Applied Mathematical Modelling 38 (2014) 2454–2462

Contents lists available at ScienceDirect

Applied Mathematical Modelling

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

An effective krill herd algorithm with migration operator in biogeography-based optimization

Gai-Ge Wang [a,*], Amir H. Gandomi [b], Amir H. Alavi [c]

[a] School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, Jiangsu 221116, China
[b] Department of Civil Engineering, The University of Akron, Akron, OH 44325, USA
[c] Department of Civil and Environmental Engineering, Engineering Building, Michigan State University, East Lansing, MI 48824, USA

ARTICLE INFO

Article history:
Received 18 December 2012
Received in revised form 17 July 2013
Accepted 8 October 2013
Available online 4 November 2013

Keywords:
Global optimization
Krill herd
Biogeography-based optimization
Krill migration operator

ABSTRACT

Krill herd (KH) is a novel search heuristic method. To improve its performance, a biogeography-based krill herd (BBKH) algorithm is presented for solving complex optimization tasks. The improvement involves introducing a new krill migration (KM) operator when the krill updating to deal with optimization problems more efficiently. The KM operator emphasizes the exploitation and lets the krill cluster around the best solutions at the later run phase of the search. The effects of these enhancements are tested by various well-defined benchmark functions. Based on the experimental results, this novel BBKH approach performs better than the basic KH and other optimization algorithms.

1. Introduction

Optimization is to select a vector for a function that makes the function reach maximum/minimum. Nowadays, modern intelligent algorithms based on randomness are utilized to solve optimization problems. Up to now, various techniques have been used to cope with these problems. Classification of these techniques can be carried out in many ways. However, in general, a simple way for classifying these techniques is to look at their nature, and they can be generally categorized as two groups: classical methods, and intelligent algorithms. The former follow a rigorous procedure, and for the same input values, they will follow the same path and eventually generate the same solutions. Contrarily, intelligent algorithms are based on randomness, and the final solutions will be different each time you run a program regardless of initial value. However, in most cases, these two types of methods can reach the same values within a given accuracy. Recently, meta-heuristic methods inspired by nature perform efficiently in solving modern complicated global optimization problems. These meta-heuristic methods have been designed according to the principle of the nature for optimization problems, like o reliability problem [1,2], knapsack problem [3], permutation flow shop scheduling [4], system identification [5], airline boarding problem [6], UCAV path planning [7], education [8], and engineering optimization [9,10]. In most cases, these kinds of meta-heuristic methods can always find optimal or near-optimal solutions coming from a population of solutions not individual. After idealizing evolution as an optimization method called genetic algorithms (GAs) [11,12], various techniques have designed for optimization, such as harmony search (HS) [13–15], cuckoo search (CS) [16–19], evolutionary strategy (ES) [20,21], differential evolution (DE) [22–25], imperialist competitive algorithm (ICA) [26], ant colony optimization (ACO) [27], bat algorithm

* Corresponding author. Tel.: +86 13852006092.
E-mail addresses: gaigewang@163.com (G.-G. Wang), a.h.gandomi@gmail.com, ag72@uakron.edu (A.H. Gandomi), ah_alavi@hotmail.com, alavi@msu.edu (A.H. Alavi).
URLs: http://gozips.uakron.edu/~ag72/ (A.H. Gandomi), http://www.egr.msu.edu/~alavi/ (A.H. Alavi).

0307-904X/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.apm.2013.10.052

(BA) [28], animal migration optimization (AMO) [29], particle swarm optimization (PSO) [30–33], artificial bee colony (ABC) [34], probability-based incremental learning (PBIL) [35], genetic programming (GP) [36], big bang-big crunch algorithm [37–40], biogeography-based optimization (BBO) [41], charged system search (CSS) [42,43], and the KH algorithm [44].

By idealizing the swarm behavior of krill, KH is proposed as a swarm intelligence approach for optimization tasks [44]. For the krill movement, the objective function used in KH is determined by the least distances from food and the highest herd density. The position of the krill consists of three main components. Comparing with other algorithms, one of the advantages of the KH algorithm is that it requires few control variables to regulate.

In general, KH can explore the search space effectively and efficiently but sometimes it may not escape some local optima. Thus, it fails to implement global search fully in a given decision space. For KH, the search in KH is primarily based on random walks; hence, it cannot always converge to a satisfactory solution.

On the other hand, firstly presented by Simon [41], BBO is a novel stochastic global optimizer based on the biogeography theory. BBO has satisfactory global optimization ability, insensitivity for parameters, and simplicity. Therefore, it has attracted much attention from academic research and application.

In the present work, an effective biogeography-based KH (BBKH) method is proposed for the purpose of accelerating convergence speed. In BBKH, firstly, a standard KH is used to cut down the search apace and choose a promising candidate solution set in order to make the search more efficiently. Subsequently, krill migration (KM) operator is combined with the method. It can exploit the search region carefully to find the best solutions for optimization problems. The proposed method is verified on 14 functions. Experiments show that the BBKH method performs better than basic KH and other twelve methods.

