Hindawi
Journal of Applied Mathematics
Volume 2018, Article ID 1237823, 9 pages
https://doi.org/10.1155/2018/1237823

Research Article

A Modified Artificial Bee Colony Algorithm with Firefly Algorithm Strategy for Continuous Optimization Problems

Amnat Panniem and Pikul Puphasuk

Department of Mathematics, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand

Correspondence should be addressed to Pikul Puphasuk; ppikul@kku.ac.th

Received 16 July 2018; Accepted 3 December 2018; Published 18 December 2018

Academic Editor: Wei-Chang Yeh

Artificial Bee Colony (ABC) algorithm is one of the efficient nature-inspired optimization algorithms for solving continuous problems. It has no sensitive control parameters and has been shown to be competitive with other well-known algorithms. However, the slow convergence, premature convergence, and being trapped within the local solutions may occur during the search. In this paper, we propose a new Modified Artificial Bee Colony (MABC) algorithm to overcome these problems. All phases of ABC are determined for improving the exploration and exploitation processes. We use a new search equation in employed bee phase, increase the probabilities for onlooker bees to find better positions, and replace some worst positions by the new ones in onlooker bee phase. Moreover, we use the Firefly algorithm strategy to generate a new position replacing an unupdated position in scout bee phase. Its performance is tested on selected benchmark functions. Experimental results show that MABC is more effective than ABC and some other modifications of ABC.

1. Introduction

Most continuous optimization problems in various application areas such as science, engineering, economics, and management are nonlinear and difficult to find the optimal solutions. Several nature-inspired optimization algorithms including particle swarm optimization algorithm [1], Bee algorithm [2], Firefly algorithm [3], Bat algorithm [4], and Artificial Bee Colony (ABC) algorithm [5] have been developed and applied to solve the problems. The ABC algorithm, proposed by Karaboga in 2005, has been shown to be competitive to some other algorithms [6–9]. It is widely used in many applications, for example, image template matching [10], virus evolution [11], inverse analysis problem [12], and clustering problem [13]. According to the experimental results in [5, 9, 14–16], ABC still has some deficiencies in dealing with the functions having narrow curving valley and the multimodal functions. It also faces a slow convergence speed [9, 11, 15, 17–21], premature convergence, and getting trapped into local optima [9, 15, 17, 22].

In order to address these deficiencies, some improved versions of ABC have been proposed. In 2010, Zhu and Kwong presented the Gbest-guided ABC (GABC) algorithm [14]. They improved the search equation to increase the exploitation in employed and onlooker bee phases by using the strategy of the particle swarm optimization. The experimental results showed that GABC outperformed ABC for most test problems. In 2012, Gao et al. proposed a modified ABC algorithm (ABC/best) [9], which is inspired by the differential evolution algorithm, to improve the exploitation by searching only around the best solution of the previous iteration. Their results showed better performance when compared with ABC. In 2016, Anuar et al. developed ABC with the rate of change technique (ABC-ROC) based on the changing of slope on the performance graph to replace the parameter *limit* in the scout bee phase of ABC [23] and ABC-ROC also produces promising results.

In this paper, we proposed the Modified Artificial Bee Colony (MABC) algorithm to improve all phases of ABC algorithm. We generate the initial population by using the search space division as in [24] to provide high quality initial solutions. We use a new search equation for employed and onlooker bees, increase an opportunity for onlooker bees to search better food source positions, replace some worst

positions by the new ones, and use the strategy of Firefly algorithm to improve unupdated positions for scout bees. The performance of MABC is tested on selected benchmark functions and compared with those of ABC, GABC, ABC-ROC, and ABC/best.

2. Algorithm Description

2.1. Artificial Bee Colony Algorithm. Artificial bee colony algorithm, proposed by Karaboga in 2005 [6], is a relatively new nature-inspired optimization algorithm which is inspired by the behaviour of honeybee swarms. There are three kinds of population bees (employed bees, onlooker bees, and scout bees) working together to search for food source positions. Each employed bee searches for a new food source by communicating with another bee. If a new better position is found, the employed bee will memorize this position instead of the old one. Then the onlooker bees make the decisions to choose the food sources for exploration by using the information from employed bees. Again, if a new better position is found, the onlooker bee will also memorize this position. The employed bee whose food source has been abandoned for a period of time becomes a scout bee with the newly generated position for the next search cycle. The ABC algorithm can be described as follows:

(1) *Initialization*: The initial i^{th} food source $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$ associated with the i^{th} bee is generated by

$$x_{i,j} = L_j + \phi_{ij}\left(U_j - L_j\right) \tag{1}$$

for $i = 1, 2, 3, \ldots, NP$ and $j = 1, 2, 3, \ldots, D$, where NP is the number of bees and D is the number of variables or dimension, ϕ_{ij} is a random number in range of $(0, 1)$, and L_j and U_j are the lower and upper bounds for the dimension j.

