An enhanced artificial bee colony algorithm for numerical function optimization

Yi Yu, Yonggang Wu* and Xinglong Liu
School of Hydropower and Information Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China
*Corresponding author’s e-mail: vehust@126.com

Abstract. In response to the problems of slow convergence, low convergence accuracy and easy to fall in local optimum of the standard artificial bee colony (ABC) algorithm, this paper proposed the following two improvement measures: Firstly, on the basis of using the current optimal solution, the gradient modification strategy was used to reduce the leading role of the current optimal solution, so that the nectar resources could be fully searched to ensure the diversity of the population; Secondly, the crossover and mutation operation in genetic algorithm was used to replace the random search of scout bees, and the prior information was effectively utilized, which not only improved the search efficiency but also ensured the diversity of population to a certain extent. The numerical experiment results show that the proposed algorithm has good convergence accuracy and robustness, and also presents an excellent performance in multi-dimensional optimization.

1. Introduction
Artificial bee colony (ABC) algorithm is a global and random optimization algorithm based on bees foraging, which is proposed by Karaboga in 2005[1]. The algorithm has a few parameters, simple operation and strong robustness. Since the algorithm has been put forward, it has attracted great attention from many scholars around the world and they have conducted in-depth research on the theoretical research and application of algorithms. So far, the algorithm has been successfully applied to artificial neural network training, combinatorial optimization, power system optimization, data mining and image recognition and so on [2-6].

As a novel swarm intelligence algorithm, ABC has obviously been widely recognized and affirmed by domestic and foreign academic circles for its excellent performance for solving high-dimensional problems. However, as other optimization algorithms, traditional ABC algorithm also has the disadvantages of premature convergence and slow convergence and tending to fall into the local optimal solution [7-9]. In response to these, researchers have put forward a large number of effective methods for achieving a better balance of the ABC algorithm’s exploration and exploitation [10-18]. GABC algorithm [19] is an example of many successful improved ABC algorithms, which is mainly inspired by the current optimal solution guidance mechanism in particle swarm optimization (PSO) algorithm. It introduces the current optimal solution in the local search formula, which obviously improves the global search ability of the algorithm. The convergence of this algorithm has also been proved by literature [19]. However, research [20] points out that the improved measure may reduce the algorithm’s local optimization ability to some extent. This is due to it will quickly guide the employees to the vicinity of the current optimal solution, while other nectar sources of mining value have not been fully searched. In order to optimize the performance of the algorithm further, this paper...
improves it from two aspects. On one hand, the gradient information of the global optimal solution is introduced into the search formula for properly weakening the global optimal solution’s guiding effect, which results in a fully search in the neighbourhood under a lower convergence speed. On the other hand, inspired by the genetic algorithm, the mechanism of crossover and mutation is introduced in the scouts’ global search, in which crossover can make full use of prior information and mutation can ensure the diversity of individuals, which make the random searchers obtain extra nectar information thus improve their search efficiency. Prior to this, some mixed algorithms about ABC and GA have been proposed in references [20], [21] and [22]. In these references, their genetic operators are all implemented to manipulate all the populations in each generation so as to prefer superior species to the next generation. These may lead to unsatisfactory optimization results due to incomplete local search and premature optimization. This paper tried to use genetic operator to generate new nectar resource in the case that any nectar resource has not been updated exceeding the limit, which can effectively avoid the premature convergence, and make full use of the prior information and ensure the diversity of the population. These measures can better achieve the balance between the exploration and exploitation ability of the ABC algorithm in theory and a series of classical test functions are used to attest the improved algorithm’s rationality and superiority in this paper at last.

2. Overview of artificial bee colony algorithm

The standard ABC algorithm which simulates behaviors of gathering nectar by bees, searches out the nectar resource which contains the maximum nectar through sharing information among the different divisions of bees. The bees are divided into three categories based on division of labor: employees, onlookers and scouts. The employees share the information of nectar resource to onlookers by their “8” type of dance in dance-floor, onlookers wait in hive and search around the self-considered ideal nectar resource since employees sharing the information and scouts randomly search for a new valuable nectar resource near the hive. The process of ABC algorithm for solving problem actually is the process of searching in potential solution space. The position of each nectar resource represents a feasible solution to the problem and the amount of nectar indicates the corresponding solution’s fitness. In each generation, the number of employees is equal to the number of nectar resources, and there is a one-to-one relationship between the employees and nectar resources.

