

An Efficient Universal Bee Colony Optimization Algorithm

Xuming HAN, Yidan WANG, Chen CAI, Limin WANG, Xiuping HOU, Linlin WANG

Abstract: The artificial bee colony algorithm is a global optimization algorithm. The artificial bee colony optimization algorithm is easy to fall into local optimal. We proposed an efficient universal bee colony optimization algorithm (EUBCOA). The algorithm adds the search factor u and the selection strategy of the onlooker bees based on local optimal solution. In order to realize the controllability of algorithm search ability, the search factor u is introduced to improve the global search range and local search range. In the early stage of the iteration, the search scope is expanded and the convergence rate is increased. In the latter part of the iteration, the algorithm uses the selection strategy to improve the algorithm accuracy and convergence rate. We select ten benchmark functions to testify the performance of the algorithm. Experimental results show that the EUBCOA algorithm effectively improves the convergence speed and convergence accuracy of the ABC algorithm.

Keywords: artificial bee colony algorithm; fitness; search factor; universal optimization

1 INTRODUCTION

In recent years, the swarm intelligent algorithms have been widely concerned. The ant colony optimization (ACO) [1], the particle swarm optimization (PSO) [2], the cuckoo search algorithm (CS) [3], the artificial bee colony algorithm (ABC) [4] and the fruit fly algorithm (FOA) [5] are proposed by scholars.

The artificial bee colony algorithm (ABC) is a swarm intelligent algorithm proposed by Turkish scholar Karaboga in 2005. The idea of the ABC algorithm is inspired by the colonies which exchange information and cooperate mutually to complete the mining task [6]. The information contains the distance between the hive and the food source, the direction, and the cost of the food source. Compared with other algorithms, the artificial bee colony algorithm has the advantages of simple operation, few control parameters, high searching precision and strong robustness [7, 8]. It has been widely used in network routing [9], robot path planning [10], linear dynamic systems [11], Linear Quadratic Optimal Controller [12] and PID controller [13].

Contributions. The EUBCOA algorithm introduces the search factor u on the basis of the adaptive factor φ . It balances effectively the global rapid development capability and the local fine mining ability, and changes the imperfection of the search ability. In the early stage of the iteration, the fitness of the nectar is smaller and the distance from the global optimal solution is far away, the search range is expanded and the probability of finding the optimal solution is accelerated. In the later stage of the iteration, the fitness of the nectar is close to the global optimal solution, and the search range is reduced and the precision is improved. At the same time, the algorithm is inspired by the PSO algorithm. Through introducing the idea of the current optimal solution, the onlooker bees are always followed by the local optimal solution found by the employed bees. The global optimal solution is found in the neighborhood of the local optimal solution, which improves the convergence speed of the algorithm. Through the experiments of different types of test functions, the results show that the EUBCOA algorithm has the highly efficient universal optimization ability.

Related works. Many experts and scholars have been exploring an efficient optimization algorithm that adapts to

a variety of problems. In many studies, some optimization algorithms show better performance in global search [14]. Some algorithms get better results in local search [15]. These optimization algorithms are implemented by introducing adaptive factors, adjusting perturbation, coordinate coding transform, extreme value optimization, and gravitational influence [16-18]. Among them, Santos et al. [16] introduced the inertia diminishing weight to change neighborhood search capabilities. It increased the weight and expanded the search range in the early iteration. In the latter part of the iteration, it searched for excluded local optimal solution, and reduced the weight. Experiments showed that the improved algorithm had achieved good experimental results and played a balance between local and global search. Zhu et al. [17] was inspired by the PSO algorithm, and proposed the Gbest-guided artificial bee colony algorithm for numerical function optimization. In the neighborhood search equation, the algorithm used global optimal solution to increase the disturbance. After several experiments, it was verified that the improved algorithm had better optimization effect. Yi et al. [18] used the quantum artificial bee colony optimization algorithm based on Bloch coordinates of quantum bit. The nectar was encoded through the quantum Bloch coordinates, and updated the nectar with a quantum revolving door, expanded the search range for the employed bees. The experimental results showed that the improved algorithm greatly improved the range of finding the optimal solution.

In view of the lack of convergence accuracy, many scholars had also proposed some improved version [19-20]. Banharnsakun et al. [19] proposed the historical global optimal solution instead of all random neighborhoods, and adjusted the neighborhood search step according to the fitness of the historical global optimal solution. As the neighborhood step decreased, it gradually converged to the optimal solution. The experimental results showed that the improved algorithm had a better optimization effect on the set of test functions, which improved the convergence precision and convergence speed. Ge et al. [20] proposed an efficient optimization mechanism based on extreme optimization strategy to redesign the local search scheme of the onlookers in the ABC algorithm. It selected the worst part of the individual to random variation, and unconditionally accepted new individuals generated by

mutation. The optimization process was regarded as the evolution of complex problems. Through the experiment of multiple set of test functions, it was proved that the improved algorithm had better performance and improved the optimization precision and convergence speed of the algorithm. TSai et al. [21] suggested gravitational effects between the employed bees and the onlooker bees. The employed bees and the onlooker bees would show different search results to improve the performance of the algorithm. When assigned different control parameter values and different numbers of the employed bees and the onlooker bees, you could find different search results under the influence of gravitational attraction.

This paper is organized as follows. In section 2, the main contents and steps of the traditional ABC algorithm is introduced. In section 3, the improvements and steps of the EUBCOA algorithm is introduced. In section 4, the experiments results of different types of test functions are revealed in order to illustrate the performance of the EUBCOA algorithm. Finally, the conclusion is given in section 5.

2 ARTIFICIAL BEE COLONY ALGORITHM

The position of the nectar i is initialized. In the search space the position randomly generates as the Eq. (1), X_{id} is a randomly generated position after the onlooker bees initialize the nectar. Where U_d and L_d are the lower and upper limits of the bees searching for honey, $d \in \{1, 2, \dots, D\}$ is a randomly selected dimension,

$$X_{id} = L_d + \text{rand}(0, 1)(U_d - L_d) \quad (1)$$

The employed bees generate the random position according to Eq. (1) and randomly search for the new nectar V_{id} , according to Eq. (2), and $j \neq i$, φ is a random number source between $[-1, 1]$,

$$V_{id} = X_{id} + \varphi(X_{id} - X_{jd}) \quad (2)$$

After calculating the new nectar, the employed bees calculate the fitness of the new nectar according to Eq. (3). f_i is the fitness value of the nectar and is also the objective function value. When the fitness of the new nectar $V_i = [v_{i1}, v_{i2}, \dots, v_{id}]$ is better than X_i , we use greedy selection method to replace X_i with V_i , otherwise X_i ,

$$\text{fit}_i = \begin{cases} 1/(1 + f_i), & f_i \geq 0 \\ 1 + \text{abs}(f_i), & \text{otherwise} \end{cases} \quad (3)$$

The onlooker bees according to the nectar information provided by the employed bees, in the light of the roulette way, the Eq. (4) calculates the probability of the nectar provided by the employed bees to be followed,

$$p_i = \text{fit}_i / \sum_{i=1}^{SP} \text{fit}_i \quad (4)$$

Determine if the algorithm satisfies the termination condition. When the number of iterations trial reaches the

limit, the algorithm is terminated if it hasn't been updated, and it goes to (2) to find the new nectar again. When the number of iterations trial does not reach the limit, it continues to search for the optimal solution.

$$X_i^{t+1} = \begin{cases} L_d + \text{rand}(0, 1)(U_d - L_d), & \text{trial}_i \geq \text{limit} \\ X_i^t, & \text{trial}_i < \text{limit} \end{cases} \quad (5)$$

3 AN EFFICIENT UNIVERSAL BEE COLONY OPTIMIZATION ALGORITHM

The search range of the traditional ABC algorithm is uncontrollable, which will make the time complexity of the algorithm higher. At the same time, when the onlooker bees update the nectar which is produced by information exchange between the employed bees, is used to update the position of the onlooker bees. It will also reduce the efficiency of the algorithm. In view of this, this paper presents an efficient universal bee colony optimization algorithm.

