An Artificial Bee Colony Algorithm Based on Hybrid Strategy

Yang Zhang

Department of Computing Center, Liaoning University, Shenyang 110036, Liaoning, China

Abstract

Artificial Bee Colony (ABC) Algorithm is an optimization method which simulates the bee behaviors and it is a specific application of swarm intelligence. However, the basic ABC algorithm has such defects as premature convergence, easiness to get trapped in local optimum and slow convergence in the post evolutionary stage. Therefore, in order to effectively reinforce the optimization performance and overcome the shortcomings of ABC algorithm, this paper introduces differential evolution operators in ABC algorithm and proposes corresponding improved ABC algorithms based on DE operators to different variants of DE algorithm by referring to the inspiration of differential evolution (DE) and the mutation idea of DE algorithm. The experiment simulation shows that the improved hybrid algorithm has improved or enhanced its accuracy in the optimal solution and the convergence to certain extent.

Key words: Artificial Bee Colony, Differential Evolution, Hybrid Strategy.

1. INTRODUCTION

Swarm intelligence is the feature that the body with no or simple intelligence demonstrates intelligent behaviors through the gathering in any form. Swarm intelligence algorithm is a probability search algorithm and its fundamental idea is to construct the stochastic optimization algorithm by simulating the collective behaviors of the lives in the nature. As a new swarm intelligence optimization algorithm, ABC algorithm has attracted the attention of numerous scholars since it has few control parameters, simple computation, easy realization, strong robustness and extensive application range (Amir and Babak et al, 2013). Its main characteristic is that no special information is needed and that all it needs is to compare the advantages and disadvantages of the problems and highlight the global optimum through global optimization behavior of each artificial bee. At present, ABC algorithm has been successfully applied to the artificial neural network training, image processing and filter design. Its excellent optimization performance and application effect have made it one of the most promising optimization algorithms. It is still in the preliminary research stage and it deserves further research (Sadegh and Hamid et al, 2015).

As a natural algorithm, ABC algorithm simulates the highly-efficient mechanism of the bee colony in the natural world in searching for food sources and it has the mechanism of role assignment and transition. In the Nobel Academic Exchange Conference in 1973, Frisch has raised that the bees share the information of nectar sources through round dance or wagging dance, including such information elements as direction, distance and quality of nectar sources (Alkın and Erdal, 2015). Seeley is the first to come up with a social behavioral model of bee colony. In this model, different bees only perform the simple tasks, however, the entire colony can work together in the complicated tasks such as honeycomb construction, reproduction and foraging. After Seeley, Abbass develops a bee mating optimization (BMO) model and Bozorg and Hadda have applied this model to the test platform of three mathematical problems. Karaboga D, a Turkish scholar, has brought forth the systematic artificial bee colony algorithm for the first time in 2005 and verified its effectiveness with benchmark function. So far, the research of ABC algorithm mainly focuses on two aspects: one is to apply this algorithm in different research fields and the other is to improve the algorithm (Wei Feng and Lingling et al, 2015; Abou and Aissa et al, 2015). The research and application of ABC algorithm is still in the preliminary stage and there are still many problems to be further improved and solved. In order to effectively improve the performance of ABC algorithm, this paper studies its hybrid strategy.

Firstly, this paper explores the principle, model and information-sharing mechanism of ABC in an in-depth manner. Then, it describes the idea of differential evolution algorithm and improves and compares the artificial bee colony algorithm by introducing the mutation strategy of differential evolution. Finally, the simulation experiment proves that the improved hybrid algorithm has stronger optimization ability and it improves in the convergence speed and accuracy.

2. DIFFERENTIAL EVOLUTION

The main operation idea of DE is to generate interim individuals based on the individual diversity of the population and then realize population evolution after random reorganization. In theory, DE uses the difference of the variance of two randomly-selected vectors (individuals) as the evolutionary dynamics; therefore, DE can
effectively control the hopping length by taking the strategy similar to proximity search. This algorithm starts from a certain randomly-generated initial population, realizes the population evolution through the cyclical iterative computation such as selection, crossover and mutation and makes the fittest particles survive via fitness selection. The researches on such aspects as the parameter setting, the evolution mechanism, the applied research and the integration mechanism with other algorithms of DE have become the key research fields on this algorithm.

