

An enhanced artificial bee colony algorithm based on fitness weighted search strategy

Yuksel Celik

To cite this article: Yuksel Celik (2021) An enhanced artificial bee colony algorithm based on fitness weighted search strategy, *Automatika*, 62:3-4, 300-310, DOI: [10.1080/00051144.2021.1938477](https://doi.org/10.1080/00051144.2021.1938477)

To link to this article: <https://doi.org/10.1080/00051144.2021.1938477>



© 2021 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 30 Jun 2021.



Submit your article to this journal [↗](#)



Article views: 999



View related articles [↗](#)



View Crossmark data [↗](#)



An enhanced artificial bee colony algorithm based on fitness weighted search strategy

Yuksel Celik

Faculty of Engineering, Department of Computer Engineering, Karabuk University, Karabuk, Turkey

ABSTRACT

Artificial Bee Colony (ABC) algorithm is a meta-heuristic algorithm, is inspired by the bee's food search behaviour based on swarm intelligence. Successful applications were performed on many optimization problems using this algorithm rising in popularity over the past few years. The update mechanism of the ABC algorithm, despite the fact that its exploration is good, faces the problem of convergence performance. For solving convergence problem of ABC Algorithm An Artificial Bee Colony Algorithm Based Fitness Weighted Search (ABCFWS) algorithm proposed in this paper. In this approach, an intelligent search space is proposed instead of the random search space of the ABC algorithm. In this method, the fitness values of the food source and the selected neighbour food source are taken as weights and a more balanced search space was found in the direction of the food source with better fitness value. The proposed method has been applied to 28 unconstrained numerical optimization test problems with different characteristics and the results were compared with the ABC algorithm variations. The results show that the proposed method is successful and competitive.

ARTICLE HISTORY

Received 16 January 2021
Accepted 31 May 2021

KEYWORDS

Artificial bee colony;
meta-heuristic optimization;
continuous optimization

1. Introduction

The artificial bee colony (ABC) algorithm has been developed in 2005 by Dervis Karaboga who inspired by bee's food search behaviour [1]. So far, the ABC algorithm has been tested on several optimization problems and successful results have been obtained [2–5]. In addition to optimization problems, the ABC algorithm is used for improving image contrast in the field of image processing [6], increasing the performance of optical CDMA networks [7], clustering of large-scale data [8], large scale data clustering [9], optimizing network attacks in cyber security [10], minimizing of network attacks in the field of cybersecurity [11], solving the vehicle routing problem [11], estimating the multi-purpose optimal power flow [12], training of artificial neural networks [13] and routing in wireless sensor networks [14].

While the search strategy of the ABC algorithm has a good exploration capability [15], its exploitation capability causes a weak convergence problem in this algorithm as in other evolutionary based algorithms [16]. The random method used to find a new food source in the ABC algorithm causes uncertainty whether the search direction is a good or a bad direction. This situation leads to instability in the search strategy of the ABC [17]. Until today, a lot of improvements have been done to improve the performance of the ABC algorithm. These improvements can basically

be considered under two main groups; the first is the hybridization with the other algorithms, the second is to develop a new search strategy or to improve the existing search strategies [17,18].

Some of the hybrid methods based on the ABC algorithm are Differential Evolution (DE) [19], Neuro-Fuzzy System [20], Extended Genetic Programming [21], Particle Swarm Optimization algorithm [22], Harmony Search Algorithm [23], Simulated Annealing algorithm [24], Quantum Evolutional Algorithm [25], Bacterial foraging optimization algorithm [26], search strategy [27].

An important factor in the success of the ABC algorithm is neighbour selection when developing a search strategy. There are fitness-based neighbour and distance-based neighbour selection mechanisms [16]. In the fitness-based method, individuals with better fitness are more likely to be selected. In the distance-based method, the individuals that are nearer to the parent individual are more likely to be selected. The performance of the ABC is improved using both of them [16]. In order to investigate the neighbourhood structure of the swarm system and the diffusing mechanism in the ABC algorithm, the search strategy of the algorithm is improved by adapting the basic components of Cellular Automata [17]. In another study that improves the search capability of the ABC algorithm, the bees are placed on the nodes of the scale-free

network in each iteration, the bees with a high-quality food source are placed on high nodes and the bees with low-quality are placed on low nodes. Then, the probability of connecting low-quality bees to high-quality bees is increased by the power law distribution property which is applied to nodes. In this way, each individual is forced to receive information from its neighbours and therefore, the convergence speed of the algorithm and exploitation ability are increased [28].

The latest research of the ABC algorithm; Kiran and et al. [29] suggested a method that selects the best update method and a learning strategy according to the optimization problem. Babaoglu [30] produced new candidate solutions using the mean and standard deviation values of two food sources selected in the food source update procedure. Gao et al. [31] proposed a method of generating new candidate solutions, which was inspired by the differential evolution algorithm, by allowing each bee to search around the global best in the previous iteration. Yang et al. [15] proposed an adaptive encoding learning (AEL) method that learns the distribution of information in the current population and produces a new candidate solution accordingly. In this method, besides the existing coordinate system of the ABC algorithm, they used eigen coordinate system guided by a covariance matrix learning strategy. A coordinate system is selected based on a probability vector that determines the selection rate. An adaptive selection mechanism based on the success rate of each coordinate system in each learning phase of the method is improved by balancing the exploration and exploitation abilities of the AEL.

In this paper, in order to eliminate the negativity of the uncertain search direction of the ABC, we propose a method for creating a centre of weight by taking into account the fitness values of the current food source and the neighbour food source. This method based on the centre of weight causes the search direction to move towards the food source with better fitness value. The proposed method provides to search in a good region of the solution space. The test results show that the proposed method is more successful than the variants of the ABC.

The remainder of the paper is organized as follows: In Section 2, we give a brief overview of the original ABC algorithm. We propose an Improved ABC Algorithm Based on Fitness Weighted Search (ABC-FWS) in Section 3. We introduce test functions in Section 4, give some computational results and analysis of our comparative study in Section 5. Finally, in Section 6, we present our conclusions.

2. Artificial Bee Colony Algorithm

The ABC algorithm is inspired by the intelligent swarming behaviour of honey bees in the foraging processes of natural life. In this algorithm, there are three

types of bees; employed bees, onlooker bees and scout bees. Employed and onlooker bees are equal in number. One half of the colony comprises employed bees and the other half consists of the onlooker bees. Each bee represents a food source, the location of a food source represents a candidate solution to the optimization problem and the amount of nectar indicates the fitness value of the corresponding solution.

