Hu et al. *EURASIP Journal on Wireless Communications and Networking* (2020) 2020:40
https://doi.org/10.1186/s13638-020-1659-y

EURASIP Journal on Wireless Communications and Networking

RESEARCH | Open Access

UAV-aided networks with optimization allocation via artificial bee colony with intellective search

Bing Hu[1], Zhixin Sun[1], Hanshu Hong[1] and Jian Liu[2*]

Abstract

In this paper, we consider a strong global search algorithm which exhibits strong exploration ability in unmanned aerial vehicle (UAV)-aided networks. UAVs in wireless communication have aroused great interest recently due to its low cost and flexibility in providing wireless connectivity in areas without infrastructure coverage. Artificial bee colony algorithm is a powerful approach for such a scene. However, due to its one-dimensional and greedy search strategy, it still suffers slow convergence speed. In the traditional version, three types of bees, including employed bees, onlooker bees, and scouts, are employed and they cooperate with each other to find the best food source position. Though different roles, these three types of bees play, there is no difference of division within the internal of each type of bees. Considering this phenomenon, this paper proposes a modified artificial bee colony algorithm with intellective search and special division (ABCIS) to enhance its performance, where different employed bees and different onlooker bees use different search strategies to search for food sources. Besides, the greedy selection method is also abandoned and the food sources' positions are updated at each iteration. Under this circumstance, the whole population's experience is fully utilized to guide bee's search. Finally, to testify the proposed algorithms' competitiveness, a series of benchmarks are adopted, and the experimental results demonstrate its superior performance among other state-of-the-art algorithm in UAV-aided networks.

Keywords: Unmanned aerial vehicles, Artificial bee colony, Global search, Convergence speed, Intellective search

Introduction

Unmanned aerial vehicles (UAVs) are being increasingly used as an innovative method to enable robust and reliable communication networks [1–4]. Due to the high mobility, flexibility, and good channel condition, UAV communication has been an emerging technique which can help to achieve better performances [5–7]. In order to deploy a UAV-aided network, it is important to model the reliability and coverage of the airborne platform. In particular, recent research has shown that the location and height of drones can severely impact the reliability of air-to-ground (A2G) links [8]. Furthermore, UAVs have great potential to be employed in long-range backscatter networks to both support more devices and increase the network efficiency and reliability. Consequently, optimizing the 3-D location of the data collecting UAV is very critical in order to provide reliable communication for backscatter devices which operate in the presence of very low power signals [9–11].

In literature, UAV communication has been extensively studied for boosting the capacity and coverage of existing wireless networks [12–15]. Li et al. [12] and Wu and Wang [13] investigated the 2-D and 3-D placement problems of a single UAV, respectively. In [14], the authors aim to optimize the UAV's altitude and antenna beamwidth for throughput maximization in three different communication models without considering the impact of altitude and beamwidth on the flight time. In [15], an optimum placement of multiple UAVs for maximum number of covered users is investigated. However, evolutionary algorithms have not been used for UAV-aided networks. In nature, creatures conduct comprehensive tasks by swarm cooperation, and each individual's simple behavior could indicate powerful capability because of the interaction

*Correspondence: by.liujian@gmail.com
[2]College of Information Engineering, Nanjing University of Finances and Economics, 210003, Nanjing, China
Full list of author information is available at the end of the article

among its swarm. Inspired by this phenomenon, evolutionary algorithms are gradually developed and they have received extensive attention in recent years.

Different from the traditional mathematical optimization algorithms, evolutionary algorithms could be applied to extensive problems because of its no requirement of problem characteristic. Among these algorithms, artificial bee colony (ABC) [16] has been demonstrated powerful competitiveness, and due to its ease of implementation, it has been successfully applied to many real-world problems, such as industrial systems [17–20], image processing [21–24], and so on. Compared with other evolutionary algorithms, especially the particle swarm optimization (PSO) and differential evolution (DE), ABC shows strong global search ability yet poor convergence speed [25–30]. To enhance its performance, plenty of works have been carried out. Zhu and Kwong [25] raised that ABC shows slow convergence speed because of its blindness search equation; thus, they introduced the best position found so far by the whole population to guide each bee's search. With the same motivation, Karaboga and Gorkemli [26] introduced the best position found so far by a newly defined neighborhood to direct onlooker bees' search. Considering oscillation phenomenon as exhibited in [25], Gao et al. [27] modified the search equation from three items into two items, which utilized two randomly selected individuals, and then they cooperated this search equation with orthogonal learning scheme to accelerate the convergence speed and further enhance the performance. Kiran et al. [28] introduced an integration of five different search equations for various optimization problems with diverse characteristics, utilizing an adaptive selection strategy. Xianneng Li and Guangfei Yang [29] kept a record of individuals' past successful search experience, and utilized them to guide the future search, and attained superior optimization performance.

In this paper, an intellective search strategy to optimize the 3-D location of the aerial base stations under various scenarios is proposed. Regarding the deficiency of this strategy, a new division for updating the best food source's position is endowed to the corresponding employed bees and onlooker bees. Besides, in the proposed ABCIS algorithm, the greedy selection mechanism is abandoned, and each food source's position is updated at each iteration; thus, the scout's role is not necessary and it is eliminated in the proposed algorithm of this paper.

The remaining part of this paper is organized as follows: Section 2 presents the traditional methods via artificial bee colony. The detailed information of the proposed ABCIS algorithm is described in Section 2. In Section 2, comprehensive experiments and discussions with the purpose of demonstrating ABCIS's effectiveness are conducted. Section 2 concludes the paper.

Methods

First proposed by Karaboga in 2005, artificial bee colony algorithm divides individuals into three groups—employed bees, onlooker bees, and scouts. Each type of bees shares the same purpose of locating the food source with maximum nectar. In the searching process, employed bees take charge of making rough search in the search space, and onlooker bees hold the task of making fine tuning around the superior food sources. For the scouts, they work for jumping out of local optima and maintaining the algorithm's exploration ability. To be more specific, this paper explains them one by one.

Food sources

Let NP denote the population size, then the number of food source is calculated as $\mathrm{NP}/2$. At the beginning of artificial bee colony algorithm, these $\mathrm{NP}/2$ food sources are initialized randomly, as shown in Eq. (1):

$$\mathrm{Food}_{ij} = \mathrm{Lower}_j + \mathrm{rand}\,(0,1) \bullet \left(\mathrm{Upper}_j - \mathrm{Lower}_j\right), \quad (1)$$

where i denotes the ith food source, and j presents the jth dimension, $i = 1,2,...,\mathrm{NP}/2$, $j = 1,2,...,D$.Lower and Upper represent the minimum and maximum bounds of the search space, respectively. $\mathrm{rand}(0,1)$ randomly generates a real value within the range of (0,1).

Employed bees

The number of Employed bees is equal to that of the food source, and it takes up half of the colony. For each employed bee, it updates food source's position by using Eq. (2):

$$\mathrm{trial}_{ij} = \mathrm{Food}_{ij} + \mathrm{rand}\,(-1,1) \bullet \left(\mathrm{Food}_{rj} - \mathrm{Food}_{\mathrm{ij}}\right), \quad (2)$$

where $r \neq i$, and r is a randomly selected integer within the range of $(1,\mathrm{NP}/2)$, $\mathrm{rand}(-1,1)$randomly generates a real value within the range of $(-1,1)$.

After a new vector $\mathrm{trial}_{\mathrm{i}}$ being generated, a greedy selection mechanism is applied between $\mathrm{trial}_{\mathrm{i}}$ and Food_i, and then the corresponding object value f_i and fitness value is computed. f_i can be calculated by the optimization problem (if it is a minimum problem), and F_i can be calculated by Eq. (3):

$$F_i = \begin{cases} 1/(1+f_i) & \text{if } f_i \geq 0 \\ 1+\left|f_i\right| & \text{if } f_i < 0 \end{cases} \quad (3)$$

Onlooker bees

When all the employed bees finish their works, onlookers start their work. They calculates each food source's selection probability by Eq. (4):

$$p_i = \frac{F_i}{\sum_1^{\mathrm{NP}} F_i} \quad (4)$$

Then the roulette wheel selection mechanism is used to help onlookers to select a food source and make exploitation around its neighbor region. In this part, the search equation is the same as Eq. (2).

