Multi-strategy artificial bee colony based on multiple population for coverage optimisation

Hui Sun*, Kun Wang and Haihua Xie

School of Information Engineering,
Nanchang Institute of Technology,
Nanchang 330099, China
Email: sun_hui2006@163.com
Email: wk880819@163.com
Email: pxlh_xhh@163.com
*Corresponding author

Abstract: In order to overcome the shortcomings of weak local search ability and slow convergence speed for the standard artificial bee colony algorithm, this paper proposes an improved multi-strategy artificial bee colony algorithm based on multiple populations (IMSABC). Firstly, the employed bees are randomly divided into three subgroups, corresponding to three evolutionary strategies. If the candidate solution obtained from searching is inferior to the current honey source, the bee is randomly assigned to other subgroups and the search strategy is changed. In this way, it not only facilitates the information exchange between populations, but also balances the global search and local development capabilities of the algorithm since the three search strategies have different characteristics. Secondly, by imitating the particle swarm algorithm, the search strategy of the following bees is improved by using the abundant information contained in the current global optimal honey source and random neighbour honey source. The simulation results of twelve benchmark test functions and 28 CEC2013 functions show that the performance of this algorithm has significant advantages compared with many similar improved algorithms. In order to improve the unreasonable distribution of sensor nodes and improve the network coverage, the above algorithm is applied to optimise the coverage of wireless sensor networks and achieve better optimisation effect.

Keywords: artificial bee colony; multiple populations; random selection strategy.

Reference to this paper should be made as follows: Sun, H., Wang, K. and Xie, H. (2018) 'Multi-strategy artificial bee colony based on multiple population for coverage optimisation', *Int. J. Wireless and Mobile Computing*, Vol. 14, No. 1, pp.47–55.

Biographical notes: Hui Sun received the BSc degree from the Hydropower Institute of Jiangxi, Nanchang, China in 1982, the MSc degree in mechanics from the Tsinghua University, Beijing, China in 1988, and the PhD degree in metal plastic working from the Nanchang University, Nanchang, China in 2002. He is currently a Professor with Nanchang Institute of Technology, Nanchang, China. His research interests include intelligent algorithms, rough sets and granular computing, variation inequality principle.

Kun Wang is currently working toward the MSc degree at the School of Information Engineering, Nanchang Institute of Technology, China. His research interest focuses on swarm intelligence.

Haihua Xie is currently working toward the MSc degree at the School of Information Engineering, Nanchang Institute of Technology, China. His research interest focuses on swarm intelligence.

1 Introduction

The feature of group intelligence is that it shows macro intelligent behaviour through the collaboration of individuals. By imitating the group behaviour of the populations, researchers designed group intelligence algorithms such as ant colony algorithm (Dorigo and Caro, 1999), cat colony algorithm (Chu and Tsai, 2007), firefly algorithm (Yang, 2010), bacterial algorithm (Passino, 2002) and particle swarm optimisation algorithm (Kennedy and Eberhart, 1995). These algorithms are used to solve the complex optimisation problems in reality.

Inspired by the foraging behaviour of bee populations, Karaboga and Akay (2005) proposed the artificial bee colony algorithm (ABC) in 2005. The algorithm has the advantages of simple structure, few parameters and strong robustness. At present, the artificial bee colony algorithm has been widely used in many complex function optimisation fields (Gao et al., 2013; Karaboga et al., 2014). It has achieved good results when dealing with the problems of data mining (Shukran et al., 2011), target recognition (Ma et al., 2011), mixed flow shop scheduling (Pan et al., 2013) and other issues (Babar and Crăciunescu, 2014; Bilal, 2013).

Although there is a lot of research work on the ABC algorithm, the theory and practice of the algorithm are still not well developed. As a result, many scholars have attempted to improve the ABC algorithm by improving its performance. Gao and Liu (2011) made the distribution of individuals in solution space more purposeful by using the chaos and reverse learning strategies. However, the chaotic distribution strategy leads to strong initial value sensitivity of the algorithm, and to some extent weaken the global search performance of the algorithm. Kang and Ma (2011) proposed a RABC algorithm. In which, the population uses two evolutionary formulas with Rosenbrock rotation perturbations interchangeably. And the algorithm dynamically adjusts the balance between the abilities of global search and local development. Saxena et al. (2014) learns from the local optimal concept in the PSO algorithm, so that the local optimal information of the honey source is involved in the composition of the offspring candidate solution, which improves the search ability of the algorithm in solution space. In order to reduce the probability of falling into local optimum due to the decrease of the population diversity, Xiang and An (2013) proposed the MABC algorithm, which uses fixed parameter P instead of the selection probability in the roulette strategy to improve the algorithm performance.

In order to improve the performance of the ABC algorithm, this paper proposes an improved multi-strategy artificial bee colony algorithm based on multiple populations (IMSABC). Firstly, the employed bees are randomly divided into three subgroups, corresponding to three evolutionary strategies. And the bees are allowed to move between different subgroups. The three search strategies have different characteristics, which can balance the global search and local development of the algorithm. Secondly, by imitating the particle swarm algorithm, the update strategy of the onlooker bees is improved, and the abundant information contained in the current global optimal honey source and random neighbour honey source is fully utilised. The search strategy of the following bees is optimised. The simulation results of twelve benchmark test functions and 28 CEC2013 functions show that the IMSABC algorithm can effectively improve the performance of the ABC algorithm. Compared with some well-known improved ABC algorithms, the new proposed algorithm has significant improvements in both convergence speed and accuracy.

