Artificial Bee Colony Algorithm with Crossover Strategies for Global Numerical Optimization

Sheng-Ta Hsieh and Jhih-Sian Chen
Department of Communication Engineering Oriental Institute of Technology, Taiwan
(Tel: +886-2-7738-0145 Ext.2325, Fax: +886-2-7738-7411)

fo013@mail.oit.edu.tw

Abstract: In this paper, a new variant of artificial bee colony (ABC) algorithm is proposed for solving numerical optimization problems. In order to increase population's solution searching ability, the crossover operation of genetic algorithm (GA) is involved to produce new potential offspring. In the experiment, the CEC 2005 test functions are adopted for test proposed method and compared it to related works. From the results, it can be observed that the proposed method performed better performance than two variants of ABC approaches.

Keywords: artificial bee colony, crossover, genetic algorithm, optimization.

1 INTRODUCTION

In order to deal with complicate real-world optimization problems, more and more researchers are focus on creating new optimizers or improving its performance. Such as Genetic Algorithm (GA) [1], Ant Colony Optimization (ACO) [2], Particle Swarm Optimization (PSO) [3] and Differential Evolution(DE) [4] etc.

Artificial bee colony algorithm was first proposed by Karaboga [5]-[6] in 2005. It simulated Bees' forage behavior. Then, Tereshko and Loengarov [7] proposed the foraging model for ABC algorithm.

The ABC algorithm is consisted of three basic elements, employed bees, onlooker bees and scouts bees. The employed bee, onlooker bee will try to find food. While employed and onlooker bees cannot find any food, then, the scout bee will start over for following food searching process and looking for new food source.

In recent years, more and more researchers are studying ABC algorithms either enhancement or applications. In 2010, Banharnsakun *et al*. proposed distributed population [9], which divide whole population into several small sub-populations. Each sub-population will perform its own solution search process in parallel. The sub-swarm will exchange their information after several iterations for improving solution quality and improving solution searching process. In 2011, Bi and Wang proposed FMABC [8], which introduced a free search algorithm to replace traditional roulette wheel for selecting onlooker bees. Further, it also combined an opposition-based strategy to replace scouts bee. Both strategies will speed up solution searching process. In the same year, Guo *et al*. proposed GABCS [10] which refer to PSO moving behavior to modify bees' moving vector. Also, a mutated strategy is involved to generate new bees for increasing bees' solution searching ability. In 2012, El-Abd proposed an interesting ABC variant named GOABC [11], which introduced a opposition-based learning and involved generalized concept into ABC for enhancing optimizer's performance. The GOABC exhibit good results on solving both uni-modal and multi-modal test functions.

However, an optimizer may has higher ability for solving uni-modal problems due to its deep search approach, but it could make optimizer has a weak ability on widely search for solving multi-modal problems, and vice versa. In other word, for the deep and widely search ability, improved one will worsen the other one. Thus, how to make choose a suitable strategy to deal with current problem will be an important issue of optimizer improvement.

In order to enhance ABC optimizer's solution searching ability. In this paper, the crossover of GA is involved into to generate more potential individual. Also, the elitist strategy is involved to speed up convergence.

This paper is consisted of five sections. The basic concept of ABC algorithm is described in section 2. The detail of proposed method is introduced in section 3. The experiments are presented in section 4. Finally, the conclusions are described in section 5.

2 ARTIFICIAL BEE COLONY

2.1 Introduction to ABC algorithm

Artificial bee colony (ABC) algorithm is a novel optimizer which simulated Bees' forage behavior in solution space and try to looking for optimal solution in reasonable computational consumption In ABC, there are 2 kinds of bees, employed and onlooker bees, are response for finding food source (also named solutions). The employed bees are going to explore new food and onlooker bees are response for better food source. The food searching process will be performed by following equation.

$$v_{ij} = x_{ij} + \emptyset_{ij} \times (x_{ij} - x_{kj}) \tag{1}$$

where both i and k are random integer, whose range are between $[1, s]$, and i is not equal to k. The s denotes population size. In other word, i and k are random selected bees. The j is also a random integer whose range is between $[1, D]$. The D denotes dimension. $\emptyset_{ij}$ is a normal distribution number which is between [-1, 1], x denotes current food source and v is the new food source.

