



Chaos-Based Second-Order BAS For Multimodal Function Optimization

Xiaoxiao Li^a, Xuefeng Zhou^{*a}, Zhihao Xu^a, Guanrong Tang^a

^aGuangdong Key Laboratory of Modern Control Technology, Guangdong Institute of Intelligent Manufacturing, Guangzhou, Guangdong, 510070, P. R. China

Abstract. Aiming at solving a drawback of the second-order beetle antenna search (SOBAS), a variant of the beetle antenna search (BAS), that it is difficult to find the global optimal solution and the low convergence accuracy when applied to the multimodal optimization functions **with high dimension or large variable region**, a chaotic-based second-order BAS algorithm (CSOBAS) is proposed by introducing chaos theory into the SOBAS. The algorithm mainly has three innovations: 1) chaos initialization: choosing the one with the smallest fitness function value from twenty beetles with different positions for iterative search; 2) using chaotic map to tune the randomization parameter in the detection rule; 3) imposing a chaotic perturbation on the current beetle to hope to help the search to jump out the local optimal solution. Eight different chaotic maps are used to demonstrate their impact on the simulation results. With six typical multimodal functions, performance comparisons between the CSOBAS and the SOBAS are conducted, validating the effectiveness of the CSOBAS and its superiority compared to the SOBAS. What's more, the CSOBAS with an appropriate chaotic map can achieve a very good convergence quality compared to other swarm intelligence optimization algorithms while maintaining an individual.

1. Introduction

The beetle antenna search (BAS) [1] is a meta-heuristic algorithm proposed by Jiang et al. in 2017 based on the foraging principle of the beetle, and requires only one individual compared to the Particle swarm optimization (PSO) that is one of the state-of-the-art algorithms, with properties of simple calculation, fast convergence speed, and few parameters that needs to be adjusted. The BAS algorithm has been extensively investigated and used in several applications, and many related variants also have been yielded to improve its convergence quality [2–8]. For instance, in [2], a variant of the BAS method (BAS-WPT) that can handle multi-objective function and have no requirement for tuning parameters was proposed, which used a static penalty function to exploit infeasible solutions for the constraint optimization problem. A combination of BAS and PSO, called BSO, was developed and was tested on 23 benchmark functions to verify its performance in [6]. Combining swarm intelligence algorithm with feedback-based step-size update strategy, the Beetle Swarm Antennae Search Algorithm (BSAS) was developed in [7].

2010 Mathematics Subject Classification. 78M32

Keywords. Second-order BAS, chaos theory, **chaos maps**, **chaos perturbation**, multimodal function

Received: 28 October 2018; Revised: 20 November 2018; Accepted: 29 November 2018

Communicated by Shuai Li

This work is supported by the National Natural Science Foundation of China (No. 51405091) and Guangdong Special Support Plan (2016TQ03X463) and Natural Science Foundation of Guangdong Province (2016A030310310) and Guangzhou Science Research Project (201707010457).

Email addresses: sandraxxli@163.com (Xiaoxiao Li), xuefengzhou@vip.qq.com (Xuefeng Zhou*), zh.xu@gia.ac.cn (Zhihao Xu), gr.tang@gia.ac.cn (Guanrong Tang)

While overcoming the disadvantages of BAS with low convergence accuracy and easily falling into the local optimum, both BSO and BSAS employ swarm intelligence concepts, i.e, using m beetles or particles instead of one individual, with high computational complexity. In the original BAS, the detecting rule is equivalent to updating the speed of the beetle, which has lower convergence accuracy, and even the finding of global solution fails for both high-dimensional optimization problems (such as $D = 30$) and large variable region. Therefore, a new detecting rule (or say the new velocity and new position) is given in this paper, which is equivalent to creating a driving force between difference of the two antennas. Thus we call the improved BAS as second-order BAS (SOBAS), and the original one as the first-order BAS (FOBAS).

Compared with the FOBAS, we found in the experimental simulation via MATLAB that the SOBAS has several advantages of high convergence quality and high success rate of finding a solution with certain accuracy as a success condition, however, at the expense of taking more time to complete calculating. To the best of our knowledge, a variety of optimization algorithms have been successfully employed to some practical applications [9–13], such as robot motion planning [9, 10], image processing [12] and so on. What's more, as we mentioned earlier, the BAS algorithm has also been found their applications in ship course control [5], energy management in microgrid [3] and wireless sensor network [4] and so on. Therefore, we believe that the SOBAS also has the same potentiality to be applied to these fields.

