Zhang and Zhang *EURASIP Journal on Wireless Communications and Networking* (2016) 2016:13
DOI 10.1186/s13638-015-0510-3

EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

A study of artificial bee colony variants for radar waveform design

Xiu Zhang and Xin Zhang*

Abstract

Waveform design and diversity is the technology that allows one or more sensors (e.g., radar or radar sensor networks) on board a platform to automatically change operating parameters, e.g., frequency, gain pattern, and pulse repetition frequency, to meet the varying environments. Optimization problems in the design of radar waveforms such as polyphase code design often bring troubles to designers. This paper proposes to hybridize the migration operator of biogeography-based optimization (BBO) approach with artificial bee colony (ABC) algorithm. Migration operator is able to promote information exchange of solutions in bee colony. This is useful for exploiting good information of searched solutions. Moreover, three state-of-the-art ABC variants are taken for study. A spread spectrum polyphase code design problem is chosen for experiment. The proposed ABCBBO algorithm as well as three state-of-the-art ABC algorithms are applied to solve polyphase code design. Results show that ABCBBO presents the overall best performance amongst the test algorithms. It is also the most reliable one.

Keywords: Polyphase code design, Artificial bee colony, Biogeography-based optimization, Radar system design, Global optimization

1 Introduction

Waveform design and diversity is the technology that allows one or more sensors (e.g., radar or radar sensor networks) on board a platform to automatically change operating parameters, e.g., frequency, gain pattern, and pulse repetition frequency (PRF), to meet the varying environments [1, 2]. It has long been recognized that judicious use of properly designed waveforms, coupled with advanced receiver strategies, is fundamental to fully utilize the capacity of the electromagnetic spectrum. However, it is the relatively recent advances in hardware technology that are enabling a much wider range of design freedoms to be explored [3]. As a result, there are emerging and compelling changes in system requirements such as more efficient spectrum usage, higher sensitivities, greater information content, improved robustness to errors, and reduced interference emissions [4, 5]. The combination of these changes is fueling a worldwide interest in the subject of waveform design and the use of waveform diversity techniques [6]. In this paper, we focus on waveform design such as polyphase code design for radar system.

Swarm intelligence approaches have been successfully applied to handle optimization problems in the design of radar waveforms. In designing radar waveforms using polyphase code method to do pulse compression, the design can be modeled as a min-max nonlinear programming problem. This problem is a NP-hard optimization problem [7] and thus is difficult to solve when problem dimension increases. Recently, tabu search , genetic algorithm (GA), and hybrid harmony search (HS) were used to solve the design problem [7, 8]. Perez-Bellido et al. studied three memetic algorithms for the design problem [9]. Memetic algorithm is a hybrid of a global search metaheuristic approach with a local search algorithm. In [9], fast evolutionary programming, particle swarm optimization, and differential evolution were taken as global search heuristics, while a gradient-guided local search method was used to exploit the solutions found by global search methods. The results show that memetic algorithm is able to obtain better solution than tabu search, GA, and HS.

*Correspondence: ecemark@mail.tjnu.edu.cn
College of Electronic and Communication Engineering, Tianjin Normal University, Tianjin, China

The purpose of this paper is twofold. First, the paradigm of artificial bee colony (ABC) has drawn great attention since it was proposed in 2005. Based on ABC paradigm, a new algorithm is proposed by hybridizing ABC with biogeography-based optimization (BBO). Second, three state-of-the-art ABC variants are studied for dealing with radar system designs. Because the design problems are difficult to handle and effective algorithms are highly desirable, the study in this paper would shed new light on the application of swarm intelligence approaches in radar and solar system designs.

The paper is organized as follows. Section 2 describes standard ABC algorithm and three state-of-the-art ABC variants. Section 3 explains the proposed algorithm in detail. Experimental results are shown in Section 4, and Section 5 gives the conclusion.