The initial values of the ABC algorithm are a population P as well as SN individuals in the D -dimensional search space. The initial values of an individual $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ which represents a food source are generated using Equation (1). New individuals in the subsequent search strategy are constructed using Equation (2).

$$x_{i,j} = x_{min,j} + rand(0,1)(x_{max,j} - x_{min,j}) \quad (1)$$

$$v_{i,j} = x_{i,j} + \emptyset_{i,j}(x_{i,j} - x_{k,j}), \quad (2)$$

where individuals $x_{min,j}$ and $x_{max,j}$ are the lower value and the upper value of the generated solution set, respectively, $\emptyset_{i,j}$ is a random number in $[-1,1]$, $x_{k,j}$ is randomly selected neighbour and $j \in 1, 2, \dots, D$, $k \in 1, 2, \dots, SN$. Afterward, by a greedy selection according to the quality of the nectar amount, the best solution is chosen from the previous and current solutions. The update process of the onlooker bees is the same as the employer bees. However, the main difference between them is based on the choice of potential food sources according to the probabilities calculated by the fitness values. The probability value p_i of an onlooker individual is calculated using Equations (3) and (4).

$$p_i = fit_i / \sum_{i=1}^{SN} fit_i \quad (3)$$

$$fit_i = \begin{cases} \frac{1}{1+|f_i|}, & \text{if } f_i \geq 0 \\ |f_i|, & \text{if } f_i < 0 \end{cases}, \quad (4)$$

where f_i is the objective function value. If a position cannot be further developed in a predetermined number of loops, then it is assumed that the food source has been left (abandoned). The value of the predetermined number of loops is an important control parameter called the abandonment limit in the ABC algorithm. Assuming that the abandoned food source is X_i , then the scout bee as X_i produces a random (Equation (1)) food source.

3. Proposed approach for the ABC algorithm

Although the ABC algorithm has a strong global search capability, it suffers from a weak convergence rate and loses the balance between exploration and exploitation in the search strategy [17]. Once the initial population of the algorithm is constructed by Equation (1), a new solution is obtained by Equation (2). If improvement

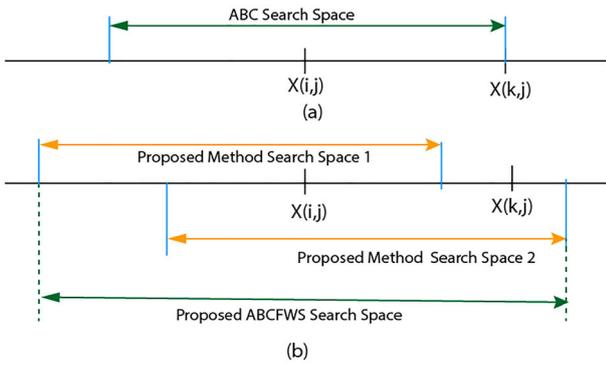


Figure 1. (a) The ABC search space (b) proposed ABCFWS search spaces.

is achieved with the new solution, then the algorithm moves towards the optimal solution according to the quality of the solution otherwise it will not progress. The choice of neighbour $x_{k,j}$ in Equation (2) plays an important role in the success of the ABC algorithm. However, a random selection of neighbour $x_{k,j}$, regardless of its success, has a negative effect on the overall success. In order to eliminate this negativity, the ABC algorithm uses a new search space instead of the original one that is obtained using the fitness values of the updated bee and its neighbour (Figure 1).

In the ABC algorithm, in order to move towards the good food source, a new value is obtained between the value of the randomly selected neighbour and the value of the current position. There is no control over whether the direction of the new value is good or bad, that is, the search strategy selects a random direction.

In this study, the search space is made more balanced by considering the fitness values of the solution for both food sources.

Figure 1(a) shows the original search space of the ABC algorithm. As can be seen in Figure 1(b), in the proposed method, we combined two search spaces, each of length the ABC search space, in two different directions which intersect with each other. It is easily seen that the ABCFWS search space is larger than the ABC search space. Moreover, it is shown in Figures 2 and 3 that the search space searches in the specified space due to the iterations. In the proposed method, a new value is obtained by considering the weight of the fitness value as in Equation (5). In Equal 5 if the value of the current fitness (Fit_i) is better than the randomly selected neighbour Fit_k fitness, the direction of the update function is directed to the current solution. If not, the direction of the update function is directed to the randomly selected neighbour. The new value is updated as in Equation (6) in a manner similar to the original ABC algorithm.

$$W_{ij} = \begin{cases} \frac{(Fit_i * x_{i,j}) - (Fit_k * x_{k,j})}{Fit_i + Fit_k}, & \text{if } Fit_i > Fit_k \\ \frac{(Fit_i * x_{i,j}) + (Fit_k * x_{k,j})}{Fit_i + Fit_k}, & \text{if otherwise} \end{cases} \quad (5)$$

where i is the index of the food source, j is a randomly selected value of the D -dimensional solution set and k is the randomly selected neighbour's index. Accordingly, W_{ij} is the new weight value obtained for the



Figure 2. Fitness values of updated method.



Figure 3. Updated values of the algorithms.

Table 1. Unconstrained benchmark functions used in computational tests.