However, along with the iteration, the SOBAS steps into the low convergence accuracy when applied to the multimodal optimization functions, and is difficult to find the global optimal solution. Chaos is a relatively common nonlinear phenomenon existing in nonlinear systems. The characteristics of chaos mainly include pseudo-randomness, ergodicity and sensitivity to initial conditions. Since ergodicity can be used as an effective mechanism to avoid the search process to stepping into the local minimum, chaos theory has become a promising optimization tool [14, 15]. The effectiveness of chaos algorithm has already been tested and verified in several applications [16–18], such as image encryption [16, 17]. The fusion of chaos theory and meta-heuristic algorithm has also been successfully developed and achieved good results [19–24]. For example, in [19], chaos theory was embedded into Firefly Algorithm (FA), twelve different chaotic maps were used for tuning the attractive movement of the fireflies. In [20], an adaptive FA was developed and combined with chaos, increasing FA's global search ability. In [24], chaos was embedded into an accelerated PSO algorithm proposed in [25], where twelve different chaotic maps were used. Driven by them, a chaotic-based SOBAS algorithm (CSOBAS) is proposed by introducing chaos theory into the SOBAS. Applying to six typical multimodal functions, the results show that the effectiveness of the CSOBAS and its superiority compared to the SOBAS in terms of convergence accuracy and iteration numbers. What's more, the CSOBAS with an appropriate chaotic map can achieve a very good convergence quality compared to other swarm intelligence optimization algorithms. For comparison, in Table 1, we conclude a comparison between the CSOBAS, SOBAS and FOBAS algorithms.

Table 1: Comparison on between the CSOBAS, SOBAS and FOBAS algorithms.

Algorithms	Convergence quality	Success rate	Time complexity	Applicable type
FOBAS	low	low	low	Two dimension
SOBAS	high	high	high	Low and high dimension
CSOBAS	higher	higher	higher	Low and high dimension

The rest of this paper is organized as follows. Section 2 introduces the second-order BAS. The CSOBAS is introduced in Section 3 with eight chaotic maps. Then, Section 4 gives the simulation comparison between the CSOBAS and the SOBAS by applying to six multimodal optimization functions. The last chapter, i.e, Section 5, summarizes the paper.

2. The Second-Order BAS Algorithm

The general formula for multimodal optimization problems is formulated as:

$$\min f(x_1, x_2, \dots, x_D) \quad \text{s.t. } \alpha_i \leq x_i \leq b_i.$$

where $\mathbf{x} = [x_1, x_2, \dots, x_D]^T$ is a vector of decision variables, the superscript T stands for transpose. D denotes the dimension; α_i and b_i are the lower bound and upper bound of the i -th decision variables, respectively; $f(\mathbf{x})$ is then the objective function to be optimized.

A beetle relies on the long antennas on both left and right to detect the odour value around itself, then flies toward the larger odour value in the next step, looping until food is found. Therefore, the resulting BAS algorithm mainly is divided into two steps, i.e, detecting and searching. The searching rule (1) is used to explore,

$$\begin{cases} \mathbf{x}_l^t = \mathbf{x}^t - \frac{1}{2}d\vec{b}, \\ \mathbf{x}_r^t = \mathbf{x}^t + \frac{1}{2}d\vec{b}. \end{cases} \quad (1)$$

where \mathbf{x}_l^t and \mathbf{x}_r^t are the left and right antenna coordinate of the beetle at time t , respectively. d denotes the distance between two antennas for a beetle, \vec{b} is a normalized random unit vector to enhance the searching ability,

$$\vec{b} = \frac{\text{rand}(D, 1)}{e + \text{norm}(\text{rand}(D, 1))}.$$

The detecting rule is:

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \xi \vec{b} \text{sign}(f(\mathbf{x}_l^t) - f(\mathbf{x}_r^t)), \quad (2)$$

where $f(\cdot)$ is the function to be optimized. $\text{sign}(\cdot)$ is a symbolic function in MATLAB. ξ denotes step length of a beetle,

$$\xi = \lambda \xi,$$

where λ is a random number close to one.

Compared to the FOBAS, the SOBAS has mainly changed in two places. First, the detection rules (2) are replaced with:

$$\mathbf{v}^{t+1} = c_1 \mathbf{v}^t - c_2 \vec{b} (f(\mathbf{x}_l^t) - f(\mathbf{x}_r^t)), \quad (3)$$

$$\begin{aligned} \mathbf{v}^{t+1} &= V_{\max}(\mathbf{v}^{t+1} > V_{\max}) - V_{\max}(\mathbf{v}^{t+1} < -V_{\max}) \\ &+ \mathbf{v}^{t+1}(\mathbf{v}^{t+1} \leq V_{\max})(\mathbf{v}^{t+1} \geq -V_{\max}), \end{aligned} \quad (4)$$

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \mathbf{v}^{t+1}, \quad (5)$$

where $V_{\max} = c_1 \xi$. The parameters $c_1 \in (0, 1)$ and $c_2 \in (0, 1)$ are valued as 0.7 and 0.2 in this paper, respectively. For ξ , in each iteration, setting

$$\xi = \lambda(\xi - \xi_0) + \xi_0; \quad (6)$$

for step size with a minimal resolution, where $\lambda = 0.95$, ξ_0 is a constant and valued as 5×10^{-5} .