2 Standard artificial bee colony algorithm

In the ABC algorithm, the honey source represents the candidate solution. The degree of excellence depends on the fitness value determined by the optimisation problem (Metlicka and Davendra, 2014). The whole population is divided into three categories: employed bees, onlooker bees and scout bees (Das et al., 2013). The employed bees search the honey source in the whole solution space, and deliver the obtained information to the onlooker bees. The onlooker bees choose to search in the vicinity of better honey source by the way of roulette (Ran and Mesut, 2013), looking for better solution greedily. Throughout the process, individual bees exchange information constantly, so as to find a better honey source, and ultimately solve the problem.

2.1 Initialise the populations

For a D-dimensional optimisation problem, assuming that the total number of bee populations is SN, the number of honey sources is FN, $SN = 2FN$. The vector $x_i = \{x_{i,1}, x_{i,2}, .., x_{i,D}\}$ represents the location of i-th honey source. The initial honey source is produced according to equation (1).

$$x_{i,j} = x_{\min,j} + rand(0,1)(x_{\max,j} - x_{\min,j}) \tag{1}$$

where $i \in \{1, 2, ..., FN\}$, $j \in \{1, 2, ..., D\}$. $x_{\min,j}$ and $x_{\max,j}$ are the lower and upper limits of the j-th dimensional solution, respectively.

2.2 The stages of employed bees

In order to find high quality honey source, the employed bees conduct searching according to equation (2). The new generated candidate honey source is compared with the original one, and the better one is selected to stay in the next generation (Gao et al., 2012).

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) \tag{2}$$

where $v_{i,j}$ represents the new generated candidate honey source, $x_{k,j}$ represents the location of the neighbor honey source. Random number $\phi_{i,j} \in [-1, 1]$, random number $k \in \{1, 2...FN\}$, and $i \neq k$.

2.3 The stages of onlooker bees

According to the information collected by the employed bees, the onlooker bees select the high quality honey source using the roulette method and conduct searching around the honey source. The update is achieved with equation (2). The greedy strategy is adopted to ensure that higher quality honey source enters into the next generation.

The quality of the i-th honey source is represented by the fitness fit_i:

$$fit_i = \begin{cases} \dfrac{1}{1+f_i} & f_i \geq 0 \\ 1 + |f_i| & f_i < 0 \end{cases} \tag{3}$$

where f_i is the fitness value of the i-th honey source. It depends on the quality of the honey source.

The probability that the *i*-th honey source is selected:

$$p_i = \frac{fit_i}{\sum_{i=1}^{FN} fit_i} \tag{4}$$

where p_i is the probability that the *i*-th honey source is selected, and fit_i represents the fitness of the *i*-th honey source.

2.4 The stages of scout bees

If a better honey source is not found after *limit* times of searches in the neighbourhood of a honey source, the honey source is abandoned, and the corresponding employed bees turn to scout bees. A new honey source is generated by equation (1) (Halder et al., 2013).

3 Improved artificial bee colony algorithm

3.1 Strategy of multiple populations

The search strategy plays an important role in the performance of the whole algorithm. Compared to the single evolutionary model of standard ABC algorithm, the overall performance of the proposed algorithm improves by using different strategies for searching focus in different stages. Thus, the employed bees are randomly divided into three subgroups in the initialisation process, and each subgroup evolves according to different search strategies. For each evolution of the employed bees, if the obtained candidate solution is better than the current honey source, indicating that the current searching strategy is suitable for the bees at this stage, then keep the bees in the current subgroup. If the obtained candidate solution is inferior to the current honey source, indicating that the current subgroup search strategy is not suitable for the evolution of bees, then the bees are randomly assigned to other subgroups, and the search strategy is changed. In this way, the evolutionary strategy of bees is adjusted adaptively to achieve better search results, and information between subgroups can be effectively communicated.

3.2 The update strategy of employed bees

In the ABC algorithm, the honey source positions are divided into three categories: individual, neighbourhood and the current global optimal honey source. Three basic search strategies can be obtained according to the classification.

$$x_{i,j} = g_j + \Delta_{i,j} \tag{5}$$

$$x_{i,j} = x_{i,j} + \Delta_{i,j} \tag{6}$$

$$x_{i,j} = x_{k,j} + \Delta_{i,j} \tag{7}$$

where $\Delta_{i,j}$ is the step size of each movement for individual bees.

In the IMSABC algorithm, the above three basic search strategies are adopted and improved to balance the global search and local development capability of the algorithm.

The ABC algorithm only updates in one dimension each time, so the algorithm has good global search capability. However, the local development capability is weak, and the convergence speed in the later stages is slow. Inspired by the GABC (Zhu and Kwong, 2010) algorithm, the information of g_{best} is added to the search strategy. g_{best} is the optimal honey source of the entire population in each iteration. Compared with other honey sources, it is most likely to be the one that is closest to the optimal solution of the problem. Using its special position information to guide the convergence of bee colony, it is possible to ensure that the search range of the employed bees is always around the current global optimal honey source. In this way, the convergence speed of the bee colony is accelerated. Therefore, the first improved strategy adopted is as follows:

$$v_{i,j} = g_j + \phi(x_{i,j} - x_{k,j}) \tag{8}$$

where, g_j represents the position information of the current global optimal honey source. The random number $\phi \in [-1,1]$. x_k represents the random neighbour honey source of x_i, and $i \neq k$.

