individual bat; x_* is the optimal solution for the current bat group; f_1 and f_2 are pulse frequencies; r_1 and r_2 are uniformly random numbers within the range of (0.5, 1.5); F_{avg} is the average fitness of all the individuals in bat group; F_{best} is the optimal individual in bat group; f_{min} is a constant, which is used to set and control the minimum value of f_1. In the paper, the linearly decreasing weight is used to endow the algorithm with the strong global search capability.

The step weight coefficient μ is added to restrain the iterative step size of an individual bat i:

$$x_i^t = x_i^{t-1} + \mu\, v_i^t \tag{4}$$

where μ is in the range of (0,1].

In this paper, a local search strategy is proposed and described as follows:

First, a random number is set as $\beta \in [0,1]$. If β is less than the pulse emission frequency R, then the individual bat performs the local search. The local search strategy is described as follows:

When the evaluation index $k<0.4$, the position vector of an individual bat i is updated as follows:

$$x_i^t = x_* + A * s * g(k) * \delta \tag{5}$$

When $k \geq 0.4$, the position vector of an individual bat i is updated as follows:

$$x_i^t = x_* + A * s * 0.1^{g(k)} * \delta \tag{6}$$

where the evaluation index is $k=t/t_{max}$ (t_{max} is the maximum number of iterations and $k \in (0,1]$); x_* is the optimal solution of the bat group; A is the average loudness of the current bat group; s is the ratio of the distance between the upper and lower boundaries of the feasible solution domain to the number of bat group and can adapt the local search step size to the scale of the problem to be solved; $\delta \in [-1,1]$ is a random vector. The function $g(k)$ is updated as follows:

$$g(k) = \begin{cases} 2, & k \leq 0.1 \\ 1.5, & 0.1 < k \leq 0.2 \\ 1, & 0.2 < k \leq 0.3 \\ 0.5, & 0.3 < k \leq 0.4 \\ 1, & 0.4 < k \leq 0.6 \\ 3, & 0.6 < k \leq 0.7 \\ 5, & 0.7 < k \leq 0.8 \\ 7, & 0.8 < k \leq 0.9 \\ 9, & 0.9 < k \end{cases} \tag{7}$$

2.2 Mutation mechanism. The step-controlled mechanism can effectively improve the global search capability of the algorithm, but it cannot ensure the global optimal solution. Moreover, in the later stage of the algorithm, f_1 decreases and the global search capability is reduced. Therefore, the algorithm still has the probability to fall into local optimum. In this paper, the loudness A and the emission frequency R are restricted, so that the SABA algorithm can perform the mutation operation in the later stage, thus reducing the probability that the algorithm falls into a local optimum. The pulse loudness A and the emission frequency R are updated as follows:

$$A_i^{t+1} = \frac{f_1}{f_{max}} \tag{8}$$

$$R_i^{t+1} = \frac{f_2}{f_{max}} \tag{9}$$

where f_{max} is the upper limit of the pulse frequency. Pulse frequencies f_1 and f_2 depends on the number of current iterations and the average fitness of the group, so the loudness A and the emission frequency R can be adaptively changed with the optimum search process of the algorithm.

Through the analysis of the behavior characteristics of bats, this paper presents a mutation mechanism combined with the pulse loudness A of bats. The mechanism is summarized as follows:

A random number $\beta_1 \in [0,1]$ is generated. If β_1 is less than A and the individual bat does not perform local search, then a random number $\beta_2 \in [0,1]$ is generated. If β_2 is greater than the mutation threshold $\rho \in (0,1)$, then the individual bat is randomly reset.

SABA algorithm flowchart is shown in Fig. 1.

3. Test results and analysis

3.1 Test environment. The control algorithms include the PSO algorithm [4], the basic BA algorithm [5], and the IBA algorithm [6]. The parameters are consistent with the original reports. The group sizes of all algorithms are 40; the maximum number of iterations is 500; independent runs are 50. The calculation results based on 30*D* are shown in Table 1. The iterative curves are shown in Fig. 2.

