INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(4), 590-601, August 2018.

An ABC Algorithm with Recombination

X. You, Y. Ma, Z. Liu, M. Xie

Xuemei You*, Yinghong Ma, Zhiyuan Liu, Mingzhao Xie
Business School,
Shandong Normal University,
Shandong, 250014, P. R. China
*Corresponding author: sdxmyou@126.com yinghongma71@163.com
liuzhiyuan@sdnu.edu.cn

Abstract: Artificial bee colony (ABC) is an efficient swarm intelligence algorithm, which has shown good exploration ability. However, its exploitation capacity needs to be improved. In this paper, a novel ABC variant with recombination (called RABC) is proposed to enhance the exploitation. RABC firstly employs a new search model inspired by the updating equation of particle swarm optimization (PSO). Then, both the new search model and the original ABC model are recombined to build a hybrid search model. The effectiveness of the proposed RABC is validated on ten famous benchmark optimization problems. Experimental results show RABC can significantly improve the quality of solutions and accelerate the convergence speed.
Keywords: Artificial bee colony (ABC), recombination, hybrid search model, global optimization.

1 Introduction

In recent years, swarm intelligence (SI) has become a research focus in optimization field. The SI refers to establish mathematical model to simulate the social behaviors from nature. In the past decades, different SI algorithms have been proposed, including ant colony optimization (ACO) [8, 17, 25], particle swarm optimization (PSO) [22], artificial bee colony (ABC) [9], firefly algorithm (FA) [20], hybrid algorithm (HA) [14], bat algorithm (BA) [2], and cuckoo search (CS) [5, 30].

Except PSO and ACO, ABC can be regarded as the most popular SI algorithm. The main reason contains: 1) ABC has less control parameters than other SI algorithms; and 2) ABC has powerful exploration ability. Due to the superiority of ABC, it has received much attention. In the original ABC, individuals are divided into three types: i.e., employed bees, onlooker bees and scouts. During the search, all individuals (bees) fly in the search space and try to improve the food sources (find better solutions). Compared to other SI algorithms, ABC uses a different search equation to generate new solutions. For the current solution, an individual (employed bee or onlooker bee) randomly chooses a different solution in the population and uses their difference to obtain a new solution by modifying one dimension. Based on this search mechanism, ABC shows slow convergence speed and exploitation ability.

To strengthen the exploitation capacity, this paper presents a new ABC variant (called RABC), which employs a recombination method between the original ABC search model and a modified search model. For the latter model, it is inspired by the updating equation of PSO. By combining the search information of the global best solution and previous best, RABC can improve the exploitation ability. Experiment is validated on ten well-known benchmark problems. Simulation results of RABC are compared with the original ABC and two improved ABC variants.

The rest of this paper is organized as follows. The original ABC is briefly introduced in Section 2. A short review of recent progress on ABC is given in Section 3. In Section 4, our proposed RABC is described. Section 5 presents the simulation results and discussions. Finally, conclusion and future work are summarized in Section 6.

2 Artificial Bee Colony (ABC)

There are two famous SI optimization algorithms: PSO and ACO, which were developed in the 1990s. Recently, different SI algorithms were proposed. Among these algorithms, ABC becomes popular because of its superiorities [10]. In the original ABC, it consists of three types of individuals: employed bees, onlooker bees and scouts. For all food sources (solutions), the employed bees conduct the first round search around food sources and try to find new better solutions. The onlooker bees conduct the second round search around some selected better food sources. For a food source, if the above bees cannot find a better one to replace it after some rounds search, the scout will randomly find a food source to replace it.

2.1 Population initialization

In the following, we will describe the original ABC in details. First, a initial population consists of N food sources (solutions) $\{X_i|i = 1, 2, ..., N\}$, where X_i is the ith food source and N is the population size. The initial food sources are randomly generated as follows.

$$x_{ij} = low_j + rand_j \cdot (up_j - low_j) \tag{1}$$

where x_{ij} denotes the jth component for ith food source X_i, $i = 1, 2, ..., N$, $j = 1, 2, ..., D$; D is the dimensional size; $rand_j$ is a real random number between 0 and 1; up_j and low_j are the boundaries for the jth dimension.