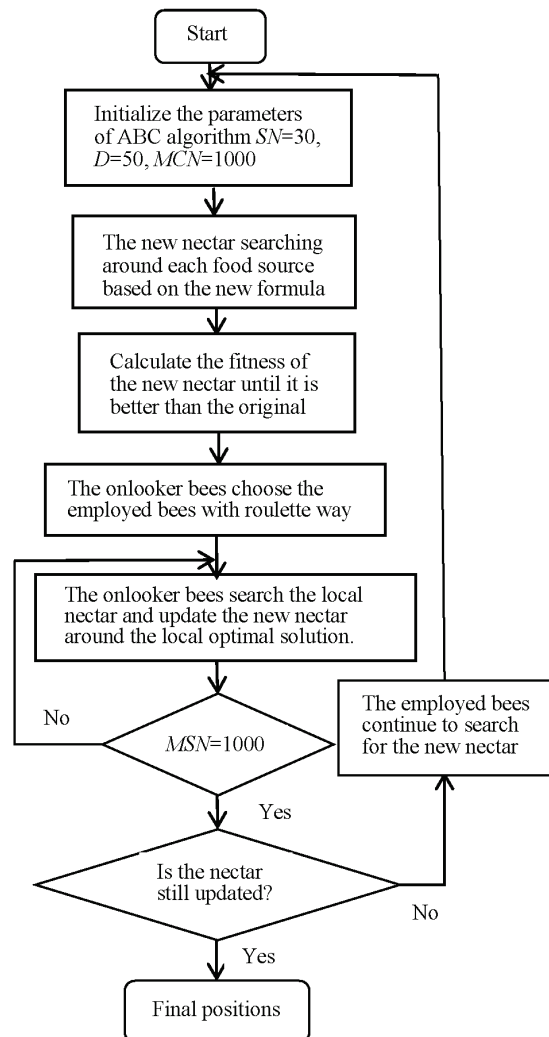


Figure 1 Flowchart of the EUBCOA algorithm

The flow chart of the EUBABC algorithm is given in Fig. 1. The pseudo-code of the EUBCOA algorithm is given in Fig. 2.

```

for  $i = 1$  to the  $SN$  do
    Initialize the population of solution  $X_i$  by using
    Eq. (1)
    Evaluate the fitness of the new nectar, according
    to Eq. (3)
end for
for  $cycle = 1$  to the  $MaxCycles$  do
    for  $i=1$  to the Nectar number do
        Randomly search for the new nectar  $V_{id}$ ,
        according to Eq. (7) and evaluate it
    end for
    for  $i = 1$  to the nectar number do
        Calculate the probability of the nectar  $X_i$ 
        provided by the employed bees to be
        followed, using Eq. (4)
    end for
     $i = 1, t = 1$ 
    While  $t \leq Limit$  do
        if  $rand < P_i$  then
             $t = t + 1$ 
            Determine whether the local search of the
            nectar is the local optimal solution
            founded by the employed bees, using
            Eq. (8)
            Produce the new solution  $V_i$  by using Eq. (9),
            with the onlooker bees selecting the
            solution  $X_i$  depending on  $P_i$  and
            evaluating
        end if
         $i = i + 1$ 
        if  $i >$  the Nectar number then
             $i = 1$ 
        end if
    end while
    Determine if the algorithm satisfies the termination
    condition
    if it hasn't been updated then
        Find the new nectar again using Eq. (5)
    end if
end for

```

Figure 2 Pseudo-code of the EUBCOA algorithm

3.1 The Improvement of the Search Factor

The adaptive factor φ of the traditional ABC algorithm is the random number of $[-1, 1]$. The search range is uncontrollable. At the same time, it increases the time in which the employed bees find the local optimal solution and the onlooker bees find the global optimal solution. It also makes the convergence rate slow. In view of this, in order to have different adjustments at different times, the algorithm can find the target area better [22]. At the same time without affecting the randomness of the range of the original adaptive factor $[-1, 1]$, this paper proposes the search factor u which plays a guiding role in the search trend of the position of the nectar [23]. We set the adjustment formula of the search factor u as follows:

$$u = e^{-k \times fit_i} \tag{6}$$

where fit_i is the fitness of the nectar after the last iteration of the employed bees. Optimization criteria which are used to evaluating fitness are to be chosen in applying optimization method. k is an adjustable random number. The exponential function is used to expand the search range in the early iteration. After introducing the search factor u , the updated nectar position is as follows:

$$V_{id} = X_{id} + u \times \varphi \times (X_{id} - X_{jd}) \tag{7}$$

It can be seen that the search range of the algorithm depends on the adaptability of different nectar. The corresponding search range can be improved at different times and the efficiency of the algorithm can be improved. At the same time, the search range decreases with the increase of fitness. In the early stage of the iteration, if the fitness of the nectar is smaller and the objective function is larger, the distance between the nectar and the local optimal solution is larger. Then the search range is expanded and the probability of finding the optimal solution is increased. In the latter part of the iteration, the fitness of the nectar is larger and the distance between the nectar and the local optimal solution is smaller. Then the search range is reduced and the rate of finding the global optimal solution is increased. The global fast development capability and the local fine mining ability are effectively balanced.

3.2 The Selection Strategy of the Onlooker Bees based on Local Optimal Solution

The ABC algorithm includes the global search of the employed bees and the local search of the onlooker bees. The number of bees used is equal to the number of food sources since each employed bee is associated with one and only one food source. When the onlooker bees update the nectar, they select the position of the nectar which is provided by the employed bees in accordance with the roulette way [24]. It will also reduce the efficiency of the algorithm. In order to make the onlooker bees find the global optimal solution quickly, the employed bees exchange the nectar position information, which is updated to the local optimal solution position. The onlooker bees search surrounding local optimal solution. In this way, the convergence speed and accuracy of the algorithm are improved. Therefore, we introduce the local optimal solution. The formula is:

$$X_{P(i),d} = \begin{cases} X_{i+1} & fit_i(X_{i+1}) \geq fit_i(X_{P(i)}) \\ X_{i,d} & fit_i(X_{i+1}) < fit_i(X_{P(i)}) \end{cases} \tag{8}$$

In Eq. (8), the method of calculating the fitness is used as the judgment condition, $X_{P(i),d}$ is the local optimal solution quoted by the onlooker bees when it updates the nectar. fit_i is the fitness value of the nectar. If the fitness of the new nectar is found by the employed bees that it is greater than the fitness of the local optimal solution in the previous iteration, the new nectar is updated as the local optimal solution. If the fitness of the new nectar is less than the fitness of the local optimal solution in the previous iteration, the original nectar is set as the local optimal solution. After finding the local optimal solution, the new updated formula of the onlooker bees is:

$$V_{id} = X_{P(i),d} + u \times \varphi \times (X_{id} - X_{jd}) \tag{9}$$

In ABC algorithm, the position of onlooker bees is updated after the employed bees exchange the information. In Eq. (9), the position of the nectar provided by the employed bees is updated according to the employed bees

with local optimums. At the same time, the part of the formula is the same as the employed bees, and the search factor u is added on the basis of the adaptive factor φ . When the optimal solution is close to the global optimal solution, the fitness of the nectar becomes larger, the search factor u becomes smaller and the variation is smaller. The range becomes smaller which increases the probability of finding the global optimal solution.

3.3 The Specific Steps of the Improved Algorithm

Step 1: Algorithm initialization. Initialize the number of population, the number of iterations, the search factor u and the new parameters of the local optimal solution $X_{P(i),d}$. The position of the nectar is initialized according to the assigned parameters.

Step 2: The employed bees according to Eq. (7) search for the new nectar in the search space and calculate the fitness of the new nectar. If it is better than the original nectar, then update the nectar, otherwise retain the original nectar.

Step 3: The onlooker bees choose the employed bees with roulette way, according to Eq. (8) to determine whether the local search of the nectar is the local optimal solution founded by the employed bees.

Step 4: If it is the local optimal solution, the onlooker bees according to the Eq. (9) search the local nectar and update the nectar. Then calculate the fitness and compared with the original fitness, finally update the global optimal position.

Step 5: After several iterations, if the fitness of the nectar has still no improvement, then the employed bees become the detective bees and continue to search for the new nectar.

4 EXPERIMENT AND ANALYSIS

In this experiment, we select ten benchmark functions [25], which are shown in Tab. 1. Among them f_1, f_2, f_3, f_4, f_5 are the single-mode benchmark functions, as shown in Fig. 3. $f_6, f_7, f_8, f_9, f_{10}$ are the multi-mode Benchmark

functions with a large number of local extreme points, as shown in Fig. 3.

4.1 Experimental Design

In order to verify the performance of the improved algorithm, the EUBCOA algorithm first compares with meta heuristic algorithms such as the PSO algorithm, the CS algorithm and the ABC algorithm. And then the EUBCOA algorithm compares performance with improved ABC algorithms such as the HABC algorithm [26] and the CABC algorithm [27]. The experimental parameters are set as follows: the maximum number of iterations of the algorithm is 1000, the function dimension defaults to 50, and the k value of the slope of the convergence rate is 6. The experimental environment is Intel (R) Core (TM) i5-4590S 3.00 GHz, 8 GB memory, Windows7 Ultimate 32-bit operating system, the programming software is the MATLAB R2010a.

4.2 Experimental Results and Analysis

Tab. 2 shows the results of the ten benchmark functions with 50 times the number of iterations of 1000 times and the population size of 30. The experimental results show that the best results of the EUBCOA algorithm are superior to the other three algorithms in the f_1, f_2, f_5 and f_4 functions for the single-mode benchmark function. The searching precision of the f_3 function is slightly stronger than that of the ABC algorithm. The multi-modal benchmark function has large local extremum. The EUBCOA algorithm improves the searching precision significantly in the functions f_6, f_7, f_8 and f_9 . The EUBCOA algorithm is better than the ABC algorithm in the f_{10} function, and the optimal mean and mean square error are also superior to the other three algorithms. On the function f_2 , the EUBCOA algorithm has an accuracy rating of 10^{-6} , while the ABC algorithm, the CS algorithm and the PSO algorithms are $10^{-4}, 10^2$ and 10^{-1} , respectively. The advantages are also obvious in the functions f_1, f_3 and f_9 .