The flowchart of basic DE is shown as below Figure 1.

**Figure 1.** Flowchart of DE.

The key evolution operations of DE include: mutation, crossover and selection.

1. **Mutation Operation**

   Generate the mutation vector \( v_i \) for any objective base vector \( x_i \) in the parent population according to the formula below.

   \[
   v_i = x_i + \omega (x_i - x_{best}) + \omega (x_i - x_i)
   \]

   In this formula, \( \omega \) is scaling factor to control the degree of variation of the differential vector, \( (x_i - x_j) \) is difference vector, \( i = 1, 2, ..., n, \not i \neq i \neq i \neq i \). \( x_{best} \) is the optimal solution found so far (Rangababu and Kiran et al, 2014).

2. **Crossover Operation**

   The purpose of crossover operation in DE algorithm is to increase the diversity of the individuals through random reorganization of mutation vector \( v_i \) and every component of the objective base vector \( x_i \). New crossover vectors \( u_i \) generate according to the following formula.

   \[
   u_i = \begin{cases} 
   v_i, & \text{if } rand[0,1] \leq CR \\
   x_i, & \text{otherwise}
   \end{cases}
   \]

   In this formula, \( CR \) is the crossover factor and \( rand[0,1] \) is variable which is randomly generated within 0 and 1 (Adam, 2014).

3. **Selection Operation**

   \( u_i \) will only be accepted by the population when and only when the fitness of the new vector \( v_i \) is better than that of the objective base vector \( x_i \), otherwise, \( x_i \) will still be preserved in the population of the next generation and mutation and crossover operations will still be implemented on it in the next iterative computation. Selection operation can be described as follows.

   \[
   x_i = \begin{cases} 
   v_i, & \text{if } fitness(v_i) < fitness(x_i) \\
   x_i, & \text{otherwise}
   \end{cases}
   \]
In this formula, fitness() is the fitness function. DE algorithm uses greedy selection method, namely guarantee that the population in the next generation will contain the vectors with better fitness(Francesco and Samuele et al, 2014).

3. PROCEDURES OF ARTIFICIAL BEE COLONY ALGORITHM

The minimum search model of bee colony in producing swarm intelligence includes three basic components: food source, employed foragers and unemployed foragers. The two most fundamental behavior models are to recruit and abandon a certain food source. The employed forager is also called as leader and there are two kinds of unemployed foragers: scouter and follower. Below are the procedures of ABC algorithm.

(1) Initialization of nectar sources
In the initialization, randomly produce \( SN \) feasible solutions, namely the quantity of employed foragers and calculate the fitness function. The formula to randomly produce feasible solutions is as follows.

\[
x_{ij} = x_{mn,j} + \text{rand}(0,1)(x_{max,j} - x_{min,j})
\]

(4)

In this formula, \( x_i \) \((i=1, 2, L, SN)\) is the D-dimensional vector, \( D \) is the number of optimization parameters and \( j \in \{1, 2, L, D\} \).

(2) Search formula for new nectar sources
The bee records its current optimal value and searches in the neighborhood of the current nectar source. The formula to search new nectar sources in the neighborhood of the ABC algorithm is as follows.

\[
x_{ij} = x_{ij} + r_i(x_{ij} - x_{ij})
\]

(5)

In this formula, \( j \in \{1, 2, L, D\}, k \in \{1, 2, L, SN\} \), \( k \) is produced randomly and \( k \neq i \), \( r_i \) is a random number among \([-1,1]\), which controls the generation range of the neighborhood and the neighborhood range reduces gradually as the search is close to the optimal solution(Can B and Surendra, 2013).

(3) Probability for the scouter to select employed foragers.

\[
P = \frac{\text{fit}(x_i)}{\sum_{n=1}^{\text{SN}} \text{fit}(x_n)}
\]

(6)

In this formula, \( \text{fit}(x_i)\) is the abundance of the corresponding nectar source to the \( i \)th solution. The more abundant the nectar source is, the higher probability to be selected by the scouter(Valinataj-Bahnemiri and Ramiar, 2015).

