International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.6, No.6 (2013), pp.259-274
http://dx.doi.org/10.14257/ijsip.2013.6.6.24

An Improved Artificial Bee Colony Algorithm and Its Application

Xiangyu Kong[1, 2,] *, Sanyang Liu[1] and Zhen Wang[2]

[1]*Department of Applied Mathematics, Xidian University, Xi'an 710071, China*
[2]*Institute of Information and System Computation Science, Beifang University of Nationalities, Yinchuan 750021, China*

1kxywz08@163.com, 2 liusanyang@126.com, 3 jen715@163.com

Abstract

To further improve the performance of artificial bee colony algorithm (ABC), an improved ABC (IABC) algorithm is proposed for global optimization via employing orthogonal initialization method. Furthermore, to balance the exploration and exploitation abilities, a new search mechanism is also designed. The performance of this algorithm is verified by using 27 benchmark functions. And the comparison analyses are given between the proposed algorithm and other nature-inspired algorithms. Numerical results demonstrate that the proposed algorithm outperforms the original ABC algorithm and other algorithms for global optimization problems.

Keywords: *Artificial bee colony, Orthogonal initialization, Search mechanism, Differential evolution*

1. Introduction

Global optimization problems arise in almost every field of science, engineering and business. By now, learning from life system, many optimization methods have been developed to solve global optimization problems, such as genetic algorithms (GAs) [1, 2], ant colony optimization (ACO) [3], differential evolution (DE) [4] and particle swarm optimization (PSO) [5]. These kinds of algorithms can be named as artificial-life computation. Recently, Karaboga [6] proposed a new kind of optimization technique called artificial bee colony (ABC) algorithm for global numerical function optimization, which simulates the foraging behavior of honey bee swarm. A set of comparison experimental results show that ABC algorithm is competitive to some conventional bio-inspired algorithms with an advantage of employing fewer control parameters [7]. Due to its simplicity, ABC algorithm has been applied to solve many kind of real-world problems, for instance, leaf-constrained minimum spanning tree problem [8], flow shop scheduling problem [9], inverse analysis problem [10], radial distribution system network reconfiguration problem [11], clustering problem [12], TSP problems [13], a large-scale capacitated facility location problem [14], and so on.

According to the applications showed above, ABC algorithm seems to be a well-performed algorithm. However, similar to other population-based algorithms, there still are insufficiencies in ABC algorithm, such as slower convergence speed for some unimodal problems and easily get trapped in local optima for some complex multimodal problems [7]. It is well known that for the population-based algorithms the exploration and the exploitation

* Corresponding author.

ISSN: 2005-4254 IJSIP

abilities are both necessary facts. The exploration ability refers to the ability to investigate the various unknown regions to discover the global optimum in solution space, while the exploitation ability refers to the ability to apply the knowledge of the previous good solutions to find better solutions. The exploration ability and the exploitation ability contradict to each other, so that the two abilities should be well balanced to achieve good performance on optimization problems. So far as we know that the search equation of ABC algorithm is good at exploration but poor in exploitation.

Therefore, accelerating convergence speed and avoiding local optima have become two most important goals in ABC algorithm modification. To overcome the issues in ABC algorithm and achieve the two goals above, inspired by DE, a new search mechanism is proposed in the improved artificial bee colony (IABC) algorithm. In order to balance the exploration ability and the exploitation ability, two search equations are designed based on DE search strategies, and the usage of these two search equation is according to whether a probability p is larger than a uniformly distributed random number. In addition, to enhance the convergence speed, the orthogonal initialization is employed [15]. Experimental results and comparisons denote the effectiveness and efficiency of the proposed IABC algorithms.

The rest of the paper is organized as follows. In Section 2, ABC algorithm is summarized briefly. In Section 3, the proposed improved artificial bee colony algorithm is described. In Section 4, experiments are presented and the results are discussed. Finally, a conclusion is provided in Section 5.

2. Overview of Artificial Bee Colony Algorithm

In 2005, Karaboga proposed a new swarm intelligence-based algorithm for numerical function optimization, artificial bee colony (ABC) algorithm, which simulates the foraging behavior of bee colonies [6]. In ABC algorithm, the colony consists of three kinds of bees: employed bees, onlooker bees and scout bees. Half of the colony is employed bees, and the other half is onlooker bees. The employed bees explore the food source and share the information of the food source with the onlooker bees. Based on the information shared by the employed bees, the onlooker bees choose a food source to exploit. The employed bee whose food source has been abandoned becomes a scout bee. The position of a food source is a possible solution to the optimization problem. The main steps of ABC algorithm can be described as follows:

Initialization

Repeat

Employed bee stage: Place the employed bees on the food sources in the memory.

Onlooker bee stage: Place the onlooker bees on the food sources in the memory.

Scout bee stage: Send the scout bees to the search area for discovering new food sources.

Until (conditions are satisfied).

Denote the food source number as SN, the position of the i th food source as x_i $(i=1,\cdots,SN)$, which is a D-dimensional vector. In the initial stage, each food source is generated as follows:

$$x_{ij} = x_j^{\min} + rand(0,1)\left(x_j^{\max} - x_j^{\min}\right), \tag{1}$$

where $x_j^{\max}$ and $x_j^{\min}$ are the upper and lower bounds of the j th dimension of the problem's search space. These food sources are randomly assigned to the employed bees.

In the employed bee stage and onlooker bee stage, in order to produce a candidate solution v_{ij} from the old one x_{ij}, the new candidate solution v_{ij} can be generated as:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \tag{2}$$

where $k \in \{1, 2, \cdots, SN\}$, $k \neq i$ and $j \in \{1, 2, \cdots, D\}$ are randomly selected indices, $\phi_{ij} \in [-1, 1]$ is a uniformly distributed random number. The candidate solution is compared with the old one, and the better one should be remained.

In ABC algorithm, the i th fitness value $fitness_i$ for a minimization problem is defined as:

$$fitness_i = \begin{cases} \dfrac{1}{1+f_i}, & f_i \geq 0, \\ 1 + abs(f_i), & f_i < 0, \end{cases} \tag{3}$$

where f_i is the cost value of the i th solution.

In the onlooker bee stage, the probability of a food source being selected by an onlooker bee is given by:

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i}. \tag{4}$$

If the abandoned food source is x_i, the scout bee exploits a new food source according to equation (1).