The rest of the paper is structured as follows. Reviews on BBO and KH are discussed in Section 2 and Section 3, respectively. The BBKH method is presented in Section 3. Details on BBKH are presented in Section 4. Our method is tested through on functions in Section 5. Finally, Section 6 provides the conclusion and points out the path of the future work.

2. BBO

Firstly proposed by Dan Simon, BBO [41] is a novel meta-heuristic evolution algorithm for optimization problems. It is based on the idealization of the migration of species between islands when searching for more satisfactory places. Each candidate solution is called a "habitat" responding to a HSI and denoted by a d-dimension vector. A population of habitat is initially generated at random in the search space. The quality of habitat is mainly affected by HSI. Low HIS can absorb some useful features from the high HSI. Habitat H in BBO is a vector of NP SIVs. Later migration and mutation operators are implemented to get the final best solution.

The migration operator used in BBO is likely to the ES [21], where many parents cooperate to generate an offspring. In essence, migration is a stochastic operator that updates X_i. The updated probability X_i is related to its immigration rate λ_i and the emigration rate μ_j.

Because mutation operator is not used in our present work, here it is not described in detail. More information about BBO can be referred as in [41].

3. KH algorithm

By idealizing by the swarm behavior of krill, KH [44] is a meta-heuristic optimization approach for solving optimization problems. In KH, the position is mainly affected by three actions [45,46]:

i. movement affected by other krill;
ii. foraging action;
iii. physical diffusion.

In KH, the Lagrangian model [44] is used within predefined search space as Eq. (1).

$$\frac{dX_i}{dt} = N_i + F_i + D_i \tag{1}$$

where N_i is the motion produced by other krill individuals; F_i is the foraging motion, and D_i is the random diffusion of the ith krill individual.

In the first one, its direction, α_i, is decided by the following parts: target effect, local effect, and a repulsive effect. In sum, its definition can be provided below:

$$N_i^{new} = N^{\max}\alpha_i + \omega_n N_i^{old} \tag{2}$$

and $N^{\max}$, ω_n and N_i^{old} are the maximum speed, the inertia weight, the last motion, respectively.

The second one can be approximately calculated by the two components: the food location and its previous experience. For the ith krill, it can be idealized below:

$$F_i = V_f \beta_i + \omega_f F_i^{old} \tag{3}$$

where

$$\beta_i = \beta_i^{food} + \beta_i^{best} \tag{4}$$

and V_f is the foraging speed, ω_f is the inertia weight, F_i^{old} is the last one.

The third part is essentially a random process. It is computed based on a maximum diffusion speed and a random directional vector. Its expression is below:

$$D_i = D^{\max} \delta \tag{5}$$

where $D^{\max}$ is the maximum diffusion speed, and δ is the random directional vector and its arrays are random numbers.

Herein, the position in KH from t to $t + \Delta t$ is formulated as follows:

$$X_i(t + \Delta t) = X_i(t) + \Delta t \frac{dX_i}{dt} \tag{6}$$

More detailed information about the KH method can be referred as in [44].

4. BBKH

As the search used in KH is mainly based on random walks, it cannot converge to the satisfactory function value all the time. To combat with this disadvantage, Gandomi and Alavi [44] have added genetic reproduction mechanisms to the algorithm. It was shown that these mechanisms notably improve the performance of basic KH [44]. In addition, KH performs well on unimodal functions and several multimodal functions. However, sometimes KH's performance on complex multimodal functions is disappointing. In general, KH can implement exploration quickly and locate the promising region. While, KH has a poor exploitation ability. Therefore, another optimization strategy that can perform exploitation well is necessary to search locally to make the method converge the best solution.

Herein, in order to enhance the exploitation ability of KH, an improved habitat migration operator originally used in BBO performing local search, called krill migration (KM) operator, is combined with KH to form an effective biogeography-based krill herd (BBKH) approach. In BBKH, the KM operator is used to regulate the new solution generated by KH for each krill; while randomness is used in KH. This improved local search technique can increase population diversity for the purpose of avoiding premature convergence and make the krill search a small promising region carefully at the later stage of the method. The step of the KM operator used in BBKH is described in **Algorithm 1**.

In BBKH, to begin with, due to its fast convergence, KH is applied to make the krill cluster to a limited area. After that, the KM operator with good exploitation is used to search locally to choose better krill. In essence, the KH in BBKH centralizes on the exploration at early stage of the optimization; while later, the KM operator emphasizes the exploitation and makes most krill cluster around the best solution at the later of the optimization. In this way, BBKH can fully explore the space with KH and exploit the useful information by KM operator. Therefore, this method can overcome the KH's poor exploitation.

