Journal of Information Hiding and Multimedia Signal Processing ©2016 ISSN 2073-4212
Ubiquitous International Volume **7**, Number **6**, November **2016**

Opposition-Based Artificial Bee Colony Using Different Learning Models

Jia Zhao[1,2], Xue-Feng Fu[1,2],Li Lv[1,2], Run-Xiu Wu[1,2],
Hui Wang[1,2], Xiang Yu[1,2], Tang-huai Fan[1,2]

1.Provincial Key Laboratory for Water Information Cooperative Sensing and Intelligent Processing, Nanchang Institute of Technology,Nanchang 330099,China;
2.School of Information Engineering,Nanchang Institute of Technology,Nanchang 330099, China
zhaojia925@163.com;fxfcn@126.com ;lvli623@163.com;
wrxhxl@qq.com; huiwang@whu.edu.cn;xyuac@ust.hk; fantanghuai@163.com

Received August, 2015; revised June, 2016

Abstract. *Artificial Bee Colony (ABC) is easy to fall into the local optimum area and has slow convergence velocity when solving some optimization problems. We propose a new method, called opposition-based ABC to improve the performance of ABC in this paper. In the process of evolution, according to the greedy selection strategy, the modified algorithm chooses the best solution as the new location of the employed bee and the onlooker bee from their present solution, new generated solution and opposition solution are used to enlarge the searching area; in addition, when the employed bee and the onlooker bee change into the scouter, the new approach proposes a new update rule, where the new location of the scouter is decided by the location of the employed bee to enhance local exploitation of the scouter. We introduce 4 common kinds of opposition-based models into our new approach and they form 4 kinds of opposition-based ABC algorithms. Then we use 8 famous benchmark functions to test the 4 different models; the results demonstrate that 4 kinds of opposition-based ABC have better performance than ABC, but there are differences on efficiency and precision among them.*

Keywords: Artificial Bee Colony, Opposition-based Learning, Model, Optimization

1. **Introduction.** Artificial bee colony (ABC), which is originally developed in 2005 by Karaboga [1], a Turkish scholar, is a swarm intelligence optimization algorithm based on bees honey gathering. Once proposed, ABC attracts many scholars attention, because ABC has many advantages such as a few parameters, easy implementation and strong global search optimum and it is applied into lots of science projects successfully [2, 3, 4, 5, 6, 7, 8].such as shop problems [9], optimal design and manufacturing [10], parameter estimation [11], economic dispatch [12, 13] and function optimization [14]. Although ABC has shown a good performance over many optimization problems, it converges slowly, especially at the middle and last stages of the search process. The main reason is that ABC is good at exploration but poor at exploitation [15].An ideal optimization algorithm should properly balance exploration and exploitation during the search process [16]. Initially, the algorithm should concentrate on exploration; as iteration increases, it would be better to exploit to find more accurate solutions. However, it is difficult to determine when the algorithm should switch from an explorative behavior to an exploitative behavior.

In order to balance exploration and exploitation of ABC during the searching process, this paper proposes opposition-based learning ABC (OLABC) algorithm. In the process, we introduce the opposition-based learning, and get best solution from three candidate solutions as the new location of the employed bee and the onlooker bee, and we adopt a new method of updating strategy to update the new location of the scouter which is decided by the employed bee which can enhance the learning ability and local exploitation of the scouter. We use 8 famous benchmark functions to test the 4 different models; the results demonstrate that 4 kinds of opposition-based ABC have better performance in algorithm performance than ABC, but there are differences on efficiency and precision among them in different optimization problems.

2. ABC algorithm and Opposition-based Learning Strategy.

2.1. ABC algorithm. There are three kinds of bees, the employed bee, the onlooker bee and the scouter. The number of the employed bee is equal to the number of the onlooker bee. After a round of global search, the employed bee return the area of information exchange to share information with other bees by swinging dance. The more abundant the nectar sourceis, the greater probability it is to be selected, the more the onlooker bee there are. Then the onlooker bee searches the areas like the employed bee. The employed bee and the onlooker bee choose the better nectar source position as the next generation solution according to the greedy rule.

