

# An improved optimization method based on krill herd and artificial bee colony with information exchange

Heqi Wang<sup>1</sup> · Jiao-Hong Yi<sup>2</sup>

Received: 7 August 2016 / Accepted: 19 July 2017 / Published online: 31 July 2017  
© Springer-Verlag GmbH Germany 2017

**Abstract** This study presents a robust optimization algorithm based on hybridization of krill herd (KH) and artificial bee colony (ABC) methods and the information exchange concept. The global optimal solutions found by the proposed hybrid KH and ABC (KHABC) algorithm are considered as a neighbor food source for onlooker bees in ABC. Thereafter, a local search is performed by the onlooker bees in order to find a better solution around the given neighbor food source. Both the methods—the KH and ABC—share the globally best solutions through the information exchange process between the krill and bees. Based on the results, the exchange process significantly improves exploration and exploitation of the hybrid method. Besides, a focused elitism scheme is introduced to enhance the performance of the developed algorithm. The validity of the KHABC method is verified using thirteen unconstrained benchmark functions, twenty-one CEC 2017 constrained real-parameter optimization problems, and ten CEC 2011 real world problems. The proposed method clearly demonstrates its ability to be a competitive optimization tool towards solving benchmark functions and real world problems.

**Keywords** Global optimization · Krill herd · Artificial bee colony · Elitism scheme · Constrained optimization · Real world problems

✉ Jiao-Hong Yi  
yijiaohong@163.com

Heqi Wang  
whq200808@gmail.com

<sup>1</sup> Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China

<sup>2</sup> School of Environmental Science and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, Jiangsu, China

## 1 Introduction

Optimization is considered as the choice of a vector for an objective function in a given domain to make an optimal solution. In the last two decades, several metaheuristic techniques have been developed to solve difficult optimization problems. Some of these problems are navigation [1], and big data optimization [2]. The potential of metaheuristic optimization approaches for addressing various maximization/minimization problems, especially the NP-hard problems, is well documented. This is evident from the sizeable number of recently proposed modern stochastic optimization methods. Some of the major metaheuristic optimization methods that have been applied to solve challenging optimization problems are: differential evolution (DE) [3], evolutionary strategy (ES) [4], genetic algorithms (GAs) [5], artificial bee colony (ABC) [6], elephant herding optimization (EHO) [7], moth search (MS) algorithm [8], harmony search (HS) [9], monarch butterfly optimization (MBO) [10], and particle swarm optimization (PSO) [11]. In fact, individuals in swarm intelligence algorithms like a meme in memetic computing [12, 13].

Krill herd (KH) is a robust swarm intelligence algorithm originally proposed by Gandomi and Alavi [14]. KH has been proved to be superior to many other metaheuristic approaches (e.g., GAs, ES, BBO, DE, PSO) for solving many benchmark problems [14]. This clearly indicates that KH is a generic stochastic optimization method with immense scope of further development. On the other side, the ABC algorithm, motivated by the swarm behaviors of bee colonies, has a quite simple yet effective structure for solving optimization problems [6]. Hence, it has attracted the attention of many researchers.

It is known that the metaheuristic methods require various exploration and exploitation schemes for solving problems

with increasing dimensions in the search space. Although the KH generally explores the search space well and appears to be fully capable of locating the global optimal value, its exploration ability has exhibited relatively poor performance at later run phase [14]. On the other hand, the ABC method has strong exploration ability with its poor exploitation. Therefore, KH or ABC method, when acts independently, does not exhibit the potential needed for the exploration and exploitation of the search space [15]. In order to address this, the present study intends to investigate a hybridization of the ABC and KH methods towards solving continuous numerical global optimization as well as discrete problems.

The rest of the paper is organized as follows: the related work about KH and ABC is provided in Sect. 2. An overview of the basic KH and ABC is presented in Sect. 3. This is followed by the detailed hybridization process in Sect. 4. The performance evaluation is carried out in Sect. 5. The manuscript ends with the conclusions and guidance, as provided in Sect. 6.

## 2 Review of related literature

Though this paper is based on krill herd (KH) algorithm and artificial bee colony (ABC) algorithm, therefore, some of the most representative algorithms of KH and ABC will be reviewed.

### 2.1 KH algorithm

Wang et al. [16] introduced the chaos theory into the KH optimization process. The range of a chaotic map is always between 0 and 1 through normalization. Twelve chaotic maps are used to tune the inertia weights ( $\omega_n$ ,  $\omega_f$ ) in KH on fourteen benchmarks. The best chaotic map (Singer map) is selected to generate the chaotic KH (CKH) algorithm [16], and it is further compared with other state-of-the-art metaheuristic algorithms.

Wang et al. [17] proposed a hybrid metaheuristic algorithm namely CSKH by a combination of the advantages of cuckoo search (CS) and KH. In CSKH, two operators inspired by the CS algorithm, krill updating (KU) and krill abandoning (KA) were introduced into the basic KH. The KU operator inspires the intensive exploitation and makes the krill individuals search the space carefully in the later run phase of the search, while the KA operator is used to further enhance the exploration of the CSKH in place of a fraction of the worse krill at the end of each generation.

Mukherjee et al. [18] used various chaotic maps to generate chaotic KH (CKH) with the aim of improving the performance of the basic KH method. It is observed that Logistic map-based CKHA offers better results as compared other chaotic maps.

Wang et al. [19] added an updated version of reproduction schemes called stud selection and crossover (SSC) operator to the basic KH algorithm. Accordingly, a new version of the KH algorithm termed as Stud Krill Herd (SKH) was proposed. The added SSC operator is inspired by the Stud genetic algorithm. It selects the best krill (Stud) to perform the crossover operator. This approach appears to be well capable of solving various functions.

Limited by the length of the paper, we just review some of the most presentative KH papers. More related work of KH algorithm can be found in Bolaji et al. [20].

### 2.2 ABC algorithm

Since the basic ABC has been proposed, it has developed very fast. The related literature of ABC algorithm will be reviewed below.

Bolaji et al. [21] proposed a novel hybrid ABC algorithm based on the integrated technique for tackling the university course timetabling problem. First of all, initial feasible solutions are generated using the combination of saturation degree and backtracking algorithm. Secondly, a hill climbing optimizer is embedded within the employed bee operator to enhance the local exploitation ability of the ABC while tackling the problem. Empirical results on these problem instances validate the effectiveness and efficiency of the proposed algorithm for addressing the university course timetabling problem.

Kıran and Gündüz [22] proposed a hybridization of PSO and ABC approaches, namely HPA. The global best solutions obtained by PSO and ABC are used for recombination, and the solution obtained from this recombination is given to the populations of the PSO and ABC as the global best and neighbor food source for onlooker bees, respectively. They utilized twelve basic benchmark functions in addition to CEC 2005 composite functions and an energy demand estimation problem to verify their proposed HPA algorithm.

Awadallah et al. [23] proposed a metaheuristic technique called a hybrid artificial bee colony (HABC) for the nurse rostering problem (NRP). In HABC, the process of the employed bee operator is replaced with the hill climbing optimizer (HCO) to empower its exploitation capability and the usage of HCO is controlled by hill climbing rate (HCR) parameter. The performance of the proposed HABC is evaluated using the standard dataset published in the first international nurse rostering competition 2010 (INRC 2010).

Bullinaria and AlYahya [24] examined the performance of ABC for optimizing the connection weights of feed-forward neural networks for classification tasks, and presented a more rigorous comparison with the traditional Back-Propagation (BP) training algorithm. The empirical results for benchmark problems demonstrate that using the standard “stopping early” approach with optimized learning parameters leads

to improved BP performance over the previous comparative study, and that a simple variation of the ABC approach provides improved ABC performance, too.

Li et al. [25] proposed an improved discrete ABC (DABC) algorithm to solve the hybrid flexible flowshop scheduling problem with dynamic operation skipping features in molten iron systems. First, each solution is represented by a two-vector-based solution representation, and a dynamic encoding mechanism is developed. Second, a flexible decoding strategy is designed. Next, a right-shift strategy considering the problem characteristics is developed, which can clearly improve the solution quality. Finally, an enhanced local search is embedded in the proposed algorithm to further improve the exploitation ability.

Another kind of bee algorithms, called bee colony optimization (BCO) algorithm, is also proposed. Krüger et al. [26] provided theoretical verification of the BCO algorithm by proving some convergence properties. As a result, the gap between successful practice and missing theory is reduced.

Limited by the length of the paper, we just review some of the most presentative ABC papers. More related work of ABC algorithm can be found in Hussein et al. [27].

### 3 Background

In this section, the background of our work, including ABC and KH, will be provided.

#### 3.1 The ABC method

ABC is one of the seminal metaheuristic methods among various intelligent optimization techniques. After the appearance of swarm intelligence of bee colony, the forage selection is modeled. Based on this model, the definition of three main can be defined as follows [28]:

- Food resource

In the simplest form, the value of a food source is described with only one quantity. In Fig. 1, A and B, C and D represent two food resources and two non-food resources, respectively. Furthermore, S, O, R, UF, and EF denote scouts, onlookers, recruits, unemployed foragers, and denote employed foragers, respectively.

- Unemployed foragers

The unemployed forages have two sorts. One is Scouts (S). A scout bee is the type of bee that begins implementing search autonomously without any a-priori knowledge. The other one

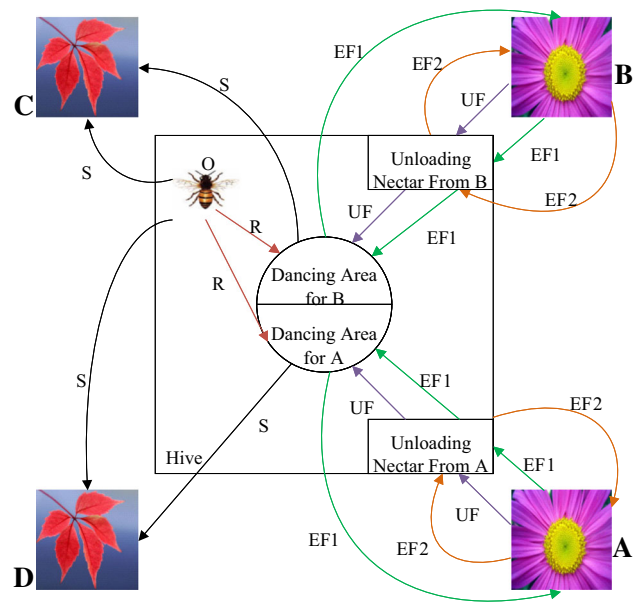


Fig. 1 The behavior of honey bee foraging for nectar

is Onlookers (O). They only stay in the nest in order to search for a food source with the help of the employed foragers.

- Employed foragers

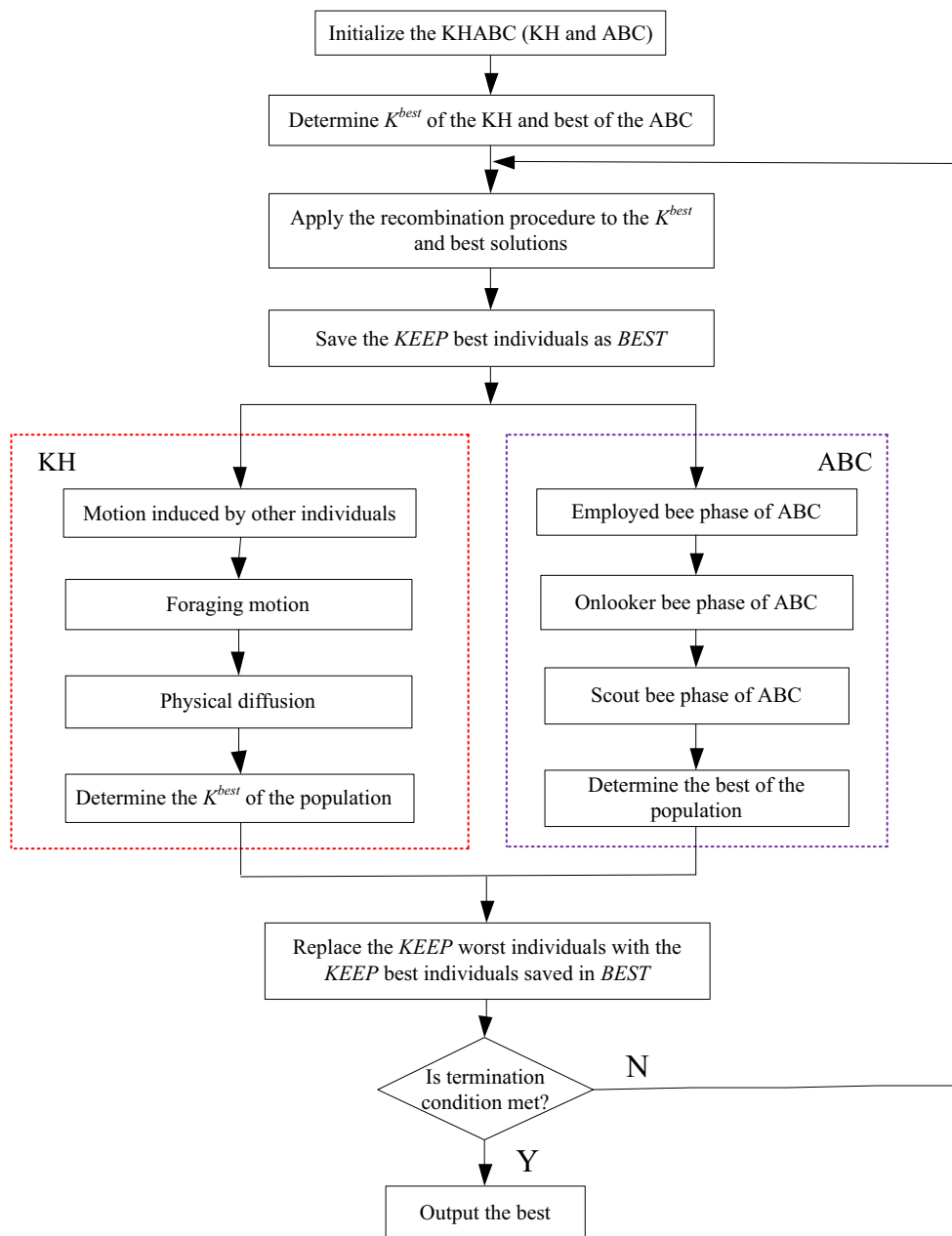
All of them are related to a food source that is being exploited. This information is shared with some probability. Three feasible choices associated with the quantity of nectar are provided for the foraging bee. One is Unemployed Forager (UF). When the nectar is less than a fixed threshold, the foraging bee gives it up and turns to an unemployed bee. The other one is Employed Forager 1 (EF1). If not, it may dance and recruit mates. The last one is Employed Forager 2 (EF2). It may forage around the food source all the time.