In the standard ABC algorithm, the search of employed bees depends on individual information, and the search is only conducted in the vicinity of the current honey source, which results in good global search capability of the algorithm. Therefore, in the IMSABC algorithm, the second adopted search strategy is as follows:

$$v_{i,j} = x_{i,j} + \varphi(x_{i,j} - x_{k,j}) \tag{9}$$

where $r \in (-1,1)$, random number.

In order to maintain the population diversity, and avoid too early convergence, the random neighbourhood search strategy is adopted. The neighbour honey source information is used to guide the search. To avoid the 'premature' of populations, the update randomness of the employed bees is increased. The third search strategy is as follows:

$$v_{i,j} = x_{k,j} + r(x_{k,j} - x_{l,j}) \tag{10}$$

where, random number $r \in (-1,1)$, x_l represents the random neighbour honey source of x_i, and $i \neq k \neq l$.

All the three search strategies have their own advantages and disadvantages. The strategy of guiding the convergence of bee colony with the current global optimal honey source, i.e. equation (8), has good local search ability. It can guarantee the convergence speed of the algorithm, but the global search ability is poor. The search strategy of the standard ABC algorithm, i.e. equation (9), can guarantee the global search ability of the population in the solution space. However, the convergence speed in the later stages is slow, resulting in low precision for the final solution. The random neighbourhood search strategy, i.e. equation (10), can strengthen the information exchange of individuals between populations. It improves the algorithm's ability of getting out of the local optimum, and maintains the population diversity. Thus, through three evolutionary strategies searches, the employed bees can effectively balance the algorithm's capabilities of global search and local development, which enhances the overall performance of the algorithm.

3.3 The update strategy of the onlooker bees

In the IMSABC algorithm, the improvement idea of the onlooker bees is derived from the standard particle swarm algorithm (Eberhart and Kennedy, 1995). The movement of the onlooker bees is affected by the neighbour honey source and the current global optimal honey source. The evolution strategies are as follows:

$$\begin{aligned} v_{i,j} &= x_{i,j} + (g_j - x_{i,j}) * rand(0,1) * c \\ &+ (x_{k,j} - x_{i,j}) * rand(0,1) * q \end{aligned} \tag{11}$$

where c and q are the acceleration constants, c is 1.5, and the random number $q \in (-1,1)$. The first part of equation (11) is the original position of the onlooker bee. The second part shows that the onlooker bees learn from the current global optimal honey source. It represents the information sharing and mutual cooperation between bees, which is similar to the social learning in the PSO algorithm. Because when the individual learns from the current global optimal honey source, the result must be favourable, so the acceleration constant c is assumed to be positive to ensure that the current global optimal information always guides the movement of onlooker bees positively. The third part is the learning process of onlooker bees from the neighbour honey source. It enhances the information exchange between individuals, maintains the population diversity, and is similar to the self-learning part in the PSO algorithm. Since the comparison result of neighbour honey source and current individual is unknown, the learning performance has uncertainty. Therefore, the acceleration constant q is used to adjust the ratio between the learning from the neighbour honey source and the total learning amount.

3.4 The stages of scout bees

If there are scout bees, the corresponding employed bees are initialised. A new honey source is regenerated by equation (1), and the bee is randomly assigned to a subgroup.

3.5 The flow of the improved bee colony algorithm:

Step 1: Determine the scale of the bee colony *SN*, the number of honey source *FN*, the dimension D of the optimisation problem. Set the limited times *limit*, initialise the population information and randomly divide the subgroups.

Step 2: Update the current global optimum information.

Step 3: According to different subgroups, the employed bees are updated according to equations (8)–(10), resulting in a new candidate honey source. The advantages and disadvantages of honey source is compared according to the fitness, and the selection operation is made.

Step 4: The onlooker bee is updated according to equation (11). The advantages and disadvantages of honey source is compared according to the fitness, and the selection operation is made.

Step 5: If the update number of a honey source exceeds the *limit*, the corresponding employed bee becomes a scout bee. A new honey source is randomly generated according to equation (1), and the bee is reassigned to a subgroup.

Step 6: Terminate the algorithm if it reaches the specified iteration number or the global optimal solution satisfies the problem. Otherwise returns to Step 2.

4 Experimental results and analysis

4.1 Classic benchmark function tests

In order to test the performance of the algorithm, 12 benchmark functions (Gao et al., 2013) listed in Table 1 were used for experiments. The first seven test functions are unimodal functions, each unimodal function has only one extreme point. The purpose of the experiments is to detect the convergence speed and local search ability of the detection algorithm. The latter five are multimodal functions, and each function has a large number of extreme points. It mainly investigates the algorithm's abilities of global search and getting out of the local optimum.