Assume that the dimension of problem is D, the positions of nectar source correspond to the points of solution space, the ith(i=1, $2, \cdots, NP$) nectar sources quality is regarded as the fitness of solution, the number of solution(NP,i.e. the nectar sources number), is total of the number of employed bees and onlooker bees. $X_i = \{x_{i1}, x_{i2}, \cdots, x_{iD}\}$ represent the location of the ith nectar source, the location, the location X_i is updated randomly in the d dimensions as follows(d=1, $2, \cdots, D$).

$$x_{id} = L_d + rand(0,1) \times (U_d - L_d) \tag{1}$$

where x_{id} is the location of the ith bee in the dth dimension; U_d and L_d stand for the Lower and upper bounds of search space respectively.

At the beginning of the search, the employed bee generates a new nectar source around the nectar source according to the formula (2).

$$v_{id} = x_{id} + rand(-1,1) \times (x_{id} - x_{jd}) \tag{2}$$

where d is a random integer which denotes that the employed bee searches in random dimension; $j \in \{1, 2, \cdots, NP\}(j \neq i)$ represents that it chooses a nectar source which is different from the ith nectar source in the NP nectar source; rand(-1,1)is a random number with uniform distribution between 0 and 1.

When the fitness value of new nectar source $V_i = \{v_{i1}, v_{i2}, \cdots, v_{id}\}$ is better than X_i, the V_i replaces X_i , otherwise, the ABC algorithm keeps X_i, and update the employed bees according to the formula (2). After updating, the employed bee feed information back to the onlooker bee. The onlooker bee chooses the employed bee to follow in roulette way according to the probability p_i, and determined the retention of nectar according to the same greedy method of the employed bee. p_i is computed as follows.

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \tag{3}$$

where fit_i stands for the fitness value of the ith nectar source.

In the processing of search, if X_i does not find the better the nectar source after reaching the threshold *limit* and *trial* times iteration, X_i would be given up and the employed bee would be changed into the scout. The scout generates randomly a new nectar source which replaces X_i in the search space. The above-mentioned process can be described as follows.

$$v_i^{t+1} = \begin{cases} L_d + rand(-1,1) \times (U_d - L_d) & trial \geq limit \\ v_i^t & trial < limit \end{cases} \tag{4}$$

Generally, ABC algorithm takes the minimum optimization problem as an example; the fitness value of solution is estimated by formula (5).

$$fit_i = \begin{cases} 1/(1+f_i) & f_i \geq 0 \\ 1 + abs(f_i) & otherwise \end{cases} \tag{5}$$

where the f_i is the function value of solution.

2.2. **Opposition-based Learning Strategy.**

Opposition-Based Learning (OBL) is presented by Tizhoosh [17]in 2005, has been applied to Genetic algorithm [18], Differential Evolution [19], Ant Colony Optimization [20] and Shuffled Frog Leaping Algorithm [21] and so on. Whats more, OBL can generate opposite candidate solution, enlarge the search scope and enhance the performance of the algorithms. Reference [22] has a mathematical deduction for OBL, and in theory, it is proved that the opposition candidate solution can be close to the global optimal solution with a greater probability.

Definition 1(Opposite Point) [23] — Let $X_i = \{x_{i1}, x_{i2}, \cdots, x_{iD}\}$ be a point in a D-dimensional space(i.e., candidate solution), where $j \in [a_j, b_j]$, j=1, 2, $\cdots, D$, the opposite point $X_i^* = \{x_{i1}^*, x_{i2}^*, \cdots, x_{iD}^*\}$ is defined below.

$$x_{ij}^* = a_{ij} + b_{ij} - x_{ij} \tag{6}$$

Definition 2(Dynamic Generalized Opposition-Based Learning, GOBL) [24]—

$$X_{ij}^*(m) = k(a_j(t) + b_j(t)) - X_{ij}(t) \tag{7}$$

where $X_{ij}^*(t)$ is the opposite solution of $X_{ij}(t)$, N is the number of particle, D represents the dimension of problem, t is the iteration number, $a_j(t)$ and $b_j(t)$ stand form the dynamic boundary in the jth dimension which can be calculated by the following formula.

$$\begin{cases} a_j^{(t)} = max(x_{ij}^m) \\ b_j^{(t)} = max(x_{ij}^m) \end{cases} \tag{8}$$

We simply adjust the (8) as followings.