3. Improved Artificial Bee Colony Algorithm

3.1 Orthogonal initialization

Population initialization is an important step in swarm intelligence-based algorithms, which can affect the quality of solution. It is desirable that the initial population be scattered uniformly over the feasible solution space, so that the algorithm can search the whole solution space evenly. Before an optimization problem is solved, there is no information about the location of the solution. Notice that an orthogonal array specifies a small number of combinations that are scattered uniformly over the space of all possible combinations. The orthogonal design can make the initial population be scattered evenly over the solution space. Therefore, in this paper we generate initial population by using the orthogonal initialization method described in [15, 16].

The algorithm for generating an initial population is given as follows.

Algorithm 1: Generation of Initial Population.

Step 1: Divide the feasible solution space $[l,u]$ into S subspaces $[l_1,u_1],[l_2,u_2],\cdots,[l_S,u_S]$ based on the following equations:

$$\begin{cases} l_i = l+(i-1)\left(\dfrac{u(s)-l(s)}{S}\right) l_S, \\ u_i = u-(S-i)\left(\dfrac{u(s)-l(s)}{S}\right) l_S, \end{cases} \qquad i=1,2,\cdots,S. \tag{5}$$

Here, $u(s)-l(s)=\max\limits_{1\le i\le D}\{u_i-l_i\}$.

Step2: Quantize subspace $[l_i,u_i]$ into Q_1 levels based on

$$\alpha_{ij}=\begin{cases} l_i, & j=1, \\ l_i+(j-1)\left(\dfrac{u_i-l_i}{Q_1-1}\right), & 2\le j\le Q_1-1, \\ u_i, & j=Q_1, \end{cases} \tag{6}$$

where Q_1 is odd. Then, construct orthogonal array $L_{M_1}(Q_1^N)=\left[a_{ij}\right]_{M_1\times N}$ to select M_1 individuals based on

$$\begin{cases} (\alpha_{1,a_{11}},\alpha_{2,a_{12}},\cdots,\alpha_{N,a_{1N}}) \\ (\alpha_{1,a_{21}},\alpha_{2,a_{22}},\cdots,\alpha_{N,a_{2N}}) \\ \cdots \\ (\alpha_{1,a_{M_1 1}},\alpha_{2,a_{M_1 2}},\cdots,\alpha_{N,a_{M_1 N}}). \end{cases} \tag{7}$$

Here, $L_{M_1}(Q_1^N)$ can be generated as follows. Select the smallest J_1 fulfilling $(Q_1^{J_1}-1)/(Q_1-1)\ge N$. If $(Q_1^{J_1}-1)/(Q_1-1)=N$, then $N'=N$ else $N'=(Q_1^{J_1}-1)/(Q_1-1)$. Then, construct the basic columns based on $j=\dfrac{Q_1^{k-1}-1}{Q_1-1}+1$, $a_{ij}=\left\lfloor\dfrac{i-1}{Q_1^{J_1-k}}\right\rfloor \bmod Q_1$, for $i=1,\cdots,M_1$, $k=1,\cdots J_1$. Construct the non-basic columns as $j=\dfrac{Q_1^{k-1}-1}{Q_1-1}+1$, $a_{j+(s-1)(Q_1-1)+t}=(a_s\times t+a_j)\bmod Q_1$, for $s=1,\cdots,j-1$, $t=1,\cdots,Q_1$. Thus, the orthogonal array $L_{M_1}(Q_1^{N'})$ is constructed. Delete the last $N'-N$ columns of $L_{M_1}(Q_1^{N'})$ to get $L_{M_1}(Q_1^N)$ where $M_1=Q_1^{J_1}$.

Step 3: Among the M_1S individuals, select SN individuals having the smallest cost as the initial population.

3.2 New Search Mechanism

As we all known that how to balance exploration and exploitation abilities to achieve good optimization performance is an important problem for the population based algorithms, such as GA, PSO, DE, and so on. The exploration refers to the ability to search the unknown regions in the solution space to find the global optimum, while the exploitation refers to the ability to discover better solutions based on the information of the previous good solutions. Actually, the exploration and exploitation abilities contradict with each other, so that the two abilities should be well balanced.

In ABC algorithm, the employed bees exploit the new food source and send the information to the onlooker bees. The onlooker bees select one food source to explore based on the information shared by the employed bees. The scout bees explore a new food source to replace the old one which abandoned by employed bees. Therefore, in the ABC algorithm, the employed bee stage and onlooker bee stage represent the exploitation ability of the algorithm, while the scout bee stage represents the exploration ability. While, the search equation proposed in ABC algorithm is good at exploration but poor at exploitation. In order to improve the exploitation, a new solution search mechanism is proposed based on DE.

Differential evolution (DE) is a population based algorithm, whose main strategy is to generate a new position for an individual by calculating vector differences between other randomly selected individuals in the population. It has been shown the efficiency for many optimization problems in real-world applications. It follows the general stages of an evolutionary algorithm. In DE algorithm, three evolutionary operations including mutation, crossover and selection will be executed. There are several kinds of mutation operation, which formulates different DE algorithms. Among them, “DE/rand/2” can effectively maintain population diversity and “DE/best/2” can improve the convergence speed. Both two strategies are used more frequently in literatures, which can be described as follows:

$$\text{DE/rand/2: } v_i = x_{r1} + F(x_{r2} + x_{r3} - x_{r4} - x_{r5}), \tag{8}$$

$$\text{DE/best/2: } v_i = x_{best} + F(x_{r1} + x_{r2} - x_{r3} - x_{r3}), \tag{9}$$

where $i \in \{1, 2, \cdots, SN\}$; $r1$, $r2$, $r3$, $r4$ and $r5$ are different random integer indices selected from $\{1, 2, \cdots, SN\}$; x_{best} is the global best solution; $F \in [0.5,1]$ is a positive real number.

Motivated by DE and based on the property of ABC algorithm, two new solution search equations are proposed as follows:

$$v_{i,j} = x_{r1,j} + \varphi_{i,j}(x_{i,j} - x_{r2,j}) + \phi_{i,j}(x_{r3,j} - x_{r4,j}), \tag{10}$$

$$v_{i,j} = x_{best,j} + \varphi_{i,j}(x_{i,j} - x_{r1,j}) + \phi_{i,j}(x_{r2,j} - x_{r3,j}), \tag{11}$$

where $i \in \{1, 2, \cdots, SN\}$; $r1$, $r2$, $r3$ and $r4$ are integers randomly selected from $\{1, 2, \cdots, SN\}$, and both of them are different from i; $x_{best,j}$ is the global best solution; $j \in \{1, 2, \cdots, D\}$ is a randomly selected index; and $\varphi_{ij} \in [-1,1]$ and $\phi_{ij} \in [0.5,1]$ are uniformly distributed random number.