Table 1 Benchmark test functions

Function	*Search space*	*optimum*
Sphere	[–100,100]	0
Schwefel 2.22	[–10,10]	0
Schwefel 1.2	[–100,100]	0
Schwefel 2.21	[–100,100]	0
Rosenbrock	[–30,30]	0
Step	[–100,100]	0
Quartic with noise	[–1.28,1.28]	0
Schwefel 2.26	[–500,500]	–418.98D
Rastrigin	[–5.12,5.12]	0
Ackley	[–32,32]	0
Griewank	[–600,600]	0
Penalised	[–50,50]	0

In the experiments, the five algorithms of IMSABC and the standard ABC, GABC, MABC, MEABC were initialised with unified parameters. The population *SN* = 100, the number of honey sources *FN* = 50, the solution dimension *D* = 30, the maximum number of iterations is 1.5e5, the maximum number of searches limit = 100. In order to reduce the error, every algorithm for each test function is calculated 30 times, and the mean is taken as the result. Table 2 lists the results of the IMSABC, ABC, GABC, MABC, and MEABC algorithms for 12 benchmark test functions (Wang et al., 2014).

Table 2 The comparison results of the five algorithms

Function		*ABC*	*GABC*	*MABC*	*MEABC*	*MMSABC*
Sphere	mean	9.58e-16	1.09e-32	4.02e-40	4.85e-40	**7.11e-43**
	std	1.07e-15	4.01e-32	1.58e-39	2.31e-40	**3.36e-42**
Schwefel 2.22	mean	1.21e-10	6.65e-18	1.67e-21	1.25e-21	**7.68e-23**
	std	2.67e-10	1.03e-17	6.37e-18	3.56e-21	**2.09e-22**
Schwefel 1.2	mean	7.66e3	7.88e3	1.03e4	9.81e3	**4.23e3**
	std	8.84e3	1.40e4	1.20e4	2.49e3	**1.07e4**
Schwefel 2.21	mean	2.25e1	1.76e1	**4.22e0**	4.89e0	1.53e1
	std	2.25e1	1.32e4	**2.62e0**	1.37e0	1.75e1
Rosenbrock	mean	4.17e-1	1.83e0	8.37e-1	2.86e-1	**1.39e-1**
	std	2.01e0	1.69e0	7.08e0	3.48e-1	**7.43e-1**
Step	mean	**0**	**0**	**0**	**0**	**0**
	std	**0**	**0**	**0**	**0**	**0**
Quartic with noise	mean	1.73e-1	8.11e-2	3.18e-2	**2.29e-2**	6.21e-2
	std	2.12e-1	9.80e-2	3.34e-2	**1.38e-2**	7.95e-2
Schwefel 2.26	Mean	–12494.4	**–12569.5**	**–12569.5**	**–12569.5**	**–12569.5**
	std	3.47e2	**6.51e-3**	**2.26e-11**	**1.59e-10**	**1.99e-011**
Rastrigin	mean	1.95e-14	**0**	**0**	**0**	**0**
	std	1.65e-13	**0**	**0**	**0**	**0**
Ackley	mean	1.17e-9	3.49e-14	**2.88e-14**	2.90e-14	3.28e-14
	std	2.66e-9	1.69e-14	**1.37e-14**	1.32e-14	1.71e-14
Griewank	mean	1.66e-7	7.38e-9	**0**	**0**	2.23e-14
	std	4.90e-6	2.17e-7	**0**	**0**	6.54e-13
Penalised	mean	7.44e-16	**1.57e-32**	**1.57e-32**	3.02e-17	**1.57e-32**
	std	6.26e-16	**1.50e-47**	**1.50e-47**	0	**1.50e-47**

Table 3 Results comparison of the five algorithms

Algorithm	*Rankings*
IMSABC	2.17
MEABC	2.46
MABC	2.46
GABC	3.50
ABC	4.42

In the test results, the average value ‘mean’ represents the precision of the algorithm, and the standard deviation ‘std’ reflects the stability of the algorithm. It can be seen from the experimental results that the IMSABC is better than or comparable to the other eight functions, indicating that the algorithm has strong abilities of global search and getting out of local optimum. In the tests of seven unimodal functions, the convergence ability of the population is improved with the guidance of g_{best}. The improved algorithm achieves the best results in the tests of five functions.

The Friedman test was used to analyse the test results. It can effectively determine the performance gap between the algorithms. The smaller the rankings value, the better the algorithm’s performance. From Table 3, compared with ABC, GABC, MABC and MEABC algorithms, the rankings value of the IMSABC algorithm is the smallest, indicating the best overall performance.