$$x^* = k(a + b - x) \tag{9}$$

where k is real number According to the values of k, the four kinds of GOB model can be expressed as follows. (1) k=0 (SS-GOB)

$$x^* = -x \tag{10}$$

where $x \in [a, b]$, $x^* \in [-b, -a]$.
(2) k=0.5 (DSI-GOB)

$$x^* = 0.5(a + b - x) \tag{11}$$

where $x \in [a, b]$, $x^* \in [-\frac{b-a}{2}, \frac{b-a}{2}]$.
(3) k=1 (D-GOB)

$$x^* = a + b - x \tag{12}$$

where $x \in [a, b]$, $x^* \in [a, -b]$.
(4) k=rand(0,1) (DR-GOB)

$$x^* = k(a + b - x) \tag{13}$$

where k is a random number between 0 and 1,$x \in [a, b]$, $x^* \in [k(a+b)-b, -k(a+b)-a]$, the center location of opposite solution is a random number between $-\frac{a+b}{2}$ and $\frac{a+b}{2}$.

3. ABC using opposition-based learning.

3.1. **The principle of algorithm.** In accordance with probability theory, each candidate solution has 50% probability away from the optimal solution compared with its opposite solution. Hence, we present a new approach called opposition-based ABC, and apply the above 4 opposition-based learning models to ABC. In each iteration, by using GOBL, the range of the current solution is far less than the initial variable range with the increase of the number of iterations, which make population fast close to the optimum. At the same time, in order to keep the advantage of the employed bee and the onlooker bee, when the scouter reach the threshold limit by the trial times iteration and cannot find better nectar source, the algorithm adopts a new updating rule to avoid the scout updating randomly. To increase the local exploration ability of our method, the scout is updated by formula (14)

$$x_{ik}^{t+1} = \begin{cases} rand(-1,1) * x_{jk}^{t} & ,trial \geq limit \\ x_{jk}^{t} & ,trial < limit \end{cases} \tag{14}$$

where k=$1, 2, \cdots, D$, x_{jk}^{t} is the location of the jth nectar source in kth dimension at the tth iteration, $j \in \{1, 2, \cdots, NP\}(j \neq i)$ represents that it chooses a nectar source which is different from the ith nectar source in the NP nectar source; the new generated nectar is decided by each dimension of the employed bee, some dimension of X_i may come from better location of the same nectar source.

3.2. **The steps of our method. Step 1:**Initialization of parameters and nectar source, including the number of employed bee and onlooker bee NP,$limit$, CR, $trial = 0$;

Step 2: Allocate the employed bee to NP nectar sources and update by the formula (2); When employed bee generate new nectar source V_i, computet he fitness value of V_i according to formula (5); then update the employed bee by (7) and generate the opposite solution, and its fitness value is required by (5); finally, choose the best fitness value according to the greedy choose strategy, if V_i or opposite value is kept, set $trial = 0$, otherwise, $trial$++.

Step 3: According to roulette way, estimate the probability by formula (3) and update according to the same greedy method of the employed bee.

Step 4: Judging $trial \geq limit$, if it is satisfiedjump to Step 5, otherwise, go to Step6;

Step 5: The scouter generates the new nectar source according to formula (3);

Step 6: If the algorithm satisfies the ending condition, output the optimum, otherwise, please turns to step 2 and continue.

4. Experiments and Results Analysis.

4.1. **Test functions.** 9 well-known test functions are shown in the table 1, $f_1 \sim f_4$ are unimodal functions, functions $f_5 \sim f_8$ are multimodal functions. Most optimization algorithms fall into local optimal solution in the process of searching the optimum.

TABLE 1. Test Functions

Function No	Function Name	Range	Optimum coordinate	Optimum
f_1	Sphere	[-100,100]	$(0,\cdots,0)$	0
f_2	Schwefel 2.22	[-10,10]	$(0,\cdots,0)$	0
f_3	Schwefel 2.21	[-100,100]	$(0,\cdots,0)$	0
f_4	Quadric Noise	[-1.28,1.28]	$(0,\cdots,0)$	0
f_5	Schwefel	[-500,500]	$(-420.9687,\cdots,420.9687)$	0
f_6	Rastrigin	[-5.12,5.12]	$(0,\cdots,0)$	0
f_7	Ackley	[-32,32]	$(0,\cdots,0)$	0
f_8	Griewank	[-600,600]	$(0,\cdots,0)$	0