In order to ensure the diversity, employees are required to carry on a local search for better nectar resources around the corresponding resources in each generation based on the following formula,

\[
\overline{x}_j = x_j + \text{rand}(-1,1) \cdot (x_j - x_k)
\]

(1)

Where \(x_j\) is the value of generated nectar resource in \(j\)th dimension, \(x_j\) is the value of \(j\)th nectar resource in \(j\)th dimension, \(x_k\) is the value of \(k\)th nectar resource in \(j\)th dimension, in which \(k\) is a random number that is less than population quantity and not equal to \(i\). Comparing the generated nectar resource with the original one, the one with higher fitness is retained. The fitness is calculated as follow.

\[
\text{Fit}_i = \begin{cases} 
1, & \text{if } f_i > 0 \\
\frac{1}{1 + |f_i|}, & \text{if } f_i \leq 0
\end{cases}
\]

(2)

According to the information of nectar resources transmitted by employees, each onlooker will choose a nectar resource based on roulette strategy. The formula of possibility of being selected is as follow.

\[
p_i = \frac{\text{Fit}_i}{\sum_{i=1}^{n} \text{Fit}_i}
\]

(3)

Where \(\text{Fit}_i\) is the fitness of \(i\)th nectar resource and \(n\) is the number of nectar resources. The onlookers search for new nectar resources according to the Eq. (1) after selecting nectar resources by
roulette strategy. Meanwhile, employees update the nectar resources by fitness on basis of the greedy strategy. If any nectar resource has not been updated within a given limit of generation, the corresponding employee gives up the nectar resource, and searches for a new one according to the following formula,

\[ x_{id} = x_{id}^{\text{min}} + \text{rand}(0,1) \cdot (x_{id}^{\text{max}} - x_{id}^{\text{min}}) \]  

(4)

Where \( x_{id} \) is the value of \( i \)th nectar resource in \( d \)th dimension. \( x_{id}^{\text{min}} \) and \( x_{id}^{\text{max}} \) are the lower and upper bounds of \( i \)th nectar resource in \( d \)th dimension, respectively.

3. Enhanced artificial bee colony algorithm

The exploitation and exploration are two important performances of swarm intelligent algorithms [23]. The former is reflected in algorithms’ local search capability and the latter is reflected in the global search capability. The ABC algorithm’s local search is performed by employees and onlookers and global search is mainly reflected in the search process of scouters. In order to achieve a better balance of exploitation and exploration, the standard ABC algorithm is improved from the below two aspects.

3.1. Global optimal solution guidance and gradient adjustment strategies

Local search is based on Eq. (1) in standard artificial bee colony algorithm. It will be found by analyzing the Eq.(1) that the standard ABC algorithm’s local search is to select one dimension from one nectar resource as its local optimization variable, take the selected dimension of this nectar resource as the center and regard projection distance between the nectar resource and another one in this dimension as the search scope. In Eq. (1), the coefficient is a totally random number in \([-1,1] \), \( x_{id} \) is a random individual in the population and the possibility of selecting a good solution is nearly equal to that of selecting a bad solution. Therefore, the new candidate solution is not promising to be a solution better than the previous one and it also confirms the statement that the artificial bee colony algorithm is tired of local search. To solve this problem, the literature [19] presents GABC to introduce the global optima into the search formula for improving the exploitation which refers to particle swarm optimization and the specific is shown in Eq. (5).

\[ x_{id} = x_{id} + \varphi_{id} \cdot (x_{id}^{\text{Global}_{i}} - x_{id}) + \beta \left( x_{id}^{\text{Global}_{i}} - x_{id} \right) \]  

(5)

Where \( x_{id}^{\text{Global}_{i}} \) is the value of current optimal solution, \( \beta \) is a random number which subjects to a uniformly distributed in the range \([0, C] \), and the study found that the performance of the algorithm is the best when C is equal to 1.5. However, the research [20] has shown that the GABC algorithm inherits the feature of fast convergence speed of PSO, but leads to incomplete local search of other valuable individuals because of the effects of quick guide to the local optima. Aimed at the problem, this paper proposes a gradient adjustment strategy which reduces the speed of the bees falling into the local optimum through the reverse effect of the current optima’s gradient, which can make full search of the other individuals and ensure a good species diversity.