Table 1 Benchmark functions

	Function Name	Formula	Feasible Region	Optimal value
Single mode	SumOfDifferent	$\sum_{i=1}^n x_i ^{i+1}$	$[-10, 10]$	0
	SumSquares	$\sum_{i=1}^n ix_i^2$	$[-10, 10]$	0
	Sphere	$\sum_{i=1}^n x_i^2$	$[-100, 100]$	0
	Rosenbrock	$\sum_{i=1}^{n-1} (100(x_{i+1} - x_i)^2 + (x_i - 1)^2)$	$[-1, 1]$	0
	Quartic	$\sum_{i=1}^n ix_i^4 + \text{rand}()$	$[-100, 100]$	0
Multi mode	Alpine	$\sum_{i=1}^n x_i \sin(x_i) + 0.1x_i $	$[-10, 10]$	0
	Rastrigin	$\sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	0

Table 1 Benchmark functions (continuation)

	Function Name	Formula	Feasible Region	Optimal value
Multi mode	Ackley	$-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2} - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e\right)$	$[-50, 50]$	0
	AxisParallel	$\sum_{i=2}^n ix_i^2$	$[-1, 1]$	0
	Pathologic	$\sum_{i=1}^{n-1} \left(0.5 + \frac{\sin^2\left(\sqrt{100x_i^2 + x_{i+1}^2}\right) - 0.5}{1 + 0.001(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2)}\right)^2$	$[-5.12, 5.12]$	0

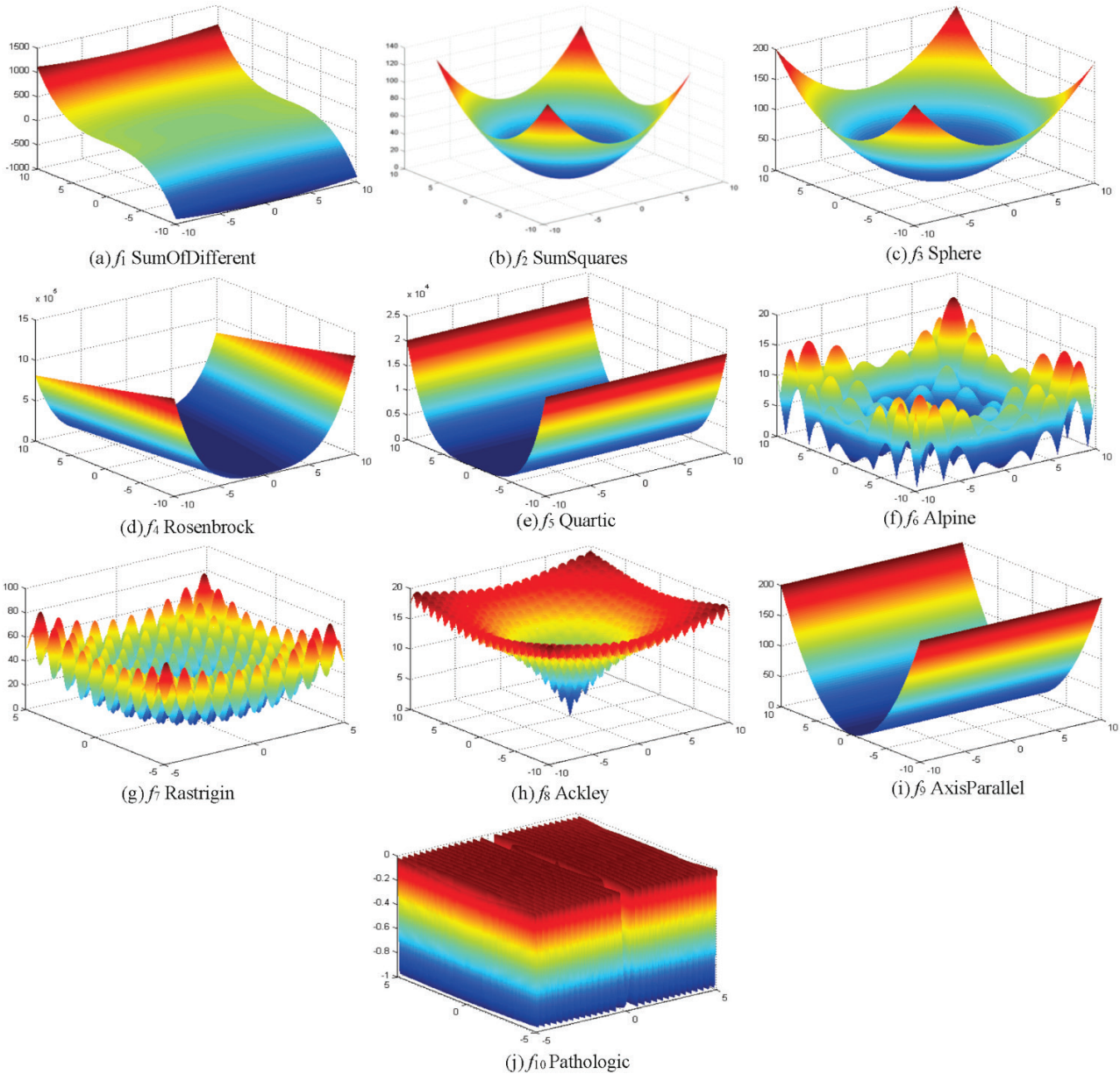


Figure 3 Visualization of 10 benchmark functions: $f_1 \sim f_5$ in the first column, $f_6 \sim f_{10}$ in the second column

For the single-mode benchmark function, the experimental results of Tab. 3 show that the average searching precision of the EUBCOA algorithm is 10^{-6} . The average searching precision of the ABC algorithm, the CS algorithm and the PSO algorithm is about 10^{-4} , 10^{-3} and 10^4 . For the multi-mode benchmark function, the average searching precision of the EUBCOA algorithm is 10^{-6} . The average searching precision of the ABC algorithm, the CS

algorithm and the PSO algorithm is about 10^{-3} , 10^{-1} and 10^2 respectively. It is shown that the EUBCOA algorithm has better performance than the other three algorithms in terms of optimization accuracy. The experimental data in Tab. 2 and Tab. 3 show that the search factor u can effectively improve the local search ability and the global search ability of the ABC algorithm.

Table 2 Results of meta heuristic algorithms for benchmark functions (size pop = 30)

Function	Algorithm	Minimum	Maximum	Mean	Std
SumOfDifferent	CS	2.3220e ⁺¹⁰	2.3891e ⁺¹⁰	2.3297e ⁺¹⁰	2.3339e ⁻¹⁷
	PSO	6.8741e ⁺⁸	6.8801e ⁺⁸	6.8765e ⁺⁸	6.8767e ⁻²²
	ABC	1.1074e ⁻⁴	1.4480e ⁺²⁷	1.0012e ⁺²⁵	4.4889e ⁺⁵¹
	EUBCOA	2.9721e ⁻⁶	8.8191e ⁺²⁶	3.8528e ⁺²⁴	3.1315e ⁺⁴⁹
SumSquares	CS	1.0369e ⁻¹	1.5221e ⁺⁶	1.3691e ⁺⁵	1.1433e ⁺¹⁰
	PSO	5.1801e ⁺²	9.1800e ⁺²	8.1800e ⁺²	1.1802e ⁻¹¹
	ABC	2.7668e ⁻⁴	8.5316e ⁺¹	1.6599e ⁻¹	3.0244e ⁻¹
	EUBCOA	3.2579e ⁻⁶	3.1097	3.4627e ⁻⁴	9.7341e ⁻⁵
Sphere	CS	6.4465e ⁻³	3.7298e ⁺⁴	1.5690e ⁺²	6.1758e ⁺¹
	PSO	2.1853e ⁺¹	3.5002e ⁺¹	2.6873e ⁻¹	2.6658e ⁻⁸
	ABC	1.3762e ⁻⁴	1.1367e ⁺⁵	1.4833e ⁺¹	3.5069e ⁻³
	EUBCOA	1.2609e ⁻⁷	2.0729e ⁺⁶	9.8480e ⁻²	8.1004e ⁻²
Rosenbrock	CS	1.0534e ⁺⁶	9.0548e ⁺¹⁰	4.6302e ⁺⁸	2.5991e ⁺⁹
	PSO	4.4886e ⁺³	1.1692e ⁺⁴	6.1646e ⁺³	1.6015e ⁻⁹
	ABC	2.1457e ⁺¹	1.4845e ⁺⁸	8.4955e ⁺³	8.9799e ⁺¹
	EUBCOA	3.1957	8.1857e ⁺⁸	5.8035e ⁺⁴	3.5766e ⁺¹
Quartic	CS	2.8100e ⁺¹	2.8010e ⁺⁹	1.7270e ⁺³	4.8900e ⁻⁸
	PSO	1.3881e ⁺²	1.3869e ⁺³	3.7563e ⁺²	2.9642e ⁻⁵
	ABC	6.5093e ⁻¹	5.1520e ⁺²	8.2211e ⁻¹	1.0199
	EUBCOA	6.6477e ⁻¹	6.8951e ⁺³	9.6984e ⁻¹	1.0578e ⁻¹
Alpine	CS	1.0768e ⁺¹	2.2662e ⁺¹	1.7575e ⁺¹	2.7570e ⁻⁶
	PSO	1.5859e ⁺¹	2.4069e ⁺¹	1.5884e ⁺¹	2.3281e ⁻¹³
	ABC	3.7329e ⁻⁴	2.5008e ⁻¹	8.0248e ⁻⁴	8.3243e ⁻⁴
	EUBCOA	2.9844e ⁻⁵	2.5018e ⁻³	8.5386e ⁻⁵	8.1903e ⁻⁵
Rastigin	CS	1.2172e ⁺¹	2.3544e ⁺²	1.6338e ⁺¹	2.3305e ⁺¹
	PSO	3.8304e ⁺²	4.7834e ⁺²	3.9576e ⁺²	6.0284e ⁻¹²
	ABC	1.0189	8.6148e ⁺²	1.0517e ⁻¹	2.9886
	EUBCOA	5.5630e ⁻¹	4.5494e ⁺²	5.8488e ⁻¹	5.5216e ⁻¹
Ackley	CS	1.9269e ⁺¹	2.1992e ⁺¹	1.9690e ⁺¹	5.0821e ⁻⁹
	PSO	4.3076	4.8153	4.4503	5.1540e ⁻¹⁴
	ABC	6.8953	2.1432e ⁺¹	9.2180	4.9868e ⁻¹
	EUBCOA	3.0760	2.0976e ⁺¹	3.3710	2.0328e ⁻¹
AxisParallel	CS	1.3770e ⁻¹	2.7242e ⁺⁶	1.9553e ⁻¹	4.4399e ⁻³
	PSO	5.0463e ⁺²	8.9563e ⁺²	5.9876e ⁺²	1.0752e ⁺²
	ABC	1.6938e ⁻¹	6.7334e ⁺⁷	4.0535e ⁺⁶	1.5105e ⁺¹⁴
	EUBCOA	1.2893e ⁻³	1.0673e ⁺⁵	7.4045e ⁺³	3.8780e ⁺⁸
Pathologic	CS	5.1864	6.3765	5.6097	2.5958e ⁻¹
	PSO	1.0479e ⁺¹	1.5518e ⁺¹	1.1538e ⁺¹	1.5284e ⁻¹³
	ABC	3.1783e ⁻⁴	6.5971e ⁻²	7.1513e ⁻¹	5.2806
	EUBCOA	1.6754e ⁻⁶	1.4107e ⁻¹	4.32714e ⁻⁴	3.94537e ⁻²