Scouts

When a food source cannot be improved for a successive number of generation, it will be abandoned and be randomly initialized by a scouts in the search space.

Artificial bee colony with intellective search and special division

In the traditional artificial bee colony, both employed bees and onlooker bees update their corresponding food source by learning from a randomly selected neighbor. Besides, the greedy selection strategy makes all the food sources' position get more and more optimal, which drops out much search experience. In the proposed ABCIS algorithm, food sources' positions are updated at each generation, and they are updated by two elite positions with better fitness value. In mathematical expression, it can be presented as:

$$\mathrm{Food}_{ij} = \frac{\mathrm{Alpha}_j + \mathrm{Beta}_j}{2} + \mathrm{SR}\begin{pmatrix} (-1)^{n1}\left(c_1\mathrm{Alpha}_j - \mathrm{rand}\,(0,1)\,\mathrm{Food}_{ij}\right) \\ +(-1)^{n2}\left(c_2\mathrm{Beta}_j - \mathrm{rand}\,(0,1)\,\mathrm{Food}_{ij}\right) \end{pmatrix} \tag{5}$$

where Alpha is the food source's position with best fitness value. In the proposed ABCIS algorithm, since the food sources are updated at each iteration, the best position found so far in history needed to be recorded and used to guide other bees' search. Under this circumstance, this paper learns from the PSO algorithm and records each food source's personal best position as pbest. When considering the selection of Beta, we firstly randomly select a food source $r\,(r \neq i)$ from the colony, then compare the corresponding pbest_r and pbest_i , and the better vector (with better fitness value) is assigned to Beta. Thus the search strategy illustrated in Eq. (5) is guided by two elite vectors, which will accelerate the convergence speed. $n1$ and $n2$ are two integers, they can be 0 or 1, they are selected randomly and independently. SR is the success rate. At the beginning of the search state, SR is larger and then the step size is larger, which will help the swarm to execute global search and open up extensive unknown regions. At the later state of optimization, the SR will be small and it helps bees to make fine tuning around the potential global optimum, which will enhance the solution accuracy. For the parameters c_1 and c_2 , they are generated by:

$$c_1, c_2 = \frac{\mathrm{rand} \bullet \mathrm{rand}}{\mathrm{rand}}, \tag{6}$$

where rand means uniform random generator, and c_1 and c_2 are generated independently.

Besides, by using Eq. (6) to generate these two parameters, there may be some undesired values. To investigate this characteristic, we adopt the Monte Carlo method [31–33]. In this experiment, the generator $\frac{\mathrm{rand}\bullet\mathrm{rand}}{\mathrm{rand}}$ is used to generate random values, and each trial are repeated for 10,000 times. That is to say, 10 000 numbers are generated. Then, their density distribution is plot, as shown in Fig. 1.

As Fig. 1 indicates, values may exceed 1000, which may not be desired in the searching process. Thus, this paper truncates them into the range of $[0, 2]$ (when the generated value is out of that range, it will be deleted and regenerated) (Fig. 2). Figure 2 shows the density distribution of the 10,000 randomly generated values. Of these numbers, More than 85% of 1000 numbers are within the range of $[0, 1]$, which is the same to the range of rand, and there are also numbers larger than 1. Figure 2 shows that individuals could learn more from the selected elite vectors rather than itself. The settings of $[0, 2]$ and the effectiveness will be discussed in Section 2 by numerical experiments.

In Eq. (5), all the dimensions are updated simultaneously at each generation. By using Eq. (5) to generate new positions, employed bees and onlooker bees will be more intellective. We further consider the case when the current food source is the global best position, then both Alpha and Beta vectors equals to Food_i itself. Then, Eq. (5) can be rewrote as:

$$\mathrm{Food}_i^j = \left[1 + (-1)^{n1}\,(c_1 - k_3) + (-1)^{n2}\,(c_2 - k_4)\right] \cdot \mathrm{Food}_i^j \tag{7}$$

In this case, Food_i will lose the ability of learning from others. Thus, a special division for the best food source should be assigned. Then, Eq. (5) cannot generate new positions, which will cause evaluation's waste. Thus, a new division for the best food source should be assigned. In ABCIS algorithm, it uses the following equation to update the best food source's position:

$$\mathrm{trial}_{ij} = \mathrm{Food}_{mj} + (-1)^{n3}\left(c_3\mathrm{Food}_{mj} - \mathrm{rand} \bullet \mathrm{Food}_{ij}\right) \tag{8}$$

where Food_m is a randomly selected food source and it is different from Food_i. Similar to n_1 and n_2 in Eq. (5), n_3 is also a integer number selected from the collection of $\{0, 1\}$. And similar to c_1 and c_2 , c_3 is also generated by $\frac{\mathrm{rand}\bullet\mathrm{rand}}{\mathrm{rand}}$.

In Eq. (8), only one dimension which is randomly selected is updated at each iteration, which is similar to that in the traditional artificial bee colony. And Eq. (8) applies the greedy selection strategy, which aims at enhancing the global search ability. For employed bees, they search for each food source and decide if it is the

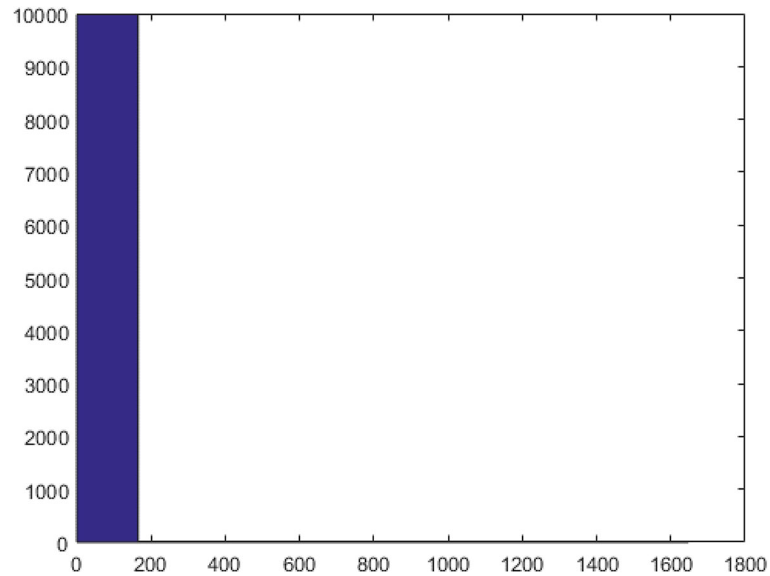

Fig. 1 Density distribution of generator $\frac{\text{rand}\bullet\text{rand}}{\text{rand}}$

global best food source. If it is not the best food source, they will randomly select a neighbor r of the current food source i and decide if pbest$_r$ is better than pbest$_i$. For Onlooker bees, they use the roulette wheel selection mechanism to select a food source to exploit. The main pseudo code of ABCIS algorithm is presented in Algorithm 1.

