Filomat 34:15 (2020), 5121–5137
https://doi.org/10.2298/FIL2015121W

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Beetle Swarm Optimization Algorithm: Theory and Application

Tiantian Wang[a], Long Yang[b], Qiang Liu*[a]

[a]*Engineering College,Ocean University of China,238Songling Road ,Laoshan District,Qingdao 266100, Shandong,China*
[b]*Chemistry Experimental Center, Xihua University,9999Hongguang Road, Pidu District,Chengdu 610039,Sichuan,China*

Abstract. In this paper, a new meta-heuristic algorithm, called beetle swarm optimization (BSO) algorithm, is proposed by enhancing the performance of swarm optimization through beetle foraging principles. The performance of 23 benchmark functions is tested and compared with widely used algorithms, including particle swarm optimization (PSO) algorithm, genetic algorithm (GA) and grasshopper optimization algorithm (GOA). Numerical experiments show that the BSO algorithm outperforms its counterparts. Besides, to demonstrate the practical impact of the proposed algorithm, two classic engineering design problems, namely, pressure vessel design problem and himmelblau's optimization problem, are also considered and the proposed BSO algorithm is shown to be competitive in those applications.

1. The first section

In the past decade, various optimization algorithms have been proposed and applied to different research fields. Procedures may vary to solve different optimization problems, but the following questions need to be considered in advance before selecting the optimization algorithm: (1) Parameters of the problem. The problem can be divided into continuous or discrete depending on the parameters. (2) Constraints of variables. Optimization problems can be classified into constrained and unconstrained ones based on the type of constraints [1]. (3) The cost function of a given problem. The problem can be divided into single-objective and multi-objective problems [2]. Based on the above three points, we need to select the optimization algorithm according to the parameter type, constraint and target number.

The development of optimization algorithms is relatively mature at present, and many excellent optimization algorithms have been applied in various fields. We can divide the optimization algorithms into two categories: gradient-based methods and meta-heuristic algorithms. For simple problems such as continuous and linear problems, some classical algorithm gradient algorithms can be utilized, such as Newton's method [3], truncated gradient method [4], gradient descent method [5],etc. For more complex problems, meta-heuristics such as genetic algorithm [6], ant colony algorithm [7] and particle swarm optimization algorithm [8] can be considered. And the meta heuristic algorithm becomes very popular because of its stability and flexibility and its ability to better avoid local optimization [9].

2010 *Mathematics Subject Classification*. 68UXX
Keywords. Heuristic algorithm;Beetle Antennae Search Algorithm;Beetle Swarm Optimization;Multi-objective Optimization
Received: 19 September 2018; Revised: 02 February 2019; Accepted: 12 February 2019
Communicated by Shuai Li
Research supported by the National Natural Science Foundation of China through Grants Nos. 41072176 and 41371496 and the National Science and Technology Support Program through Grant No. 2013BAK05B04.
Email addresses: 18366135507@163.com (Tiantian Wang), yanglongren@163.com (Long Yang), liuqiang@ouc.edu.cn (Qiang Liu*)

Table 1: Algorithm Classification

Meta-heuristic Algorithms	Evolutionary Algorithms	Genetic Algorithm [6] Evolution Strategies[10] Probability-Based Incremental Learning[11] Genetic Programming[12] Biogeography-Based Optimizer[13]
	Physics-based Algorithms	Simulated Annealing[14] Gravitational Local Search[15] Big-Bang Big-Crunch[16] Gravitational Search Algorithm[17] Charged System Search[18] Central Force Optimization[19] Artificial Chemical Reaction Optimization Algorithm[20] Black Hole algorithm[21] Ray Optimization algorithm[22] Small-World Optimization Algorithm[23] Galaxy-based Search Algorithm[24] Curved Space Optimization[25]
	Swarm-based Algorithms	particle swarm optimization algorithm[8] Honey Bees Optimization Algorithm[26] Artificial Fish-Swarm Algorithm[27] Termite Algorithm[28] Wasp Swarm Algorithm[29] Monkey Search[30] Bee Collecting Pollen Algorithm[31] Cuckoo Search[32] Dolphin Partner Optimization[33] Firefly Algorithm[34] Bird Mating Optimizer[35] Fruit fly Optimization Algorithm[36]

People usually divide the meta-heuristic algorithm into three types, which are based on the principles of biological evolution, population and physical phenomena. The evolutionary approach is inspired by the concept of natural evolution. The population based optimization algorithm is mainly inspired by the social behavior of animal groups, while the physical phenomenon based method mainly imitates the physical rules of the universe.Table 1 summarizes the algorithms included in each category.

In face of so many existing meta-optimization algorithms, a concern naturally rises. So far, there have been many different types of optimization algorithms. Why do we need more algorithms? We will mention that there is no free lunch (NFL) [37] theorem, no matter how smart or how clumsy the optimization algorithm is, their performance is logically equivalent. That is, there is no optimization algorithm that can solve all optimization problems. This theorem makes the number of algorithms increase rapidly over the past decade, which is one of the motivations of this paper.

In this paper, a new optimization, namely Beetle Swarm Optimization (BSO) algorithm, is proposed by combining beetle foraging mechanism with group optimization algorithm. The rest of the paper is structured as follows. Section 2 describes the Beetle Swarm Optimization algorithm developed in this study. Section 3 tests the performance of the algorithm on the unimodal functions, multimodal functions and fixed-dimension multimodal functions. Section 4 applies the BSO algorithm to the multi-objective problems to further test the performance of the algorithm. Section 5 draws conclusions.

2. Beetle Swarm Optimization (BSO)

2.1. Beetle Antennae Search Algorithm

A meta-heuristic optimization algorithm based on the search behavior of long-horned beetles was proposed by Jiang X et al. [38, 39]. When using BAS to optimize nonlinear systems, a simple two-step building procedure is employed as follows: (i) model the searching behavior; (ii) formulate the behavior of detecting. In this section, the position of beetle at time t (t=1,2,...) is denoted as x^t, denote the concentration of odor at position x to be $f(x)$ known as a fitness function, where the maximum (or minimum) value corresponds to the point of odor source.

Mathematically, BAS model is stated as follows. The random search directions of beetles are shown as follows[38]:

$$\overrightarrow{b} = \frac{rands(n,1)}{||rands(n,1)||}. \tag{1}$$

where $rands(.)$ denote the random function, and indicates the space dimension. Then create the beetle's left and right spatial coordinates[38, 40]:

$$x_{rt} = x^t + d_0 * \overrightarrow{b} \Big/ 2, \tag{2}$$

$$x_{lt} = x^t - d_0 * \overrightarrow{b} \Big/ 2. \tag{3}$$

where x_{rt} represents the position coordinates of the right antennae at time t, and x_{lt} represents the coordinates of the left antennae at time t. d_0 represents the distance between two antennae. Use the fitness function value to represent the scent intensity at the right and left antennae, we denote them as $f(x_{rt})$ and $f(x_{lt})$.