Table 3 Results of meta heuristic algorithms for benchmark functions (size pop = 50)

Function	Algorithm	Minimum	Maximum	Mean	Std
SumOfDifferent	CS	1.5545e ⁺¹⁰	7.8973e ⁺¹⁰	3.3752e ⁺¹⁰	1.4203e ⁻¹⁵
	PSO	6.8763e ⁺⁸	6.8785e ⁺⁸	6.8776e ⁺⁸	6.8764e ⁻²²
	ABC	1.1122e ⁻⁴	1.4245e ⁺²⁷	6.9543e ⁺²⁵	3.6711e ⁺⁵²
	EUBCOA	1.8781e ⁻⁶	2.6612e ⁺²⁶	9.5478e ⁺²⁴	8.5935e ⁺⁵⁰
Sum Squares	CS	1.5023e ⁻¹	3.4874e ⁺⁶	2.2212e ⁺⁵	1.5170e ⁺¹¹
	PSO	5.1802e ⁺²	9.1801e ⁺²	8.1801e ⁺²	1.1801e ⁻¹¹
	ABC	8.8527e ⁻⁴	1.9042e ⁺¹	3.3102e ⁻¹	5.8133e ⁻¹
	EUBCOA	7.2163e ⁻⁶	6.0019	9.6173e ⁻⁵	1.8635e ⁻³
Sphere	CS	1.2233e ⁻³	3.4691e ⁻³	2.2354e ⁻³	5.6150e ⁻¹
	PSO	2.4455e ⁺¹	3.9995e ⁺¹	2.4484e ⁺¹	3.6255e ⁻⁶
	ABC	2.0128e ⁻⁴	1.2045e ⁺⁵	2.5096e ⁺²	3.5603e ⁺³
	EUBCOA	5.4852e ⁻⁶	7.2470e ⁺⁶	2.6285e ⁻¹	2.4380e ⁺²
Rosenbrock	CS	1.8473e ⁺⁶	3.2802e ⁺¹⁰	2.3900e ⁺⁸	1.8136e ⁺⁹
	PSO	6.6631e ⁺⁴	7.5631e ⁺⁴	6.9749e ⁺⁴	1.0919e ⁻¹⁶
	ABC	1.0478e ⁺²	7.8932e ⁺⁸	5.9625e ⁺³	7.8589e ⁺²
	EUBCOA	1.5700e ⁻¹	1.1125e ⁺⁸	7.9642e ⁻¹	8.3206
Quartic	CS	1.5473e ⁺¹	8.2820e ⁺⁹	1.7826e ⁺⁶	5.7095e ⁺²
	PSO	7.1976e ⁺²	1.1976e ⁺³	5.3140e ⁺²	2.8481e ⁻⁵
	ABC	6.3694e ⁻¹	5.3636e ⁺³	8.4866e ⁻¹	1.0061e ⁺²
	EUBCOA	6.2030e ⁻¹	5.3865e ⁺²	8.0073e ⁻¹	4.8343e ⁺¹
Alpine	CS	1.3717e ⁺¹	2.4244e ⁺¹	1.9006e ⁺¹	2.1526e ⁻⁵
	PSO	1.8351e ⁺¹	2.4588e ⁺¹	1.8360e ⁺¹	6.7535e ⁻¹³
	ABC	5.9697e ⁻⁴	7.8037e ⁻¹	8.6586e ⁻⁴	1.9698e ⁻⁵
	EUBCOA	8.3821e ⁻⁵	3.6633e ⁻³	5.1579e ⁻⁴	1.0522e ⁻⁴
Rastigin	CS	1.3635e ⁺¹	2.2254e ⁺²	1.7736e ⁺¹	1.9176e ⁺¹
	PSO	3.9174e ⁺²	4.6013e ⁺²	4.1256e ⁺²	5.0947e ⁻¹⁰
	ABC	4.0214	1.3009e ⁺²	6.8772e ⁺¹	3.6273e ⁺¹
	EUBCOA	8.5048e ⁻¹	5.4956e ⁺²	7.2364	5.4842e ⁺¹
Ackley	CS	3.5854e ⁺¹	5.3302e ⁺¹	3.6115e ⁺¹	2.5304e ⁻⁸
	PSO	4.1371	4.8951	4.7396	7.8198e ⁻¹²
	ABC	8.8657	5.9001e ⁺¹	9.6196	4.5612
	EUBCOA	5.2730	3.2795e ⁺¹	8.0201	8.7648

Table 3 Results of meta heuristic algorithms for benchmark functions (size pop = 50) (continuation)

Function	Algorithm	Minimum	Maximum	Mean	Std
AxisParallel	CS	$7.3670e^{-1}$	$3.7252e^{+5}$	$9.2615e^{-1}$	$3.4399e^{-2}$
	PSO	$4.0363e^{+2}$	$8.6332e^{+2}$	$4.8507e^{+2}$	$5.0463e^{+2}$
	ABC	$1.8403e^{-1}$	$2.6334e^{+7}$	$4.0223e^{+6}$	$3.2175e^{+12}$
	EUBCOA	$1.7846e^{-3}$	$8.6184e^{+5}$	$7.6045e^{+5}$	$3.6780e^{+9}$
Pathologic	CS	5.7712	6.5016	5.8456	$7.3800e^{-1}$
	PSO	$1.2548e^{+1}$	$1.8411e^{+1}$	$1.6508e^{+1}$	$8.5307e^{-5}$
	ABC	$2.1570e^{-4}$	$1.7690e^{-2}$	$3.3800e^{-1}$	$4.7387e^{+1}$
	EUBCOA	$6.8285e^{-6}$	$7.3138e^{-1}$	$2.6933e^{-3}$	$1.5326e^{-1}$