Algorithm 1 Krill migration (KM) operator.

Begin
  Select krill i (its position X_i) with probability based on λ_i
  if rand(0,1)< λ_i **then**
    for j=1 to d (all elements) **do**
      Select X_j with probability based on μ_j
      if rand(0,1)< μ_j **then**
        Randomly select an element σ from X_j
        Replace a random element in X_i with σ
      end **if**
    end for j
  end if
End.

In addition, as with other algorithms, we add some kind of elitism for the purpose of keeping the optimal krill in the population all the time. It can forbid the optimal krill from being ruined by three motions and the KM operator. We must point out that, in BBKH, an elitism strategy is used to save the property of the krill with optimal fitness, and thus even if KH and/or the KM operator corrupt its previous good krill, we have saved it and can easily return it to its former condition.

By combining KM operator and focused elitism into the KH algorithm, the BBKH method has been developed as **Algorithm 2**. λ_i and μ_i are functions of *NP*.

Algorithm 2 Biogeography-based KH mehtod.

Begin
  Step 1: Initialization. Set the generation counter t = 1; initialize the population P of NP krill randomly; set V_f, D^{max}, N^{max}, S_{max} and p_{mod}.
  Step 2: Fitness evaluation. Evaluate each krill.
  Step 3: While t < MaxGeneration **do**
    Sort the krill from best to worst.
    Store the best krill.
    for i=1:NP (all krill) **do**
      Perform the three motion calculation.
      Update the krill position by Eq. (6).
      Fine-tune X_{i+1} by performing KM operator in **Algorithm 1**.
      Evaluate each krill by X_{i+1}.
    end for i
    Replace the worst krill with the best krill.
    Sort the krill and find the current best.
    t = t+1;
  Step 4: end while
  Step 5: Output the best solutions.
End.

Table 1
Benchmark functions.

No.	Name	Definition
F01	Ackley	$f(\vec{x}) = 20 + e - 20 \cdot e^{-0.2 \cdot \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)}$
F02	Fletcher–Powell	$f(\vec{x}) = \sum_{i=1}^{n}(A_i - B_i)^2, \quad A_i = \sum_{j=1}^{n}(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j)$ $B_i = \sum_{j=1}^{n}(a_{ij}\sin x_j + b_{ij}\cos x_j)$
F03	Griewank	$f(\vec{x}) = \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F04	Penalty #1	$f(\vec{x}) = \frac{\pi}{30}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4), \quad y_i = 1 + 0.25(x_i + 1)$
F05	Penalty #2	$f(\vec{x}) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2\left[1 + \sin^2(3\pi x_{i+1})\right] + (x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$
F06	Quartic with noise	$f(\vec{x}) = \sum_{i=1}^{n}(i \cdot x_i^4 + U(0,1))$
F07	Rastrigin	$f(\vec{x}) = 10 \cdot n + \sum_{i=1}^{n}(x_i^2 - 10 \cdot \cos(2\pi x_i))$
F08	Rosenbrock	$f(\vec{x}) = \sum_{i=1}^{n-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$
F09	Schwefel 2.26	$f(\vec{x}) = 418.9829 \times D - \sum_{i=1}^{D} x_i \sin\left(\lvert x_i\rvert^{1/2}\right)$
F10	Schwefel 1.2	$f(\vec{x}) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$
F11	Schwefel 2.22	$f(\vec{x}) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$
F12	Schwefel 2.21	$f(\vec{x}) = \max_i\{\lvert x_i\rvert, 1 \le i \le n\}$
F13	Sphere	$f(\vec{x}) = \sum_{i=1}^{n} x_i^2$
F14	Step	$f(\vec{x}) = 6 \cdot n + \sum_{i=1}^{n}\lfloor x_i\rfloor$

*In benchmark function F02, the matrix elements $\mathbf{a}_{n\times n}, \mathbf{b}_{n\times n} \in (-100, 100), \alpha_{n\times 1} \in (-\pi, \pi)$ are draw from uniform distribution.
*In benchmark functions F04 and F05, the definition of the function $u(x_i,a,k,m)$ is as follows:

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$$

Table 2
Mean normalized optimization results.