Similar to DE, the search equation (10) can maintain population diversity efficiently, and the search equation (11) can improve the convergence performance which means can improve the exploitation ability of the algorithm. Therefore, in order to take advantage of these two search equations and avoid the shortages of them, a new search mechanism is proposed by hybridizing the two search equations. In the new search mechanism, a selective probability p is introduced to control the frequency of using search equation (10) and (11).

Moreover, as can be seen from the ABC search equation, there is only one different element between the new food source and the old one, which is short of sufficient and new information such that the search efficiency is poor. To overcome this issue, in the new search mechanism, we use the search equation (10) and (11) for every element of food source. Thus, the main steps of the new search mechanism are given as follows.

Algorithm 2: The new search mechanism

Step 1: Give a food source x_i and selective probability p. Produce new food source v_i.

Step 2: If $rand < p$, then using equation (10) to generate a new food source v_i based on the old one x_i.

Step 3: If $rand \geq p$, then using equation (11) to generate a new food source v_i based on the old one x_i.

3.3 The proposed algorithm

Based on the above analysis, the main procedure of improved artificial bee colony algorithm is as follows.

Algorithm 3: Improved artificial bee colony algorithm

Initialize the food sources by Algorithm 1 proposed in subsection 3.1, evaluate the population, and memorize the best food sources found so far, $trail_i = 0$, $(i = 1, 2, \cdots, SN)$. $Cycle = 1$.

Repeat

Step 1: Search the new food source for employed bee according to Algorithm 2 and evaluate its quality.

Step 2: Apply a greedy selection process and select the better solution between the new food source and the old one.

Step 3: If solution does not improve $trail_i = trail_i + 1$, otherwise $trail_i = 0$.

Step 4: Calculate the probability according to (3) and apply roulette wheel selection scheme to choose a food source for onlooker bees.

Step 5: Search the new food source for onlooker bees according to Algorithm 2 and evaluate its quality.

Step 6: Apply a greedy selection process and select the better solution between the new food source and the old one.

Step 7: If solution does not improve $trail_i = trail_i + 1$, otherwise $trail_i = 0$.

Step 8: If $\max(trail_i) > \text{limit}$, replace this food source with a new food source produced by equation (1).

Memorize the best solution achieved so far.

$Cycle = Cycle + 1$

Until($Cycle =$ Maximum Cycle Number)

4. Numerical Experiments

4.1 Test functions

In this section, the IABC algorithm proposed in this paper is applied to minimize 27 benchmark functions, as shown in Table 1 and 2. In Table 1, the dimensions of the benchmark functions are given in the third column.

Table 1. Benchmark functions f_1- f_{16} used in experiments. D: Dimension, C: Characteristic, U: Unimodal, M: Multimodal, S: Separable, N: Non-Separable.

Function	Range	D	C	Formulation
Beale	[-4.5, 4.5]	2	UN	$f_1 = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$
Bohachevsky	[-100, 100]	2	MS	$f_2 = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$
Booth	[-10,10]	2	MS	$f_3 = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
Branin	[-5,10] × [0,15]	2	MS	$f_4 = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$
Colville	[-10,10]	4	UN	$f_5 = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2)$
Easom	[-100, 100]	2	UN	$f_6 = -\cos x_1 \cos x_2 \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$
GoldStein-Price	[-2,2]	2	MN	$f_7 = \begin{bmatrix} 1 + (x_1 + x_2 + 1)^2 \\ (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \end{bmatrix} \cdot \begin{bmatrix} 30 + (2x_1 - 3x_2)^2 \\ (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \end{bmatrix}$
Hartman3	[0,1]	3	MN	$f_8 = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right]; c = [1.0, 1.2, 3.0, 3.2]$
Six Hump Camel Back	[-5,5]	2	MN	$f_9 = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
Matyas	[-10,10]	2	UN	$f_{10} = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$
Perm	[- D, D]	2	MN	$f_{11} = \sum_{k=1}^{n}\left[\sum_{i=1}^{2}(i^k + 0.5)((x_i/i)^k - 1)\right]^2$

Powell	[-4,5]	4	UN	$f_{12}=\sum_{i=1}^{n/k}(x_{4i-3}+10x_{4i-2})^2+5(x_{4i-1}+10x_{4i})^2+(x_{4i-2}+10x_{4i-1})^4 +10(x_{4i-3}+10x_{4i})^4$
PowerSum	[0, D]	24	MN	$f_{13}=\sum_{k=1}^{n}\left[\sum_{i=1}^{4}(x_i^k)-b_k\right]^2$; $b=[8,18,44,114]$
Shekel	[0,10]	4	MN	$f_{14}=-\sum_{j=1}^{m}\left[\sum_{i=1}^{4}(x_i-a_{ij})^2+c_i\right]^{-1}$
Shubert	[-10,10]	2	MN	$f_{15}=\left(\sum_{i=1}^{5} i\cos((i+1)x_1+i)\right)\cdot\left(\sum_{i=1}^{5} i\cos((i+1)x_2+i)\right)$
Trid6	[-36,36]	6	UN	$f_{16}=\sum_{i=1}^{n}(x_i-1)^2-\sum_{i=2}^{n}x_i x_{i-1}$

In function Hartman3, $a=\begin{bmatrix}3.0 & 10 & 30\\ 0.1 & 10 & 35\\ 3.0 & 10 & 30\\ 0.1 & 10 & 35\end{bmatrix}^T$, $p=\begin{bmatrix}0.3689 & 0.1170 & 0.2673\\ 0.4699 & 0.4387 & 0.7470\\ 0.1091 & 0.8732 & 0.5547\\ 0.03815 & 0.5743 & 0.8828\end{bmatrix}^T$; and in function Shekel, $a=\begin{bmatrix}4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0\\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 5.0 & 1.0 & 2.0 & 3.6\\ 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 3.0 & 8.0 & 6.0 & 7.0\\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6\end{bmatrix}$, $c=\frac{1}{10}[1,2,2,4,4,6,3,7,5,5]^T$. The benchmark functions presented in Table 2 are tested of dimension $D=30$.

Table 2. Benchmark functions f_{17} - f_{27} used in experiments. C: Characteristic, U: Unimodal, M: Multimodal, S: Separable, N: Non-Separable, $n=D$.