(2) *Employed bee phase*: The i^{th} bee shares a piece of information with the k^{th} bee and generates a new food source v_i by using the following equation:

$$v_{i,j} = \begin{cases} x_{i,j} + \phi_{ij}\left(x_{i,j} - x_{k,j}\right); & j = j^*, \\ x_{i,j}; & j \neq j^*, \end{cases} \tag{2}$$

where k is randomly chosen from 1 to NP such that $k \neq i$, j^* is randomly chosen from 1 to D, and ϕ_{ij} is a random number in $[-1, 1]$. Note that v_i is different from x_i only at the j^{*th} component. The new position v_i is evaluated and compared with the old x_i. If $f(v_i) < f(x_i)$, then replace x_i by v_i; otherwise, hold x_i and set $trial(i) = trial(i) + 1$ where $trial$ is the counter number of unimproved trials.

(3) *Onlooker bee phase*: Onlooker bees select the food sources depending on their quality using the probability values $p(i)$ which are computed by

$$p(i) = \frac{fit\left(x_i\right)}{\sum_{j=1}^{NP} fit\left(x_j\right)}, \tag{3}$$

where

$$fit\left(x_i\right) = \begin{cases} \dfrac{1}{1 + f\left(x_i\right)} & ; f\left(x_i\right) \geq 0, \\ 1 + \left|f\left(x_i\right)\right| & ; \text{otherwise.} \end{cases} \tag{4}$$

If $\text{rand}(0, 1) < p(i)$, then generate a new v_i by the same equation (2). If $f(v_i) < f(x_i)$, then replace x_i by v_i; otherwise, retain x_i and set $trial(i) = trial(i) + 1$.

(4) *Scout bee phase*: If the food source x_i cannot be improved through the limitation number of trails ($limit$), then generate a new position for x_i by using (1).

(5) Find the best position x_{best} and the best value f_{best}.

(6) Repeat steps (2)–(5) until the stopping criterion is reached.

2.2. Firefly Algorithm. Firefly algorithm strategy is used in the scout bee phase of MABC algorithm, and its concept which is introduced by Yang in 2008 [3] is mentioned here. Firefly algorithm (FA) is inspired by the flashing behaviour of fireflies where their movements depend on the light intensity (brightness) and the attractiveness. The attractiveness is proportional to the brightness of a firefly; i.e., for any two fireflies the less bright one is attracted by the brighter one. On the other hand, the attractiveness is inversely proportional to the distance between two fireflies where the distance between any two fireflies x_i and x_j is given by the Euclidean norm

$$r_{ij} = \left(\sum_{k=1}^{D}\left(x_{ik} - x_{jk}\right)^2\right)^{1/2}. \tag{5}$$

The attractiveness β_{ij} is computed by

$$\beta_i = \beta_0 e^{-\gamma r_{ij}^2} \tag{6}$$

where β_0 is the original light attractiveness at $r = 0$ and γ is the light absorption coefficient. For simplicity, β_0 and γ are usually set to 1. From the movement of firefly x_i to another more attractive firefly x_j, a new position is given by

$$x_i = x_i + \beta_{ij}\left(x_j - x_i\right) + \alpha\left(\text{rand}\left(0, 1\right) - 0.5\right) \tag{7}$$

where $\text{rand}(0, 1)$ and α are the random numbers between 0 and 1.

3. Modified Artificial Bee Colony Algorithm

To construct MABC algorithm, all phases of ABC are determined. For the initialization, the population is generated based on search space division (SSD) proposed by He et al. [24]. For employed bee phase, we improve the search equation of ABC by gradually using the information of the best solution to accelerate the search. For onlooker bee phase, 25% of employed bees are selected with the same probability for additional search moves. Then 5% of worst positions are replaced by the new positions constructed by using the

information of the current best solution and the number of the best solutions (M) as the scaling factor for providing long distance moves in the case that many best solutions are found for the multimodal functions. For scout bee phase, Firefly algorithm strategy is applied to construct the new position by moving an unupdated position to a new position based on the distance to a better solution. MABC algorithm is proposed as follows.

(1) *Initialization*: To provide high quality initial solutions, generate the i^{th} food source x_i by using the search space division

$$x_{i,j} = L_j + \frac{\left(\phi_{ij} + 2i - 1\right)\left(U_j - L_j\right)}{2NP} \tag{8}$$

for $i = 1, 2, 3, \ldots, NP$ and $j = 1, 2, 3, \ldots, D$, where ϕ_{ij} is a random number in $[-1, 1]$.