3. Chaos-Based SOBAS

3.1. Analysis on SOBAS Algorithm

- 1) Since the initialization process is random, when the initial solution of the beetle is far away from the optimal solution, plus that the number of iterations is a certain amount, there is no guarantee that the beetle will eventually step to the optimal solution. If the initial solution is better, it may help to improve the convergence quality of the SOBAS.
- 2) For the detecting rule (3), when the odor strength values of the left and right antenna around the beetle are equal, updating new speed and position will depend largely on the parameter c_1v , due to $c_1 \in (0, 1)$, this makes the SOBAS easily fall into the local optimal solution.

Therefore, by using the ergodicity of chaos theory, we first generate a large number of initial beetle swarms, and select a beetle with the best fitness function value to perform iterative search with the current position. We use chaotic maps to tune the parameter c_1 and impose a chaotic perturbation on position of the current beetle to jump out the dilemma of the poor convergence quality and the local minimum when the SOBAS is used for the multimodal optimization function.

3.2. The CSOBAS

The CSOBAS is now reformulated with more detail as follows:

1. Chaos initialization:

- (1) Randomly generating a D -dimensional vector $\mathbf{x}_1 = (x_{1,1}, \dots, x_{1,D})$ with each component value $x_{1,i} \in (0, 1), i = 1, 2, \dots, D$.
- (2) Mapping the decision variables $x_{1,i}$ to chaotic variables $cx_{1,i}$ based on equation (7):

$$cx_{1,i} = \frac{x_{1,i} - x_{\min,i}}{x_{\max,i} - x_{\min,i}}, \tag{7}$$

where $x_{\max,i}$ and $x_{\min,i}$ are the upper and lower bounds of the i -th decision variable, respectively.

- (3) According to the chaotic map, $cx_{i+1,j} = \mu cx_{i,j}$, (taking the Logistics equation as an example), $i = 1, 2, \dots, N, j = 1, 2, \dots, D - 1$, we obtain cx_1, cx_2, \dots, cx_N . Here, N denotes the number of the beetles.
- (4) Converting the chaotic variables $cx_i, i = 1, 2, \dots, N$, to decision variables \mathbf{x}_i based on (8)

$$\mathbf{x}_{i,j} = x_{\min,j} + (x_{\max,j} - x_{\min,j})cx_i. \tag{8}$$

- (5) Calculating the objective function value, and choosing the corresponding solution of the beetle having the smallest function value from the N beetles as an initial solution to start an iterative search.

2. Chaotic disturbance:

- (1) Based on the employed chaotic map, generating a D -dimensional vector $\mathbf{z}_1 = (z_{1,1}, \dots, z_{1,D})$, where

$$z_{1,i} = 4z_{0,i}(1 - z_{0,i}).$$

- (2) Converting each component of \mathbf{z}_1 into an allowable perturbation range $[-\beta, \beta]$ with a perturbation $\Delta \mathbf{x} = (\Delta x_1, \dots, \Delta x_D)$, where

$$\Delta x_i = -\beta + 2\beta z_{1,i}.$$

$$\mathbf{z}_0 = \mathbf{z}_1.$$

(3) Generating new position

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \mathbf{v}^{t+1},$$

and new position with a disturbance

$$\mathbf{x}_{\text{new}}^{t+1} = \mathbf{x}^t + \mathbf{v}^{t+1} + \Delta \mathbf{x}.$$

(3) Evaluating the corresponding fitness function values, if $f(\mathbf{x}_{\text{new}}^{t+1}) < f(\mathbf{x}^{t+1})$, then $\mathbf{x}^{t+1} = \mathbf{x}_{\text{new}}^{t+1}$.

The pseudo-code of the CSOBAS algorithm is shown in Algorithm 1.

Algorithm 1 Chaotic SOBAS.

Input: The objection function $f(\mathbf{x})$, $\mathbf{x} = [x_1, x_2, \dots, x_D]^T$, and the required parameters;

Chaos initialization;

Initialize speed of the beetle as zero, and $c_1 = 0.7$;

Initialize each element of vector $\mathbf{x}_{\text{best}} \in \mathbb{R}^m$ to \mathbf{x}^0 ;

Initialize the best objective function value, f_{best} , to $\check{f}(\mathbf{x}_{\text{best}})$;

Randomly generating a D -dimensional vector $\mathbf{z}_0 = (z_{0,1}, \dots, z_{0,D})$ with each component value $z_{0,i} \in (0, 1)$, $i = 1, 2, \dots, D$.

while $t < \text{iteration number}$ **do**

$d = \xi/5$;

Generate the direction vector unit \vec{b} ;

Search the location of two antennas for a beetle according to (1);

Map the variable c_1 to chaotic variable based on a chaotic map;

Generate the new position \mathbf{x}^t according to (5);

Impose a chaotic disturbance on the current position;

Update \mathbf{x}^t ;

if $f(\mathbf{x}^t) < f_{\text{best}}$ **then**

$\mathbf{x}_{\text{best}} = \mathbf{x}^t$

$f_{\text{best}} = f(\mathbf{x}^t)$

end if

Update step length ξ based on (6);

end while

Output: \mathbf{x}_{best} and the corresponding function value f_{best} .