Nr	Name	Interval	C	Function
F1	Sphere	$[-100, 100]$	US	$f_1(\vec{X}) = \sum_{i=1}^n x_i^2$
F2	Elliptic	$[-100, 100]$	UN	$f_2(\vec{X}) = \sum_{i=1}^n (10^6)^{(i-1)/(n-1)} x_i^2$
F3	SumSquares	$[-10, 10]$	US	$f_3(\vec{X}) = \sum_{i=1}^n ix_i^2$
F4	SumPower	$[-10, 10]$	MS	$f_3(\vec{X}) = \sum_{i=1}^n ix_i^2$
F5	Schwefel2.22	$[-10, 10]$	UN	$f_5(\vec{X}) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
F6	Schwefel2.21	$[-100, 100]$	UN	$f_6(\vec{X}) = \max_{1 \leq i \leq n} x_i $
F7	Step	$[-100, 100]$	US	$f_7(\vec{X}) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$
F8	Quartic	$[-1.28, 1.28]$	US	$f_8(\vec{X}) = \sum_{i=1}^n ix_i^4$
F9	QuarticWN	$[-1.28, 1.28]$	US	$f_9(\vec{X}) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1]$
F10	Rosenbrock	$[-10, 10]$	UN	$f_{10}(\vec{X}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
F11	Rastrigin	$[-5.12, 5.12]$	MS	$f_{11}(\vec{X}) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$
F12	Non-continuous rastrigin	$[-5.12, 5.12]$	MS	$f_{12}(\vec{X}) = \sum_{i=1}^n [y_i^2 - 10\cos(2\pi y_i) + 10]$ $y_i = \begin{cases} x_i x_i < \frac{1}{2} \\ \frac{\text{round}(2x_i)}{2} x_i \geq \frac{1}{2} \end{cases}$
F13	Griewank	$[-600, 600]$	MN	$f_{13}(\vec{X}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F14	Schwefel2.26	$[-500, 500]$	UN	$f_{14}(\vec{X}) = 418.98 * n - \sum_{i=1}^n x_i \sin(\sqrt{ x_i })$
F15	Ackley	$[-32, 32]$	MN	$f_{15}(\vec{X}) = -20 \exp\left\{-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right\} - \exp\left\{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right\} + 20 + e$
F16	Penalized1	$[-50, 50]$	MN	$f_{16}(\vec{X}) = \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\}$ $+ \sum_{i=1}^n u(x_i, 10, 100, 4) y_i = 1 + \frac{1}{4} (x_i + 1) u_{x_i, a, k, m} = \begin{cases} k(x_i - a)^m x_i > a \\ 0 - a \leq x_i \leq a \\ k(x_i - a)^m x_i < -a \end{cases}$
F17	Penalized2	$[-50, 50]$	MN	$f_{17}(\vec{X}) = \frac{1}{10} \left\{ \sin^2(\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_{i+1})] \right\}$ $+ \sum_{i=1}^n u(x_i, 5, 100, 4)$
F18	Alpine	$[-10, 10]$	MS	$f_{18}(\vec{X}) = \sum_{i=1}^n x_i \cdot \sin(x_i) + 0.1 \cdot x_i $
F19	Levy	$[-10, 10]$	MN	$f_{19}(\vec{X}) = \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + x_n - 1 [1 + \sin^2(3\pi x_n)]$
F20	Weierstrass	$[-0.5, 0.5]$	MN	$f_{20}(\vec{X}) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right)$ $-D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k 0.5)] a = 0.5, b = 3, k_{\max} = 20$
F21	Schaffer	$[-100, 100]$	MN	$f_{20}(\vec{X}) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right)$ $-D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k 0.5)] a = 0.5, b = 3, k_{\max} = 20$
F22	Himmelblau	$[-5, 5]$	MS	$f_{22}(\vec{X}) = \frac{1}{n} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$
F23	Michalewicz	$[0, \pi]$	MS	$f_{23}(\vec{X}) = -\sum_{i=1}^n \sin(x_i) \sin^{20}\left(\frac{ix_i^2}{\pi}\right)$

(continued).

Table 1. Continued.

Nr	Name	Interval	C	Function
F24	Shifted sphere	[-100, 100]	US	$f_{24}(\vec{X}) = \sum_{i=1}^n z_i^2 z = x - o$
F25	Shifted rastrigin	[-5.12, 5.12]	MS	$f_{25}(\vec{X}) = \sum_{i=1}^n [z_i^2 - 10\cos(2\pi z_i) + 10]z = x - o$
F26	Shifted griewank	[-600, 600]	MN	$f_{26}(\vec{X}) = \frac{1}{4000} \sum_{i=1}^n z_i^2 - \prod_{i=1}^n \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1z = x - o$
F27	Shifted ackley	[-32, 32]	MN	$f_{27}(\vec{X}) = -20\exp\left\{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n z_i^2}\right\} - \exp\left\{\frac{1}{n}\sum_{i=1}^n \cos(2\pi z_i)\right\}$ $z = x - o$
F28	Shifted alpine	[-10, 10]	MN	$f_{28}(\vec{X}) = \sum_{i=1}^n z_i \cdot \sin(z_i) + 0.1 \cdot z_i z = x - o$

proposed method, Fit_i is the fitness value of the current solution and Fit_k is the fitness value of the neighbour solution which is chosen randomly. The new weight value obtained by Equation (5) is used in Equation (6) in the ABC algorithm.

$$v_{i,j} = W_{i,j} + \emptyset_{i,j}(x_{i,j} - x_{k,j}) \quad (6)$$

Here, by replacing the current value $x_{i,j}$ in Equation (2) of the original ABC algorithm with the weighted $W_{i,j}$ value, the search strategy becomes balanced and able to search better. The fitness values used in the weighting of the proposed method for each iteration are shown in Figure 2. A comparison of the new values obtained according to these fitness weights with the old values is given in Figure 3.

It is seen in Figures 2 and 3 that the proposed method obtains more balanced values in the intervals indicated in Figure 1 than the ABC algorithm.

Figure 2 shows the fitness values of the food source of the selected neighbour to be updated. Figure 3 shows the new values obtained by the original ABC algorithm and the proposed method depending on the same fitness value. When Figure 3 is examined, it can be seen that the new values obtained by the proposed method are closer to the value of the good food source. This principle is the key success factor of the proposed method.

4. Benchmark functions

In order to investigate the convergence and analyse the performance of the ABCFWS algorithm, 28 unconstrained numerical optimization test functions which are commonly used in the literature are given in Table 1 [29]. The selected problems have unimodal (U), multimodal (M) or separable (S) and non-separable (N) properties as indicated in column C in Table 1 so that they can test different properties of the algorithms. As additional information, the shifted vector in the functions F24-F28 is generated randomly in the search intervals of the problems.

5. Experimental results and discussion

5.1. Experimental settings

The proposed ABCFWS algorithm has been tested on 28 test functions. We perform 30 independent runs for each instance to get reliable statistical results. The dimensions of the test functions other than F22 and F23 are 30, 60 and 100. For functions F22 and F23, the dimensions are set to 100, 200 and 300.

In all experimental tests and comparisons, the swarm size is chosen as $NP = 40$, the number of employee bees or food sources is chosen as half of the swarm, i.e. $SN = NP / 2$. Scout bees are chosen as the other half of the swarm.

