Applied Soft Computing 26 (2015) 454–462

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A directed artificial bee colony algorithm

Mustafa Servet Kıran [a,*], Oğuz Fındık [b]

[a] Department of Computer Engineering, Faculty of Engineering, University of Selcuk, 42075 Konya, Turkey
[b] Department of Computer Engineering, Faculty of Engineering and Architecture, Abant İzzet Baysal University, 14280 Bolu, Turkey

ARTICLE INFO

Article history:
Received 23 May 2014
Received in revised form
11 September 2014
Accepted 9 October 2014
Available online 24 October 2014

Keywords:
Swarm intelligence
Artificial bee colony
Direction information
Numerical optimization

ABSTRACT

Artificial bee colony (ABC) algorithm has been introduced for solving numerical optimization problems, inspired collective behavior of honey bee colonies. ABC algorithm has three phases named as employed bee, onlooker bee and scout bee. In the model of ABC, only one design parameter of the optimization problem is updated by the artificial bees at the ABC phases by using interaction in the bees. This updating has caused the slow convergence to global or near global optimum for the algorithm. In order to accelerate convergence of the method, using a control parameter (modification rate-MR) has been proposed for ABC but this approach is based on updating more design parameters than one. In this study, we added directional information to ABC algorithms, instead of updating more design parameters than one. The performance of proposed approach was examined on well-known nine numerical benchmark functions and obtained results are compared with basic ABC and ABCs with MR. The experimental results show that the proposed approach is very effective method for solving numeric benchmark functions and successful in terms of solution quality, robustness and convergence to global optimum.

1. Introduction

Swarm intelligence is a subfield of artificial intelligence and the algorithms of swarm intelligence have been developed by inspiring natural behavior of real ants [1], bees, birds, fishes [2], etc. Artificial bee colony algorithm is one of the swarm intelligence algorithms and has been developed by using waggle dance and foraging behaviors of real honey bee colonies [3]. In the nature, the honey bees search and forage food sources around the hive and share position information about the food sources. The honey bees which work in the foraging labor are divided into three groups. First group is employed bees and they move to hive nectar foraged from food source and position information about the food sources. Second group consists of onlooker bees and onlookers forage food sources by considering information shared by employed bees. The last group of the bees is scout bees. 5–10% of a bee population is scout bee [3,4] and scout bees search new food source around the hive and share position information of found new food sources with the other bees. For sharing information, the bees use waggle dance in the dance area of the hive and the time and glow of the dance depend on amount and distance from hive of food source.

Karaboga [3] used the aforementioned natural behaviors of the real honey bee in order to develop the artificial bee colony algorithm and solve numerical optimization problems. In the ABC algorithm, half of the population is first scout bee. For each scout bee, a new food source which is possible solution for the optimization problem is generated. After generating new food source position, all the scout bees become employed bees and all employed bees try to improve food sources by using interaction between them. If a food source could not be improved in a certain time named as limit which is a control parameter for ABC algorithm, the employed bee of this food source becomes a scout bee. For this scout bee, a new food source is produced and the scout bee becomes employed bee, again. The onlooker bees wait to be shared food sources position by the employed bees in the hive. After employed bees share position information about the food sources, each onlooker bee select one of the food source position and tries to improve the food source position.

ABC algorithm is an iterative algorithm and only one design parameter of the optimization problem is updated by the each employed or onlooker bee at the each ABC iteration and updating only one design parameter has caused slow convergence for the algorithm. In order to overcome this issue, Akay and Karaboga [5] have proposed a control parameter called as modification rate-MR. In this work, we used directional information for each design parameter in order to cope slow convergence of the algorithm and the performance of the proposed approach is investigated on the well-known numerical benchmark optimization problems.

The paper is organized as follows: Section 1 introduces the study and gives literature review on artificial bee colony algorithm. The

* Corresponding author. Tel.: +90 332 223 1992; fax: +90 332 223 2106.
E-mail address: mskiran@selcuk.edu.tr (M.S. Kıran).

http://dx.doi.org/10.1016/j.asoc.2014.10.020
1568-4946/

basic ABC algorithm and modifications are explained in Section 2 and the experiments and experimental results are presented in Section 3. The study is discussed in Section 4 and finally, the conclusions and future works are given in Section 5.