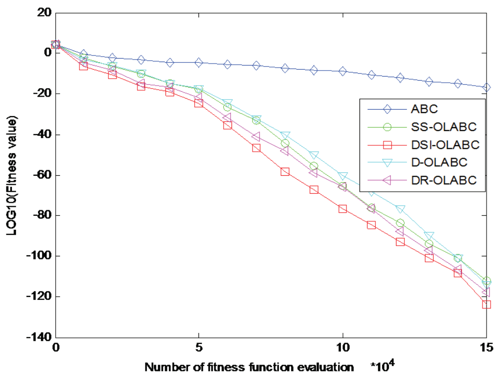
4.2. **the results and analysis.** 4 kinds of GOBL models called SS-GOB, DSI-GOB, D-GOB and DR-GOB are applied to ABC, so we can get 4 different ABC algorithms using opposition-based learning, the abbreviations of 4 ABC variants are respectively SS-OLABC, DSI-OLABC, D-OLABC and DR-OLABC. In order to judge which ABC variant is the best according to the experiments, we assume that the scale of population is equal to $N = 30$, the maximum number of iteration is 150000, the number of population is 200, the number of the employer bee and the onlooker bee is respectively 100, and the bee turns into the scouter when the iteration number is more than 100.

Table 2 presents results of 5 kinds of algorithms over 30 dimensions, where Mean stands for the mean optimal fitness value and Std.Dev represents the standard deviation. The Mean reflects the precision at the fixed times, and the Std.Dev reflects the stability. In order to eliminate the influence of the random of the algorithm, the algorithm runs 30 times independently, and the final average value is the final result of the algorithm.

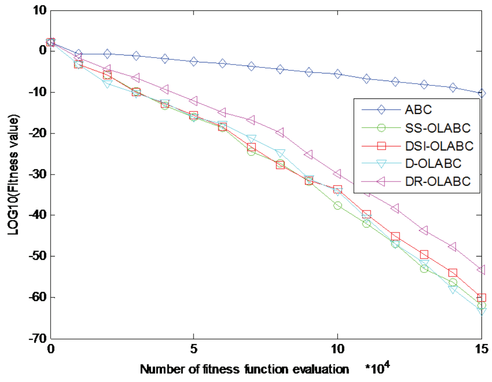
TABLE 2. The comparison results of 5 algorithms on famous benchmark functions

Function Name		ABC	SS-OLABC	DSI-OLABC	D-OLABC	DR-OLABC
f_1	Mean	1.05e-16	4.12E-113	**2.17e-124**	1.04e-114	7.74e-120
	std.Dev	8.93e-15	5.26e-1122	**2.57e-123**	5.24e-113	5.621e-120
f_2	Mean	1.30e-10	2.58e-51	1.10e-60	**4.16e-64**	6.59e-54
	std.Dev	3.15e-10	7.86e-50	5.78e-60	**8.47e-63**	1.87e-53
f_3	Mean	3.97e0	1.13e-24	**6.28e-34**	4.08e-29	7.62e-18
	std.Dev	2.93e0	5.17e-24	**2.17e-33**	7.48e-28	5.28e-17
f_4	Mean	2.05e1	**5.07e-96**	7.53e-51	2.26e-15	8.99e-52
	std.Dev	2.26e1	4.78E-96	2.57e-51	5.11e-15	7.85e-37
f_5	Mean	-12501.4	-12569.5	-12095.7	-12213.4	**-12569.5**
	std.Dev	2.75e1	2.55e-13	5.23e2	3.25e2	**1.97e-13**
f_6	Mean	2.64e-14	**0**	**0**	**0**	**0**
	std.Dev	2.75e-13	**0**	**0**	**0**	**0**
f_7	Mean	1.13e-11	**5.89e-16**	**5.89e-16**	**5.89e-16**	**5.89e-16**
	std.Dev	2.82e-11	**0**	**0**	**0**	**0**
f_8	Mean	3.53e-13	**0**	**0**	**0**	**0**
	std.Dev	7.24e-12	**0**	**0**	**0**	**0**

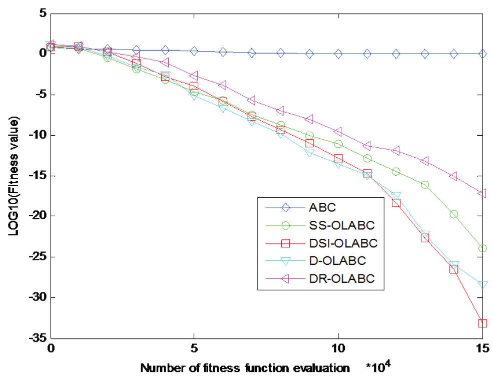
From the results of table 2, 4 different kinds of ABC using opposition-based learning outperforms standard ABC on 8 benchmark functions. 4 kinds of improved ABC attain