The value of abandonment limit is taken as $limit = D * SN$ in the ABC, GABC and ABCVSS algorithms and as $limit = 0.6 * D * N$ in the ABC/Best/2 algorithm [31].

For a fair comparison, the parameter maxFEs (maximum number of the function evaluations) is set to 150,000, 300,000 and 500,000 for the functions with $D = 30$, $D = 60$ and $D = 100$, respectively.

5.2. Experimental results

In addition to the standard deviation and average performance data which are used to investigate the accuracy and robustness of the solutions obtained in optimization algorithms, convergence graphs are an important metric for the comparison of algorithms in the computational test.

The comparison of the proposed ABCFWS algorithm with the original ABC algorithm and the ABCVSS algorithm [29] in terms of their evolutionary processes and convergence characteristics for the first 5000 iterations of the solutions on the 15 test problems is given in Figure 4.

According to Figure 4, the proposed ABCFWS algorithm yields convergence rates worse than the ABCVSS only for F4, worse than the ABC and the ABCVSS for F10 and worse than the ABC for F14. When we analyse the convergence graphs, we can easily see that the proposed ABCFWS algorithm converges more quickly for 12 of the problems. Considering these

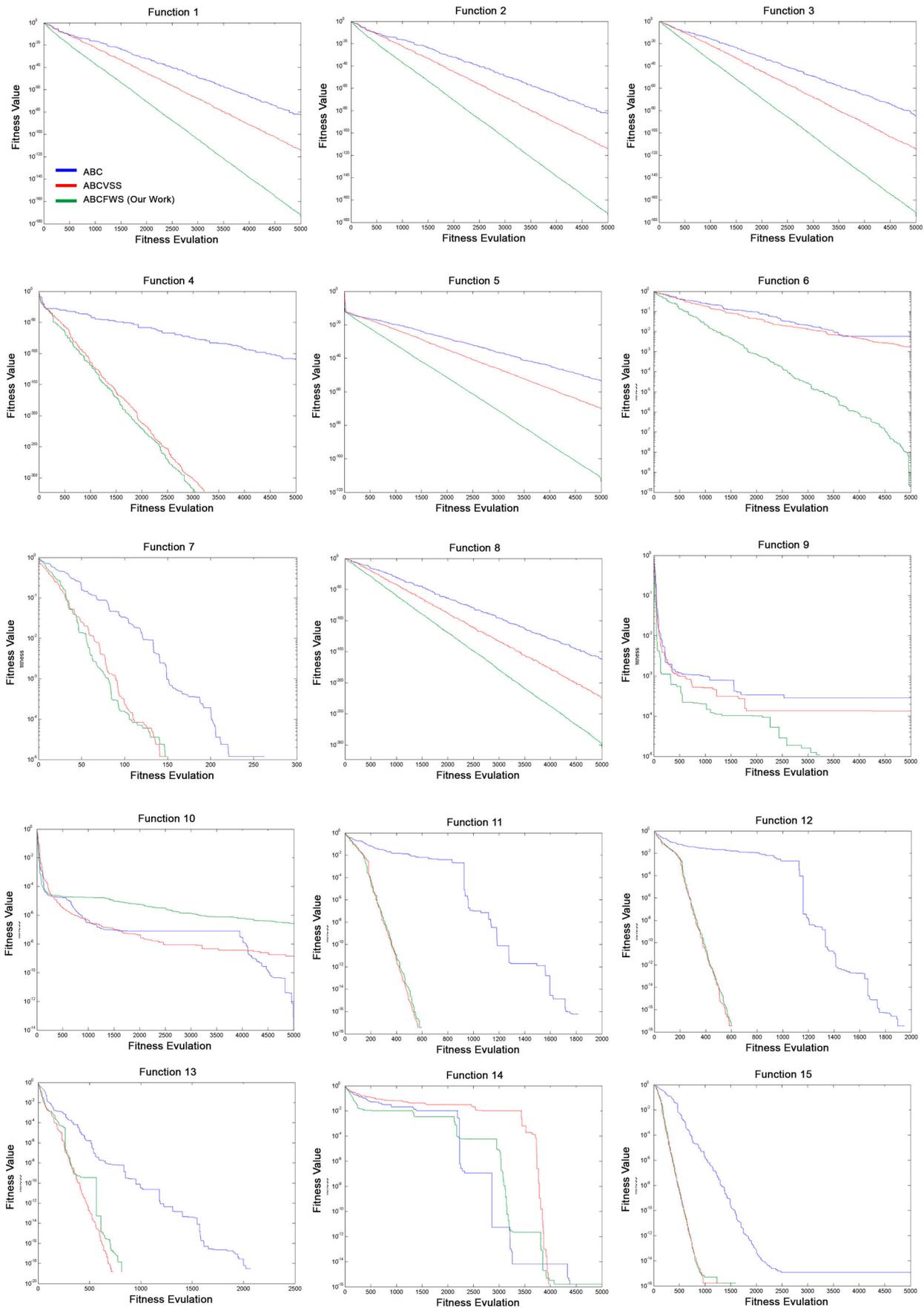


Figure 4. The convergence performance of the algorithms on the first 15 test functions with 30 dimension.

convergence graphs in general, it is seen that the proposed algorithm obtains a better convergence rate than the other algorithms.

In addition to the convergence graphs, mean and standard deviation values of 30 independent experimental test results are obtained to analyse the

Table 2. Comparison of the basic ABC variants on the 30 and 100 (F22, F23) dimensional function.