2 Standard artificial bee colony and state-of-the-art variants

Besides initialization, the main cycle in standard ABC algorithm contains three phases: employed bee phase, onlooker bee phase, and scout bee phase. Without loss of generality, let the optimization be a minimization throughout the paper.

At the initialization phase, a population of N_p solutions is randomly created within search space Ω. Each solution in population is considered as a food source for honey bees. Each food source can attract honey bees to collect nectar and produce honey. The objective function value of a solution is alike the nectar amounts of a food source. After evaluating all initialized solutions, their fitness values are calculated as follows:

$$fit(\mathbf{x}_i) = \begin{cases} \frac{1}{1+f(\mathbf{x}_i)}, & \text{if } f(\mathbf{x}_i) \geq 0 \\ 1 + |f(\mathbf{x}_i)|, & \text{otherwise} \end{cases}, \tag{1}$$

where $f(\mathbf{x}_i)$ is the function value of solution $\mathbf{x}_i$, $i = 1, 2, \cdots, N_p$. The fitness function in (1) is a strictly decreasing one.

Employed bee phase. A swarm of N_p employed bees is sent out to search around each food source. Usually, one employed bee is assumed to search around one food source. Candidate solution $\mathbf{v}_i$ produced by ith employed bee is implemented as follows:

$$v_{i,j} = \begin{cases} x_{i,j} + \varphi_{i,j}(x_{i,j} - x_{r1,j}), & \text{if } j = j1 \\ x_{i,j}, & \text{otherwise} \end{cases}, \tag{2}$$

where $v_{i,j}$, $x_{i,j}$ and $x_{r1,j}$ denote the j$^{\text{th}}$ variable of $\mathbf{v}_i$, $\mathbf{x}_i$ and $\mathbf{x}_{r1}$, respectively; $\varphi_{i,j} \in [-1, 1]$ is a random number. In (2), $j1 \in [1, D]$ is a random integer and D is the number of variables. $\mathbf{x}_i \neq \mathbf{x}_{r1}$. After the evaluation of $\mathbf{v}_i$, greedy selection is performed between $\mathbf{v}_i$ and $\mathbf{x}_i$. The winner survives and substitutes old solution $\mathbf{x}_i$ as in (3).

$$\mathbf{x}_i = \begin{cases} \mathbf{v}_i, & \text{if } f(\mathbf{v}_i) < f(\mathbf{x}_i) \text{or } fit(\mathbf{v}_i) > fit(\mathbf{x}_i) \\ \mathbf{x}_i, & \text{otherwise} \end{cases}, \tag{3}$$

Onlooker bee phase. A swarm of N_p onlooker bees is sent out. The behavior of onlooker bees differs from that of employed bees. An onlooker bee chooses a food source depending on the quality/goodness of food sources. High-quality food sources would attract more onlooker bees, while low-quality sources have low chances to be selected. The quality of food sources can be defined according to their fitness. This behavior is realized by first calculating a probability for each solution as in (4); then, roulette wheel selection is used to choose a food source. The probability value of $\mathbf{x}_i$ is calculated by the following equation:

$$p_i = \frac{fit_i}{\sum_{j=1}^{N_p} fit_j}. \tag{4}$$

After choosing a food source, an onlooker bee produces a modification following (2). Then, a greedy selection is performed between the newly produced solution and the old one, and the winner survives as at employed bee phase.