Onlooker bees will then select a food source according to equation (2) and perform food searching process by equation (1).

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{2}$$

where fit_i is fitness value and i denotes the i_{th} bee. The fitness value will be updated by following equation.

$$fit_i = \begin{cases} \frac{1}{(1+f_i)}, & if\ f_i \geq 0 \\ 1 + \mathrm{abs}(f_i), & if\ f_i < 0 \end{cases} \tag{3}$$

where f_i represents objective value of i_{th} bee. After several generations, if there is no better food source be found. The scout bees will take over the food search process and try to looking for new food source (better solutions) by following equation.

$$x_i^{rand} = x_i^{min} + rand(0,1)(x_i^{max} - x_i^{min}) \tag{4}$$

where x^{rand} denotes random selected bee, and x^{min} and x^{max} present minimal and maximal boundary of search range respectively.

The procedure of ABC algorithm is listed as follows.

Step 1: Initialization for generated food source randomly.
Step 2: Calculate fitness value for each food source.
Step 3: Employed bees looking for new food source by (1) and evaluate x_i and v for select better food source.
Step 4: Calculate probability of fitness value by (2).
Step 5: Onlooker bees will select food source by roulette wheel and keep looking for better food by (1).
Step 6: Evaluate x_i and v for select better food source.
Step 7: Record the best food source.
Step 8: If no better be found in limited generations, use (4) to find new food source.
Step 9: Repeat step 3 to 8, until terminal condition is reached

3 PROPOSED METHOD

In order to enhance ABC optimizer's solution searching ability. In this paper, the crossover of GA is involved into to generate more potential individual. Also, the elitist strategy is involved to speed up convergence.

3.1 Crossover

In genetic algorithm, crossover is going to produce offspring by re-combine parents' information. In this paper, multi-point crossover is involved to generate new individual for searching potential/unsearched solution space.

The multi-point crossover will applied on random selected dimension and perform information exchange for selected bees. Thus, it will drive bees toward to right direction and perform deep and widely search. The same as GA, the crossover rate *Cr* is also set.

In order to prevent solution from fall into local optimum, the multi-point crossover will not perform every generation but every *g* generations.

3.2. Elitist strategy

The elitist strategy is to select the best bee for guide other bees toward to right direction and perform solution search for finding better food source. Thus, the equation (1) is modified as following equation.

$$v_{ij} = x_{ij} + \emptyset_{ij} \times (x_{ij} - x_j^{best}) \tag{5}$$

where x^{best} is the best bee of population in current generation.

Although the elitist strategy will speed up convergence of solution searching process, it could also make all the bees are clustered in a small area of solution space. In order to avoid this situation, the elitist strategy will not be activated every generation but by an activation rate *Er*.

3.3. Procedure of proposed method

The procedure of proposed method is listed as follows.

Step 1: Initialization for generated food source randomly.
Step 2: Calculate fitness value for each food source.
Step 3: If the random value is smaller than *Er*, jump to step 5, else execute step 4.
Step 4: Employed bees looking for new food source by (1)

and evaluate x_i and v for select better food source.

Step 5: Employed bees looking for new food source by (5) and evaluate x_i and v for select better food source.

Step 6: Calculate probability of fitness value by (2).

Step 7: Onlooker bees will select food source by roulette wheel and keep looking for better food. If the random value is smaller than Er, use (5), else use (1).

Step 8: Evaluate x_i and v for select better food source.

Step 9: If the random value is smaller than Cr, perform crossover, else jump to step 10.

Step 10: Record the best food source.

Step 11: If no better be found in limited generations, use (4) to find new food source.

Step 12: Repeat step 3 to 11, until terminal condition is reached

4 EXPERIMENT RESULTS

4.1 Test functions

In order to test the performance of proposed method and compare it with other variants of ABC, the twenty five test function of CEC 2005 [12] are selected which include uni-modal functions (f_1 ~ f_5), multi-modal functions (f_6 ~ f_{12}), expanded functions (f_{13} ~ f_{14}) and hybrid composition functions (f_{15} ~ f_{25}). All the test functions are set as 30 dimensions. In order to easier compare the performance between optimizers, the error e between the real global optimum f^* and function value f found by optimizer will be presented. Thus, the bias of CEC 2005 test functions is not involved.

$$e = f - f^* \qquad (6)$$

The search range and global optimum of all test function is listed in Table 1.