Function	ABC		GABC		ABCBest2		MABC		ABCVSS		ABCFWS	
	Mean	Std.Dev.										
F1	5,10E-16	8,40E-17	4,62E-16	7,12E-17	5,96E-35	3,61E-35	9,43E-32	6,67E-32	1,53E-81	8,37E-81	1,2E-123	1,4E-123
F2	4,79E-16	9,88E-17	3,62E-16	7,88E-17	1,70E-28	2,35E-28	3,66E-28	5,96E-28	4,82E-82	2,63E-81	8,33E-119	9,87E-119
F3	5,06E-16	9,20E-17	4,55E-16	7,00E-17	5,55E-36	3,36E-36	2,1E-32	1,56E-32	3,19E-89	1,48E-88	1,22E-124	1,32E-124
F4	2,85E-17	9,69E-18	1,64E-17	8,07E-18	3,00E-46	1,07E-45	2,7E-69	5,38E-69	5,55E-115	3,04E-114	8,57E-129	2,41E-128
F5	1,28E-15	1,44E-16	1,35E-15	1,36E-16	1,36E-18	4,27E-19	2,4E-17	9,02E-18	7,89E-43	4,32E-42	8,24E-65	7,04E-65
F6	7,27E-01	3,25E-01	2,18E-01	4,01E-02	3,55E+00	4,79E-01	1,02E+01	1,49E+00	4,08E-02	2,20E-02	1,85E-03	2,53E-04
F7	0	0	0	0	0	0	0	0	0	0	0	0
F8	2,01E-16	4,74E-17	1,21E-16	3,99E-17	3,1E-76	2,89E-76	1,45E-67	2,28E-67	3,25E-154	1,78E-153	3,89E-246	0
F9	4,86E-02	1,49E-02	2,03E-02	5,74E-03	0,0253	4,67E-03	0,0371	8,53E-03	1,81E-02	5,27E-03	1,54E-02	3,85E-03
F10	4,32E-02	4,71E-02	3,21E-01	8,21E-01	5,45E+00	8,40E+00	0,611	4,55E-01	3,87E-01	1,54E+00	2,89E+00	2,02E+00
F11	0	0	0	0	0	0	0	0	0	0	0	0
F12	0	0	0	0	0	0	0	0	0	0	0	0
F13	7,62E-11	4,18E-10	3,70E-17	5,32E-17	1,81E-08	6,29E-08	0	0	0	0	0	0
F14	1,09E-12	9,06E-13	9,42E-02	5,16E-01	1,76E-12	3,32E-13	1,21E-13	4,53E-13	4,85E-13	8,18E-13	7,07E-06	2,38E-05
F15	3,79E-14	3,99E-15	3,20E-14	3,36E-15	3,07E-14	3,43E-15	4,13E-14	2,17E-15	2,45E-14	4,00E-15	2,38E-14	3,38E-15
F16	5,08E-16	5,15E-17	4,12E-16	8,36E-17	1,57E-32	5,57E-48	1,9E-32	3,70E-33	1,57E-32	5,57E-48	1,57E-32	5,57E-48
F17	4,88E-16	7,45E-17	4,01E-16	8,19E-17	1,35E-32	5,57E-48	2,23E-31	1,46E-31	1,35E-32	5,57E-48	1,35E-32	5,57E-48
F18	8,82E-10	2,19E-09	3,41E-09	1,13E-08	3,23E-14	9,14E-14	1,58E-16	2,48E-16	3,66E-44	1,93E-43	8,43E-35	2,52E-34
F19	4,21E-16	8,31E-17	3,28E-16	5,03E-17	1,35E-31	6,68E-47	1,48E-31	2,30E-32	1,35E-31	6,68E-47	1,35E-31	6,68E-47
F20	0	0	0	0	9,47E-16	2,46E-15	0	0	0	0	0	0
F21	3,27E-01	4,44E-02	2,66E-01	4,39E-02	0,281	3,92E-02	0,295	3,17E-02	2,84E-01	5,69E-02	1,81E-01	3,72E-02
F22	-7,83E-01	1,02E-10	-7,83E+01	2,94E-14	-7,83E+01	4,86E-09	-7,83E+01	2,06E-07	-7,83E+01	3,02E-10	-7,83E+01	8,25E-14
F23	-9,63E-01	4,42E-01	-9,94E+01	4,18E-02	-8,94E+01	4,72E-01	-9,07E+01	5,03E-01	-9,94E+01	8,84E-02	-9,42E+01	2,42E-01
F24	4,91E-16	7,25E-17	4,38E-16	8,43E-17	0	0	0	0	0	0	0	0
F25	0	0	0	0	0	0	0	0	0	0	0	0
F26	8,17E-10	4,47E-09	3,33E-17	5,17E-17	1,46E-07	7,78E-07	0	0	0	0	1,54E-04	2,57E-04
F27	3,73E-14	4,45E-15	3,20E-14	2,80E-15	3,01E-14	3,70E-15	4,92E-14	5,31E-15	2,53E-14	4,55E-15	1,61E-14	2,23E-15
F28	1,35E-09	3,48E-09	6,65E-08	2,39E-07	1,33E-13	4,89E-13	1,38E-16	8,11E-17	7,49E-17	1,48E-17	8,30E-06	5,84E-06

Table 3. Comparison of the basic ABC variants on the 60 and 200 (F22, F23) dimensional functions.