In the next step, Setting the beetle's iterative mechanism to formulate the detect behavior, the model as follows[38]:

$$x^{t+1} = x^t + \delta^t * \overrightarrow{b} * sign(f(x_{rt}) - f(x_{lt})). \tag{4}$$

where δ^t represents the step factor, the step size usually decreases as the number of iterations increases. $sign(.)$ represents a sign function.

It is worth pointing out that searching distance d_0 and δ. In general, setting the initial step length as a constant, and the initial step length increases as the fitness function dimension increases. To simplify the parameter turning further more, we also construct the relationship between searching distance d and step size δ as follows [39]:

$$\delta^t = c_1\delta^{t-1} + \delta^0 \text{or} \delta^t = eta * \delta^{t-1}. \tag{5}$$

$$d^t = \delta^t \Big/ c_2. \tag{6}$$

where c_1 , c_2 and are constants to be adjusted by designers, we recommend eta's value is 0.95.

2.2. Beetle Swarm Optimization Algorithm

With the continuous deepening of the experiment, we found that the performance of the BAS algorithm in dealing with high-dimensional functions is not very satisfactory, and the iterative result is very dependent on the initial position of the beetle. In other words, the choice of initial position greatly affects the efficiency and effectiveness of optimization. Inspired by the swarm optimization algorithm, we have made further improvements to the BAS algorithm by expanding an individual to a group.That is the beetle swarm optimization (BSO) algorithm we will introduce.

In this algorithm, each beetle represents a potential solution to the optimization problem, and each beetle corresponds to a fitness value determined by the fitness function. Similar to the particle swarm

algorithm, the beetles also share information, but the distance and direction of the beetles are determined by their speed and the intensity of the information to be detected by their long antennae.

In mathematical form, we borrowed the idea of particle swarm algorithm. There is a population of n beetles represented as $X = (X_1, X_2, \cdots, X_n)$ in an S-dimensional search space, where the ith beetle represents an S-dimensional vector $X_i = (x_{i1}, x_{i2}, \cdots, x_{iS})^T$,represents the position of the ith beetle in the S-dimensional search space, and also represents a potential solution to the problem. According to the target function, the fitness value of each beetle position can be calculated. The speed of the ith beetle is expressed as $V_i = (V_{i1}, V_{i2}, \cdots, V_{iS})^T$ .The individual extremity of the beetle is represented as $P_i = (P_{i1}, P_{i2}, \cdots, P_{iS})^T$, and the group extreme value of the population is represented as $P_g = (P_{g1}, P_{g2}, \cdots, P_{gS})^T$ [41]. Mathematical model for simulating its behavior is as follows:

$$X_{is}^{k+1} = X_{is}^k + \lambda V_{is}^k + (1 - \lambda)\xi_{is}^k. \tag{7}$$

where $s = 1, 2, \cdots, S$; $i = 1, 2, \cdots, n$; k is the current number of iterations. V_{is}^k is expressed as the speed of beetles, and ξ_{is} represents the increase in beetle position movement. λ is a positive constants.

Then the speed formula is written as [8, 42, 43]:

$$V_{is}^{k+1} = \omega V_{is}^k + c_1 r_1 (P_{is}^k - X_{is}^k) + c_2 r_2 (P_{gs}^k - X_{gs}^k). \tag{8}$$

where c_1 and c_2 are two positive constants, r_1 and r_2 are two random functions in the range[0,1]. ω is the inertia weight. In the standard PSO algorithm, ω is a fixed constant, but with the gradual improvement of the algorithm, many scholars have proposed a changing inertia factor strategy [41, 44, 45].

This paper adopts the strategy of decreasing inertia weight, and the formula is as follows [41]:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{K} * k. \tag{9}$$

Where $\omega_{\min}$ and $\omega_{\max}$ respectively represent the minimum and maximum value of ω. k and K are the current number of iterations and the maximum number of iterations. In this paper, the maximum value of ω is set to 0.9, and the minimum value is set to 0.4 [46],so that the algorithm can search a larger range at the beginning of evolution and find the optimal solution area as quickly as possible. As ω gradually decreases, the beetle's speed decreases and then enters local search.

The ξ function, which defines the incremental function, is calculated as follows:

$$\xi_{is}^{k+1} = \delta^k * V_{is}^k * sign(f(X_{rs}^k) - f(X_{rs}^k)). \tag{10}$$

In this step, we extend the update (4) to a high dimension. δ indicates step size. The search behaviors of the right antenna and the left antenna are respectively expressed as:

$$X_{rs}^{k+1} = X_{rs}^k + V_{is}^k * d/2, \tag{11}$$

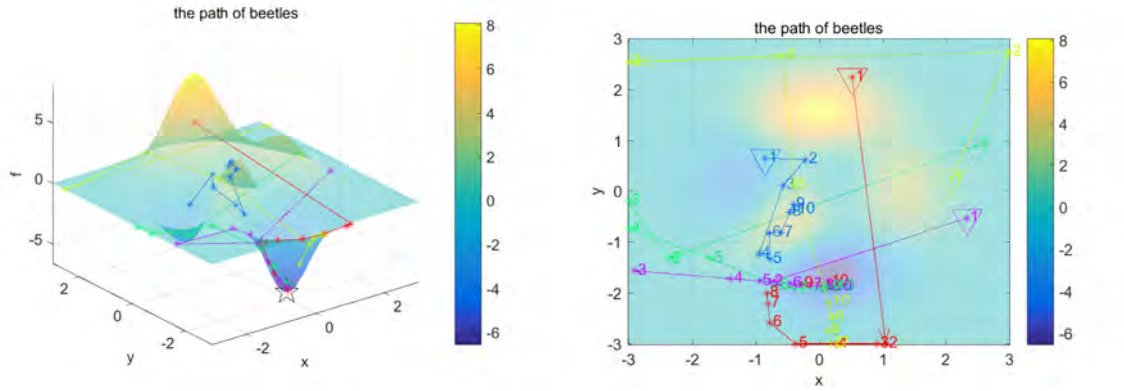
$$X_{ls}^{k+1} = X_{ls}^k - V_{is}^k * d/2. \tag{12}$$

Figure 1 shows the trajectories of the beetle swarm in three-dimensional space. To represent the search path more visually, we used a small population size and showed the location change process of 10 iterations in 3D space. Because factors such as step length and inertial weight coefficient are decreasing in the iterative process, the algorithm will not converge to the target point too quickly, thus avoiding the group falling into the local optimum greatly.