The artificial bee colony includes three types of bees: (1) employed bees, (2) onlookers and (3) scouts. In the artificial bee colony, a food source corresponds to an employed bee. That is to say, the employed bees and the food sources have the same number. The main steps of the search conducted by the artificial bees can be described as follows:

- Step 1 Initialize the population  $x_{ij}$ ;
- Step 2 Repeat
- Step 3 Generate new solutions  $v_{ij}$  around  $x_{ij}$  for the employed bees as

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj}) \tag{1}$$

Here  $k$  is a solution around  $i$ ,  $\Phi$  is a random number  $[-1, 1]$ .



**Fig. 2** Flowchart of the KHABC algorithm

Step 4 The greedy selection is used between  $x_i$  and  $v_i$ ;  
 Step 5 Calculate the probability  $P_i$  for  $x_i$  according to their fitness:

$$P_i = \frac{f_i}{\sum_{i=1}^{SN} f_i} \quad (2)$$

$SN$  is the number of food sources, and  $f$  is its fitness;

Step 6 Normalize  $P_i$  into  $[0, 1]$ ;  
 Step 7 Generate the new solutions  $v_i$  for the onlookers from  $x_i$ , selected depending on  $P_i$ ;

Step 8 The greedy selection is used for the onlookers between  $x_i$  and  $v_i$ ;

Step 9 Check if a solution is abandoned. If it is, replace it with a novel one  $x_i$  for the scout

$$x_{ij} = \min_j + \varphi_{ij} * (\max_j - \min_j) \quad (3)$$

Here  $\varphi_{ij}$  is a random number in  $[0, 1]$ .

Step 8 Save the best solution obtained up to now;

Step 9 Go to Step 2 until termination criteria is satisfied.

### 3.2 The KH method

KH [14] is a classic metaheuristic method for function optimization. KH is based on the simulation of the herding behavior of the krill individuals. The KH algorithm repeats the implementation of the three movements and takes search directions that converge to the best solution. The position is mostly influenced by three movements:

(i) Movement induced by other krill;

- (ii) Foraging action; and
- (iii) Random diffusion.

In KH, the Lagrangian model is used as shown below:

$$\frac{dX_i}{dt} = N_i + F_i + D_i \tag{4}$$

where  $N_i$  is the motion induced by other krill;  $F_i$  is the foraging motion, and  $D_i$  is the physical diffusion.

For the first motion, its direction ( $\alpha_i$ ) is estimated by the following three factors: target effect, local effect, and a repulsive effect. For a krill, its definition can be provided as:

$$N_i^{new} = N^{max}\alpha_i + \omega_n N_i^{old} \tag{5}$$

where

$$\alpha_i = \alpha_i^{local} + \alpha_i^{target} \tag{6}$$

and  $N^{max}$  is the maximum induced speed,  $\omega_n$  is the inertia weight in  $[0,1]$ ,  $N_i^{old}$  is the former motion,  $\alpha_i^{local}$  is the local effect and  $\alpha_i^{target}$  is the target direction effect.

The formulation of the second motion is mostly determined by two main components: the food location and the previous experience. It can be expressed for the  $i$ -th krill as follows:

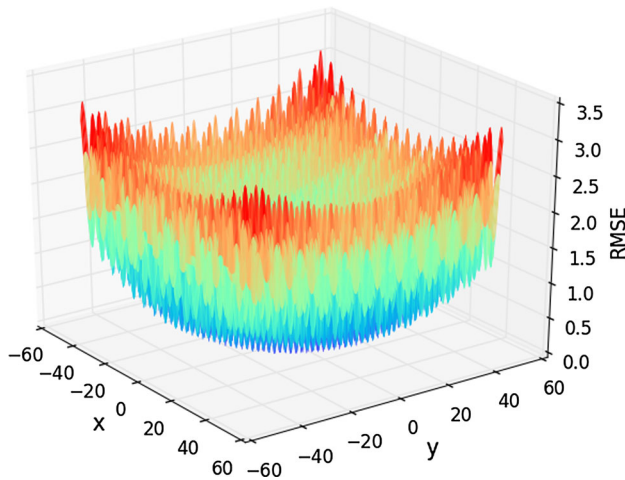


Fig. 3 3-D contours of the Griewank function

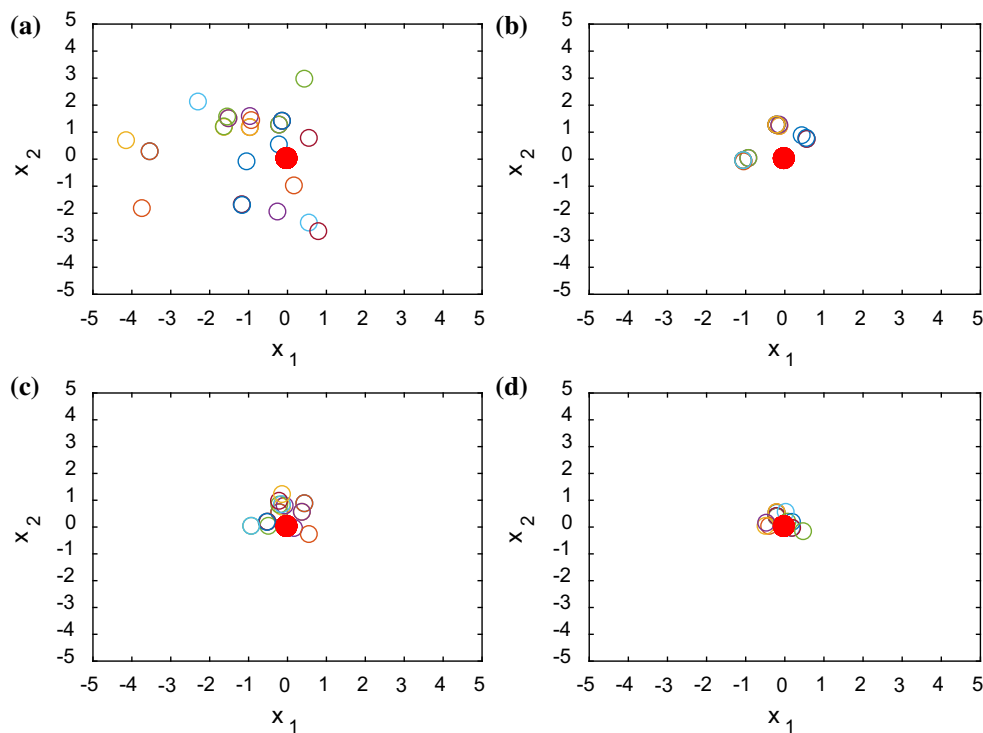


Fig. 4 The positions of the individuals and best position after a 1st iteration, b 2th iteration, c 3rd iteration, and d 4th iteration

**Table 1** Unconstrained benchmark functions

No.	Function	lb	ub	opt	Separability	Modality
F01	Griewank	−600	600	0	Nonseparable	Multimodal
F02	Holzman 2	−10	10	0	Separable	Multimodal
F03	Levy	−10	10	0	Nonseparable	Multimodal
F04	Penalty #1	−50	50	0	Nonseparable	Multimodal
F05	Penalty #2	−50	50	0	Nonseparable	Multimodal
F06	Perm 1	− <i>D</i>	<i>D</i>	0	Separable	Multimodal
F07	Perm 2	− <i>D</i>	<i>D</i>	0	Separable	Multimodal
F08	Powell	−4	5	0	Separable	Unimodal
F09	Rosenbrock	−2.048	2.048	0	Nonseparable	Unimodal
F10	Schwefel 1.2	−100	100	0	Nonseparable	Unimodal
F11	Schwefel 2.21	−100	100	0	Nonseparable	Unimodal
F12	Sphere	−5.12	5.12	0	Nonseparable	Unimodal
F13	Sum	−10	10	0	Separable	Unimodal

**Table 2** Best function values of thirteen unconstrained benchmark functions

	CCS	DE	ES	GA	HS	HSBBO	KHE	KHABC	MBA	PSO	VNBA
F01	20.00	52.77	227.30	41.99	319.70	<b>1.00</b>	11.03	1.18	59.23	142.10	1.09
F02	5.74E3	1.33E4	2.32E5	2.25E4	1.23E5	<b>2.20E−16</b>	2.01E3	0.29	85.26	1.48E4	1.94E−4
F03	19.59	29.05	127.00	55.60	96.94	<b>2.20E−16</b>	10.20	0.96	2.71	49.63	3.49E−3
F04	20.31	3.36E6	6.00E7	2.90E6	1.33E8	<b>2.20E−16</b>	6.10E4	2.01	1.21E4	4.13E6	0.87
F05	3.33E4	8.16E6	1.58E8	1.43E7	2.27E8	<b>2.20E−16</b>	8.46E5	14.15	1.50E5	1.66E7	1.07
F06	4.31E79	8.87E77	7.12E80	2.78E87	6.06E78	1.86E80	7.18E72	<b>6.27E57</b>	1.55E77	2.89E81	9.28E81
F07	1.72E−9	3.16E−8	1.98E−5	121.00	8.78E−4	1.37E−3	4.05E−5	<b>2.20E−16</b>	4.78E−8	3.25E−4	7.94E−3
F08	799.80	1.68E3	7.20E3	557.00	4.10E3	<b>2.20E−16</b>	783.40	5.33	9.37	1.45E3	0.02
F09	1.17E3	818.70	7.84E3	820.00	3.37E3	<b>2.20E−16</b>	273.50	30.45	60.80	1.03E3	7.34
F10	2.66E3	4.11E4	3.92E4	2.55E4	3.77E4	<b>2.20E−16</b>	2.07E4	258.90	7.92E3	2.41E4	1.44E4
F11	17.68	60.25	66.72	53.00	66.37	<b>2.20E−16</b>	8.61	<b>2.20E−16</b>	24.11	48.18	17.44
F12	25.58	20.36	126.40	72.00	100.00	<b>2.20E−16</b>	3.66	0.06	0.53	25.30	3.16E−4
F13	868.70	1.05E3	6.19E3	3.64E3	4.56E3	<b>2.20E−16</b>	307.30	5.25	171.80	1.57E3	0.13
Total	0	0	0	0	0	11	0	3	0	0	0

$$F_i = V_f \beta_i + \omega_f F_i^{old} \tag{7}$$

where

$$\beta_i = \beta_i^{food} + \beta_i^{best} \tag{8}$$

and  $V_f$  is the foraging speed,  $\omega_f$  is the inertia weight in [0, 1],  $F_i^{old}$  is the last motion,  $\beta_i^{food}$  is the food attractive and  $\beta_i^{best}$  is the effect of the  $i$ -th krill.

In essence, the third motion is looked upon as a random process. It can be expressed as:

$$D_i = D^{max} \delta \tag{9}$$

where  $D^{max}$  is the maximum diffusion speed, and  $\delta$  is the random directional vector.

According to the formulations of these actions for the  $i$ -th krill, the change in position of a krill from  $t$  to  $t + \Delta t$  can be represented by Eq. (10):

$$X_i(t + \Delta t) = X_i(t) + \Delta t \frac{dX_i}{dt} \tag{10}$$

### 4 The KHABC method

In this section, firstly, we will provide the main idea of KHABC algorithm, and then an example will be used to show how KHABC can works.