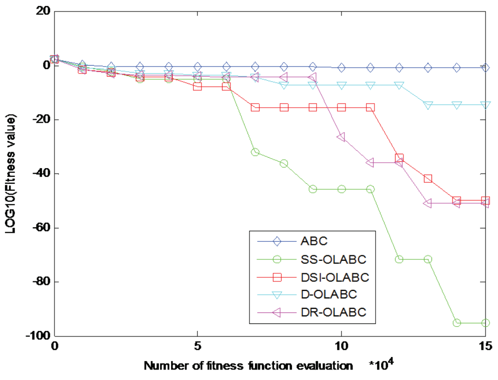
(a) f_1



(b) f_2



(c) f_3



(d) f_4

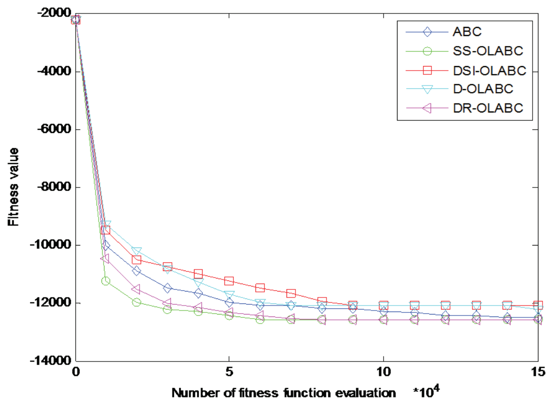
Figure 1. Evolution curves of unimodal functions

the same optimization effects on $f_6 \sim f_8$ functions, as for the rest 5 functions, they get different optimization results. For the comparison of SS-OLABC,DSI-OLABC,D-OLABC and DR-OLABC, DSI-OLABC performs better than the rest 3 ABC variants on f_1 function; and SS-OLABC has the best performance among 4 ABC variants. So we can know from the table 2 that 4 ABC variants have their own optimization problems and different optimal effects.

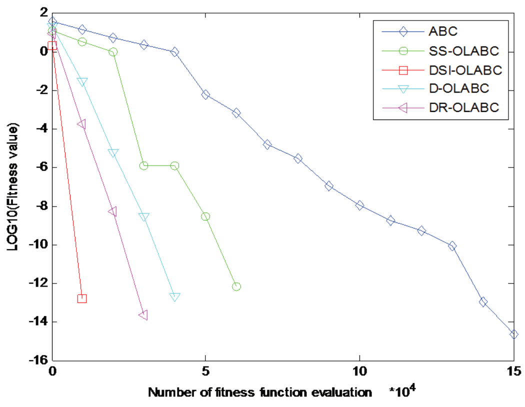
In order to compare the performance of 5 algorithms, Friedman test is employed to analyze results. Table 3 presents the average ranking of ABC, SS-OLABC, DSI-OLABC, D-OLABC and DR-OLABC on 8 test functions. The lower ranking is, the better performance and the higher rank will be. The best values are shown in bold. From the results in the Table 3, the DR-OLABC has the best performance among 5 algorithms.

Table 3. results of 5 algorithms using Friedman test

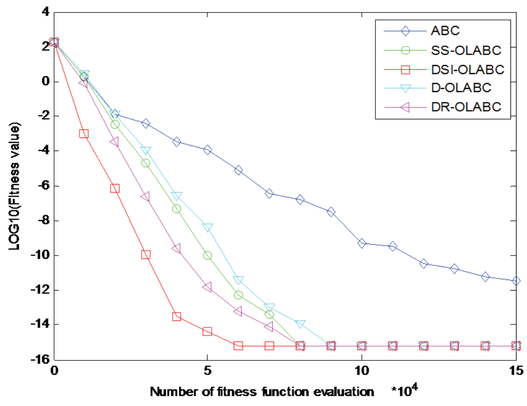
Algorithm	Ranking
DR-OLABC	**2.38**
SS-OLABC	2.50
D-OLABC	2.75
DSI-OLABC	2.88
ABC	4.50