Scout bee phase. Scout bees are sent out occasionally looking for new food sources. If a source has been continuously searched for a long time, its nectar amount did not decrease and it might be abandoned by honey bees. In this case, scout bees are send out to explore for new sources. In standard ABC, a predefined parameter called *limit* is set to determine if a source should be abandoned. In case a solution cannot be improved after *limit* times, then it is abandoned and the associated employed bee becomes a scout. The abandoned solution is replaced by a randomly created solution. Note that the number that a solution (food source) is searched but not improved is counted as in (5).

$$l_i = \begin{cases} 0, & \text{if } f(\mathbf{v}_i) < f(\mathbf{x}_i) \text{or } fit(\mathbf{v}_i) > fit(\mathbf{x}_i) \\ l_i + 1, & \text{otherwise} \end{cases}. \tag{5}$$

Standard ABC contains two parameters: N_p and *limit*. The impact of both parameters on the algorithm has been studied on a set of benchmark functions. Studies show that proper settings of both parameters is related with the properties of problems [10, 11]. Parameter *limit* is not sensitive to the difficulty of problems; *limit* $= 0.5NpD$ could solve many benchmark functions and thus becomes default setting [10, 12].

In standard ABC, both employed bees and onlooker bees use the same search equation (2). This formula could not fully reflect the behavior of onlooker bees. Thus, quick ABC (qABC) algorithm is introduced for mimicking the behavior of onlooker bees [13].

$$v_{i,j} = \begin{cases} x_{i,j}^{nbest} + \phi_{i,j}(x_{i,j}^{nbest} - x_{r1,j}), & \text{if } j = j1 \\ x_{i,j}, & \text{otherwise} \end{cases}, \tag{6}$$

where $\mathbf{x}_i^{nbest}$ represents the best solution amongst the neighbors of $\mathbf{x}_i$ and itself. In (6), Euclidean distance is used to define neighborhood of a solution. Denote $d(i,m)$ as the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_m$. The mean Euclidean distance of $\mathbf{x}_i$ is computed as in (7).

$$md_i = \frac{\sum_{m=1}^{N_p} d(i,m)}{N_p - 1}. \tag{7}$$

Then, the neighbor of $\mathbf{x}_i$ is defined as the set of solutions with distance $d(i,m) \neq r \times md_i$. Parameter r is "neighborhood radius." When $r = 0$, qABC reduces to standard ABC. Experimental results show that qABC improves the convergence performance of stand ABC when r is properly set. Note that r is suggested to be 1 in qABC after an empirical experiment [13].

In [14], a new ABC algorithm is proposed which includes a one-position inheritance (OPI) scheme and an opposite directional search method (OOABC). In the OPI scheme, a candidate solution $\mathbf{v}_i$ accepts a one-position transfer as follows:

$$v_{i,j} = \begin{cases} x_{i,j} + \varphi_{i,j}(x_{i,j} - x_{r1,j}), & \text{if } j = j1 \\ x_{r2,j}, & \text{if } j = j2 \\ x_{i,j}, & \text{otherwise} \end{cases}, \tag{8}$$

where $j1, j2 \in [1, D]$ are random integers but $j1 \neq j2$. The OPI scheme in (8) substitutes the search equation at the employed bee phase in standard ABC. Opposite directional (OD) search method works on onlooker bee phase. If an onlooker bee could not detect a better solution using (2), OD search is activated as in (9).

$$v_{i,j}^{o} = \begin{cases} x_{i,j} - \varphi_{i,j}(x_{i,j} - x_{r1,j}), & \text{if } j = j1 \\ x_{i,j}, & \text{otherwise} \end{cases}, \tag{9}$$

Observed from (2) and (9), $v_{i,j}$ and $v_{i,j}^{o}$ are symmetric points with respect to $x_{i,j1}$. OOABC has been taken to deal with electromagnetic device designs with promising results obtained.