Experimental results and discussions

To investigate the proposed ABCIS algorithm's effectiveness, integral and comprehensive experiments are conducted in this subsection. In Section 2, 14 classical benchmark functions, functioning as the evaluation criterion, are listed and introduced. Section 2 conducts comparison experiments for ABCIS and other state-of-the-art algo-

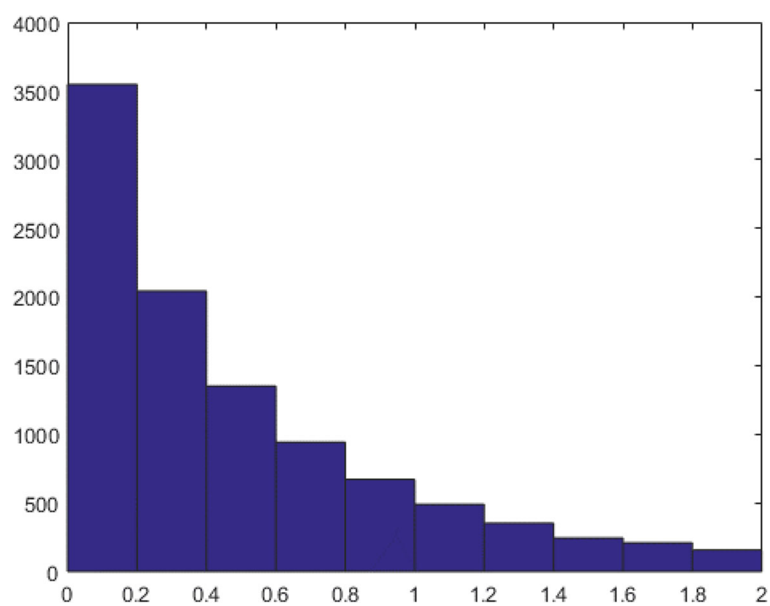

Fig. 2 Density distribution of generator $\frac{\text{rand}\bullet\text{rand}}{\text{rand}}$ within the range of [0, 2]

Initialization: SR=1
For each iteration:
Set succ=0;
Employed bees:
  For each food source:
    If this is not the global best food source:
      Set *Alpha* as the best food source;
      Randomly select a neighbor r of the current food source i ;
      If $pbest_r$ is better than $pbest_i$:
        $Beta = pbest_r$;
      Otherwise:
        $Beta = pbest_i$;
      Calculate c_1 and c_2 using Eqs. (6) and (7);
      For all the dimensions:
        Using Eq. (5) to generate new position;
      If the new generated position is better than the previous one:
        succ=succ+1;
    Else
      Select a neighbor m of the current food source;
      Select one dimension in the total dimension;
      Using Eq. (8) to generate a new trial vector;
      If the trial vector is better than the current one:
        succ=succ+1
        update the food position
Onlooker bees:
  For each onlooker bee:
    Using the roulette wheel selection mechanism to select a food source to exploit;
    If this is not the global best food source:
      Set *Alpha* as the best food source;
      Randomly select a neighbor r of the current food source i ;
      If $pbest_r$ is better than $pbest_i$:
        $Beta = pbest_r$;
      Otherwise:
        $Beta = pbest_i$;
      Calculate c_1 and c_2 using Eqs. (6) and (7);
      For all the dimensions:
        Using Eq. (5) to generate new position;
      If the new generated position is better than the previous one:
        succ=succ+1;
    Else
      Select a neighbor m of the current food source;
      Select one dimension in the total dimension;
      Using Eq. (8) to generate a new trial vector;
      If the trial vector is better than the current one:
        succ=succ+1
      update the food position
Calculate SR as $succ/NP$
If the termination condition is not meet, continue the loop.

Algorithm 1: The main pseudo code of ABCIS algorithm rithms to testify its performance. Section 2 carries out experiments to demonstrate each component's effectiveness of the proposed ABCIS.

Benchmarks' illustration

To testify the proposed algorithm's efficacy, this paper adopts fourteen classical benchmark functions which are widely used in literature [34–36]. Their detailed information, including mathematical expression, search range, and optimal value, is provided in Table 1. All of these benchmarks' optimal value is zero. Among them, the first nine benchmark functions are unimodal, which means there is only one local minimum point and it is also the global minimum point,and the last five functions are multimodal functions whose local optimal positions are numerous and the global optimal position is difficult to access. CEC2015 learning-based real-parameter single objective optimization problems are employed to further verify the proposed algorithm's performance.

Comparison experiments with other state-of-the-art algorithms

In this subsection, this paper conducts experiments for the purpose of demonstrating the proposed ABCIS's competitiveness among other evolutionary algorithms. In this comparison, the traditional particle swarm optimization, differential evolution algorithm, artificial bee colony and five ABC variants, including GABC [21], qABC [22], OCABC [23], ABCVSS [28], and ABCM [29], are adopted.

In this comparison, the population size is set to 40 and all the algorithms are repeated for 50 trails in order to be justice. Experiments are conducted on 10, 30, and 50 dimensions, with the corresponding maximum iteration setting as 1000, 3000, and 5000. For the maximum function evaluation number, it is set as the product of population size and maximum iteration number. When the maximum function evaluation number is achieved, the algorithm stops iteration, and after 50 trials, their final average fitness value (the first row) and standard deviation (the second row) results on these 14 benchmarks are recorded, as presented in Tables 2, 3, 4, and 5, corresponding to 10, 30, and 50 dimensions. Besides, the Wilcoxon rank-sum test [37–39] is also conducted to demonstrate the statistical effectiveness. In this measurement, the significant value is set to 5%. The result "=" means the proposed ABCIS algorithm attains results which are similar to the corresponding compared algorithm, "+" means the corresponding algorithm exhibits better performance than ABCIS and "-" means worse. This outcome is also provided in Tables 2, 3, 4, and 5. Further, Figs. 3, 4, and 5 also show the convergence curves of these nine evolutionary algorithm on some typical benchmarks.

From the results, it could be easily noticed that the proposed ABCIS algorithm exhibits best optimization performance almost on all the benchmarks and on all 10, 30, and 50 dimensions. It obtains the optimal value on several benchmarks, and achieves best accuracy on other problems. Most algorithms could find the global optima; thus, the convergence speed should be a criterion to evaluate an algorithm's performance. For the unimodal functions, the proposed ABCIS algorithm obtains the optimal value on several benchmarks and achieves best accuracy on other problems, which means that the two elite vectors' guiding could better perfect the convergence speed, which can also be verified in the convergence curves presented in Figs. 3, 4, and 5. For the multimodal functions, the proposed ABCIS algorithm also achieves great results, which may contribute to the update strategy of global best food source. Further demonstration will be discussed in the following subsection.

Each component of ABCIS's effectiveness

In this subsection, we aim at verifying each component's effectiveness of the proposed ABCIS. Two components are considered: the first is of which the total dimensional update strategy and the second is the special division strategy. For the experimental settings, the population size is chosen as 40 and the test dimension is 10. Each trial is repeated for 50 times and the average fitness value (the first row) and standard deviation (the second row) are recorded. The maximum iteration number is set as 1000 and the corresponding maximum number of function evaluation is calculated as the product of population size and maximum iteration number.

Total dimensional update strategy

In the proposed ABCIS algorithm, food sources except the best one are updated on all the dimensions at each generation. To testify this component's effectiveness, this subsection designs another ABC algorithm named as ABCIS-sd (ABCIS with single dimension) to perform this comparison. Experimental results are stated in Table 5, from which it could be observed that total dimensional update strategy could achieve much better solution accuracy on unimodal functions.

Special division strategy

In the previous subsection, we point out that the special division strategy does a great contribution to the multimodal functions' optimization, and this subsection aims at making demonstration. Thus, ABCIS-wd (ABCIS algorithm without division) is generated to function as the comparison algorithm. Experimental results are presented in Table 4.

