Intelligent Automation And Soft Computing, 2019
Copyright © 2019, TSI® Press
Vol. 25, no. 2, 385–394
https://doi.org/10.31209/2019.100000100

An Enhanced Exploitation Artificial Bee Colony Algorithm in Automatic Functional Approximations

Peizhong Liu[1], Xiaofang Liu[1], Yanming Luo[2], Yongzhao Du[1], Yulin Fan[1] and Hsuan-Ming Feng[3]

[1]College of Engineering, Huaqiao University, Quanzhou 362021-China
[2]College of Computer Science and Technology, Huaqiao University, Xiamen 361021–China
[3]Department of Computer Science and Information Engineering, National Quemoy University, No.1 University Rd, Kin-Ning Vallage Kinmen, 892 Taiwan, R.O.C.

ABSTRACT
Aiming at the drawback of artificial bee colony algorithm (ABC) with slow convergence speed and weak exploitation capacity, an enhanced exploitation artificial bee colony algorithm is proposed, EeABC for short. Firstly, a generalized opposition-based learning strategy (GOBL) is employed when initial population is produced for obtaining an evenly distributed population. Subsequently, inspired by the differential evolution (DE), two new search equations are proposed, where the one is guided by the best individuals in the next generation to strengthen exploitation and the other is to avoid premature convergence. Meanwhile, the distinction between the employed bee and the onlooker bee is not made, unified as a bee and controlled by the probability P. The performance of proposed approach was examined on 14 benchmark functions, and results are compared with basic ABC and other ABC variants. As documented in the experimental results, the proposed algorithm has good optimization performance and can improve both the accuracy and the convergence speed.

KEY WORDS: Artificial bee colony algorithm, Bionic algorithms, search equation, optimization algorithm

1 INTRODUCTION

THE optimization problem is often met in the financial, economic, management, computer and other fields to seek the best solution in a finite or infinite feasible scheme. As a vital branch of applied mathematics and operational research, it has aroused wide concern and has been deeply infiltrated almost everywhere.

The traditional optimization methods include the simplex method (Dantzig, 1951), the steepest descent method (Chatterjee, 2013), etc. These approaches have shown excellent performance in solving some mathematical models, but the actual activities optimization model is established gradually to large-scale, multi-dimensional problem. Moreover, the traditional algorithms mostly depend on the initial point of objective function, continuity and differentiability of functions. Consequently, these algorithms are often powerless for the large-scale complex optimization problems without explicit mathematical expression, which prompts people to quest for new algorithms. Later on, the biological behavior is abstracted into the mathematical model, and intelligent optimization algorithm is invented to the specific optimization problem, which shows strong vitality and adaptability. Such algorithms have no above requirements, and these advantages have attracted great attention.

The intelligent optimization algorithms mainly include genetic algorithm (GA) (Lin, et al., 2017), ant colony optimization algorithm (ACO) (Zhou, 2009), particle swarm optimization algorithm (PSO) (Tian, 2017), differential evolution algorithm (DE) (Mayer, et al., 2005), etc. The artificial bee colony algorithm (ABC) (You, et al., 2017) is also such an algorithm, proposed by Karaboga in 2005 and especially outstanding in intelligent optimization field. The ABC has been widely concerned because of its simplicity

CONTACT Hsuan-Ming Feng hmfeng@nqu.edu.tw, hmfenghmfeng@gmail.com

and easy-implementation. It has been widely used in unconstrained numerical optimization, artificial neural network training, image segmentation, etc. Hence, study on algorithm improvement and theory analysis should be further work.

However, ABC also has deficiencies such as slow convergence speed and easy premature (Gao, et al., 2015). The reasons are as follows. The initial solution affects the quality of final solution to a certain extent. The more uniform the initial solution is, the wider the coverage is, and thus searching the neighborhood of the optimal solution is more likely. Yet, the basic ABC adopts random method which has blindness and is not conducive to find the optimal solution. In addition, we all know that exploration and exploitation for swarm intelligent optimization algorithm is indispensable but contradictory, with great influence on the optimization effect. When the two capabilities are in a suitable balance, the optimization effect can be the best. But the search equation of basic ABC is well exploratory, and the exploitation is poor. Subsequently, a generalized opposition-based learning strategy (GOBL) and two new search equations are applied to improve performance. That is the main contributions of this paper.

The rest of this paper is organized as follows. Section 2 depicts basic ABC and summarizes the related works. The modified ABC called EeABC is proposed and analyzed in Section 3. Section 4 discusses the experimental results. Finally, the conclusion is drawn in Section 5.

2 BASIC ABC ALGORITHM AND RELATED WORKS

2.1 Description of Basic ABC

THE artificial bee colony algorithm (Basturk, and Karaboga, 2006) divides the bees into employed bees, onlooker bees and scouter bees, and their roles in the optimization process are distinct.