Func.	Original ABC		GABC		ABCBest2		MABC		ABCVSS		ABCFS	
	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std,Dev,	Mean	Std.Dev.	Mean	Std.Dev.
F1	1,24E-15	1,16E-16	1,06E-15	1,21E-16	4,82E-33	2,59E-33	6,03E-29	4,31E-29	1,09E-83	5,01E-83	1,29E-122	1,67E-122
F2	1,15E-15	1,50E-16	8,97E-16	9,29E-17	5,86E-27	1,13E-26	3,51E-25	2,72E-25	1,01E-82	5,54E-82	1,54E-117	1,64E-117
F3	1,21E-15	1,59E-16	1,04E-15	1,27E-16	9,10E-34	3,87E-34	1,39E-29	8,84E-30	8,17E-86	4,47E-85	7,80E-123	9,24E-123
F4	4,31E-17	1,47E-17	2,85E-17	1,01E-17	7,53E-39	3,95E-38	3,00E-62	3,87E-62	1,59E-118	8,70E-118	3,89E-127	9,17E-127
F5	2,80E-15	2,40E-16	2,96E-15	1,85E-16	1,58E-17	3,32E-18	6,96E-16	1,20E-16	1,47E-45	7,00E-45	4,74E-64	3,01E-64
F6	9,56E+00	2,32E+00	4,47E+00	6,09E-01	2,40E+01	2,16E+00	3,77E+01	3,14E+00	1,68E+00	4,05E-01	1,62E-01	2,14E-02
F7	0	0										
F8	4,89E-16	6,20E-17	3,73E-16	6,67E-17	6,76E-72	6,26E-72	5,00E-62	9,38E-62	7,09E-171	0	3,03E-242	0
F9	9,66E-02	1,84E-02	5,43E-02	7,03E-03	6,79E-02	9,38E-03	1,14E-01	1,16E-02	4,35E-02	7,69E-03	3,64E-02	4,48E-03
F10	9,28E-02	1,37E-01	3,30E+00	1,28E+01	5,10E+01	3,77E+01	1,51E+00	1,34E+00	5,27E-01	1,18E+00	4,34E+01	1,32E+01
F11	0	0										
F12	0	0										
F13	3,74E-16	2,92E-16	2,47E-04	1,35E-03	3,96E-09	2,04E-08	0	0	0	0	0	0
F14	9,55E-01	5,23E+00	3,97E+01	6,47E+01	3,95E+00	2,16E+01	3,56E-11	2,18E-12	3,64E-11	0	4,72E-05	7,73E-05
F15	8,68E-14	8,48E-15	7,31E-14	5,57E-15	7,47E-14	4,12E-15	1,37E-13	1,24E-14	5,93E-14	7,65E-15	5,44E-14	3,86E-15
F16	1,18E-15	1,13E-16	1,05E-15	1,21E-16	7,85E-33	2,78E-48	6,19E-31	3,62E-31	7,85E-33	2,78E-48	7,85E-33	2,78E-48
F17	1,14E-15	1,33E-16	1,01E-15	1,28E-16	1,35E-32	5,57E-48	3,80E-29	1,87E-29	1,35E-32	5,57E-48	1,35E-32	5,57E-48
F18	2,31E-07	7,68E-07	7,34E-07	1,70E-06	2,23E-11	3,77E-11	8,20E-16	4,69E-16	5,42E-47	2,02E-46	1,83E-26	2,04E-26
F19	1,07E-15	1,26E-16	8,89E-16	8,73E-17	1,41E-31	1,47E-32	4,08E-30	2,58E-30	1,35E-31	6,68E-47	1,35E-31	6,68E-47
F20	1,75E-14	1,03E-14	9,00E-15	7,90E-15	2,65E-14	8,94E-15	9,94E-15	5,68E-15	0	0	0	0
F21	4,76E-01	7,84E-03	4,62E-01	1,79E-02	4,68E-01	9,17E-03	4,84E-01	3,62E-03	4,72E-01	1,42E-02	3,93E-01	6,73E-02
F22	-7,83E+01	5,77E-09	-7,83E+01	4,89E-14	-7,83E+01	1,76E-08	-7,83E+01	2,40E-07	-7,83E+01	6,84E-06	-7,83E+01	2,11E-13
F23	-1,93E+02	6,18E-01	-1,96E+02	2,84E-01	-1,76E+02	6,92E-01	-1,74E+02	9,91E-01	-1,99E+02	6,38E-01	-1,87E+02	3,30E-01
F24	1,16E-15	1,63E-16	1,01E-15	1,23E-16	0	0	5,61E-29	4,18E-29	0	0	0	0
F25	0	0										
F26	2,63E-16	2,71E-16	6,66E-17	1,08E-16	1,44E-08	7,17E-08	0	0	0	0	0	0
F27	8,38E-14	7,20E-15	7,54E-14	5,00E-15	7,39E-14	3,54E-15	2,00E-13	3,07E-14	5,97E-14	8,52E-15	3,88E-14	2,27E-15
F28	1,04E-07	2,24E-07	1,24E-05	5,65E-05	2,53E-10	1,17E-09	9,71E-16	5,70E-16	2,86E-16	3,41E-16	4,59E-05	2,78E-05

Table 4. Comparison of the ABC variants on the 100 and 300 (F22, F23) dimensional functions.

Func.	Original ABC		GABC		ABCBest2		MABC		ABCVSS		ABCFSW	
	Mean	Std.Dev.										
F1	2,18E-15	2,35E-16	1,84E-15	1,72E-16	5,09E-32	2,03E-32	1,43E-27	8,12E-28	1,01E-83	5,52E-83	1,39E-124	1,06E-124
F2	2,03E-15	1,98E-16	1,66E-15	1,72E-16	2,81E-26	1,61E-26	3,52E-24	2,47E-25	5,29E-75	2,80E-74	1,56E-119	1,83E-119
F3	2,22E-15	2,08E-16	1,85E-15	2,00E-16	2,15E-32	1,24E-32	4,46E-28	2,08E-28	5,31E-85	2,68E-84	1,19E-124	1,13E-124
F4	5,25E-17	1,38E-17	3,66E-17	9,63E-18	1,92E-29	6,05E-29	1,92E-48	3,42E-48	4,86E-109	2,66E-108	6,89E-160	0
F5	5,01E-15	2,95E-16	5,17E-15	2,24E-16	7,21E-17	1,36E-17	4,41E-15	1,50E-15	3,94E-40	2,12E-39	7,79E-69	3,82E-69
F6	2,39E+01	3,32E+00	1,59E+01	1,55E+00	5,06E+01	2,67E+00	5,98E+01	1,60E+00	7,91E+00	1,32E+00	1,36E+00	1,48E-01
F7	0	0										
F8	9,81E-16	1,02E-16	7,50E-16	9,05E-17	1,61E-69	1,73E-69	5,72E-60	5,32E-60	0	0	0	0
F9	1,65E-01	2,60E-02	9,47E-02	1,26E-02	1,42E-01	1,58E-02	2,31E-01	2,79E-02	7,87E-02	1,16E-02	6,79E-02	7,63E-03
F10	1,97E-01	3,53E-01	2,40E+01	3,39E+01	1,18E+02	5,76E+01	1,98E+00	1,30E+00	5,14E-01	1,05E+00	1,04E+02	1,86E+01
F11	0	0										
F12	0	0										
F13	4,66E-16	4,19E-16	1,26E-16	1,56E-16	2,94E-10	1,16E-09	0	0	0	0	0	0
F14	4,55E-10	1,79E-09	1,20E+02	1,22E+02	1,32E+00	7,21E+00	1,19E-10	4,06E-12	1,12E-10	3,67E-12	1,14E-10	3,63E-12
F15	1,51E-13	6,97E-15	1,32E-13	9,74E-15	1,39E-13	5,75E-15	3,56E-13	2,29E-14	1,11E-13	9,93E-15	1,14E-13	6,01E-15
F16	2,16E-15	2,62E-16	1,90E-15	1,95E-16	4,71E-33	1,39E-48	1,89E-30	8,42E-31	4,71E-33	1,39E-48	4,71E-33	1,39E-48
F17	2,11E-15	2,07E-16	1,85E-15	1,68E-16	2,18E-32	6,62E-33	1,81E-28	6,44E-29	1,35E-32	5,57E-48	1,35E-32	5,57E-48
F18	1,18E-05	2,84E-05	1,92E-05	3,56E-05	8,44E-10	1,74E-09	5,83E-15	1,97E-15	2,78E-17	1,17E-16	2,27E-06	1,25E-06
F19	2,02E-15	1,57E-16	1,70E-15	1,76E-16	2,42E-31	1,85E-31	8,49E-29	3,57E-29	1,35E-31	6,68E-47	1,35E-31	6,68E-47
F20	8,15E-14	2,33E-14	6,06E-14	1,79E-14	1,29E-13	2,33E-14	5,21E-14	6,69E-15	1,89E-15	7,21E-15	5,31E-14	9,83E-15
F21	4,97E-01	7,90E-04	4,95E-01	1,87E-03	4,97E-01	6,58E-04	4,99E-01	1,75E-04	4,98E-01	8,15E-04	4,89E-01	3,24E-03
F22	-7,83E+01	8,57E-11	-7,83E+01	1,72E-02	-7,83E+01	1,26E-09	-7,83E+01	1,84E-08	-7,83E+01	6,85E-06	-7,83E+01	1,72E-13
F23	-2,89E+02	1,14E+00	-2,88E+02	6,19E-01	-2,67E+02	7,75E-01	-2,63E+02	7,14E-01	-2,98E+02	1,20E+00	-2,84E+02	3,05E-01
F24	2,16E-15	1,81E-16	1,90E-15	1,96E-16	0	0	1,44E-27	9,16E-28	0	0	0	0
F25	0	0										
F26	4,48E-16	4,71E-16	9,25E-17	1,09E-16	1,01E-12	5,23E-12	0	0	0	0	0	0
F27	1,52E-13	9,00E-15	1,31E-13	7,35E-15	1,38E-13	6,90E-15	5,33E-13	7,58E-14	1,15E-13	1,08E-14	7,20E-14	3,08E-15
F28	4,29E-06	1,47E-05	1,03E-05	2,50E-05	1,01E-09	2,12E-09	3,01E-15	1,39E-15	3,48E-16	2,08E-16	1,43E-04	6,62E-05