(2) *Employed bee phase*: The best position x_{best} is used to generate a new food source v_i by the following equation:

$$v_{i,j} = \begin{cases} x_{best,j} + \phi_{ij}\left(x_{i,j} - x_{k,j}\right); & j = j^*, \\ x_{i,j}; & j \neq j^*, \end{cases} \tag{9}$$

where k is randomly chosen from 1 to NP such that $k \neq i$, j^* is randomly chosen from 1 to D and ϕ_{ij} is a random number in $[-1, 1]$. If $f(v_i) < f(x_i)$, then replace x_i by v_i; otherwise, hold x_i and set $trial(i) = trial(i) + 1$, where $trial$ is the counter number of unimproved trials.

(3) *Onlooker bee phase*: The probability values used to make the decisions for onlooker bees are set to be a constant $p(i) = 0.25$, if $\text{rand}(0, 1) < p(i)$; then (9) is used to generate a new position v_i. If $f(v_i) < f(x_i)$, then replace x_i by v_i; otherwise, retain x_i and set $trial(i) = trial(i) + 1$. In addition, the worst positions are replaced by the new ones using the equation

$$x_{z_t} = M\left[x_{best} + \phi_t\left(x_{z_t} - x_{r_1}\right) + \omega_t\left(x_{best} - x_{r_2}\right)\right] \tag{10}$$

where z_t, $t = 1, 2, \ldots, \lfloor 0.05NP \rfloor$, are the indexes of 5% worst positions, r_1 and r_2 are the randomly chosen indexes from 1 to NP such that $r_1 \neq r_2 \neq z_t$ for all t, ϕ_t and ω_t are the random numbers in range of $[-1, 1]$, and M is the number of the best positions obtained from previous generation.

(4) *Scout bee phase*: Generate a new position for an unupdated position x_i of a scout bee by using the following Firefly algorithm strategy:

$$x_i = x_i + e^{-r_{iq}^2}\left(x_q - x_i\right) + (\text{rand}(0, 1) - 0.5) \tag{11}$$

where q is the first index such that $f(x_q) < f(x_i)$.

4. Experimental Results and Discussion

To compare the performance of MABC algorithm with those of other methods, three experiments are conducted using different settings and performance measurements. We select 8 benchmark functions consisting of 2 functions from one of 4 different types: unimodal and separable (US), unimodal and nonseparable (UN), multimodal and separable (MS), and multimodal and nonseparable (MN). Their descriptions and 2D surface plots are shown in Table 1 and Figure 1, respectively.

For the first experiment, we compare the performances of MABC, ABC [5], GABC [14], ABC/best/1, and ABC/best/2 [9]. As in [5, 9, 14], we set $NP = 40$, $limit = NP \times D$, and $MG = 5000$, where MG is the maximum number of generations. The MABC algorithm is performed 30 runs. Table 2 shows the mean and SD of f_{best} values of our MABC compared with those of ABC, GABC, and ABC/best/1 and ABC/best/2 as reported in [5, 9, 14]. The best values are indicated in bold, and the values less than 10^{-20} are reported as 0. In addition, a two-tailed t-test at a 0.05 level of significance is used to compare the performances of MABC with those of ABC, GABC, ABC/best/1, and ABC/best/2 in this order. The values "+", "0", and "–" denote that MABC performs significantly better than, similarly to, and worse than a compared method. The results show that MABC clearly outperforms ABC and GABC for almost all 10 cases. Compared to ABC/best/1 and ABC/best/2, MABC performs better for 3 cases and performs similarly for 6 cases, and there is only one case that it performs slightly worse than ABC/best/1.

For the second experiment, we compare the performances of MABC, ABC [5], and ABC-ROC [23]. The parameters are set the same as in [23] where $NP = 50$, $D = 30$, $MG = 5000$, and $limit = NP \times D$. The MABC algorithm is performed 30 runs. Table 3 presents the mean and SD of f_{best} values of MABC compared with those of ABC and ABC-ROC as reported in [23]. The best value for each function is highlighted in boldface. We use a two-tailed t-test at a significance level of 0.05 to compare the performances of MABC with ABC and ABC-ROC, respectively. The results show that MABC gives the best values for all 5 test functions and significantly outperforms both ABC and ABC-ROC for 2 cases and performs similarly for 3 cases.