2.2 Employed bee phase

For each food source X_i in the current population, each employed bee flies to its neighborhood and tries to find a new solution V_i. This process can be described as below [9].

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

where $j\epsilon[1, D]$ is a random index of the population; X_k is a randomly selected food source and it is mutually different with X_i; t represents the iteration index; ϕ_{ij} is a real random number in the range [-1, 1].

The ABC employs a greedy selection method to determine whether the new candidate solution (food source) V_i should be entered in the next generation. If V_i is better than X_i, then X_i is updated by V_i; otherwise X_i is unchangeable. The greedy selection can accelerate the population convergence.

2.3 Onlooker bee phase

The onlooker bees only fly to some selected food sources and search their neighborhood to find new food sources. The selection of each food source X_i is related to its fitness quality. A better food source has a higher selection probability. In the original ABC, the selection probability p_i for each food source X_i is defined by [9]:

$$p_i = \frac{fit_i}{\sum_{i=1}^{N} fit_i} \tag{3}$$

where fit_i is the fitness value of X_i. When X_i is selected, the onlooker bee searches the neighborhood of X_i and obtain a new solution V_i according to Eq.(2).

Similar to the employed bees, the same greedy selection is utilized by the onlooker bees to determine whether the new solution V_i should be entered in the next generation.

2.4 Scout bee phase

If the employed or onlooker bees cannot improve the quality of X_i in limit iterations, X_i may be stagnated. Then, a scout bee re-initializes X_i according to Eq.(1).

3 A brief review of ABC

Although the original ABC has shown good performance, it still has some drawbacks. To tackle these issues, many improved ABCs have been proposed. In this section, a brief review of recent advance on ABC is presented.

Karaboga and Akay [10] compared ABC with some evolutionary algorithms on a number of benchmark problems. Simulation results demonstrate ABC is competitive to those compared algorithms. Zhu and Kwong [32] proposed a new ABC called GABC , which introduced the global best individual into the original solution search equation. Akay and Karaboga [1] designed a new parameter MR to adjust the probability of dimension perturbation. Results demonstrate the modified approach can accelerate the search. Wang et al. [23] combined multi-strategy ensemble learning and ABC. It aims that multiple strategies can effectively balance the global and local search. Inspired by Gaussian DE [21], Zhou et al. [31] introduced Gaussian sampling into ABC to obtain good performance. Cui et al. [4] used a ranking method to choose the parent solutions when generating new solutions. Simulation results show the ranking based ABC is very effective. In [6], Cui et al. developed another version of ABC, which employs a dynamic population mechanism. During the search, the population size is not fixed and dynamically updated. In [3], Chen et al. combined teaching learning based optimization into ABC, and proposed an improved ABC variant to optimize the parameters estimation of photovoltaic.

To improve the exploitation, Xiang et al. [26] used a grey relational model to choose the neighbor individuals. Then, the DE mutation operator is employed to generate new candidate solutions. In [27], Xiang et al. proposed another ABC variant based on the cosine similarity. To select some good neighbor individuals, a novel solution search model is designed. The frequency of parameters perturbation is also modified to share more search information between different solutions. Simulation results on twenty-four benchmark functions show that the proposed Cos-ABC is competitive. Yaghoobi and Esmaeili [29] designed an improved ABC by using three new strategies: chaos theory, multiple searches, and modified perturbation. Song et al. [18] presented an improved ABC based on objective function value information, which employs two modified solution search models. The objective function value is incorporated to adjust the step size. Experiments on thirty test functions show the proposed approach is better than other six ABCs. Li et al. [13] designed a new gene recombination operation to accelerate the convergence of ABC. Some good solutions in the population are chosen to generate offspring through the gene recombination. Results demonstrate the gene recombination operation can effectively strengthen the exploitation capacity of ABC. Kong et al. [12] presented a new ABC variant with two strategies: elite group guidance and combined breadth-depth search. The proposed algorithm was verified on twenty-two benchmark functions. Sulaiman et al. [19] presented a robust ABC to balance exploitation and exploration. Experiments on 27 test functions and economic environmental dispatch (EED) problems show the effectiveness of the approach.