3.3. Chaotic Map

Different chaotic maps have different enhancement capabilities for CSOBAS in multimodal function optimization. In this paper, we have chosen eight different chaotic maps [24] to show their impact on the optimization results.

1) Circle map

$$x_{i+1} = x_i + 0.2 - \frac{0.5}{2\pi} \sin(2\pi x_i) \quad \text{mod } (1),$$

where $\text{mod } (1)$ denotes the remainder of division of the number by 1.

2) Iterative map

$$x_{i+1} = \sin\left(\frac{\alpha\pi}{x_i}\right),$$

where $\alpha \in (0, 1)$. $\alpha = 0.7$ is used for the experiments.

3) Logistic map

$$x_{i+1} = \mu x_i(1 - x_i),$$

where $\mu = 4$.

4) Piecewise map

$$x_{i+1} = \begin{cases} \frac{x_i}{p} & 0 \leq x_i < p \\ \frac{x_i - p}{0.5 - p} & p \leq x_i < 0.5 \\ \frac{1 - p - x_i}{0.5 - p} & 0.5 \leq x_i < 1 - p \\ \frac{1 - x_i}{p}, & 1 - p \leq x_i < 1, \end{cases}$$

where $p \in [0, 0.5]$, $x \in (0, 1)$. p is valued as 0.2 in this paper.

5) Sine map

$$x_{i+1} = \frac{\alpha}{4} \sin(\pi x_i),$$

where $0 < \alpha \leq 4$, and $\alpha = 3.2$.

6) Singer map

$$x_{i+1} = \alpha(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.302875x_i^4),$$

where $\alpha \in [0.9, 1.08]$ and $\alpha = 1.02$.

7) Sinusoidal map

$$x_{i+1} = \alpha x_i^2 \sin(\pi x_i).$$

If $\alpha = 2.3$ and $x_0 = 0.7$, it can be rewritten as

$$x_{i+1} = \sin(\pi x_i).$$

8) Tent map

$$x_{i+1} = \begin{cases} \frac{x_i}{0.7} & x_i < 0.7 \\ \frac{10}{3}(1 - x_i), & x_i \geq 0.7 \end{cases}$$

4. Multimodal Function

The ability to search for multimodal functions has always been one of the bases for measuring the superiority of an optimization algorithm. In this study, we use six typical multi-modal functions to verify the effectiveness and superiority of the proposed SCOBAS algorithm compared to other optimization algorithms.

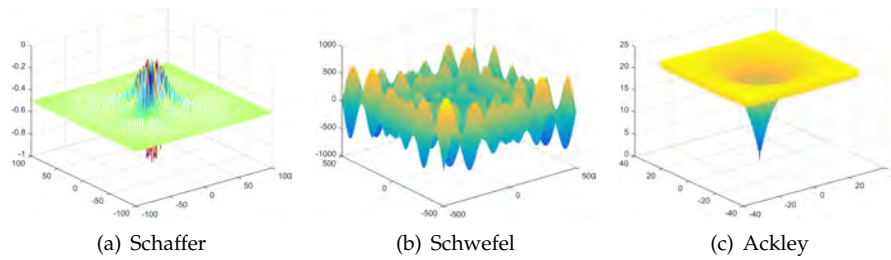


Figure 1: Typical 2D representations of multimodal functions;

4.1. Schaffer Function

$$f_1 = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{1 + 0.001(x_1^2 + x_2^2)^2} - 0.5, \quad -100 \leq x_i \leq 100.$$

The Schaffer function is a multimodal function with strong oscillations (as shown in Fig.1a), whose global minimum is -1 . Due to it has numerous local minima surrounding the optimal value, it is so difficult that the globally best value is found. Independently running CSOBAS and SOBAS (the iteration number are set as 10^3 and 10^4 , respectively.) each with 100 times, the acquired results are listed in Table 2. Among which, we can observe that convergence performance of the CSOBAS is better than the SOBAS, although it may not reach the minimum -1 due to only an individual involved in the evolution. Compared to the SOBAS, however, the convergence quality of the CSOBAS can be increased as the number of iterations increases. Table 2 also shows the optimization results of the FOBAS for the Schaffer function. It is not difficult to find that the SOBAS has no obvious advantage over the FOBAS. For better comparison, the optimal function value less than -0.9 is viewed as the success condition and under 1000 iteration, we give in Table 3 that the success rates of the CSOBAS, SOBAS and FOBAS by running these three algorithms separately 100 times. Taking the results of running 10 times, based on Table 3, the advantages of the CSOBAS compared to the other two algorithms can be easily found. Also, we found via MATLAB that the average time required for the FOBAS algorithm to execute once is approximately 8.4 seconds, while the SOBAS is around 10.5 seconds. For the CSOBAS, the computation time is increased to 13.37 seconds due to the fusion of chaos theory. Compared to the FOBAS, the second-order BAS and the CSOBAS are increased in terms of time complexity. However, the success rate of optimization and convergence quality are better than the first order. The above observations are combined, revealing the effectiveness of the improved algorithm combining chaos theory with second-order BAS. In addition, Table 4 shows that statistical results obtained by the CSOBAS under different chaotic maps for Schaffer function. We obtain that the sine map and singer map can achieve the best performance than others.