Function	Range	C	Formulation
Ackley	[-32, 32]	MN	$f_{17}=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$
Dixon-Price	[-10, 10]	UN	$f_{18}=(x_1-1)^2+\sum_{i=2}^{n}i(2x_i^2-x_{i-1})^2$
Griewank	[-600, 600]	MN	$f_{19}=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$
Levy	[-10, 10]	MN	$f_{20}=\sin^2(\pi y_1)+\sum_{i=1}^{n-1}\left[(y_i-1)^2(1+10\sin^2(\pi y_i+1))\right] +(y_n-1)^2(1+10\sin^2(2\pi y_n))$ $y_i=1+\frac{x_i-1}{4},\ i=1,\cdots,n.$
Michalewicz	[0, π]	MS	$f_{21}=-\sum_{i=1}^{n}\sin(x_i)\left(\sin(ix_i^2/\pi)\right)^{2m}$, $m=10$

Rastrigin	[-5.12 ,5.12]	MS	$f_{22}=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$
Rosenbrock	[-30, 30]	UN	$f_{23}=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$
Schwefel	[-500, 500]	MS	$f_{24}=\sum_{i=1}^{n}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$
Sphere	[-100, 100]	US	$f_{25}=\sum_{i=1}^{n}x_i^2$
SumSquares	[-10, 10]	US	$f_{26}=\sum_{i=1}^{n}ix_i^2$
Zakharov	[-5, 10]	UN	$f_{27}=\sum_{i=1}^{n}x_i^2+\left(\sum_{i=1}^{n}0.5ix_i\right)^2+\left(\sum_{i=1}^{n}0.5ix_i\right)^4$

4.2 Effects of selective probability p

In this section, we investigate the impact of selective probability p on the new algorithm. Note that the test function Matyas, Powell, PowerSum and Schewefel are representative, so selective probability p is tested according to these four functions. The IABC algorithm runs 30 times on each function, and the mean values of the final results are plotted in Figure 1. As all the test functions are minimization problems, the smaller the mean values, the better it is.

From Figure 1, we can see that the selective probability p can affect the results. For these four test functions, better results are obtained when p is around 0.25. Hence, the selective probability p will be equal to 0.25 for all test functions in the experiments.

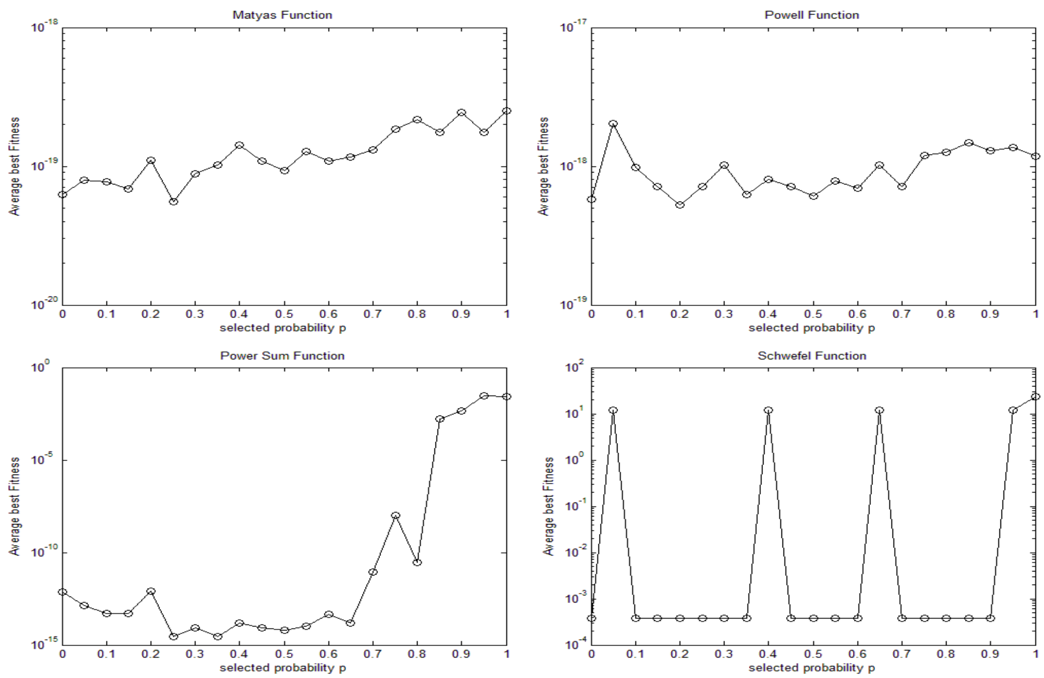


Figure 1: Results on four test functions with different selective probability p

4.3 Comparison of IABC with ABC

In order to verify the performance of IABC algorithm proposed in this paper, this section presents a comparison of IABC algorithm with original ABC algorithm. In the experiments, both IABC and ABC use the same parameter settings. The population size SN, limit, and maximum number of cycle (MSN) are set to 50, $(SN/2)*D$, 1000, respectively. All experiments are repeated 30 times and run 5.0×10^4 function evaluations (FEs) for each test function.

Table 3 shows the optimization results in terms of best, worst, mean and std, which represent the best, the worst, the mean, standard deviation of function value, respectively. The best results are marked in bold.

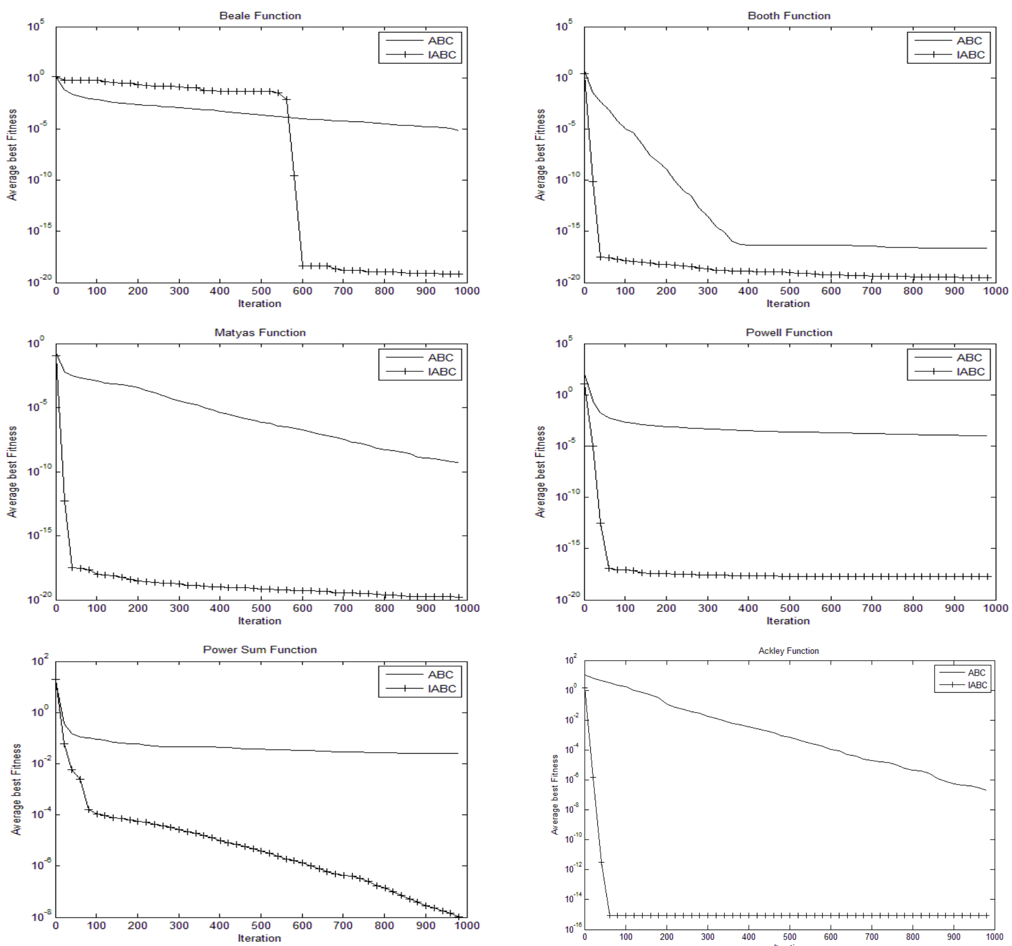
As shown in Table 3, the mean function values of the IABC algorithm are equal or closer to the optimal ones than which of the ABC algorithm, and the standard deviations are relatively small. Particularly, IABC algorithm outperforms ABC algorithm on unimodal function f_1, f_{10}, f_{12}, f_{25}, f_{26}, f_{27} and multimodal function f_3, f_{13}, f_{17}, f_{19}, f_{20}, f_{22}, f_{24}. At the mean time, the two algorithms have the same mean function values on function f_2, f_4, f_7 and f_8, which equal to the optimal ones. All these results show that IABC algorithm has the better performance than ABC algorithm on unimodal and multimodal problems.