Regardless of the proposed search Eq. (5) for bees, there are two main differences between the proposed ABCIS and the standard ABC: one is the total dimensional update strategy for bees except for the best one and another is

Table 1 Detailed information of fourteen benchmark functions

$f1$	$\sum_{i=1}^{n} {x_i}^2$	[− 100,100]	0
$f2$	$\sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	[− 100,100]	0
$f3$	$10^6 * {x_1}^2 + \sum_{i=2}^{n} {x_i}^2$	[− 100,100]	0
$f4$	$\sum_{i=1}^{n} i * {x_i}^2$	[− 100,100]	0
$f5$	$\sum_{i=1}^{n} \lfloor x_i + 0.5 \rfloor^2$	[− 100,100]	0
$f6$	$\sum_{i=1}^{n} i{x_i}^4 + random[0,1)$	[− 1.28,1.28]	0
$f7$	$\sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[− 10,10]	0
$f8$	$\sum_{i=1}^{n} {x_i}^2 + \prod_{i=1}^{n} {x_i}^2$	[− 100,100]	0
$f9$	$\max(\lvert x_1 \rvert, \lvert x_2 \rvert, ..., \lvert x_n \rvert)$	[− 100,100]	0
$f10$	$\sum_{i=1}^{n} \left[{x_i}^2 - 10\cos(2\pi x_i) + 10\right]$	[− 5.12,5.12]	0
$f11$	$\frac{1}{4000} \sum_{i=1}^{n} {x_i}^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[− 600,600]	0
$f12$	$-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} {x_i}^2}) + \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	[− 32,32]	0
$f13$	$\frac{\pi}{30}\{10 \cdot \sin[1 + 0.25(x_1+1)^2] + \sum_{i=1}^{n-1} \{[0.25\,(x_i+1)]^2\,[1 + 10(\sin(\pi(1 + 0.25\,(x_{i+1}+1)))^2)]\}\} + \sum_{i=1}^{n} y_i$ $y_i = \begin{cases} 100(x_i - 10)^4 & x_i > 10 \\ 0 & -10 \le x_i \le 10 \\ 100(-x_i - 10)^4 & x_i < -10 \end{cases}$	[− 50,50]	0
$f14$	$0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} y_i$ $y_i = \begin{cases} 100(x_i - 10)^4 & x_i > 5 \\ 0 & 5 \le x_i \le 5 \\ 100(-x_i - 10)^4 & x_i < -5 \end{cases}$	[− 50,50]	0

Table 2 Comparison results, $D = 10$

F	PSO	DE	ABC	GABC	ABCIS
$f1$	3.44E−25 8.09E−25 -	1.67E−49 2.46E−49 -	1.04E−16 3.47E−17 -	6.79E−17 1.50E−17 -	0 0
$f2$	5.30E−07 9.65E−07 -	1.07E−18 1.67E−18 -	7.27E+01 3.74E+01 -	9.17E+01 6.76E+01 -	1.65E−52 7.38E−52
$f3$	3.82E−25 7.36E−25 -	2.20E−48 6.22E−48 -	1.38E−16 5.51E−17 -	6.10E−17 1.48E−17 -	0 0
$f4$	6.34E−24 2.06E−23 -	7.35E−49 1.03E−48 -	1.35E−16 6.66E−17 -	7.71E−17 3.24E−17 -	0 0
$f5$	0 0 =	0 0 =	0 0 =	0 0 =	0 0
$f6$	2.87E−03 1.75E−03 -	2.03E−03 8.82E−04 -	2.12E−02 6.24E−03 -	6.44E−03 2.74E−03 -	2.27E−04 1.62E−04
$f7$	2.28E−15 2.51E−15 -	1.44E−26 1.73E−26 -	3.91E−16 8.79E−17 -	3.00E−16 4.50E−17 -	2.09E−243 0
$f8$	8.07E−23 3.29E−22 -	2.38E−46 4.71E−46 -	1.40E−16 5.75E−17 -	7.67E−17 1.62E−17 -	0 0
$f9$	4.98E−07 3.23E−07 -	3.87E−06 1.68E−05 -	1.19E−01 8.40E−02 -	3.03E−04 6.73E−04 -	6.45E−129 2.31E−128
$f10$	3.44E+00 1.78E+00 -	9.69E+00 4.28E+00 -	0 0 =	60 0 =	0 0
$f11$	8.27E−02 3.34E−02 -	7.37E−02 6.50E−02 -	1.13E−02 4.19E−03 +	9.18E−03 2.05E−03 +	2.73E−02 1.52E−02
$f12$	3.56E−01 1.28E−01 -	5.40E−01 8.48E−02 -	1.05E−14 4.70E−15 -	6.04E−15 7.94E−16 -	3.20E−15 1.30E−15
$f13$	2.33E−28 5.06E−28 -	1.57E−32 8.21E−48 -	8.58E−17 2.62E−17 -	5.16E−17 1.83E−17 -	1.57E−32 2.81E−48
$f14$	7.23E−24 1.45E−23 -	1.35E−32 2.74E−48 =	1.13E−16 4.85E−17 -	7.36E−17 2.62E−17 -	1.35E−32 2.74E−48
$f1$	3.51E−17 4.78E−17 -	3.13E−58 9.19E−58 -	1.14E−47 5.11E−47 -	8.75E−51 1.95E−50 -	0 0
$f2$	1.45E+02 7.71E+01 -	1.49E+01 1.64E+01 -	1.24E+02 1.17E+02 -	4.04E+01 2.53E+01 -	1.65E−52 7.38E−52
$f3$	1.42E−12 6.01E−12 -	2.54E−39 1.13E−38 -	3.25E−70 1.08E−69 -	3.18E−48 8.02E−48 -	0 0
$f4$	5.73E−17 6.20E−17 -	1.98E−51 8.85E−51 -	2.95E−68 9.11E−68 -	8.79E−48 3.92E−47 -	0 0

Table 2 Comparison results, $D = 10$ (*Continued*)

F	PSO	DE	ABC	GABC	ABCIS
$f5$	0	0	0	0	0
	0	0	0	0	0
	=	=	=	=	
$f6$	8.23E−03	2.71E−03	6.76E−03	1.75E−02	2.27E−04
	5.65E−03	1.20E−03	3.73E−03	6.66E−03	1.62E−04
	-	-	-	-	
$f7$	1.44E−16	6.50E−34	1.30E−38	7.76E−28	2.09E−243
	2.03E−16	1.50E−33	3.85E−38	1.07E−27	0
	-	-	-	-	
$f8$	2.29E−12	5.15E−50	4.63E−72	3.95E−48	0
	1.03E−11	2.30E−49	1.82E−71	9.71E−48	0
	-	-	-	-	
$f9$	3.63E−05	2.52E−07	1.16E−01	1.26E−01	6.45E−129
	6.45E−05	2.06E−07	1.64E−01	7.99E−02	2.31E−128
	-	-	-	-	
$f10$	0	0	0	0	0
	0	0	0	0	0
	=	=	=	=	
$f11$	8.25E−03	1.14E−02	8.69E−03	1.05E−02	2.73E−02
	1.20E−03	4.70E−03	1.39E−03	3.55E−03	1.52E−02
	+	+	+	+	
$f12$	2.49E−15	6.27E−03	3.73E−15	1.10E−14	3.20E−15
	2.44E−15	2.80E−02	1.67E−15	3.87E−15	1.30E−15
	+	-	-	-	
$f13$	1.25E−18	1.57E−32	3.37E−11	1.57E−32	1.57E−32
	1.65E−18	2.81E−48	1.51E−10	2.81E−48	2.81E−48
	-	=	-	=	
$f14$	8.47E−18	1.35E−32	1.74E−06	1.35E−32	1.35E−32
	9.90E−18	2.81E−48	7.76E−06	2.81E−48	2.74E−48
	-	-	-	-	

Table 3 Comparison results, $D = 30$

F	PSO	DE	ABC	GABC	ABCIS
$f1$	2.05E−17	1.88E−47	5.29E−16	3.83E−16	0
	5.06E−17	4.89E−47	7.13E−17	9.45E−17	0
	-	-	-	-	
$f2$	1.18E+02	1.98E−01	5.87E+03	7.06E+03	7.24E−03
	5.64E+01	1.48E−01	1.34E+03	1.82E+03	1.16E−02
	-	-	-	-	
$f3$	6.76E−17	1.02E−47	5.00E−16	3.69E−16	0
	1.67E−16	1.24E−47	8.35E−17	7.14E−17	0
	-	-	-	-	
$f4$	1.68E−16	1.28E−46	5.22E−16	3.98E−16	0
	2.08E−16	1.67E−46	9.79E−17	8.55E−17	0
	-	-	-	-	
$f5$	0	0	0	0	0
	0	0	0	0	0
	=	=	=	=	
$f6$	2.20E−02	5.55E−03	3.97E−02	1.99E−02	2.76E−04
	8.44E−03	1.68E−03	1.08E−02	5.98E−03	1.20E−04
	-	-	-	-	
$f7$	6.65E−13	1.96E−25	1.29E−15	1.21E−15	0
	5.57E−13	1.93E−25	1.28E−16	1.47E−16	0
	-	-	-	-	