	ABC	ACO	BA	BBO	BKH	CS	DE	ES	GA	HS	KH	PBIL	PSO	SGA
F01	4.09	4.66	5.95	2.61	**1.00**	5.26	3.70	5.77	5.04	5.89	1.34	5.99	4.97	2.68
F02	2.85	10.13	14.26	1.14	**1.00**	6.95	3.88	10.45	3.85	9.30	4.04	9.14	8.70	1.26
F03	32.41	9.58	172.37	7.15	**1.00**	66.85	16.01	79.27	29.78	154.40	4.58	174.47	61.81	6.47
F04	4.2E5	2.4E7	3.0E7	7.4E3	**1.00**	1.4E6	8.0E4	1.0E7	1.2E5	1.8E7	4.4E3	2.4E7	1.6E6	13.94
F05	2.1E3	3.4E4	6.2E4	80.58	**1.00**	6.5E3	626.79	2.9E4	1.1E3	4.7E4	66.30	5.7E4	6.5E3	7.65
F06	166.68	165.62	2.5E3	12.25	**1.00**	422.47	67.03	2.1E3	144.99	2.0E3	19.61	2.5E3	481.93	5.56
F07	3.79	7.03	10.59	1.59	**1.00**	7.97	6.19	9.85	6.70	9.34	4.00	10.05	7.40	2.16
F08	9.37	52.16	53.16	3.21	**1.00**	14.47	7.55	68.54	13.87	48.43	3.50	54.13	16.28	3.03
F09	3.70	2.43	8.49	1.22	**1.00**	6.32	4.76	5.87	2.09	7.11	4.48	7.49	7.15	1.40
F10	7.12	6.26	15.54	3.83	**1.00**	4.04	8.87	9.88	7.21	9.16	4.39	9.89	6.91	5.76
F11	6.19	17.03	1.3E4	2.47	**1.00**	16.61	7.33	26.41	12.90	21.13	9.65	21.18	17.22	3.47
F12	6.64	4.10	7.01	4.50	**1.00**	4.87	5.35	6.61	5.66	6.79	1.07	6.97	5.64	3.92
F13	102.44	262.15	534.34	19.78	**1.00**	207.50	50.34	545.84	180.44	497.72	14.02	551.73	195.57	22.09
F14	27.54	12.34	154.52	5.43	**1.00**	53.29	13.87	94.19	26.83	137.22	3.67	151.01	54.68	4.48
Time	2.38	3.17	1.07	1.56	3.10	2.01	1.93	2.03	2.37	2.77	4.66	**1.00**	2.39	2.33
Total	0	0	0	0	**14**	0	0	0	0	0	0	0	0	0

Table 3
Best normalized optimization results.

	ABC	ACO	BA	BBO	BKH	CS	DE	ES	GA	HS	KH	PBIL	PSO	SGA
F01	76.18	105.55	145.83	52.32	**1.00**	126.19	86.37	149.27	90.29	152.38	19.25	159.44	128.76	52.12
F02	3.54	26.37	35.94	2.08	**1.00**	18.82	10.43	28.39	7.53	19.50	9.16	25.50	25.47	1.99
F03	8.24	4.64	103.22	2.84	**1.00**	45.85	10.34	60.04	10.02	136.85	2.53	157.75	49.08	3.05
F04	20.16	**1.00**	2.0E7	19.57	3.98	1.1E6	144.30	1.6E7	38.09	9.2E7	95.55	1.2E8	4.6E4	8.41
F05	4.0E20	**1.00**	2.3E23	6.6E17	4.7E16	8.2E21	2.9E20	9.6E22	5.6E19	2.4E23	8.5E19	3.4E23	9.1E21	6.0E16
F06	372.90	2.4E3	3.7E4	59.18	**1.00**	7.5E3	1.2E3	5.6E4	557.92	7.5E4	326.92	8.8E4	1.2E4	26.38
F07	6.33	15.03	21.04	2.27	**1.00**	16.11	12.42	21.60	10.49	19.70	6.64	19.74	13.93	2.46
F08	7.19	46.18	19.85	2.18	**1.00**	11.91	5.54	57.05	8.51	48.56	3.45	34.63	13.59	1.97
F09	13.23	7.78	32.12	2.99	**1.00**	29.39	22.27	26.23	3.18	34.02	17.88	34.82	28.84	2.27
F10	11.31	10.20	21.79	4.33	**1.00**	6.34	13.79	20.89	9.82	18.34	7.28	23.52	12.19	10.40
F11	31.34	97.85	162.89	10.30	**1.00**	85.95	49.06	147.30	50.53	145.28	36.37	148.70	83.30	18.35
F12	20.92	10.95	20.75	10.41	**1.00**	14.42	14.71	21.70	8.75	22.89	1.60	23.76	17.08	7.71
F13	1.1E3	4.0E3	7.0E3	181.12	**1.00**	3.1E3	713.46	9.4E3	2.2E3	1.2E4	209.99	1.3E4	3.5E3	273.12
F14	83.79	29.84	593.26	15.89	**1.00**	166.16	62.16	506.74	56.84	597.05	11.11	591.53	295.32	7.58
Total	0	2	0	0	12	0	0	0	0	0	0	0	0	0