1.1. Literature review

The ABC algorithm was first introduced in 2005 and its performance is analyzed on three numerical problems [3]. It is mentioned that ABC algorithm has developed for solving numeric optimization problems [6] and proposed modifications and improvements for the methods have also been tested by using numeric problems. We give the literature review based on modifications and improvements of the method, and the studies on applications and hybridizations which use basic ABC algorithm can be found in a comprehensive literature review on ABC in [6]. The performance of ABC has been investigated on the numeric benchmark functions in [7–10]. Akay and Karaboga [5] introduced a modified version of ABC and used it for real-parameter optimization. In the modified ABC, Karaboga and Akay added two new control parameters named as modification rate-MR which is used for increasing convergence rate of ABC and scaling factor-SF which is used for controlling magnitude of perturbation to ABC. Dongli et al. [11] proposed three modified versions of ABC in order to obtain better quality results for the optimization problems. In the first modification, the neighborhood structure is changed in the solution updating equation of ABC, in the second modification, a new selection equation is proposed for onlooker bees in order to choose an employed bee and the last modified version of ABC is based on modification #1 and #2. Tsai et al. [12] proposed a model based on ABC, by employing Newtonian law of universal gravitation in onlooker bee phase of ABC. Alatas [13] proposed an ABC model that uses chaotic maps for parameter adaptation so as to prevent the ABC to get stuck local minimums. Zhu and Kwong [14] modified ABC algorithm (named as GABC) by appending the global best information of the population to exploitation equation of ABC in order to increase exploitation ability of ABC. Gao and Liu [15] modified search equation of the basic ABC by using chaotic systems and opposition-based learning methods and applied the modified ABC (called as MABC) to 28 benchmark functions. Banharnsakun et al. [16] improved the capability of convergence of ABC to a global optimum by using the best-so-far selection for onlooker bees and they tested performance of their method on the numerical benchmark functions and image registration. The Rosenbrock's rotational direction method which was designed to cope with specific features of “Rosenbrock's banana function” was applied to ABC in order to increase exploitation and local search abilities of the basic ABC [17]. Karaboga and Akay [18] adapted the basic ABC for constrained optimization problems by using the Deb's Rules and evaluated the performance of the adapted model on the 13 constrained optimizations in the literature. Kıran and Gündüz [19] proposed a crossover-based improvement for neighbor bee selection in onlooker bee phase of basic ABC algorithm. Horng [20] proposed maximum entropy thresholding based on the ABC for image segmentation and Omkar et al. [21] presented vector evaluated ABC (VEABC) for multi-objective design optimization of laminated composite components and compared performance of VEABC with other population based methods. Liu et al. [22] have published a variant of ABC algorithm which is improved by using mutual learning which tunes the produced candidate food source with the higher fitness between two individuals selected by a mutual learning factor. Gao et al. [23] proposed two ABC-based algorithms which use two update rules of differential evolution (DE), and they are called as ABC/best/1 and ABC/best/2. The global best-based ABC methods also use chaotic initialization in order to properly distribute the agents to the search space, and the performance and accuracy of the methods are examined on 26 numerical benchmark functions [23]. Due to premature convergence and getting trap of local minima in ABC/best/1, Gao and Liu [24] proposed to use the update rule of ABC/best/1 algorithm for employed bees and the update rule of basic ABC for onlooker bees in order to reinforce exploration ability of the method, and they tested the variant of ABC on 28 numerical benchmark functions. In another study, Gao et al. [25] defined a new update rule for ABC algorithm, and the new update rule uses random solutions for obtaining the candidate solution. New update rule looks like to crossover operator of GA, and the method is named as CABC. In this study, the orthogonal learning strategy is proposed for ABC methods such as basic ABC (OABC), GABC (OGABC), CABC (OCABC), and their accuracies and performances are examined on numerical benchmark functions and compared with the other nature-inspired optimization algorithm.

Being analyzed literature review, it is seen that the update rule of ABC algorithm was modified, and the local search capability of the method is tried to improve in most of the papers which are based on improvement of the ABC algorithm. Therefore, we propose the same update rule with a bit modification in order to improve convergence characteristic of the basic ABC algorithm instead of local search capability. But it should be mentioned that this modification provides local search capability besides improvement convergence characteristics of the basic ABC algorithm.

2. ABC algorithm

By simulating intelligent behavior of real honey bee colonies, ABC algorithm tries to find a global optimum or near optimum solution for the optimization problems. In the ABC algorithm, number of food sources is equals to number of employed bees and also number of employed bees equals to number of onlooker bees. All the employed bees are scout bees in the starting of the algorithm and a food source position is produced for each scout bee using Eq. (1):

$$P_{ij} = X_j^{\min} + r \times (X_j^{\max} - X_j^{\min}), \quad i = 1, 2, \ldots, \mathrm{NE} \quad \text{and} \quad j = 1, 2, \ldots, D \tag{1}$$

where P_{ij} is the jth dimension of ith food source which will be assigned to ith employed bee, $X_j^{\min}$ and $X_j^{\max}$ are the lower and upper bounds of the jth dimension, respectively, r is a random number between [0,1], NE is the number of employed bee and D is the dimensionality (the number of decision variables) of the problem or function optimized.

After producing a food source position for each scout bee, all the scout bees become employed bee. The qualities of the food sources of the employed bees are measured by using Eq. (2):

$$\mathrm{fit}_i = \begin{cases} 1/(1+f_i) & \text{if}\,(f_i \geq 0) \\ 1+\mathrm{abs}(f_i) & \text{if}\,(f_i, 0) \end{cases} \tag{2}$$

where fit_i is the fitness of the ith food source and f_i is the objective function value specific for the optimization problem. In addition, a trial counter is defined and reset for each food source and limit value for the population is described in the initialization of the algorithm.

The employed bees search around the self-food sources for new food sources. A new food source position around the food source of employed bee is obtained as follows:

$$V_{ij} = S_{ij} + \varphi \times (S_{ij} - N_j), \quad i{=}1, 2, \ldots, \mathrm{NE}, \quad k \in \{1, 2, \ldots, \mathrm{NE}\} \quad \text{and} \quad j \in \{1, 2, \ldots, D\} \tag{3}$$

where V is the candidate food source position produced for food source position S, N is the randomly selected neighbor food source for food source S and φ is a random number in range of [−1,1]. It is mentioned that only one dimension of the food source position

Initialization Phase
Determine the number of food sources
Define *Limit* for the population
Produce the food sources using equation 1.
Define trial counters for the food sources.
Assign the food sources to the employed bees
Calculate fitness of the food sources using Eq. 2.
Employed Bee Phase
For Each Employed Bee
Produce new food source position using Eq.3.
Calculate fitness of the candidate food source.
If the fitness of candidate food source is better than old one, memorize new position and reset trial counter; otherwise increase its trial counter by 1.
Onlooker Bee Phase
Calculate being selected probabilities of the employed bees using Eq. 4
For each onlooker bee
Produce new food source position using Eq.3.
Calculate fitness of the candidate food source.
If the fitness of candidate food source is better than old one, memorize new position and reset trial counter; otherwise increase its trial counter by 1.
Save the best solution obtained so far.
Scout Bee Phase
If a scout bee occurs
Produce a new food source position by using Eq.1.
Calculate fitness of the produced food source position by using Eq.2.
Reset its trial counter.
IF a termination condition is met **THEN** report the best solution
ELSE go to Employed Bee Phase

Fig. 1. The ABC algorithm.

S is updated for each the iteration and the dimension is randomly selected. The fitness of the candidate food source is obtained by using Eq. (2) and if the fitness of candidate food source is better than the old one, the employed bee memorizes the new food source position and trial counter of the food source is reset; otherwise the trial counter of the food source is increased by 1.