Employed bees: correspond to the honey source, record information about nectar, and share information with other bees through swing dance. The nectar position is obtained by the following formula,

$$\mathbf{V}_{ij} = \mathbf{x}_{ij} + rand(\mathbf{x}_{ij} - \mathbf{x}_{kj}) \tag{1}$$

where $\mathbf{V}_{ij}$ is the location of new nectar, $\mathbf{x}_{ij}$ and $\mathbf{x}_{kj}$ are i_{th} and k_{th} nectar's j_{th} position, *rand* is random number subjected to [-1,1].

Onlooker bees: share the honey source information brought by employed bees. Choose a better nectar and search new source in the vicinity by Eq.(1).

Scouter bees: explore new nectar. If a nectar for a continuous generation is not updated, it will start scouter bees, randomly generated new honey instead of the original source.

The ABC algorithm searches for optimal solution through repeated search and conversion of the three bees.

Due to space limitations, a detailed description of the ABC is given by reference (Basturk and Karaboga, 2006) (Karaboga, 2005) (Karaboga and Basturk, 2007) (Karaboga and Basturk, 2008) (Karaboga and Akay, 2009).

2.2 Related Works about ABC

ABC is presented by Karaboga, a Turkish scholar in his technical report in 2005 (Karaboga, 2005). In 2006, Basturk and Karaboga(2006) first introduced ABC algorithm at the International Conference. In 2007, the research was published in academic journal for the first time, which described ABC algorithm, and compares with other well-known intelligent algorithms (Karaboga and Basturk, 2007). In 2008, Karaboga and Basturk(2008) were studied on optimization performance of ABC algorithm in detail, and then illustrated the performance]. In 2009, the ABC algorithm website (http://mf.erciyes.edu.tr/abc) was built to provide information for the researchers (Karaboga and Akay,2009). Since ABC was proposed, numerous scholars conducted researches. Zhu and Kwong(2010) proposed a gbest-guided ABC algorithm(denoted as GABC) which introduced current global optimal solution information into the search equation. Gao, Liu and Huang (2010) presented a modified ABC algorithm(denoted as ABC/best) inspired by DE algorithm, where novel search equation, chaotic systems and opposition-based learning method were introduced to enhance the global convergence. Gao and Liu(2012) improved the algorithm's exploitation capability by searching around current best solution. Kiran et al(2015) proposed the integration of multiple solution update rules with ABC, which adopted five search strategies to efficiently solve different types of optimization problems. Shan. Yasuda and Ohkura (2015) proposed a self-adaptive hybrid enhanced ABC algorithm to improve the convergence ability, search speed and control the balance between exploration and exploitation. Cui et al. (2016) introduced two novel search equations and a depth-first search (DFS) framework which was to allocate more computing resources for nectar and obtain better quality solution. Zhang et al. (2017) developed a distributed dynamic ABC based on fuzzy C-means clustering. In addition, a search equation based on the Gaussian attractor was proposed to further accelerate the diffusion of optimal solution.

The recent research on ABCs discussed above cannot be covered in this section. More recent studies can be found in related literature.

3 PROPOSED APPROACH

3.1 The Generalized Opposition-Based Learning Strategy (GOBL)

THE uniformity of initial population distribution directly affects the convergence speed and the solution quality of the algorithm. Therefore, it plays a pivotal role to design a reasonable initialization method for improving optimization performance. In the initial phase, it is blind to the solution spatial distribution information, which requires the initial population is to be evenly distributed in the solution space so that the algorithm can uniformly search. Generally, the initial population is randomly generated, so it cannot guarantee the uniformity of population distribution. Consequently, this paper presents the generalized opposition-based learning strategy (GOBL), which can simultaneously generate a solution and the corresponding inverse solution and then ensure that the initial population is evenly distributed in the search space(Zhou, et al., 2015)(Wang, et al., 2011). The details are described as follows.

Let $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ be a feasible solution of the current optimization problem, and its corresponding inverse solution $OX_i = (ox_{i,1}, ox_{i,2}, \dots, ox_{i,D})$ can be defined as:

$$\mathbf{ox}_{i,j} = k\left(\mathbf{a}_j + \mathbf{b}_j\right) - \mathbf{x}_{i,j} \tag{2}$$

$$\mathbf{a}_j = \min\left(\mathbf{x}_{i,j}\right), \mathbf{b}_j = \max\left(\mathbf{x}_{i,j}\right) \tag{3}$$

$$i = 1, 2, \dots, SN, \quad j = 1, 2, \dots, D$$

where $\mathbf{x}_{i,j} \in \left[\mathbf{x}_{min,j}, \mathbf{x}_{max,j}\right]$, $k \in [0,1]$ is the generalized coefficient, $\left[\mathbf{a}_j, \mathbf{b}_j\right]$ is the dynamic boundary of the j^{th} dimension search space.