(4) Production of scouter
In order to prevent the algorithm from getting trapped into local optimum, the nectar source will be abandoned when no improvement is made in the limit iterations and it will be recorded in the tabu table. In the meanwhile, the corresponding employed forager to this nectar source will become a scouter and a new position will be produced to replace the previous nectar source according to Formula (4).

(5) The follower selects the food source (solution) according to the probability value, conducts neighborhood search to produce new solution according to the formula above and calculate its fitness value.

(6) If the fitness value of the new solution is better than the original solution, replace the former with the latter and take the new solution as the current best solution, otherwise, keep it unchanged.

(7) Judge whether there is any solution to be abandoned. If any, the scouter will produce a new solution in its place according to the formula of \( X^j = X^j + \text{rand}(0,1)(X^j - X^j_{\text{min}}) \).

(8) Record the best solution ever.

(9) Judge whether the circle termination condition is satisfied. If so, output the optimal result; otherwise, return to (2).

The flowchart of ABC algorithm is indicated as Figure 2.
4. COMBINATION OF DIFFERENTIAL EVOLUTION AND ARTIFICIAL BEE COLONY ALGORITHM

4.1. Basic Idea

Inspired by the idea of differential evolution (DE), this paper introduces DE operators into ABC algorithm. The scouter selects the nectar source with higher yield at a certain probability. In the subsequent search, produce new candidate solutions by using the mutation and crossover strategies in DE, compare the new candidate solutions and the target nectar source with greedy selection strategy where the better one wins, improve the optimization ability of the algorithm and it is better for the bees to find better nectar sources. Its basic idea is as follows.

(1) After the colony initialization, calculate the corresponding fitness value to every nectar source $X$, for a global minimization problem $\text{min}(X)$, bring the position vector $X$ of each nectar source into the fitness function $F$ and calculate its fitness value, which stands for the yield of the nectar source.

(2) Introduce the mutation strategy of DE algorithm into ABC algorithm. Use the following mode in the search of the leader and Formula (5) in ABC algorithm will be replaced by Formula (7).

$$x'_{ij} = x_{best,j} + \varphi \cdot (x_{ij} - x_{ij}) \quad (7)$$

In this formula, $x_{best,j}$ is the best individual in the population of $j$ generation, $\varphi$ is no longer a constant, but an adaptive and dynamic variable, as indicated by Formula (8).

$$\varphi = 1 - \frac{G}{\text{max}T} \quad (8)$$

Here, $G$ is the current iterations of the algorithm $\text{max}T$ is the maximum iterations. In the preliminary stage of the algorithm, $G$ is small while $\varphi$ is big. The mutation intensity of the algorithm is big, which is helpful in the evolution towards the optimal value in a more effective and rapid manner in the post evolution. With the continuous evolution, $G$ is growing bigger while $\varphi$ smaller. As the individual evolves towards the optimal value, the function can converged to the optimal value rapidly and stably.

(3) Every bee in the colony needs to remember the position and the corresponding fitness value of its own nectar source. In the update search of the $i$th nectar source, if the corresponding fitness of the new candidate nectar source is better than the current value, then update the corresponding fitness value $\text{fitness}$ and position vector. In order to increase the diversity of the solutions, Formula (6) is still used in the search for the solutions of the follows in order to maintain the randomness of the search and reduce the chance for the solution to get trapped in the local optimum.
4.2. Optimization Test of Benchmark Function

In order to verify the optimization performance and convergence speed of DEABC algorithm proposed in this paper, four benchmark functions are selected to conduct optimization test. These four standard benchmark functions have different characteristics and they can test the optimization ability of the algorithm. Their expression formulas and characteristics are shown in Table 1. Among them, Function 1 is a uni-modal function and it is used to test the convergence speed of the optimization algorithm while Formulas 2, 3 and 4 are complicated non-linear functions and they have many local optimal solutions, which are deemed as the most difficult optimization problems. \( f_2 - f_4 \) are these functions. The number of their local optimal solutions will rapidly increase with the increase of dimensions and they are always used to test the group diversity, the global search and the convergence ability to jump out of local extremum and to avoid prematurity. Benchmark functions are shown in Table 1.