After the employed bees return to the hive, the employed bees share self-food source positions with the onlooker bees. An onlooker bee selects an employed bee and memorizes its food source position in order to improve its food source by using roulette-wheel selection mechanism given as follows:

$$p_i = \frac{\text{fit}_i}{\sum_{j=1}^{\text{NE}} \text{fit}_j} \tag{4}$$

where p_i is the being selected probability of the ith employed bee by an onlooker bee. Thereafter, the onlooker bee searches around the food source position of the employed bee by using Eq. (3). If the fitness of the food source found by onlooker bee is better than fitness of the food source of the employed bee, the employed bee memorizes the food source position of the onlooker bee and trial counter of this food source is reset; otherwise the trial counter of the food source is increased by 1.

The occurrence of the scout bee in the ABC depends on limit and trial counters of the food sources. After onlooker's search, the trial counter with maximum content (H) is fixed and if H is higher than the limit, a new food source position is produced for this bee by using Eq. (1) and its trial counter is reset. It is mentioned that only one scout bee can occur at the each ABC iteration.

The ABC algorithm is an iterative algorithm and consists of four phases sequentially realized named as initialization, employed bee, onlooker bee and scout bee phases. In order to terminate the algorithm, the maximum iteration number, meeting an error tolerance, etc can be used. The detailed algorithm of ABC is also shown in Fig. 1.

2.1. ABC algorithm with MR control parameter

In order to increase convergence rate of the method, Akay and Karaboga [5] proposed a control parameter named as modification rate-MR. In the basic ABC, only one dimension of the food source position is updated by the employed or onlooker bees, but in the ABC with MR (called as ABC_{MR}), whether a dimension will be updated is decided by using MR value which is a number in range of [0,1]. By using MR parameter, the Eq. (2) is changed as follows:

$$V_{ij} = \begin{cases} S_{ij} + \varphi \times (S_{ij} - N_j) & \text{if}(R_{ij} < \text{MR}) \\ S_{ij} & \text{otherwise} \end{cases} \tag{5}$$

where R_{ij} is a random number produced in range of [0,1]. If random number is less than MR, the dimension j is modified and at least one dimension is updated by using Eq. (3). The lower value for MR may cause solutions to improve slowly while higher value for MR can be caused too much diversification in the population [5]. Therefore, we used 0.3 and 0.7 values for the MR in the experiments by obtaining from [5].

2.2. Directed ABC (dABC)

The searching around the food source in the basic ABC is fully random in terms of direction because φ is a random number between [−1,1]. This undirected search has caused the slow convergence of the algorithm to the optimum or near optimum. Therefore, we added direction information for each dimension of for each food source position. By using direction information for the dimensions, the Eq. (3) is modified as follows:

$$V_{ij} = \begin{cases} S_{ij} + \varphi \times (S_{ij} - N_j) & \text{if}(d_{ij} = 0) \\ S_{ij} + r \times \text{abs}(S_{ij} - N_j) & \text{if}(d_{ij} = 1) \\ S_{ij} - r \times \text{abs}(S_{ij} - N_j) & \text{if}(d_{ij} = -1) \end{cases} \tag{6}$$

where abs is absolute function, d_{ij} is the direction information for jth dimension of the ith food source position and while φ is a random

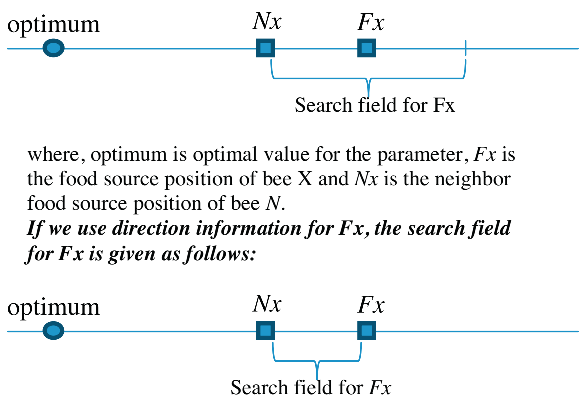

Fig. 2. An illustrative example of using direction information.

number in range of [−1,+1], r is a random number produced in range of [0,1].

The Eq. (6) identifies the direction of searching. In the initialization of the algorithm, the direction of information for all dimensions equals to 0. If the new solution obtained by Eq. (6) is better than old one (the better solution is determined by using fitness values of the old and new solutions via Eq. (2)), the direction information is updated. If previous value of the dimension is less than current value, the direction information of this dimension is set to −1; otherwise the direction information of this dimension is set to 1. If new solution obtained by Eq. (6) is worse than old one, the direction information of the dimension is set to 0. In this way, the direction information of each dimension of each food source position is used and also, the local search capability and convergence rate of the algorithm are improved. This situation is also shown by using an illustrative example in Fig. 2. Based on Fig. 2, a worse value can be obtained for *Fx* because an undirected search is performed on the search field for *Fx*. If we use direction information for *Fx*, the search tends to optimum.

3. Experiments

The performance of the method is investigated on the well-known eight benchmark functions taken from [5] and collected from the literature. The experiments are conducted on IBM-compatible PC with 3.01 GHZ, 4GB Ram and Matlab® 7.04 platform. Each experiment is repeated 30 times with random seeds and the best, the worst, mean and standard deviations are reported on the comparisons.

3.1. Benchmark functions

The benchmark functions are given in Table 1. *D*, *C*, Range and $f(x^*)$ in Table 1, are dimensions, characteristics, lower and upper bounds of search spaces and global minimum values of the functions, respectively. The numerical functions used in the experiments have some characteristics. If a function has more than one local minimum, this function is called as multimodal (*M*) and the multimodal functions such as Rastrigin, Griewank tests search ability of the algorithms. Unimodal functions (*U*) such as Sphere has only one local optimum and this is global optimum. The exploitation ability of the algorithms is examined on this kind of functions. If a function with n-variable can be written as sum of the n functions of one variable, then this function is called as separable (*S*) function (Sphere, Rastrigin). Non-separable functions such as Rosenbrock functions cannot be written in this form because there is interrelation among variables of these functions. Therefore, to optimize non-separable functions is more difficult than optimizing the separable functions. The dimensionality of the search space is also an important issue with the problem for the algorithms [9,26]. If the global optimum of the function is in the narrow curving valley such as Rosenbrock's Banana function, the methods should keep up the direction changes in the functions. In the experiments, we investigated and compared the performance of the methods on the numeric functions with 10, 30 and 50 dimensionalities.