The BSO algorithm first initializes a set of random solutions. At each iteration, the search agent updates its location based on its own search mechanism and the best solution currently available. The combination of these two parts can not only accelerate the population's iteration speed, but also reduce the probability of the population falling into the local optimum, which is more stable when dealing with high-dimensional problems.

The pseudo code of the BSO algorithm is presented in Algorithm **1**.

In theory, the BSO algorithm includes exploration and exploitation ability, so it belongs to global optimization. Furthermore, the linear combination of speed and beetle search enhances the rapidity and accuracy of population optimization and makes the algorithm more stable. In the next section, we will examine the performance of the proposed algorithm through a set of mathematical functions.



(a) three-dimensional diagram (b) vertical view

Figure 1: Beetles Search Path in 2D Space(a) and 3D Space(b)

Algorithm 1 Procedure:

Input: Initialize the swarm$X_i(i = 1, 2, ..., n)$;
Initialize population speed v ;
Set step size δ , speed boundary $v_{\max}$ and $v_{\min}$, population size *sizepop* and maximum number of iterations K ;
Calculate the fitness of each search agent;
while $k < K$ **do**
Set inertia weight ω using (9);
Update d using (6);
for each search agent **do**
Calculate $f(X_{rs})$ and $f(X_{ls})$ using (11)(12);
Update the incremental function ξ by the (10);
Update the speed formula V by the (8);
Update the position of the current search agent by the (7);
end for
Calculate the fitness of each search agent$f(x)$;
Record and store the location of each search agent;
for each search agent **do**

if $f(x) < f_{pbest}$ **then**
$f_{pbest} = f(x)$
end if
if $f(x) < f_{gbest}$ **then**
$f_{gbest} = f(x)$
end if
end for
Update x^* if there is a better solution;
Update step factor δ by the (5);
end while
Output: x_{best} and f_{best}.

3. Results and Discussion

In the optimization field, a set of mathematical functions with optimal solutions is usually used to test the performance of different optimization algorithms quantitatively. And the test functions should be diverse so that the conclusions are not too one-sided. In this paper, three groups of test functions with different characteristics are used to benchmark the performance of the proposed algorithm which are unimodal functions, multimodal functions and fixed-dimension multimodal functions [47–49] .The specific form of the function is given in table 2-4, where *Dim* represents the dimension of the function, *Range* represents the range of independent variables, that is, the range of population, and f_{min} represents the minimum value of the function. In the comparative experiment, the experimental environment was the same. Matlab version was 2016a, win10 operating system, and the processor model was Inter(R) Core(TM) i7-8550u CPU @1.80GHz 2.00GHz.

Table 2: Description of unimodal benchmark functions

Function	*Dim*	*Range*	f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10,10]	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	30	[-100,100]	0
$f_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30	[-100,100]	0
$f_5(x) = \sum_{i=1}^{n-1} [100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2]$	30	[-30,30]	0
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[-100,100]	0
$f_7 = \sum_{i=1}^{n} ix^4 + random[0, 1)$	30	[-1.28,1.28]	0

Table 3: Description of multimodal benchmark functions

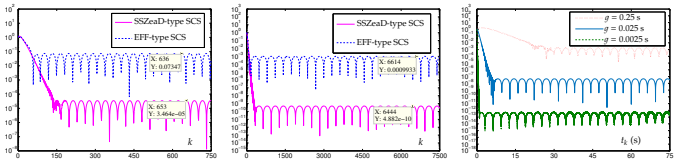
Function	*Dim*	*Range*	f_{min}
$f_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	[-500,500]	-418.9829*Dim
$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
$f_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30	[-32,32]	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[-600,600]	0
$f_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\}$ $+ \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i+1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m x_i < -a \end{cases}$	30	[-50,50]	0
$f_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n} \begin{matrix}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)]+ \\ (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\end{matrix}\}$ $+ \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[-50,50]	0

Figure 2 shows the two-dimensional versions of unimodal function, multimodal function and fixed-dimension multimodal function respectively. The unimodal test function has only one global optimal solution, which is helpful to find the global optimal solution in the search space, and it can test the convergence speed and efficiency of the algorithm well. From Figure 2, it can also be seen that the multimodal function and the fixed-dimension multimodal test function have multiple local optimal solutions, which can be

Table 4: Description of fixed-dimension multimodal benchmark functions

Function	*Dim*	*Range*	f_{min}
$f_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25} (j + \sum_{i=1}^{2} (x_i - a_{ij})^6)^{-1})^{-1}$	2	[-65,65]	0.998
$f_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}]^2$	4	[-5,5]	0.0003
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316
$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	[-5,5]	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2,2]	3
$f_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[1,3]	-3.86
$f_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	-3.32
$f_{21}(x) = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.1532
$f_{22}(x) = -\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.4028
$f_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.5363

used to test the algorithm to avoid the performance of the local optimal solution, and the fixed-dimension multimodal function compared with unimodal test function is more challenging.



(a) unimodal function (b) multimodal function (c) fixed-dimension function

Figure 2: 2-D version of unimodal function,multimodal function and fixed-dimension multimodal function

In the part of qualitative analysis, six typical test functions are provided, including optimal trajectory map, contour map and convergence curve of search path. In the quantitative analysis part, 50 search agents were used, the maximum number of iterations was set to 1000, and each test function was run 30 times to generate statistical results. Quantitative evaluation was performed using the mean, standard deviation, and program performance time of three performance indicators. Statistical results are reported in Table 5. BSO was compared with PSO [8] ,GA [6] and GOA [50].

3.1. Qualitative Results and Discussion

In this paper, six unimodal , multimodal and fixed-dimension multimodal functions are selected to observe the BSO algorithm's optimization behavior. In order to express the optimization trajectory more intuitively, we use five search agents.

Figure 3 shows the optimal trace of each test function, the contour map of the search path, and the convergence curves. The optimal trajectory gives the best beetle optimization route. Since the initial position of the beetle is randomly generated, the optimal trajectory may be different when reproducing the result. The contour map of the search path can more intuitively display the beetle's trajectory, and connecting the same z-values on the x, y plane makes it easier to observe beetle movements. The convergence curve shows the function value of the best solution obtained so far.

From Figure 3 it can be seen that beetles gradually move to the best point and eventually gather around the global best point. This phenomenon can be observed in unimodal, multimodal, and fixed-dimension multimodal functions. The results show that the BSO algorithm has a good balance between exploration and exploitation capabilities to promote the beetle to move to the global optimum. In addition, in order to more clearly represent the trajectory of the beetle, some of the function images are processed. Such as f_{10}, this paper selects the opposite form and can more intuitively observe the optimal trajectory.

The BSO algorithm of the beetle self-optimization mechanism has been added, which can more intelligently avoid local optimums. During the optimization process, we found that some beetles always move quickly toward the maximum value, and then reach the maximum value and then perform normal iterations. This mechanism makes the beetle cleverly avoid the local optimum during the optimization process. For unimodal and multimodal functions, the advantage of the self-optimization mechanism is even more pronounced.