Table 3 Comparison results, $D = 30$ (*Continued*)

F	PSO	DE	ABC	GABC	ABCIS
*f*8	1.11E−07 4.84E−07 -	3.32E−43 7.69E−43 -	5.49E−16 1.24E−16 -	3.99E−16 7.13E−17 -	0 0
*f*9	3.63E+00 1.07E+00 -	9.28E+00 6.66E+00 -	2.19E+00 6.04E−01 -	2.45E−01 5.08E−02 -	5.79E−02 7.29E−02
*f*10	2.84E+01 8.06E+00 -	1.05E+02 3.19E+01 -	0 0 =	0 0 =	0 0
*f*11	1 3.43E−16	1 2.22E−16	1 2.12E−16	1 1.25E−16	1 2.60E−16
*f*12	8.46E−01 1.36E−01 -	9.03E−01 1.72E−01 -	6.22E−14 1.66E−14 -	3.09E−14 3.73E−15 -	6.57E−15 2.55E−15
*f*13	3.63E−02 6.78E−02 -	1.58E−32 2.81E−34 -	5.08E−16 7.96E−17 -	3.61E−16 7.89E−17 -	1.57E−32 2.81E−48
*f*14	2.30E+02 1.76E+02 -	1.17E+02 1.11E+02 -	6.12E−16 9.83E−17 -	3.80E−16 8.65E−17 -	1.35E−32 2.81E−48
*f*1	2.45E−16 2.38E−16 -	4.11E−20 1.81E−19 -	1.87E−09 8.34E−09 -	1.61E−45 5.27E−45 -	0 0
*f*2	7.23E+03 1.58E+03 -	4.40E+03 1.46E+03 -	7.96E+03 1.50E+03 -	4.68E+03 1.14E+03 -	7.24E−03 1.16E−02
*f*3	1.47E−17 1.93E−17 -	2.85E−19 1.27E−18 -	4.94E−11 2.21E−10 -	4.00E−44 9.56E−44 -	0 0
*f*4	6.38E−16 2.09E−16 -	2.29E−24 1.00E−23 -	1.55E−66 6.89E−66 -	1.40E−43 5.59E−43 -	0 0
*f*5	0 0 =	0 0 =	0 0 =	0 0 =	0 0
*f*6	3.35E−02 1.49E−02 -	5.35E−03 1.64E−03 -	8.50E−02 2.68E−02 -	4.47E−02 1.28E−02 -	2.76E−04 1.20E−04
*f*7	9.15E−16 4.59E−16 -	2.66E−16 1.17E−15 -	1.33E−07 5.93E−07 -	1.29E−25 1.79E−25 -	0 0
*f*8	2.68E−16 2.63E−16 -	3.49E−22 1.55E−21 -	2.81E−12 1.26E−11 -	6.15E−44 1.76E−43 -	0 0
*f*9	2.97E−01 1.65E−01 -	8.38E+00 4.57E+00 -	1.08E+01 3.65E+00 -	2.49E+00 5.71E−01 -	5.79E−02 7.29E−02
*f*10	0 0 =	1.60E−01 3.63E−01 -	4.12E−01 1.84E+00 -	0 0 =	0 0
*f*11	1 5.11E−16	1 1.55E−16	1 1.23E−14	1 5.27E−16	1 2.60E−16

Table 3 Comparison results, $D = 30$ (*Continued*)

F	PSO	DE	ABC	GABC	ABCIS
$f12$	2.03E−14 8.27E−15 -	8.14E−02 4.00E−02 -	6.69E−03 2.99E−02 -	4.21E−14 5.27E−15 -	6.57E−15 2.55E−15
$f13$	1.28E−17 1.01E−17 -	1.88E−07 8.40E−07 -	1.66E−32 4.04E−33 -	1.57E−32 2.81E−48 =	1.57E−32 2.81E−48
$f14$	2.23E−17 1.77E−17 -	7.39E−21 2.53E−20 -	2.96E−04 1.33E−03 -	1.35E−32 2.81E−48 =	1.35E−32 2.81E−48

Table 4 Comparison results, $D = 50$

F	PSO	DE	ABC	GABC	ABCIS
$f1$	4.01E−13 4.99E−13 -	1.82E−48 3.67E−48 -	9.88E−16 1.32E−16 -	7.09E−16 9.76E−17 -	0 0
$f2$	5.69E+03 2.08E+03 -	3.86E+02 1.59E+02 -	2.16E+04 4.20E+03 -	2.40E+04 4.04E+03 -	1.46E+00 2.04E+00
$f3$	1.70E−12 5.45E−12 -	3.58E−48 5.09E−48 -	9.27E−16 1.03E−16 -	7.63E−16 9.59E−17 -	0 0
$f4$	9.38E−12 1.63E−11 -	2.22E−47 3.82E−47 -	1.03E−15 1.40E−16 -	7.21E−16 1.19E−16 -	0 0
$f5$	0 0 =	0 0 =	0 0 =	0 0 =	0 0
$f6$	7.25E−02 1.88E−02 -	1.16E−02 6.88E−03 -	7.82E−02 1.30E−02 -	3.81E−02 8.42E−03 -	3.85E−04 2.00E−04
$f7$	5.39E−10 8.23E−10 -	5.19E−27 7.09E−27 -	2.26E−15 2.19E−16 -	2.12E−15 2.77E−16 -	0 0
$f8$	45276.2962 65913.5423 -	5.69E−45 1.70E−44 -	1.04E−15 1.33E−16 -	7.62E−16 1.02E−16 -	0 0
$f9$	2.16E+01 4.03E+00 -	2.12E+01 6.63E+00 -	1.06E+01 2.09E+00 -	2.71E+00 2.88E−01 -	2.98E+00 4.80E+00
$f10$	7.15E+01 1.21E+01 -	1.39E+02 4.66E+01 -	0 0 =	0 0 =	0 0
$f11$	1 1.48E−14 -	1 1.99E−16 -	1 3.95E−16 -	1 2.70E−16 -	1 7.20E−17
$f12$	1.15E+00 1.51E−01 -	7.96E−01 2.56E−01 -	1.21E−13 2.76E−14 -	2.07E−13 6.57E−13 -	7.11E−15 3.02E−15
$f13$	4.37E−02 6.78E−02 -	3.11E−02 9.33E−02 -	9.42E−16 1.36E−16 -	6.61E−16 9.46E−17 -	1.57E−32 2.81E−48
$f14$	1.48E+03 3.55E+02 -	7.83E+02 3.19E+02 -	9.95E−16 1.16E−16 -	7.11E−16 9.30E−17 -	1.35E−32 2.81E−48

Table 4 Comparison results, $D = 50$ (*Continued*)