If the opposite solution is out of dynamic bounds, it is reset by randomly method:

$$\mathbf{ox}_{i,j} = \mathrm{rand}\left(\mathbf{x}_{min,j}, \mathbf{x}_{max,j}\right) \tag{4}$$

where $\mathrm{rand}(\bullet)$ is random number between $\mathbf{x}_{min,j}$ and $\mathbf{x}_{max,j}$.

The pseudo-code description of GOBL is shown in Table 1.

Table 1. The pseudo-code description of GOBL

```
SN: Number of Foods
D: Dimensionality of problem
        // Initialization
        for i = 1 to SN do
                for j = 1 to D do
                Randomly generated solutions X_ij by X_ij = x_min,j + rand ( x_max,j - x_min,j ) ;
                end
        end
Set the individual counter i = 1, j = 1;
        for i = 1 to SN do
                for j = 1 to D do
                Generate opposite solutions OX_ij by Eq. (2);
                end
        end
Selecting SN best individuals from the set {X_ij ∪ OX_ij} as initial population.
```

By this strategy, a feasible solution to be optimized is calculated and its opposite solution is evaluated, and then the better solution is chosen as the candidate solution. The method can improve the probability of finding the global optimal solution. What's more, the literature (Rahnamayan, et al, 2008) has mathematically proved that the GOBL strategy is a good method to estimate the original candidate solution.

3.2 Novel Search Mechanism

The imbalance of search ability leads to decline the algorithm's performance, so the trade-off of exploitation and exploration is in urgent need. In formula 1, it is clear that *rand* is a coefficient randomly obtained in between [-1,1], and the parameters *j* and *k* are random numbers in [1, *D*], the random factors causes lack exploitation. In a word, the basic ABC does well in exploration, but badly in exploitation. To balance the exploitation and exploration, two new search equations are designed based on the DE algorithm(Price, 2005). In the process of algorithm implementation, the distinction between the employed bee and the onlooker bee is not made, unified as a bee. Meanwhile, the probability *P* is used to control the above two equations and the parameter is determined by the benchmark functions(Gao and Liu, 2012). New search strategies are given as follows:

$$\mathbf{v}_i^j = \mathbf{x}_{best}^j + \phi_i^j\left(\mathbf{x}_{best}^j - \mathbf{x}_{r1}^j\right) \tag{5}$$

$$\mathbf{v}_i^j = \mathbf{x}_i^j + \phi_i^j\left(\mathbf{x}_i^j - \mathbf{x}_{r2}^j\right) \tag{6}$$

where the index r_1 and r_2 are random integer which belongs to {1, 2, ... , *SN*}, and varies with *i*. The implication of *i* and *j* is as in the above case of Eq.(1). The coefficient φ_i^j is chosen from the range of [-1,1]. The variable x_r^j is the j^{th} dimension of r^{th} particle. The variable x_{best}^j refers to the j^{th} dimension of best particle, guided the next iteration individual's evolution direction, which can improve exploitation.

3.3 Main Steps of EeABC

Based on the above explanation of improvement strategy, the pseudo-code of the EeABC algorithm is given as Table 2:

Table 2. The pseudo-code of the EeABC algorithm

```
Data: set control parameters and concepts
SN: Number of Foods
D: Dimensionality of problem
limit: Maximum numbers of trial for abandoning a nectar
MCN: Maximum numbers of cycle
MFE: Maximum number of fitness evaluations, where MFE = D *MCN
Begin
        // Initialization
        FES = 0;
        Generate initial population by Algorithm 1;
        trial(i) = 0;
        FES = FES +SN ;
      i = 1;
        Repeat
                While i < SN do
                        Generate a new solution X'(i) by Eq. (5);
                        Evaluate new solution fit(X'(i));
                        FES ++;
                        if fit(X'(i)) < fit(X(i)) then
                                X(i) = X'(i);
                          trial(i) = 0;
                                if FES == MFE
                                        Record the best solution achieved so far and exit main repeat;
                                end
                        else then
                                if rand < p then
                                        Generate a new solution X'(i) by Eq. (6);
                                        Evaluate new solution fit(X'(i));
                                        FES ++;
                                        if fit(X'(i)) < fit(X(i)) then
                                                X(i) = X'(i);
                                       trial(i) = 0;
            else
                          trial(i) ++;
                                        end
                                end
                        end
                        if FES == MFE
                                Record the best solution achieved so far and exit main repeat;
                        end
                end
        // Scouter bee phase
      if max(trial (i) ) > limit then
        Replace X(i) with a new randomly produced solution by
        X(i)=Xmin + rand ( Xmax - Xmin );
      end
        until FES = MFE;
end
```