3.2. Setting control parameters for methods

In order to make a clear and consistent comparison, the control parameters values of the methods are equal to each other. Akay and Karaboga [27] show that there is no need to a huge colony size for basic ABC algorithm. Therefore, the population size is taken as 40 in the experiments. The limit value which is a specific control parameter for ABC algorithms for the population is calculated as follows [9]:

$$\text{limit} = \text{NE} \times D \tag{7}$$

where limit is used for controlling occurrence of scout bee, NE is the number of food source or employed bee and *D* is dimensionality of the optimization problem. By using Eq. (7), the occurrence of scout

Table 1
Benchmark functions used in the experiments.

Function	C	$f([x]^D)$	Range	Formulae
F1-Sphere	US	0	$[-100,100]^D$	$f(x)=\sum_{i=1}^{D} x_i^2$
F2-Rosenbrock	UN	0	$[-2.048,2.048]^D$	$f(x)=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$
F3-Ackley	MN	0	$[-32.768,32.768]^D$	$f(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+e$
F4-Griewank	MN	0	$[-600,600]^D$	$f(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$
F5-Weierstrass	MN	0	$[-0.5,0.5]^D$	$f(x)=\sum_{i=1}^{D}\left(\sum_{k=0}^{kmax}[a^k\cos(2\pi b^k(x_i+0.5))]\right)-D\sum_{k=0}^{kmax}[a^k\cos(2\pi b^k 0.5)]$ $a=0.5,\quad b=3,\quad k_{\max}=20$
F6-Rastrigin	MS	0	$[-5.12,5.12]^D$	$f(x)=\sum_{i=1}^{D}[x_i^2-10\cos(2\pi x_i)+10]$
F7-Non-Cont. Rastrigin	MS	0	$[-5.12,5.12]^D$	$f(x)=\sum_{i=1}^{D}[y_i^2-10\cos(2\pi y_i)+10]$ $y_i=\begin{cases} x_i & \lvert x_i\rvert>\frac{1}{2} \\ \frac{\text{round}(2x_i)}{2} & \lvert x_i\rvert\le\frac{1}{2}\end{cases}$
F8-Schwefel	MN	0	$[-500,500]^D$	$f(x)=418.9829\times D-\sum_{i=1}^{D}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$
F9-Sumsquares	US	0	$[-10,10]^D$	$f(x)=\sum_{i=1}^{D} ix_i^2$

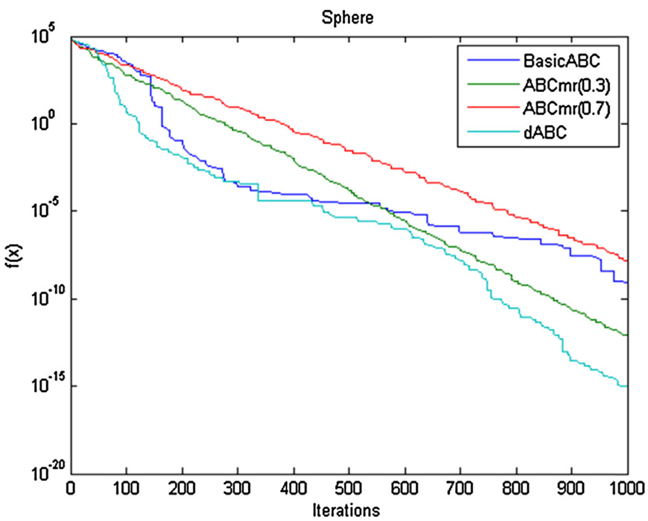

Fig. 3. The convergence graph of the methods on the sphere function with 30-D.

bee is properly controlled by depending on population size and dimensionality of the problem.

The ABC_{MR} algorithm with lower MR value likes to basic ABC algorithm and ABC_{MR} algorithm with higher MR value has caused more diversification in the population [5]. Therefore, we used 0.3 and 0.7 values for the MR parameter in the experiments.

The termination condition for the methods is used maximum cycle number (MCN) and MCN is taken as 500, 1000 and 1500 for 10, 30 and 50-dimensional numeric functions, respectively.

3.3. Comparison of the methods

Comparisons of the basic ABC, ABC_{MR}, dABC algorithms are given in Tables 2–4. Based on the comparisons, the dABC algorithm is very effectiveness for solving numeric functions.

While the dimensionality of the functions is increased, the performance of the methods is decreased but when the performance of the methods is compared, it is shown that the dABC algorithm is better than the other methods in terms of solution quality and robustness by considering mean results and standard deviations. Wilcoxon non-parametric signed-rank statistical test with 0.05 *p* value is performed to the results of 30 independent runs and the statistical tests results are shown in Tables 5–7 for 10, 30 and 50-dimensional functions, respectively. According to statistical test results, the proposed method is significantly different from the basic or other ABC variants in most cases. The convergence graphs of the methods (Figs. 3–10) are also figured for the 30-D functions and the convergence rate of dABC algorithm is better than the other methods except Weierstrass function. In addition, the experimental results and convergence graphs of the ABC_{MR} show that 0.3 value for MR parameter is more appropriate than 0.7 value.

4. Discussion

The swarm intelligence-based optimization methods start with random initial solutions to search solution space. For obtaining an optimum or near optimum solution, the interactions between the agents in the population are used. In the ABC algorithm, the dance behavior is performed for sharing position information about the food sources. In this point, the direction information is important factor for finding a good solution although basic ABC algorithm is undirected and this information is not shared in the artificial hive of ABC. This issue has caused to slow convergence and decreased local search ability of basic ABC algorithm. In this work, for each

Table 2
The best, worst, mean and standard deviations of results obtained by 30 independent runs on numeric functions. D:10 and MCN: 500.