TABLE 1. GLOBAL OPTIMUM AND SEARCH RANGE OF CEC 2005 TEST FUNCTIONS

Test Functions	Global Optimum	Search range
F_1 ~ F_6	0	$[-100, 100]^D$
F_7	0	$[0, 600]^D$
F_8	0	$[-32, 32]^D$
F_9 ~ F_{10}	0	$[-5, 5]^D$
F_{11}	0	$[-0.5, 0.5]^D$
F_{12}	0	$[-100, 100]^D$
F_{13}	0	$[-3, 1]^D$
F_{14}	0	$[-100, 100]^D$
F_{15} ~ F_{24}	0	$[-5, 5]^D$
F_{25}	0	$[-2, 5]^D$

TABLE 2. EXPERIMENT RESULTS

Algorithms	Results	F_1	F_2	F_3	F_4	F_5
ABC	Mean	**0.00E+00**	6.00E+03	8.81E+06	3.04E+04	1.26E+04
	Std.	**0.00E+00**	2.87E+03	2.50E+06	6.74E+03	1.45E+03
GOABC	Mean	**0.00E+00**	2.72E+03	6.15E+06	3.18E+04	1.20E+04
	Std.	**0.00E+00**	9.13E+02	2.07E+06	6.52E+03	1.61E+03
Proposed Method	Mean	**0.00E+00**	**2.98E+02**	**1.89E+06**	**9.59E+03**	**3.21E+03**
	Std.	**0.00E+00**	**1.57E+02**	**6.83E+05**	**2.86E+03**	**6.63E+02**

Algorithms	Results	F_6	F_7	F_8	F_9	F_{10}
ABC	Mean	**5.17E+00**	4.70E+03	2.08E+01	2.89E-11	3.89E+02
	Std.	**7.93E+00**	1.74E-12	4.65E-02	1.41E-10	5.59E+01
GOABC	Mean	8.46E+00	**3.29E+03**	2.08E+01	**0.00E+00**	3.48E+02
	Std.	1.23E+01	**2.22E+02**	4.05E-02	**0.00E+00**	5.63E+01
Proposed Method	Mean	1.93E+01	4.70E+03	**2.04E+01**	**0.00E+00**	**8.73E+01**
	Std.	1.89E+01	5.75E-13	**4.81E-02**	**0.00E+00**	**2.10E+01**

Algorithms	Results	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
ABC	Mean	2.83E+01	1.27E+04	1.10E+00	1.31E+01	1.85E+01
	Std.	1.44E+00	5.83E+03	1.50E-01	1.31E-01	5.47E+01
GOABC	Mean	2.97E+01	8.43E+03	**8.08E-01**	1.30E+01	2.94E+01
	Std.	1.47E+00	3.86E+03	**1.48E-01**	1.13E-01	6.29E+01
Proposed Method	Mean	**2.37E+01**	**2.84E+03**	8.59E-01	**1.26E+01**	**4.00E+00**
	Std.	**1.09E+00**	**2.19E+03**	1.85E-01	**1.35E-01**	**1.96E+01**

Algorithms	Results	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}
ABC	Mean	3.43E+02	2.75E+02	9.24E+02	9.22E+02	9.21E+02
	Std.	3.99E+01	3.61E+01	6.32E+00	5.19E+00	6.95E+00
GOABC	Mean	3.47E+02	3.75E+02	9.25E+02	9.23E+02	9.25E+02
	Std.	3.05E+01	4.30E+01	6.00E+00	4.39E+00	6.20E+00
Proposed Method	Mean	**1.64E+02**	**2.07E+02**	**9.08E+02**	**9.08E+02**	**9.04E+02**
	Std.	**2.93E+01**	**5.82E+01**	**1.45E+00**	**2.12E+00**	**7.96E+00**