#### 4.1 The mainframe of KHABC method

Based on the above-analyses of the ABC, it can be seen that the standard ABC algorithm does not directly utilize



**Table 3** Mean function values of thirteen unconstrained benchmark functions

	CCS	DE	ES	GA	HS	HSBBO	KHE	KHABC	MBA	PSO	VNBA
F01	38.11	95.49	320.70	122.80	409.50	11.39	18.14	<b>1.98</b>	134.80	183.70	17.38
F02	1.77E4	3.02E4	3.92E5	1.14E5	2.57E5	389.20	5.75E3	<b>6.43</b>	4.48E3	6.09E4	3.63E3
F03	48.23	57.57	208.80	96.91	145.50	7.49	22.43	<b>1.96</b>	13.30	80.95	6.53
F04	2.06E4	1.78E7	2.94E8	2.88E7	2.42E8	1.87E5	4.37E5	<b>4.47</b>	7.62E6	2.95E7	4.46E5
F05	9.26E5	5.59E7	6.38E8	1.14E8	5.32E8	8.16E5	2.61E6	<b>33.59</b>	3.53E7	8.65E7	3.41E6
F06	1.48E83	7.01E82	7.33E85	2.78E87	1.25E84	4.45E83	2.79E87	3.23E83	<b>5.29E81</b>	1.21E85	5.53E87
F07	0.01	<b>2.71E−3</b>	11.26	121.00	2.71	0.51	4.07E20	12.10	3.82E−3	3.68	1.09E21
F08	1.91E3	3.60E3	1.28E4	1.78E3	7.37E3	581.30	1.54E3	<b>15.53</b>	81.01	3.19E3	1.09E3
F09	2.81E3	1.38E3	1.25E4	2.55E3	7.89E3	1.25E3	591.90	<b>42.02</b>	126.90	2.59E3	190.70
F10	5.52E3	5.91E4	7.01E4	4.78E4	6.07E4	1.85E4	3.38E4	<b>1.15E3</b>	3.32E4	4.56E4	5.55E4
F11	25.86	74.55	77.63	73.58	75.03	36.32	14.33	<b>8.10</b>	43.95	79.51	52.19
F12	53.12	30.07	183.90	115.50	137.90	4.30	8.09	<b>0.44</b>	4.10	54.76	3.07
F13	1.62E3	1.36E3	8.82E3	5.78E3	6.65E3	136.20	498.00	<b>21.07</b>	649.00	2.65E3	378.10
Total	0	1	0	0	0	0	0	11	1	0	0

**Table 4** Worst function values of thirteen unconstrained benchmark functions

	CCS	DE	ES	GA	HS	HSBBO	KHE	KHABC	MBA	PSO	VNBA
F01	59.13	121.50	400.30	231.30	490.10	66.65	28.82	<b>3.24</b>	303.20	520.00	95.07
F02	4.13E4	4.71E4	5.72E5	2.94E5	3.76E5	5.28E3	1.06E4	<b>53.54</b>	3.15E4	6.72E5	3.19E4
F03	72.47	81.73	263.70	155.20	191.70	29.23	39.15	<b>5.03</b>	26.72	127.40	44.07
F04	2.58E5	4.54E7	4.94E8	7.30E7	4.21E8	8.20E6	1.29E6	<b>10.43</b>	6.95E7	7.35E7	1.06E7
F05	4.98E6	1.04E8	9.36E8	3.30E8	8.53E8	1.38E7	5.88E6	<b>52.25</b>	2.04E8	2.16E8	7.67E7
F06	2.83E84	9.07E83	4.19E86	2.78E87	1.92E85	4.37E84	4.77E88	1.51E85	<b>3.43E82</b>	8.14E85	8.42E88
F07	0.08	0.06	99.92	121.00	52.12	3.56	1.09E22	121.00	<b>0.03</b>	47.67	5.34E22
F08	3.32E3	5.23E3	1.89E4	3.16E3	1.17E4	2.89E3	3.23E3	<b>69.15</b>	432.60	5.03E3	9.52E3
F09	6.19E3	2.14E3	1.57E4	5.84E3	1.06E4	1.07E4	876.90	<b>69.11</b>	224.60	4.33E3	674.10
F10	1.00E4	7.65E4	9.33E4	6.96E4	7.80E4	4.46E4	5.45E4	<b>4.51E3</b>	7.24E4	7.08E4	1.02E5
F11	34.01	84.88	83.41	87.00	81.00	89.00	22.97	<b>15.25</b>	71.17	93.00	85.09
F12	79.11	40.39	211.20	165.00	159.00	21.88	12.87	<b>1.36</b>	14.63	72.38	13.69
F13	2.50E3	1.75E3	1.04E4	8.11E3	8.15E3	816.00	769.20	<b>85.97</b>	2.55E3	1.00E4	1.82E3
Total	0	0	0	0	0	0	0	11	2	0	0

the global optimal individual. In addition, in KH, if any krill gets trapped in the local values, it cannot escape from local minimum by itself. To overcome these limitations, a hybrid meta-heuristic method based on information exchange is presented. The hybridization process is similar to that proposed by Kiran and Gündüz [22].

Information exchange or crossover operation is one of the most-famous evolution operators. Here, it is used for yielding a new solution, called *TheBest*. *TheBest* is considered to be  $K^{best}$  for the KH and food source of onlooker bees for the ABC. To get *TheBest*, the  $K^{best}$  of the KH and the optimal individual of the ABC are computed by Eq. (2). Probabilities used to select the two solutions are given by Eqs. (11) and (12):

$$P_{best} = \frac{fit_{best}}{fit_{K^{best}} + fit_{best}} \tag{11}$$

where  $P_{best}$  is the probability to choose the optimal individual of the ABC,  $fit_{best}$  and  $fit_{K^{best}}$  are the  $K^{best}$  of the KH and the optimal individual of the ABC achieved according to Eq. (2).

$$P_{K^{best}} = \frac{fit_{K^{best}}}{fit_{K^{best}} + fit_{best}} \tag{12}$$

where  $P_{K^{best}}$  is the  $K^{best}$  of the KH.

When generating the best solution, random numbers in the range of [0, 1] are utilized for the dimensions of the standard test function. If it is not above  $P_{best}$ , the value for this

**Table 5** The Std of thirteen unconstrained benchmark functions

	CCS	DE	ES	GA	HS	HSBBO	KHE	KHABC	MBA	PSO	VNBA
F01	10.05	14.11	34.58	46.25	35.04	17.24	4.34	<b>0.50</b>	44.71	51.89	19.12
F02	8.29E3	8.55E3	8.51E4	5.11E4	5.18E4	1.08E3	1.68E3	<b>9.00</b>	6.37E3	1.07E5	6.22E3
F03	13.54	10.69	24.82	24.81	21.28	7.45	6.96	<b>0.90</b>	5.73	14.99	8.10
F04	4.64E4	8.83E6	8.22E7	1.98E7	6.48E7	1.18E6	3.20E5	<b>1.77</b>	1.33E7	1.64E7	1.59E6
F05	9.24E5	2.25E7	1.57E8	7.67E7	1.38E8	2.63E6	1.27E6	<b>7.56</b>	4.48E7	4.65E7	1.17E7
F06	4.50E83	1.63E83	1.16E86	<b>8.94E71</b>	3.19E84	9.30E83	7.99E87	2.10E84	8.05E81	2.03E85	1.65E88
F07	0.02	8.80E−3	18.49	<b>1.22E−13</b>	8.60	0.87	1.98E21	36.67	6.73E−3	9.77	7.69E21
F08	634.50	811.30	2.79E3	660.00	1.67E3	691.10	614.90	<b>10.97</b>	72.86	939.60	1.82E3
F09	983.80	293.80	1.84E3	1.22E3	1.43E3	3.25E3	119.30	<b>8.26</b>	44.67	773.10	151.70
F10	1.78E3	8.95E3	1.28E4	1.16E4	1.04E4	1.32E4	7.61E3	<b>1.03E3</b>	1.66E4	1.16E4	2.01E4
F11	3.86	4.56	3.91	7.47	3.86	25.02	<b>2.94</b>	3.03	9.51	11.07	16.14
F12	12.04	4.94	17.67	22.02	13.00	6.65	2.00	<b>0.26</b>	2.95	8.03	3.72
F13	357.90	182.80	1.03E3	1.05E3	894.20	197.30	104.40	<b>13.09</b>	478.00	1.15E3	411.50
Total	0	0	0	2	0	0	1	10	0	0	0

**Table 6** Comparisons between KHABC and other methods at  $\alpha = 0.05$  on a two-tailed  $t$ -tests for thirteen unconstrained benchmark functions

	CCS	DE	ES	GA	HS	HSBBO	KHE	MBA	PSO	VNBA
F01	<b>25.39</b>	<b>46.83</b>	<b>65.16</b>	<b>18.46</b>	<b>82.23</b>	<b>3.86</b>	<b>26.18</b>	<b>21.00</b>	<b>24.77</b>	<b>5.69</b>
F02	<b>15.08</b>	<b>24.98</b>	<b>33.04</b>	<b>15.24</b>	<b>34.71</b>	<b>2.50</b>	<b>24.20</b>	<b>4.97</b>	<b>3.94</b>	<b>4.12</b>
F03	<b>24.11</b>	<b>36.64</b>	<b>58.89</b>	<b>27.04</b>	<b>47.63</b>	<b>5.20</b>	<b>20.62</b>	<b>13.83</b>	<b>37.19</b>	<b>3.96</b>
F04	<b>3.10</b>	<b>13.99</b>	<b>25.01</b>	<b>10.31</b>	<b>26.41</b>	1.14	<b>9.34</b>	<b>4.10</b>	<b>12.67</b>	<b>2.07</b>
F05	<b>7.10</b>	<b>17.59</b>	<b>28.46</b>	<b>10.61</b>	<b>27.11</b>	<b>2.20</b>	<b>15.27</b>	<b>5.67</b>	<b>13.18</b>	<b>2.11</b>
F06	−0.58	−0.85	4.41	<b>9243.65</b>	1.67	0.35	<b>2.44</b>	−1.06	<b>4.12</b>	<b>2.35</b>
F07	−2.33	−2.33	−0.14	<b>21.00</b>	−1.76	−2.23	1.43	−2.33	−1.57	1.00
F08	<b>21.12</b>	<b>31.27</b>	<b>32.65</b>	<b>18.89</b>	<b>31.13</b>	<b>5.79</b>	<b>17.55</b>	<b>6.28</b>	<b>23.89</b>	<b>4.18</b>
F09	<b>19.86</b>	<b>32.16</b>	<b>46.83</b>	<b>14.49</b>	<b>38.74</b>	<b>2.64</b>	<b>32.51</b>	<b>13.22</b>	<b>23.27</b>	<b>6.92</b>
F10	<b>15.06</b>	<b>46.11</b>	<b>40.41</b>	<b>28.72</b>	<b>38.60</b>	<b>8.71</b>	<b>30.16</b>	<b>14.05</b>	<b>27.79</b>	<b>19.30</b>
F11	<b>25.58</b>	<b>85.76</b>	<b>99.32</b>	<b>57.43</b>	<b>96.44</b>	<b>7.92</b>	<b>10.42</b>	<b>25.41</b>	<b>44.01</b>	<b>18.98</b>
F12	<b>30.94</b>	<b>42.36</b>	<b>73.43</b>	<b>36.94</b>	<b>74.79</b>	<b>4.10</b>	<b>26.88</b>	<b>8.75</b>	<b>47.81</b>	<b>5.00</b>
F13	<b>31.63</b>	<b>51.70</b>	<b>60.45</b>	<b>38.66</b>	<b>52.43</b>	<b>4.12</b>	<b>32.06</b>	<b>9.29</b>	<b>16.21</b>	<b>6.13</b>
Better	<b>11</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>11</b>	<b>10</b>	<b>12</b>	<b>11</b>	<b>12</b>	<b>12</b>
Equal	1	1	1	0	2	2	1	1	1	1
Worst	1	1	0	0	0	1	0	1	0	0

dimension is selected from the optimal individual of ABC. Otherwise, this value is selected from  $K^{best}$  of KH. This selection process can be formulated as Eq. (13):

$$TheBest_i = \begin{cases} Best_i & \text{if } r < P_{best} \\ Kbest_i & \text{otherwise} \end{cases} \quad (13)$$

where  $TheBest_i$  is the  $i$ -th dimension of  $TheBest$ ,  $Best_i$  is the  $i$ -th dimension of the best solution found by ABC,  $Kbest_i$  is the  $i$ -th dimension of  $K^{best}$  of the KH.  $r$  is a random number in the range of  $[0, 1]$ .

Based on the information exchange described above, the connection between the krill and bees in the KHABC method can be stated as follows.

The global part of the KHABC method is “the best”. Through the best, not only the ability of the KH from escaping from the local minima showed substantial improvement, but also the exploitation of ABC got significantly enhanced by the direct utilization of the global best solution.  $K^{best}$  of the KH is updated in terms of the best accordingly and the same is passed on to onlooker bees of ABC as neighbor.

Besides, a concentrated elitism strategy is introduced into KHABC to preserve the optimal solutions and not being ruined by the method. This is carried out in order to guarantee that the whole population is capable of proceeding with a better status than before. By introducing this concentrated elitism strategy into the algorithm, the KHABC has been



**Table 7** CEC 2017 constrained benchmark functions

Problem	Search range	Type of objective	Number of constraints	
			$E$	$I$
C01	$[-100, 100]D$	Non separable	0	1 Separable
C02	$[-100, 100]D$	Non separable, rotated	0	1 Non separable, rotated
C03	$[-100, 100]D$	Non separable	1 Separable	1 Separable
C04	$[-10, 10]D$	Separable	0	2 Separable
C05	$[-10, 10]D$	Non separable	0	2 Non separable, rotated
C06	$[-100, 100]D$	Separable	2 Non separable	0
C07	$[-10, 10]D$	Separable	2 Non separable	0
C08	$[-100, 100]D$	Separable	2 Non separable	0
C09	$[-100, 100]D$	Separable	1 Non separable	1 Non separable
C10	$[-100, 100]D$	Separable	0	2 Separable
C11	$[-100, 100]D$	Non separable	0	3 Separable
C12	$[-100, 100]D$	Separable	1 1	1 1
C13	$[-100, 100]D$	Non separable	Non separable	Separable 2
C14	$[-100, 100]D$	Separable	1	Non separable 2
C15	$[-100, 100]D$	Rotated	0	Rotated 3
C16	$[-100, 100]D$	Rotated	0 1	Rotated 1
C17	$[-100, 100]D$	Rotated	Rotated 1	Rotated 1
C18	$[-100, 100]^D$	Rotated	Rotated 1	Rotated 1
C19	$[-100, 100]D$	Rotated	Rotated 1	Rotated 1
C20	$[-100, 100]D$	Rotated	Rotated 1	Rotated 2
C21	$[-100, 100]D$	Rotated	Rotated	Rotated

$D$  is the number of decision variables,  $I$  is the number of inequality constraints,  $E$  is the number of equality constraints