F	PSO	DE	ABC	GABC	ABCIS
$f1$	6.22E− 11 2.78E− 10 -	1.22E− 06 5.45E− 06 -	2.73E− 11 8.50E− 11 -	4.20E− 42 1.77E− 41 -	0 0
$f2$	2.22E+ 04 4.12E+ 03 -	2.16E+ 04 6.28E+ 03 -	2.64E+ 04 3.90E+ 03 -	1.79E+ 04 2.72E+ 03 -	1.46E+ 00 2.04E+ 00
$f3$	1.86E− 10 8.30E− 10 -	2.33E− 12 1.04E− 11 -	1.43E− 11 6.38E− 11 -	3.15E− 41 1.21E− 40 -	0 0
$f4$	1.19E− 15 4.25E− 16 -	1.51E− 09 6.45E− 09 -	1.04E− 09 4.67E− 09 -	5.10E− 40 2.05E− 39 -	0 0
$f5$	0 0 =	0 0 =	0 0 =	0 0 =	0 0
$f6$	7.21E− 02 2.76E− 02 -	6.36E− 03 1.73E− 03 -	2.29E− 01 3.99E− 02 -	7.56E− 02 1.33E− 02 -	3.85E− 04 2.00E− 04
$f7$	2.06E− 15 7.46E− 16 -	4.58E− 04 2.05E− 03 -	1.52E− 34 2.59E− 34 -	5.22E− 23 1.17E− 22 -	0 0
$f8$	5.38E− 16 3.46E− 16 -	0.09975354 0.36790203 -	8.72E− 65 2.18E− 64 -	3.86E− 41 1.19E− 40 -	0 0
$f9$	3.31E+ 00 1.25E+ 00 -	2.73E+ 01 5.43E+ 00 -	2.25E+ 01 6.35E+ 00 -	1.05E+ 01 2.14E+ 00 -	2.98E+ 00 4.80E+ 00
$f10$	0 0 =	1.09E+ 00 1.08E+ 00 -	8.88E− 17 3.97E− 16 =	0 0 -	0 0
$f11$	1 1.20E− 15 -	1.00000002 1.03E− 07 -	1 4.47E− 16 -	1 3.60E− 16 -	1 7.20E− 17
$f12$	4.37E− 14 1.65E− 14 -	6.81E− 02 4.00E− 02 -	7.83E− 14 1.18E− 14 -	8.08E− 14 9.51E− 15 -	7.11E− 15 3.02E− 15
$f13$	6.16E− 11 2.75E− 10 -	5.43E− 04 2.43E− 03 -	5.15E− 07 2.31E− 06 -	1.57E− 32 2.81E− 48 =	1.57E− 32 2.81E− 48
$f14$	5.49E− 17 7.37E− 17 -	6.95E− 08 2.85E− 07 -	1.32E− 03 5.90E− 03 -	1.35E− 32 2.81E− 48 =	1.35E− 32 2.81E− 48

the special division for the best one. Thus, this subsection demonstrates these two aspects' effectiveness first. With this purpose, two comparison algorithms, ABCIS-sd (ABCIS with single dimension) and ABCIS-wd (ABCIS without special division for the best bee) , are constructed. ABCIS-sd is a variant of ABCIS where bees update food sources only on one randomly selected dimension in both Eqs. (5) and (8), and ABCIS-wd is an ABCIS variant where all the bees using Eq. (5) as the update equation to find food sources.

Conclusion

In this paper, we studied the problem about the allocation of UAVs in UAV-aided wireless communications. We have used the artificial bee colony algorithm to search for the optimal UAV allocation scheme. To accelerate artificial bee colony's convergence speed and improve solution accuracy, we first propose an intellective search strategy for bees searching food sources and then introduce a special division for the global best food sources to compensate the intellective search strategy's draw-

Table 5 Effectiveness demonstration - total dimension update strategy

	$f1$	$f2$	$f3$	$f4$	$f5$	$f6$	$f7$	$f8$	$f9$	$f10$	$f11$	$f11$	$f13$	$f14$
ABCIS	0	1.65E−52	0	0	0	2.27E−04	2.09E−243	0	6.45E−129	0	2.73E−02	3.20E−15	1.57E−32	1.35E−32
	0	7.38E−52	0	0	0	1.62E−04	0	0	2.31E−128	0	1.52E−02	1.30E−15	2.81E−48	2.74E−48
ABCIS-sd	4.68E−60	2.49E−20	4.79E−60	2.07E−59	0	2.78E−04	2.49E−33	7.12E−60	1.58E−02	3.80E+00	1.37E−01	1.40E−01	1.32E−02	1.46E+00
	1.64E−59	4.31E−20	1.37E−59	6.32E−59	0	1.43E−04	3.35E−33	2.49E−59	1.02E−02	9.30E+00	9.37E−02	2.03E−01	1.56E−02	1.01E+00