Figure 3 provides a convergence curve to further prove that this mechanism can improve the search results. The convergence curve clearly shows the descending behavior of all test functions. Observe that the BSO search agent suddenly changes during the early stage of the optimization process, and then gradually converges. According to Berg et al. [51], this behavior ensures that the algorithm quickly converges to the optimal point to reduce the iteration time.

3.2. Quantitative Results and Discussion

The above discussion proves that the proposed algorithm can solve the optimization problem, but pure qualitative test can not prove the superiority of the algorithm. This section raises the dimensions of test functions other than fixed dimensions to 5 dimensions and gives quantified results. Table 5 gives the experimental results of the test function.

As shown in Table 5 , when dealing with the unimodal functions, the processing speed of BSO is comparable to that of PSO, but it is obviously better than GA and GOA algorithm. In addition, compared with the other three algorithms, BSO algorithm is more stable in performance. Adding the beetle search mechanism in the process of optimization makes the algorithm have better global optimization performance, accelerates the convergence speed of the algorithm, and effectively avoids the phenomenon of "premature".

When dealing with multimode functions, BSO algorithm shows good performance again. Because multimodal functions have multiple local optimal solutions, the results can be directed to show that BSO algorithm is effective and efficient in avoiding local optimal solutions.

For the fixed-dimension multimodal functions, the proposed algorithm gives very competitive results. The BSO algorithm has the ability to balance the exploration and exploitation of the individual and can solve more challenging problems.

3.3. Analysis of Convergence Behavior

Convergence curves of BSO,GA,GOA and PSO are compared in Figure 4 for all of the test functions. The figure shows that BSO has good processing ability for unimodal functions, multimodal functions and fixed-dimension functions, and the processing process is very stable. Especially when solving more complex fixed-dimension functions, BSO shows more obvious advantage than other algorithms. It can be seen that BSO is enough competitive with other state-of-the-art meta-heuristic algorithms.

As a summary, the results of this section revealed different characteristics of the proposed BSO algorithm. Efficient and stable search capabilities benefit from beetle-specific optimization features. The increase in the exploration function of the left and right must greatly improve the stability of the search, making the exploration and exploitation capabilities more balanced, and the BSO can handle better for high-dimensional and more complex problems. Overall, the success rate of the BSO algorithm seems to be higher in solving challenging problems. In the next sections, BSO performance is validated on more challenging multi-objective issues.

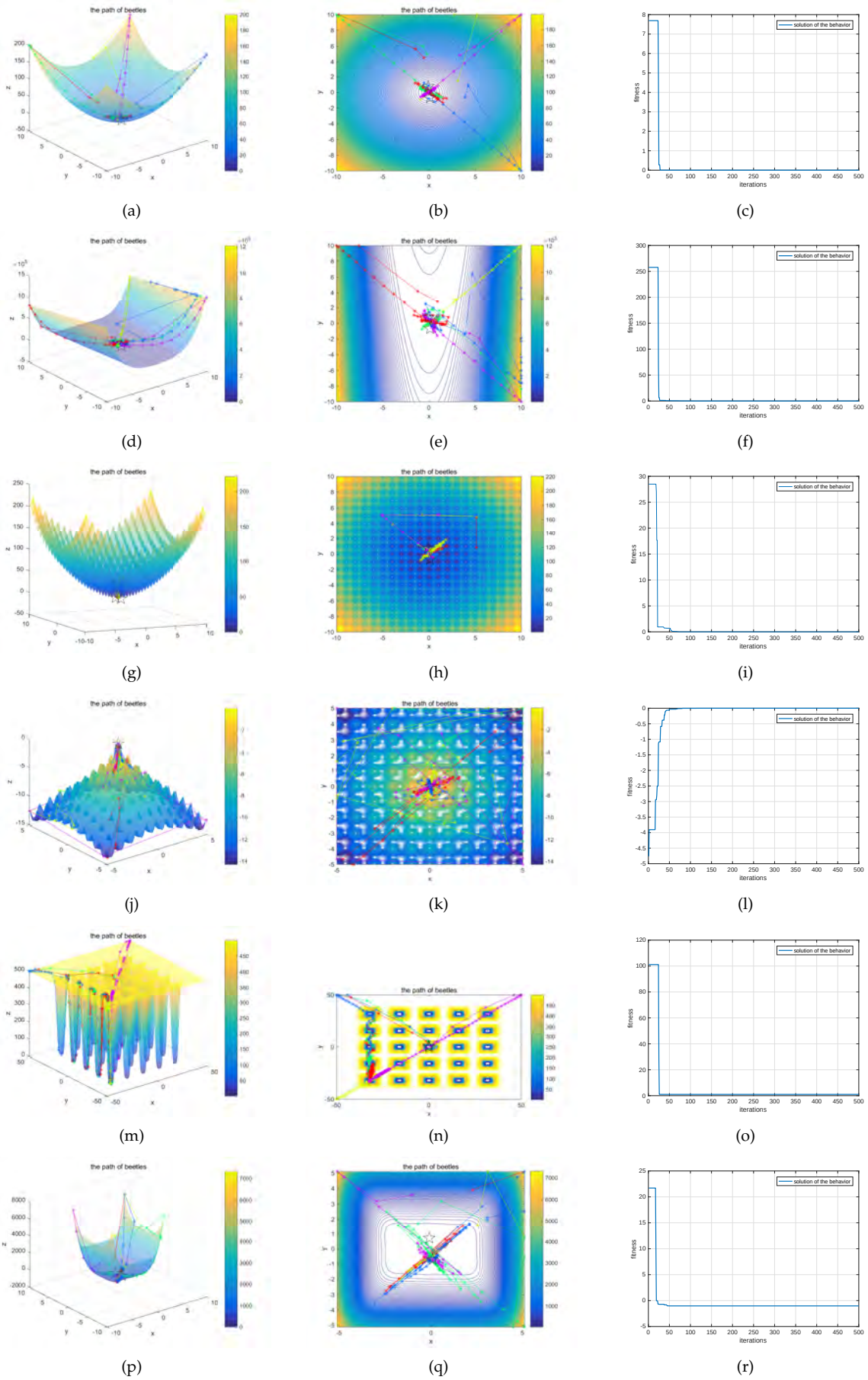


Figure 3: Behavior of BSO on the 2D benchmark problems

Table 5: Comparision of optimization results obtained for the unimodal, multimodal, and fixed-dimension multimodal functions