Function	Basic ABC				ABC_{MR} (MR = 0.3)				ABC_{MR} (MR = 0.7)				dABC			
	Best	Worst	Mean	Std. Dev.	Best	Worst	Mean	Std. Dev.	Best	Worst	Mean	Std. Dev.	Best	Worst	Mean	Std. Dev.
F1	7.40E−17	3.05E−16	2.08E−16	6.49E−17	3.17E−17	1.11E−16	8.37E−17	2.00E−17	**3.80E−17**	**1.04E−16**	**6.43E−17**	**1.55E−17**	7.07E−17	2.77E−16	1.51E−16	6.47E−17
F2	2.34E−02	3.93E+00	8.93E−01	1.02E+00	3.08E−01	6.79E+00	3.69E+00	1.90E+00	6.74E−01	6.24E+00	5.47E+00	9.67E−01	**1.62E−02**	**2.00E+00**	**2.80E−01**	**4.15E−01**
F3	7.88E−11	4.98E−09	7.67E−10	9.34E−10	3.61E−12	5.33E−11	1.79E−11	1.16E−11	2.21E−13	4.73E−12	1.33E−12	1.12E−12	**1.87E−14**	**2.42E−13**	**7.30E−14**	**4.53E−14**
F4	1.72E−11	2.71E−02	1.08E−02	8.92E−03	3.15E−06	3.75E−02	1.09E−02	9.06E−03	1.01E−02	3.02E−01	1.22E−01	7.74E−02	**1.90E−13**	**2.71E−02**	**6.09E−03**	**7.31E−03**
F5	1.13E−11	2.24E−09	5.23E−10	5.99E−10	0.00E+00	1.81E−13	2.58E−14	3.64E−14	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F6	0.00E+00	1.23E−11	8.03E−13	2.31E−12	2.64E−09	1.28E−04	1.03E−05	2.56E−05	3.09E+00	1.68E+01	1.04E+01	3.05E+00	**0.00E+00**	**2.84E−14**	**1.42E−15**	**5.62E−15**
F7	0.00E+00	1.28E−09	1.18E−10	2.97E−10	8.58E−07	1.00E+00	3.35E−02	1.80E−01	6.38E+00	1.15E+01	8.33E+00	1.28E+00	**0.00E+00**	**1.48E−11**	**5.58E−13**	**2.66E−12**
F8	1.27E−04	1.18E+02	1.18E+01	3.55E+01	1.27E−04	2.97E+01	1.06E+00	5.33E+00	1.52E−03	9.09E+02	3.12E+02	2.43E+02	**1.27E−04**	**7.35E−04**	**1.56E−04**	**1.16E−04**
F9	7.00E−17	2.47E−16	1.27E−16	4.98E−17	2.00E−17	2.69E−16	1.05E−16	5.87E−17	**2.15E−17**	**8.78E−17**	**6.46E−17**	**1.65E−17**	5.09E−17	2.47E−16	1.33E−16	6.22E−17

Table 3
The best, worst, mean and standard deviations of results obtained by 30 independent runs on numeric functions. D:30 and MCN: 1000.

Function	Basic ABC				ABC_{MR} (MR = 0.3)				ABC_{MR} (MR = 0.7)				dABC			
	Best	Worst	Mean	Std. Dev.	Best	Worst	Mean	Std. Dev.	Best	Worst	Mean	Std. Dev.	Best	Worst	Mean	Std. Dev.
F1	1.88E−10	2.53E−08	3.14E−09	5.38E−09	1.85E−13	3.02E−12	8.93E−13	5.42E−13	5.06E−09	8.37E−08	3.20E−08	2.21E−08	**9.53E−16**	**5.98E−15**	**1.92E−15**	**1.08E−15**
F2	1.77E−01	2.79E+01	1.79E+01	7.44E+00	2.16E+01	2.65E+01	2.45E+01	1.57E+00	2.07E+01	2.88E+01	2.59E+01	1.42E+00	**1.32E−01**	**2.40E+01**	**1.02E+01**	**7.35E+00**
F3	4.14E−06	6.58E−05	2.36E−05	1.48E−05	1.37E−07	5.94E−07	2.69E−07	9.20E−08	2.44E−05	1.24E−04	5.46E−05	2.61E−05	**1.35E−08**	**1.32E−07**	**6.76E−08**	**3.26E−08**
F4	2.23E−10	2.33E−02	3.03E−03	7.13E−03	**1.51E−10**	**4.65E−06**	**4.29E−07**	**1.10E−06**	3.88E−07	8.67E−02	4.67E−03	1.59E−02	1.89E−15	7.53E−03	2.59E−04	1.35E−03
F5	4.16E−04	1.27E−03	7.82E−04	2.20E−04	**8.88E−07**	**7.55E−06**	**2.90E−06**	**1.75E−06**	7.39E−04	2.04E−03	1.33E−03	3.67E−04	2.13E−05	6.19E−05	3.87E−05	1.17E−05
F6	5.97E−09	1.99E+00	2.98E−01	5.03E−01	4.60E+01	7.62E+01	6.28E+01	7.13E+00	1.40E+02	1.77E+02	1.58E+02	8.85E+00	**1.57E−10**	**1.25E+00**	**2.42E−01**	**4.39E−01**
F7	1.10E−08	2.37E+00	1.14E+00	8.46E−01	3.24E+01	5.29E+01	4.32E+01	4.94E+00	1.04E+02	1.65E+02	1.38E+02	1.25E+01	**3.57E−08**	**3.03E+00**	**1.01E+00**	**9.66E−01**
F8	1.19E+02	5.94E+02	3.99E+02	1.29E+02	1.88E+03	3.43E+03	2.78E+03	4.09E+02	4.51E+03	6.79E+03	5.89E+03	5.06E+02	**1.18E−01**	**5.97E+02**	**3.68E+02**	**1.42E+02**
F9	1.75E−11	2.97E−10	9.11E−11	7.67E−11	1.51E−14	2.62E−13	1.02E−13	7.65E−14	5.93E−10	7.05E−09	2.71E−09	1.50E−09	**5.54E−16**	**1.20E−15**	**8.62E−16**	**1.53E−16**

Table 4
The best, worst, mean and standard deviations of results obtained by 30 independent runs on numeric functions. D:50 and MCN: 1500.