Those two experiments use relatively low maximum number of generations (MG) when we consider the best values obtained for the Rosenbrock f_3 in high dimensions. To be able to solve f_3 function and to better compare the convergence performances of MABC and ABC, we conduct the third experiment by setting the more accurate value to reach VTR=10^{-10} and $MG = 2000 \times D$ which is related to dimension D. The dimensions are also varied as $D = 2, 5, 10, 30, 60$ while NP and $limit$ are set the same as the second experiment. Both algorithms are performed for 30 runs and the number of successful runs (NS), the mean of number of function evaluations (NF), and the percentage of standard deviations of the function evaluations (%SD) are reported in Table 4. The last column of the table shows the performances of MABC compared with ABC at a 0.05 level of significance. The values "+" and "0" denote that MABC

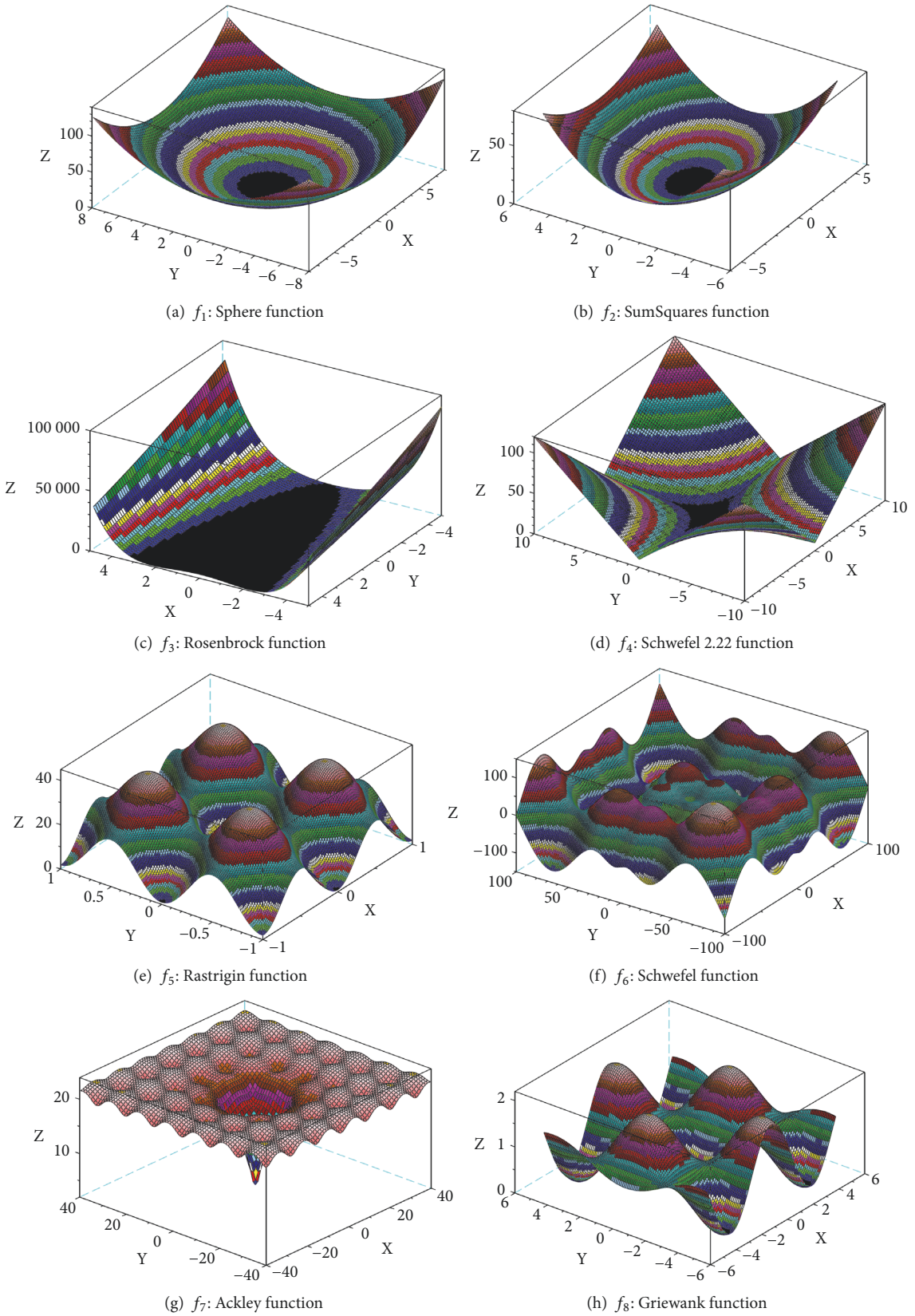
Table 1: Benchmark functions [8, 9].