In [15], history-driven ABC (HdABC) is proposed to find high-quality regenerated solutions. HdABC stores the entire search history information in a binary space partitioning (BSP) tree and applies a guided anisotropic search (GAS) module to produce a candidate solution. GAS works in two ways: local restart and global restart. The probability of doing local restart is

$$p_l = igen/mgen, \tag{10}$$

where *igen* is current number of generations, *mgen* is total number of generations in (10), p_l is the local restart probability, and global restart probability is $1 - p_l$. Solution $\mathbf{v}$ produced by GAS is

$$\mathbf{v} = \begin{cases} \mathbf{x} - \alpha(\mathbf{p} - \mathbf{x}), & \text{if } r1 < p_l \\ \mathbf{x}^{rnd} - \alpha\left(\mathbf{p}^{rnd} - \mathbf{x}^{rnd}\right), & \text{otherwise} \end{cases}, \tag{11}$$

where $\mathbf{x}$ is the abandoned food source and $\mathbf{p}$ is the local optimum of $\mathbf{x}$ traced by tree memory in (11); $\mathbf{x}^{rnd}$ is a randomly generated solution and $\mathbf{p}^{rnd}$ is the associated local optimum traced by BSP tree. Experimental results show that HdABC significantly outperforms ABC in most test problems. When the number of scout bees increases, HdABC performs even better.

qABC, OOABC, and HdABC are state-of-the-art ABC variants. In this paper, they are taken to deal with radar system design problems.

3 The proposed algorithm

Since ABC does not use an operator like crossover as in GA or differential evolution (DE), the spread of useful information amongst solutions is not very effective [16]. This issue is resolved by integrating information migration scheme of biogeography-based optimization (BBO) approach as follows.

3.1 Information migration scheme

BBO is inspried from biogeography [17]. It characterizes how species migrate from one island to another, how new species arise, and how species become extinct. In BBO, each solution is considered as a habitat with a habitat suitability index (HSI). HSI is utilized to measure the quality of solutions. Good solutions correspond to habitats with high HSI, while poor solutions have low HSI. The migration is like this: high HSI solutions tend to disperse their features (emigration), while low HSI solutions accept new features from high HSI solutions (immigration) and seldom share their features with high HSI solutions.

A typical implementation of migration model in BBO is described as follows. Each solution has its own immigration rate λ and emigration rate μ. A good solution has higher μ and lower λ, and vice versa. λ and μ are functions of the number of species in a habitat. All solutions in the population are sorted by their function values from the best to the worst. The immigration and emigration rates of the sorted solutions are calculated as follows:

$$\lambda_i = I\left(1 - \frac{i}{n}\right), \tag{12}$$

$$\mu_i = E\left(\frac{i}{n}\right), \tag{13}$$

where I is the maximum immigration rate and E is the maximum emigration rate. In BBO, E is usually set to 1 and n is set to N_p. When $I = E$, combining (12) and (13), we have

$$E = \lambda_i + \mu_i \tag{14}$$

Immigration and emigration are collectively called migration. With the migration operator, information can be shared amongst the solutions in the BBO algorithm. Thus, BBO is good at exploiting the information of existing solutions.

3.2 Hybrid ABC with migration operator

As mentioned in the introduction, ABC does not do well in exploiting the existing information of solutions. On the other hand, BBO is good at exploiting the information of existing solutions. Based on these observations, we propose to hybridize ABC with migration operator of the BBO algorithm to combine their strengths, called ABCBBO. This algorithm combines the migration operator with the employed bee phase of ABC. In addition, Rechenberg's one-fifth success rule is employed to effectively control the adaptation of immigration probability.

Algorithm 1 Pseudo code of the ABCBBO algorithm

```
Input: $f(\cdot)$, $D$, $\mathbf{x}^{min}$, $\mathbf{x}^{max}$, $N_p$, $limit$
Output: the best solution obtained by the algorithm
randomly create $N_p$ solutions $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_p}$;
evaluate function values of solutions and their fitness by (1);
set counter $l_i, i = 1, 2, \cdots, N_p$ to 0;
repeat
    send out employed bees (see Algorithm 2);
    send out onlooker bees depending on their nectar amounts by (2);
    evaluate candidate solutions and their fitness by (1);
    do greedy selection by (3) and update $l_i$ by (5);
    send out scout bees if $l_i$ reaches $limit$;
    evaluate candidate solutions and their fitness by (1);
    reset $l_i$ to 0;
until termination criteria are met
```