Table 4 The comparison results of the five algorithms

Functions	*ABC*	*GABC*	*MABC*	*MEABC*	*IMSABC*
f_1	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
f_2	1.44e+07	1.23e+07	1.47e+07	**1.23e+06**	7.07e+06
f_3	7.87e+08	4.27e+08	1.37e+09	1.40e+08	**1.14e+08**
f_4	7.34e+04	6.32e+04	1.08e+05	8.35e+04	**6.24e+04**
f_5	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
f_6	1.49e+01	1.33e+01	2.10e+01	**1.01e+01**	1.46e+01
f_7	1.23e+02	9.92e+01	1.33e+02	**9.23e+01**	9.36e+01
f_8	**2.09e+01**	**2.09e+01**	**2.09e+01**	**2.09e+01**	**2.09e+01**

Table 4 The comparison results of the five algorithms (continued)

Functions	*ABC*	*GABC*	*MABC*	*MEABC*	*IMSABC*
f_9	2.96e+01	2.96e+01	3.25e+01	**2.88e+01**	2.89e+01
f_{10}	1.18e+00	1.49e+00	2.66e+00	5.57e+00	**2.87e-01**
f_{11}	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
f_{12}	2.70e+02	1.58e+02	1.75e+02	2.07e+02	**1.47e+02**
f_{13}	3.17e+02	2.12e+02	2.45e+02	2.29e+02	**2.09e+02**
f_{14}	4.78e+00	7.65e-01	6.14e+00	1.37e+01	**3.96e-01**
f_{15}	4.10e+03	4.53e+03	3.92e+03	**3.41e+02**	4.05e+03
f_{16}	1.60e+00	2.04e+00	**1.20e+00**	1.44e+00	1.64e+00
f_{17}	**3.04e+01**	**3.04e+01**	**3.04e+01**	**3.04e+01**	**3.04e+01**
f_{18}	3.33e+02	2.50e+02	2.16e+02	**1.80e+02**	2.28e+02
f_{19}	8.82e-01	6.20e-01	4.34e-01	3.94e-01	**1.68e-01**
f_{20}	1.45e+01	**1.39e+01**	1.47e+01	1.56e+01	1.41e+01
f_{21}	**2.05e+02**	2.75e+02	2.78e+02	2.10e+02	2.30e+02
f_{22}	9.05e+01	9.16e+01	8.63e+01	**1.78e+01**	9.62e+01
f_{23}	5.08e+03	5.48e+03	5.19e+03	5.16e+03	**4.70e+03**
f_{24}	2.88e+02	2.82e+02	2.87e+02	**2.81e+02**	**2.81e+02**
f_{25}	3.07e+02	2.96e+02	3.05e+02	**2.74e+02**	2.99e+02
f_{26}	**2.01e+02**	**2.01e+02**	**2.01e+02**	**2.01e+02**	**2.01e+02**
f_{27}	**4.00e+02**	**4.00e+02**	**4.00e+02**	4.02e+02	**4.00e+02**
f_{28}	**2.13e+02**	3.00e+02	3.00e+02	3.00e+02	3.00e+02