4 ABC with recombination (RABC)

In Section 2, it can be seen that ABC mainly uses Eq.(2) to find new food sources (solutions) during the iterations. From Eq.(2), the new solution V_i is very similar to its parent solution X_i, because they are different on only one dimension. Based this search mechanism, ABC shows powerful exploration ability, but its convergence speed is slow. To improve this case, some researchers introduced some good search experiences into ABC to accelerate the search. In [32], the global best solution is used to improve the search equation. Results show the modification can significantly improve the performance.

PSO is a successful SI optimization algorithm. The original PSO references has more than 100,000 citations. The main advantages of PSO focus on fast convergence speed and strong search ability. In PSO, each individual (also called particle) flies to its previous best (pbest) and the global best (*gbest*) found so far. So, PSO takes full advantage of the search experiences of those best individuals. The PSO updating model is defined by [11].

$$v_{ij}(t+1) = w \cdot v_{ij}(t) + c_1 r_1 (pbest_{ij}(t) - x_{ij}(t)) + c_2 r_2 (gbest_j(t) - x_{ij}(t)) \tag{4}$$

where V_i and X_i are the velocity and position, respectively; $w \epsilon [0, 1]$ is inertia weight; c_1 and c_2 are two learning factors; $r_1, r_2 \epsilon [0, 1]$ are two random numbers.

In this paper, a new solution updating equation is proposed inspired by PSO. However, this paper is not the first time to introduce PSO model into ABC. In [32], a gbest guided ABC (GABC) was proposed. In [28], Xiang et al. used the gbest and an elitist method to modify the solution search model. Liu [16] used pbest to update the employed bees and gbest to the onlooker bees. In [15], Li et al. defined a modified model as follows.

$$v_{ij}(t) = w \cdot x_{ij}(t) + 2\phi_{ij}(x_{ij}(t) - x_{kj}(t))\Phi_1 + \varphi_{ij}(gbest_j(t) - x_{kj}(t))\Phi_2 \tag{5}$$

where Φ_1 and Φ_2 are two positive values; ϕ_{ij} and φ_{ij} are two random numbers in [0,1].

Differs from these existing models, we design a new one by taking full advantage of *pbest* and *gbest*. The detailed model is defined as follows.

$$v_{ij}(t) = w \cdot x_{ij}(t) + r_1(pbest_{ij}(t) - x_{ij}(t)) + r_2(gbest_j(t) - x_{ij}(t)) \tag{6}$$

where j is a randomly selected dimension index, $r_1, r_2 \epsilon [0, 1]$ are two random numbers, and the weight factor $w \epsilon [0, 1]$.

Based on Eq.(6), we propose an alternative model by combining the original ABC model. The recombined model is described as below.

$$v_{ij}(t) = \begin{cases} x_{ij}(t) + \phi_{ij}(x_{ij}(t) - x_{kj}(t)), & if rand(0,1) < pr \\ w \cdot x_{ij}(t) + r_1(pbest_{ij}(t) - x_{ij}(t)) + r_2(gbest_j(t) - x_{ij}(t)), & Otherwise. \end{cases} \tag{7}$$

where $rand(0, 1)\epsilon[0, 1]$ is a random value, and $pr\epsilon[0, 1]$ is the probability rate.

In our approach RABC, both employed and onlooker bees use Eq. 6 to generate new solutions during iterations. Like the original ABC, a greedy selection method is also utilized to accelerate the search.

In comparison to the original ABC, RABC adds a probability parameter *pr* to control the usage of *pbest* and *gbest*. When *pr* is large, bees mainly use the original ABC model to generate new solutions. Then, RABC is similar to ABC. When *pr* is small, bees mainly employ the modified model to generate new solutions. Then, more search experiences of *pbest* and *gbest* are utilized in the search process. Therefore, *pr* plays an significant role in balancing exploration and exploitation in RABC. Moreover, RABC only modifies the search model, and does not employ other operations. So, ABC and RABC have the same computational time complexity.