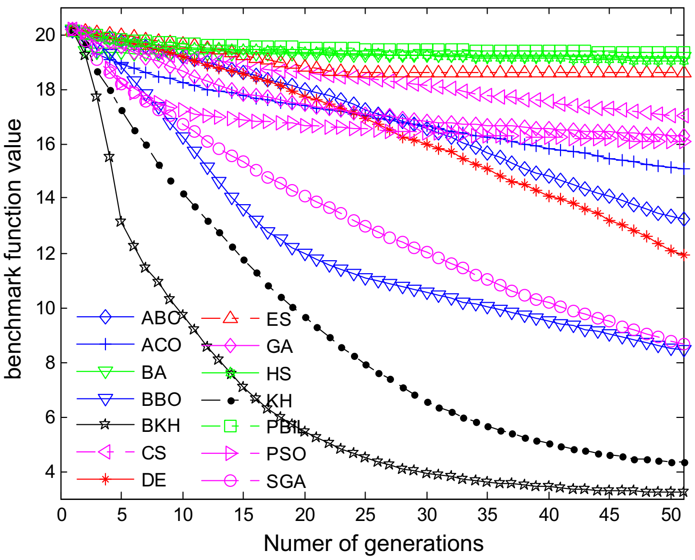

Fig. 1. Performance comparison for the F01 Ackley function.

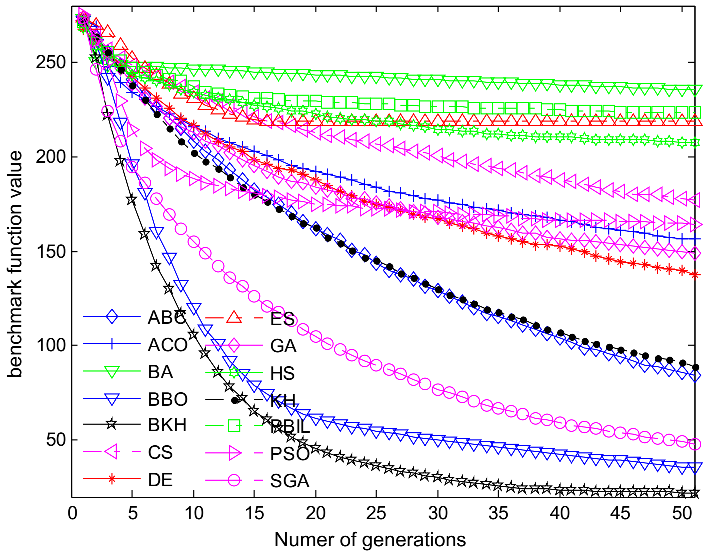

Fig. 2. Performance comparison for the F07 Rastrigin function.

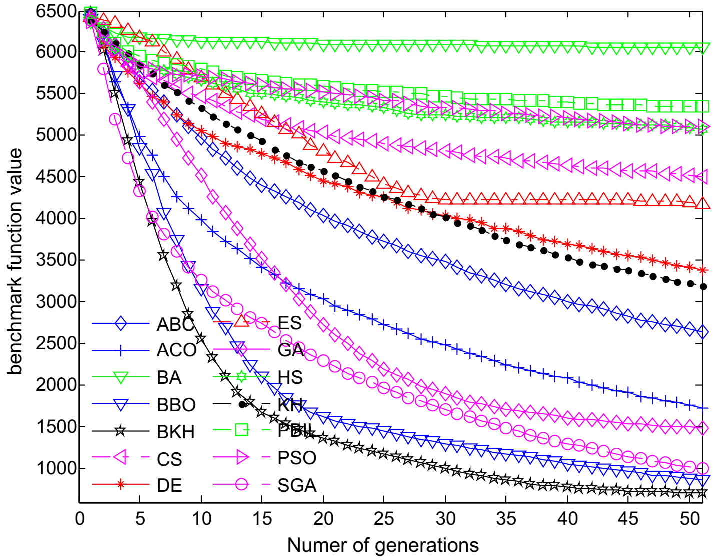

Fig. 3. Performance comparison for the F09 Schwefel 2.26 function.

5. Experiments

BBKH is evaluated through a variety of simulations in functions as shown in Table 1. Their properties can be found in Table 2. More information about these functions can be referred as in [41,47,48].

For the purpose of proving the superiority of BBKH, its performance is compared on optimization problems with thirteen methods, which are ABC [34], ACO [27,49], BA [28], BBO [41], CS [17,50], DE [22], ES [21], GA [11], HS [13], KH [44], PBIL [35], PSO [30,51] and SGA [52]. In addition, note that, in [44], Gandomi and Alavi concluded that, comparing all the other methods, the KH II performed the best. Herein, KH II is considered as basic KH algorithm.

Here, we will use the following parameters for KH and BBKH that are $V_f = 0.02$, $D^{max} = 0.005$, $N^{max} = 0.01$, habitat modification probability = 1 (only for BBKH). For others, we set the same parameters as [41,45,46].

We did 100 implementations to obtain representative performances. The results are recorded in Table 2 and Table 3. Note that, different scales are used to normalize the values in the tables, so they cannot be compared each other. Moreover, the detailed normalization process can be found in [53]. Here, the dimension of each function is 20.