F	BSO			PSO			GA			GOA		
	ave	*std*	$ave_{time(s)}$	*ave*	*std*	$ave_{time(s)}$	*ave*	*std*	$ave_{time(s)}$	*ave*	*std*	$ave_{time(s)}$
F1	0	9.36E-76	0.5153	0	0	0.4597	0.0025	0.0017	3.7335	0.4004	0.3342	144.5615
F2	1.02E-04	3.92E-04	0.6127	1.3333	3.4575	0.5099	0.008	0.0068	3.7362	1.3612	2.0519	29.6388
F3	0	3.31E-72	0.8765	1.67E+02	912.8709	0.636	7.66E+03	2.34E+03	6.0088	0	0	29.8757
F4	3.55E-09	1.07E-08	0.4999	0	0	0.459	15.7727	4.8173	3.6927	2.50E-05	1.20E-05	29.5846
F5	0.6578	1.4017	0.6432	1.51E+04	3.41E+04	0.5247	43.927954	32.6768	3.7723	3.01E+03	1.64E+04	29.5752
F6	0	0	0.5081	0	0	0.4591	0.0007	0.0011	3.7253	0	0	29.5096
F7	5.17E-04	4.47E-04	0.6382	2.98E-04	0.0003	0.5219	0.0019	0.0009	3.9028	0.0737	0.1023	29.5672
F8	-1.79E+03	173.3453	0.6503	-1.40E+03	85.7482	0.532	-9.78E+03	373.5056	3.8002	-1.74E+03	183.2	29.7437
F9	0.4311	0.9305	0.5215	5.1785	9.0057	0.4683	59.7404	8.75764	3.7708	5.3052	2.9227	29.5213
F10	0.1097	0.4177	0.6282	4.6379	8.4257	0.5253	0.007	0.0051	3.7441	0.6931	0.9474	29.5833
F11	0.1267	0.0849	0.7203	0.1348	0.0926	0.5779	0.0725	0.1001	3.7637	0.1227	0.0638	29.7993
F12	7.00E-06	3.76E-05	1.424	0	0	0.9052	36.1241	9.0446	4.0032	0.0011	0.0059	29.9328
F13	0.0011	0.0034	1.4382	0	0	0.9123	57.65	12.9744	4.0068	0.0022	0.0044	29.9379
F14	0.998	1.54E-16	1.9211	0.998	0	3.1104	0.998	0	3.8205	0.998	0	12.3002
F15	0.0015	0.0036	0.586	0.0042	0.0117	0.4993	0.0039	0.00718	1.5158	0.0035	0.0067	19.9701
F16	-1.0316	6.71E-16	0.4534	-1.0316	0	0.4272	-1.0316	0	1.2441	-1.0316	0	10.306
F17	0.3979	0	0.5045	0.3979	0	0.5767	0.3979	0	1.2171	0.3979	0	10.2556
F18	3	1.03E-15	0.3853	3	0	0.4031	3.9	4.9295	1.2144	5.7	14.7885	10.3014
F19	-3.8609	0.0034	0.8683	-3.6913	0.1247	0.64	-3.8627	0	1.5927	-3.8369	0.1411	20.205
F20	-3.1256	0.3735	0.8685	-2.1198	0.5567	0.6541	-3.2625	0.0605	1.894	-3.2698	0.0607	29.4666
F21	-9.8164	1.2818	0.5519	-1.0902	0.8326	0.9072	-5.9724	3.37309	1.9346	-7.0499	3.2728	20.2475
F22	-10.0513	1.3381	0.6845	-1.0196	0.4063	1.0713	-7.3119	3.4237	2.1298	-7.3062	3.4705	20.4859
F23	-9.1069	2.4111	0.9185	-1.2161	0.6276	1.3545	-5.7112	3.5424	2.4214	-8.6298	3.0277	20.5744

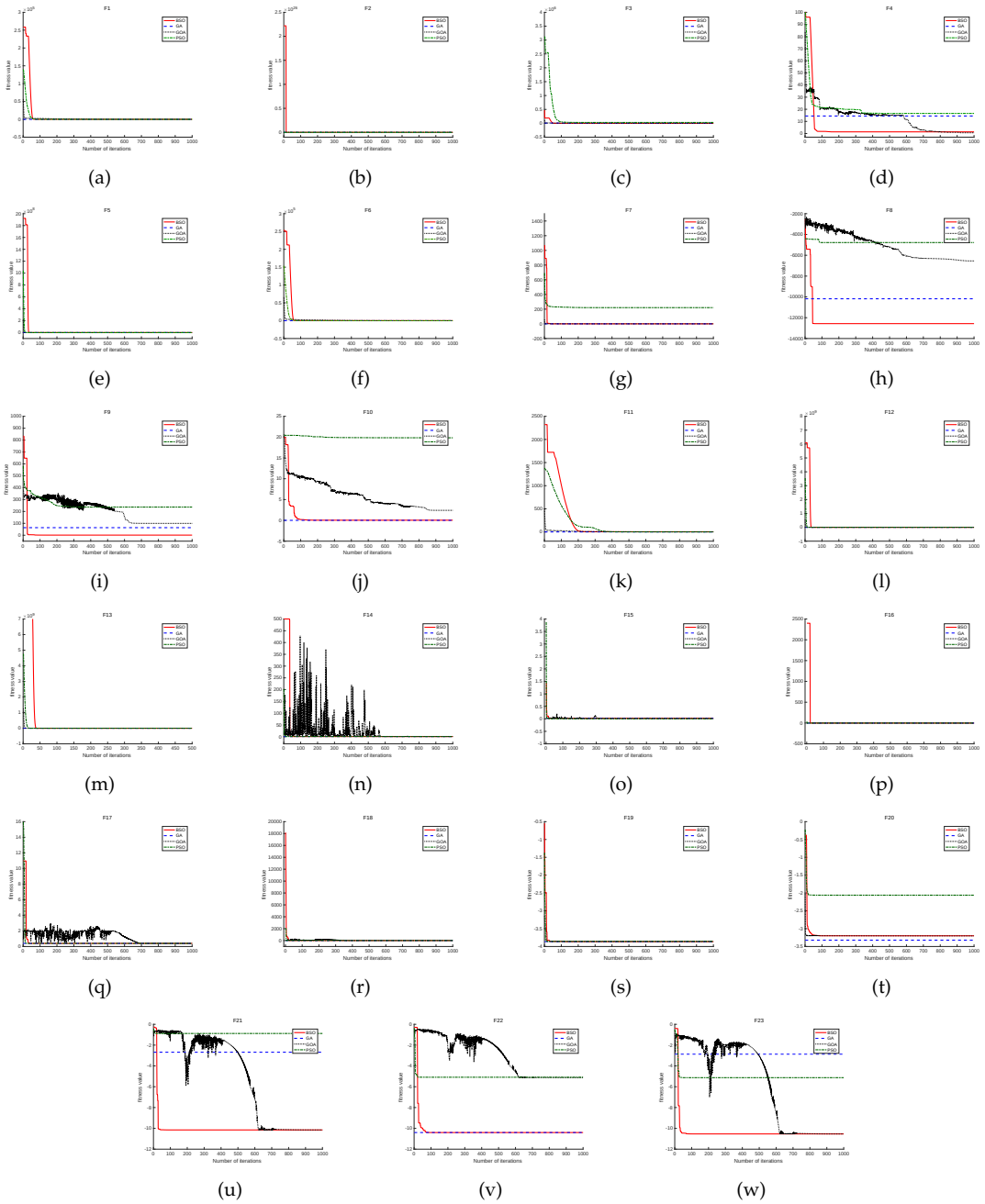


Figure 4: Comparison of convergence curves of BSO and literature algorithms obtained in all of the benchmark problems