5 Experimental study

5.1 Test functions

In this paper, ten famous classical benchmark functions are utilized to validate the performance of RABC. These functions were early used in many optimization papers [24]. The dimension D is set to 30 in the experiments. Table 1 briefly describes the ten benchmark problems. All functions should be minimized and their mathematical definitions are listed as below.

f_1: Sphere

$$f_1(x) = \sum_{i=1}^{D} x_i^2$$

f_2: Schwefel 2.22

$$f_2(x) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} x_i$$

f_3: Schwefel 1.2

$$f_3(x) = \sum_{i=1}^{D}(\sum_{j=1}^{i} x_j)^2$$

f_4: Schwefel 2.21

$$f_4(x) = max_i(|x_i|, 1 \leq i \leq D)$$

f_5: Rosenbrock

$$f_5(x) = \sum_{i=1}^{D-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

f_6: Step

$$f_6(x) = \sum_{i=1}^{D}(\lfloor x_i + 0.5 \rfloor)^2$$

f_7: Quartic with noise

$$f_7(x) = \sum_{i=1}^{D} ix_i^4 + rand(0, 1)$$

f_8: Schwefel 2.26

$$f_8(x) = \sum_{i=1}^{D} -x_i \sin(\sqrt{|x_i|})$$

f_9: Rastrigin

$$f_9(x) = \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i) + 10]$$

f_{10}: Ackley

$$f_{10}(x) = -20 \cdot exp(-0.2 \cdot \sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - exp(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi x_i)) + 20 + e$$

Table 1: Search range and global optimum for the benchmark problems.

Functions	Search range	Global optimum
f_1	[-100,100]	0
f_2	[-10,10]	0
f_3	[-100,100]	0
f_4	[-100,100]	0
f_5	[-30,30]	0
f_6	[-100,100]	0
f_7	[-1.28,1.28]	0
f_8	[-500,500]	$-418.98 \cdot D$
f_9	[-5.12,5.12]	0
f_{10}	[-32,32]	0

Table 2: PComputational results of RABC with different pr values.

Functions	$pr = 0.1$	$pr = 0.3$	$pr = 0.5$	$pr = 0.7$	$pr = 0.9$
	Mean	Mean	Mean	Mean	Mean
f_1	**5.94E-75**	6.51E-57	6.75E-48	7.46E-40	4.35E-33
f_2	**8.78E-36**	9.61E-33	1.54E-26	8.44E-20	5.65E-14
f_3	5.87E+03	6.40E+03	6.32E+03	**5.56E+03**	5.83E+03
f_4	**1.47E+00**	3.37E+00	9.89E+00	1.84E+01	3.32E+01
f_5	2.74E+01	2.54E+01	2.31E+01	1.61E+01	**1.43E-01**
f_6	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	2.79E-02	**2.25E-02**	3.59E-02	4.57E-02	1.06E-01
f_8	-10925.2	-12545.9	**-12569.5**	**-12569.5**	**-12569.5**
f_9	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{10}	2.90E-14	3.26E-14	**2.55E-14**	2.89E-14	1.73E-12

Sensitivity analysis of the parameter pr

According to Eq.(7), a new parameter *pr* is introduced in our approach RABC. As mentioned in Section 4, *pr* is beneficial for balancing the exploration and exploitation. A large *pr* is good for exploration and a small *pr* is helpful for exploitation. So, how to choose a suitable *pr* is worthy to be investigated.

To analyze the parameter *pr*, we try to test different *pr* on the benchmark set. The parameter *pr* is set to 0.1, 0.3, 0.5, 0.7, and 0.9, respectively. For other parameters N and $limit$, they are set to 50 and 100, respectively. The termination criterion for running an algorithm is maximum number of function evaluations ($MaxFEs$). When the number of function evaluations reaches to $MaxFEs$, the algorithm is stopped. According to the literature [23], $MaxFEs$ is equal to 1.5E+05. For each parameter *pr*, RABC is run 25 trials.