**Table 8** Best function values of twenty-one CEC 2017 constrained benchmark functions

	CCS	DE	ES	GA	HS	HSBBO	KHE	KHABC	MBA	PSO	VNBA
C01	5.94E3	3.48E4	5.01E4	2.82E4	3.98E4	9.72E3	1.92E4	<b>1.74E3</b>	1.78E4	2.99E4	2.75E4
C02	4.48E3	4.17E4	4.69E4	3.33E4	3.98E4	6.41E3	1.48E4	<b>3.40E3</b>	1.56E4	3.15E4	1.97E4
C03	9.07E3	4.65E4	4.72E4	2.22E4	3.82E4	1.13E4	2.26E4	<b>7.52E3</b>	3.11E4	3.15E4	3.43E4
C04	353.10	280.50	700.60	457.60	664.80	161.80	247.30	<b>125.20</b>	186.20	442.10	358.00
C05	8.63E4	7.83E4	2.09E6	1.17E5	1.02E6	1.12E4	4.23E4	<b>8.45E3</b>	2.24E4	2.70E5	1.65E5
C06	983.00	980.50	1.03E3	924.60	1.04E3	<b>911.70</b>	999.20	998.70	1.02E3	1.03E3	929.80
C07	0.87	<b>0.10</b>	6.38	0.42	3.68	0.61	3.25	2.35	3.12	4.94	0.61
C08	963.50	975.00	1.03E3	953.40	1.02E3	<b>946.40</b>	994.00	975.50	1.02E3	1.02E3	957.50
C09	7.61E3	7.09E3	8.03E3	7.19E3	8.60E3	7.03E3	7.86E3	<b>6.02E3</b>	8.79E3	8.50E3	7.03E3
C10	3.30E3	6.47E3	3.74E4	1.04E4	3.49E4	<b>2.08E3</b>	4.35E3	2.17E3	9.20E3	1.48E4	7.35E3
C11	8.55E7	6.41E8	7.11E9	7.27E8	8.55E9	4.69E7	3.34E8	<b>9.73E6</b>	1.24E9	2.36E9	1.08E9
C12	28.30	56.63	69.16	56.12	67.69	32.72	26.04	<b>17.90</b>	44.55	51.97	40.57
C13	1.73	2.49	9.14	4.96	10.03	1.48	1.83	<b>1.31</b>	3.61	5.33	2.89
C14	4.36E3	7.24E3	3.50E4	1.61E4	3.68E4	2.32E3	3.48E3	<b>1.76E3</b>	6.23E3	1.61E4	7.98E3
C15	1.29E4	4.90E4	1.18E5	5.05E4	1.09E5	<b>4.52E3</b>	2.10E4	5.94E3	3.93E4	4.73E4	2.16E4
C16	1.77E9	2.12E10	8.81E10	7.52E9	8.02E10	5.98E8	8.94E9	<b>1.75E8</b>	1.59E10	2.24E10	8.68E9
C17	20.72	21.02	20.94	20.98	20.97	20.91	20.98	<b>20.63</b>	20.71	21.01	21.11
C18	44.20	101.70	136.60	86.80	123.90	46.36	59.53	<b>33.24</b>	76.40	84.09	86.95
C19	446.10	1.01E3	1.52E3	982.10	1.42E3	382.80	588.60	<b>324.10</b>	873.90	1.02E3	768.20
C20	3.81	9.92	31.25	13.09	32.17	3.02	5.75	<b>2.06</b>	9.06	13.15	10.27
C21	1.39E4	5.00E4	1.07E5	5.56E4	1.25E5	7.28E3	2.06E4	<b>4.92E3</b>	3.24E4	4.31E4	4.50E4
Total	0	1	0	0	0	4	0	16	0	0	0

further developed. The main steps of the proposed KHABC algorithm can be described as follows:

- Step 1 Initialize the KHABC (KH and ABC);
- Step 2 Determine  $K^{best}$  of the KH and best of the ABC;
- Step 3 Repeat
- Step 4 Apply the recombination procedure to the  $K^{best}$  and best solutions;
- Step 5 Save the *KEEP* best individuals as *BEST*;
- Step 6 KH process**

- 1) Motion Induced by other individuals;
- 2) Foraging motion;
- 3) Physical diffusion;
- 4) Determine the  $k^{best}$  of the population;

#### Step 7 ABC process

- 1) Employed bee phase of abc;
- 2) Onlooker bee phase of abc;
- 3) Scout bee phase of abc;
- 4) Determine the best of the population;

- Step 8 Replace the *KEEP* worst individuals with the *KEEP* best individuals saved in *BEST*;
- Step 9 Is termination condition met? If yes, output the best solution; or go to Step 3.

From the analysis above, we can see, the Step 6 KH progress and Step 7 ABC progress are simultaneously implemented. In this way, KHABC can find the best solution in a much faster way. In addition, the corresponding flowchart of the process is illustrated in Fig. 2.

Furthermore, for the complexity of the proposed KHABC algorithm, because KHABC did not introduce new operators except the original operators used in KH and ABC, the complexity of KHABC is no more than KH and ABC. Our proposed KHABC algorithm only used the best individual in KH and ABC, not introduced any new operators, therefore, KHABC is as simple as the basic KH and ABC.

#### 4.2 Convergent process of KHABC algorithm

As an example, the proposed KHABC algorithm is benchmarked using the 2D Griewank function (see Fig. 3). The formulation of the 2D Griewank function can be given as:

$$f(\vec{x}) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (14)$$

For the purpose of clarifying the movement process of the individuals in KHABC algorithm, less individuals (10) and less generations (4) are used to find the minima in this

**Table 9** Mean function values of twenty-one CEC 2017 constrained benchmark functions

	CCS	DE	ES	GA	HS	HSBBO	KHE	KHABC	MBA	PSO	VNBA
C01	1.17E4	6.77E4	7.65E4	5.91E4	6.57E4	2.30E4	3.81E4	<b>8.74E3</b>	4.87E4	5.51E4	5.58E4
C02	1.09E4	6.58E4	7.76E4	5.79E4	6.92E4	2.32E4	3.67E4	<b>8.58E3</b>	4.75E4	5.60E4	6.09E4
C03	<b>1.64E4</b>	6.45E4	8.86E4	5.93E4	7.67E4	2.55E4	4.09E4	1.81E4	8.85E4	6.04E4	7.49E4
C04	438.40	360.90	957.50	584.40	802.80	274.10	318.00	<b>234.40</b>	280.70	575.80	476.90
C05	2.11E5	1.69E5	3.55E6	3.44E5	2.19E6	4.63E4	1.59E5	<b>3.12E4</b>	1.31E5	8.15E5	5.44E5
C06	1.01E3	1.00E3	1.05E3	<b>953.90</b>	1.05E3	985.10	1.01E3	1.01E3	1.03E3	1.04E3	989.20
C07	2.31	1.57	8.41	<b>1.24</b>	7.40	4.53	5.73	4.70	5.85	6.71	4.82
C08	1.00E3	992.10	1.05E3	<b>977.00</b>	1.04E3	996.20	1.02E3	1.00E3	1.04E3	1.03E3	997.80
C09	8.21E3	7.15E3	8.30E3	7.31E3	8.88E3	7.24E3	8.23E3	7.51E3	9.09E3	8.81E3	<b>7.03E3</b>
C10	8.35E3	1.04E4	4.89E4	2.37E4	4.63E4	4.29E3	7.39E3	<b>3.68E3</b>	2.00E4	2.50E4	1.65E4
C11	3.97E8	1.60E9	1.83E10	3.04E9	1.56E10	3.09E8	9.03E8	<b>7.53E7</b>	4.45E9	5.12E9	3.38E9
C12	36.12	66.49	80.80	73.12	77.55	52.71	38.00	<b>27.80</b>	57.72	63.07	62.21
C13	2.78	3.47	12.60	7.05	13.12	1.98	2.93	<b>1.79</b>	6.67	6.88	4.99
C14	8.20E3	1.02E4	4.73E4	2.42E4	4.73E4	<b>4.49E3</b>	6.86E3	4.78E3	1.53E4	2.57E4	1.47E4
C15	2.47E4	6.84E4	1.67E5	9.00E4	1.68E5	1.67E4	3.78E4	<b>1.36E4</b>	8.28E4	8.96E4	8.04E4
C16	4.90E9	5.36E10	2.17E11	3.68E10	1.98E11	3.31E9	2.69E10	<b>1.89E9</b>	7.58E10	7.38E10	5.75E10
C17	21.09	21.17	21.14	21.15	21.16	21.11	21.16	20.95	<b>20.90</b>	21.16	21.20
C18	62.68	122.10	155.90	122.20	148.90	71.47	82.18	<b>55.31</b>	112.20	122.60	120.90
C19	659.90	1.25E3	1.76E3	1.40E3	1.71E3	634.30	769.20	<b>474.20</b>	1.21E3	1.33E3	1.26E3
C20	7.27	18.00	42.05	23.65	40.33	5.83	10.13	<b>4.00</b>	20.93	22.39	22.04
C21	2.64E4	7.07E4	1.61E5	8.66E4	1.64E5	1.91E4	3.68E4	<b>1.40E4</b>	6.65E4	8.37E4	7.56E4
Total	1	0	0	3	0	1	0	14	1	0	1

problem. The optimization process of KHABC algorithm can be shown in Fig. 4, where the position of individuals and the best solution are shown by blue  $\circ$  marks and red  $\bullet$  mark, respectively. The positions in the spread of the individuals after 1st, 2nd, 3rd and 4th generations are respectively shown in Fig. 4a–d. As is evident, firstly the positions of the individuals are scattered over the search space and far away from the best solution (see Fig. 4a). Subsequently, all the individuals move towards the best solution with increasing the generations (see Fig. 4b–c). Ultimately, all the individuals are located around the best solution (see Fig. 4d).

## 5 Simulations

In this section, the proposed KHABC algorithm will be benchmarked by thirteen basic unstrained benchmark functions, twenty-one CEC 2017 constrained optimization functions, and ten CEC 2011 real world problems (RWPs). And to ensure a fair comparison, all the simulations were implemented in the same environments as shown in [29].

Here, the performance of KHABC was compared with ten nature-inspired methods viz. chaotic cuckoo search

(CCS) [30], DE [3], ES [4], GA [5], HS [9], HSBBO [31], KH with elitism (KHE) [32], multi-swarm bat algorithm (MBA) [33], PSO [11], and bat algorithm with variable neighborhood search (VNBA) [34].

For KH and KHABC, the same parameters are set:  $V_f = 0.02$ ,  $D^{\max} = 0.005$ ,  $N^{\max} = 0.01$ , and  $KEEP=2$ . For the other methods, the parameters are set suggested in [31]. For CCS, HSBBO, KHE, MBA, and VNBA, their parameters are set as the original paper.

It is well-known that most of the metaheuristic methods are based on certain type of stochastic distribution. To obtain typical performances, fifty trials are implemented for each method on each function. The optimal function value is highlighted in bold. If no special clarification is provided, the dimension of the benchmark is set to thirty. In addition, both the population size and maximum generation are set to 50.

### 5.1 Unconstrained optimization

Here, the proposed KHABC algorithm is verified by thirteen basic standard benchmark functions.

**Table 10** Worst function values of twenty-one CEC 2017 constrained benchmark functions

	CCS	DE	ES	GA	HS	HSBBO	KHE	KHABC	MBA	PSO	VNBA
C01	1.82E4	9.11E4	1.06E5	1.19E5	8.22E4	4.85E4	6.25E4	<b>1.47E4</b>	9.82E4	8.16E4	1.04E5
C02	<b>1.88E4</b>	8.33E4	1.10E5	9.21E4	8.91E4	4.31E4	5.51E4	2.30E4	1.05E5	8.62E4	1.14E5
C03	<b>2.79E4</b>	8.10E4	1.35E5	1.26E5	1.17E5	4.18E4	7.24E4	4.26E4	1.79E5	8.71E4	1.66E5
C04	512.10	411.40	1.12E3	730.60	903.00	<b>367.60</b>	384.60	378.50	386.40	692.00	655.90
C05	4.54E5	2.93E5	4.98E6	1.04E6	3.49E6	2.27E5	3.43E5	<b>8.64E4</b>	7.48E5	1.67E6	2.03E6
C06	1.03E3	1.03E3	1.07E3	<b>996.90</b>	1.06E3	1.07E3	1.03E3	1.02E3	1.05E3	1.05E3	1.06E3
C07	4.68	<b>2.86</b>	10.30	5.10	9.31	11.08	7.86	6.62	9.33	9.26	9.35
C08	1.02E3	<b>1.01E3</b>	1.05E3	1.03E3	1.06E3	1.06E3	1.03E3	1.02E3	1.06E3	1.05E3	1.06E3
C09	8.63E3	7.22E3	8.57E3	7.54E3	9.12E3	7.69E3	8.47E3	8.62E3	9.38E3	9.03E3	<b>7.03E3</b>
C10	1.42E4	1.51E4	5.93E4	3.40E4	5.92E4	7.16E3	1.45E4	<b>5.53E3</b>	3.39E4	3.68E4	3.36E4
C11	1.07E9	3.06E9	3.21E10	8.48E9	2.34E10	1.45E9	2.00E9	<b>2.70E8</b>	1.14E10	1.01E10	1.87E10
C12	46.98	71.43	89.80	89.32	85.99	88.48	48.56	<b>42.27</b>	75.86	75.92	87.26
C13	4.48	4.35	16.15	11.29	16.38	2.65	4.89	<b>2.37</b>	11.63	9.30	7.90
C14	1.45E4	1.52E4	5.96E4	3.73E4	5.83E4	<b>7.56E3</b>	1.09E4	8.82E3	2.90E4	3.73E4	3.48E4
C15	3.60E4	8.67E4	2.15E5	1.35E5	2.18E5	3.09E4	6.24E4	<b>2.65E4</b>	1.32E5	1.31E5	1.47E5
C16	1.11E10	9.06E10	3.41E11	9.98E10	3.10E11	9.51E9	4.84E10	<b>8.29E9</b>	1.99E11	1.59E11	2.08E11
C17	21.19	21.24	21.28	21.25	21.28	21.21	21.25	21.24	<b>21.00</b>	21.25	21.30
C18	<b>80.73</b>	144.50	182.10	152.30	163.70	99.80	116.10	84.22	156.60	152.30	160.70
C19	945.20	1.52E3	2.08E3	1.83E3	1.88E3	878.50	1.00E3	<b>634.30</b>	1.60E3	1.62E3	1.90E3
C20	12.49	26.60	52.26	32.51	53.93	11.38	15.79	<b>6.88</b>	44.17	33.21	50.98
C21	4.23E4	9.39E4	2.18E5	1.49E5	1.93E5	4.77E4	6.33E4	<b>2.65E4</b>	1.23E5	1.12E5	1.59E5
Total	3	2	0	1	0	2	0	11	1	0	1

### 5.1.1 Benchmark evaluation

In order to validate KHABC, it has been applied to optimize a series of benchmark functions from previous studies (see Table 1) [35].