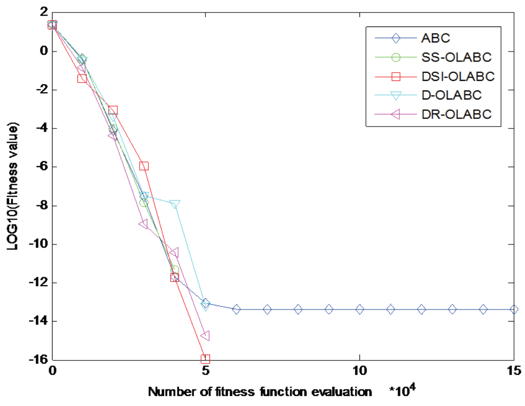
(a)f_5



(b) f_6



(c) f_7



(d) f_8

FIGURE 2. Evolution curves of multimodal functions

To illustrate the convergence velocity of 4 ABC variants in the evolution process, we give the convergence performance curve of SS-OLABC, DSI-OLABC, D-OLABC, DR-OLABC and standard ABC algorithm in 30 dimensions on 8 test functions. Abscissa means function evaluations, ordinate shows logarithm of fitness value.

As shown from fig.1 and fig.2, the standard ABC has trapped to local optimum on 4 unimodal functions when the number of iterations is about 10000 times, 4 ABC variants have not only faster convergence velocity than standard ABC, but also they are near the optimum. 4 ABC variants have faster convergence velocity than the standard ABC on multimodal functions such as f_6 , f_7 and f_8 , but the standard ABC converge faster than 4 ABC variants on function. Whether unimodal functions or multimode functions, the 4 variants convergence velocity is different.

5. **Conclusions.**

The paper proposed opposition-based ABC, our approach generates the opposite solution of the employed bee and the onlooker bee, choose the best solution from three candidate solutions as the new location of the above two kinds of bees according to the greedy selection strategy, and put forward a new updating rule of the scouter, the new location of the scouter is decided by the location of the employer bee using the rule. Then we use the dynamic generalized opposition-based learning into ABC, which form 4 ABC variants according to the value of k. Finally, From analysis and experiments based on well-known 8 benchmark functions, we find that SS-OLABC, DSI-OLABC,D-OLABC and DR-OLABC attain better precision and faster convergence velocity when compared with standard ABC in one sense. But the optimization effects of the 4 ABC variants are different on benchmark functions, DR-OLABC outperforms SS-OLABC, DSI-OLABC,D-OLABC and ABC according to the results of average rankings. So possible future work [25, 26] is how to choose different ABC using opposition-based learning models according to different optimization problems.

Acknowledgment. This work is supported by the Jiangxi Province Department of Education Science and Technology Project under Grant (Nos. GJJ151133, GJJ09612), the National Natural Science Foundation of China under Grant (Nos. 51669014, 61663029), Science Foundation of Jiangxi Province under Grant (Nos. 20161BAB212037), Science and Technology Support Program of Jiangxi Province under Grant (No.20142BBG70034), Science and Technology Planning Project of Nanchang City under Grant (No. 2014HZZC008), College students innovation and entrepreneurship training program(No. 201611319059).