Table 2: Statistical results obtained by the CSOBAS and SOBAS for the Schaffer function.

	1000 iterations			10000 iterations		
	Best	Mean	Worst	Best	Mean	Worst
CSOBAS	-0.99801259	-0.87308159	-0.59286701	-0.99986608	-0.94348926	-0.94348926
SOBAS	-0.96351203	-0.57189222	-0.50134958	-0.98875641	-0.56831554	-0.50147538
FOBAS	-0.94996383	-0.51478505	-0.50000957	-0.95308699	-0.53733437	-0.50000776

Table 3: Success rate of the CSOBAS, SOBAS and FOBAS for the Schaffer function.

	Success number										Success rate
	1	2	3	4	5	6	7	8	9	10	
CSOBAS	51	47	43	45	43	44	51	39	37	54	100%
SOBAS	1	4	1	2	2	5	0	2	2	5	90%
FOBAS	0	1	1	1	2	6	0	3	2	0	70%

Table 4: Statistical results obtained by the CSOBAS under different chaotic maps for the Schaffer function.

	1000 iterations			10000 iterations		
	Best	Mean	Worst	Best	Mean	Worst
Circle map	-0.99801259	-0.89269717	-0.72219215	-0.99368911	-0.94382899	-0.81927744
Iterative map	-0.99026212	-0.85059894	-0.58324955	-0.99027576	-0.94073673	-0.82834493
Logistic map	-0.99801259	-0.87308159	-0.59286701	-0.99986608	-0.94348926	-0.94348926
Piecewise map	-0.99002903	-0.80427833	-0.65429050	-0.99560813	-0.82593855	-0.65429050
Sine map	-0.99572887	-0.94852500	-0.93791217	-0.9967109448	-0.96117373	-0.93999396
Singer map	-0.99801259	-0.93181641	-0.79468510	-0.99027066	-0.95176927	-0.84602778
Sinusoidal map	-0.99398597	-0.88524889	-0.62251305	-0.99026846	-0.95044441	-0.87241953
Tent map	-0.99470661	-0.81624003	-0.65429050	-0.99391891	-0.83530897	-0.65429050

4.2. Schwefel Function

$$f_2 = \sum_{i=1}^D (x_i \sin(\sqrt{|x_i|})), \quad -500 \leq x_i \leq 500.$$

Fig.1b shows typical 2D representations of the Schwefel function, whose global minimum is related to the dimension D , which is $-418.9829D$. Independently running CSOBAS and SOBAS (the iteration number is set as 10^3) each with 100 times, the acquired results are listed in Table 5. Statistical results obtained by the CSOBAS under different chaotic maps are shown in Table 6. Following them, it is clear that for the Schwefel function, the CSOBAS is significantly better than the SOBAS in terms of convergence quality in three dimensions. Especially, the CSOBAS with an appropriate chaotic mapping achieves global optimal solutions within tiny error. For the Schwefel function, the singer map and sinusoidal map achieve the best performance than others.

Table 5: Statistical results obtained by the CSOBAS and SOBAS for the Schwefel function.

	f_{min}	CSOBAS			SOBAS		
		Best	Mean	Worst	Best	Mean	Worst
d=2	-837.9658	-837.9657	-785.7350	-599.5720	-790.9040	-516.6586	-361.1783
d=10	-4189.829	-3331.1754	-2430.1700	-1805.8915	-2905.5422	-2321.9474	-1852.0266
d=30	-12569.487	-8138.5710	-6301.5662	-5415.6316	-6758.5250	-5609.1751	-4527.6119

Table 6: Statistical results obtained by the CSOBAS under different chaotic maps for the Schwefel function.