4. BSO for Multi-objective Optimization

In order to better illustrate the superiority and competitiveness of BSO algorithm in solving constrained optimization problems, two multi-objective functions in BAS algorithm are used in this paper, and the results are compared with the results of other algorithms.

4.1. BSO for a Pressure Vessel Design Problem

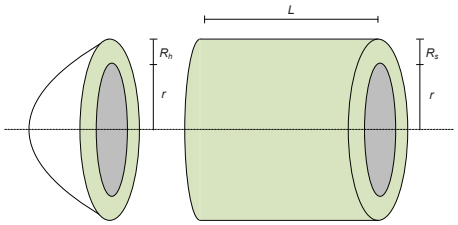


Figure 5: schematic of pressure vessel

As shown in Figure 5, two hemispheres cover the ends of the cylinder to form a pressure vessel. The goal is to minimize the total cost including material costs, welding costs and molding costs [52]:

$$\min f_{\cos t}(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

There are four variables in pressure vessel problem where x_1 is the thickness of the shell(R_s), x_2 is the thickness of the head(R_h) , x_3 is the inner radius (r), and x_4 is the length of the section of the cylinder of the container (L). R_s and R_h are integral times of 0.0625, the available thickness of rolled steel plates, and r and L are continuous. The constraint function can be stated as follows:

$$s.t.g_1(x) = -x_1 + 0.0193x_3 \le 0,$$

$$g_2(x) = -x_2 + 0.00954x_3 \le 0,$$

$$g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0,$$

$$g_4(x) = x_4 - 240 \le 0,$$

$$x_1 \in \{1, 2, 3, ..., 99\} \times 0.0625,$$

$$x_2 \in \{1, 2, 3, ..., 99\} \times 0.0625,$$

$$x_3 \in [10, 200],$$

$$x_4 \in [10, 200].$$

Table 6 illustrates the best results obtained by the BSO algorithm just using 100 iterations and other various existing algorithm to solve the pressure vessel optimization. And most of these results are taken from Jiang et al.(2017).The results show that the best results of BSO algorithm are better than most existing algorithms and in the case where the population number is properly selected (we suggest 50 individuals), the convergence rate is faster and has good The robustness. The BSO algorithm iterative process is shown in Figure 6.

Table 6: comparisons results for pressure vessel function

methods	x_1	x_2	x_3	x_4	$g_1(x)$	$g_2(x)$	$g_3(x)$	$g_4(x)$	f^*
[53]	0.8125	0.4375	42.0984	176.6378	-8.8000e-7	-0.0359	-3.5586	-63.3622	6059.7258
[54]	1	0.625	51.2519	90.9913	-1.0110	-0.136	-18759.75	-149.009	7172.3
[55]	0.8125	0.4375	42.0870	176.7791	-2.210e-4	-0.0360	-3.5108	-63.2208	6061.1229
[56]	1.0000	0.6250	51.0000	91.0000	-0.0157	-0.1385	-3233.916	-149.0000	7079.0370
[57]	0.8125	0.4375	41.9768	182.2845	-0.0023	-0.0370	-22888.07	-57.7155	6171.0000
[58]	0.9375	0.5000	48.3290	112.6790	-0.0048	-0.0389	-3652.877	-127.3210	6410.3811
[59]	0.8125	0.4375	40.3239	200.0000	-0.0343	-0.05285	-27.10585	-40.0000	6288.7445
[60]	1.1250	0.6250	58.1978	44.2930	-0.0018	-0.0698	-974.3	-195.707	7207.4940
[61]	1.1250	0.6250	48.9700	106.7200	-0.1799	-0.1578	97.760	-132.28	7980.8940
[62]	1.1250	0.6250	58.2789	43.7549	-0.0002	-0.0690	-3.71629	-196.245	7198.4330
[39]	0.8125	0.4375	42.0936	176.7715	-9.43e-05	-0.0359	-413.6252	-63.2285	6062.0468
BSO	0.8125	0.4375	42.0984	176.6366	0.0000	-0.0359	0.0000	-63.3634	6059.7000

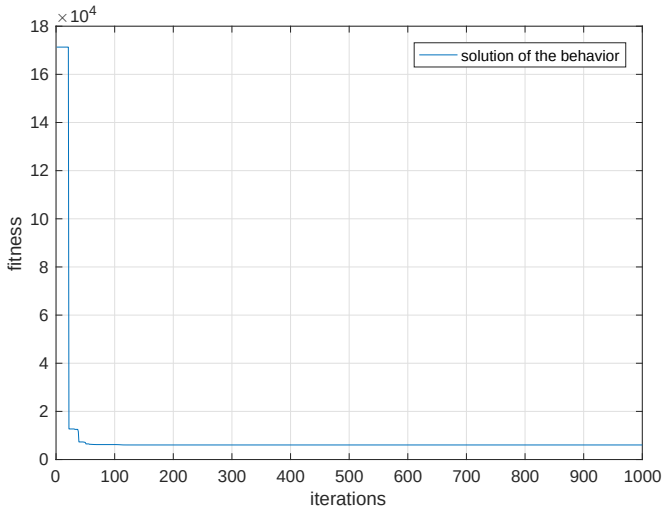


Figure 6: Iteration process for pressure vessel design problem

Table 7: comparisons results for himmelblau function

methods	x_1	x_2	x_3	x_4	x_5	$g_1(x)$	$g_2(x)$	$g_3(x)$	f^*
[64]	78	33	29.995256	45	36.775813	92	98.8405	20 .0000	-30665.54
[65]	78	33	29.995256	45	36.775813	92	98.8405	20	-30665.539
[67]	81.49	34.09	31.24	42.2	34.37	91.78	99.3188	20.0604	-30183.576
[67]	78	33	29.995256	45	36.7258	90.71	98.8287	19.9929	-30665.539
[39]	78	33	27.1131	45	45	92	100.417	20.0206	-31011.3244
BSO	78	33	27.071	45	44.9692	92	100.4048	20	-31025.5563