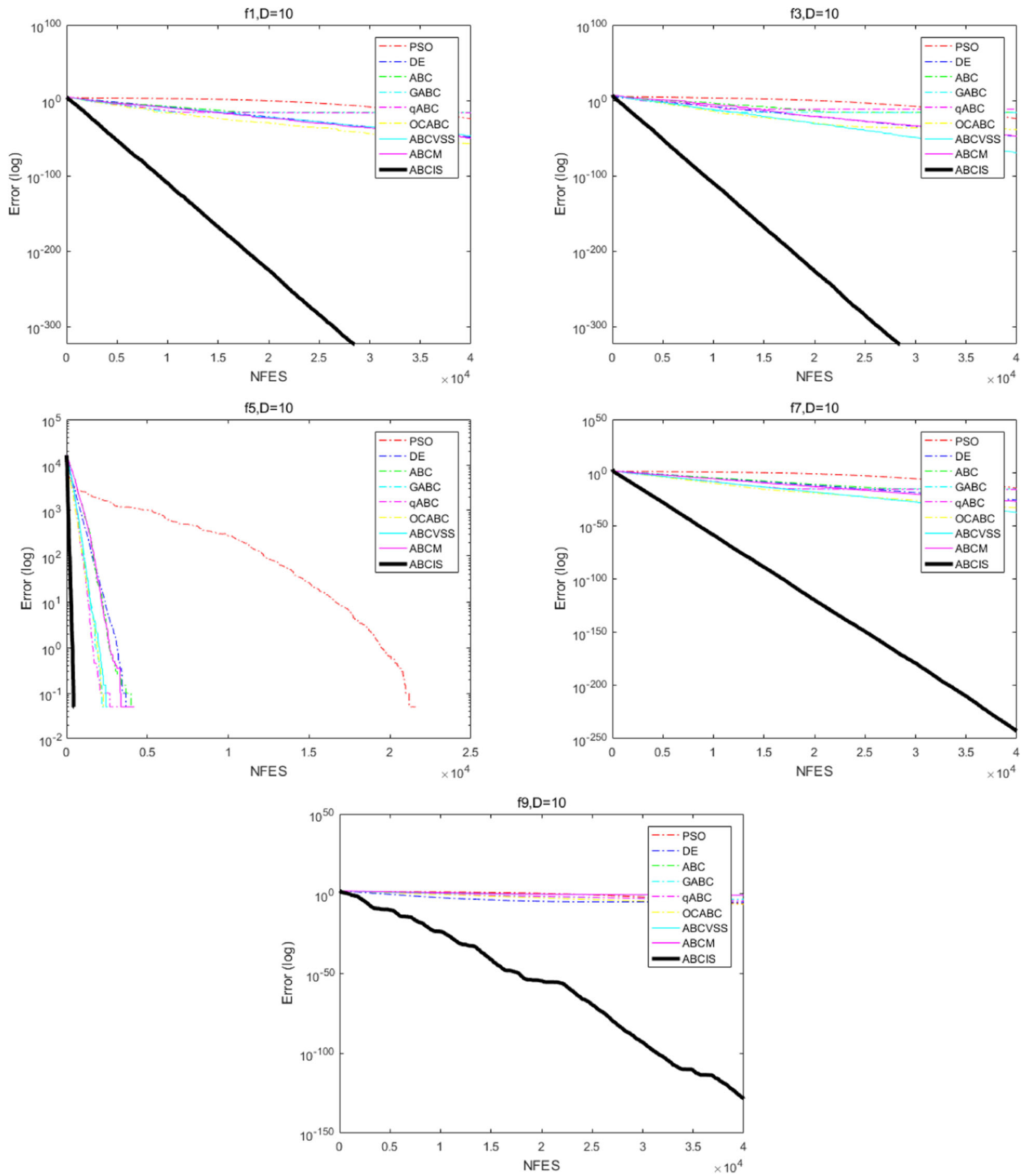

Fig. 3 Convergence curves, $D = 10$

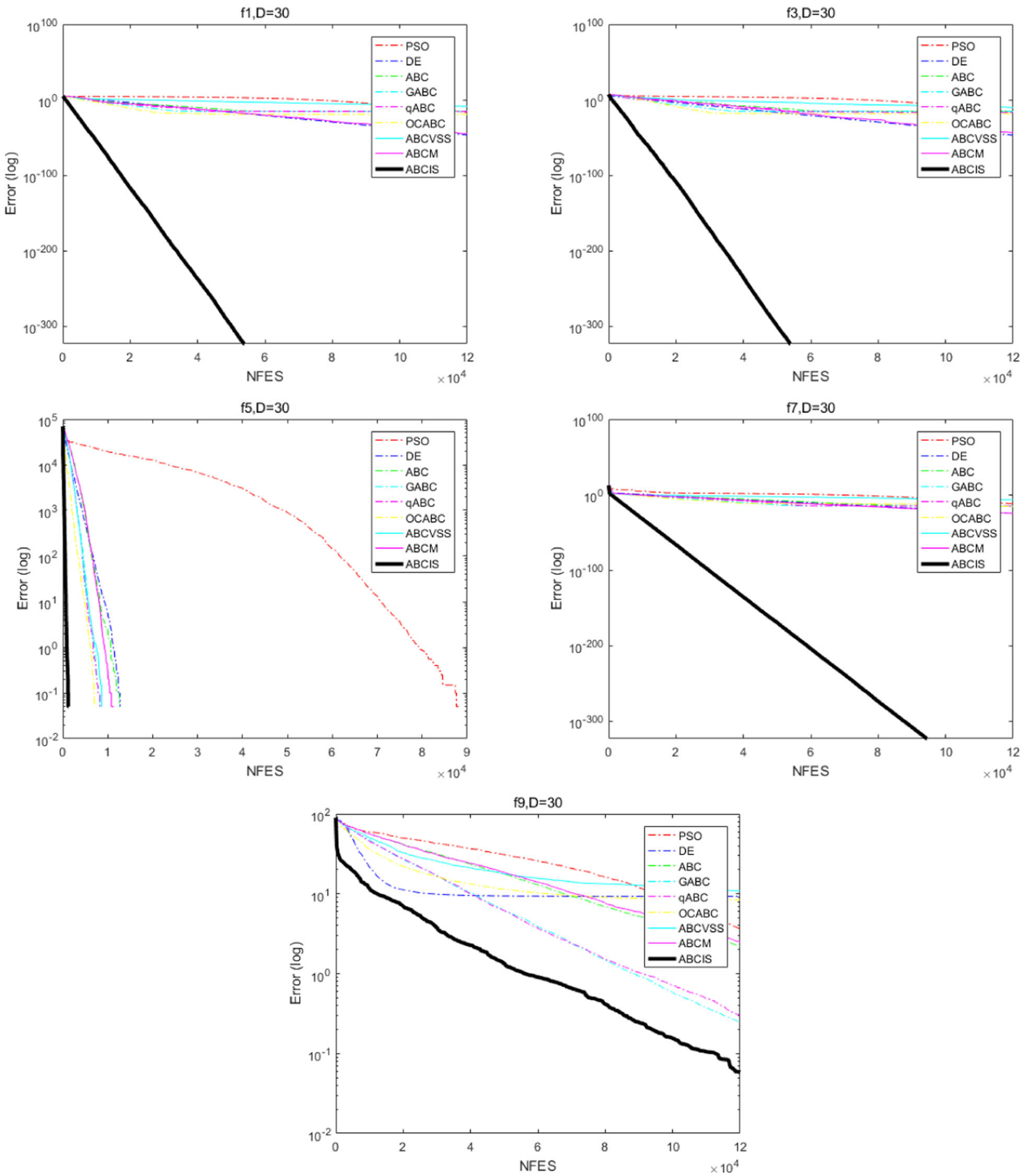

Fig. 4 Convergence curves, $D = 30$

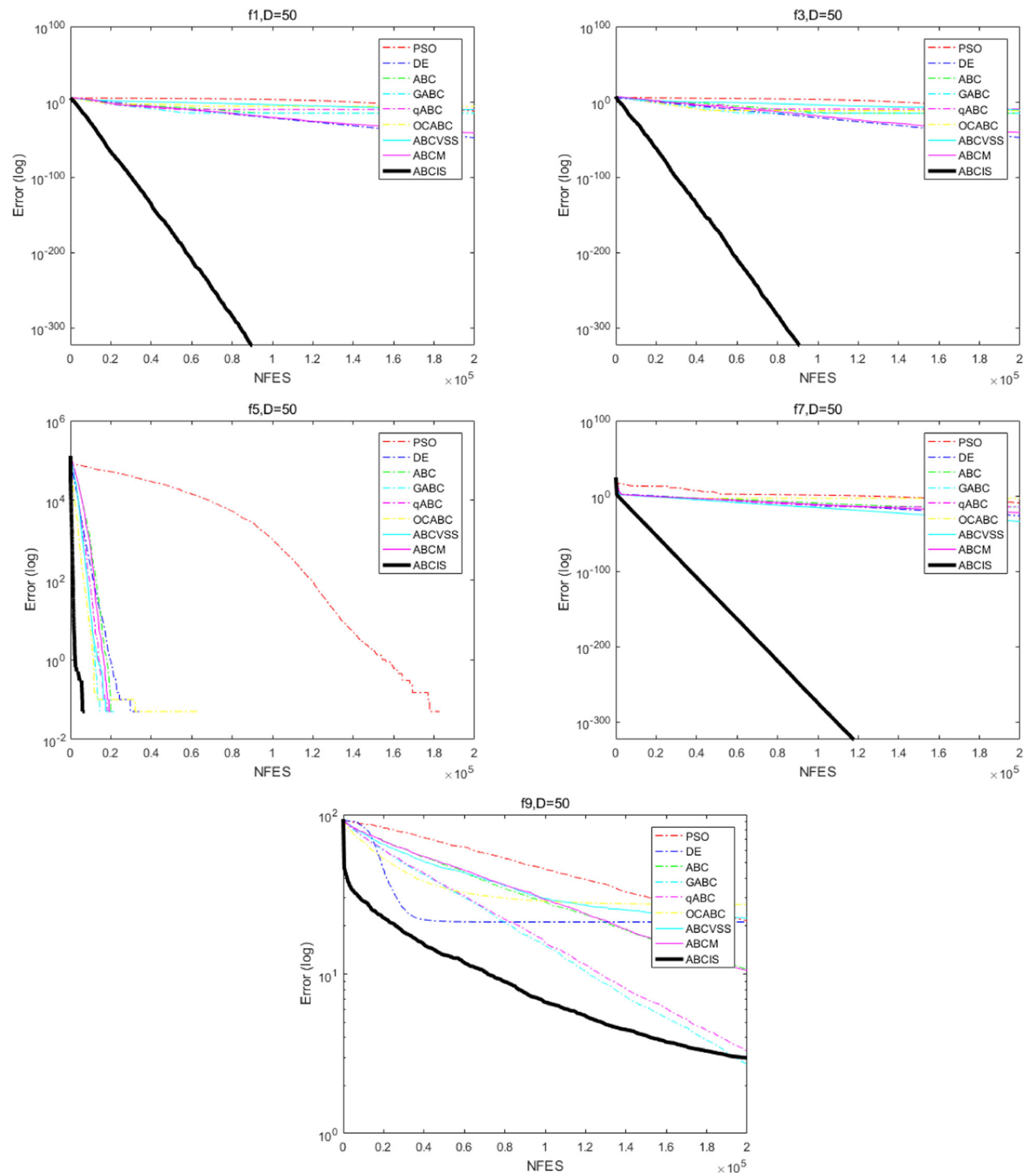

Fig. 5 Convergence curves, $D = 50$

backs. Experimental results demonstrate that this proposed ABCIS algorithm could achieve great improvements on both unimodal and multimodal functions, which improve its performance over UAV-aided wireless communications.

Abbreviations

A2G: Air-to-ground; ABC: Artificial bee colony; DE: Differential evolution; LoS: Line-of-sight; PSO: Particle swarm optimization; UAV: Unmanned aerial vehicle

Acknowledgments

Not applicable

Authors' contributions

BH is the main author of the current paper. BH contributed to the development of the ideas, design of the study, theory, result analysis, and article writing. ZS carried out the experimental work and the data collection and interpretation. HH finished the analysis and interpretation of data and drafted the manuscript. JL conceived and designed the experiments and undertook revision works of the paper. All authors read and approved the final manuscript .