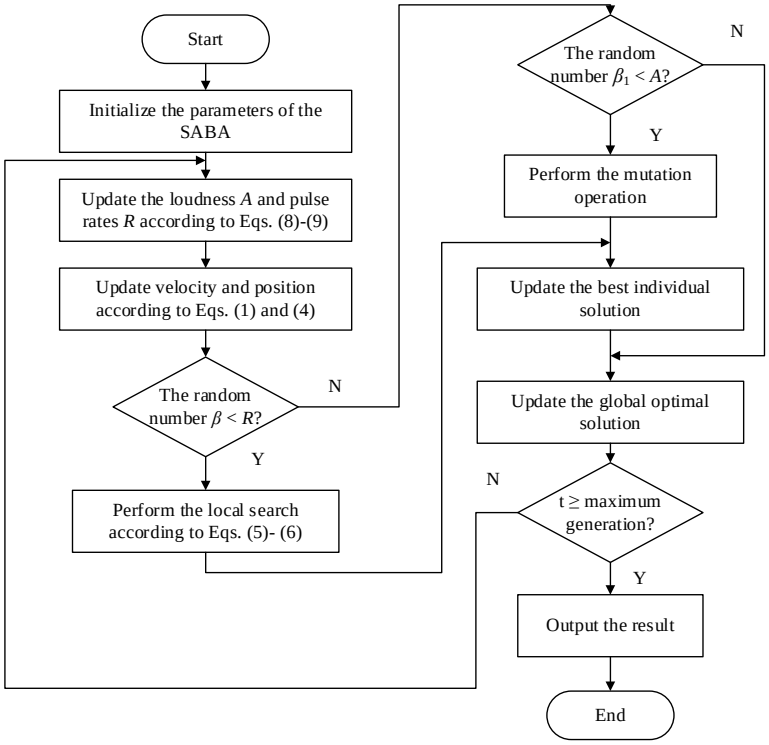

Fig. 1. Flowchart of SABA algorithm

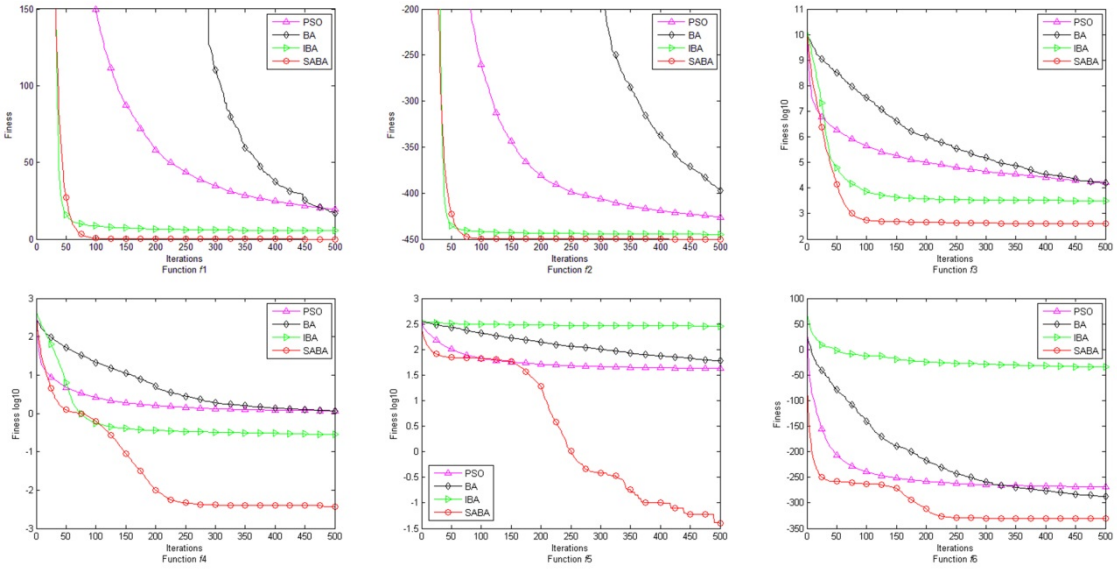

Fig. 2. Test results (30*D*)

3.2 Result analysis. According to the data in Table 1, in the optimum search performance, the basic BA algorithm shows the slightly worse solution precision for the unimodal functions (f_1-f_2) than the PSO algorithm. In multimodal functions, the overall performance of the basic BA algorithm is better than that of the PSO algorithm. The IBA algorithm has good test results in the multimodal functions.