It is assumed that the N-ary function has first-order continuous partial derivative in the space region G, where \( g(x_1, x_2, \ldots, x_N) \in G \), the gradient of point \( g \) can be expressed as follow.

\[ \text{grad}^g = \frac{\partial F}{\partial x^i} x_1 + \frac{\partial F}{\partial x^2} x_2 + \cdots + \frac{\partial F}{\partial x^N} x_N \]  

(6)

Similarly, if the current spatial solution of the optimal solution is marked as \( x_{\text{global}}=(x_1, x_2, \ldots, x_D) \), the gradient of the current optimal solution is expressed below \( \text{grad}^{\text{Global}} \).

\[ \text{grad}^{\text{Global}} = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_D}) \]  

(7)

\[ \frac{\partial f}{\partial x_i} = \lim_{\Delta x \to 0} \frac{f(x_{i-1}, x_i + \Delta x, x_{i+1}, \cdots) - f(x_1, x_2, \cdots, x_D)}{\Delta x}, i = 1, 2 \cdots D \]  

(8)
After introducing the current optimal solution and gradient adjustment strategy into the local search of the ABC algorithm, the local search of the algorithm is transformed from formula (1) to formula (9).

$$\bar{x}_{id} = x_{id} + \varphi_d \cdot (x_{id} - x_d) + \beta \cdot (x_{id}^{Global} - x_{id}) + (-1)^k \cdot \gamma(t) \cdot \left| x_{id}^{Global} - x_d \right| \cdot \left| \text{grad}^{Global}_{e,d} \right|$$ (9)

Where $k$ is the direction control parameter, if $x_{id} > x_{id}^{Global}$, $k$ is even; otherwise, $k$ is odd. $\gamma(t)$ is the adaptive parameter, which changes with the number of iterations; $\left| x_{id}^{Global} - x_d \right|$ is the control item of step size; $\left| \text{grad}^{Global}_{e,d} \right|$ is the scalar of gradient unit vector in d dimension, and it is the control parameter of step size change. The gradient unit vector is expressed as follows.

$$\text{grad}^{Global}_e = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_d} \right) / \sqrt{\left( \frac{\partial f}{\partial x_1} \right)^2 + \left( \frac{\partial f}{\partial x_2} \right)^2 + \ldots + \left( \frac{\partial f}{\partial x_d} \right)^2}$$ (10)

3.2. Crossover and mutation strategy

In ABC algorithm, the nectar source can be considered to be exhausted if the number of nectar source that has not been updated reaches the limit. At this time, the employee corresponding to the nectar source will change its role as a scout bee, abandon the mined nectar source, and then randomly search in the search space to generate a new one. The mechanism makes the algorithm possible jump out of the trap when it falls into the local optimal state, which is also a highlight in the design of ABC algorithm. In the standard ABC algorithm, global random search is conducted according to formula (4), but it is undoubtedly inefficient to search for nectar sources in the complex solution space, especially in the later stage of optimization.

As is known to all, the GA has the characteristics of self-organization, self-adaptation and self-learning [24], and it realizes the continuous evolution of population through selection, crossover and mutation. Selection is to reserve some preferred individuals directly to the next generation, or to provide an operational basis for crossover operators and mutation operators, so as to protect the good genes. Crossover generates two new individuals through the recombination of the selected parent generation, which is used for the crossover combination to generate new individuals to search the solution space effectively and reduce the probability of destroying the effective mode. Mutation randomly selects an individual from the population to exchange its gene position to increase the diversity of the population. It has good global search ability, and can quickly search out the approximate optimal solution in the solution space [24-25].