Function	Basic ABC				ABC_{MR} (MR = 0.3)				ABC_{MR} (MR = 0.7)				dABC			
	Best	Worst	Mean	Std. Dev.	Best	Worst	Mean	Std. Dev.	Best	Worst	Mean	Std. Dev.	Best	Worst	Mean	Std. Dev.
F1	4.08E−09	3.46E−07	4.75E−08	6.48E−08	3.22E−09	1.83E−08	7.76E−09	3.67E−09	5.09E−03	2.37E−02	1.33E−02	4.60E−03	**1.76E−14**	**2.91E−12**	**5.32E−13**	**6.20E−13**
F2	2.45E+01	9.05E+01	4.55E+01	1.48E+01	4.29E+01	5.13E+01	4.53E+01	1.63E+00	4.72E+01	1.40E+02	6.74E+01	2.47E+01	**9.25E+00**	**8.78E+01**	**4.02E+01**	**2.20E+01**
F3	4.62E−05	5.71E−04	2.24E−04	1.06E−04	1.23E−05	4.16E−05	2.46E−05	6.48E−06	1.34E−02	4.94E−02	2.97E−02	8.50E−03	**3.53E−07**	**3.47E−06**	**1.60E−06**	**7.61E−07**
F4	5.14E−09	1.68E−02	9.00E−04	3.46E−03	2.78E−08	3.75E−05	5.46E−06	7.92E−06	1.10E−02	2.59E−01	7.40E−02	5.80E−02	**5.99E−13**	**1.20E−06**	**4.56E−08**	**2.15E−07**
F5	2.07E−03	6.57E−03	4.19E−03	9.68E−04	4.00E−04	1.09E−03	7.55E−04	1.76E−04	1.61E−01	3.36E−01	2.11E−01	3.98E−02	**1.82E−04**	**6.93E−04**	**4.53E−04**	**1.10E−04**
F6	1.51E−07	4.12E+00	1.86E+00	1.30E+00	1.82E+02	2.26E+02	2.09E+02	1.26E+01	2.90E+02	4.08E+02	3.65E+02	2.70E+01	**1.73E−07**	**4.98E+00**	**1.62E+00**	**1.36E+00**
F7	2.72E−01	7.30E+00	4.19E+00	1.76E+00	1.23E+02	1.83E+02	1.60E+02	1.74E+01	2.52E+02	3.86E+02	3.43E+02	3.05E+01	**1.00E+00**	**7.00E+00**	**3.56E+00**	**1.76E+00**
F8	5.50E+02	1.30E+03	1.02E+03	1.80E+02	6.96E+03	8.87E+03	8.00E+03	4.26E+02	9.54E+03	1.29E+04	1.16E+04	1.07E+03	**5.43E+02**	**1.25E+03**	**9.66E+02**	**2.07E+02**
F9	1.06E−09	1.17E−07	1.34E−08	2.10E−08	6.19E−10	3.01E−09	1.42E−09	5.37E−10	8.00E−04	4.62E−03	2.02E−03	9.85E−04	**5.47E−15**	**2.79E−13**	**3.80E−14**	**5.09E−14**

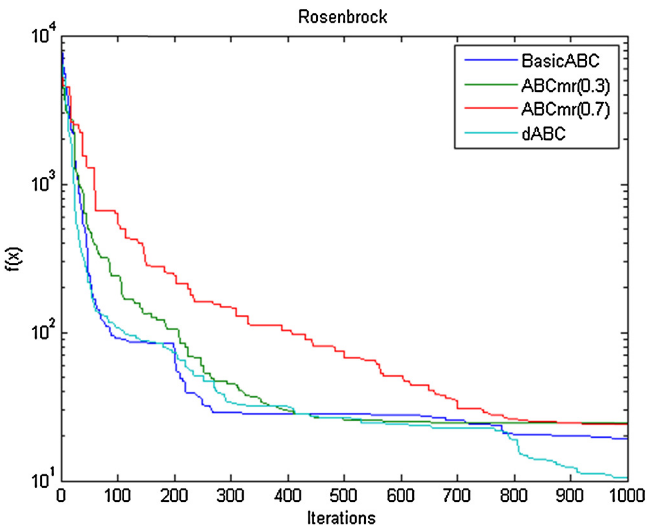

Fig. 4. The convergence graph of the methods on the Rosenbrock function with 30-D.

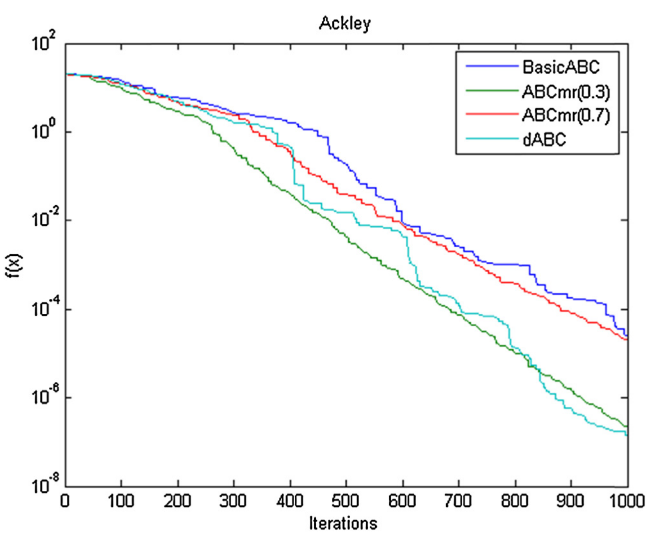

Fig. 5. The convergence graph of the methods on the Ackley function with 30-D.

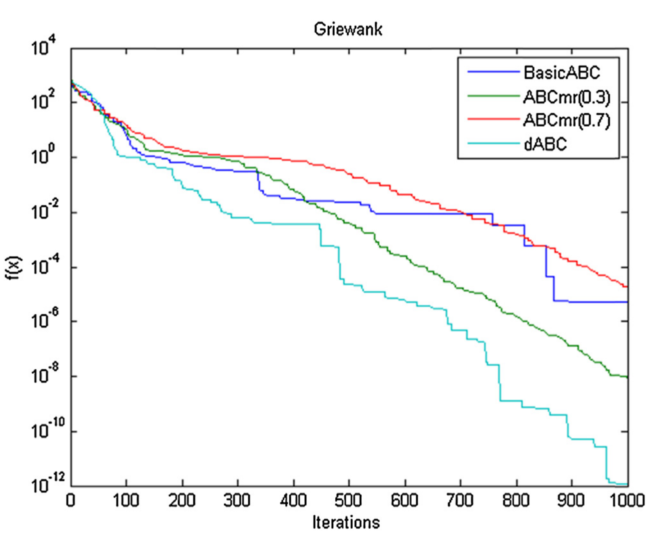

Fig. 6. The convergence graph of the methods on the Griewank function with 30-D.