Table 3. Benchmark Function

Number	Name	C	Search Range
F1	Sphere	U	$[-100, 100]^D$
F2	Elliptic	U	$[-100, 100]^D$
F3	SumSquares	U	$[-10, 10]^D$
F4	SumPower	M	$[-10, 10]^D$
F5	Schwefel2.22	U	$[-10, 10]^D$
F6	Schwefel2.21	U	$[-100, 100]^D$
F7	Quartic	U	$[-0.5, 0.5]^D$
F8	QuarticWN	U	$[-1.28, 1.28]^D$
F9	Ackley	M	$[-32, 32]^D$
F10	Penalized1	M	$[-50, 50]^D$
F11	Penalized2	M	$[-50, 50]^D$
F12	Alpine	M	$[-10, 10]^D$
F13	Levy	M	$[-10, 10]^D$
F14	Shifted sphere	U	$[-100, 100]^D$

4 EXPERIMENTAL RESULTS

4.1 Benchmark functions

IN this paper, 14 benchmark functions with dimensions D=30 are selected to validate the performance of proposed algorithm (EeABC), as listed in Table 1. These functions are divided into two categories: unimodality (U), multimodality (M), where the function characteristic are given in column C of Table 3(Kiran, 2015).

4.2 Experimental Comparison with Basic ABC

To validate the performance of proposed algorithm (EeABC), we compare the experimental results of EeABC with that of ABC. When experiments are to be made, the population size *SN*=40, *limit*=100, the maximum number of fitness evaluations *MFE*=5000*D*. Through the simulation experiment, it is found that the better experimental results are obtained when *p*=0.7. Accordingly, *p* is taken as 0.7. The two algorithms run 30 times on each function independently, recording the mean and standard deviations of the results.

Table 4 presents the comparison results between the ABC and the EeABC with *D*=30. It can be seen: For the unimodal function, the two algorithms can obtain the theoretical optimal value of F1 and F14, and for other unimodal functions, the EeABC is superior to the ABC in the accuracy and stability. For the multimodal function, the EeABC is as efficient as ABC algorithm about F9, and the former outperforms the latter in both the accuracy and stability for other complex multimodal function.

Besides the solution quality and stability, the convergence curves are another essential measure of the performance. The convergence curves for some benchmark functions are shown in Figure 1. It can be seen that the convergence curve of EeABC is faster and can converge to a higher precision solution.

4.3 Experimental Comparison with ABC Variants

The experimental results of EeABC are compared with gbest-guided ABC(GABC)(Zhu and Kwong, 2010), ABC/Best/1 (Gao, et al., 2012), ABC/Best/2 (Gao, et al., 2012), and modified ABC (MABC)(Gao and Liu, 2012). These ABC variants are chosen for comparison because all the above mentioned algorithms are improved about the search equation. In the GABC algorithm, global best solution is adopted to update individuals of employed bee and onlooker bee phase. The ABC/Best/1 and ABC/Best/2 algorithms utilize various update strategy to enhance the optimization effect. Through analysis to the above algorithms, two of these strategies are picked to generate candidate solutions, combining with generalized opposition-based learning strategy (GOBL) to initial population. The comparison results illustrate that the proposed algorithm is better than the compared algorithms with regard to solution quality.