The characteristics of the thirteen functions are provided in Table 1, including optimal value, dimension, separability, modality, lower bound and upper bound. Their best, average, worst, and Std values are recorded, as shown in Tables 2, 3, 4 and 5.

From Table 2, it can be seen that HSBBO has the best performance on eleven of the thirteen test problems. Furthermore, the performance of KHABC is only worse than HSBBO. For average solutions shown in Table 3, KHABC provides the best results for eleven of the thirteen test problems, while DE and MBA are only inferior to KHABC, and have the best performance on only one function. Furthermore, for the worst performance, as shown in Table 4, KHABC performs the best on eleven functions out of thirteen functions, and MBA performs much better than other comparative algorithms, which can find the best solutions on two functions. For the Std of eleven algorithms on thirteen functions (see Table 5), KHABC has the minimum on ten out of thirteen functions. This indicates that KHABC can be implemented in a more stable way.

### 5.1.2 Comparisons with other optimization methods by using *t*-test

According to the final function values of fifty independent runs on 30-D thirteen functions, as shown in Sect. 5.1.1, the *t* values on thirteen test problems of the two-tailed test with the 5% level of significance between KHABC method and other ten metaheuristic methods are provided in this section. The results are recorded in Table 6. In the table, the value of *t* with 98 degrees of freedom is significant at  $\alpha = 0.05$  by a two-tailed test. The result is highlighted in bold font for showing the better performance of KHABC w.r.t. comparative method. The “Better”, “Equal”, and “Worse” in the last three rows indicate better than, equal to and worse performance respectively of the KHABC as compared to the comparative one. Here, the comparison between KHABC and MBA is taken an example. KHABC has better and worse performance than MBA on eleven and one test problems, respectively. The performance between KHABC and KH has no significant differences on one test problem. To summarize, it can be said that KHABC outperforms MBA on most test problems. In addition, for KHABC and HSBBO, HSBBO yields better and worse performance than KHABC on one and ten test problems respectively and they have similar performance on two test problems. The two examples

**Table 11** The Std of twenty-one CEC 2017 constrained benchmark functions

	CCS	DE	ES	GA	HS	HSBBO	KHE	KHABC	MBA	PSO	VNBA
C01	<b>2.96E3</b>	1.03E4	1.32E4	1.81E4	1.09E4	9.28E3	1.01E4	3.24E3	1.79E4	1.18E4	1.86E4
C02	<b>2.71E3</b>	7.90E3	1.57E4	1.58E4	1.08E4	8.79E3	9.23E3	4.01E3	2.01E4	1.21E4	2.19E4
C03	<b>4.44E3</b>	9.12E3	1.73E4	1.90E4	1.70E4	7.69E3	9.51E3	8.29E3	3.31E4	1.31E4	2.63E4
C04	41.26	<b>25.16</b>	89.65	63.80	64.84	44.00	31.39	46.15	49.25	55.79	78.00
C05	8.77E4	4.38E4	7.94E5	1.86E5	4.78E5	3.57E4	6.09E4	<b>2.02E4</b>	1.21E5	3.05E5	3.45E5
C06	8.47	10.32	7.90	19.18	7.02	44.53	5.11	<b>3.36</b>	8.49	6.13	30.06
C07	0.90	<b>0.62</b>	1.11	1.10	1.00	3.25	1.13	0.93	1.07	0.98	2.43
C08	8.46	8.01	<b>5.82</b>	18.47	8.30	34.84	8.47	8.21	9.63	7.05	22.19
C09	199.60	28.22	142.30	78.51	124.20	215.20	136.70	613.20	140.20	99.48	<b>9.17E–13</b>
C10	2.41E3	1.63E3	5.53E3	4.94E3	5.49E3	1.25E3	1.89E3	<b>991.60</b>	5.51E3	5.06E3	6.03E3
C11	2.37E8	4.99E8	4.49E9	1.83E9	3.27E9	2.66E8	3.20E8	<b>4.87E7</b>	2.44E9	1.85E9	3.29E9
C12	4.42	<b>3.24</b>	4.79	8.31	3.99	13.83	5.32	4.45	7.88	5.81	11.51
C13	0.55	0.39	1.64	1.42	1.28	0.29	0.54	<b>0.28</b>	1.56	1.02	1.12
C14	2.04E3	1.71E3	5.61E3	4.68E3	5.37E3	<b>1.31E3</b>	1.64E3	1.46E3	4.61E3	4.31E3	4.40E3
C15	6.47E3	8.79E3	2.04E4	1.64E4	2.12E4	6.04E3	7.99E3	<b>4.51E3</b>	2.28E4	1.83E4	2.63E4
C16	2.20E9	1.41E10	5.32E10	2.15E10	4.69E10	2.00E9	9.63E9	<b>1.38E9</b>	4.84E10	3.07E10	3.80E10
C17	0.08	<b>0.05</b>	0.07	0.06	0.06	0.07	0.06	0.17	0.06	0.05	0.05
C18	<b>8.52</b>	11.16	10.61	14.51	10.33	12.95	11.27	12.42	19.46	14.44	17.69
C19	106.70	107.80	130.20	194.50	102.40	109.40	86.79	<b>78.46</b>	180.60	145.30	234.70
C20	1.73	3.31	4.93	4.85	4.40	1.85	2.06	<b>1.19</b>	6.85	4.54	10.01
C21	6.99E3	9.86E3	2.18E4	1.82E4	1.77E4	8.13E3	8.05E3	<b>5.45E3</b>	1.99E4	1.80E4	2.53E4
Total	4	4	1	0	0	1	0	10	0	0	1

above indicate that KHABC significantly outperforms MBA and HSBBO on most benchmarks. Looking carefully at the results as shown in Table 6, it is safe to say, KHABC is a competitive and promising method on most cases when compared to the other ten methods.

### 5.2 Constrained optimization

Here, the proposed KHABC algorithm is verified by twenty-one CEC 2017 constrained benchmark functions (C01–C21).

#### 5.2.1 Benchmark evaluation

The characteristics of the twenty-one CEC 2017 constrained benchmark functions (C01–C21) can be found in Table 7, including the number of decision variables and constraints (inequality constraints and equality constraints). Their best, average, worst, and Std values through fifty independent runs are recorded, as shown in Tables 8, 9, 10 and 11.

From Table 8, it can be seen that HSBBO has the best performance on four of the twenty-one test problems, while KHABC performs the best on sixteen functions which is much better than HSBBO. For average solutions shown in Table 9, KHABC can find the best results for fourteen of the twenty-one test problems, while GA is only inferior to

KHABC, and has the best performance on only three functions (C06–C08). Furthermore, for the worst performance, as shown in Table 10, KHABC performs the best on eleven functions out of twenty-one functions, and CCS can find the best solutions on three functions (C02, C03, and C18). For other algorithms, DE and HSBBO have the similar performance, and perform the best on two functions (C07, C08 and C04, C14). For the Std of eleven algorithms on twenty-one constrained functions, although CCS and de have the minimum Std on four functions, they are far worse than KHABC, which can find the solutions within the least range on ten functions. This indicates that KHABC is more suitable algorithm to solve the real world problems.

#### 5.2.2 Comparisons with other optimization methods by using *t*-test

According to the final function values of fifty independent runs on 30-D twenty-one CEC 2017 constrained functions as shown in Sect. 5.2.1, the *t* values on thirteen test problems of the two-tailed test with the 5% level of significance between KHABC method and other ten metaheuristic methods are provided in this section. The results are recorded in Table 12. In Table 12, the value of *t* with 98 degrees of freedom is significant at  $\alpha = 0.05$  by a two-tailed test. The

**Table 12** Comparisons between KHABC and other methods at  $\alpha = 0.05$  on a two-tailed  $t$ -tests for twenty-one CEC 2017 constrained benchmark functions

	CCS	DE	ES	GA	HS	HSBBO	KHE	MBA	PSO	VNBA
C01	<b>4.40</b>	<b>36.76</b>	<b>34.81</b>	<b>19.10</b>	<b>36.80</b>	<b>10.77</b>	<b>18.37</b>	<b>15.66</b>	<b>26.80</b>	<b>17.73</b>
C02	<b>3.51</b>	<b>45.24</b>	<b>30.85</b>	<b>20.82</b>	<b>39.13</b>	<b>11.00</b>	<b>19.53</b>	<b>13.00</b>	<b>26.28</b>	<b>16.87</b>
C03	-2.00	<b>26.02</b>	<b>25.48</b>	<b>14.03</b>	<b>21.03</b>	<b>4.03</b>	<b>12.04</b>	<b>14.13</b>	<b>19.32</b>	<b>14.10</b>
C04	<b>23.31</b>	<b>17.02</b>	<b>50.71</b>	<b>31.43</b>	<b>50.51</b>	<b>4.40</b>	<b>10.59</b>	<b>4.85</b>	<b>33.34</b>	<b>18.92</b>
C05	<b>14.32</b>	<b>19.77</b>	<b>31.65</b>	<b>12.08</b>	<b>31.61</b>	<b>2.73</b>	<b>13.56</b>	<b>6.16</b>	<b>17.96</b>	<b>10.35</b>
C06	<b>2.91</b>	-0.53	<b>40.15</b>	-18.50	<b>42.73</b>	-3.12	<b>10.36</b>	<b>21.73</b>	<b>33.87</b>	-3.63
C07	-13.04	-19.82	<b>18.11</b>	-17.01	<b>13.93</b>	-0.37	<b>4.94</b>	<b>5.70</b>	<b>10.50</b>	0.31
C08	0.89	-5.83	<b>30.81</b>	-8.60	<b>23.62</b>	-1.06	<b>8.08</b>	<b>19.05</b>	<b>20.18</b>	-1.13
C09	<b>7.69</b>	-4.14	<b>8.89</b>	-2.25	<b>15.48</b>	-2.90	<b>8.14</b>	<b>17.78</b>	<b>14.88</b>	-5.48
C10	<b>12.69</b>	<b>26.23</b>	<b>56.80</b>	<b>27.23</b>	<b>54.74</b>	<b>2.71</b>	<b>12.33</b>	<b>21.36</b>	<b>29.39</b>	<b>14.91</b>
C11	<b>9.56</b>	<b>22.45</b>	<b>28.85</b>	<b>11.40</b>	<b>34.03</b>	<b>5.88</b>	<b>17.68</b>	<b>12.68</b>	<b>19.52</b>	<b>7.12</b>
C12	<b>9.38</b>	<b>49.68</b>	<b>57.27</b>	<b>34.00</b>	<b>58.83</b>	<b>12.13</b>	<b>10.40</b>	<b>23.37</b>	<b>34.08</b>	<b>19.72</b>
C13	<b>11.38</b>	<b>24.89</b>	<b>46.00</b>	<b>25.75</b>	<b>61.15</b>	<b>3.16</b>	<b>13.28</b>	<b>21.75</b>	<b>34.02</b>	<b>19.57</b>
C14	<b>9.65</b>	<b>17.14</b>	<b>52.54</b>	<b>28.21</b>	<b>53.87</b>	-1.05	<b>6.72</b>	<b>15.56</b>	<b>31.53</b>	<b>15.53</b>
C15	<b>9.97</b>	<b>39.13</b>	<b>52.60</b>	<b>31.47</b>	<b>50.76</b>	<b>2.17</b>	<b>17.97</b>	<b>21.10</b>	<b>28.61</b>	<b>17.89</b>
C16	<b>8.25</b>	<b>25.05</b>	<b>27.95</b>	<b>11.14</b>	<b>29.18</b>	<b>4.28</b>	<b>17.74</b>	<b>10.78</b>	<b>16.54</b>	<b>10.27</b>
C17	<b>5.25</b>	<b>8.71</b>	<b>7.34</b>	<b>7.89</b>	<b>8.27</b>	<b>6.43</b>	<b>8.40</b>	-1.96	<b>8.32</b>	<b>10.18</b>
C18	<b>3.46</b>	<b>28.31</b>	<b>43.53</b>	<b>24.78</b>	<b>40.95</b>	<b>6.37</b>	<b>11.33</b>	<b>17.42</b>	<b>24.98</b>	<b>21.46</b>
C19	<b>9.91</b>	<b>41.16</b>	<b>59.65</b>	<b>31.17</b>	<b>67.82</b>	<b>8.41</b>	<b>17.83</b>	<b>26.48</b>	<b>36.74</b>	<b>22.30</b>
C20	<b>11.03</b>	<b>28.15</b>	<b>53.01</b>	<b>27.84</b>	<b>56.34</b>	<b>5.90</b>	<b>18.22</b>	<b>17.23</b>	<b>27.74</b>	<b>12.66</b>
C21	<b>9.69</b>	<b>35.51</b>	<b>48.41</b>	<b>27.25</b>	<b>57.53</b>	<b>4.20</b>	<b>16.35</b>	<b>18.27</b>	<b>25.65</b>	<b>16.91</b>
Better	<b>18</b>	<b>17</b>	<b>21</b>	<b>17</b>	<b>21</b>	<b>16</b>	<b>21</b>	<b>20</b>	<b>21</b>	<b>17</b>
Equal	2	1	0	0	0	3	0	1	0	2
Worse	1	3	0	4	0	2	0	0	0	2