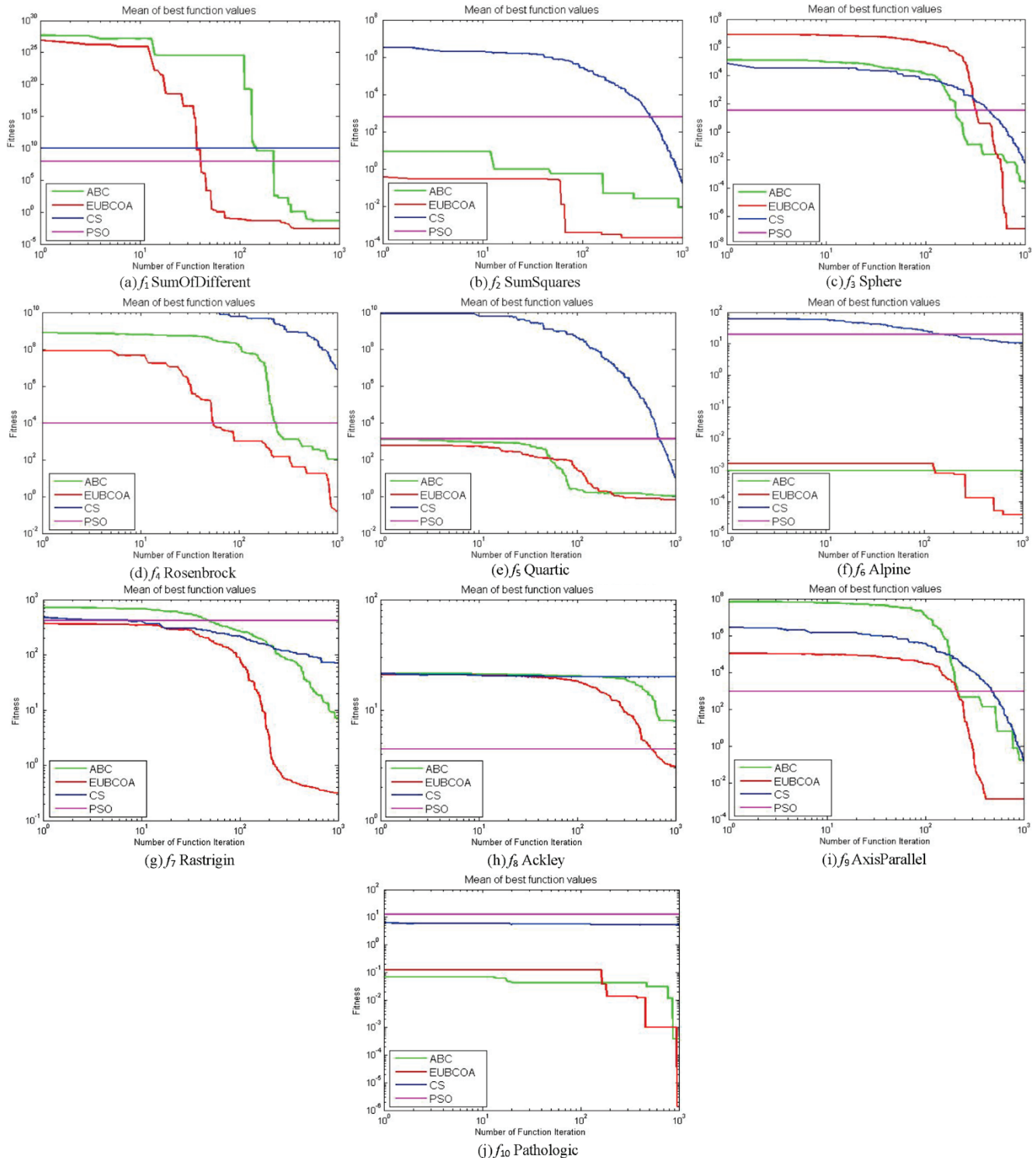


Figure 4 Convergence curves of 10 benchmark functions

Fig. 4 compares the search routes of the EUBCOA algorithm, the ABC algorithm, the CS algorithm and the PSO algorithm. It is verified that the EUBCOA algorithm

can improve the convergence speed, convergence precision and the performance of the ABC algorithm in the number of iterations.

Fig. 4 is an experimental result of the optimization of the function by the EUBCOA algorithm, the ABC algorithm, the CS algorithm and the PSO algorithm. For the convenience of observation, the output value is taken as the base 10 logarithm. It can be seen from the experimental results that the EUBCOA algorithm is superior to the other three algorithms in convergence performance. When the two-dimensional functions f_1 , f_2 and f_6 are optimized, the convergence level reaches 10^{-6} , while the other three algorithms have poor convergence performance. When the ABC algorithm optimizes the f_4, f_5, f_7 and f_8 functions, the convergence result is greater than 10^0 . When the CS algorithm optimizes f_1, f_4 functions, the convergence result is greater than 10^6 . When the PSO

algorithm optimizes the f_1 function, the convergence result is greater than 10^8 . The convergence speed of the EUBCOA algorithm is better than that of the other three algorithm algorithms. For example, from the convergence curve of the f_3 function, the convergence accuracy of the EUBCOA algorithm can reach 10^{-7} or less in 50 iterations. It can be seen from the convergence curves of the f_2, f_6 and f_{10} functions that after the EUBCOA algorithm is iterated to 100 times, the optimal value does not change, and the convergence curve is horizontal. But after mutating, the algorithm quickly jumps out of the local extremum and continues to find the global optimal value. Therefore, the introduction of the EUBCOA algorithm can effectively improve the convergence speed and convergence accuracy.

Table 4 Results of improved ABC algorithms for benchmark functions (size pop = 30)

Function	Algorithm	Minimum	Maximum	Mean	Std
SumOfDifferent	ABC	4.8973e ⁻⁴	3.2462e ⁺²⁷	3.6177e ⁻³	5.3264e ⁻³
	CABC	1.1627e ⁻⁵	2.9113e ⁺²⁷	2.6248e ⁻⁴	5.3262e ⁻³
	HABC	1.9681e ⁻⁴	3.3140e ⁺²⁸	3.6104e ⁻³	4.8275e ⁻³
	EUBCOA	1.4855e ⁻⁸	2.5995e ⁺²⁷	3.9512e ⁻⁶	3.4987e ⁻⁵
SumSquares	ABC	1.8200e ⁻²	2.3190e ⁺¹	1.8788e ⁻²	1.5451e ⁺¹
	CABC	1.9072e ⁻⁴	6.2221e ⁺¹	7.4847e ⁻⁴	1.5498e ⁺²
	HABC	1.0052e ⁻⁵	3.1719e ⁺¹	6.0299e ⁻⁵	7.1020e ⁻⁴
	EUBCOA	1.2120e ⁻⁶	3.1097e ⁺¹	1.6347e ⁻⁵	1.9028e ⁻⁵
Sphere	ABC	1.7000e ⁻³	1.1555e ⁺²	1.3264e ⁻²	2.3154e ⁻¹
	CABC	9.0723e ⁻³	1.1029e ⁺⁵	2.5475e ⁻²	3.6497e ⁻³
	HABC	1.5000e ⁻⁴	1.2554e ⁺⁵	1.3724e ⁻³	2.1318e ⁻⁴
	EUBCOA	6.2450e ⁻⁶	1.3294e ⁺³	4.2788e ⁻⁴	6.0505e ⁻²
Rosenbrock	ABC	9.1196e ⁺¹	5.6351e ⁺⁸	4.6302e ⁺²	2.5990e ⁺¹
	CABC	4.8627e ⁺¹	1.2188e ⁺⁶	6.1646e ⁺³	1.6015e ⁺¹
	HABC	5.9848e ⁺¹	1.4161e ⁺⁴	6.5007e ⁺¹	6.8467e ⁺¹
	EUBCOA	5.8000e ⁻³	3.3506e ⁺⁶	6.3112	1.0790
Quartic	ABC	1.7952e ⁺¹	5.2534e ⁺³	2.6542e ⁺¹	4.8900e ⁺¹
	CABC	7.0817e ⁺¹	3.1051e ⁺⁴	9.7563e ⁺¹	2.9642e ⁺¹
	HABC	4.1579e ⁺¹	7.8184e ⁺²	9.1631e ⁺¹	7.4687e ⁻¹
	EUBCOA	4.3550e ⁻¹	4.3979e ⁺²	8.5436e ⁻¹	7.8294e ⁻¹
Alpine	ABC	2.3000e ⁻³	1.6020e ⁻¹	2.6530e ⁻³	2.6450
	CABC	3.7315e ⁻⁴	1.2950e ⁻¹	1.5884e ⁻³	1.5540
	HBC	8.9290e ⁻⁵	1.9800e ⁻²	9.0826e ⁻⁵	2.6118e ⁻⁵
	EUBCOA	2.3750e ⁻⁶	1.6600e ⁻¹	3.5025e ⁻⁵	4.5442e ⁻⁵
Rastigin	ABC	1.7952e ⁺¹	5.2534e ⁺³	4.8859e ⁺¹	7.5497e ⁻¹
	CABC	3.4079e ⁺¹	7.7813e ⁺²	3.8462e ⁺¹	5.9454e ⁻¹
	HABC	4.1579e ⁺¹	7.8184e ⁺²	4.2436e ⁺¹	1.2114
	EUBCOA	4.3550e ⁻¹	4.2979e ⁺²	4.8859e ⁻¹	6.5351e ⁻²
Ackley	ABC	1.9670e ⁺¹	2.1505e ⁺¹	1.9885e ⁺¹	1.5246e ⁻²
	CABC	1.7248e ⁺¹	2.1380e ⁺¹	1.7289e ⁺¹	1.5484e ⁻²
	HABC	1.4512e ⁺¹	2.1348e ⁺¹	1.4860e ⁺¹	2.1461e ⁻¹
	EUBCOA	6.8953e ⁻¹	2.1465e ⁺¹	1.2180	4.9868
AxisParallel	ABC	1.4800e ⁻²	7.8557e ⁺⁵	1.6592e ⁻²	3.7925e ⁻³
	CABC	5.1890e ⁻¹	9.0034e ⁺³	5.7696e ⁻¹	1.3659e ⁺²
	HABC	4.8200e ⁻²	2.6085e ⁺²	4.8596e ⁻²	9.7129e ⁻³
	EUBCOA	1.4427e ⁻⁴	8.3269e ⁺⁵	1.0222e ⁻²	1.5733e ⁻²
Pathologic	ABC	1.1000e ⁻³	8.7800e ⁻²	2.8540e ⁻³	5.6421e ⁻¹
	CABC	1.8261e ⁻⁴	1.5430e ⁻¹	1.6236e ⁻³	1.5498e ⁻⁵
	HABC	3.1783e ⁻⁵	3.7300e ⁻²	3.2550e ⁻⁴	3.2009e ⁻⁴
	EUBCOA	2.3083e ⁻⁶	1.5730e ⁻¹	6.4377e ⁻⁵	7.7438e ⁻⁶