Figure 1 The convergence speeds of five kinds of algorithms

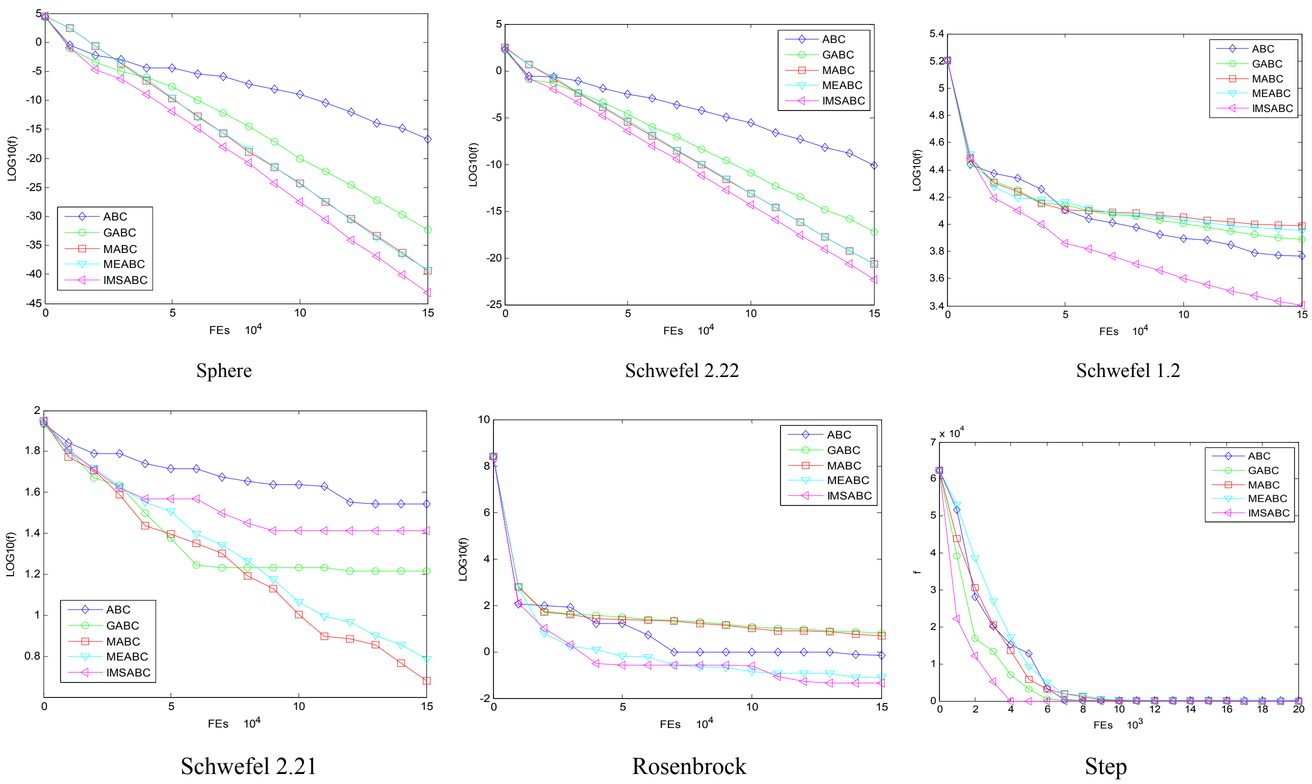
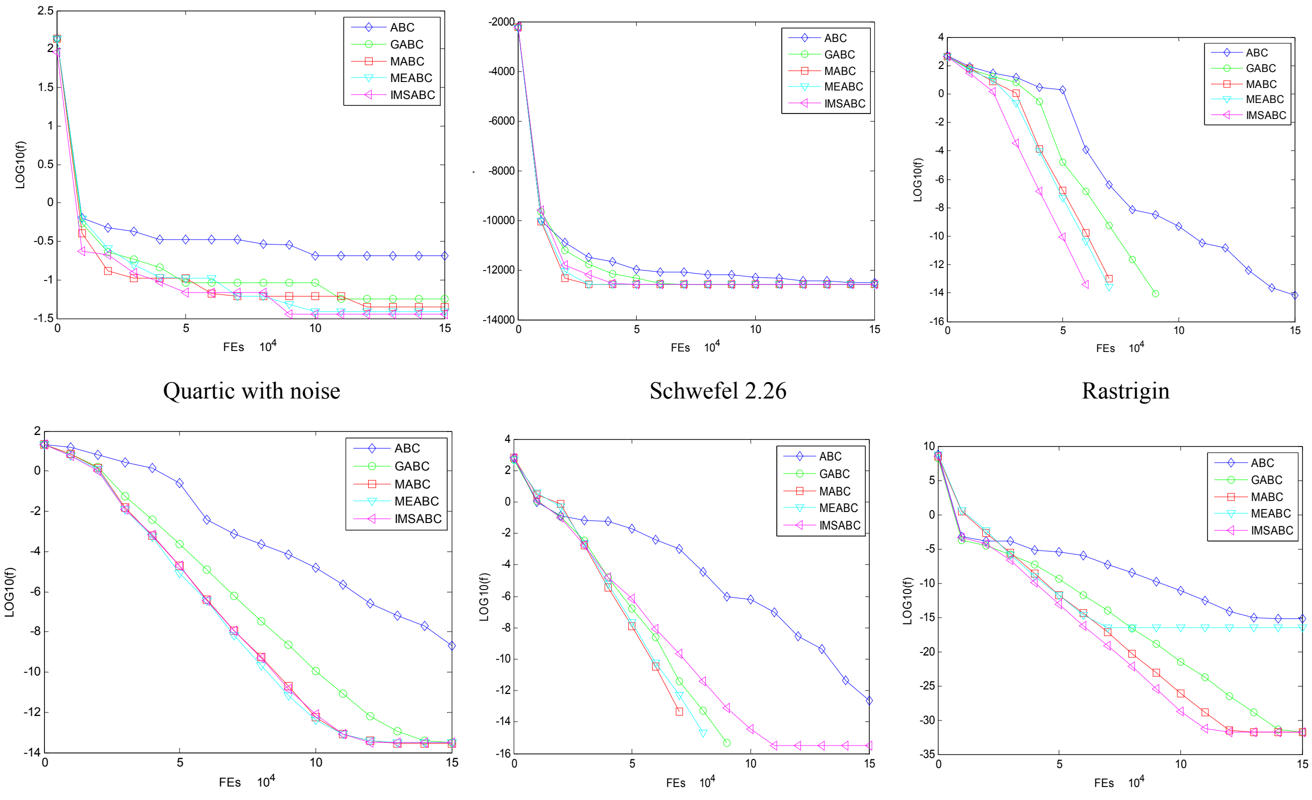


Figure 1 The convergence speeds of five kinds of algorithms (continued)



Quartic with noise | Schwefel 2.26 | Rastrigin

Ackley | Griewank | Penalised

4.2 *Convergence curves*

In order to compare the convergence speed of each algorithm intuitionally, Figure 1 shows the convergence curves of the five algorithms for 12 benchmark test functions. The convergence speed is an important criterion to judge the performance of the algorithm, and it represents the movement process of feasible solution in the algorithm operation process. Among them, the abscissa is the number of evaluations, and the ordinate is the fitness value. Compared to other algorithms, in most of the function tests, the convergence speed of IMSABC has obvious advantages. When solving real problems, if the termination condition of the algorithm is set so that the final solution reaches a certain precision, the improved algorithm will complete the search in the shortest time.

4.3 *CEC2013 complex function tests*

In the twelve benchmark test functions, except for the Schwefel 2.26, the theoretical optimal values of the remaining functions are zero. In order to further verify the effectiveness of the IMSABC algorithm, the more complex CEC2013 function (Zhang et al., 2014) was used for testing. Among the 28 CEC2013 functions, $f_1 - f_5$ are unimodal functions, $f_6 - f_{20}$ are multimodal functions, $f_{21} - f_{28}$ are composite functions.