Table 2 displays the results of RABC with different *pr*, where "$Mean$" is the mean best function value. As seen, when $pr = 0.1$, RABC outperforms other *pr* values on two functions f_1 and f_2. For functions $f_3 - f_5$, all *pr* values cannot help RABC find good solutions. on function f_3, $pr = 0.7$ is slightly better than other pr values, and $pr = 0.1$ is better on function f_4. For function f_5, $pr = 0.9$ can find reasonable solution, while other *pr* values fail. All *pr* values and find the global optimum on f_6 and f_9. When $pr > 0.3$, RABC can converge to the global optimum, but RABC with $pr = 0.1$ and 0.3 falls into local minima. For function f_7, RABC with $pr = 0.2$ is better than other *pr* values. When $pr = 0.5$, RABC achieves better results on function f_10. From the above analysis, RABC with a fixed pr value cannot obtain better results than other *pr* values. So, it is not easy to select which *pr* is suitable for the benchmark set.

In order to choose the relatively best *pr*, Friedman test is used to calculate the mean rank of each pr on the benchmark set. Table 3 shows the mean rank values of RABC with different *pr* values. As shown, RABC with $pr = 0.1$ achieves the best mean rank. It demonstrates that $pr = 0.1$ is the relatively best choice among five different *pr* values. In the following experiment, $pr = 0.1$ is used for RABC.

Table 3: Mean rank of RABC with different *pr* values.

RABC	Mean rank
$pr = 0.1$	**2.70**
$pr = 0.3$	3.00
$pr = 0.5$	2.80
$pr = 0.7$	2.90
$pr = 0.9$	3.60

Comparison of RABC and other well-known ABC algorithms

In this section, we compare RABC with the standard ABC and two other well-known ABCs. The compared algorithms are listed as below.

- ABC [9]
- GABC GABC [32]
- MABC [7]
- Our approach RABC

In the following experiments, all algorithms use the same population size and stopping condition. Both N and $limit$ are set to 100. $MaxFEs$ is equal to 1.5E+05. The parameter C is equal to 1.5 in GABC [32]. In MABC, the probability P is set to 0.7 by the suggestions of [7]. The parameter pr in RABC is equal to 0.1 based on experimental study. Each algorithm is run 25 trial on each function.

Table 4 presents the results among RABC, ABC, GABC, and MABC on the ten benchmark functions, where "$Mean$" indicates the mean best function value. From the table, RABC outperforms the standard ABC on 7 functions, while ABC is better than RABC on f_5 and f_8. For the last function f_6, all ABCs can find the global optimum zero. Compared to GABC, RABC achieves better solutions on 5 functions. GABC performs better than RABC on 3 functions. Besides the function f_6, RABC, GABC, MABC can converge to the global minima on f_9. MABC is better than RABC on 2 functions, but RABC outperforms MABC on six functions.

Table 4: Comparison results among RABC, ABC, GABC and MABC.

Functions	ABC	GABC	MABC	RABC
	Mean	Mean	Mean	Mean
f_1	9.67E-16	6.86E-16	2.98E-40	**5.94E-75**
f_2	2.36E-10	1.39E-15	2.13E-21	**8.78E-36**
f_3	9.21E+03	**4.29E+03**	1.01E+04	5.87E+03
f_4	3.73E+01	1.91E+01	5.71E+00	**1.47E+00**
f_5	1.21E+00	6.77E-01	**2.27E-01**	2.74E+01
f_6	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	1.68E-01	7.98E-02	4.02E-02	**2.79E-02**
f_8	-12332.6	**-12569.5**	**-12569.5**	-10925.2
f_9	5.33E-14	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{10}	1.65E-09	3.97E-14	3.26E-14	**2.90E-14**

Fig.1 lists the convergence graphs of RABC and three other ABCs on six test functions. As shown, RABC converges faster than MABC, GABC, and ABC on f_1, f_2, f_4, f_9 and f_{10} on the whole search process. For function f_3, RABC is faster than other three algorithms at the beginning of the search. GABC is faster than RABC at the last stage. Because RABC converges to a local minima when FEs reaches to 3.0E+04. For function f_{10}, RABC converges much faster than other three algorithms at the beginning and middle stages. The convergence curves of RABC, MABC, and GABC are similar at the last stage of the search.