Considering the inefficiency of scout’s random search in the standard ABC algorithm and the good global search capability of GA, the idea of crossover and mutation is introduced, instead of the original random searching. On one hand, crossover can make full use of the prior information of the population and improve the search efficiency of scout bee. On the other hand, mutation retains the original character of random search which can still guarantee the diversity of population with a certain probability. So the formula is changed from equation (4) to equation (11).

$$x_{id} = \begin{cases} \lambda \cdot x_{id} + (1-\lambda) \cdot x_{id} & \text{if } 0 \leq p \leq p_c \\ L_d + \text{rand}(0,1) \cdot (U_d - L_d) & \text{if } p_c < p \leq 1 \end{cases}$$ (11)

In the formula, $\lambda$ is a uniformly distributed random number within the range of $[0, 1]$, $k$ and $l$ are two nectar sources randomly selected by roulette based on sorting probability; $p_c$ is the probability of cross, $p_m$ is the probability of mutation, $p_m=1-p_c$.

4. Enhanced artificial bee colony algorithm flow

1) Total number of bees is initialized $N$, in which one half are employees and the others are onlookers. The number of nectar resources is initialized $N/2$. Maximum residence times is initialized limit. Iteration is marked iter=0, Maximum iteration is initialized $Max_{Cycle}$. 


2) The bees randomly select $N$ nectar resources in solution space. Calculate fitness values with Eq. (3) and record the initial sign the res_times(1)=0.

3) Each employer randomly search around its located nectar resource according to the Eq.(9). Calculate each fitness value, if it is better than the original source, update the position of the employed bees and res_times(i)=0. Otherwise, res_times(i+1)= res_times(i)+1.

4) Calculate selection probability according to the Eq.(2), each onlooker selects a nectar reource to follow according to the Eq.(3), update the nectar resource and res_times(i).

5) Determine whether res_times(i) is greater than limit. If it is, continues. Otherwise, switch to 7.

6) The i-th employee gives up its nectar resource and searches for a new one according to Eq.(11).

7) Record the current optimal solution and iterations iter=iter+1. Determine whether iterations is greater than Max_Cycle. If it is, termination. Otherwise, skip to 4.

5. Experiments

5.1. Benchmark functions

In order to test the performance of EABC algorithm on numerical function optimization, a series of benchmark functions shown in table 1 are used here.

| Functions | Search space | The minimum solution |
|-----------|--------------|----------------------|
| $f_1(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$ | $[-600,600]^D$ | $f_1(\vec{0}) = 0$ |
| $f_2(x) = \sum_{i=1}^{D} [x_i^2 - 10 \cos(2\pi x_i) + 10]$ | $[-5.12,5.12]^D$ | $f_2(\vec{0}) = 0$ |
| $f_3(x) = \sum_{i=1}^{D} [100(x_i - x_i^*)^2 + (x_i - 1)^2]$ | $[-50,50]^D$ | $f_3(\vec{0}) = 0$ |
| $f_4(x) = -20 \exp(-0.2 \sqrt{\sum_{i=1}^{D} x_i^2}) - \exp(\sum_{i=1}^{D} \cos(2\pi x_i)) + 20 + e$ | $[-32,32]^D$ | $f_4(\vec{0}) = 0$ |
| $f_5 = D \times 418.9829 + \sum_{i=1}^{D} -x_i \sin \left( \sqrt{|x_i|} \right)$ | $[-500,500]^D$ | $f_5(420.9867) = 0$ |

5.2. Parameter setting

The experiment in this paper includes two parts: one is to evaluate the improvement effect of EABC algorithm by comparing with the results of ABC and GABC algorithm in the reference [19]; the other one is to evaluate the optimization performance of EABC algorithm by comparing with the results of the representative optimization algorithms, such as GA, PSO and PS-EA in the reference [26]. In order to ensure the objectivity, the parameters in this paper are set as consistent with the relevant literatures. The population size is 80 and the maximum generations is 5000 in the first part, the population size is 125 in the other one. Each experiment was repeated 30 times independently.