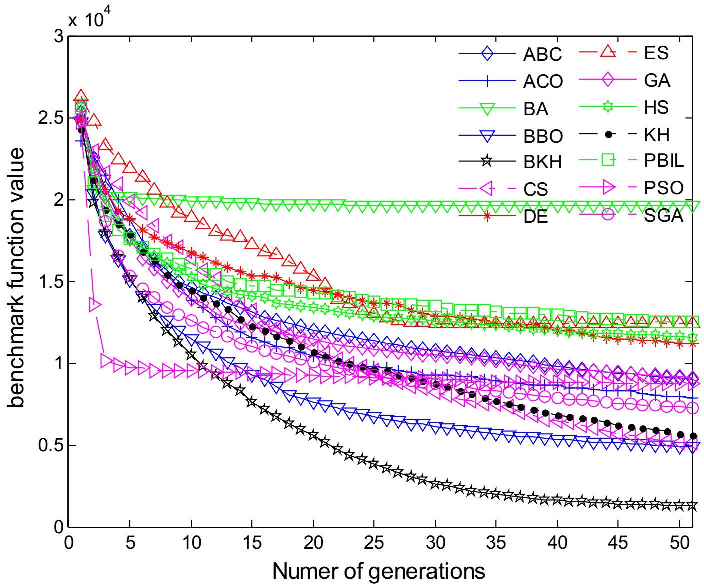

Fig. 4. Performance comparison for the F10 Schwefel 1.2 function.

From Table 2, on average, BBKH can find the most satisfactory function value on all the functions (F01-F14). Table 3 can find the best solutions on twelve benchmarks which are F01-F03, F06-F14. ACO is inferior to BBKH, and can find the optimal solution on the benchmarks F04 and F05.

Moreover, to future prove the superiority of the BBKH, the process of optimization for fourteen methods is provided in Figs. 1–5.

Fig. 1 illustrates that BBKH is always superior to all others from start to finish. Though slow, KH eventually converges to the function value close to BBKH. For others, the figure illustrates that the performance of BBO has little difference with SGA. Table 3 and Fig. 1 show that BBO performs slightly better than SGA, while others cannot search for the global minimum. Here, all the methods start the optimization process from almost the same value, however BBKH outperforms them soon.

For this case, BBKH, BBO and SGA perform the best and have ranks of 1, 2, and 3, respectively. Further looking at the Fig. 2, BBKH has the fastest convergence towards the global minimum and significantly overtakes all others. In addition, other methods do not find the satisfactory function value under given condition.

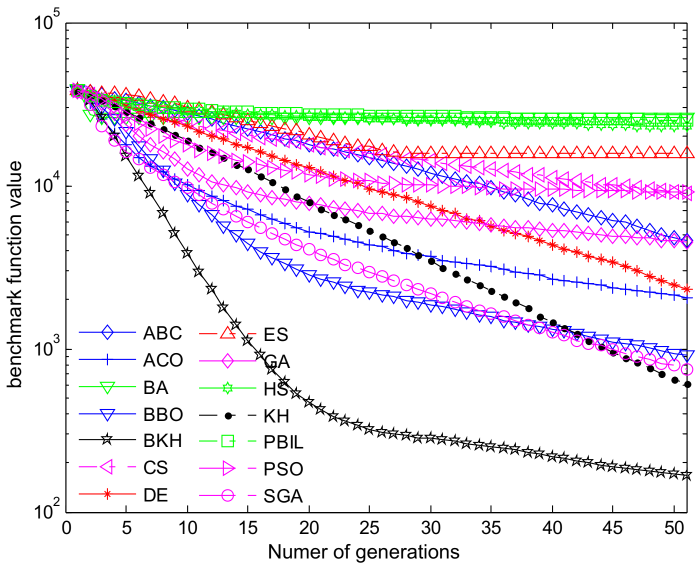

Fig. 5. Performance comparison for the F14 Step function.

From Fig. 3, BBKH, BBO, SGA and GA perform the best and have ranks of 1, 2, 3, and 4, respectively.

Fig. 4 shows that BBKH performs far better than others in the optimization process. For other methods, similarly, the performance of BBO, CS, and KH have little difference each other. Looking carefully at Table 3 and Fig. 4, BBO, KH and CS have ranks of 2, 3, and 4, respectively. In addition, PSO converges faster initially than BBKH; however, it is overtaken by BBKH method after 12 generations.

From Fig. 5, BBKH performs the best in this unimodal function. In addition, KH, SGA and BBO perform very well and rank 2, 3, and 4, respectively.

Tables 2 and 3 and Figs. 1–5 show that our hybrid BBKH algorithm significantly outperforms the other thirteen methods. In most cases, BBO, KH and SGA are only inferior to BBKH, and perform the second best among fourteen approaches. ABC, ACO, CS and GA perform the third best only inferior to BBKH and KH.