Function	Formulation	Type	Search range	Optimum value
Sphere	$f_1(X) = \sum_{i=1}^{D} x_i^2$	US	$[-100, 100]^D$	0
SumSquares	$f_2(X) = \sum_{i=1}^{D} i x_i^2$	US	$[-10, 10]^D$	0
Rosenbrock	$f_3(X) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	UN	$[-30, 30]^D$	0
Schwefel 2.22	$f_4(X) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	UN	$[-10, 10]^D$	0
Rastrigin	$f_5(X) = 10D + \sum_{i=1}^{D} \left[x_i^2 - 10\cos\left(2\pi x_i\right) \right]$	MS	$[-5.12, 5.12]^D$	0
Schwefel	$f_6(X) = 418.9828872724339D - \sum_{i=1}^{D} x_i \sin\sqrt{(\lvert x_i \rvert)}$	MS	$[-500, 500]^D$	0
Ackley	$f_7(X) = -20\exp\left[-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2} \right] - \exp\left[\frac{1}{D}\sum_{i=1}^{D} \cos\left(2\pi x_i\right) \right] + 20 + e$	MN	$[-32, 32]^D$	0
Griewank	$f_8(X) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	MN	$[-600, 600]^D$	0

Table 2: Performance comparison of ABC, GABC, ABC/best/1, ABC/best/2, and MABC algorithms.

Functions	D	Statistics	ABC	GABC	ABC/best/1	ABC/best/2	MABC	Significant
f_1	30	Mean	6.38E-16	4.18E-16	**0**	**0**	**0**	+,+,0,0
		SD	1.20E-16	7.37E-17	0	0	0	
	60	Mean	2.78E-15	1.43E-15	**0**	**0**	**0**	+,+,0,0
		SD	3.18E-16	1.38E-16	0	0	0	
f_3	2	Mean	9.93E-03	1.68E-04	4.99E-04	4.42E-04	**3.85E-11**	+,+,+,+
		SD	8.14E-03	1.45E-04	5.22E-06	2.39E-04	1.24E-10	
	3	Mean	6.45E-02	2.66E-03	5.52E-06	9.90E-04	**2.84E-11**	+,+,+,+
		SD	4.85E-02	2.12E-03	3.03E-06	6.92E-04	6.27E-11	
f_5	30	Mean	1.34E-13	1.33E-15	**0**	**0**	**0**	+,0,0,0
		SD	7.97E-14	2.45E-14	0	0	0	
	60	Mean	2.06E-08	3.52E-13	**0**	**0**	**0**	0,+,0,0
		SD	1.12E-07	1.24E-13	0	0	0	
f_7	30	Mean	4.70E-14	3.22E-14	**1.72E-14**	2.50E-14	2.73E-14	+,+,−,0
		SD	5.95E-15	3.25E-15	2.84E-15	3.48E-15	1.76 E-14	
	60	Mean	1.66E-13	1.00E-13	6.62E-14	7.12E-14	**5.58E-14**	+,+,+,+
		SD	2.22E-14	6.09E-15	1.74E-15	4.14E-15	1.86E-14	
f_8	30	Mean	1.27E-15	2.96E-17	**0**	**0**	**0**	+,+,0,0
		SD	1.46E-15	4.99E-17	0	0	0	
	60	Mean	2.51E-13	7.55E-16	**0**	**0**	**0**	+,+,0,0
		SD	7.51E-13	4.13E-17	0	0	0	

performs significantly better than and similarly to ABC and the value "++" denotes that our MABC's solutions can reach VTR=10^{-10} while those of ABC cannot. The results show that MABC performs successfully for 30 runs for all test functions whereas ABC can perform successfully for almost all functions except f_3 for which it fails for all runs. Moreover, the NF values obtained by MABC are less than those obtained by ABC for all cases and our MABC significantly outperforms ABC for 33 out of 35 cases and performs similarly for 2 cases. For 5 cases of f_3, MABC's solutions can reach VTR=10^{-10} while those of ABC cannot. This indicates that MABC algorithm is more effective.



(a) f_1: Sphere function

(b) f_2: SumSquares function

(c) f_3: Rosenbrock function

(d) f_4: Schwefel 2.22 function

(e) f_5: Rastrigin function

(f) f_6: Schwefel function

(g) f_7: Ackley function

(h) f_8: Griewank function

Figure 1: 2*D* Surface plots of the test functions.