Tab. 4 shows the performance comparison between the EUBCOA algorithm and other improved versions of the ABC algorithm with a population size of 30, a maximum number of iterations of 1000, and a function dimension of 50. It can be seen from Tab. 4 that for the single-mode Benchmark function, the EUBCOA algorithm outperforms the other three algorithms, and the minimum average optimization accuracy is obtained on the five functions. Especially for the functions f_1 and f_2 , the mean precision and mean square error of the EUBCOA algorithm are significantly better than other algorithms, indicating that the EUBCOA algorithm is better than other comparison algorithms in solving the single-pole problem. In the multi-

mode Benchmark function with more local extremum, the optimization average accuracy of the EUBCOA algorithm on the function f_{10} reaches 10^{-5} and the optimal mean square error accuracy is 10^{-6} , which is much better than the algorithm for comparison. It shows that the EUBCOA algorithm has stronger global search ability than the improved algorithm when solving the multi-extreme problem and it can achieve better optimization effect in the limited iteration. Further research shows that for the ten Benchmark functions, the average optimization accuracy of the EUBCOA algorithm reaches 10^{-6} , while the ABC, HABC and CABC algorithms are 10^{-3} , 10^{-5} and 10^{-4} , respectively. The above experimental results show that the

optimization performance of the EUBCOA algorithm is generally superior to other improved ABC algorithms.

Tab. 5 shows the performance comparison of the EUBCOA algorithm with other improved versions of the ABC algorithm with a population size of 50, a maximum number of iterations of 1000, and a function dimension of 50. It can be seen from Tab. 5 that the EUBCOA algorithm outperforms the other three algorithms for the single-mode Benchmark function. On the function f_1 , the EUBCOA algorithm has an accuracy rating of 10^{-6} , while the HABC and CABCO algorithms are 10^{-5} and 10^{-4} , respectively. The advantages are also obvious in the functions f_2, f_3 and f_4 . For the multi-modal Benchmark function, the optimization performance of the EUBCOA algorithm is also excellent. The average optimization accuracy of the function f_9

reaches 10^{-6} , which is obviously superior to the other three algorithms. On the five multi-mode Benchmark functions, the minimum optimization accuracy is obtained, indicating that the improved algorithm performs better overall when solving the multi-value optimization problem. Further research shows that for the ten Benchmark functions, the average optimization accuracy of the EUBCOA algorithm reaches 10^{-6} , while the ABC, HABC and CABCO algorithms are $10^{-1}, 10^{-4}$ and 10^{-3} , respectively. The above experimental results show that the optimization performance of the EUBCOA algorithm is generally superior to other improved ABC algorithms. It is further proved that the EUBCOA algorithm has a great improvement in convergence speed and optimization accuracy.

Table 5 Results of improved ABC algorithms for benchmark functions (size pop = 50)

Function	Algorithm	Minimum	Maximum	Mean	Std
SumOfDifferent	ABC	5.3000e ⁻³	7.5504e ⁺²⁵	2.3154e ⁻⁴	2.3339e ⁻³
	CABC	1.6143e ⁻⁵	1.6192e ⁺²⁶	6.3247e ⁻³	8.6249e ⁻³
	HABC	4.7557e ⁻⁴	1.8769e ⁺²¹	1.6543e ⁻³	1.6670e ⁻³
	EUBCOA	5.7664e ⁻⁶	7.5855e ⁺²⁵	6.3837e ⁻³	8.9829e ⁻³
SumSquares	ABC	3.3200e ⁻²	1.0819e ⁺¹	5.3691e ⁻²	1.1433e ⁻⁴
	CABC	1.7816e ⁻⁴	3.2368e ⁺¹	1.7865e ⁻³	1.6892e ⁺²
	HABC	1.7386e ⁻⁴	1.6329	1.7924e ⁻⁵	7.6090e ⁻⁵
	EUBCOA	1.0257e ⁻⁵	6.9005	1.7828e ⁻⁴	1.5757e ⁻⁴
Sphere	ABC	5.2467e ⁻⁴	1.3392e ⁺⁵	9.3221e ⁻⁴	5.3246e ⁻³
	CABC	8.3015e ⁻⁴	1.1700e ⁺⁵	4.3659e ⁻³	2.3335e ⁻³
	HABC	1.0000e ⁻³	1.0457e ⁺³	8.0222e ⁻³	3.3100e ⁻⁴
	EUBCOA	1.5708e ⁻⁷	1.2674e ⁺⁵	2.0803e ⁻³	3.5979e ⁻³
Rosenbrock	ABC	4.4780e ⁺¹	4.3910e ⁺⁸	4.5134e ⁺²	4.3284e ⁺¹
	CABC	4.7618e ⁺¹	1.3983e ⁺⁶	6.1646e ⁺³	1.6015e ⁺³
	HABC	4.9714e ⁺¹	1.8821e ⁺⁵	4.9915e ⁺²	9.8855e ⁺¹
	EUBCOA	1.2270e ⁻¹	4.0879e ⁺⁶	1.6446	1.3281e ⁺³
Quartic	ABC	8.1490e ⁻¹	1.4352e ⁺³	1.5554e ⁺¹	5.6249e ⁺¹
	CABC	6.1840e ⁻¹	8.5817e ⁺⁴	7.2985e ⁻¹	5.5231e ⁺¹
	HABC	6.9570e ⁻¹	1.4619e ⁺³	6.3722e ⁺¹	8.2631e ⁺¹
	EUBCOA	6.8090e ⁻¹	9.7386e ⁺⁴	7.7270e ⁺¹	4.8950e ⁺¹
Alpine	ABC	3.0081e ⁻⁴	6.3600e ⁻²	3.1187e ⁻⁴	1.5496e ⁻¹
	CABC	1.6000e ⁻³	1.2200e ⁻²	1.8551e ⁻³	1.2164
	HABC	3.1745e ⁻⁵	5.9943e ⁻⁴	4.5014e ⁻⁵	9.5184e ⁻⁶
	EUBCOA	1.4244e ⁻⁵	5.8500e ⁻²	2.5626e ⁻⁵	1.2556e ⁻⁵
Rastigin	ABC	3.7447e ⁺¹	7.9194e ⁺²	4.2356e ⁺¹	6.9426
	CABC	4.8415e ⁺¹	4.7087e ⁺³	8.9576e ⁺¹	6.0284e ⁺¹
	HABC	8.1225e ⁺¹	5.7413e ⁺³	9.5614e ⁺¹	5.4167e ⁺¹
	EUBCOA	4.1410e ⁻¹	4.2859e ⁺²	6.1686e ⁻¹	1.1413
Ackley	ABC	1.7766e ⁺¹	2.1508e ⁺¹	1.8789e ⁺¹	1.5685e ⁻³
	CABC	1.7637e ⁺¹	2.1400e ⁺¹	1.7869e ⁺¹	1.3264e ⁻⁴
	HABC	1.4732e ⁺¹	2.1364e ⁺¹	1.5487e ⁺¹	1.0678
	EUBCOA	5.4490e ⁻¹	2.1454e ⁺¹	2.1272	1.3983
AxisParallel	ABC	1.6900e ⁻²	5.7197e ⁺⁶	5.6895e ⁻²	6.3495e ⁻²
	CABC	5.8790e ⁻¹	7.2007e ⁺³	5.9876e ⁻²	1.3645e ⁻¹
	HABC	1.2800e ⁻²	3.0846e ⁺²	1.4214e ⁻²	2.0112e ⁻³
	EUBCOA	4.4927e ⁻⁶	7.8434e ⁺⁵	4.7642e ⁻⁴	7.3449e ⁻⁴
Pathologic	ABC	3.6000e ⁻³	1.5210e ⁻¹	5.6895e ⁻³	6.3495e ⁻²
	CABC	3.9738e ⁻⁴	1.7220e ⁻¹	9.8465e ⁻³	1.2165e ⁺¹
	HABC	9.9157e ⁻⁵	8.9500e ⁻²	3.2550e ⁻⁴	3.2009e ⁻⁴
	EUBCOA	2.9160e ⁻⁶	8.6300e ⁻²	5.8489e ⁻⁴	6.5255e ⁻⁴

In order to further illustrate the faster convergence speed of the EUBCOA algorithm, the optimization convergence curves of the ABC algorithm, the HABC algorithm and the CABCO algorithm on ten Benchmark functions are graphically displayed. The population size is 30, the maximum number of iterations is 1000, the function dimension is 50, and the number of optimization times is 50. It can be seen from Fig. 5 that on the functions f_2, f_6, f_9 and f_{10} , the convergence curve of the EUBCOA algorithm is more twisted, and the curve is stepped. The step of the curve increases with the increase of the number of iterations. The improved algorithm has strong ability to

jump out of the local extremum, so that the algorithm has stronger global survey capability in the later stage of optimization. At the same time, on the functions f_1, f_2, f_4, f_6 and f_7 , it is found that the convergence curve of the EUBCOA algorithm has a larger slope and more fluctuations in the curve than the HABC algorithm and the CABCO algorithm in the early stage of optimization. The EUBCOA algorithm has a faster convergence speed in the early iteration period, which improves the local search efficiency in the early stage of optimization. The above conclusions prove that the EUBCOA algorithm has

stronger global search ability and local search ability when solving multidimensional function optimization problems.