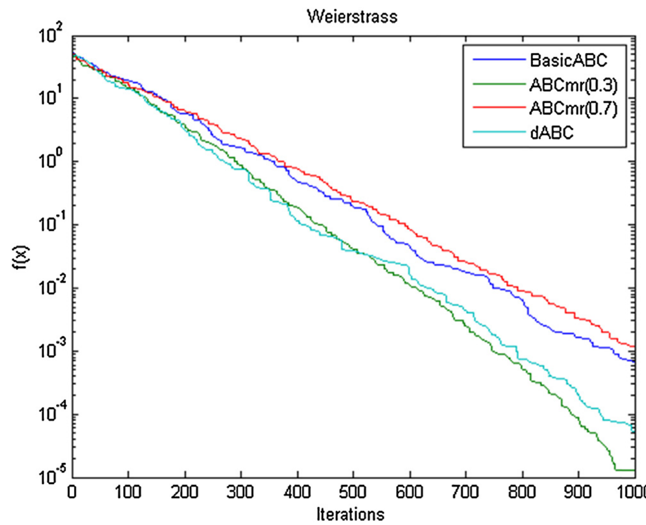

Fig. 7. The convergence graph of the methods on Weierstrass function with 30-D.

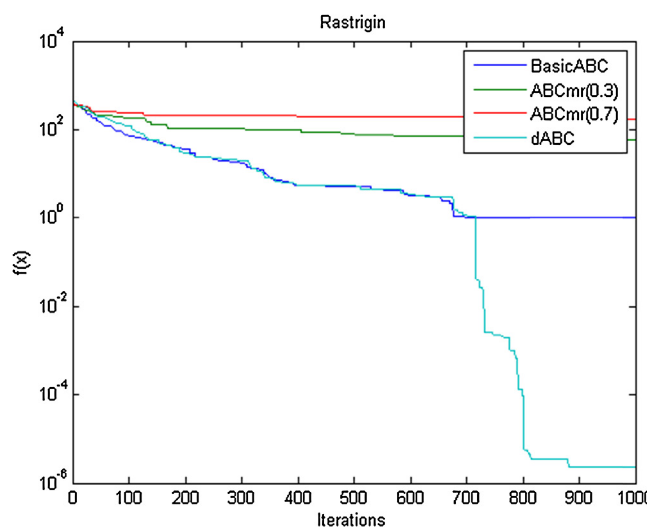

Fig. 8. The convergence graph of the methods on the Rastrigin function with 30-D.

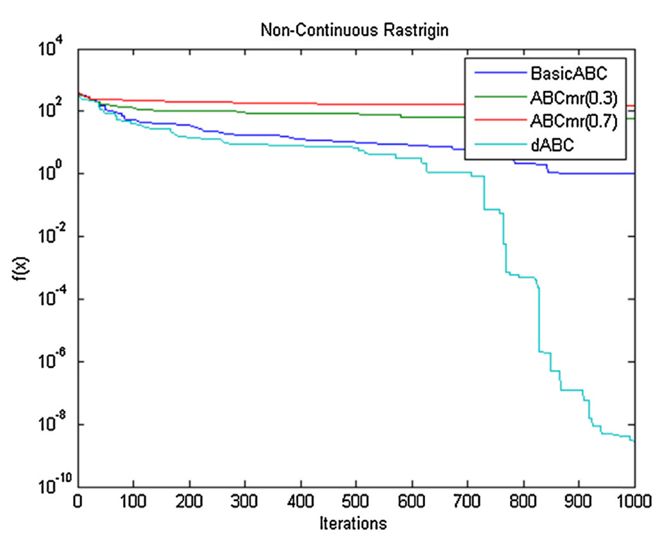

Fig. 9. The convergence graph of the methods on Non-continuous Rastrigin function with 30-D.

Table 5
Statistical significant test results among the ABC variants for 10-dimensional functions.

Function.	Basic ABC	ABC_{MR} (MR = 0.3)	ABC_{MR} (MR = 0.7)
F1	–	–	–
F2	+	+	+
F3	+	–	–
F4	–	+	+
F5	+	+	NA
F6	+	+	+
F7	+	+	+
F8	+	+	+
F9	–	+	+

Table 6
Statistical significant test results among the ABC variants for 30-dimensional functions.

Function.	Basic ABC	ABC_{MR} (MR = 0.3)	ABC_{MR} (MR = 0.7)
F1	+	+	+
F2	–	–	–
F3	–	–	–
F4	+	+	+
F5	+	+	+
F6	–	+	+
F7	–	+	+
F8	+	+	+
F9	+	+	+

Table 7
Statistical significant test results among the ABC variants for 50-dimensional functions.

Function	Basic ABC	ABC_{MR} (MR = 0.3)	ABC_{MR} (MR = 0.7)
F1	+	+	+
F2	–	–	–
F3	+	+	+
F4	+	+	+
F5	+	+	+
F6	–	+	+
F7	–	+	+
F8	+	+	+
F9	+	+	+

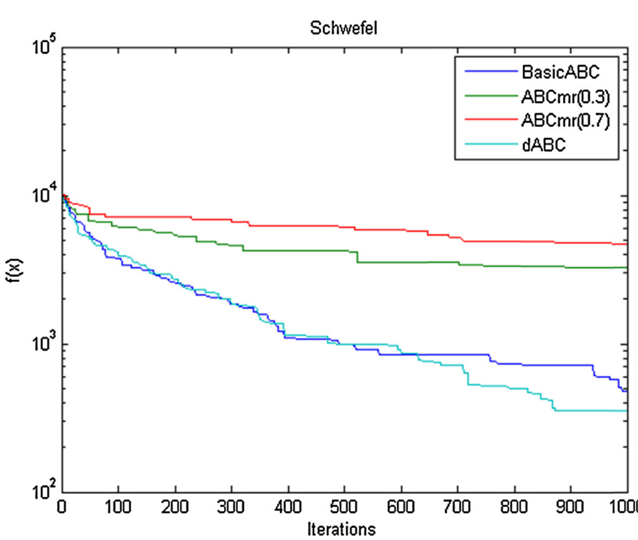

Fig. 10. The convergence graph of the methods on the Schwefel function with 30-D.