In order to show the performance of IABC algorithm more clearly, Figure 2 shows the mean best function value of ten test functions. It is clear that the IABC algorithm has higher convergence rate than the ABC algorithm has.

Table 3. Best, worst, mean and standard deviation obtained by ABC and IABC

	ABC				IABC			
	Best	Mean	Worst	Std	Best	Mean	Worst	Std
f_1	8.80e-07	6.37e-06	5.97e-05	1.10e-05	2.68e-22	**6.62e-20**	2.71e-19	**7.75e-20**
f_2	0.00e+00	**0.00e+00**	0.00e+00	**0.00e+00**	0.00e+00	**0.00e+00**	0.00e+00	**0.00e+00**
f_3	5.82e-20	2.44e-17	7.00e-17	2.27e-17	1.50e-23	**3.09e-20**	1.61e-19	**4.10e-20**
f_4	3.98e-01	**3.98e-01**	3.98e-01	**0.00e+00**	3.98e-01	**3.98e-01**	3.98e-01	**0.00e+00**
f_5	-2.80e+11	-1.10e+10	-1.63e+05	5.02e+10	-1.00e+166	**-5.00e+164**	-4.30e+07	**6.55e+04**
f_6	-1.00e+00	-0.99e+00	-0.99e+00	1.38e-03	-1.00e+00	**-1.00e+00**	-1.00e+00	**0.00e+00**
f_7	3.00e+00	**3.00e+00**	3.00e+00	5.43e-04	3.00e+00	**3.00e+00**	3.00e+00	**2.11e-15**
f_8	-3.86e+00	**-3.86e+00**	-3.86e+00	**2.24e-15**	-3.86e+00	**-3.86e+00**	-3.86e+00	2.63e-15
f_9	4.65e-08	**4.65e-08**	4.65e-08	**4.05e-17**	4.65e-08	4.65e-08	4.65e-08	6.78e-17
f_{10}	8.93e-15	4.58e-10	4.82e-09	1.01e-09	6.54e-22	**1.60e-20**	5.22e-20	**1.53e-20**
f_{11}	1.12e+80	**2.28e+83**	1.93e+84	**4.50e+83**	1.35e+76	4.59e+85	4.28e+86	1.31e+86
f_{12}	8.58e-06	9.57e-05	1.94e-04	4.21e-05	2.26e-20	**1.83e-18**	8.75e-18	**2.36e-18**
f_{13}	1.66e-04	2.47e-02	9.26e-02	2.29e-02	6.51e-16	**7.99e-09**	1.73e-07	**3.16e-08**
f_{14}	-10.5364	**-10.5347**	-10.4967	**7.50e-02**	-10.5364	-9.10373	-5.12848	2.42e+00
f_{15}	-186.731	**-186.731**	-186.731	**3.58e-14**	-186.731	**-186.731**	-186.731	3.84e-14

f_{16}	-50	**-50**	-49.9999	**1.60e-05**	-49.9997	-46.7195	-39.9574	3.27e+00
f_{17}	7.03e-09	1.57e-07	8.94e-07	2.13e-07	8.88e-16	**8.88e-16**	8.88e-16	**0.00e+00**
f_{18}	2.47e-03	**2.88e-02**	0.193e+00	**3.49e-02**	0.16e+00	4.32e-01	6.67e-01	2.55e-01
f_{19}	2.13e-11	3.36e-04	9.86e-03	1.79e-03	0.00e+00	**0.00e+00**	0.00e+00	**0.00e+00**
f_{20}	1.07e-14	1.82e-13	6.51e-13	1.45e-13	5.98e-20	**1.94e-18**	1.50e-17	**2.87e-18**
f_{21}	-29.3027	**-28.9797**	-28.6755	**1.68e-01**	-17.4652	-13.8766	-11.6965	1.49e+00
f_{22}	3.50e-08	2.53e-01	1.05e+00	4.09e-01	0.00e+00	**0.00e+00**	0.00e+00	**0.00e+00**
f_{23}	3.55e-02	**9.49e-01**	7.16e+00	1.42e+00	3.12e-20	6.75e+00	2.84e+01	**1.12e+01**
f_{24}	5.20e-04	2.77e+02	5.91e+02	1.48e+02	3.82 e-04	**3.82e-04**	3.82 e-04	**0.00e+00**
f_{25}	3.86e-15	2.33e-14	1.09e-13	2.42e-14	3.99e-20	**2.39e-18**	9.55e-18	**2.62e-18**
f_{26}	1.04e-13	2.43e-12	1.62e-11	2.94e-12	1.01e-19	**2.00e-18**	7.28e-18	**2.07e-18**
f_{27}	2.16e+02	2.77e+02	3.45e+02	3.17e+01	2.00e-18	**4.19e-01**	4.11e+00	**9.69e-01**