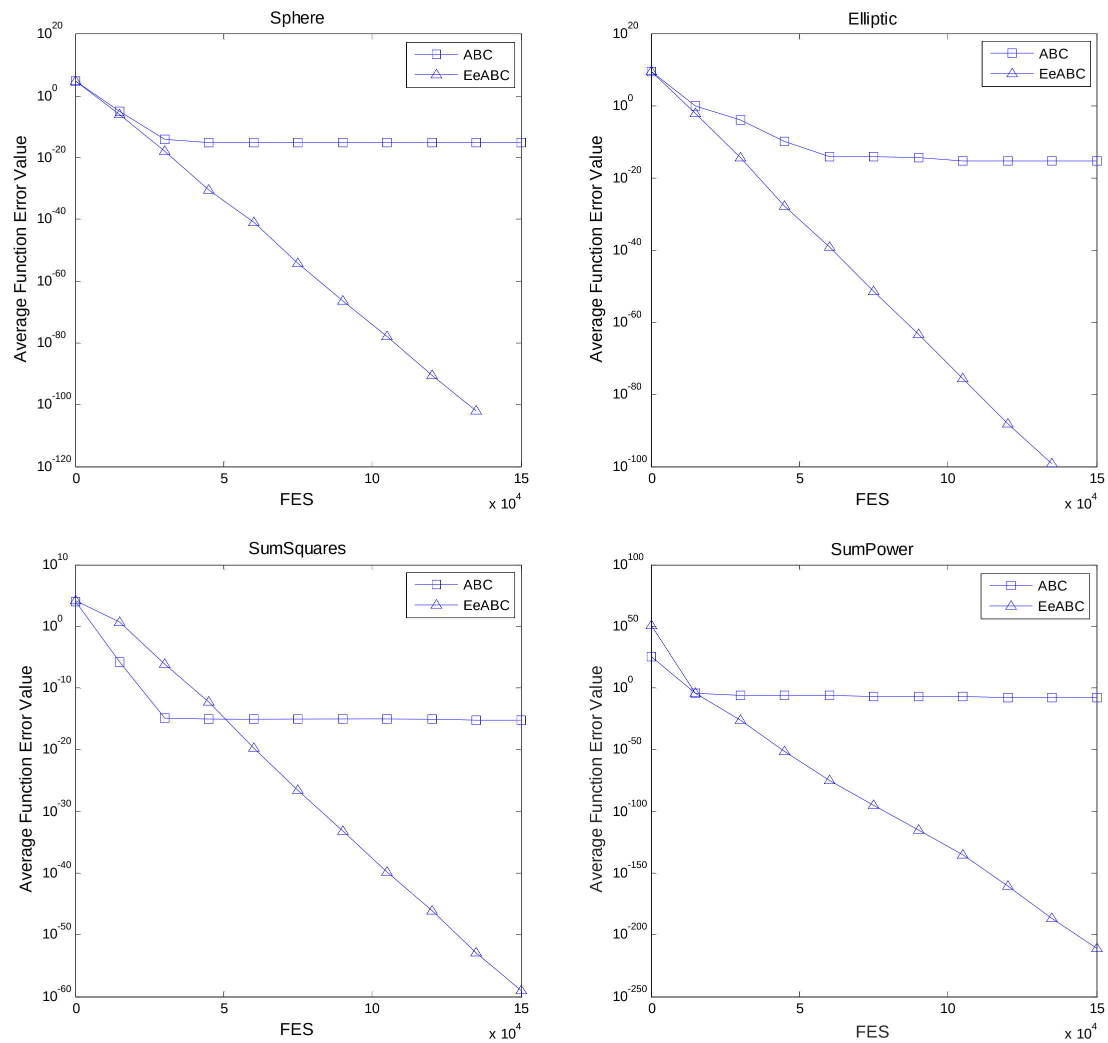
To make a clear and fair comparison, the setting parameters are in keeping with those of their corresponding papers, and the termination condition is to meet the maximum number of fitness evaluations (*MFE*), setting to 5000*D*. For EeABC algorithm, the population size *SN*=40, *limit*=100. In the comparison tables, the mean and the standard deviation of the algorithms are given, and the contrast effect is recorded as "+/=/-" which means that the performance is better than, equal to, and worse than the corresponding ABCs, respectively. For the other four contrast algorithms, the experimental data are taken directly from the literature (Kiran, et. al., 2015).

Table 5 presents the comparison results between the ABC variants and the EeABC with *D*=30. For F1-F5, F7, the proposed algorithm is superior to all other comparison algorithms in accuracy and stability. Especially, the EeABC can achieve the theoretical optimal value on F1. For F6, the proposed algorithm is ahead of ABC/Best/1, ABC/Best/2 and MABC, and has the same effect with GABC. For F8, F9 and F14, the EeABC is as efficient as the comparison algorithm. Especially, the proposed algorithm can

Table 4. Experimental Results(EeABC & ABC)

		F1	F2	F3	F4	F5	F6	F7
ABC	Mean	5.10E-16	4.79E-16	5.06E-16	2.85E-17	1.28E-15	7.27E-01	2.01E-16
	SD	8.40E-17	9.88-E17	9.20E-17	9.69E-18	1.44E-16	3.25E-01	4.74E-17
	Sig.	+	+	+	+	+	+	+
EeABC	Mean	0	7.86E-102	1.84E-106	6.49E-223	1.43e-54	4.46E-02	2.40E-212
	SD	0	1.88E-101	4.07E-106	0	2.11e-54	1.74E-02	0

		F8	F9	F10	F11	F12	F13	F14
ABC	Mean	4.86E-02	3.79E-14	5.08E-16	4.88E-16	8.82E-10	4.21E-16	4.91E-16
	SD	1.49E-02	3.99E-15	5.15E-17	7.45E-17	2.19E-09	8.31E-17	7.25E-17
	Sig.	+	=	+	+	+	+	+
EeABC	Mean	1.62E-02	3.29E-14	1.57E-32	1.35E-32	9.83E-16	1.18E-31	0
	SD	3.50E-03	1.15E-14	5.74E-48	5.47E-48	1.17E-15	6.56E-47	0