6. Conclusions

In the present work, the KM operator is combined with the basic KH to propose an improved BBKH approach for optimization problems. In BBKH, the critical operator is the krill migration operator, which roots in migration operator in BBO. In BBKH, KH is applied to search globally and make the krill cluster into the promising area. Herein, KM operator performs local search to find the final best solution. In principle, KH can make full use of the population information from the three motions and performs very well on most multimodal problems. When solving complicated multimodal function, KH may not succeed in proceeding to better solutions [44]. Then, KM operator is automatically launched to restart the process. The experiments conducted here show that the BBKH method performed the best and most effectively for the global numerical optimization problems. However, similar to other intelligent methods, BBKH has a clear disadvantage that it is related to the necessity of adjusting parameters when solving different problems.

In optimization, many issues deserve further scrutiny. Three issues may be considered as the focus of the future research. Firstly, BBKH can be used to deal with engineering problems. On the other hand, some self-adaptive mechanisms may be added to make BBKH adaptively solve different problems. Thirdly, some new optimization techniques should be developed based on the specific engineering problem.

References

[1] D. Zou, L. Gao, J. Wu, et al., A novel global harmony search algorithm for reliability problems, Comput. Ind. Eng. 58 (2) (2010) 307–316.
[2] P. Wu, L. Gao, D. Zou, et al., An improved particle swarm optimization algorithm for reliability problems, ISA Trans. 50 (1) (2011) 71–81.
[3] D. Zou, L. Gao, S. Li, et al., Solving 0–1 knapsack problem by a novel global harmony search algorithm, Appl. Soft Comput. 11 (2) (2011) 1556–1564.
[4] X. Li, M. Yin, An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure, Adv. Eng. Softw. 55 (2013) 10–31.
[5] A. Alfi, H. Modares, System identification and control using adaptive particle swarm optimization, Appl. Math. Model. 35 (3) (2011) 1210–1221.
[6] M. Soolaki, I. Mahdavi, N. Mahdavi-Amiri, et al., A new linear programming approach and genetic algorithm for solving airline boarding problem, Appl. Math. Model. 36 (9) (2012) 4060–4072.
[7] G. Wang, L. Guo, H. Duan, et al., Path planning for UCAV using bat algorithm with mutation, Sci. World J. 2012 (2012) 1–15.
[8] H. Duan, W. Zhao, G. Wang, et al., Test-sheet composition using analytic hierarchy process and hybrid metaheuristic algorithm TS/BBO, Math. Probl. Eng. 2012 (2012) 1–22.
[9] X.S. Yang, A.H. Gandomi, S. Talatahari, et al., Metaheuristics in Water, Geotechnical and Transport Engineering, Elsevier, Waltham, MA, 2013.
[10] A.H. Gandomi, X.S. Yang, S. Talatahari, et al., Metaheuristic Applications in Structures and Infrastructures, Elsevier, Waltham, MA, 2013.
[11] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine learning, Addison-Wesley, New York, 1998.
[12] H. Nazif, L.S. Lee, Optimised crossover genetic algorithm for capacitated vehicle routing problem, Appl. Math. Model. 36 (5) (2012) 2110–2117.
[13] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60–68.
[14] S. Gholizadeh, A. Barzegar, Shape optimization of structures for frequency constraints by sequential harmony search algorithm, Eng. Optim. 45 (6) (2013) 627–646.
[15] G. Wang, L. Guo, A novel hybrid bat algorithm with harmony search for global numerical optimization, J. Appl. Math. 2013 (2013) 1–21.
[16] A.H. Gandomi, X.-S. Yang, A.H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (1) (2013) 17–35.
[17] G. Wang, L. Guo, H. Duan, et al., A hybrid meta-heuristic DE/CS algorithm for UCAV three-dimension path planning, Sci. World J. 2012 (2012) 1–11.
[18] X.S. Yang, S. Deb, Cuckoo search via Lévy flights, in: Proceeding of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), IEEE Publications, Coimbatore, India, 2009, pp. 210–214.
[19] A.H. Gandomi, S. Talatahari, X.S. Yang, et al., Design optimization of truss structures using cuckoo search algorithm, Struct. Des. Tall Spec. 22 (17) (2013) 1330–1349.
[20] T. Back, Evolutionary algorithms in theory and practice, Oxford University Press, Oxford, 1996.
[21] H. Beyer, The theory of evolution strategies, Springer, New York, 2001.
[22] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[23] A.H. Gandomi, X.-S. Yang, S. Talatahari, et al., Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization, Comput. Math. Appl. 63 (1) (2012) 191–200.
[24] X. Li, M. Yin, Application of differential evolution algorithm on self-potential data, PLoS ONE 7 (12) (2012) e51199.
[25] D. Zou, J. Wu, L. Gao, et al., A Modified Differential Evolution Algorithm for Unconstrained Optimization Problems, Neurocomputing 120 (2013) 469–481.
[26] S. Talatahari, B. Farahmand Azar, R. Sheikholeslami, et al., Imperialist competitive algorithm combined with chaos for global optimization, Commun. Nonlinear Sci. Numer. Simul. 17 (3) (2012) 1312–1319.
[27] M. Dorigo, T. Stutzle, Ant Colony Optimization, MIT Press, Cambridge, 2004.
[28] X.S. Yang, A.H. Gandomi, Bat algorithm: a novel approach for global engineering optimization, Eng. Comput. 29 (5) (2012) 464–483.
[29] X. Li, J. Zhang, M. Yin, Animal migration optimization: an optimization algorithm inspired by animal migration behavior, Neural Comput. Appl., In press.