The parameters of IMSABC, ABC, GABC, MABC, MEABC algorithms are initialised. The populations $SN = 60$, the number of honey sources $FN = 30$, the solution dimension $D = 30$, the maximum number of iterations is 3.0e5, the maximum number of searches limit = 100. For each test function, every algorithm runs 51 times. The obtained result is the difference between the calculated final solution and the theoretical optimum. The error less than 10^{-8} is regarded as 0. The average value of the test results is selected to represent the performance of the algorithm. Table 4 shows the comparison results of IMSABC with four improved bee colony algorithms (Horng and Lin, 2013). From the experimental results it can be obtained that the IMSABC algorithm achieved the best results for 16 test functions.

Table 5 is the Friedman test results for 28 CEC2013 functions. The differences of average rank values between algorithms are large, which indicates that the performance gaps between the algorithms are large. The ranking is: IMSABC, MEABC, GABC, ABC and MABC. The average rank value of the IMSABC algorithm is the lowest, therefore, it has the best overall performance.

Table 5 The comparison results of the five algorithms

Algorithm	*Rankings*
IMSABC	2.34
MEABC	2.61
GABC	3.09
ABC	3.39
MABC	3.57

5 Application of the improved algorithm in wireless sensor network coverage optimisation

A wireless sensor network consists of a large number of sensors deployed in the monitoring area in a self-organised and multi-hop manner that sense, collect, transmit and process the information of monitored objects in a network coverage area in a collaborative manner. Network coverage is an important measure of providing quality of monitoring and target tracking services. In order to verify the effect of the improved algorithm, it is applied to the wireless sensor network coverage optimisation problem. For sensor node deployment optimisation model and parameter settings see Sun and Zhao (2011).

In order to compare and analyse the coverage of the algorithm, we choose chaotic particle swarm optimisation algorithm (Liu and Fan, 2011), particle sharing based particle swarm frog leaping hybrid optimisation algorithm (Sun and Zhao, 2011) and the proposed algorithm to deploy sensor nodes in the monitoring area. In the case of different initial node distributions, ten independent optimisation experiments were carried out respectively. The average coverage of the five optimisation algorithms is shown in Table 6.

Table 6 The coverage performance of three algorithms over 10 independent optimisations

Algorithm	*Coverage rate (%)*
Chaos Particle Swarm Algorithm	84.1
Particle Sharing Particle Swarm Frog Leaping Hybrid Algorithm	85.4
The proposed algorithm	86.8

From Table 5, chaotic particle swarm optimisation algorithm combines the advantages of chaotic algorithm and particle swarm optimisation algorithm and its network coverage is high, reaching 84.1%. But chaotic particle swarm optimisation (PSO) adds chaos optimisation phase, which leads to a certain increase in computational complexity and time consuming. Particle sharing based particle swarm frog leaping hybrid optimisation algorithm combines the advantages of particle swarm optimisation algorithm and hybrid leapfrog algorithm to avoid making the particles fall into the local optimum prematurely and the network coverage is 1.3% higher than that of the former algorithm, but the algorithm is still in the local optimum. The algorithm proposed in this paper effectively avoids trapping into the local optimum and the convergence speed is fast and the network coverage rate is 86.8%.

6 Conclusions

In order to improve the performance of standard artificial bee colony algorithm, this paper proposes an improved multi-strategy artificial bee colony algorithm. Firstly, the original basic search mode is improved. The improved search strategies for individual, neighbourhood and current global optimal honey sources are proposed. The employed bees are randomly divided into three subgroups, corresponding to three evolutionary strategies. Since the three search strategies have different characteristics, it can ensure that the bee colony dynamically adjusts the search gravity centre in the evolution process. It can also balance the algorithm's capabilities of global search and local development. Secondly, by imitating the particle swarm algorithm, the abundant information contained in the current global optimal and random neighbour honey sources is fully utilised. The search strategy of the following bees is optimised, and the local search ability of the algorithm is enhanced. The experimental results show that the proposed algorithm has a better network coverage than the chaotic particle swarm optimisation algorithm and the particle sharing based particle swarm frog leaping hybrid optimisation algorithm. The next research content is the improvement of evolution strategy for bee colony.

Acknowledgement

This work is supported by Jiangxi Province Department of Education Science and Technology Project under Grant (Nos. GJJ161108, KJLD13096), the National Natural Science Foundation of China under Grant (No. 61663029).

References

Babar, B. and Crăciunescu, A. (2014) 'Comparison of artificial bee colony algorithm with other algorithms used for tracking of maximum power point of photovoltaic arrays', *International Conference on Renewable Energies and Power Quality (ICREPQ'14)*, Vol. 38, No. 12, pp.36–39.

Bilal, B. (2013) 'Implementation of artificial bee colony algorithm on maximum power point tracking for PV modules', *Advanced Topics in Electrical Engineering (ATEE), 2013 8th International Symposium*, pp.1–4.

Chu, S.C. and Tsai, T.W. (2007) 'Computation intelligence based on behavior of cats', *International Journal of Innovative Computing Information and Control*, Vol. 3, No. 1, pp.163–173.

Das, S., Biswas, S. and Kundu, S. (2013) 'Synergizing fitness learning with proximity-based food source selection in artificial bee colony algorithm for numerical optimization', *Applied Soft Computing*, Vol. 13, No. 12, pp.4676–4694.