Funding

This work was supported by the National Natural Science Foundation of China (grant nos. 61972208, 61802200, 61672299), the Natural Science Foundation of Jiangsu (grant nos. BK20180745, BK20190789), and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (grant nos. 18KJB520035, 19KJB130006). The authors would like to thank those who have provided helpful suggestions.

Availability of data and materials

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]College of Modern Posts, Nanjing University of Posts and Telecommunications, 210003, Nanjing, China. [2]College of Information Engineering, Nanjing University of Finances and Economics, 210003, Nanjing, China.

Received: 27 October 2019 Accepted: 26 January 2020
Published online: 11 February 2020

References

1. C. Li, H. J. Yang, F. Sun, J. M. Cioffi, L. Yang, Multiuser overhearing for cooperative two-way multiantenna relays. IEEE Trans. Veh. Technol. **65**(5), 3796–3802 (2016)
2. W. Wu, B. Wang, Efficient transmission solutions for MIMO wiretap channels with SWIPT. IEEE Commun. Lett. **19**(9), 1548–1551 (2015)
3. D. Karaboga, *An idea based on honey bee swarm for numerical optimization*, (2005). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department
4. H. Chen, et al., Artificial bee colony optimizer based on bee life-cycle for stationary and dynamic optimization. IEEE Trans. Syst. Man Cybern. Syst. **47**(2), 327–346 (2017)
5. T. H. S. Li, P. H. Kuo, Y. F. Ho, M. C. Kao, L. H. Tai, A biped gait learning algorithm for humanoid robots based on environmental impact assessed artificial bee colony. IEEE Access. **3**, 13–26 (2015)
6. J. q. Li, Q. k. Pan, P. y. Duan, An improved artificial bee colony algorithm for solving hybrid flexible flowshop with dynamic operation skipping. IEEE Trans. Cybern. **46**(6), 1311–1324 (2016)
7. C. Li, S. Zhang, P. Liu, F. Sun, J. M. Cioffi, L. Yang, Overhearing protocol design exploiting inter-cell interference in cooperative green networks. IEEE Trans. Veh. Technol. **65**(1), 441–446 (2016)
8. C. Li, P. Liu, C. Zou, F. Sun, J. M. Cioffi, L. Yang, Spectral-efficient cellular communications with coexistent one- and two-hop transmissions. IEEE Trans. Veh. Technol. **65**(8), 6765–6772 (2016)
9. W. Wu, F. Zhou, R. Q. Hu, B. Wang, Energy-Efficient Resource Allocation for Secure NOMA-Enabled Mobile Edge Computing Networks. IEEE Trans. Commun. **68**(1), 493-505 (2020). https://doi.org/10.1109/TCOMM.2019.2949994
10. C. Li, F. Sun, J. M. Cioffi, L. Yang, Energy efficient MIMO relay transmissions via joint power allocations. IEEE Trans. Circ. Syst. **61**(7), 531–535 (2014)
11. W. Wu, X. Yin, P. Deng, T. Guo, B. Wang, Transceiver Design for Downlink SWIPT NOMA Systems With Cooperative Full-Duplex Relaying. IEEE Access. **7**, 33464–33472 (2019)
12. C. Li, H. J. Yang, F. Sun, J. M. Cioffi, L. Yang, Adaptive overhearing in two-way multi-antenna relay channels. IEEE Signal. Proc. Lett. **23**(1), 117–120 (2016)
13. W. Wu, B. Wang, Robust secrecy beamforming for wireless information and power transfer in multiuser MISO communication system. EURASIP J. Wirel. Commun. Netw. **2015.1**, 161 (2015)
14. S. C. Horng, Combining artificial bee colony with ordinal optimization for stochastic economic lot scheduling problem. IEEE Trans. Syst. Man Cybern. Syst. **45**(3), 373–384 (2015)
15. L. Yang, X. Sun, L. Peng, X. Yao, T. Chi, An agent-Based artificial bee colony (ABC) algorithm for hyperspectral image endmember extraction in Parallel. IEEE J. Sel. Top Appl. Earth Obs. Remote Sens. **8**(10), 4657–4664 (2015)
16. A. Bose, K. Mali, Fuzzy-based artificial bee colony optimization for gray image segmentation. SIViP. **10**(6), 1089–1096 (2016)
17. A. K. Bhandari, A. Kumar, G. K. Singh, Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, Otsu and Tsallis functions. Expert Syst. Appl. **42**(3), 1573–1601 (2015)
18. B. Akay, D. Karaboga, A survey on the applications of artificial bee colony in signal, image, and video processing. SIViP. **9**(4), 967–990 (2015)
19. Y. Shi, R. Eberhart, in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*. A modified particle swarm optimizer, (1998), pp. 69–73. https://doi.org/10.1109/ICEC.1998.699146
20. R. Storn, K. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. J. Global. Optim. **11**(4), 341–359 (1997)
21. G. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization. Appl. Math. Comput. **217**(7), 3166–3173 (2010)
22. D. Karaboga, B. Gorkemli, A quick artificial bee colony (qABC) algorithm and its performance on optimization problems. Appl. Softw. Comput. **23**, 227–238 (2014)
23. 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. **43**(3), 1011–1024 (2013)
24. F. Kuang, Z. Jin, W. Xu, et al., in *IEEE Congress on Evolutionary Computation (CEC)*. A novel chaotic artificial bee colony algorithm based on tent map, (2014), pp. 235–241. https://doi.org/10.1109/cec.2014.6900278
25. W. F. Gao, L. L. Huang, S. Y. Liu, et al., Artificial bee colony algorithm based on information learning. IEEE Trans. Cybern. **45**(12), 2827–2839 (2015)
26. M. S. Kran, O. Fndk, A directed artificial bee colony algorithm. App. Soft. Comput. **26**, 454–462 (2015)
27. I. Babaoglu, Artificial bee colony algorithm with distribution-based update rule. App. Soft. Comput. **34**, 851–861 (2015)
28. M. S. Kiran, H. Hakli, M. Gunduz, et al., Artificial bee colony algorithm with variable search strategy for continuous optimization. Inform. Sci. **300**, 140–157 (2015)
29. X. Li, G. Yang, Artificial bee colony algorithm with memory. App. Soft. Comput. **41**, 362–372 (2016)
30. Y. Shi, C. M. Pun, H. Hu, et al., An improved artificial bee colony and its application. Knowl. Syst. **107**, 14–31 (2016)
31. K. Binder, D. Heermann, L. Roelofs, et al., Monte carlo simulation in statistical physics. Comput. Phys. **7**(2), 156–157 (1993)
32. R. H. Swendsen, J. S. Wang, Nonuniversal critical dynamics in monte carlo simulations. Phys. Rev. Lett. **58**(2), 86 (1987)
33. P. P. Boyle, Options: A monte carlo approach. J. Finan. Econ. **4**(3), 323–338 (1977)
34. X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster. IEEE Trans. Evol. Comput. **3**(9), 82–102 (1999)
35. M. M. Ali, C. Khompatraporn, Z. B. Zabinsky, A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. J. Global. Optim. **31**(4), 635–672 (2005)
36. H. Gao, S. Kwong, J. Yang, et al., Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation. Inform. Sci. **250**, 82–112 (2013). https://doi.org/10.1016/j.ins.2013.07.005
37. P. Wu, Y. Han, T. Chen, et al., Causal inference for Mann-Whitney-Wilcoxon rank sum and other nonparametric statistics. Stat. Med. **33**(8), 1261–1271 (2014)
38. L. De Capitani, D. De Martini, Reproducibility probability estimation and testing for the Wilcoxon rank-sum test. J. Stat. Comput. Simul. **85**(3), 468–493 (2015)
39. R. C. Blair, J. J. Higgins, A comparison of the power of Wilcoxon's rank-sum statistic to that of student's *t* statistic under various nonnormal distributions. J. Educ. Behav. Stat. **5**(4), 309–335 (1980)

Publisher's Note