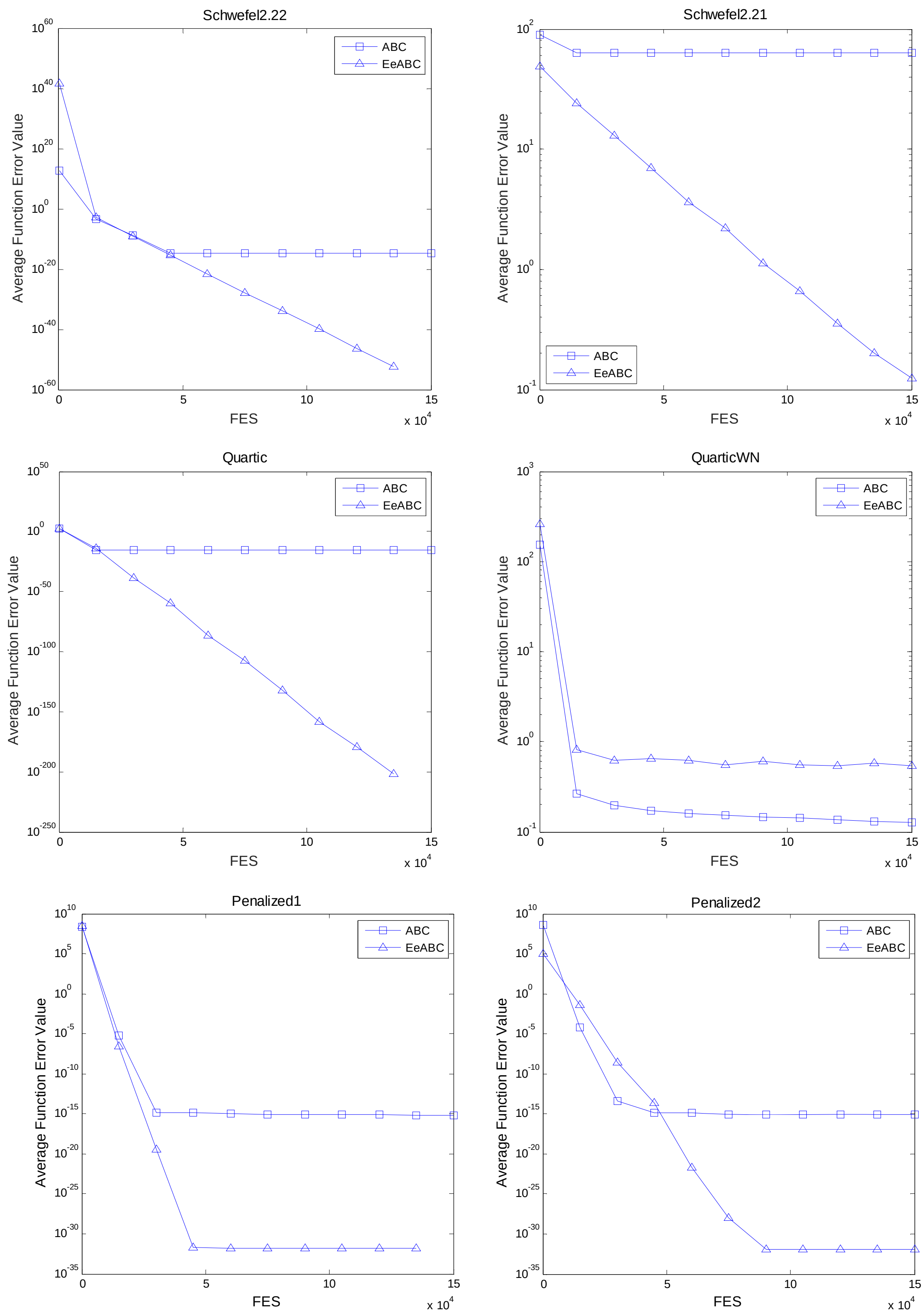


Figure 1. **Partial Convergence Curve**

Table 5. **Experimental Results for EeABC & ABC**

Func	GABC			ABCBest1			ABCBest2			MABC			EeABC	
	Mean	SD	Sig.	Mean	SD	Sig.	Mean	SD	Sig.	Mean	SD	Sig.	Mean	SD
F1	4.62E-16	7.12E-17	+	3.11E-47	3.44E-47	+	5.96E-35	3.61E-35	+	9.43E-32	6.67E-32	+	0	0
F2	3.62E-16	7.88E-17	+	5.35E-44	4.91E-44	+	1.70E-28	2.35E-28	+	3.66E-28	5.96E-28	+	7.86E-102	1.88E-101
F3	4.55E-16	7.00E-17	+	6.50E-48	6.04E-48	+	5.55E-36	3.36E-36	+	2.10E-32	1.56E-32	+	1.84E-106	4.07E-106
F4	1.64E-17	8.07E-18	+	1.77E-86	7.02E-86	+	3.00E-46	1.07E-45	+	2.70E69	5.38E-69	+	6.49E-223	0
F5	1.35E-15	1.36E-16	+	2.10E-25	9.08E-26	+	1.36E-18	4.27E-19	+	2.40E-17	9.02E-18	+	1.43E-54	2.11E-54
F6	2.18E-01	4.01E-02	=	2.18E+00	3.27E-01	+	3.55E+00	4.79E-01	+	1.02E+01	1.49E+00	+	4.46E-02	1.74E-02
F7	1.21E-16	3.99E-17	+	2.63E-97	3.75E-97	+	3.10E-76	2.89E-76	+	1.45E-67	2.28E-67	+	2.40E-212	0
F8	2.03E-02	5.74E-03	=	2.06E-02	4.75E-03	=	2.53E-02	4.67E-03	=	3.71E-02	8.53E-03	=	1.62E-02	3.50E-03
F9	3.20E-14	3.36E-15	=	3.01E-14	2.91E-15	=	3.07E-14	3.43E-15	=	4.13E-14	2.17E-15	=	3.29E-14	1.15E-14
F10	4.12E-16	8.36E-17	+	1.57E-32	5.57E-48	=	1.57E-32	5.57E-48	=	1.90E-32	3.70E-33	=	1.57E-32	5.74E-48
F11	4.01E-16	8.19E-17	+	1.35E-32	5.57E-48	=	1.35E-32	5.57E-48	=	2.23E-31	1.46E-31	=	1.35E-32	5.47E-48
F12	3.41E-09	1.13E-08	+	3.00E-16	8.99E-16	+	3.23E-14	9.14E-14	+	1.58E-16	2.48E-16	=	9.83E-16	1.17E-15
F13	3.28E-16	5.03E-17	+	1.35E-31	6.68E-47	=	1.35E-31	6.68E-47	=	1.48E-31	2.30E-32	=	1.18E-31	6.56E-47
F14	4.38E-16	8.43E-17	=	0	0	=	0	0	=	0	0	=	0	0

achieve the theoretical optimal value on F14. For other functions, the performance of EeABC is suboptimal or equivalent to other algorithms.

On the basis of the above experimental results, the EeABC can be a very promising algorithm. And the experiments on D=60 are not listed as a result of space issues.

5 CONCLUSION