**Table 13** CEC 2011 real world application problems

No.	Function
R01	Parameter estimation for frequency-modulated (FM) sound waves
R02	Lennard-Jones potential problem
R03	Optimal control of a non-linear stirred tank reactor
R04	Tersoff potential for model Si (B)
R05	Spread spectrum radar polly phase
R06 <sup>a</sup>	DED instance 1
R07 <sup>a</sup>	ELD instance 1
R08 <sup>a</sup>	ELD instance 2
R09 <sup>a</sup>	ELD instance 3
R10 <sup>a</sup>	ELD instance 4

<sup>a</sup> f06–f10 are five instances of the ELD problems in CEC 2011. DED and ELD denote dynamic economic dispatch and economic load dispatch, respectively

result is highlighted in bold font for showing the better performance of KHABC w.r.t. comparative method. The “Better”, “Equal”, and “Worse” in the last three rows indicate better than, equal to and worse performance respectively of the

KHABC as compared to the comparative one. Here, the comparison between KHABC and HSBBO is taken an example. KHABC has better and worse performance than HSBBO on sixteen and two test problems, respectively. The performance between KHABC and HSBBO has no significant differences on two test problems. To summarize, it can be said that KHABC outperforms HSBBO on most test problems. In addition, for KHABC and CCS, CCS yields better and worse performance than KHABC on one and eighteen test problems respectively and they have similar performance on two test problems. The two examples above indicate that KHABC significantly outperforms HSBBO and CCS on almost all the benchmarks. Looking carefully at the results as shown in Table 12, it is safe to say, KHABC is a competitive and promising method on most cases when compared to the other ten methods.

### 5.3 Real world problems

The target of designing a new metaheuristic algorithm is to solve the practical engineering problems. Therefore, except the benchmark evaluation conducted in Sects. 5.1 and 5.2, ten real world problems selected from CEC 2011 (R01–R10) is used to further verify the proposed KHABC algorithm.



**Table 14** Best function values of ten CEC 2011 real world application problems

	CCS	DE	ES	GA	HS	HSBBO	KHE	KHABC	MBA	PSO	VNBA
R01	15.61	23.12	23.53	14.85	20.80	12.24	17.20	12.31	<b>10.75</b>	22.14	19.51
R02	-7.95	-8.37	-5.42	-12.34	-9.12	-8.30	-10.02	<b>-17.61</b>	-13.03	-4.02	-8.70
R03	14.37	14.38	13.95	17.56	14.34	14.76	14.33	<b>13.77</b>	13.94	13.94	18.78
R04	203.20	173.40	368.70	355.50	214.30	209.60	203.20	<b>-17.65</b>	98.79	539.30	222.90
R05	1.50	1.87	2.12	1.01	1.84	1.56	1.54	<b>0.50</b>	1.14	1.83	1.80
R06	2.19E6	1.78E8	2.23E8	4.70E6	2.03E8	<b>1.06E6</b>	3.49E7	1.86E6	4.63E6	6.00E7	8.29E7
R07	1.55E4	1.57E4	1.52E4	1.57E4	1.54E4	1.55E4	1.55E4	<b>1.55E4</b>	1.55E4	1.50E4	1.53E4
R08	1.95E4	1.98E4	<b>1.85E4</b>	1.97E4	1.97E4	1.92E4	1.96E4	1.85E4	1.94E4	1.95E4	1.80E4
R09	<b>3.21E4</b>	3.32E4	4.05E4	3.29E4	6.90E4	3.36E4	3.36E4	3.35E4	8.83E4	4.65E4	3.38E4
R10	<b>1.37E5</b>	1.41E5	1.55E5	1.36E5	3.11E5	1.43E5	1.32E5	1.53E5	1.37E7	4.53E5	1.51E5
Total	2	0	1	0	0	1	0	5	1	0	0

**Table 15** Mean function values of ten CEC 2011 real world application problems

	CCS	DE	ES	GA	HS	HSBBO	KHE	KHABC	MBA	PSO	VNBA
R01	24.40	25.45	28.54	24.13	26.50	24.21	23.96	<b>22.31</b>	22.98	26.71	27.39
R02	-5.51	-5.18	-2.08	24.56	-5.58	-5.32	-5.49	<b>-13.09</b>	-7.07	-0.38	-5.27
R03	18.21	19.69	20.51	21.20	20.01	20.47	18.99	18.96	19.68	<b>18.16</b>	21.86
R04	487.60	646.30	434.70	400.30	652.00	513.10	618.10	<b>83.11</b>	462.10	804.90	675.40
R05	2.01	2.17	2.67	1.72	2.25	1.93	2.06	<b>1.38</b>	1.57	2.32	2.28
R06	2.93E6	2.20E8	2.84E8	9.93E6	2.51E8	3.92E6	5.62E7	<b>2.13E6</b>	4.35E7	1.00E8	2.79E8
R07	1.50E4	1.56E4	1.94E4	1.59E4	2.10E4	1.52E4	1.53E4	<b>1.50E4</b>	5.77E4	1.70E4	2.37E4
R08	1.92E4	2.61E4	<b>1.89E4</b>	2.29E4	3.11E4	2.56E4	2.46E4	1.91E4	1.95E4	2.91E4	4.54E4
R09	4.38E4	<b>3.35E4</b>	1.97E5	3.79E5	3.28E5	1.98E5	1.07E5	2.12E5	2.45E6	2.74E5	3.48E5
R10	<b>1.41E5</b>	1.61E5	9.19E5	1.58E5	2.30E7	1.63E5	2.09E5	1.22E6	5.35E7	1.71E7	5.82E5
Total	1	1	1	0	0	0	0	6	0	1	0

**Table 16** Worst function values of ten CEC 2011 real world application problems

	CCS	DE	ES	GA	HS	HSBBO	KHE	KHABC	MBA	PSO	VNBA
R01	27.77	<b>27.18</b>	30.46	31.01	29.71	31.01	28.11	27.79	27.75	29.01	30.54
R02	-3.09	-2.61	-0.55	1.05E3	-2.81	-1.53	-2.63	<b>-9.78</b>	-4.73	5.03	-1.82
R03	<b>21.01</b>	21.08	21.89	22.44	22.91	21.74	21.54	21.08	21.08	21.08	24.47
R04	613.80	1.18E3	542.70	473.60	1.14E3	1.16E3	1.56E3	<b>347.10</b>	746.50	1.21E3	1.12E3
R05	2.45	2.38	3.08	2.37	2.52	2.39	2.46	<b>1.84</b>	2.22	2.76	2.68
R06	8.45E6	2.62E8	3.36E8	4.08E7	2.87E8	7.59E6	7.87E7	<b>2.46E6</b>	1.86E8	1.53E8	3.73E8
R07	<b>1.57E4</b>	1.50E4	3.52E4	1.53E4	6.34E4	1.53E4	1.57E4	1.55E4	2.36E5	3.18E4	2.75E5
R08	2.03E4	4.44E4	<b>1.95E4</b>	3.20E4	7.04E4	3.76E4	4.82E4	1.99E4	1.99E4	5.77E4	1.19E5
R09	1.74E5	<b>3.36E4</b>	3.68E5	2.82E6	1.21E6	4.94E6	3.22E5	5.81E5	1.21E7	5.02E5	5.02E6
R10	<b>1.67E5</b>	2.76E5	2.84E6	1.83E5	4.99E7	2.02E5	4.61E5	1.91E7	9.25E7	3.32E7	2.63E6
Total	3	2	1	0	0	0	0	4	0	0	0

### 5.3.1 Performance

Ten selected CEC 2011 real world problems can be shown in Table 13. More information about these real world problems

can be found in [36]. Their best, average, worst, and Std values through fifty independent runs are recorded, as shown in Tables 14, 15, 16 and 17.

**Table 17** The Std of ten CEC 2011 real world application problems

	CCS	DE	ES	GA	HS	HSBBO	KHE	KHABC	MBA	PSO	VNBA
R01	2.38	<b>0.89</b>	1.52	3.36	1.86	3.58	2.44	3.53	3.94	1.30	2.46
R02	1.19	1.12	<b>1.08</b>	154.70	1.38	1.28	1.28	1.85	1.78	1.79	1.36
R03	2.50	2.45	1.92	<b>0.71</b>	2.40	1.66	2.82	2.95	2.42	2.59	0.92
R04	101.50	216.20	39.37	<b>31.25</b>	252.20	163.60	284.00	83.34	120.30	137.10	224.30
R05	0.21	<b>0.14</b>	0.24	0.33	0.17	0.20	0.22	0.28	0.23	0.17	0.19
R06	1.29E6	2.05E7	2.19E7	7.54E6	1.92E7	1.90E6	1.03E7	<b>1.08E5</b>	3.31E7	2.30E7	7.47E7
R07	<b>29.99</b>	66.31	4.62E3	97.60	7.46E3	46.81	47.42	37.00	5.32E4	2.71E3	3.84E4
R08	309.50	6.40E3	<b>146.90</b>	3.46E3	1.08E4	4.68E3	6.01E3	174.10	166.90	1.04E4	2.98E4
R09	2.71E4	<b>131.10</b>	8.20E4	6.07E5	1.78E5	7.29E5	7.11E4	1.32E5	2.95E6	9.92E4	9.02E5
R10	<b>7.01E3</b>	2.03E4	7.13E5	1.01E4	1.30E7	1.13E4	7.49E4	2.99E6	1.96E7	9.65E6	4.62E5
Total	2	3	2	2	0	0	0	1	0	0	0

**Table 18** Comparisons between KHABC and other methods at  $\alpha = 0.05$  on a two-tailed  $t$ -tests for ten CEC 2011 real world application problems

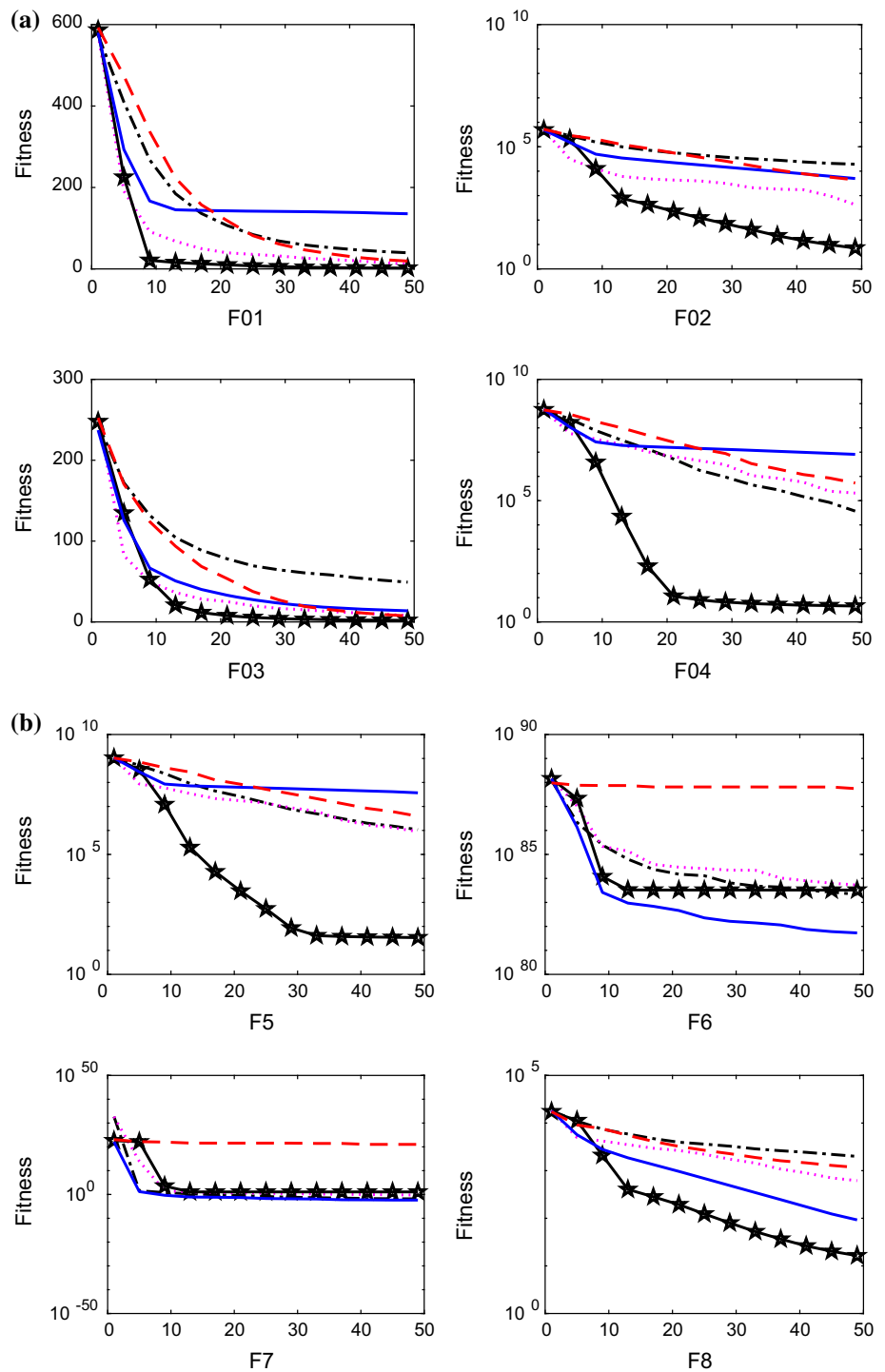
	CCS	DE	ES	GA	HS	HSBBO	KHE	MBA	PSO	VNBA
R01	<b>3.46</b>	<b>6.10</b>	<b>11.45</b>	<b>2.64</b>	<b>7.42</b>	<b>2.67</b>	<b>2.72</b>	0.89	<b>8.27</b>	<b>8.35</b>
R02	<b>24.40</b>	<b>25.84</b>	<b>36.32</b>	1.72	<b>22.99</b>	<b>24.42</b>	<b>23.88</b>	<b>16.60</b>	<b>34.95</b>	<b>24.11</b>
R03	-1.36	1.36	<b>3.12</b>	<b>5.23</b>	1.96	<b>3.16</b>	0.06	1.34	-1.43	<b>6.64</b>
R04	<b>21.77</b>	<b>17.18</b>	<b>26.98</b>	<b>25.20</b>	<b>15.15</b>	<b>16.56</b>	<b>12.78</b>	<b>18.31</b>	<b>31.81</b>	<b>17.51</b>
R05	<b>12.62</b>	<b>17.74</b>	<b>24.50</b>	<b>5.47</b>	<b>18.61</b>	<b>11.15</b>	<b>13.45</b>	<b>3.58</b>	<b>19.94</b>	<b>18.53</b>
R06	<b>4.56</b>	<b>78.47</b>	<b>94.72</b>	<b>7.31</b>	<b>94.29</b>	<b>6.50</b>	<b>36.46</b>	<b>8.60</b>	<b>29.56</b>	<b>25.82</b>
R07	0.48	<b>6.27</b>	<b>5.42</b>	<b>6.58</b>	<b>5.41</b>	<b>3.07</b>	<b>3.74</b>	<b>5.64</b>	<b>4.98</b>	1.50
R08	<b>4.15</b>	<b>7.52</b>	-9.85	<b>7.11</b>	<b>8.14</b>	<b>8.85</b>	<b>5.93</b>	1.22	<b>7.05</b>	<b>6.27</b>
R09	-8.78	-9.50	-0.88	1.83	<b>3.68</b>	-0.18	-4.83	<b>5.47</b>	<b>2.64</b>	1.06
R10	-2.70	-2.66	-0.86	-2.68	<b>11.11</b>	-2.65	-2.54	<b>18.43</b>	<b>11.40</b>	-1.64
Better	6	7	7	7	9	8	7	7	9	7
Equal	2	1	2	2	1	1	1	3	1	3
Worse	2	2	1	1	0	1	2	0	0	0