In order to compare with other search algorithms and increase the optimization complexity, optimization test will be conducted with the standard ABC and the DEABC in this paper, the leader number is 50, the follower number is 50, and the maximum number of iterations is 1000. Perform 30 experiments on every standard test function (30-dimensional) and take the mean value as the final result. The test result is shown in Table 2.

<table>
<thead>
<tr>
<th>Test function</th>
<th>n</th>
<th>Scope ([-100,100]^n)</th>
<th>Minimum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1(x) = \sum_{i=1}^{n} x_i^2 )</td>
<td>30</td>
<td>([-100,100]^n)</td>
<td>0</td>
</tr>
<tr>
<td>( f_2(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} x_i \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1 )</td>
<td>30</td>
<td>([-600,600]^n)</td>
<td>0</td>
</tr>
<tr>
<td>( f_3(x) = \frac{1}{n} \left( 10 \sin^2 \left( \pi x_1 \right) + \sum_{i=1}^{n-1} \left( y_i - 1 \right)^2 \left[ 1 + \sin^2 \left( \pi y_{i+1} \right) \right] \right) + (y_n - 1)^2 + \sum_{i=1}^{n} u \left( x_i, 10, 100, 4 \right) )</td>
<td>30</td>
<td>([-50,50]^n)</td>
<td>0</td>
</tr>
<tr>
<td>( u(x, a, k, m) = \begin{cases} \displaystyle k (x_i - a)^m, &amp; x_i &gt; a, \ 0, &amp; -a \leq x_i \leq a, \ k (-x_i - a)^m, &amp; x_i &lt; -a. \end{cases} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_4(x) = 0.1 \left[ \sin^2 \left( 3 \pi x_1 \right) + \sum_{i=1}^{n-1} \left( x_i - 1 \right)^2 \left[ 1 + \sin^2 \left( 3 \pi x_{i+1} \right) \right] \right) + (x_n - 1) \left[ 1 + \sin^2 \left( 2 \pi x_n \right) \right] + \sum_{i=1}^{n} u \left( x_i, 5, 100, 4 \right) )</td>
<td>30</td>
<td>([-50,50]^n)</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test function</th>
<th>ABC</th>
<th>DEABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Standard deviation</td>
<td>Mean</td>
</tr>
<tr>
<td>( f_1 )</td>
<td>1.72e^{-19}</td>
<td>2.65e^{-19}</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>8.65e^{-12}</td>
<td>4.31e^{-11}</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>2.17e^{-11}</td>
<td>2.56e^{-11}</td>
</tr>
<tr>
<td>( f_4 )</td>
<td>2.54e^{-12}</td>
<td>2.62e^{-12}</td>
</tr>
</tbody>
</table>
Figure 3. Two-dimensional space diagram of four test functions $f_1 - f_4$

It can be seen from Table 2 that all these algorithms can achieve the optimization accuracy of most functions, suggesting that they all have strong global search ability. For the complicated non-linear global optimization functions, namely Functions 2, 3 and 4, there exist local minimum or it is easy to get trapped into local minimum. DEABC has better optimization effect and higher optimization accuracy, indicating that DEABC has not only excellent global search ability, but also strong local optimization ability and the anti-prematurity is also improved. As for the test problem $f_2$, both the mean and variance are big because there is certain probability for the DE operators to get trapped in local optimum. However, generally speaking, the performance of DEABC is better and it searches the optimal solution to most problems, proving that DEABC is effective in multi-modal problems.

5. CONCLUSIONS

Artificial bee colony algorithm is a new bionic optimization method by simulating the behaviors of bee and it is also an application of swarm intelligence. This paper has integrated differential evolution algorithm with artificial bee colony algorithm to make the hybrid algorithm, namely DEABC, have a good balance between the global and local search capacities so that it improves the comprehensive optimization ability of the hybrid algorithm. The simulation experiment shows that DEABC algorithm has better performance in optimizing uni- and multi-modal problems and it can well balance the global and local optimization.

REFERENCES