TO sort out the issue of artificial bee colony algorithms, such as slow convergence speed and weak exploitation capacity, an enhanced exploitation artificial bee colony algorithm is proposed, called EeABC. The GOBL strategy is applied to obtain uniform initial population, and modified solution search equations are introduced to achieve a relative balance between exploitation and exploration. In addition, the performance of proposed approach was examined on 14 benchmark functions, and results are compared with basic ABC and other ABCs. As documented in the experimental results, the proposed algorithm has good optimization performance. As a consequence, EeABC may be a promising and viable tool to deal with numerical optimization problems. It is advisable to further adopt EeABC to deal with real-world problems. The studies on how to extend EeABC to handle classification of textile defects and to solve multi-objective optimization problems are our ongoing projects.

6 REFERENCES

B. Basturk, D. Karaboga, (2006). An artificial bee colony (ABC) algorithm for numeric function optimization, *IEEE swarm intelligence symposium 2006*.

B. Chatterjee, (2013). *Steepest Descent Method,* Springer US.

L. Cui, G. Li, Q. Lin, Z. Du, W. Gao, J. Chen, and N. Lu, (2016). A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation, *Information Sciences.* 367-368, 1012-1044.

G. B. Dantzig, (1951). Application of the simplex method to a transportation problem, *Activity Analysis and Production and Allocation*. 359-373.

W. F. Gao, T. S. Chan, L. Huang, and S. Liu, (2015). Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood, *Information Sciences.* 316(C), 180-200.

W. F. Gao, S. Y. Liu, L. L. Huang, (2012). A global best artificial bee colony algorithm for global optimization, *Journal of Computational & Applied Mathematics*. 236(11), 2741-2753.

W. F. Gao, S. Y. Liu, (2012). A modified artificial bee colony algorithm, *Computers & Operations Research*. 39(3), 687-697.

D. Karaboga, (2005). An Idea Based on Honey Bee Swarm for Numerical Optimization.

D. Karaboga, B. Basturk, (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Kluwer Academic Publishers*.

D. Karaboga, B. Basturk, (2008). On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing*. 8(1), 687-697.

D. Karaboga, B. Akay, (2009). A comparative study of Artificial Bee Colony algorithm, *Applied Mathematics & Computation*. 214(1), 108-132.