From Table 13, it can be seen that CCS has the best performance on two of the ten real world problems, while KHABC performs the best on five functions which is much better than CCS. For average solutions shown in Table 14, KHABC can find the best results on six of the ten real world problems, while CCS, DE, ES, and PSO have the similar performance, and all of them have the best performance on only one real world problem (R10, R09, R08, and R03). Furthermore, for the worst performance, as shown in Table 15, KHABC performs the best on four out of ten real world problems, and CCS, DE, and ES rank two, three, and four, respectively, which can find the best solutions on three, two, and on real world problems (C02, C03, and C18). For the Std of eleven algorithms on ten real world problems, DE has the best performance and has the minimum Std values on three real world problems. CCS, DE, and GA have the similar performance, and they can find the solutions with the minimum Std values on two real world problems. KHABC can find the solutions within the least range on one real world problem.

### 5.3.2 Comparisons with other optimization methods by using $t$ -test

According to the final function values of fifty independent runs on ten CEC 2011 real world problems, as shown in Sect. 5.3.1, the  $t$  values on ten real world problems of the two-tailed test with the 5% level of significance between KHABC method and other ten metaheuristic methods are provided in this section. The results are recorded in Table 18. In Table 18, the value of  $t$  with 98 degrees of freedom is significant at  $\alpha = 0.05$  by a two-tailed test. The result is highlighted in bold font for showing the better performance of KHABC w.r.t. comparative method. The “Better”, “Equal”, and “Worse” in the last three rows indicate better than, equal to and worse performance respectively of the KHABC as compared to the comparative one. Here, the comparison between KHABC and CCS is taken an example. KHABC has better and worse performance than CCS on six and two real world problems, respectively. The performance between KHABC and CCS

**Fig. 5** Performance of different methods for the thirteen unconstrained benchmark functions and twenty-one CEC 2017 constrained functions. **a** F01–F04. **b** F05–F08. **c** F09–F12. **d** F13 and C01–C03. **e** C04–C07. **f** C08–C11. **g** C12–C15. **h** C16–C19. **i** C20–C21 and legend

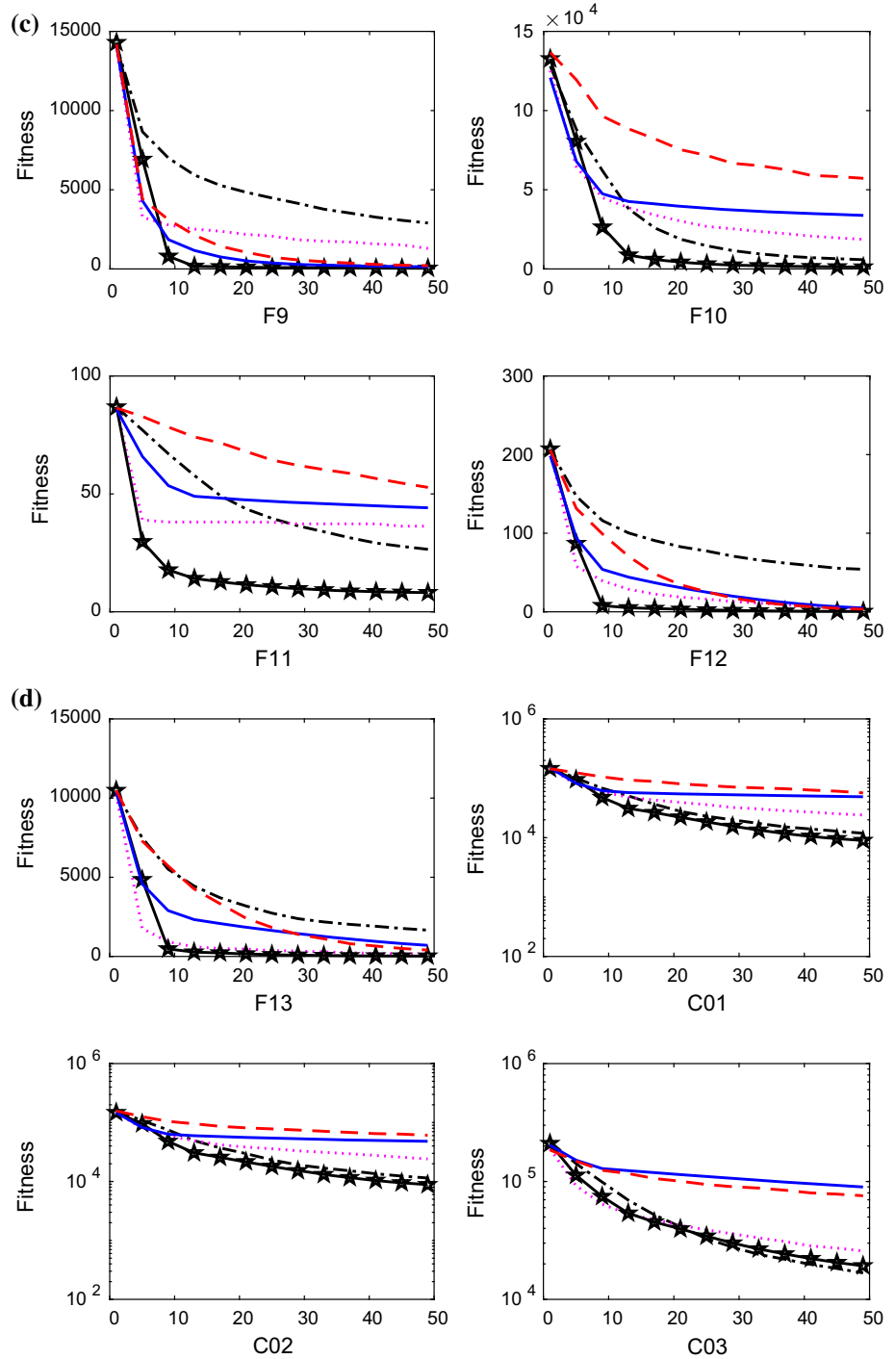


has no significant differences on two real world problems. To summarize, it can be said that KHABC outperforms CCS on most real world problems. In addition, for KHABC and VNBA, VNBA yields worse performance than KHABC on seven real world problems and they have similar performance on three RWPs. The two examples above indicate that KHABC significantly outperforms CCS and VNBA on

almost all the benchmarks. Looking carefully at the results as shown in Table 18, it is safe to say, KHABC is a competitive and promising method on most cases when compared to the other ten methods.

Furthermore, in order to prove the advantages of KHABC method over other algorithms, convergence maps of the five selected representative methods (CCS, HSBBO, KHABC,

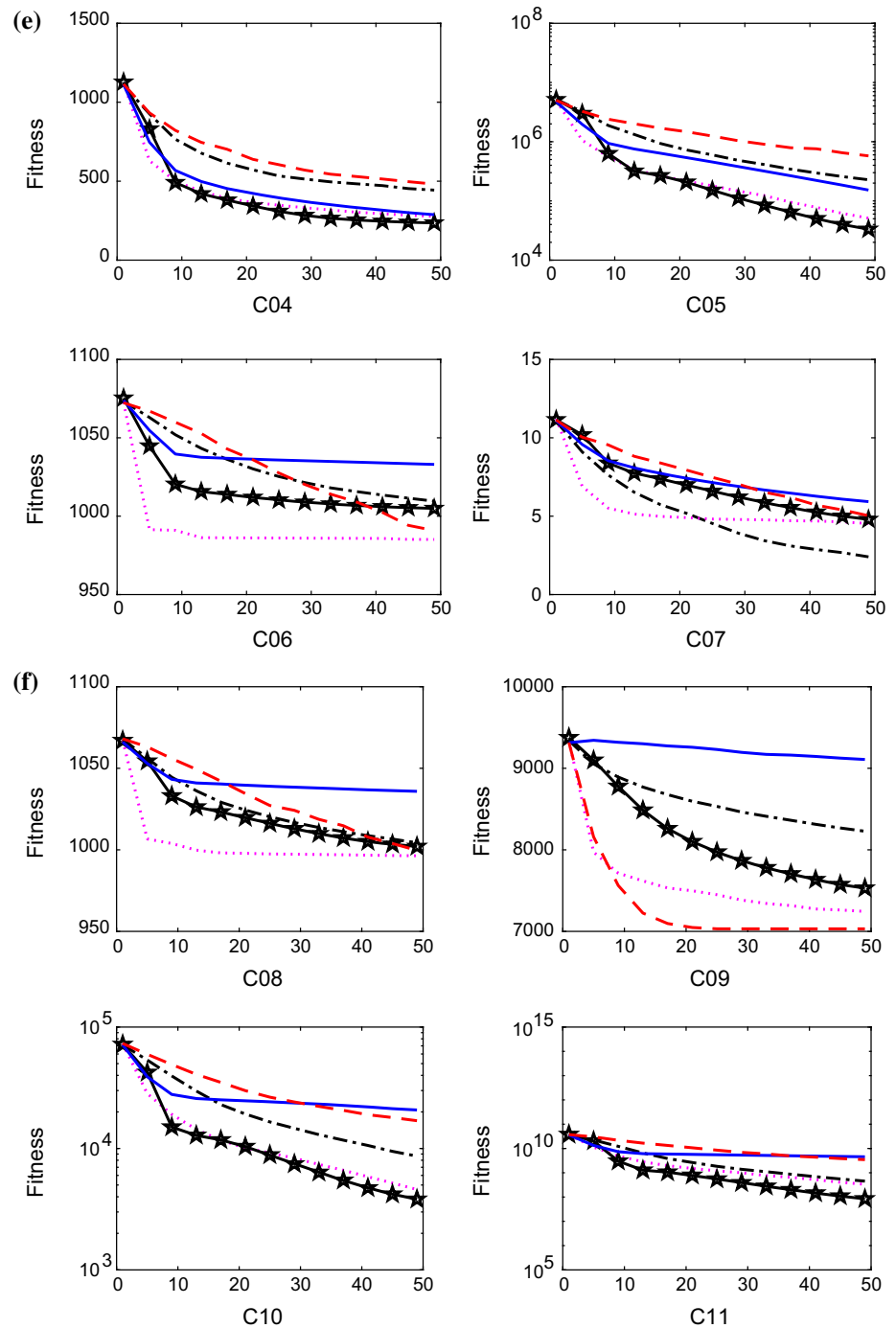
Fig. 5 continued



MBA, and VNBA) on thirty-four benchmark problems (F01–F13, and C01–C21) are given in Fig. 5. It can be observed from Fig. 5, that KHABC is capable of finding minimum on most benchmark functions. For other four algorithms, HSBBO is little inferior to KHABC, though HSBBO has the similar performance with KHABC on some benchmark functions.

All the experiments conducted in this section have proven the effectiveness and efficiency of the proposed KHABC algorithm. This also indicates KHABC is a promising, robust optimization strategy for unconstrained, constrained, and real world problems.