5.3. Experimental results

The comparison with the results in the reference [19] is shown in table 2. As can be seen that EABC is obviously superior to the traditional ABC algorithm in the optimization results of different dimensions of each test function, which proved that the optimization performance of EABC has been greatly improved compared with the traditional ABC algorithm. Compared with GABC test results, EABC is also significantly better than other algorithms in terms of mean and standard deviation, except for that in the optimization of 30-dimensional test Griewank function. Therefore, compared with ABC and GABC, the optimization performance of EABC algorithm has been greatly improved, indicating the effectiveness of the improvement measures in this paper. For the sake of intuition, figures 1 and 2 respectively show the convergence of the optimal solutions of ABC, GABC and EABC algorithm in
30-dim and 60-dim Griewank, Rastrigin, Ackley and Schwefel functions. It is easy to find that the optimization ability of the EABC algorithm is obviously better than that of ABC and GABC.

Table 2. Comparison with ABC and GABC algorithms

| Function | Alg. | ABC | GABC | EABC |
|----------|------|-----|------|------|
| Dim      | Mean | SD  | Mean | SD   | Mean | SD   |
| $f_1$    | 30   | 1.27306E-15 | 1.46400E-15 | 2.96059E-17 | 4.99300E-17 | 1.66531E-16 | 1.56354E-16 |
| 60       | 2.51040E-13 | 7.51400E-13 | 7.54952E-16 | 4.12700E-16 | 3.70000E-16 | 4.12432E-16 |
| $f_2$    | 30   | 1.34529E-13 | 7.96600E-14 | 1.32635E-14 | 2.44500E-14 | 3.78953E-15 | 1.41791E-14 |
| 60       | 2.06479E-08 | 1.12100E-07 | 3.52429E-13 | 1.24300E-13 | 1.44004E-13 | 1.63259E-13 |
| $f_3$    | 2    | 9.93136E-03 | 8.14300E-03 | 1.68497E-04 | 1.45400E-04 | 3.66933E-08 | 1.97600E-07 |
| 3        | 6.44947E-02 | 4.85200E-02 | 2.65914E-03 | 2.22000E-03 | 1.09957E-04 | 3.66540E-04 |
| $f_4$    | 30   | 4.69550E-14 | 5.95400E-15 | 3.21521E-14 | 3.25200E-15 | 1.45785E-14 | 2.35015E-15 |
| 60       | 1.66089E-13 | 2.21700E-14 | 1.00099E-13 | 6.08900E-15 | 9.97120E-14 | 1.81829E-14 |
| $f_5$    | 30   | -   | -   | -   | -   | 3.81830E-04 | 1.62630E-19 |
| 60       | -   | -   | -   | -   | -   | 7.65998E-04 | 9.93060E-06 |

Figure 1. Functions with Dim=30
In order to further verify the superiority of the EABC algorithm, the algorithm is compared with representative algorithms such as GA, PSO and PS-EA, and the results are shown in table 2. Experimental data of GA, PSO and PS-EA algorithms in table 3 are extracted from reference [26]. As can be seen from table 3, compared with other algorithms, EABC algorithm has high convergence accuracy and strong robustness in different dimensions in each test function, and the results of EABC algorithm are obviously superior to other algorithms, reflecting the superiority of EABC algorithm.