REFERENCES

[1] D. Karaboga, An idea based on honey bee swarn for numerical optimization, *Kayseri: Erciyes University*, 2005.
[2] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, A comprehensive survey: artificial bee colony (ABC) algorithm and applications, *Artificial Intelligence*, vol. 42, pp. 21-57, 2014.
[3] P. W. Tsai, J.-S. Pan, B.Y. Liao, S.-C. Chu, Enhanced Artificial Bee Colony Optimization, *International Journal of Innovative Computing, Information and Control,* vol.5, no. 12(B), pp. 5081-5092, 2009.
[4] C. Li, H. Xiong, Improved Artificial Bee Colony Algorithm and its Application in Function Optimization, *Journal of Computational Information Systems*, vol. 10, no. 6, pp. 2341-2349, 2014.
[5] P. W. Tsai, M. K. Khan, J. S. Pan, B. Y. Liao, Interactive Artificial Bee Colony Supported Passive Continuous Authentication System, *IEEE Systems Journal,* vol. 8, no. 2, pp. 395-405, 2014.
[6] J.-S. Pan, H. B. Wang, H. Q. Zhao, L.-L. Tang, Interaction Artificial Bee Colony Based Load Balance Method in Cloud Computing, *The 8th International Conference on Genetic and Evolutionary Computing*, pp. 49-57, 2014.
[7] J. Zhao, L.Li, H. Sun, Artificial Bee Colony using Opposition-Based Learning, *The 8th International Conference on Genetic and Evolutionary Computing*, pp. 3-10, 2014.
[8] G. Jian, L. Li, J. Zhao, H. H. Xu, Y. Hu, P. Fu, Artificial bee colony algorithm based on special central and random dimension learning, *Journal of Nanchang Institute of Technology*, vol. 34, no.4, pp. 52-56, 2015.
[9] C. Zhe, X. S. Gu, An improved discrete artificial bee colony algorithm to minimize the makespan on hybrid flow shop problems, *Neurocomputing*, vol. 148, pp. 248-259, 2014.
[10] A. R. Yildiz, A new hybrid artificial bee colony algorithm for robust optimal design and manufacturing, *Applied Soft Computing*, vol. 13, no. 5, pp. 2906-2912, 2013.
[11] S. Talatahari, H. Mohaggeg, Kh. Najafi, A. Manafzadeh, Solving Parameter Identification of Nonlinear Problems by Artificial Bee Colony Algorithm, *Mathematical Problems in Engineering,* vol. 2014, pp. 1-7, 2014.
[12] Do. Aydin, S. C. Yaar, T. Liao, Artificial bee colony algorithm with dynamic population size to combined economic and emission dispatch problem, *International Journal of Electrical Power & Energy Systems*, vol. 54, pp. 144-153, 2014.
[13] H. Shayeghi, A. Ghasemi, A modified artificial bee colony based on chaos theory for solving non-convex emission/economic dispatch, *Energy Conversion and Management*, vol. 79, pp. 344-354, 2014.
[14] H. Wang, Z. J. Wu, S. Rahnamayan, H. Sun, Y. Liu, J.-S. Pan, Multi-strategy ensemble artificial bee colony algorithm, *Information Sciences*, vol. 279, no. 20, pp. 587-603, 2014.
[15] G. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166-3173, 2010.
[16] A. P. Engelbrecht, Heterogeneous particle swarm optimization, *The 7th International Conference on Swarm Intelligence*, pp. 191-202, 2010.
[17] H. R. Tizhoosh , Opposition-based Learning: a new scheme for machine intelligence, *International Conference on, Computational Intelligence for Modeling, Control and Automation*, pp. 695-701, 2005.
[18] H. R. Tizhoosh , Reinforcement Learning Based on Actions and Opposite Actions, *International Conference on Artificial Intelligence and Machine Learning*, pp. 19-21, 2005.

[19] Q. Z. Xu,L. Wang, B. M. He, N. Wang, Opposition-Based Differential Evolution Using the Current Optimum for Function Optimization, *Journal of Applied Sciences*, vol. 29, no. 3, pp. 308-315, 2011.
[20] H. P. Ma, X. Y. Ruan, B. Jin, Oppositional ant colony optimization algorithm and its application to fault monitoring, *Chinese Control Conference*, pp. 3895-3898, 2010.
[21] L. Juan, Y. W. Zong, S. L. Ma, Improved opposition-based shuffled frog leaping algorithm for function optimization problems, *Application Research of Computers*, vol. 30, no. 3, pp. 760-763, 2013.
[22] R. Shahryar,R. T. Hamid, M. A. S. Magdy , Opposition versus randomness in soft computing techniques, *Applied Soft Computing*, vol. 8, no. 2, pp. 906-918, 2008
[23] S. Rahnamayan, H. R. Tizhoosh, M. Salama, Opposition-based differential evolution, *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64-79, 2008.
[24] S. W. Wang , L. X. Ding, D. T. Xie, Group Search Optimizer Applying Opposition-based Learning, *Computer Science,* vol. 9, pp. 183-187, 2012.
[25] L. H. Xu, X. F. Li, Simon X.Yang, Intelligent Information Processing and System Optimization[J]. *Intelligent Automation and Soft Computing*, vol. 17, no. 7, pp. 829-833, 2011.
[26] L. Z. Xu, X. L. Feng, C. L. Wen, Sequential Fusion Filtering for Networked Multi-Sensor Systems Based on Noise Estimation[J]. *ACTA ELECTRONICA SINIC*, vol. 42, no. 1, pp. 160-168, 2014.