Fig. 5 continued

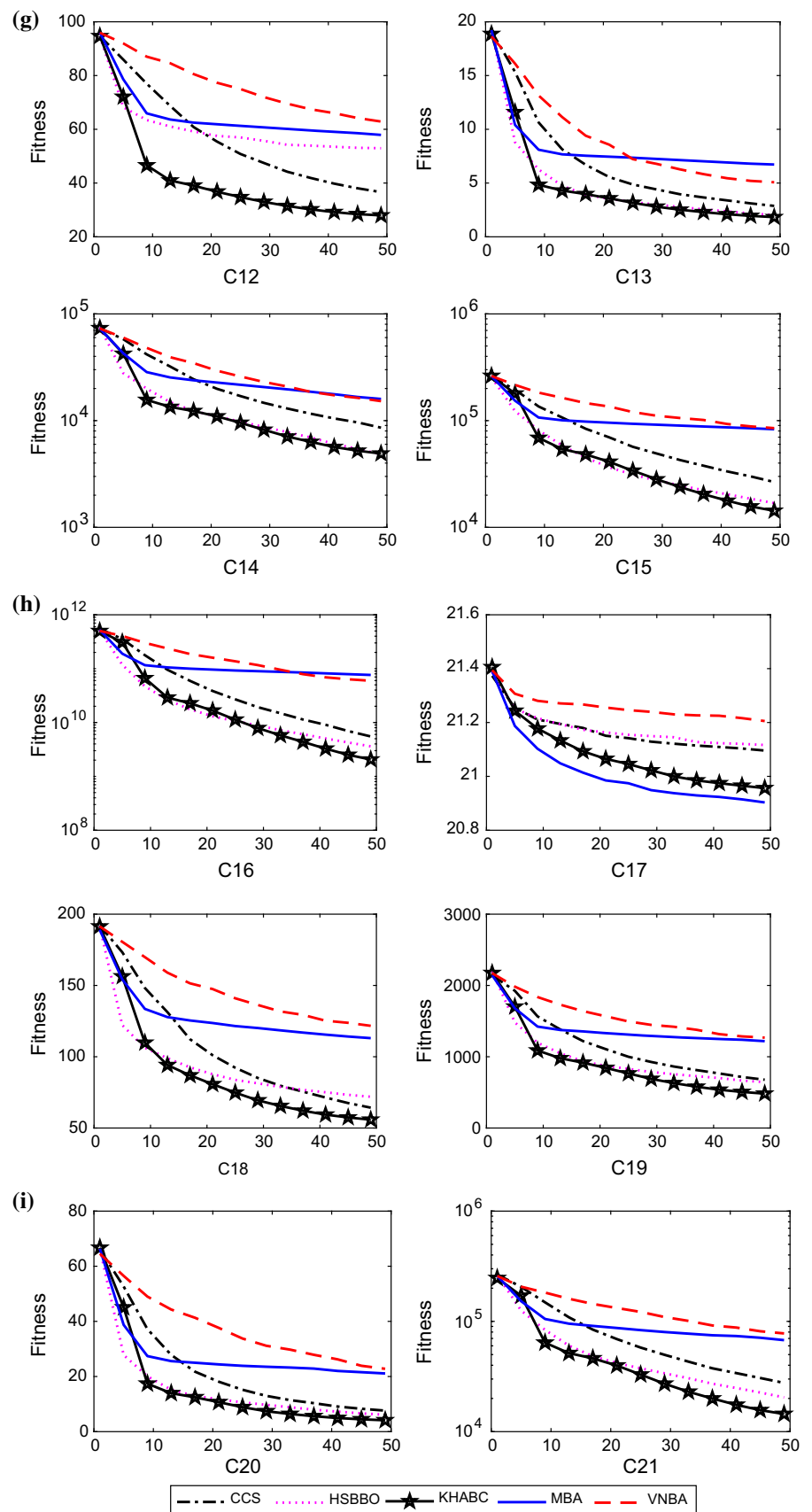


### 6 Conclusions

In the present study, a hybridization of the KH and ABC methods, namely KHABC, is proposed for the continuous and discrete optimization. The KHABC integrates the capabilities of the KH and the ABC in order to stop all krill from being attracted to local values. Moreover, a focused elitism scheme is applied to the method to further enhance its performance. The effectiveness of the proposed methodology

is tested using twenty-five high-dimensional test problems, as well as a discrete problem. The results clearly demonstrate the superiority of KHABC over KH, ABC, and other meta-heuristic algorithms. However, there are quite a few issues that merit further investigation such as analyzing the parameters used in the KHABC method. The future study can focus on solving a more ubiquitous set of different continuous optimization and discrete problems. The combination of other search strategies based on various

Fig. 5 continued





robust meta-heuristic techniques, such as the ACO and PSO, is a direction that is worth investigation. Finally, the study of CPU time used by the meta-heuristic approaches needs attention in order to make the proposed method more feasible for solving the practical engineering problems.

**Acknowledgements** This work was supported by the Natural Science Foundation of Jiangsu Province (No. BK20150239) and National Natural Science Foundation of China (No. 61503165).

## References

- Kundu S, Parhi DR (2016) Navigation of underwater robot based on dynamically adaptive harmony search algorithm. *Memet Comput* 8(2):125–146. doi:[10.1007/s12293-016-0179-0](https://doi.org/10.1007/s12293-016-0179-0)
- Zhang Y, Liu J, Zhou M, Jiang Z (2016) A multi-objective memetic algorithm based on decomposition for big optimization problems. *Memet Comput* 8(1):45–61. doi:[10.1007/s12293-015-0175-9](https://doi.org/10.1007/s12293-015-0175-9)
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359. doi:[10.1023/A:1008202821328](https://doi.org/10.1023/A:1008202821328)
- Beyer H, Schwefel H (2002) *Nat Comput*. Kluwer Academic Publishers, Dordrecht
- Goldberg DE (1989) *Genetic algorithms in search, optimization, and machine learning*, vol 412. Addison-Wesley, Boston
- Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Glob Optim* 39(3):459–471. doi:[10.1007/s10898-007-9149-x](https://doi.org/10.1007/s10898-007-9149-x)
- Wang G-G, Deb S, Gao X-Z, Coelho LdS (2016) A new metaheuristic optimization algorithm motivated by elephant herding behavior. *Int J of Bio Inspir Comput* 8(6):394–409. doi:[10.1504/IJBIC.2016.10002274](https://doi.org/10.1504/IJBIC.2016.10002274)
- Wang G-G (2016) Moth search algorithm: a bio-inspired meta-heuristic algorithm for global optimization problems. *Memet Comput*. doi:[10.1007/s12293-016-0212-3](https://doi.org/10.1007/s12293-016-0212-3)
- Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. *Simulation* 76(2):60–68. doi:[10.1177/003754970107600201](https://doi.org/10.1177/003754970107600201)
- Wang G-G, Deb S, Cui Z (2015) Monarch butterfly optimization. *Neural Comput Appl*. doi:[10.1007/s00521-015-1923-y](https://doi.org/10.1007/s00521-015-1923-y)
- Kennedy J, Eberhart R (1995) Particle swarm optimization. Paper presented at the proceeding of the IEEE international conference on neural networks, Perth, Australia, 27 November–1 December
- Le MN, Ong Y-S, Jin Y, Sendhoff B (2009) Lamarckian memetic algorithms: local optimum and connectivity structure analysis. *Memet Comput* 1(3):175–190. doi:[10.1007/s12293-009-0016-9](https://doi.org/10.1007/s12293-009-0016-9)
- Meuth R, Lim M-H, Ong Y-S, Wunsch DC (2009) A proposition on memes and meta-memes in computing for higher-order learning. *Memet Comput* 1(2):85–100. doi:[10.1007/s12293-009-0011-1](https://doi.org/10.1007/s12293-009-0011-1)
- Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. *Commun Nonlinear Sci Numer Simul* 17(12):4831–4845. doi:[10.1016/j.cnsns.2012.05.010](https://doi.org/10.1016/j.cnsns.2012.05.010)
- Črepinšek M, Liu S-H, Mernik M (2013) Exploration and exploitation in evolutionary algorithms. *ACM Comput Surv* 45(3):1–33. doi:[10.1145/2480741.2480752](https://doi.org/10.1145/2480741.2480752)
- Wang G-G, Guo L, Gandomi AH, Hao G-S, Wang H (2014) Chaotic krill herd algorithm. *Inf Sci* 274:17–34. doi:[10.1016/j.ins.2014.02.123](https://doi.org/10.1016/j.ins.2014.02.123)
- Wang G-G, Gandomi AH, Yang X-S, Alavi AH (2016) A new hybrid method based on krill herd and cuckoo search for global optimization tasks. *Int J of Bio Inspir Comput* 8(5):286–299. doi:[10.1504/IJBIC.2016.10000414](https://doi.org/10.1504/IJBIC.2016.10000414)
- Mukherjee A, Mukherjee V (2015) Solution of optimal power flow using chaotic krill herd algorithm. *Chaos Solitons Fractals* 78:10–21. doi:[10.1016/j.chaos.2015.06.020](https://doi.org/10.1016/j.chaos.2015.06.020)
- Wang G-G, Gandomi AH, Alavi AH (2014) Stud krill herd algorithm. *Neurocomputing* 128:363–370. doi:[10.1016/j.neucom.2013.08.031](https://doi.org/10.1016/j.neucom.2013.08.031)
- Bolaji ALa, Al-Betar MA, Awadallah MA, Khader AT, Abualigah LM (2016) A comprehensive review: krill herd algorithm (KH) and its applications. *Appl Soft Comput* 49:437–446. doi:[10.1016/j.asoc.2016.08.041](https://doi.org/10.1016/j.asoc.2016.08.041)
- Bolaji AL, Khader AT, Al-Betar MA, Awadallah MA (2014) University course timetabling using hybridized artificial bee colony with hill climbing optimizer. *J Comput Sci* 5(5):809–818. doi:[10.1016/j.jocs.2014.04.002](https://doi.org/10.1016/j.jocs.2014.04.002)
- Kiran SM, Gündüz M (2013) A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems. *Appl Soft Comput* 13(4):2188–2203. doi:[10.1016/j.asoc.2012.12.007](https://doi.org/10.1016/j.asoc.2012.12.007)
- Awadallah MA, Bolaji ALa, Al-Betar MA (2015) A hybrid artificial bee colony for a nurse rostering problem. *Appl Soft Comput* 35:726–739. doi:[10.1016/j.asoc.2015.07.004](https://doi.org/10.1016/j.asoc.2015.07.004)
- Bullinaria JA, AlYahya K (2014) Artificial bee colony training of neural networks: comparison with back-propagation. *Memet Comput* 6(3):171–182. doi:[10.1007/s12293-014-0137-7](https://doi.org/10.1007/s12293-014-0137-7)
- Li JQ, Pan QK, Duan PY (2016) An improved artificial bee colony algorithm for solving hybrid flexible flowshop with dynamic operation skipping. *IEEE Trans Cybern* 46(6):1311–1324. doi:[10.1109/TCYB.2015.2444383](https://doi.org/10.1109/TCYB.2015.2444383)
- Krüger TJ, Davidović T, Teodorović D, Šelmić M (2016) The bee colony optimization algorithm and its convergence. *Int J Bio Inspir Comput* 8(5):340–354. doi:[10.1504/IJBIC.2016.079573](https://doi.org/10.1504/IJBIC.2016.079573)
- Hussein WA, Sahran S, Sheikh Abdullah SNH (2017) The variants of the bees algorithm (BA): a survey. *Artif Intell Rev* 47(1):67–121. doi:[10.1007/s10462-016-9476-8](https://doi.org/10.1007/s10462-016-9476-8)
- Zhang Y, Wu L (2012) Artificial bee colony for two dimensional protein folding. *Adv Electr Eng Syst* 1(1):19–23
- Wang G, Guo L, Wang H, Duan H, Liu L, Li J (2014) Incorporating mutation scheme into krill herd algorithm for global numerical optimization. *Neural Comput Appl* 24(3–4):853–871. doi:[10.1007/s00521-012-1304-8](https://doi.org/10.1007/s00521-012-1304-8)
- Wang G-G, Deb S, Gandomi AH, Zhang Z, Alavi AH (2016) Chaotic cuckoo search. *Soft Comput* 20(9):3349–3362. doi:[10.1007/s00500-015-1726-1](https://doi.org/10.1007/s00500-015-1726-1)
- Wang G, Guo L, Duan H, Wang H, Liu L, Shao M (2013) Hybridizing harmony search with biogeography based optimization for global numerical optimization. *J Comput Theor Nanos* 10(10):2318–2328. doi:[10.1166/jctn.2013.3207](https://doi.org/10.1166/jctn.2013.3207)
- Li Z-Y, Yi J-H, Wang G-G (2015) A new swarm intelligence approach for clustering based on krill herd with elitism strategy. *Algorithms* 8(4):951–964. doi:[10.3390/a8040951](https://doi.org/10.3390/a8040951)
- Wang G-G, Chang B, Zhang Z (2015) A multi-swarm bat algorithm for global optimization. In: 2015 IEEE congress on evolutionary computation (CEC 2015), Sendai, Japan, May 25–28, 2015. IEEE, pp 480–485. doi:[10.1109/CEC.2015.7256928](https://doi.org/10.1109/CEC.2015.7256928)
- Wang G-G, Lu M, Zhao X-J (2016) An improved bat algorithm with variable neighborhood search for global optimization. Paper presented at the 2016 IEEE congress on evolutionary computation (IEEE CEC 2016), Vancouver, 25–29 July, 2016

35. Wang G-G, Deb S, Gandomi AH, Alavi AH (2016) Opposition-based krill herd algorithm with Cauchy mutation and position clamping. *Neurocomputing* 177:147–157. doi:[10.1016/j.neucom.2015.11.018](https://doi.org/10.1016/j.neucom.2015.11.018)
36. Das S, Suganthan P (2010) Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Jadavpur Univ., Nanyang Technol. Univ., Kolkata, India