Compared with the above-mentioned control algorithms, the SABA algorithm shows the better test results. Moreover, the standard variances of various test functions indicate that the SABA algorithm has the good robustness.

Table 1 Calculation results (30*D*) of various algorithms

Functions		PSO	BA	IBA	SABA
Sphere (f_1)	best	4.9437	0.5440	3.5608	2.0815e-19
	worst	52.7758	96.5196	6.6412	4.9257e-18
	mean	19.4191	16.9526	5.4623	1.5382e-18
	std	11.3091	16.3720	0.6466	1.0966e-18
Shifted Sphere (f_2)	best	-445.5051	-444.7815	-447.0343	-450.0000
	worst	-404.6207	-191.0373	-443.8029	-450.0000
	mean	-426.1541	-396.9489	-444.7664	-450.0000
	std	9.1465	44.7184	0.7164	-1.3466e-13
Shifted Rosenbrock (f_3)	best	2.3813e+03	442.5229	986.2928	390.0000
	worst	5.2670e+04	1.4042e+05	1.2093e+04	500.0399
	mean	1.7078e+04	1.5326e+04	3.1965e+03	418.6112
	std	1.0837e+04	2.7361e+04	3.2924e+03	23.2614
Griewank (f_4)	best	1.0695	0.4405	0.1564	4.4409e-16
	worst	1.3641	1.7464	0.3722	0.0517
	mean	1.1697	1.1660	0.2830	0.0070
	std	0.0760	0.1916	0.0377	0.0118
Rastrigin (f_5)	best	26.3531	1.0680	228.9885	1.4211e-14
	worst	67.5933	170.0479	350.0273	0.9950
	mean	42.7317	59.5188	287.2139	0.0399
	std	10.4599	48.1039	30.1155	0.1969
Shifted Rastrigin (f_6)	best	-300.8787	-329.9551	-105.3721	-330.0000
	worst	-235.8998	-144.2777	21.5326	-328.0101
	mean	-268.9089	-287.8379	-34.5640	-329.8010
	std	13.8519	36.6534	27.5158	0.4495

4. Conclusions

In the SABA algorithm, the moving step size of a bat can be adjusted according to the feasible solution domain and the current iterative progress and the solution accuracy of the algorithm is improved. The test results of various test functions show that the SABA algorithm has the better performance. In the future, we will verify the engineering applications of the SABA algorithm.

Acknowledgements

This work was financially supported by the National Natural Science Foundation of China (Grant Nos. 61601189 and 61602187), the Special Fund of Modern Technology System of Agricultural Industry (Grant No. CARS-27), the Science and Technology Planning Project of Guangdong Province, China (Grant Nos. 2014A020208108, 2015A020209161, 2016A020210088 and 2016A020210093), and the Science and Technology Program of Guangzhou, China (Grant No. 201605030013).

References

[1] Xiao H H, Duan Y M. Research and Application of Improved Bat Algorithm Based on DE Algorithm[J]. Computer Simulation, 2014(1):62-65.
[2] Jinfeng Wang. A Hybrid Bat Algorithm for Process Planning Problem[J]. IFAC-PapersOnLine 48-3 (2015) 1708–1713.
[3] Gai-Ge Wang, HaiCheng Eric Chu, Seyedali Mirjalili. Three-dimensional path planning for UCAV using an improved bat algorithm[J]. Aerospace Science and Technology 49 (2016) 231–238.
[4] J. Kennedy, and R. Eberhart. Particle swarm optimization[J]. IEEE International Conference on Neural Networks, 1995. Proceedings, Year Published, pp. 1942-1948.

[5] X.S. Yang. A New Metaheuristic Bat-Inspired Algorithm[J]. Computer Knowledge and Technology, vol. 284, 2010, pp. 65-74.
[6] Wang G G, Lu M, Zhao X J. An improved bat algorithm with variable neighborhood search for global optimization[C]. IEEE Congress on Evolutionary Computation. IEEE, 2016:1773-1778.