Fig. 6 is the experimental results of the EUBCOA algorithm and other improved ABC algorithms with a population size of 50, a maximum number of iterations of 1000, a function dimension of 50, and a number of optimization of 50. It can be seen from the experimental results that the EUBCOA algorithm is superior to the other three algorithms in convergence performance. When the two-dimensional functions f_1, f_9 and f_{10} are optimized, the convergence level reaches 10^{-6} , while the other three

algorithms have poor convergence performance. When the ABC algorithm optimizes the f_5, f_7 and f_8 functions, the convergence result is greater than 10^0 . When the HABC algorithm optimizes the f_1, f_2 functions, the convergence result is greater than 10^{-4} . When the CABC algorithm optimizes the f_6 function, the convergence result is greater than 10^{-3} . At the same time, when the EUBCOA algorithm optimizes the functions f_1, f_7 and f_{10} , the curve converges faster. Therefore, the introduction of the EUBCOA algorithm can effectively improve the convergence speed and convergence accuracy.

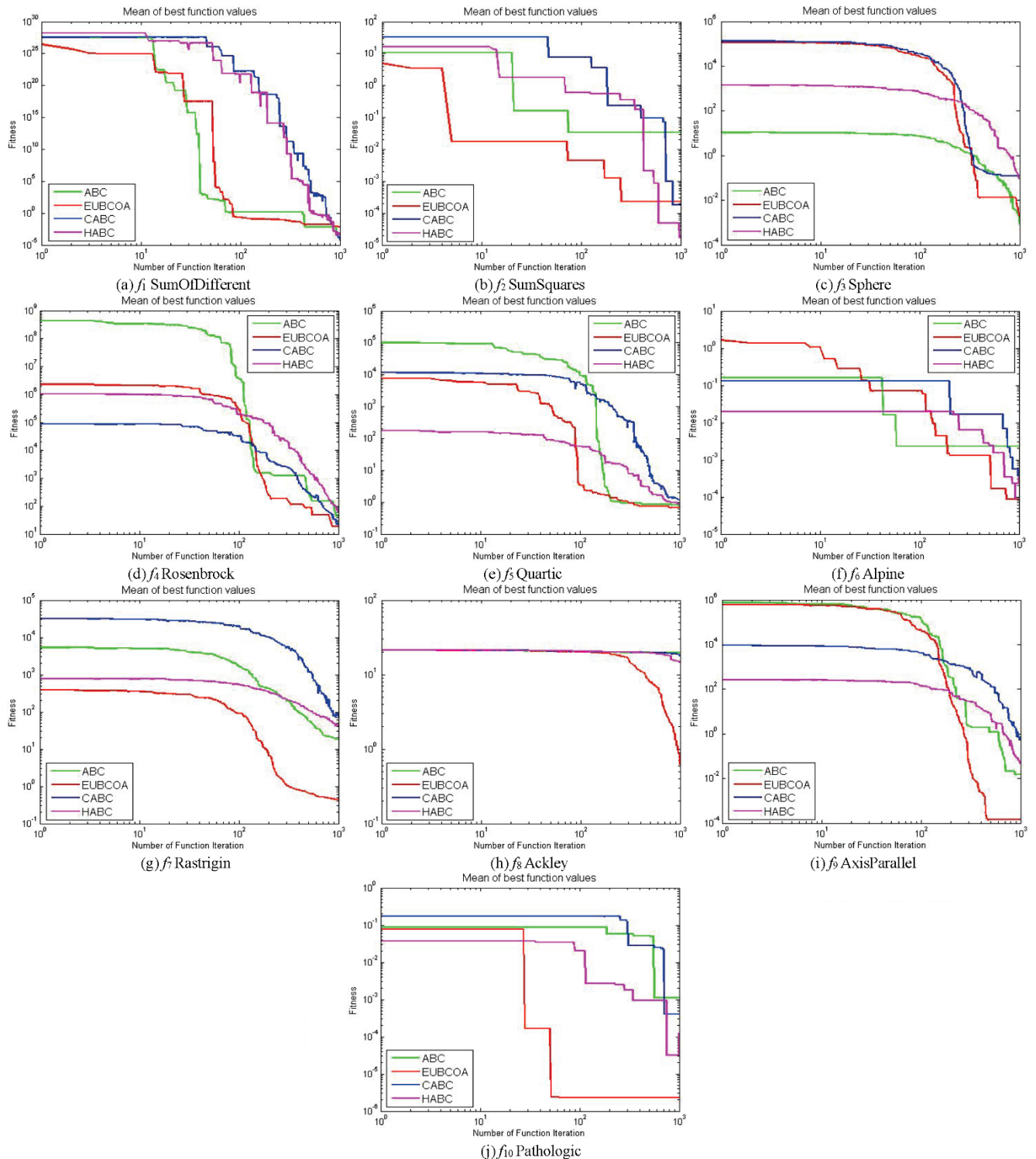


Figure 5 Convergence curves of 10 benchmark functions

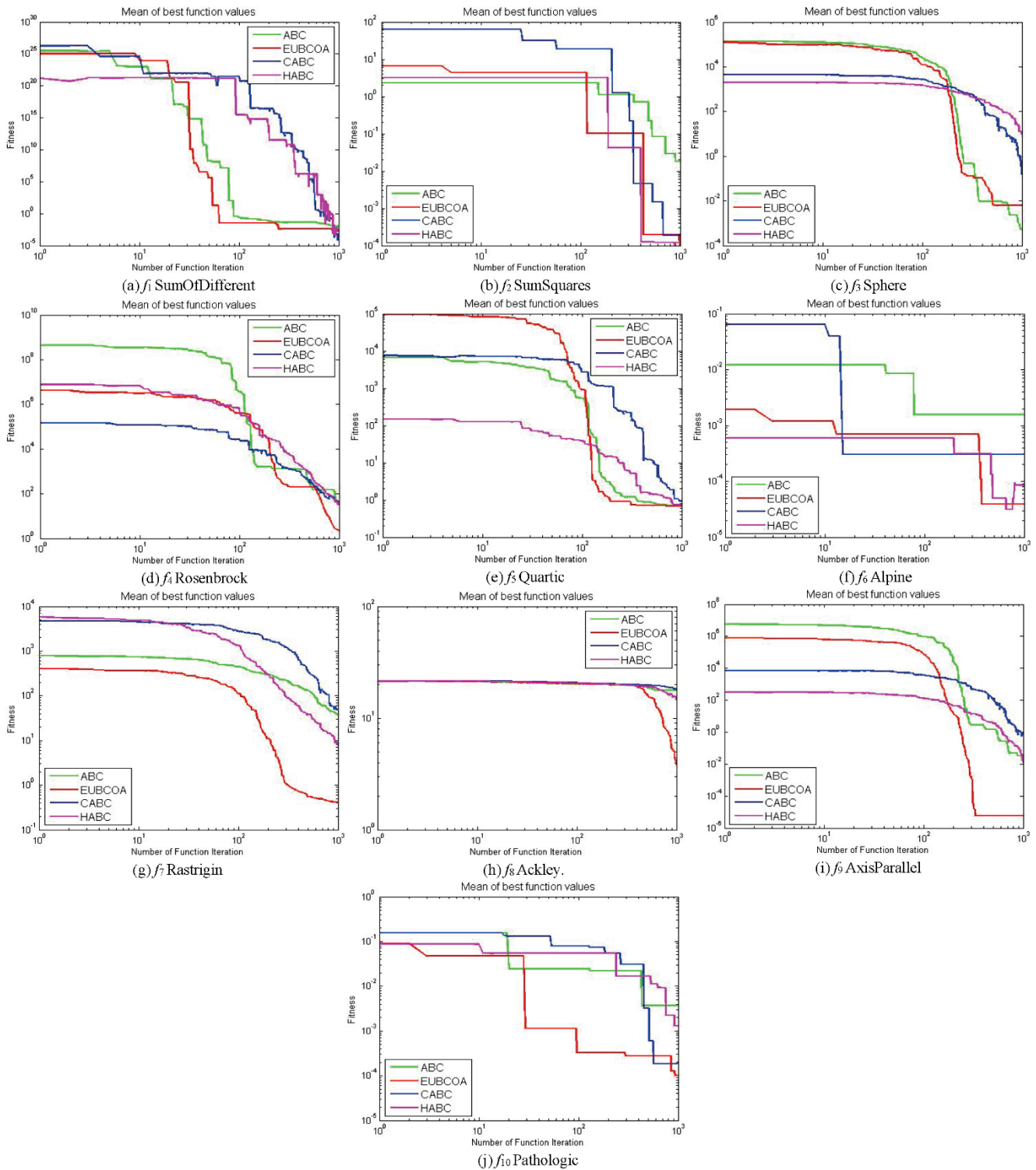


Figure 6 Convergence curves of 10 benchmark functions

5 CONCLUSION

This paper proposes an efficient universal bee colony optimization algorithm, which incorporates the search factor u and the selection strategy of the onlooker bees based on local optimal solution. Through the search factor u , the search range of the algorithm is more controllable, so that the search range of the employed bees is expanded, the search range of the onlooker bees is reduced and the convergence rate of the algorithm is improved. At the same time, the EUBCOA algorithm adopts the selection strategy of the onlooker bees based on local optimal solution, which