	D=10			D=30		
	Best	Mean	Worst	Best	Mean	Worst
Circle map	-4140.7318	-4133.9665	-4127.7499	-12413.4182	-12401.9560	-12392.1602
Iterative map	-3479.1771	-2462.6692	-1805.8915	-8909.3380	-6630.8275	-4817.6648
Logistic map	-3331.1754	-2430.1700	-1805.8915	-8138.5710	-6301.5662	-5415.6316
Piecewise map	-2149.5322	-1779.7561	-1309.3467	-5417.6747	-4012.8150	-2290.7057
Sine map	-3071.0364	-23607.417	-1858.6114	-7285.1326	-5796.2906	-4462.0319
Singer map	-4187.1411	-4185.5580	-4183.6375	-12559.5028	-12556.8613	-12553.2917
Sinusoidal map	-4183.3942	-4176.8718	-4175.0514	-12537.6491	-12528.8477	-12525.5593
Tent map	-3222.2081	-3157.1287	-3093.2322	-9598.1698	-9474.4398	-9367.5328

4.3. Ackley Function

$$f_3 = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)) + e + 20, \quad -32 \leq x_i \leq 32.$$

Ackley function as shown in Fig.1c has global minimum 0 at $\mathbf{x} = 0$. Independently running CSOBAS and SOBAS (the iteration number is set as 10^3) each with 100 times, the acquired results are listed in Table 7 corresponding to different chaotic maps. It seems that the used nine chaotic maps have no ability to promote the CSOBAS to evolve towards global solution. Nonetheless, the CSOBAS has good convergence compared to some optimization results reported on the published literatures (see Table 8).

Table 7: Statistical results obtained by the CSOBAS for the Ackley function.

	D=10			D=30		
	Best	Mean	Worst	Best	Mean	Worst
Circle map	2.78926044	3.84395946	4.34573354	3.35760474	3.90342474	4.28759109
Iterative map	2.81079403	3.83627088	4.52597306	3.55103932	3.86281907	4.23645169
Logistic map	2.31936109	3.84311711	4.40991821	3.53694629	3.88469066	4.27347645
Piecewise map	2.75576992	3.82085980	4.42282728	3.45483688	3.86946217	4.21110517
Sine map	2.18245896	3.85177438	4.51014045	3.21384731	3.87340812	4.21746946
Singer map	2.49577164	3.34583611	3.82160929	2.92993465	3.49550091	3.83245892
Sinusoidal map	2.61266444	3.89757496	4.55430115	3.49839287	3.89879102	4.36952746
Tent map	2.80197338	3.88177101	4.54087356	3.41690571	3.86849009	4.35649447

4.4. Comparison Between The CSOBAS and Other Algorithms.

We place the best results of the CSOBAS in Table 8 from Table 4, Table 6 and Table 7, and compare them with other optimization algorithms such as FA, Chaotic Firefly algorithm (CFA), Whale Optimization algorithm (WOA). We obtained their respective program code from [2, 26, 27] to conduct compare experiment to better show the SCOBAS. From which, it is seen that the results optimized by the CSOBAS under singer map are clearly better than ones obtained by other algorithm for Schwefel function. For both Schaffer and Ackley functions, the convergence quality is worse than others, the reason is that only one beetle in the CSOBAS participates in search, which cuts the convergence accuracy and quality.

4.5. Two-Dimensional Multimodal Function with Fixed Global Minimum

1) Goldstein-Price function:

$$f_4 = (1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2))(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)) \quad -2 \leq x_i \leq 2.$$

The global minimum is 3 occurring at $\mathbf{x} = (0, -1)$.

2) Six-Hump Camel-Back function:

$$f_5 = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4, \quad -5 \leq x_i \leq 5.$$

The global minimum is -1.0316285 occurring at $\mathbf{x} = (0.08983, -0.7126)$ or $\mathbf{x} = (-0.08983, 0.7126)$.

3) Branin function:

$$f_6 = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10, \quad -5 \leq x_i \leq 5.$$

The global minimum is approximately 0.398 and it is reached at the three points $(-3.142, 2.275)$, $(3.142, 2.275)$ and $(9.425, 2.425)$.

Table 8: Comparison between the CSOBAS and other algorithms.

	population	Iteration	$f_1(D = 2)$	$f_2(D = 30)$	$f_3(D = 30)$
CSOBAS	1	1000	-0.99801259	-12559.5028	2.92993465
SOBAS	1	1000	-0.96351203	-6758.5250	19.42693137
BAS-WPT [2]	1	1000	-0.68251	-5165.1503	19.7447
PSO [26]	30	1000	-0.99028	-8752.5627	1.1865
FA [26]	30	1000	-1	-9561.4278	0.0017231
CFA [26]	30	1000	-1	-5862.5174	8.925
CFA-II [26]	30	1000	-1	-8265.8782	2.5094
WOA [27]	30	1000	-1	-9016.2679	8.8818E-16
CAPSO [24]	30	5000	N/A	N/A	1.2
CAPSO [23]	30	5000	N/A	N/A	8.57