Dorigo, M. and Caro, G.D. (1999) 'Ant colony optimization: a new meta-heuristic', *Evolutionary Computation, 1999. CEC99. Proceedings of the 1999 Congress on*, Vol. 2, pp.19–26.

Eberhart, R.C. and Kennedy, J. (1995) 'A new optimizer using particle swarm theory', *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp.39–43.

Gao, W.F. and Liu, S.Y. (2011) 'Improved artificial bee colony algorithm for global optimization', *Information Processing Letters*, Vol. 111, No. 17, pp.871–882.

Gao, W.F., Liu, S.Y. and Huang, L. (2012) 'A global best artificial bee colony algorithm for global optimization', *Journal of Computational and Applied Mathematics*, Vol. 236, No. 11, pp.2741–2753.

Gao, W.F., Liu, S.Y. and Huang, L.L. (2013) 'A novel artificial bee colony algorithm with Powell's method', *Soft Comput*, Vol. 13, No. 9, pp.3763–3775.

Halder, U., Das, S. and Maity, D. (2013) 'A cluster-based differential evolution algorithm with external archive for optimization in dynamic environments', *Cybernetics*, Vol. 43, No. 3, pp.881–897.

Horng, S.C. and Lin, S.Y. (2013) 'Evolutionary algorithm assisted by surrogate model in the framework of ordinal optimization and optimal computing budget allocation', *Information Sciences*, Vol. 233, No. 1, pp.214–229.

Kang, F. and Ma, Z. (2011) 'Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical function', *Information Sciences*, Vol. 181, No. 16, pp.3508–3531.

Karaboga, D. and Akay, B. (2009) 'A comparative study of artificial bee colony algorithm', *Applied Mathematics and Computation*, Vol. 214, No. 1, pp.108–132.

Karaboga, D., Gorkemli, B., Ozturk, C. and Karaboga, N. (2014) 'A comprehensive survey: artificial bee colony (ABC) algorithm and application', *Artificial Intelligence Review*, Vol. 42, No. 1, pp.21–57.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', *Neural Networks, 1995. Proceedings. IEEE International Conference on*, Vol. 4, pp.1942–1948.

Liu, W.T. and Fan, Z.Y. (2011) 'Chaotic particle swarm optimization based wireless sensor network coverage optimization', *Computer Applications*, Vol. 31, No. 2, pp.338–340.

Ma, M., Liang, J., Guo, M., Fan, Y. and Yin, Y. (2011) 'SAR image segment based on artificial bee colony algorithm', *Applied Soft Computing*, Vol. 11, No. 8, pp.5205–5214.

Metlicka, M. and Davendra, D. (2014) 'Chaos-driven discrete artificial bee colony', *Evolutionary Computation*, Vol. 6, No. 11, pp.2954–2947.

Pan, Q.K., Wang, L., Mao, K., Zhao, J.H. and Zhang, M. (2013) 'An effective artificial bee colony algorithm for a real-world hybrid flow shop problem in steelmaking process', *Automation Science and Engineering*, Vol. 10, No. 2, pp.307–322.

Passino, K.M. (2002) 'Biomimicry of bacterial foraging for distributed optimization and control', *Control Systems Magazine*, Vol. 22, No. 3, pp.52–67.

Ran, M.S. and Mesut, Z. (2013) 'A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems', *Applied Soft Computing*, Vol. 13, No. 4, pp.2188–2203.

Saxena, S., Sharma, K., Shiwani, K. and Sharma, H. (2014) 'Lbest artificial bee colony using structured swarm', *Advance Computing Conference (IACC)*, *2014 IEEE International*, Vol. 4, pp.1354–1360.

Shukran, M.A.M., Chung, Y.Y., Yeh, W.C. and Wahid, N. (2011) 'Artificial bee colony based data mining algorithm for classification tasks', *Modern Applied Science*, Vol. 5, No. 4, pp.217–231.

Sun, H. and Zhao, J. (2011) 'Application of particle sharing based particle swarm frog leaping hybrid optimization algorithm in wireless sensor network coverage optimization', *Journal of Information and Computational Science*, Vol. 8, No. 14, pp.3181–3188.

Wang, H., Wu, Z., Rahnamayan, S., Sun, H., Liu, Y. and Pan, J.S. (2014) 'Multi-strategy ensemble artificial bee colony algorithm', *Information Science*, Vol. 279, No. 20, pp.587–603.

Xiang, W.L. and An, M.Q. (2013) 'An efficient and robust artificial bee colony algorithm for numerical optimization', *Computers & Operations Research*, Vol. 40, No. 5, pp.1256–1265.

Yang, X.S. (2010) 'Firefly algorithm, stochastic test functions and design optimization', *International Journal of Bio-Inspired Computation*, Vol. 2, No. 2, pp.78–84.

Zhang, X., Zhang, X., Ho, S.L. and Fu, W.N. (2014) 'A modification of artificial bee colony algorithm applied to loudspeaker design problem', *IEEE Transactions on Magnetics*, Vol. 50, No. 2, pp.737–740.

Zhu, G. and Kwong, S. (2010) '*Gbest*-guided artificial bee colony algorithm for numerical function optimization', *Applied Mathematics and* Computation, Vol. 217, pp.3166–3173.