4.2. BSO for Himmelblau's Optimization Problem

This problem is proposed by Himmelblau [63] and is a common function for nonlinear constrained optimization problems. It is widely used in the optimization field. It consists of five variables, three equality constraints and six inequality constraints. The specific forms are as follows:

$$\min f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.29329x_1 - 40792.141,$$

$$s.t.g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.0022053x_3x_5,$$

$$g_2(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2,$$

$$g_3(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4,$$

$$0 \le g_1(x) \le 92,$$

$$90 \le g_2(x) \le 110,$$

$$20 \le g_3(x) \le 25,$$

$$78 \le x_1 \le 102,$$

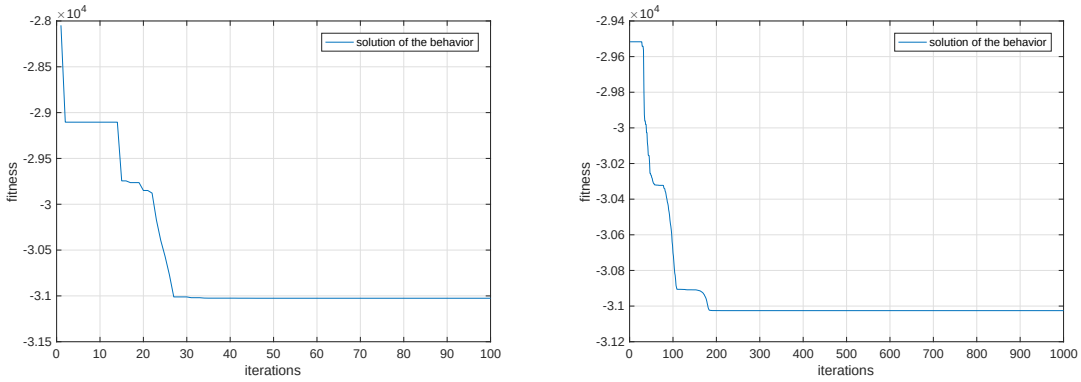
$$33 \le x_2 \le 45,$$

$$27 \le x_3 \le 45,$$

$$27 \le x_4 \le 45,$$

$$27 \le x_5 \le 45.$$

Table 7 shows the performance results of the existing algorithm and the BSO algorithm. The number of iterations is set to 100. Evidently, the best result generated from the BSO shows the most excellent performance among all the results listed in Table . The above experiments justify that the proposed BSO algorithm is effective to handle constraint optimum problem and could achieve a good performance with high convergence rate. In the experiment process, when the population size is 50 and the number of iterations is 1000, the effect is the most stable. The BSO algorithm iterative process is shown in Figure 7.



(a) iterations=100 (b) iterations=1000

Figure 7: Iteration process for himmelblau's optimization problem

5. Conclusions

This paper proposes a new meta-heuristic algorithm called beetle group optimization. The algorithm combines the beetle's foraging mechanism with the group optimization algorithm, and establishes a mathematical model and applies it to unimodal functions, multimodal functions, fixed-dimension multimodal benchmark functions. The results show that compared with the current popular optimization algorithms, the BSO algorithm still gives very competitive results, and has good robustness and running speed. In addition, the BSO algorithm also exhibits higher performance when dealing with nonlinear constraints. Compared with other optimization algorithms, BSO can handle multi-objective optimization problems efficiently and stably.

Finally, in the research process, we found that the change in step size and speed will affect the efficiency and effectiveness of BSO optimization. Therefore, in the next work, we will further study the impact of different parameter settings on BSO.

References

[1] Coello C A C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Meth. Appl. Mech. Eng.(2002) 191(11-12): 1245-1287.
[2] Marler R T, Arora J S. Survey of multi-objective optimization methods for engineering, Struct. Multidiscip. Optim.(2004) 26(6): 369-395.
[3] Bertsekas D P. Nonlinear programming, Belmont: Athena scientific, 1999.
[4] Langford J, Li L, Zhang T. Sparse online learning via truncated gradient, J. Mach. Learn. Res.(2009)10(Mar): 777-801.
[5] Tseng P, Yun S. A coordinate gradient descent method for nonsmooth separable minimization, Math. Program.(2009) 117(1-2): 387-423.
[6] Holland J H. Genetic algorithms, Scientific American.(1992) 267(1): 66-73.
[7] Dorigo M, Birattari M: Ant colony optimization, In:Encyclopedia of machine learning,Boston, MA, 2011,pp. 36-39.
[8] Kennedy J, Eberhart R. PSO optimization, in:Proc. IEEE Int. Conf. Neural Networks. IEEE Service Center, Piscataway, NJ, 1995, pp.4: 1941-1948.
[9] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer, Adv. Eng. Softw.(2014) 69: 46-61.
[10] Rechenberg I. Evolutionsstrategien,in:Simulationsmethoden in der Medizin und Biologie. Springer, Berlin, Heidelberg, 1978,pp.83-114.
[11] Evolutionary algorithms in engineering applications,in: Springer Science and Business Media, 2013.
[12] Koza J R. Genetic Programming II, Automatic Discovery of Reusable Subprograms. MIT Press, Cambridge, MA, 1992.
[13] Simon D. Biogeography-based optimization, IEEE Trans. Evol. Comput. (2008) 12(6): 702-713.
[14] Van Laarhoven P J M, Aarts E H L. Simulated annealing,in:Simulated annealing: Theory and applications. Springer, Dordrecht, 1987,pp. 7-15.
[15] Webster B, Bernhard P J. A local search optimization algorithm based on natural principles of gravitation. 2003.
[16] Erol O K, Eksin I. A new optimization method: big bang–big crunch, Adv. Eng. Softw.(2006) 37(2): 106-111.
[17] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm ,Inf. Sci.(2009)179(13): 2232-2248.
[18] Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search, Acta Mech.(2010) 213(3-4): 267-289.