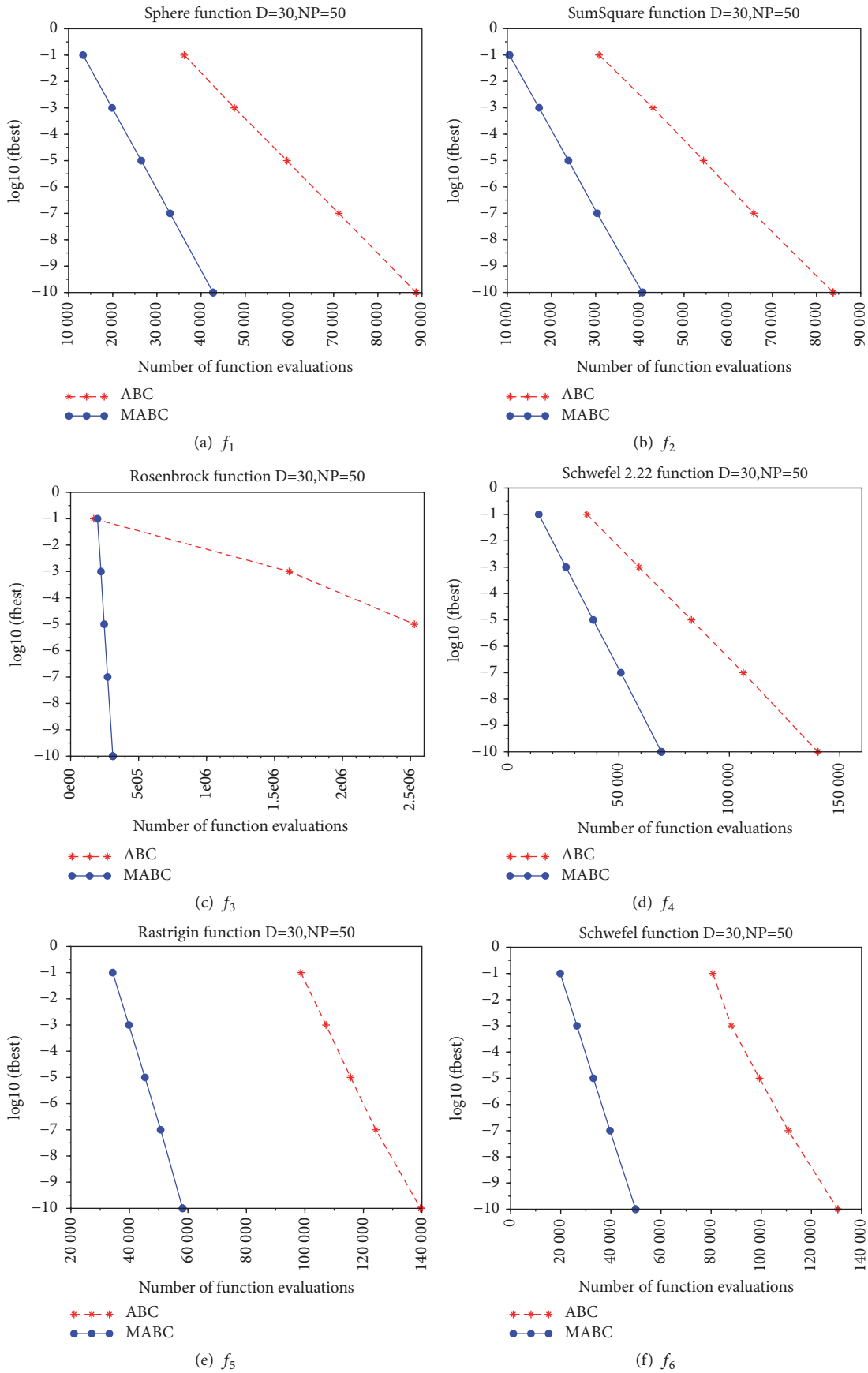
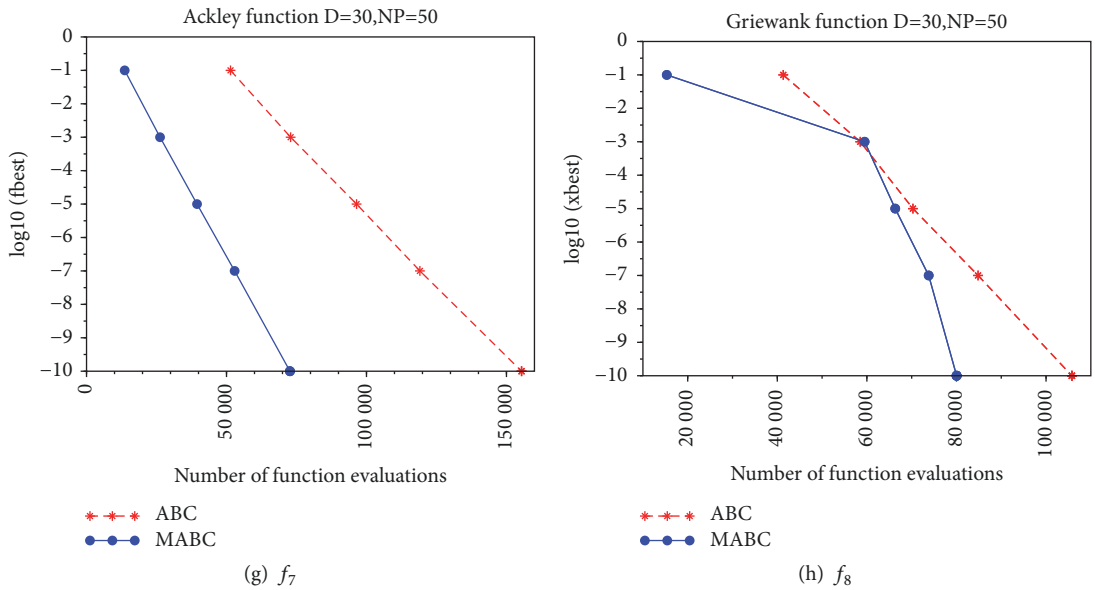


Figure 2: Continued.