performance of the ABCFWS algorithm. We compare these results with the values of the same test functions obtained by the ABC algorithm and its some variants such as the GABC [18], ABC/Best/2 [31], MABC [32], ABCVSS [29] algorithms.

A comparison of experimental results for the test functions with $D = 30$, $D = 60$ and $D = 100$ is presented in Tables 2–4, respectively.

When the tables comparing ABCFWS algorithm, which is presented above, with ABC, GABC, ABCBest2, MABC, ABCVSS algorithms were examined, the ABCFWS algorithm gives better or the same results in 23 of 28 problems, worse results in 5 problems compared to the other algorithms in which the problem dimension has been taken as $D = 30$ in Table 2. Of these problems, while 15 of them give better results, in 8 of them optimal results have been reached, thus they are equivalent to other algorithms. However, when Table 3 which has the problem dimension as $D = 60$ was examined, better or the same results have been obtained in 24 of the 28 problems, worse results have been obtained in 4 of them. Of these successful and the same results while 13 of them are better results, in 11 problems, the same results have been achieved in some or one of the other algorithms. Finally, when Table 4, which has problem dimension as $D = 100$, was studied, while of the 28 problems tested better or the same results are gained in 21 of them with the proposed ABCFWS algorithm, worse results are found in 7 of them. Among these 21 problems 10 problems give better results and 11 problems give the same results.

Considering all the comparison tables, the same results were found for F7, F11, F12 and F22 in all tests. In Tables 2–4 for F10, F23 and F28 a worse performance, for F7, F11, F12, F22 and F25 an equivalent performance and for the rest of the problems a better performance was achieved compared to the original ABC algorithm.

While obtaining good performances in most of the problem dimensions with the proposed ABCFWS method, this method has shown lower performance in F14, F18 and F28 test functions, and equal performance in F7, F11, F12 and F25 test functions.

As a consequence of the experimental test results and comparisons, the proposed ABCFWS has shown better convergence performance in most of the tested problems and better mean and standard deviation values have been obtained in comparison tables. In light of these good results, this new improvement in the ABC algorithm has been shown to be more stable and successful.

6. Conclusion

In this study, the performance of the ABC algorithm has been improved by making the search direction balanced in the search strategy of the ABC algorithm

towards individuals with better fitness. In order to evaluate the performance of the proposed method, experimental tests were performed on 28 numerical unconstrained test functions with different characteristics. According to the comparison results in the convergence graphs, it was observed that the proposed ABCFWS algorithm has achieved better convergence success. In addition to the convergence success, mean and standard deviation values of 30 independent experimental test results on the test functions with $D = 30$, $D = 60$ and $D = 100$ were compared between the proposed method ABCFWS and the original ABC algorithm and its some variants which are the GABC, ABC/Best/2, MABC, ABCVSS algorithms. Considering the results presented in the comparison tables, we obtained better or the same results for 23 problems out of 28 problems, however, we obtained worse results for 5 problems.

In conclusion, the search strategy of the proposed ABCFWS algorithm in the solution (search) space is successful and competitive.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Yuksel Celik  <http://orcid.org/0000-0002-7117-9736>

References

- [1] Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University (2005) (TR06) 10; 2005. https://abc.erciyes.edu.tr/pub/tr06_2005.pdf.
- [2] Dokeroglu T, Sevinc E, Cosar A. Artificial bee colony optimization for the quadratic assignment problem. *Appl Soft Comput.* 2019;76(March):595–606. doi:10.1016/J.ASOC.2019.01.001.
- [3] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial Bee Colony (ABC) algorithm. *J Global Optim.* 2007;39(3):459–471. doi:10.1007/s10898-007-9149-x.
- [4] Kumar A, Kumar D, Jarial SK. A review on artificial bee colony algorithms and their applications to data clustering. *Cybern Inf Technol.* 2017;17(3):3–28. doi:10.1515/cait-2017-0027.
- [5] Karaboga D, Gorkemli B, Ozturk C, et al. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artif Intell Rev.* 2014;42(1):21–57. doi:10.1007/s10462-012-9328-0.
- [6] Chen J, Yu W, Tian J, et al. Image contrast enhancement using an artificial bee colony algorithm. *Swarm Evol Comput.* 2018;38(February):287–294. doi:10.1016/J.SWEVO.2017.09.002.
- [7] Panda S. Performance improvement of optical CDMA networks with stochastic artificial bee colony optimization technique. *Opt Fiber Technol.* 2018;42(May):140–150. doi:10.1016/J.YOFTE.2018.03.005.
- [8] Qi A-L, Zhang G-M, Dong M, et al. An artificial bee colony optimization based matching pursuit approach for ultrasonic echo estimation. *Ultrasonics.* 2018;88(August):1–8. doi:10.1016/J.ULTRAS.2018.03.002.