dimension of each food source is described a field for direction information and this is used for updating position of food sources. Obtained results show that the proposed approach is better than the other versions of ABC algorithm in terms of solution quality, robustness and convergence characteristics. ABC_{MR} method use the technique based on updating more than one decision variable or dimension at the each iteration of ABC for improving convergence and local search characteristics of basic ABC algorithm. Instead of more than one decision variable, dABC method uses direction information for improving convergence characteristic and local search capability of basic ABC algorithm and this mechanism is better than the other technique in terms of solution quality and convergence to optima according to experiments.

5. Conclusion and future works

In this study, a new version of basic ABC algorithm named as dABC is described, and performance of the proposed approach is investigated on the numeric benchmark functions. Obtained results are compared with basic ABC and ABC_{MR} algorithms and experimental results show that dABC algorithm is better than the other methods in terms of solution quality and convergence characteristics. This is originated from giving the direction information to the bee population. The direction information is used to constrict the search space for obtaining better solutions. Therefore, obtained new solutions guide for the search process and the solution information about search space are shared in the bee population. This version of dABC algorithm uses the update rule of basic ABC algorithm and the other update rules proposed in the literature can be replaced with the basic update rule of ABC algorithm. We will apply the other update rules for dABC algorithm in our future works.

References

[1] M. Dorigo, V. Maniezzo, A. Colorni, Positive Feedback as a Search Strategy, Technical Report No. 91-016, Politecnico di Milano, Italy, 1991.
[2] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proc. of IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942–1948.
[3] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report – TR06, Erciyes University, Turkey, 2005.
[4] T.D. Seeley, The Wisdom of the Hive, Harvard University Press, Cambridge, MA, 1995.
[5] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Inf. Sci. 192 (2012) 120–142.
[6] D. Karaboga, B. Gorkemli, C. Ozturk, D. Karaboga, A comprehensive survey: artificial bee colony (ABC) algorithm and applications, Artif. Intell. Rev. (2012), http://dx.doi.org/10.1007/s10462-012-9328-0.
[7] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, J. Glob. Optim. 39 (2007) 459–471.
[8] D. Karaboga, B. Basturk, On the performance of artificial bee colony (abc) algorithm, Appl. Soft Comput. 8 (2008) 687–697.
[9] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Appl. Math. Comput. 214 (2009) 108–132.
[10] D. Karaboga, B. Akay, Artificial bee colony (abc), harmony search and bees algorithms on numerical optimization, in: 2009 Innovative Production Machines and Systems Virtual Conference, 2009, http://conference.iproms.org/conference/download/4153/76 (accessed 01.08.12).
[11] Z. Dongli, G. Xinping, T. Yinggan, T. Yong, Modified artificial bee colony algorithms for numerical optimization, in: Proc. of 3rd International Workshop on Intelligent Systems and Applications, 2011, pp. 1–4.
[12] P.-W. Tsai, J.-S. Pan, B.-Y. Liao, S.-C. Chu, Enhanced artificial bee colony optimization, Int. J. Innov. Comput. Inf. Control 5 (2009) 5081–5092.
[13] B. Alatas, Chaotic bee colony algorithms for global numerical optimization, Expert Syst. Appl. 37 (2010) 5682–5687.
[14] G. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, Appl. Math. Comput. 217 (2010) 3166–3173.
[15] W.-F. Gao, S.-Y. Liu, A modified artificial bee colony algorithm, Comput. Oper. Res. 39 (2012) 687–697.
[16] A. Barnharnsakun, T. Achalakul, B. Sirinaovakul, The best-so-far selection in artificial bee colony algorithm, Appl. Soft Comput. 11 (2011) 2888–2901.
[17] F. Kang, J. Lie, Z. Ma, Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions, Inf. Sci. 181 (2011) 3508–3531.
[18] D. Karaboga, B. Akay, A modified artificial bee colony (ABC) algorithm for constrained optimization problems, Appl. Soft Comput. 11 (2011) 3021–3031.
[19] M.S. Kıran, M. Gündüz, A novel artificial bee colony-based algorithm for solving the numerical optimization problems, Int. J. Innov. Comput. Inf. Control 9 (2012) 6107–6121.
[20] M.-H. Horng, Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation, Expert Syst. Appl. 38 (2011) 13785–13791.

[21] S.N. Omkar, J. Senthilnath, R. Khandelwal, G.N. Naik, S. Gopalakrishman, Artificial bee colony (ABC) for multi-objective design optimization of composite structures, Appl. Soft Comput. 11 (2011) 489–499.
[22] Y. Liu, X. Ling, G. Liu, Improved artificial bee colony algorithm with mutual learning, J. Syst. Eng. Electron. 23 (2012) 265–275.
[23] W. Gao, S. Liu, L. Huang, A global best artificial bee colony algorithm for global optimization, J. Comput. Appl. Math. 236 (2012) 2741–2753.
[24] W. Gao, S. Liu, A modified artificial bee colony algorithm, Comput. Oper. Res. 39 (2012) 687–697.
[25] W. Gao, S. Liu, L. Huang, A novel artificial bee colony algorithm based on modified search equation and orthogonal learning, IEEE Trans. Syst. Man Cybern. B (2012), http://dx.doi.org/10.1109/TSMCB. 2012.2222373.
[26] D.O. Boyer, C.H. Martinez, N.G. Pedrajas, Crossover operator for evolutionary algorithms based on population features, J. Artif. Intell. Res. 24 (2005) 1–48.
[27] B. Akay, D. Karaboga, Parameter tuning for the artificial bee colony algorithm, in: Proc. of the 1st International Conference on Computational Collective Intelligence, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 608–619.