[30] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceeding of the IEEE International Conference on Neural Networks, vol. 4, IEEE, 1995, pp. 1942–1948.
[31] S. Talatahari, M. Kheirollahi, C. Farahmandpour, et al., A multi-stage particle swarm for optimum design of truss structures, Neural Comput. Appl. 23 (5) (2013) 1297–1309.
[32] M. Sadeghpour, H. Salarieh, A. Alasty, Minimum entropy control of chaos via online particle swarm optimization method, Appl. Math. Model. 36 (8) (2012) 3931–3940.
[33] Y. Zhang, D. Huang, M. Ji, et al., Image segmentation using PSO and PCM with Mahalanobis distance, Expert Syst. Appl. 38 (7) (2011) 9036–9040.
[34] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471.
[35] B. Shumeet, Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning, Carnegie Mellon University, Pittsburgh, PA, 1994.
[36] A.H. Gandomi, A.H. Alavi, Multi-stage genetic programming: a new strategy to nonlinear system modeling, Inf. Sci. 181 (23) (2011) 5227–5239.
[37] O.K. Erol, I. Eksin, A new optimization method: Big Bang-Big Crunch, Adv. Eng. Softw. 37 (2) (2006) 106–111.
[38] A. Kaveh, S. Talatahari, Size optimization of space trusses using Big Bang-Big Crunch algorithm, Comput. Struct. 87 (17–18) (2009) 1129–1140.
[39] A. Kaveh, S. Talatahari, Optimal design of Schwedler and ribbed domes via hybrid Big Bang-Big Crunch algorithm, J. Constr. Steel Res. 66 (3) (2010) 412–419.
[40] A. Kaveh, S. Talatahari, A discrete big bang-big crunch algorithm for optimal design of skeletal structures, Asian J. Civ. Eng. 11 (1) (2010) 103–122.
[41] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713.
[42] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta Mech. 213 (3–4) (2010) 267–289.
[43] S. Talatahari, R. Sheikholeslami, M. Shadfaran, et al., Optimum design of gravity retaining walls using charged system search algorithm, Math. Probl. Eng. 2012 (2012) 1–10.
[44] A.H. Gandomi, A.H. Alavi, Krill herd: a new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simulat. 17 (12) (2012) 4831–4845.
[45] G. Wang, L. Guo, H. Wang, et al., Incorporating mutation scheme into krill herd algorithm for global numerical optimization, Neural Comput. Appl., In press.
[46] G. Wang, L. Guo, A.H. Gandomi, et al., Lévy-flight krill herd algorithm, Math. Probl. Eng. 2013 (2013) 1–14.
[47] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (2) (1999) 82–102.
[48] X.-S. Yang, Z. Cui, R. Xiao, et al., Swarm Intelligence and Bio-Inspired Computation, Elsevier, Waltham, MA, 2013.
[49] M. Niksirat, M. Ghatee, S. Mehdi Hashemi, Multimodal K-shortest viable path problem in Tehran public transportation network and its solution applying ant colony and simulated annealing algorithms, Appl. Math. Model. 36 (11) (2012) 5709–5726.
[50] X.S. Yang, S. Deb, Engineering optimisation by cuckoo search, Int. J. Math. Model. Numer. Optim. 1 (4) (2010) 330–343.
[51] A.H. Gandomi, G.J. Yun, X.-S. Yang, et al., Chaos-enhanced accelerated particle swarm optimization, Commun. Nonlinear Sci. Numer. Simulat. 18 (2) (2013) 327–340.
[52] W. Khatib, P. Fleming, The stud GA: a mini revolution?, in: A. Eiben, T. Back, M. Schoenauer, H. Schwefel (Eds.) Proc. of the 5th International Conference on Parallel Problem Solving from Nature, Springer-Verlag, New York, USA, 1998, pp. 683–691.
[53] G.-G. Wang, A.H. Gandomi, A.H. Alavi, G.-S. Hao, Hybrid krill herd algorithm with differential evolution for global numerical optimization, Neural Comput. Appl., in press.