Algorithms	Results	F_{21}	F_{22}	F_{23}	F_{24}	F_{25}
ABC	Mean	5.00E+02	1.12E+03	5.34E+02	**2.00E+02**	**1.02E+03**
	Std.	3.68E-03	4.41E+01	6.08E-04	**8.76E-13**	**7.47E+00**
GOABC	Mean	5.00E+02	1.09E+03	**5.33E+02**	2.33E+02	1.26E+03
	Std.	1.09E-12	4.63E+01	**7.86E-04**	2.97E+01	1.17E+01
Proposed Method	Mean	**5.00E+02**	**9.43E+02**	5.34E+02	2.56E+02	1.23E+03
	Std.	**4.62E-13**	**2.18E+01**	3.23E-04	3.29E+01	2.00E+01

4.2 Parameter settings

In order to fair comparison, all the parameters are according to their original settings. The population size of proposed method is referring to [11]. The crossover will be activated every 300 generations, and the crossover rate is set as 0.5. The maximum Fitness Evaluations (FEs) are set as 300,000.

4.3 Experiment results

Table 2 presents the mean, standard deviation, and average computation time of 25 runs of the 3 variants of ABC approaches on the 25 test functions with 30 dimensions. The best results among the three approaches are shown in bold.

From the results, the proposed method performed with better results on most test functions can be observed. The proposed method surpasses all other algorithm in solving all functions and a significant improvement on the results of functions 2, 4, 5, 15 and 22. The Proposed method performed the same best result as the ABC [5] and GOABC [11] on function 1. The Proposed method performed the same best result as the GOABC on function 9.

5 CONCLUSIONS

In this paper, both the crossover and elitist strategy are involved into ABC algorithm for solving numerical optimizations. The crossover can produce more potential individuals for improving bees' searching abilities, for easier to find the global optimal solution. It also makes ABC more robust, prevents bees from falling into the local optimum.

The experimental results proved that the proposed method can find better solutions than other ABC variants. Twenty five test functions were adopted for testing through a reasonable average. Form the results; it can be observed that the proposed has high ability for solving uni-modal functions, multi-modal functions, expanded functions and hybrid composition functions.

ACKNOWLEDGEMENT

This work was supported in part by National Science Council of Taiwan, Taiwan, R.O.C. under Grant NSC 101-2221-E-161-011.

REFERENCES

[1] J. H. Holland, "*Adaptation in Natural and Artificial Systems*, University of Michigan Press," Ann Arbor, MI, 1975.

[2] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a colony of cooperaing agents," *IEEE Trans. on SMC*, pp. 29-41, 1996.

[3] J. Kennedy and R. C. Eberhart, "in: Particle Swarm Optimization," in *Proc.* of 1995 IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948, 1995.

[4] R. Storn and K. Price, "Differential evolution – a simple and efficient daptive scheme for global optimization over continuous spaces," *Technical report*, International Computer Science Institute, 1995.

[5] D. Karaboga, "An idea based on honeybee swarm for numerical optimization," Technical Report TR06, 2005.

[6] D. Karaboga and B. Akay, "Artificial bee colony (ABC), harmony search and bees algorithms on numerical optimization," in Innovative Production Machines and Systems Virtual Conference, 2009.

[7] V. Tereshko and A. Loengarov, "Collective Decision-Making in Honey Bee Foraging Dynamics," *Computing and Information Systems Journal*, vol. 9, No 3, ISSN 1352-9404, 2005.

[8] X. Bi and Y. Wang, "An improved artificial bee colony algorithm," in *Proc.* of 2011 3rd International Conference on Computer Research and Development (ICCRD), vol.2, pp.174-177, 2011.

[9] A. Banharnsakun, T. Achalakul and B. Sirinaovakul, "Artificial bee colony algorithm on distributed environments," in *Proc.* of 2010 Second World Congress on Nature and Biologically Inspired Computing (NaBIC), pp.13-18, 2010.

[10] P. Guo, W. Cheng and J. Liang , "Global artificial bee colony search algorithm for numerical function optimization," in *Proc.* of 2011 Seventh International Conference on Natural Computation (ICNC), vol.3, pp.1280-1283, 2011.

[11] M. El-Abd, "Generalized opposition-based artificial bee colony algorithm," in *Proc.* of 2012 IEEE Congress on Evolutionary Computation (CEC), pp.1-4, 2012.

[12] http://www.ntu.edu.sg/home/EPNSugan