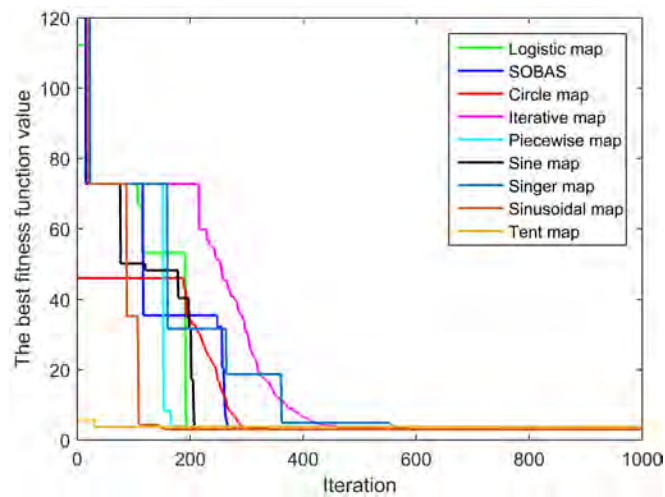


Figure 2: The Goldstein-Price function.

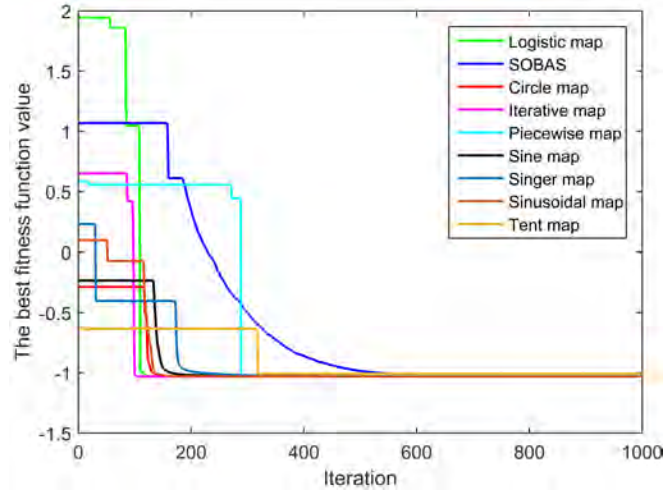


Figure 3: The Branin function.

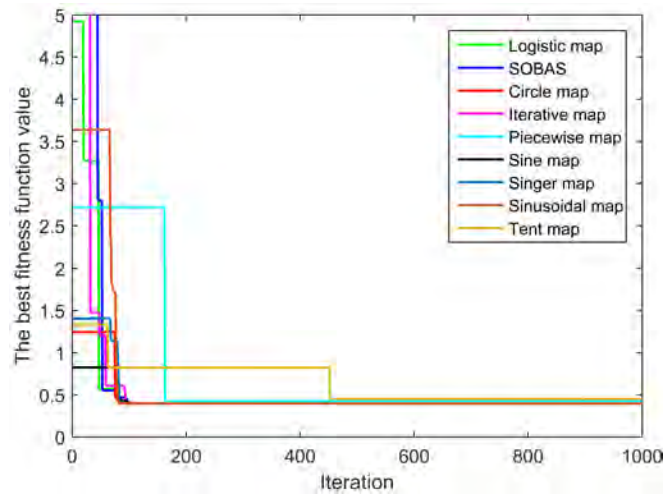


Figure 4: The Branin function.

With 1000 iterations, Fig. 2, Fig. 3 and Fig. 4 show the evolutionary process for both the different chaotic maps based CSOBAS and the SOBAS that reach the minimum value for f_3 , f_4 and f_5 , respectively. The conclusion that can be shared is that the number of iterations required for the CSOBAS that reaches the global minimum of the optimization functions for the first time with an appropriate chaotic map is much smaller than the SOBAS. For f_3 , the performance of the sinusoidal map is best than the others. Iterative map is suitable for f_4 , but it requires more runs. Tent map fails to make the CSOBAS evolve to the global minimum for f_3 , f_4 and f_5 .

5. Conclusions

A chaotic-based second-order BAS algorithm (CSOBAS) that contributes to urging the SOBAS to escape from the local minima and improve convergence quality for multimodal functions has been developed. Eight different chaotic maps have been used, from the corresponding simulation results, it is obtained that the singer map can achieve the best convergence with high probability. Six typical multimodal functions, i.e, Schaffer, Schwefel, Ackley, Goldstein-Price, Six-Hump Camel-Back and Branin function, have been introduced. For the Schaffer function, convergence quality of the CSOBAS is clearly better than the SOBAS, although it may not reach the minimum -1 due to only an individual involved in the evolution. Compared to the SOBAS, however, the convergence quality of the CSOBAS can be increased as the number of iterations increases. For the two functions Schwefel and Ackley, the convergence qualities achieved by the CSOBAS significantly outperform the SOBAS regardless of the dimension $D = 10$ or $D = 30$. For the last three functions with a fixed minimum, the iterations number required for the CSOBAS is smaller than the SOBAS. Combined with the above observations, the CSOBAS algorithm proposed in this paper is effective for multimodal function optimization.

Compared with swarm intelligent optimization algorithms such as BSO and BSAS, only one individual is one of the advantages of second-order BAS, however it also has caused disadvantage for the SOBAS. Although it is now capable of handling high-dimensional optimization problems, lower convergence quality is still a problem for large variable region. This is also what we need to improve in the future.