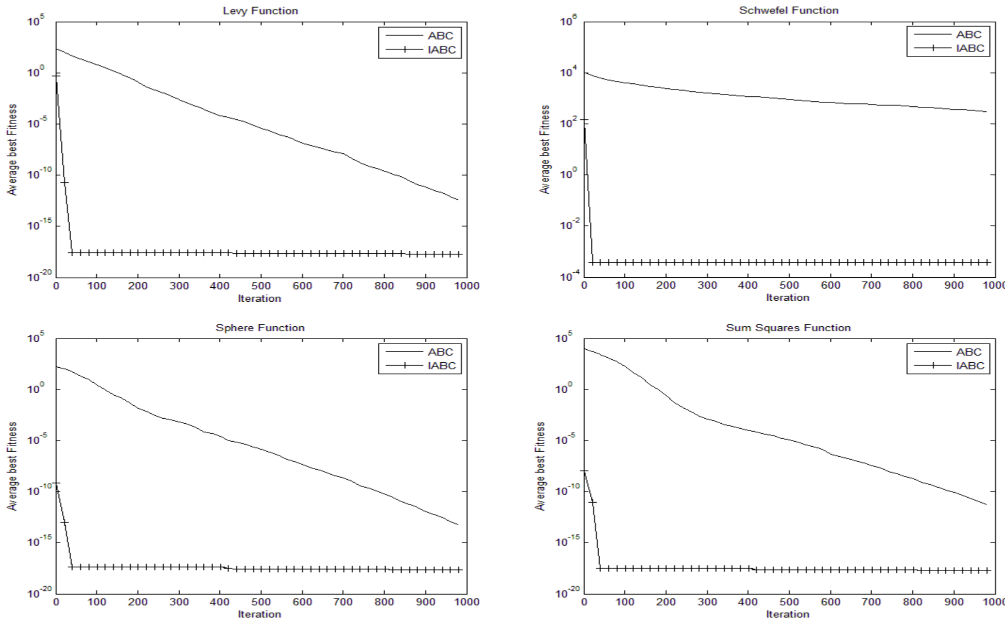
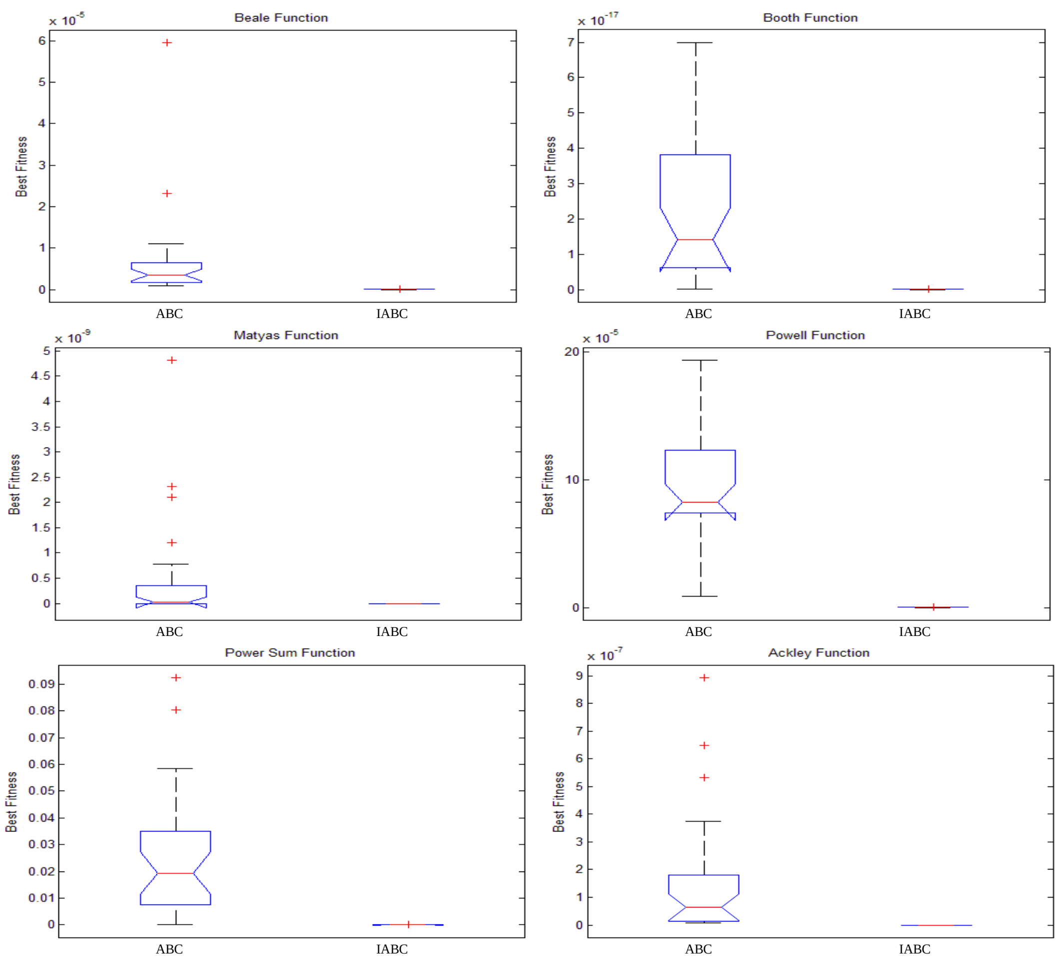