ABCBBO is described in Algorithm 1. Except employed bee phase, ABCBBO is the same as standard ABC. In hybridized employed bee phase (see Algorithm 2) of ABCBBO, the immigration and emigration rates of all solutions are calculated by (12) and (13). A new candidate solution $\mathbf{v}_i$ could obtain new features from different solutions. It is implemented as follows:

$$v_{i,j} = \begin{cases} x_{i,j} + \varphi(x_{i,j} - x_{r1,j}) & \text{if } j = j1 \\ x_{r2,j} & \text{if } rand(0,1) < \lambda_i/\sigma and\ j \neq j1 \\ x_{i,j} & \text{otherwise} \end{cases}, \tag{15}$$

where σ is utilized to control the immigration rate. $\mathbf{x}_{r2}$ with $r2 \neq i$ is selected using roulette wheel selection method depending on the emigration rates of solutions except $\mathbf{x}_i$. A colony of N_p candidate solutions is attained by repeating this procedure N_p times. Each candidate solution $\mathbf{v}_i$ is evaluated, and then compared with $\mathbf{x}_i$. If $f(\mathbf{v}_i) < f(\mathbf{x}_i)$, it means that a more suitable solution is found and we regard this as a success; otherwise, we regard the hybridized employed bee phase fails to improve solution $\mathbf{x}_i$.

Algorithm 2 Pseudo-code of the employed bee phase in ABCBBO

```
set $S_c = 0$, $\sigma = 0.5D$ and $c_d = 0.82$
calculate $\lambda_i$ and $\mu_i$ for each solution in the bee colony
for $i = 1$ to $N_p$ do
    randomly choose $j1 \in [1, D]$ and $r1 \in [1, N_p]$
    for $j = 1$ to $D$ do
        if $j == j1$ then
            $v_{i,j1} = x_{i,j1} + \varphi(x_{i,j1} - x_{r1,j1})$
        else if $rand(0,1) < \lambda_i/\sigma$ then
            choose a solution $x_{r2}$ with $r2 \neq i$ using
            roulette wheel selection method based
            on emigration rates $\mu_i, i = 1, 2, \ldots, N_p$
            $v_{i,j} = x_{r2,j}$
        else
            $v_{i,j} = x_{i,j}$
        end if
    end for
    evaluate candidate solutions and their fitness by (1);
    if $f(\mathbf{v}_i) < f(\mathbf{x}_i)$ then
        replace $\mathbf{x}_i$ by $\mathbf{v}_i$
        $S_c$++
    end if
end for
if $S_c/N_p < 1/5$ then
    $\sigma = c_d \cdot \sigma$
else if $S_c/N_p > 1/5$ then
    $\sigma = \sigma/c_d$
end if
reset $S_c = 0$
```

After greedy selection step, the number of successes is counted and denoted by S_c. Then, σ is adapted based on the one-fifth success rule as shown below.

$$\sigma = \begin{cases} c_d \cdot \sigma & \text{if } S_c/N_p < 1/5 \\ \sigma/c_d & \text{if } S_c/N_p > 1/5 \\ \sigma & \text{otherwise} \end{cases} \tag{16}$$

where c_d is a decay factor. Schwefel gives reasons to use $c_d = 0.82$ for the adaptation [18].

In the hybridized employed bee phase, σ is used to control the adaptation of immigration rate. As shown in (16), if more than 20 % solutions in the bee colony are improved, which means the employed bees have found nectar-rich food sources and these food sources deserves more search efforts, the value of σ will increase. From (15), the probability of $\mathbf{v}_i$ immigrating an element from other solutions becomes smaller. Thus, the employed bees can focus on searching around solutions instead of immigrating from other solutions. In contrast, if less than 20 % solutions are improved, σ will decrease and the probability of $\mathbf{v}_i$ immigrating from other solutions becomes bigger. Thus, the employed bees interact with each other for finding promising solutions. λ_i/σ of (15) is truncated to 1 in case that it is larger than 1, since the probability cannot be larger than 1. In Algorithm 2, σ is initialized to $0.5D$, which is determined based on experiment on toy functions.