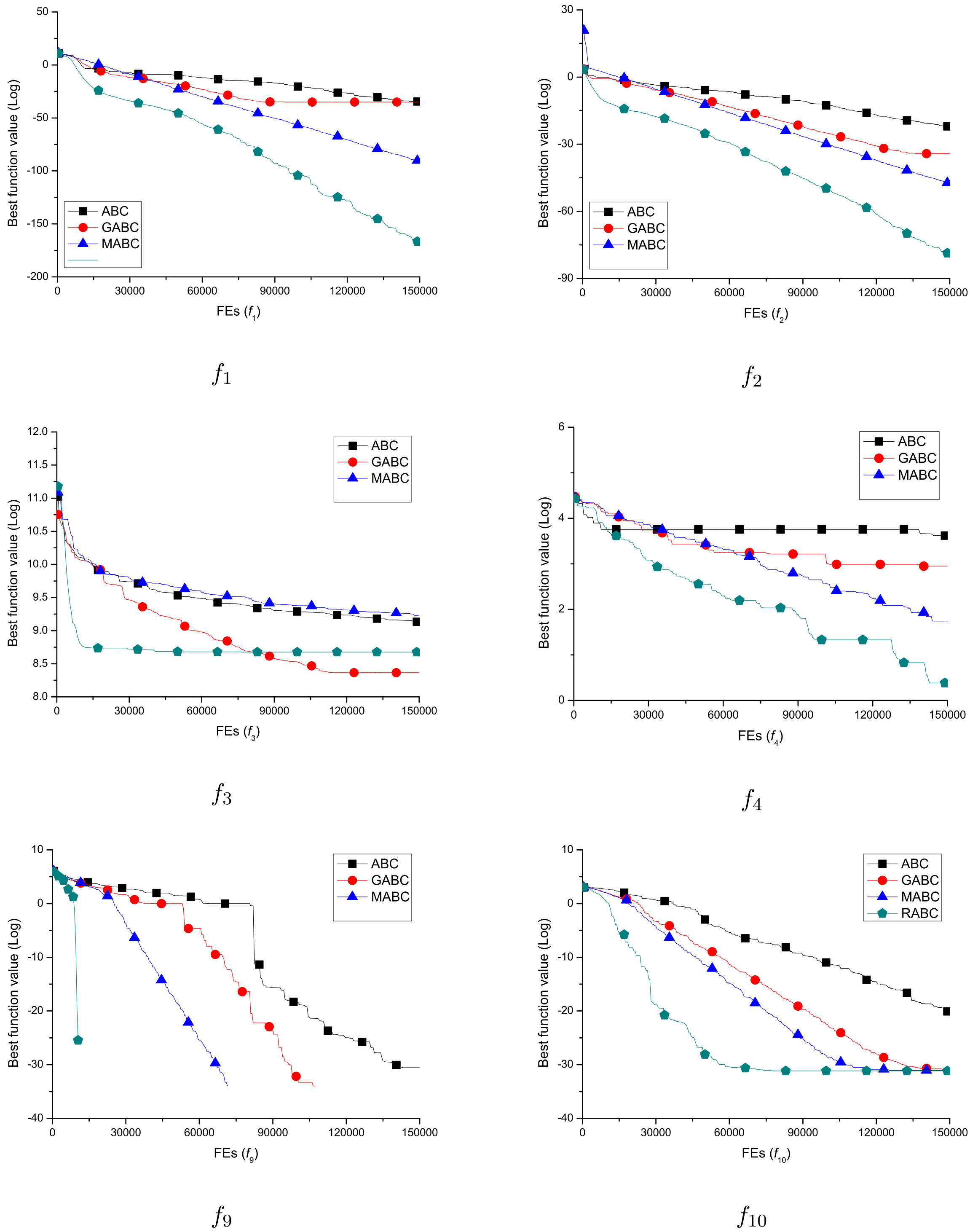

Figure 1: The convergence graphs of four ABCs on six selected functions.