References

- [1] X. Jiang, S. Li, Bas: beetle antennae search algorithm for optimization problems, *International Journal of Robotics and Control*. DOI: <https://doi.org/10.5430/ijrc.v1n1p1>.
- [2] X. Jiang, S. Li, Beetle antennae search without parameter tuning (bas-wpt) for multi-objective optimization, arXiv preprint arXiv:1711.02395.
- [3] Z. Zhu, Z. Zhang, W. Man, X. Tong, J. Qiu, F. Li, A new beetle antennae search algorithm for multi-objective energy management in microgrid, in: 2018 IEEE Conference on Industrial Electronics and Applications (ICIEA), 2018, pp. 1599–1603.
- [4] D. Song, Application of particle swarm optimization based on beetle antennae search strategy in wireless sensor network coverage, in: 2018 International Conference on Network, Communication, Computer Engineering (NCCE), 2018.
- [5] L. Wang, Q. Wu, S. Li, J. Liu, Ship course control based on bas self-optimizing pid algorithm.
- [6] T. Wang, L. Yang, Q. Liu, Beetle swarm optimization algorithm: theory and application, arXiv preprint arXiv:1808.00206.
- [7] J. Wang, H. Chen, Bsas: Beetle swarm antennae search algorithm for optimization problems, arXiv preprint arXiv:1807.10470.
- [8] M.-j. LIN, Q.-h. LI, A hybrid optimization method of beetle antennae search algorithm and particle swarm optimization, *DEStech Transactions on Engineering and Technology Research (ecar)*.
- [9] M. P. Garcia, O. Montiel, O. Castillo, R. Sepúlveda, P. Melin, Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation, *Applied Soft Computing* 9 (3) (2009) 1102–1110.
- [10] E. Masehian, D. Sedighzadeh, Multi-objective pso-and npsso-based algorithms for robot path planning, *Advances in electrical and computer engineering* 10 (4) (2010) 69–76.
- [11] T. Apostolopoulos, A. Vlachos, Application of the firefly algorithm for solving the economic emissions load dispatch problem, *International journal of combinatorics* 2011.
- [12] M.-H. Horng, Vector quantization using the firefly algorithm for image compression, *Expert Systems with Applications* 39 (1) (2012) 1078–1091.
- [13] Z. Bingül, O. Karahan, A fuzzy logic controller tuned with pso for 2 dof robot trajectory control, *Expert Systems with Applications* 38 (1) (2011) 1017–1031.
- [14] L. Wang, D. Z. Zheng, Q. S. Lin, Survey on chaotic optimization methods, *Computing Technology & Automation*.
- [15] D. Levy, Chaos theory and strategy: Theory, application, and managerial implications, *Strategic Management Journal* 15 (S2) (2010) 167–178.

- [16] Y. Wang, K. W. Wong, X. Liao, G. Chen, A new chaos-based fast image encryption algorithm, *Applied Soft Computing Journal* 11 (1) (2011) 514–522.
- [17] T. Gao, Z. Chen, A new image encryption algorithm based on hyper-chaos, *Physics Letters A* 372 (4) (2008) 394–400.
- [18] P. Voglewede, A. H. C. Smith, A. Monti, Dynamic performance of a scara robot manipulator with uncertainty using polynomial chaos theory, *IEEE Transactions on Robotics* 25 (1) (2009) 206–210.
- [19] A. H. Gandomi, X. S. Yang, S. Talatahari, A. H. Alavi, Firefly algorithm with chaos, *Communications in Nonlinear Science & Numerical Simulation* 18 (1) (2013) 89–98.
- [20] F. B. Ozsoydan, Adaptive firefly algorithm with chaos for mechanical design optimization problems, Elsevier Science Publishers B. V., 2015.
- [21] W. Gong, S. Wang, Chaos ant colony optimization and application, in: 2009 International Conference on Internet Computing for Science & Engineering, 2009, pp. 301–303.
- [22] M. Gao, J. Xu, J. Tian, H. Wu, Path planning for mobile robot based on chaos genetic algorithm, in: 2008 International Conference on Natural Computation, 2008, pp. 409–413.
- [23] B. Liu, L. Wang, Y. H. Jin, F. Tang, D. X. Huang, Improved particle swarm optimization combined with chaos, *Chaos Solitons & Fractals* 25 (5) (2005) 1261–1271.
- [24] A. H. Gandomi, G. J. Yun, X. S. Yang, S. Talatahari, Chaos-enhanced accelerated particle swarm optimization, *Communications in Nonlinear Science & Numerical Simulation* 18 (2) (2013) 327–340.
- [25] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, 2008.
- [26] <https://github.com/harshit2997>, updated May 24, 2018.
- [27] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Advances in Engineering Software* 95 (2016) 51–67.