3.3 Computational complexity

The computational complexity of the proposed hybridized employed bee phase is analyzed in terms of average CPU time (in seconds). The analysis is performed on synthetic mathematical functions over 25 independent runs. CPU time is counted by executing the ABCBBO algorithm until a predefined number of function evaluations is met. Note that the time of function evaluation is excluded from the CPU time. Typically, the percentage of the time that the ABCBBO algorithm spends on the proposed hybrid employed bee phase is about 17 %. The increase of computation time is acceptable in case that the proposed algorithm could attain promising solutions.

4 Experimental evaluation and analysis

In this section, the proposed algorithm is applied to deal with a polyphase code design problem.

4.1 Polyphase code design problem

Based on the attributes of aperiodic autocorrelation function and the assumption of coherent radar pulse processing in the receiver, polyphase code design is modeled as a min-max nonlinear nonconvex continuous optimization problem [9, 19]. This model contains numerous local optima and is expressed as follows:

$$\begin{aligned} \min_{x\in\Omega} \quad & f(\mathbf{x}) = \max\{\varphi_1(\mathbf{x}), \ldots, \varphi_{2m}(\mathbf{x})\} \\ s.t. \quad & \varphi_{2i-1}(\mathbf{x}) = \sum_{j=i}^{D} \cos\left(\sum_{k=|2i-j-1|+1}^{j} x_k\right),\ i = 1, \ldots, D \\ & \varphi_{2i}(\mathbf{x}) = 0.5 + \sum_{j=i+1}^{D} \cos\left(\sum_{k=|2i-j|+1}^{j} x_k\right),\ i = 1, \ldots, D-1^{,} \\ & \varphi_{m+i}(\mathbf{x}) = -\varphi_i(\mathbf{x}),\ i = 1, 2, \ldots, m \\ & \Omega = \{(x_1, \ldots, x_n) \in \Re^D | 0 \le x_j \le 2\pi, j = 1, \ldots, D\} \end{aligned} \tag{17}$$

where $m = 2D - 1$. In this model, variables stands for the differences of symmetrized phase. The problem objective is to minimize the biggest module amongst the samples of autocorrelation function. Previously, this model has been studied with $2 \le D \le 15$ [9], [8], while the number of variables is set $D = 20$ in this experiment, which makes the problem instance harder for testing state-of-the-art ABC algorithms.

4.2 Experimental configuration

qABC, HdABC, and OOABC as well as the proposed ABCBBO algorithm are chosen for study. The parameter setting for all ABC variants is N_p=25, $limit = 0.5N_pD$ as in [10, 11]. Additionally, $r = 1$ in qABC as suggested in [13]. All algorithms are implemented in Matlab and simulated on a PC with 3.4 GHz 4-core CPU and 4 GB of memory. Each problem is conducted 25 independent runs. The source code of ABCBBO can be obtained from the first author upon request.

4.3 Experimental results

First, problem (17) is tested with the maximum number of function evaluations (MFE) at 50,000. The experimental results are shown in Table 1. This table presents the best function values in four kinds of statistics, which are minimum (min), median (med), maximum (max), and standard deviation (std). The rank of each algorithm (listed in parentheses) associated with each metric is also given in this table. For the polyphase code design problem, HdABC obtains the minimum function value compared with the other algorithms over 25 trials. As to med, max, and std metrics, ABCBBO outperforms others and ranks first. To gain a statistical view of the results, Mann-Whitney U test (U test) is utilized with significant level $\alpha = 0.05$. The p-values comparing ABCBBO with each of the other algorithms is given in the last column of Table 1. Observe that