6 Conclusion

In order to strengthen the exploitation ability of ABC, a new ABC variant with recombination (called RABC) is proposed in this paper. RABC firstly employs a new search model inspired by the updating equation of PSO. Then, both the new search model and the original ABC model are recombined to build a hybrid search model. A set of ten famous benchmark optimization problems are tested in the experiments. Results show RABC performs better than ABC, MABC, and GABC on most test functions.

The parameter pr aims to control the frequency of using $pbest$ and $gbest$. How to choose the best pr is studied in the experiments. Results show that RABC with a fixed pr value cannot obtain better results than other pr values. The statistical test demonstrates that $pr = 0.1$ is the relatively best choice among five different pr values. However, a fixed pr is not a good choice. A dynamic pr may be more suitable. This will be studied in the future work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 71701115, 71471106, 61502284, and 71704098), the Natural Science Foundation of Shandong Province (Nos. ZR2017MF058 and ZR2016GQ03), and the Higher School Science and Technology Foundation of Shandong Province (No. J17KA172).

Bibliography

[1] Akay, B.; Karaboga, D. (2012); A modified Artificial bee colony algorithm for real-parameter optimization, *Information Sciences*, 192, 120-142, 2012.

[2] Cai, X.; Wang, H.; Cui, Z.; Cai, J.; Xue, Y.; Wang, L.(2018); Bat algorithm with triangle-flipping strategy for numerical optimization, *International Journal of Machine Learning and Cybernetics*, 9(2), 199-215, 2018.

[3] Chen, X.; Xu, B.; Mei, C.; Ding, Y.; Li, K. (2018); Teaching Clearning Cbased artificial bee colony for solar photovoltaic parameter estimation, *Applied Energy*, 212, 1578-1588, 2018.

[4] Cui, L.; Li, G.; Wang, Z.; Lin, Q.; Chen, J.; Lu, N.; Lu, J. (2017); A ranking-based adaptive artificial bee colony algorithm for global numerical optimization, *Information Sciences*, 417, 169-185, 2017.

[5] Cui, Z.H.; Sun, B.; Wang, G.G.; Xue, Y.; Chen, J.J. (2017); A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber-physical systems, *Journal of Parallel and Distributed Computing*, 103, 42-52, 2017.

[6] Cui, L.; Li, G.; Zhu, Z.; Lin, Q.; Chen, J. (2017); A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization, *Information Sciences*, 414, 53-67, 2017.

[7] Gao, W.; Liu, S. (2012); A modified artificial bee colony algorithm, *Computers & Operations Research*, 39, 687-697, 2012.

[8] Huang, P.; Lin, F.; Xu, L.J.; Kang, Z.L.; Zhou, J.L.; Yu, J.S. (2017); Improved ACO-bsed seep coverage scheme considering data delivery, *International Journal of Simulation Modelling*, 16(2), 289-301, 2017.

[9] Karaboga, D. (2005); An idea based on honey bee swarm for numerical optimization, *Technical Report-TR06, Erciyes University, engineering Faculty, Computer Engineering Department*, 2005.

[10] Karaboga, D.; Akay, B. (2009); A comparative study of artificial bee colony algorithm, *Applied Mathematics and Computation*, 214, 108-132, 2009.

[11] Kennedy, J.; Eberhart, R. (1995); Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks*, 1942-1948, 1995.

[12] Kong, D.; Chang, T.; Dai, W.; Wang, Q.; Sun, H. (2018); An improved artificial bee colony algorithm based on elite group guidance and combined breadth-depth search strategy, *Information Sciences*, 442-443, 54-71, 2018.

[13] Li, G.; Cui, L.; Fu, X.; Wen, Z.; Lua, N.; Lu, J. (2017); Artificial bee colony algorithm with gene recombination for numerical function optimization, *Applied Soft Computing*, 52, 146-159, 2017.

[14] Li, J.; Pan, Q.; Xie, S.; Wang, S. (2011); A Hybrid Artificial Bee Colony Algorithm for Flexible Job Shop Scheduling Problems, *International Jotrnal of Computers Communications & Control*, 6(2), 286-296, 2011.