- [9] Banharnsakun A. A map reduce-based artificial bee colony for large-scale data clustering. *Pattern Recognit Lett.* 2017;93(July):78–84. doi:10.1016/j.patrec.2016.07.027.
- [10] Lozano M, García-Martínez C, Rodríguez FJ, et al. Optimizing network attacks by artificial bee colony. *Inf Sci (Ny)*. 2017;377(January):30–50. doi:10.1016/J.INS.2016.10.014.
- [11] Szeto WY, Wu Y, Ho SC. An artificial bee colony algorithm for the capacitated vehicle routing problem. *Eur J Oper Res.* 2011;215(1):126–135. doi:10.1016/J.EJOR.2011.06.006.
- [12] Ding M, Chen H, Lin N, et al. Dynamic population artificial bee colony algorithm for multi-objective optimal power flow. *Saudi J Biol Sci.* 2017;24(3):703–710. doi:10.1016/J.SJBS.2017.01.045.
- [13] Karaboga D, Akay B. Artificial bee colony (ABC) algorithm on training artificial neural networks. In 2007 IEEE 15th Signal Processing and Communications Applications, IEEE; 2007, p. 1–4. <https://doi.org/10.1109/SIU.2007.4298679>.
- [14] Okdem S, Karaboga D, Ozturk C. An application of wireless sensor network routing based on artificial bee colony algorithm. In 2011 IEEE Congress of Evolutionary Computation (CEC); 2011, IEEE. p. 326–330. <https://doi.org/10.1109/CEC.2011.5949636>.
- [15] Yang J, Jiang Q, Wang L, et al. An adaptive encoding learning for artificial bee colony algorithms. *J Comput Sci.* 2019;30(January):11–27. doi:10.1016/j.jocs.2018.11.001.
- [16] Gao W, Wei Z, Luo Y, et al. Artificial bee colony algorithm based on parzen window method. *Appl Soft Comput.* 2019;74(January):679–692. doi:10.1016/J.ASOC.2018.10.024.
- [17] Zhang M, Tian N, Palade V, et al. Cellular artificial bee colony algorithm with Gaussian distribution. *Inf Sci (Ny)*. 2018;462(September):374–401. doi:10.1016/J.INS.2018.06.032.
- [18] Zhu G, Kwong S. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Appl Math Comput.* 2010;217(7):3166–3173. doi:10.1016/j.amc.2010.08.049.
- [19] Jadon SS, Tiwari R, Sharma H, et al. Hybrid artificial bee colony algorithm with differential evolution. *Appl Soft Comput J.* 2017;58(September):11–24. doi:10.1016/j.asoc.2017.04.018.
- [20] Ghosh Roy A, Peyada NK. Aircraft parameter estimation using hybrid neuro fuzzy and artificial bee colony optimization (HNFABC) algorithm. *Aerosp Sci Technol.* 2017;71(December):772–782. doi:10.1016/J.AST.2017.10.030.
- [21] Bouaziz S, Dhahri H, Alimi AM, et al. Evolving flexible beta basis function neural tree using extended genetic programming & hybrid artificial bee colony. *Appl Soft Comput.* 2016;47(October):653–668. doi:10.1016/J.ASOC.2016.03.006.
- [22] Sedighzadeh D, Mazaheripour H. Optimization of multi objective vehicle routing problem using a new hybrid algorithm based on particle swarm optimization and artificial bee colony algorithm considering precedence constraints. *Alexandria Eng J.* 2018;57(4):2225–2239. doi:10.1016/j.aej.2017.09.006.
- [23] Wu B, Qian C, Ni W, et al. Hybrid harmony search and artificial bee colony algorithm for global optimization problems. *Comput Math Appl.* 2012;64(8):2621–2634. doi:10.1016/j.camwa.2012.06.026.
- [24] Shokouhifar M, Jalali A. Simplified symbolic transfer function factorization using combined artificial bee colony and simulated annealing. *Appl Soft Comput.* 2017;55(June):436–451. doi:10.1016/J.ASOC.2017.02.029.
- [25] Duan HB, Xu CF, Xing ZH. A Hybrid artificial bee colony optimization and quantum evolutionary algorithm for continuous optimization problems. *Int J Neural Syst.* 2010;20(1):39–50. doi:10.1142/S012906571000222X.
- [26] Zhong Y, Lin J, Ning J, et al. Hybrid artificial bee colony algorithm with chemotaxis behavior of bacterial foraging optimization algorithm. In 2011 Seventh International Conference on Natural Computation; 2011. IEEE, p. 1171–1174. <https://doi.org/10.1109/ICNC.2011.6022147>.
- [27] Ding ZH, Huang M, Lu ZR. Structural damage detection using artificial bee colony algorithm with hybrid search strategy. *Swarm Evol Comput.* 2016;28(June):1–13. doi:10.1016/J.SWEVO.2015.10.010.
- [28] Ji J, Song S, Tang C, et al. An artificial bee colony algorithm search guided by scale-free networks. *Inf Sci (Ny)*. 2019;473(January):142–165. doi:10.1016/J.INS.2018.09.034.
- [29] Kiran MS, Hakli H, Gunduz M, et al. Artificial bee colony algorithm with variable search strategy for continuous optimization. *Inf Sci (Ny)*. 2015;300(April):140–157. doi:10.1016/J.INS.2014.12.043.
- [30] Babaoglu I. Artificial bee colony algorithm with distribution-based update rule. *Appl Soft Comput J.* 2015;34(September):851–861. doi:10.1016/j.asoc.2015.05.041.
- [31] Gao W, Liu S, Huang L. A global best artificial bee colony algorithm for global optimization. *J Comput Appl Math.* 2012;236(11):2741–2753. doi:10.1016/j.cam.2012.01.013.
- [32] Gao W-f, Liu S-y. A modified artificial bee colony algorithm. *Comput Oper Res.* 2012;39(3):687–697. doi:10.1016/J.COR.2011.06.007.