(g) f_7 (h) f_8

Figure 2: Convergence graphs of ABC and MABC algorithms.

Table 3: Performance comparison of ABC, ABC-ROC, and MABC algorithms.

Functions	D	Statistics	ABC	ABC-ROC	MABC	Significant
f_1	30	Mean	**0**	4.76E-16	**0**	0,+
		SD	0	7.37E-17	0	
f_3	30	Mean	7.01E-02	3.93E-02	**6.08E-05**	+,+
		SD	4.15E-02	8.40E-04	1.59E-04	
f_5	30	Mean	**0**	**0**	**0**	0,0
		SD	0	0	0	
f_7	30	Mean	3.45E-14	3.27E-14	**2.73E-14**	0,0
		SD	4.46E-15	3.30E-15	7.89E-14	
f_8	30	Mean	6.66E-17	4.07E-17	**0**	+,+
		SD	1.49E-16	5.44E-17	0	

We also show the convergence graphs of MABC and ABC for $D = 30$ in Figure 2 where the NF values are plotted against their corresponding VTR=10^{-1}, 10^{-3}, 10^{-5}, 10^{-7}, and 10^{-10}. The graphs show that MABC converges faster than ABC for all test functions.

5. Conclusions

We have presented an efficient modification of ABC algorithm called MABC which improves all phases of ABC algorithm: the initial population is generated by using search space division, the new search equations are used in employed bee and onlooker bee phases, the opportunity of onlooker bee to find better solutions is increased and some worst positions are replaced by the new ones depending on the best position, and the unupdated positions are improved in scout bee phase. Extensive experiments show that MABC converges faster than ABC and it performs better than ABC and several previously proposed modification methods of ABC.

Data Availability

The data used to support the findings of this current study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

Amnat Panniem acknowledges the Development and Promotion for Science and Technology talents project (DPST) for the financial support.

Table 4: Performance comparison of ABC and MABC algorithms using NS, NF, and %SD values and $VTR = 10^{-10}$.

Functions	D	ABC			MABC			Significant
		NS	NF	%SD	NS	NF	%SD	
f_1	2	30	4051	7.60	30	1878	5.38	+
	5	30	12010	3.83	30	6079	3.54	+
	10	30	26356	4.15	30	13608	2.49	+
	30	30	87678	2.65	30	43695	2.75	+
	60	30	188597	1.98	30	124882	1.66	+
f_2	2	30	3427	4.64	30	1588	0.06	+
	5	30	10611	4.79	30	5203	11.02	+
	10	30	23874	3.22	30	12340	6.26	+
	30	30	83755	6.78	30	40377	4.89	+
	60	30	180206	5.94	30	125596	17.04	+
f_3	2	0	-	-	30	31723	7.22	++
	5	0	-	-	30	31487	46.42	++
	10	0	-	-	30	65308	51.11	++
	30	0	-	-	30	310197	50.90	++
	60	0	-	-	30	1897652	21.86	++
f_4	2	30	6938	0.02	30	3197	0.05	+
	5	30	19854	9.64	30	9716	5.47	+
	10	30	43100	4.23	30	22187	4.12	+
	30	30	140991	4.52	30	69473	3.92	+
	60	30	293036	1.41	30	188176	4.09	+
f_5	2	30	5962	9.46	30	2234	8.10	+
	5	30	18047	4.26	30	7310	5.95	+
	10	30	40131	4.73	30	17175	7.47	+
	30	30	140089	4.68	30	59496	19.55	+
	60	30	323945	6.17	30	177682	20.57	+
f_6	2	30	6173	9.65	30	2129	6.58	+
	5	30	17836	6.30	30	6969	3.66	+
	10	30	40015	4.54	30	15611	2.63	+
	30	30	132011	3.91	30	49996	3.05	+
	60	30	290327	5.64	30	139990	4.85	+
f_7	2	30	8292	4.99	30	3551	2.56	+
	5	30	22948	2.99	30	10611	2.70	+
	10	30	48612	1.86	30	23381	1.89	+
	30	30	154908	1.35	30	72955	2.47	+
	60	30	318003	1.34	30	217442	4.05	+
f_8	2	30	8033	11.14	30	5985	37.89	+
	5	30	47947	52.66	30	41999	73.14	0
	10	30	95032	67.34	30	63610	39.31	+
	30	30	111795	14.40	30	76601	20.79	+
	60	30	207006	4.27	30	201120	15.46	0

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, pp. 1942–1948, Perth, Western Australia, November-December 1995.