Table 1 Statistics of the best function values found by ABC algorithms with MFE = 50000

Algorithm	min	med	max	std	p
qABC	1.1175(3)	1.3454(3)	1.5769(3)	0.0933(2)	0.0016
HdABC	0.9179(1)	1.3496(4)	1.6077(4)	0.1683(4)	0.1206
OOABC	1.0522(2)	1.3335(2)	1.4714(2)	0.1156(3)	0.0808
ABCBBO	1.1313(4)	1.2959(1)	1.4237(1)	0.0776(1)	

ABCBBO performs better than qABC at 99.99 % confidence, better than OOABC at 90 % confidence, and better than HdABC at 85 % confidence.

Second, we repeat the experiment with a larger budget and the design problem is tested with MFE = 150,000. The results are presented in Table 2. The organization of Table 2 is similar to Table 1. HdABC still achieves the best min result compared with other algorithms over 25 trials. As to med, max, and std metrics, ABCBBO performs the best and ranks first. According to U test, ABCBBO outperforms HdABC at 99.95 % confidence. It is better than qABC at 90 % confidence and at 70 % confidence better than OOABC.

Based on the above two experiments, we conclude that the proposed ABCBBO is a very competitive algorithm. It is more reliable than qABC, HdABC, and OOABC in repeated trials. HdABC finds the best solution amongst the four algorithms in repeated runs, whereas its variance is larger than other algorithms, which indicates that it is not very reliable.

Table 3 lists the min, med, max, and std of the computational time (in seconds). Clearly, ABCBBO costs the least running time amongst the test algorithms. The time consumed by HdABC and OOABC is about three times longer than that of qABC and ABCBBO.

5 Conclusions

To solve the design of radar waveforms such as polyphase codes design, this paper proposes to combine the migration operator of biogeography-based optimization (BBO) approach with the employed bee phase of artificial bee colony (ABC). The resulting algorithm is called ABCBBO. The proposed algorithm has reasonable physical meaning. Moreover, we study three state-of-the-art ABC variants in literature on the design problem. They are qABC, HdABC, and OOABC. Simulation results show that HdABC attains the best solution amongst the four algorithms in repeated trials. The proposed ABCBBO presents overall best performance. It is also more reliable than other algorithms. Furthermore, ABCBBO is faster than others in terms of computer running time (in seconds).

Table 2 Statistics of the best function values found by ABC algorithms with MFE = 150,000

Algorithm	min	med	max	std	*p*
qABC	1.0419(4)	1.2694(3)	1.3857(2)	0.1016(2)	0.0842
HdABC	0.9179(1)	1.2767(4)	1.4173(3)	0.1203(4)	0.0328
OOABC	0.9665(2)	1.2517(2)	1.4266(4)	0.1113(3)	0.2859
ABCBBO	1.0154(3)	1.2227(1)	1.3702(1)	0.0779(1)	

Table 3 Statistics of the computational time consumed by ABC algorithms with MFE = 150,000

Algorithm	min	med	max	std
qABC	4.784	4.793	4.802	0.004
HdABC	14.319	14.468	15.834	0.361
OOABC	17.119	17.151	17.847	0.184
ABCBBO	3.047	3.063	3.222	0.034

Through the study in this paper, we could conclude that information exchange at employed bee phase is very useful to enhance the performance of ABC. History-based scout bee could find good solution but not reliable. A future direction may hybridize history-based scout bee method and the proposed employed bee phase.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This research was supported in part by the Tianjin Thousand Youth Talents Plan Project of Tianjin Normal University (ZX110023), and the Applied Basic Research Program of Tianjin (15JCYBJC51500, 15JCYBJC52300).