M. S. Kiran, H. Hakli, M. Gunduz, and H. Uguz, (2015). Artificial bee colony algorithm with variable search strategy for continuous optimization, *Information Sciences. 300*(C), 140-157.

W. C. Lin, C. C. Wu, K. Yu, Y. H. Zhuang, and S. C. Liu, (2017). On the use of genetic algorithm for solving re-entrant flowshop scheduling with sum-of-processing-times-based learning effect to minimize total tardiness, *Intelligent Automation & Soft Computing.* 1-11.

D. G. Mayer, B. P. Kinghorn, and Archer, A. A. (2005). Differential evolution–an easy and efficient evolutionary algorithm for model optimisation, *Agricultural Systems.* 83(3), 315-328.

K. Price, R. M. Storn, and J. A. Lampinen, (2005). Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series), *Springer-Verlag New York*.

S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, (2008). Opposition versus randomness in soft computing techniques, *Applied Soft Computing*. 8(2), 906-918.

H. Shan, T. Yasuda, K. Ohkura, (2015). A self adaptive hybrid enhanced artificial bee colony algorithm for continuous optimization problems, *Biosystems*. 132-133(7), 43-53.

D. Tian, (2017). Particle swarm optimization with chaos-based initialization for numerical optimization, *Intelligent Automation & Soft Computing.* (1), 1-12.

H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, (2011). Enhancing particle swarm optimization using generalized opposition-based learning, *Information Sciences.* 181(20), 4699-4714.

X. You, X. He, and X. Han, (2017). A novel solution to the cognitive radio decision engine based on improved multi-objective artificial bee colony algorithm and fuzzy reasoning, *Intelligent Automation & Soft Computing.* 23(4), 1-9.

M. Zhang, Z. Ji, and Y. Wang, (2017). Artificial bee colony algorithm with dynamic multi-population, *Modern Physics Letters B.* 31(19-21), 1740087.

Y. Zhou, (2009). Runtime analysis of an ant colony optimization algorithm for tsp instances, *IEEE*

Transactions on Evolutionary Computation. 13(5), 1083-1092.

X. Zhou, Z. Wu, C. Deng, and H. Peng, (2015). Neighborhood search-based artificial bee colony algorithm, *Journal of Central South University*. 46(2), 534-546.

G. Zhu, S. Kwong, (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization, *Applied Mathematics & Computation*. 217(7), 3166-3173.

7 NOTES ON CONTRIBUTORS

Xiaofang Liu is working toward the M.S. degree at the Huaqiao University, Quanzhou, Fujian, China. Her research interests are in Bionic intelligent computing and optimization theory.

Peizhong Liu (S'98 - M'07) received a B.E. degree from the Huaqiao University, Fujian, China, in 1998, and M.S. and Ph.D. degrees from Xiamen University, Xiamen, Fujian, China, in 2007 and 2010, respectively. Since 2011, he has been a Visiting Scholar with Duke University, USA. He is currently a Professor with the College of Engineering and acts as the Secretary General of the Neural Network and Computational Intelligence Committee of the Artificial Intelligence Society of China. His research interests include multi-dimensional space biomimetic informatics, visual media retrieval, network model, and information security.

Yanmin Luo, received B.S. and M.S. degrees in computer science, Huaqiao University, Quanzhou, Fujian, China, in 1993 and 2001, respectively, as well as Ph.D. degree in intelligent science, Xiamen University, Xiamen, Fujian, China, in 2013. He is currently an associate professor with the College of Computer Science and Technology, Huaqiao University, China. His research interests include artificial intelligence, machine learning, image processing, data mining and artificial immune learning algorithm.

Yongzhao Du, received a B.E. degree from the Huaqiao University, Fujian, China, in 2008, and a Ph.D. degree from the Sichuan University, Chengdu, Sichuan, China, in 2014. From Sep. 2012 to Oct. 2014, he has been a visiting student (supported by CSC) with UC Irvine, USA. He is currently an associate professor with the College of Engineering, Huaqiao University, China, and his research interests include optical imaging, medical image processing and computer-aided medical diagnosis.

Fan Yuling is an M.Sc candidate in Huaqiao University, Quanzhou, Fujian, China. His present research interest includes Bionic intelligent computing and Image Processing.

Hsuan-Ming Feng received the B.S. degree in automatic control engineering from Feng-Chia University, Taichung, Taiwan, R.O.C., in 1992. He received M.S. and Ph.D. degrees in Computer Science and Information Engineering from Tamkang University, Tamsui, Taipei Hsien, Taiwan, R.O.C., in 1994 and 2000, respectively. He is currently the full professor with the Department of Computer Science and Information Engineering, National Quemoy University. His current research interests include fuzzy systems, neural networks, wireless networks, optimal learning algorithms, image processing and robot system.