[2] D. T. Pham, A. Ghanbarzadeh, E. Koç, and S. Otri, *The Bees Algorithm Technical Note*, Manufacturing Engineering Centre, Cardiff university, UK, 2005.

[3] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*, vol. 5792 of *Lecture Notes in Computer Science*, pp. 169–178, Springer, Berlin, Germany, 2009.

[4] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, J. R. Gonzalez, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds., vol. 284 of *Studies in Computational Intelligence*, pp. 65–74, Springer, Berlin, Germany, 2010.

[5] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report-TR06, Erciyes University, Kayseri, Turkey, 2005.

[6] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[7] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing*, vol. 8, no. 1, pp. 687–697, 2008.

[8] D. Karaboga and B. Akay, "A comparative study of artificial Bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.

[9] W. Gao, S. Liu, and L. Huang, "A global best artificial bee colony algorithm for global optimization," *Journal of Computational and Applied Mathematics*, vol. 236, no. 11, pp. 2741–2753, 2012.

[10] B. Li, L. G. Gong, and Y. Li, "A novel artificial bee colony algorithm based on internal-feedback strategy for image template matching," *The Scientific World Journal*, vol. 2014, Article ID 906861, 14 pages, 2014.

[11] Y. Cheng, "Modified ABC algorithm in virus evolution," in *Proceedings of the 2016 IEEE International Symposium on Computer, Consumer and Control, IS3C 2016*, pp. 805–808, China, July 2016.

[12] F. Kang, J. Li, and Q. Xu, "Structural inverse analysis by hybrid simplex artificial bee colony algorithms," *Computers & Structures*, vol. 87, no. 13-14, pp. 861–870, 2009.

[13] W. Zou, Y. Zhu, H. Chen, and X. Sui, "A clustering approach using cooperative artificial bee colony algorithm," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 459796, 16 pages, 2010.

[14] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166–3173, 2010.

[15] G. Li, P. Niu, and X. Xiao, "Development and investigation of efficient artificial bee colony algorithm for numerical function optimization," *Applied Soft Computing*, vol. 12, no. 1, pp. 320–332, 2012.

[16] F. Kang, J. J. Li, and H. J. Li, "Artificial bee colony algorithm and pattern search hybridized for global optimization," *Applied Soft Computing*, vol. 13, no. 4, pp. 1781–1791, 2013.

[17] X. Kong, S. Liu, and Z. Wang, "An Improved Artificial Bee Colony Algorithm and Its Application," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 6, no. 6, pp. 259–274, 2013.

[18] M. Alam, M. Islam, and K. Murase, "Artificial bee colony algorithm with improved exploration for numerical function optimization," in *Proceeding of the 13th International Conference Intelligent Data Engineering and Automated Learning*, pp. 1–8, Natal, Brazil, 2011.

[19] N. Sulaiman, J. Mohamad-Saleh, and A. G. Abro, "A modified artificial bee colony (JA-ABC) optimization algorithm," in *Proceedings of the 2013 International Conference on Applied Mathematics and Computational Methods in Engineering*, pp. 74–79, Rhodes Island, Greece, 2013.

[20] W. Liu, "A Multistrategy Optimization Improved Artificial Bee Colony Algorithm," *The Scientific World Journal*, vol. 2014, Article ID 129483, 10 pages, 2014.

[21] Y. Gaowei and L. Chuangqin, "An effective refinement Artificial Bee Colony optimization algorithm based on chaotic search and application for PID control tuning," *Journal of Computational Information Systems*, vol. 7, no. 9, pp. 3309–3316, 2011.

[22] S. Shapla, H. Haque, and M. Alam, "Explorative artificial bee colony algorithm: A novel swarm intelligence based algorithm for continuous function optimization," *International Journal of Science and Research*, vol. 4, no. 7, pp. 1339–1344, 2015.

[23] S. Anuar, A. Selamat, and R. Sallehuddin, "A modified scout bee for artificial bee colony algorithm and its performance on optimization problems," *Journal of King Saud University - Computer and Information Sciences*, vol. 28, no. 4, pp. 395–406, 2016.

[24] Z. He, C. Ma, X. Wang et al., "A Modified Artificial Bee Colony Algorithm Based on Search Space Division and Disruptive Selection Strategy," *Mathematical Problems in Engineering*, vol. 2014, Article ID 432654, 14 pages, 2014.