Received: 27 September 2015 Accepted: 20 December 2015
Published online: 12 January 2016

References

1. Q Liang, X Cheng, S Samn, NEW: Network-enabled electronic warfare for target recognition. IEEE Trans Aerosp. Electron. Syst. **46**(2), 558–568 (2010)
2. Q Liang, X Cheng, S Huang, D Chen, Opportunistic sensing in wireless sensor networks: theory and application. Comput. IEEE Trans. **63**(8), 2002–2010 (2014)
3. S Singh, Q Liang, D Chen, L Sheng, Sense through wall human detection using UWB radar. EURASIP J. Wirel. Commun. Netw. **2011**(1), 503–520 (2011)
4. Q Liang, Situation understanding based on heterogeneous sensor networks and human-inspired favor weak fuzzy logic system. Syst. J. IEEE. **5**(2), 156–163 (2011)
5. Q Liang, Radar sensor wireless channel modeling in foliage environment: UWB versus narrowband. Sensors J. IEEE. **11**(6), 1448–1457 (2011)
6. L Xu, Q Liang, Zero correlation zone sequence pair sets for MIMO radar. Aerosp. Electron. Syst. IEEE Trans. **48**(3), 2100–2113 (2012)
7. J Del Ser, S Gil-Lopez, A Perez-Bellido, S Salcedo-Sanz, JA Portilla-Figueras, in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*. On the application of a novel hybrid harmony search algorithm to the radar polyphase code design problem (IEEE Computer Society, Budapest, Hungary, 2011), pp. 1–5
8. S Gil-Lopez, JD Ser, S Salcedo-Sanz, AM Perez-Bellido, JM Cabero, JA Portilla-Figueras, A hybrid harmony search algorithm for the spread spectrum radar polyphase codes design problem. Expert Syst. Appl. **39**(12), 11089–11093 (2012)
9. AM Perez-Bellido, S Salcedo-Sanz, EG Ortiz-Garcia, JA Portilla-Figueras, F Lopez-Ferreras, A comparison of memetic algorithms for the spread spectrum radar polyphase codes design problem. Eng. Appl. Artif. Intell. **21**(8), 1233–1238 (2008)
10. D Karaboga, B Basturk, On the performance of artificial bee colony (ABC) algorithm. Appl. Soft Comput. **8**(1), 687–697 (2008)
11. K Diwold, A Aderhold, A Scheidler, M Middendorf, Performance evaluation of artificial bee colony optimization and new selection schemes. Memetic Comput. **3**(3), 149–162 (2011)
12. X Zhang, X Zhang, SL Ho, WN Fu, A modification of artificial bee colony algorithm applied to loudspeaker design problem. IEEE Trans. Magn. **50**(2), 737–740 (2014)

13. D Karaboga, B Gorkemli, A quick artificial bee colony (qABC) algorithm and its performance on optimization problems. Appl. Soft Comput. **23**, 227–238 (2014)
14. X Zhang, X Zhang, SY Yuen, SL Ho, WN Fu, An improved artificial bee colony algorithm for optimal design of electromagnetic devices. IEEE Trans. Magn. **49**(8), 4811–4816 (2013)
15. X Zhang, Z Wu, in *Advances in Swarm and Computational Intelligence*. Lecture Notes in Computer Science, ed. by Y Tan, Y Shi, F Buarque, A Gelbukh, S Das, and A Engelbrecht. An artificial bee colony algorithm with history-driven scout bees phase, vol. 9140 (Springer, Berlin, Heidelberg, 2015), pp. 239–246
16. D Karaboga, B Gorkemli, C Ozturk, N Karaboga, A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artif. Intell. Rev. **42**(1), 21–57 (2014)
17. D Simon, Biogeography-based optimization. IEEE Trans. Evol. Comput. **12**(6), 702–713 (2008)
18. H-P Schwefel, *Numerical Optimization of Computer Models*. (Wiley, Chichester, 1981)
19. ML Dukic, ZS Dobrosavljevic, A method of a spread spectrum radar polyphase code design. IEEE J. Sel. Areas Comm. **8**, 743–749 (1990)