[19] Formato R A. Central Force Optimization, Prog. Electromagn. Res.(2007)77: 425-491.
[20] Alatas B. ACROA: artificial chemical reaction optimization algorithm for global optimization,Expert Syst. Appl.(2011)38(10): 13170-13180.
[21] Hatamlou A. Black hole: A new heuristic optimization approach for data clustering, Inf. Sci.(2013) 222: 175-184.
[22] Kaveh A, Khayatazad M. A new meta-heuristic method: ray optimization, Comput. Struct.(2012) 112: 283-294.
[23] Du H, Wu X, Zhuang J. Small-world optimization algorithm for function optimization,in:International Conference on Natural Computation. Springer, Berlin, Heidelberg, 2006,pp. 264-273.
[24] Shah-Hosseini H. Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation, International Journal of Computational Science and Engineering(2011) 6(1-2): 132-140.
[25] Moghaddam F F, Moghaddam R F, Cheriet M. Curved space optimization: a random search based on general relativity theory, arXiv preprint arXiv:1208.2214, 2012.
[26] Abbass H A. MBO: Marriage in honey bees optimization-A haplometrosis polygynous swarming approach[M],in:Evolutionary Computation, 2001. Proceedings of the 2001 Congress on. IEEE, 2001,pp. 1: 207-214.
[27] Li X L. A new intelligent optimization-artificial fish swarm algorithm, Doctor thesis, Zhejiang University of Zhejiang, China, (2003).
[28] Roth M. Termite: A swarm intelligent routing algorithm for mobile wireless ad-hoc networks, 2005.
[29] Pinto P C, Runkler T A, Sousa J M C. Wasp swarm algorithm for dynamic MAX-SAT problems,in:International Conference on Adaptive and Natural Computing Algorithms. Springer, Berlin, Heidelberg, 2007,pp. 350-357.
[30] Mucherino A, Seref O. Monkey search: a novel metaheuristic search for global optimization,in:AIP conference proceedings. AIP, 2007,pp. 953(1): 162-173.
[31] Lu X, Zhou Y. A novel global convergence algorithm: bee collecting pollen algorithm,in:International Conference on Intelligent Computing. Springer, Berlin, Heidelberg, 2008,pp. 518-525.
[32] Yang X S, Deb S. Cuckoo search via Lévy flights,in:Nature and Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on. IEEE, 2009,pp. 210-214.
[33] Shiqin Y, Jianjun J, Guangxing Y. A dolphin partner optimization,in:Intelligent Systems, 2009. GCIS'09. WRI Global Congress on. IEEE, 2009,pp. 1: 124-128.
[34] Yang X S. Firefly algorithm, stochastic test functions and design optimisation,Int. J.Bio-Inspired Comput.(2010) 2(2): 78-84.
[35] Askarzadeh A. Bird mating optimizer: an optimization algorithm inspired by bird mating strategies, Commun. Nonlinear Sci. Numer.(2014) 19(4): 1213-1228.
[36] Pan W T. A new fruit fly optimization algorithm: taking the financial distress model as an example,Knowledge-Based Syst.(2012) 26: 69-74.
[37] Wolpert D H, Macready W G. No free lunch theorems for optimization, IEEE Trans. Evol. Comput.(1997) 1(1): 67-82.
[38] Jiang X, Li S. BAS: Beetle Antennae Search Algorithm for Optimization Problems, 2017.
[39] Jiang X, Li S. Beetle Antennae Search without Parameter Tuning (BAS-WPT) for Multi-objective Optimization,2017.
[40] Wang TT, Liu Q. Surge Storm Disaster Loss Prediction Based on BAS-BP Model,Marine Environmental Science (2018)37(03):457-463.
[41] Shi Y, Eberhart R. A modified particle swarm optimizer,in:Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., 1998,pp. 69-73.
[42] Eberhart R, Kennedy J. A new optimizer using particle swarm theory,in:Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on. IEEE, 1995,pp. 39-43.
[43] Poli R, Kennedy J, Blackwell T. Particle swarm optimization,Swarm intelligence, 2007, 1(1): 33-57.
[44] Hu JX, Zeng JC. Inertia Weight Adjustment Strategy in Particle Swarm Optimization,Computer Engineering(2007) 33(11):193-195.
[45] Trelea I C. The particle swarm optimization algorithm: convergence analysis and parameter selection,Information processing letters, 2003, 85(6): 317-325.
[46] Shi Y, Eberhart R C. Empirical study of particle swarm optimization,in:Evolutionary computation, 1999. CEC 99. Proceedings of the 1999 congress on. IEEE, 1999,pp. 3: 1945-1950.
[47] Yao X, Liu Y, Lin G. Evolutionary programming made faster,IEEE Trans. Evol. Comput.(1999) 3(2): 82-102.
[48] Digalakis J G, Margaritis K G. On benchmarking functions for genetic algorithms, Int. J. Comput. Math.(2001) 77(4): 481-506.
[49] Molga M, Smutnicki C. Test functions for optimization needs,Test functions for optimization needs(2005) 101.
[50] Saremi S, Mirjalili S, Lewis A. Grasshopper optimisation algorithm: Theory and application, Adv. Eng. Softw.(2017)105: 30-47.
[51] Van Den Bergh F, Engelbrecht A P. A study of particle swarm optimization particle trajectories,Inf. Sci.(2006)176(8): 937-971.
[52] Kannan B K, Kramer S N. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, Journal of mechanical design(1994)116(2): 405-411.
[53] Kaveh A, Talatahari S. An improved ant colony optimization for constrained engineering design problems,Eng. Comput. (2010)27(1): 155-182.
[54] Li H L, Chou C T. A global approach for nonlinear mixed discrete programming in design optimization,Eng. Optimiz. (1993) 22(2): 109-122.
[55] Coello C A C, Cortés N C. Hybridizing a genetic algorithm with an artificial immune system for global optimization, Eng. Optimiz.(2004) 36(5): 607-634.
[56] Tsai J F, Li H L, Hu N Z. Global optimization for signomial discrete programming problems in engineering design,Eng. Optimiz. (2002) 34(6): 613-622.
[57] Akhtar S, Tai K, Ray T. A socio-behavioural simulation model for engineering design optimization, Eng. Optimiz. (2002) 34(4): 341-354.
[58] Deb K. GeneAS: A robust optimal design technique for mechanical component design,in:Evolutionary algorithms in engineering

applications. Springer, Berlin, Heidelberg, 1997,pp.497-514.
[59] Coello C A C. Use of a self-adaptive penalty approach for engineering optimization problems,Comput. Ind.(2000) 41(2): 113-127.
[60] Wu S J, Chow P T. Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization, Eng. Optimiz.(1995) 24(2): 137-159.
[61] Sandgren E. Nonlinear integer and discrete programming in mechanical design optimization, J. Mech. Des.(1990) 112(2): 223-229.
[62] Lee K S, Geem Z W. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Comput. Meth. Appl. Mech. Eng.(2005) 194(36-38): 3902-3933.
[63] Himmelblau D M. Applied nonlinear programming, McGraw-Hill Companies, 1972.
[64] Dimopoulos G G. Mixed-variable engineering optimization based on evolutionary and social metaphors, Comput. Meth. Appl. Mech. Eng.(2007) 196(4-6): 803-817.
[65] He S, Prempain E, Wu Q H. An improved particle swarm optimizer for mechanical design optimization problems, Eng. Optimiz.(2004) 36(5): 585-605.
[66] Gen M, Cheng R. Foundations of genetic algorithms, Genetic Algorithms and Engineering Design (1997) 1-41.
[67] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization, IEEE Trans. Evol. Comput.(2000) 4(3): 284-294.