enhances the ability of the onlooker bees to search locally. It makes the onlooker bees find the global optimal solution faster and more accurately and improve the convergence precision of the algorithm. The experimental results show that the EUBCOA algorithm can significantly improve the convergence accuracy and convergence speed of the original algorithm, whether it is a single mode Benchmark function or a multi-mode Benchmark function. The EUBCOA algorithm can improve the searching range of the original ABC algorithm. However it has the problem that the search ability is unstable and cannot solve the problem of small part optimization. How to solve these

problems will become the focus of the next stage. In addition, more practical problems need to be solved by further analysis and verification of the improved algorithm.

Acknowledgements

The research was supported by the National Natural Science Foundation of China under grant No. 61472049, 61402193 and 61572225; the National Society Science Foundation of China under grant No. 15BGL090; the Jilin Development and Reform Commission Project; the Education Department Science and Technology Project No. JJKH20181048KJ; the Jilin Department of Science and Technology Research Project No. 20190302071GX.

6 REFERENCES

- [1] Dorigo, M., Maniezzo, V., & Coloni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1), 29-41. <https://doi.org/10.1109/3477.484436>
- [2] Kennedy, R. & Eberhart, J. (1995, November). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks IV*, (1000). https://doi.org/10.1007/978-0-387-30164-8_630
- [3] Yang, X. S. & Deb, S. (2009). Cuckoo Search via Lévy flights. *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. IEEE. https://doi.org/10.1007/978-0-387-30164-8_630
- [4] Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*, (200). Technical report-tr06, Erciyes University, engineering faculty, computer engineering department.
- [5] Pan, W. T. (2012). A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems*, 26, 69-74. <https://doi.org/10.1016/j.knsys.2011.07.001>
- [6] Qin, Q., Cheng, S., Li, L. I., & Shi, Y. (2014). Artificial bee colony algorithm: a survey. *CAAI Transactions on Intelligent Systems*, 9(2), 127-135. <https://doi.org/10.3969/j.issn.1673-4785.201309064>
- [7] Karaboga, D. & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, 214(1), 108-132. <https://doi.org/10.1016/j.amc.2009.03.090>
- [8] Karaboga, D. & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied soft computing*, 8(1), 687-697. <https://doi.org/10.1016/j.asoc.2007.05.007>
- [9] Tatomir, B. & Rothkrantz, L. (2004, September). Dynamic routing in mobile wireless networks using ABC-AdHoc. *International Workshop on Ant Colony Optimization and Swarm Intelligence*, 334-341. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-28646-2_32
- [10] Tusi, Y. & Chung, H. Y. (2016, August). Using ABC andRRT algorithms to improve mobile robot path planning with danger degree. In *Future Generation Communication Technologies (FGCT), 2016 Fifth International Conference on*, 21-26. <https://doi.org/10.1109/FGCT.2016.7605068>
- [11] Ercin, Ö. & Coban, R. (2012). Identification of linear dynamic systems using the artificial bee colony algorithm. *Turkish Journal of Electrical Engineering & Computer Sciences*, 20(Sup. 1), 1175-1188. <https://doi.org/10.3906/elk-1012-956>
- [12] Ata, B. & Coban, R. (2015). Artificial bee colony algorithm based linear quadratic optimal controller design for a nonlinear inverted pendulum. *International Journal of Intelligent Systems and Applications in Engineering*, 3(1), 1-6.
- [13] ÇOBAN, R. & ERÇİN, Ö. (2012). Multi-objective Bees Algorithm to Optimal Tuning of PID Controller. *Çukurova Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 27(2), 13-26.
- [14] Dzyabchenko, A. V. (1987). Theoretical structures of crystalline benzen. VI. Global search in the bisystemic structural class. *Journal of Structural Chemistry*, 28(6), 862-869. <https://doi.org/10.1007/BF00745258>
- [15] Min, X., Xu, X., & Wang, Z. (2014, June). Combining von Neumann neighborhood topology with approximate-mapping local search for ABC-based service composition. In *Services Computing (SCC), 2014 IEEE International Conference on*, 187-194. <https://doi.org/10.1109/SCC.2014.33>
- [16] Dos Santos Coelho, L. & Alotto, P. (2011). Gaussian artificial bee colony algorithm approach applied to Loney's solenoid benchmark problem. *IEEE Transactions on Magnetics*, 47(5), 1326-1329. <https://doi.org/10.1109/TMAG.2010.2087317>
- [17] Zhu, G. & Kwong, S. (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied mathematics and computation*, 217(7), 3166-3173. <https://doi.org/10.1016/j.amc.2010.08.049>
- [18] Yi, Z. J., He, R. H., & Hou, K. (2012). Quantum artificial bee colony optimization algorithm based on Bloch coordinates of quantum bit. *Jisuanji Yingyong/Journal of Computer Applications*, 32(7), 1935-1938.
- [19] Banharnsakun, A., Achalakul, T., & Sirinaovakul, B. (2011). The best-so-far selection in artificial bee colony algorithm. *Applied Soft Computing*, 11(2), 2888-2901. <https://doi.org/10.1016/j.asoc.2010.11.025>
- [20] Ge, Y., Liang, J., & Wang, X. P. (2013). Improved Artificial Bee Colony Algorithm Based on Extremal Optimization Strategy. *Journal of Computer Applications*, 40(6), 247-251. <https://doi.org/10.3969/j.issn.1002-137X.2013.06.054>
- [21] TSai, P. W., Pan, J. S., Liao, B. Y., & Chu, S. C. (2009). Enhanced artificial bee colony optimization. *International Journal of Innovative Computing, Information and Control*, 5(12), 5081-5092.
- [22] Zang, M., Xuan, M. A., & Duan, Y. (2015). Improved artificial bee colony algorithm. *Journal of Xidian University*, 42(2), 65-70 and 139. <https://doi.org/10.3969/j.issn.1001-2400.2015.02.011>
- [23] Hu, K., Li, X. B., & Wang, Z. L. (2011). Performance of an improved artificial bee colony algorithm. *Jisuanji Yingyong/Journal of Computer Applications*, 31(4), 1107-1110. Zhang, D. L. (2014). Improvement and related application of artificial bee colony algorithm. (*Doctoral dissertation, Yanshan University*).
- [24] Han, X., Liu, Q., Wang, H., & Wang, L. (2018). Novel fruit fly optimization algorithm with trend search and co-evolution. *Knowledge-Based Systems*, 141, 1-17. <https://doi.org/10.1016/j.knsys.2017.11.001>
- [25] Zou, W., Zhu, Y., Chen, H., & Sui, X. (2010). A clustering approach using cooperative artificial bee colony algorithm. *Discrete dynamics in nature and society*. <http://dx.doi.org/10.1155/2010/459796s>
- [26] Yan, X., Zhu, Y., Zou, W., & Wang, L. (2012). A new approach for data clustering using hybrid artificial bee colony algorithm. *Neurocomputing*, 97, 241-250. <https://doi.org/10.1016/j.neucom.2012.04.025>

Contact information:

Xuming HAN, professor
School of Computer Science and Engineering,
Changchun University of Technology
Changchun 130012, China
hanxuming@163.com

Yidan WANG, master
School of Computer Science and Engineering,
Changchun University of Technology
Changchun 130012, China
1115347210@qq.com

Chen CAI, master
School of Control and Computer Engineering,
North China Electric Power University
Changchun 130012, China
caichen_work@163.com

Limin WANG, professor
(Corresponding author)
School of Management Science and Information,
Engineering, Jilin University Finance and Economics
Changchun 130117, China
wlm_new@163.com

Xiuping HOU, professor
School of Computer Science and Engineering,
Changchun University of Technology
Changchun 130012, China
461837067@qq.com

Linlin WANG, master
School of Computer Science and Engineering,
Changchun University of Technology
Changchun 130012, China
wl20161218@163.com