[15] Li, G.; Niu, P.; Xiao, X. (2012); Development and investigation of efficient artificial bee colony algorithm for numerical function optimization, *Applied Soft Computing*, 12(1), 320-332, 2012.

[16] Liu, J.J.; Zhu, H.Q.; Ma, Q.; Zhang, L.L.; Xu, H.L. (2015); An artificial bee colony algorithm with guide of global & local optima and asynchronous scaling factors for numerical optimization, *Soft Computing*, 37, 608-618, 2015.

[17] Rajput, U.; Kumari, M. (2017); Mobile robot path planning with modified ant colony optimisation, *International Journal of Bio-Inspired Computation*, 9(2), 106-113, 2017.

[18] Song, X.; Yan, Q.; Zhao, M. (2017); An adaptive artificial bee colony algorithm based on objective function value information, *Applied Soft Computing*, 55, 384-401, 2017.

[19] Sulaiman, N.; Mohamad-Saleh, J.; Abro, A.G. (2017); Robust variant of artificial bee colony (JA-ABC4b) algorithm, *International Journal of Bio-Inspired Computation*, 10(2), 99-108, 2017.

[20] Wang, H.;Wang, W.; H. Sun, H.; Rahnamayan, S. (2016); Firefly algorithm with random attraction, *International Journal of Bio-Inspired Computation*, 8(1), 33-41, 2016.

[21] Wang, H.; Rahnamayan, S.; Sun, H.; Omran, M.G.H. (2013); Gaussian bare-bones differential evolution, *IEEE Transactions on Cybernetics*, 43(2), 634-647, 2013.

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

[23] Wang, H.; Wu, Z.J.;Rahnamayan, S.; Sun, H.; Liu, Y.; Pan, J.S. (2014); Multi-strategy ensemble artificial bee colony algorithm, *Information Sciences*, 279, 587-603, 2014.

[24] H. Wang; H. Sun; C, Li; S. Rahnamayan; J.S. Pan; Diversity enhanced particle swarm optimization with neighborhood search, *Information Sciences*, 223, 119-135, 2013.

[25] Wu, J.; Wu, G.D.; Wang, J.J. (2017); Flexible job-shop scheduling problem based on hybrid ACO algorithm, *International Journal of Simulation Modelling*, 16(3), 497-505, 2017.

[26] Xiang, W.; Li, Y.; Meng, X.; Zhang, C.; An, M. (2017); A grey artificial bee colony algorithm, *Applied Soft Computing*, 60, 1-17, 2017.

[27] Xiang, W.; Li, Y.; He, R.; Gao, M.; An, M. (2018); A novel artificial bee colony algorithm based on the cosine similarity, *Computers & Industrial Engineering*, 115, 54-68, 2018.

[28] Xiang, Y.; Peng, Y.M.; Zhong, Y.B.; Chen, Z.Y.; Lu, X.W.; Zhong, X.J. (2014); A particle swarm inspired multi-elite artificial bee colony algorithm for real-parameter optimization, *Computational Optimization and Applications*, 57, 493-516, 2014.

[29] Yaghoobi, T.; Esmaeili, E. (2017); An improved artificial bee colony algorithm for global numerical optimisation, *International Journal of Bio-Inspired Computation*, 9(4), 251-258, 2017.

[30] Zhang, M.; Wang, H.; Cui, Z.; Chen, J. (2017); Hybrid Multi-objective cuckoo search with dynamical local search, *Memetic Computing*, doi: 10.1007/s12293-017-0237-2, 2017.

[31] Zhou, X.; Wu, Z.; Wang, H.; Rahnamayan, S. (2016); Gaussian bare-bones artificial bee colony algorithm[J], *Soft Computing*, 20(3), 907-924, 2016.

[32] Zhu, G.; Kwong, S. (2010); Gbest-guided artificial bee colony algorithm for numerical function optimization, *Applied Mathematics and Computation*, 217, 3166-3173, 2010.