Figure 2. The convergence processes of ABC and IABC on some test functions



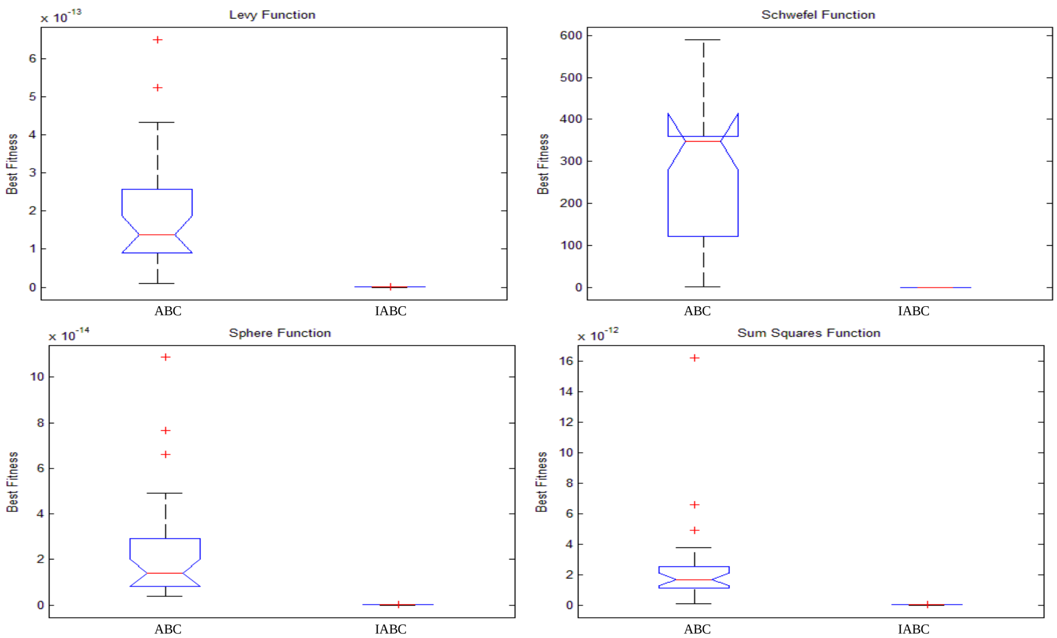


Figure 3. Statistical values of the function values of ABC and IABC on some test functions

Figure 3 shows the statistical results of the function values for ten test functions. Here, box plots are used to illustrate the distribution of these samples obtained from 30 independent runs. The upper and lower ends of the box are the upper and lower quartiles. The line within the box represents the median, and thin appendages summarize the spread a shape of the distribution. Symbol "+" indicate for outlier and the notches denote a robust estimation of the uncertainty about the medians for box-to-box comparison. From Figure 3, we can see that IABC algorithms can obtain the better and more stable solutions than ABC algorithm does, which further verifies the discussion obtained in Table 3 and Figure 2.

4.4 Comparison of IABC with other algorithms

To further test the performance of IABC algorithm, we compare IABC algorithm with other population based algorithms, including some proposed ABC algorithms. The experiments focus on the comparison of IABC algorithm with DE [7], PSO [7], CLPSO [17], CES [18], FES [19], ESLAT [20], CMA-ES [20], GABC [21], I-ABC [22], PS-ABC [22] and NABC [23]. For DE and PSO, the parameters are chosen as in [7]. For the rest algorithms, the parameter settings are followed the original papers of these algorithm.

Table 4. Comparison of IABC with other algorithms

	Ackley	Griewank	Rastrigin	Rosenbrock	Schwefel	Sphere
	Mean	Mean	Mean	Mean	Mean	Mean
DE	3.99e-08	6.15e-04	1.47e+02	4.71e+03	7.27e+03	3.43e-14
PSO	3.23e-01	1.34e-02	3.85e+01	5.74e+03	4.16e+03	2.13e-16
CLPSO	2.01e-12	6.45e-13	2.57e-11	1.10e+01	1.19e+01	1.89e-19
CES	6.00e-13	6.00e-14	1.34e+01	2.77e+01	4.57e+03	**1.70e-26**
FES	1.20e-02	3.70e-02	1.60e-01	3.33e+01	1.31e+01	2.50e-04

ESLAT	1.80e-08	1.40e-03	4.65e+00	1.93e+00	1.03e+04	2.00e-17
CMA-ES	6.90e-12	7.40e-04	5.18e+01	**4.00e-01**	4.93e+03	9.70e-23
IABC	**8.88e-16**	**0.00e+00**	**0.00e+00**	6.75e+00	**3.82e-04**	2.39e-18
Sig.	+	+	+	·	+	·

Table 5. Comparison of IABC with other ABC algorithms

Fun	GABC	I-ABC	PS-ABC	NABC	IABC	Sig.
	Mean	Mean	Mean	Mean	Mean	
Ackley	7.78e-10	**8.88e-16**	**8.88e-16**	1.07e-13	**8.88e-16**	NA
Griewank	6.96e-04	**0.00e+00**	**0.00e+00**	1.11e-16	**0.00e+00**	NA
Rastrigin	3.31e-02	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	NA
Rosenbrock	7.48e+00	2.64e+01	1.59e+00	**1.45e-01**	6.75e+00	·
Schwefel	1.62e+02	3.18e+02	5.30e+00	5.73e-01	**3.82e-04**	+
Sphere	6.26e-16	**0.00e+00**	**0.00e+00**	5.43e-16	2.39e-18	·

Table 4 presents the comparison results of DE, PSO, CES, FES, ESLAT, CMA-ES and IABC. Table 5 represents the comparison results of IABC with four other ABC algorithms. In Table 4, results of DE, PSO, CES, FES, ESLAT and CMA-ES can be found in [7]. Results of GABC, I-ABC, PS-ABC and NABC in Table 5, can be found in [23]. In Table 4 and 5, the best results among these algorithms are shown in bold. The last row of Table 4 and the last column of Table 5 show the statistical significance level of the difference of the mean value of IABC and the best algorithm among the other algorithms in the table. Here, "+" indicates the t value of 49 degrees of freedom which is significant at a 0.05 level of significance by two-tailed test; " · " represents the difference of mean values which is not statistically significant; and "NA" means two algorithms achieve the same accuracy results [24].

From Table 4, it can be seen that IABC algorithm outperforms DE, PSO, and FES on all six test functions. For function Sphere, CLPSO, CES and CMA-ES achieve better results; while for function Rosenbrock, ESLAT and CMA-ES obtain better solutions. Except function Sphere and Rosenbrock, for the rest functions, IABC algorithm performs better than CLPSO, CES, ESLAT and CMA-ES. The comparison results show that some algorithms perform better than IABC on unimodal functions, such as function Sphere and Rosenbrock. On all multimodal functions, *i.e.*, Ackley, Griewank, Rastrigin and Schwefel, IABC outperforms these algorithms compared in Table 4.

From the results in Table 5, IABC outperforms GABC on all six test functions. Even more, IABC outperforms all four ABC algorithms on function Schwefel. For function Ackley, IABC performs better than NABC and the same as I-ABC and PS-ABC. For function Griewank and Rastrigin, IABC has the same results with I-ABC and PS-ABC which achieve the optimal results, while I-ABC and PS-ABC perform better than IABC on Rosenbrock and Sphere. Compared with NABC, IABC outperforms on all six functions except function Rosenbrock.

5. Conclusion

Artificial bee colony is a new swarm-based optimization technique which has shown to be competitive to other population-based stochastic algorithms. However, ABC and other stochastic algorithms suffer from the same problems, such as lower convergence speed and easily trapped in local optima when handling complex multimodal problems. The main reason is that the search pattern is good at exploration but poor at exploitation. To overcome this issue, an improved artificial bee colony algorithm (IABC) is proposed. In IABC algorithm, orthogonal initial method is employed and a new search mechanism is designed.