### Table 3. Comparison with GA, PSO and PS-EA algorithms

| Function | Alg. | GA | PSO | PS-EA | EABC |
|----------|------|----|-----|-------|------|
| Dim | Mean | SD | Mean | SD | Mean | SD | Mean | SD |
| 10 | 5.02280E-02 | 2.95230E-02 | 7.93930E-02 | 3.34510E-02 | 2.22366E-01 | 7.81000E-02 | 1.36607E-03 | 2.47313E-03 |
| 20 | 1.01390E+00 | 2.69660E-02 | 3.05650E-02 | 2.54190E-02 | 5.90360E-01 | 2.03000E-01 | 9.45744E-04 | 2.92831E-03 |
| 30 | 1.23420E+00 | 1.10450E-01 | 1.11510E-02 | 1.42090E-02 | 1.39400E-01 | 1.39400E-01 | 3.93297E-04 | 1.78842E-03 |
| 10 | 1.39280E+00 | 7.63190E-01 | 2.65590E+00 | 1.38960E+00 | 4.34040E-01 | 2.55100E-01 | 0.00000E+00 | 0.00000E+00 |
| 20 | 6.03090E+00 | 1.45370E+00 | 1.20590E+01 | 3.32160E+00 | 1.81350E+00 | 2.55100E-01 | 2.55795E-14 | 8.56904E-14 |
| 30 | 1.04388E+01 | 2.63860E+00 | 7.73820E+01 | 3.24760E+01 | 9.49010E+01 | 3.05270E+00 | 2.73441E+01 | 9.79914E-02 |
| 10 | 4.63184E+01 | 3.38217E+01 | 4.37130E+00 | 2.38110E+00 | 2.38110E+00 | 2.53030E+00 | 2.97964E+01 | 2.63031E+00 |
| 20 | 1.03930E+02 | 6.03090E+00 | 7.73820E+01 | 9.49010E+01 | 7.24520E+01 | 2.55100E-01 | 2.55795E-14 | 8.56904E-14 |
| 30 | 1.66283E+02 | 2.63860E+00 | 7.73820E+01 | 9.49010E+01 | 7.24520E+01 | 2.55100E-01 | 2.55795E-14 | 8.56904E-14 |
| 10 | 1.39280E+02 | 7.63190E-01 | 2.65590E+00 | 1.38960E+00 | 4.34040E-01 | 2.55100E-01 | 0.00000E+00 | 0.00000E+00 |
| 20 | 6.03090E+02 | 1.45370E+00 | 1.20590E+01 | 3.32160E+00 | 1.81350E+00 | 2.55100E-01 | 2.55795E-14 | 8.56904E-14 |
| 30 | 1.04388E+03 | 2.63860E+00 | 7.73820E+01 | 3.24760E+01 | 9.49010E+01 | 3.05270E+00 | 2.73441E+01 | 9.79914E-02 |
| 10 | 5.92670E+01 | 2.24820E-01 | 9.84990E+13 | 6.92020E+13 | 1.92090E+01 | 1.95100E-01 | 5.39032E-15 | 6.70333E-15 |
| 20 | 2.95050E+02 | 2.55100E+00 | 1.25090E+01 | 6.33560E+00 | 1.81350E+00 | 2.55100E-01 | 2.55795E-14 | 8.56904E-14 |
| 30 | 1.09980E+03 | 2.49560E-01 | 1.49170E+06 | 9.87620E+06 | 5.99010E+06 | 2.55100E-01 | 2.55795E-14 | 8.56904E-14 |
| 10 | 1.39280E+03 | 4.95340E+00 | 9.90770E+02 | 5.81140E+02 | 3.27200E+00 | 1.61850E+00 | 3.21173E-03 | 1.48244E-02 |

The values obtained by GA, PSO, PS-EA and EABC after 500, 750 and 1,000 cycles for dimensions 10, 20 and 30.

### 6. Conclusion

An enhanced artificial bee colony algorithm is proposed in this paper, which is optimized by using the gradient correction strategy and the crossover and mutation of genetic algorithm. The experimental
results show that the improved algorithm has high convergence accuracy, strong robustness, and remarkable effect. And compared with the current advanced algorithms, the EABC algorithm has some superiority, and has certain advantages for the high dimensional optimization problems.

Acknowledgment
This work is partially supported by the National Natural Science Foundation of China (No.51379081) and Project of Science and Technology of SGCC (DKJS-13-00220).