To verify the performance of the proposed algorithm, a set of 27 test functions are used in the experiments. Comparison of IABC with ABC indicates that IABC can effectively accelerate the convergence speed and improve the accuracy of solutions. Another comparison denotes that IABC is significantly better or at least comparable to other population-based algorithms. We also compared the results obtained by IABC algorithm with other improved ABC algorithms. The experimental results show that IABC algorithm performs better. Therefore, the IABC algorithm proposed in our paper is more effective for global optimization problems.

Acknowledgements

This work is supported by 2013 Narure Science Foundation of Ningxia (No. NZ13096), 2013 scientific research project of Beifang University of Nationalities (2013XYZ021), institute of information and system computation science of Beifang University (13xyb01), National Nature Science Foundation of China (No. 61373174) and Foundation of State Key Lab. of Integrated Services Networks of China.

References

[1] K. S. Tang, K. F. Man, S. Kwong and Q. He, "Genetic algorithms and their applications", IEEE Signal Processing Magazine, vol. 13, no. 6, **(1996)**, pp. 22-37.
[2] X. Xue and Y. Gu, "Global Optimization Based on Hybrid Clonal Selection Genetic Algorithm for Task Scheduling", Journal of Computational Information Systems, vol. 6, no. 1, **(2010)**, pp. 253-261.
[3] M. Dorigo and T. Stutzle, "Ant colony optimization", Cambridge: MAMIT Press, **(2004)**.
[4] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, vol. 11, no. 4, **(1997)**, pp. 341-359.
[5] J. Kennedy and R. Eberhart, "Particle swarm optimization", In IEEE International Conference on Neural Networks, **(1995)**, pp. 1942-1948.
[6] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Technical report-tr06, Kayseri, Turkey: ErciyesUniversity, **(2005)**.
[7] D. Karaboga and B. Basturk, "A comparative study of artificial bee colony algorithm", Applied Mathematics and Computation, vol. 214, no. 1, **(2009)**, pp. 108-132.
[8] A. Singh, "An artificial bee colony algorithm for the leaf constrained minimum spanning tree problem", Applied Soft Computing Journal, vol. 9, no. 2, **(2009)**, pp. 625-631.
[9] Q. K. Pan, M. F. Tasgetiren, P. Suganthan and T. Chua, "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem", Information Sciences, vol. 181, no. 12, **(2011)**, pp. 2455-2468.
[10] F. Kang, J. Li and Q. Xu, "Structural inverse analysis by hybrid simplex artificial bee colony algorithms", Computers and Structures, vol. 87, no. 13, **(2009)**, pp. 861-870.
[11] R. S. Rao, S. V. L. Narasimham and M. Ramalingaraju, "Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm", International Journal of Electrical Power and Energy Systems Engineering, vol. 1, no. 2, **(2008)**, pp. 116-122.
[12] W. Zou, Y. Zhu, H. Chen and X. Sui, "A Clustering Approach Using Cooperative Artificial Bee Colony Algorithm", Discrete Dynamics in Nature and Society, vol. 2010, Article ID 459796, 16 pages, **(2010)**. doi:10.1155/2010/459796.
[13] Z. Hu and M. Zhao, "Simulation on traveling salesman problem (TSP) based on artificial bees colony algorithm", Transaction of Beijing Institute of Technology, vol. 29, no. 11, **(2009)**, pp. 978-982.
[14] G. Cabrera, E. Cabrera, R. Soto, L. J. M. Rubio, B. Crawford and F. Paredes, "A Hybrid Approach Using an Artificial Bee Algorithm with Mixed Integer Programming Applied to a Large-Scale Capacitated Facility Location Problem", Mathematical Problems in Engineering, vol. 2012, Article ID 954249, 14 pages, **(2012)**, doi:10.1155/2012/954249.
[15] Y. W. Leung and Y. P. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization", IEEE Transactions of Evolutionary Computation, vol. 5, no. 1, **(2001)**, pp. 41-53.
[16] X. Kong, *et al.*, "Hybrid Artificial Bee Colony Algorithm for Global Numerical Optimization", Journal of Computational Information System, vol. 8, no. 6, **(2012)**, pp. 2367-2374.

[17] J. J. Liang, A. K. Qin, P. N. Suganthan and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions", IEEE Transactions of Evolutionary Computation, vol. 10, **(2006)**, pp. 281-295.
[18] X. Yao and Y. Liu, "Fast evolution strategies", Control and Cybernetics, vol. 26, no. 3, **(1997)**, pp. 467-496.
[19] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation", In Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC' 96), **(1996)** May, pp. 312-317.
[20] A. Hedar and M. Fukushima, "Evolution strategies learned with automatic termination criteria", In Proceedings of the Conference on Soft Computing and Intelligent Systems and the International Symposium on Advanced Intelligent Systems, Tokyo, Japan, **(2006)**, pp. 1-9.
[21] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization", Applied Mathematics and Computation, vol. 217, no. 7, **(2010)**, pp. 3166-3173.
[22] G. Li, P. Niu and X. Xiao, "Development and investigation of efficient artificial bee colony algorithm for numerical function optimization", Applied Soft Computing, vol. 12, no. 1, **(2012)**, pp. 320-332.
[23] Y. Xu, P. Fan and L. Yuan, "A Simple and Efficient Artificial Bee Colony Algorithm", Mathematical Problems in Engineering, vol. 2013, Article ID 526315, 9 pages, **(2013)**, doi:10.1155/2013/526315.
[24] S. Das, A. Abraham, U. K. Chakraborty and A. Konar, "Differential evolution using a neighborhood-based mutation operator", IEEE Transactions on Evolutionary Computation, vol. 13, no. 3, **(2009)**, pp. 526-553.

Authors

Xiangyu Kong. He received his B.A. in Applied Mathematics from Hunan University, Changsha, China, in 2004, the M.S. degree in Applied Mathematics from Xidian University, Xi'an, China, in 2009. Since the year of 2011, he started his learning in Applied Mathematics from Xidian University for his Ph. D. degree. He is currently with Institute of Information and System Computation Science, Beifang University of Nationalities, China. His main interests include Intelligent Optimization, Theory and Methods of Optimization.

Sanyang Liu. He is a professor and a tutor of Ph.D. in Xidian University. His current research interests include optimization theory and algorithm, machine learning and pattern recognition.

Zhen Wang. Her lecturer received her Ph.D. degree in Applied Mathematics from Xidian University in 2012. She is currently with Institute of Information and System Computation Science, Beifang University of Nationalities, China. Her main research areas include Mathematical Finance, Portfolio Optimization, and Intelligent Optimization.