References
[1] Karaboga D. (2005) An idea based on honey bee swarm for numerical optimization. Erciyes University Technical Report.
[2] Lučić P, Teodorović D. (2003) Transportation Modeling:An Artificial Life Approach. In:IEEE International Conference with Artificial Intelligence.
[3] Teodorović D. (2009) Bee colony optimization. Innovations in Swarm Intelligence.
[4] H D, S S, S Y. (2005)Cooperative Bees Swarm for Solving the Maximum Weighted Satisfiability Problem[Z]. Vilanova i la Geltrú.
[5] Wedde H F, Farooq M, Yue Z. (2004) BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior. In:4th International Workshop Ant Colony Optimization and Swarm Intelligence.
[6] Shaik S, Xu F Y, Lai C S, Sinha N, Lai L L. (2012) Short-term hydro -thermal scheduling with Artificial Bee Colony. In:2012 International Conference on Machine Learning and Cybernetics (ICMLC) IEEE.
[7] Yi Y, Yonggang W, Binqi H, Xinglong L, Quan Z. (2018) An enhanced artificial bee colony algorithm (EABC) for solving dispatching of hydro-thermal system (DHTS) problem. Plos One, 13: e189282.
[8] Karaboga D, Akay B. (2009) A comparative study of Artificial Bee Colony algorithm. Applied Mathematics & Computation, 214: 108-132.
[9] Yuan X, Ji B, Chen Z, Chen Z. (2014) A novel approach for economic dispatch of hydrothermal system via gravitational search algorithm. Applied Mathematics & Computation, 247: 535-546.
[10] Cui L, Li G, Zhu Z, Lin Q, Wen Z, Lu N, Wong K, Chen J. (2017) A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization. Information Sciences: S1915193703.
[11] Akbari R, Mohammadi A, Ziarati K. (2010) A novel bee swarm optimization algorithm for numerical function optimization. Communications in Nonlinear Science & Numerical Simulation, 15: 3142-3155.
[12] Bouaziz A, Draa A, Chikhi S. (2013) A Quantum-inspired Artificial Bee Colony algorithm for numerical optimisation. In:2013 11th International Symposium on Programming and Systems.
[13] Peng G, Cheng W, Jian L. (2011) Global artificial bee colony search algorithm for numerical function optimization. In:2011 Seventh International Conference on Natural Computation IEEE.
[14] Bi X J, Wang Y J. (2011) Niche artificial bee colony algorithm for multi-peak function optimization. Systems Engineering and Electronics, 33: 2564-2568.
[15] Gao W, Liu S, Fei J, Zhang J. (2011) Hybrid artificial bee colony algorithm. Systems Engineering & Electronics, 33: 1167-1170.
[16] Bao L, Zeng J. (2010) Self-adapting search space chaos-artificial bee colony algorithm. Application Research, 27: 1330-1334.
[17] Luo J, Li Y. (2010) Artificial bee colony algorithm with chaotic-search strategy. Control & Decision, 25: 1913-1916.
[18] Bi X, Wang Y. (2011) An improved artificial bee colony algorithm. In:International Conference
on Computer Research & Development.

[19] Zhu G, Kwong S. (2010) Gbest-guided artificial bee colony algorithm for numerical function optimization. Applied Mathematics & Computation, 217: 3166-3173.

[20] M J, D Y. (2014) Artificial bee colony algorithm and its application. Science Press, Beijing.

[21] Yanhua Y. Genetic bee colony algorithm and its application [D]. Nanjing: Nanjing University of Science and Technology, 2012.

[22] Gunasekaran L, Subramaniam S. (2016) FAGA: hybridization of fractional order ABC and GA for optimization. Int Arab J Inf Techn, 13: 1045-1053.

[23] Shi Y. (2014) Diversity and Mechanisms in Swarm Intelligence. International Journal of Swarm Intelligence Research, 5: 1-12.

[24] Richardson J T. (1989) Some guidelines for genetic algorithms with penalty functions. In: Proceedings of the third international conference on Genetic algorithms.

[25] Orero S O, Irving M R. (1998) A genetic algorithm modelling framework and solution technique for short term optimal hydrothermal scheduling. IEEE Transactions on Power Systems, 13: 501-518.

[26] Karaboga D, Basturk B. (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Global Optimization, 39: 459-471.