Neural Comput & Applic
DOI 10.1007/s00521-016-2334-4

ORIGINAL ARTICLE

Human behavior-based optimization: a novel metaheuristic approach to solve complex optimization problems

Seyed-Alireza Ahmadi[1]

Received: 10 August 2015 / Accepted: 25 April 2016

Abstract Optimization techniques, specially evolutionary algorithms, have been widely used for solving various scientific and engineering optimization problems because of their flexibility and simplicity. In this paper, a novel metaheuristic optimization method, namely human behavior-based optimization (HBBO), is presented. Despite many of the optimization algorithms that use nature as the principal source of inspiration, HBBO uses the human behavior as the main source of inspiration. In this paper, first some human behaviors that are needed to understand the algorithm are discussed and after that it is shown that how it can be used for solving the practical optimization problems. HBBO is capable of solving many types of optimization problems such as high-dimensional multimodal functions, which have multiple local minima, and unimodal functions. In order to demonstrate the performance of HBBO, the proposed algorithm has been tested on a set of well-known benchmark functions and compared with other optimization algorithms. The results have been shown that this algorithm outperforms other optimization algorithms in terms of algorithm reliability, result accuracy and convergence speed.

Keywords Human behavior-based optimization (HBBO) · Metaheuristic optimization methods · Evolutionary algorithms · Global optimization problems

✉ Seyed-Alireza Ahmadi
se.ahmadi@mail.sbu.ac.ir; salireza.ahmadi@yahoo.com;
http://a-ahmadi.com

[1] Electrical Engineering Department, Shahid Beheshti University, Tehran, Iran

1 Introduction

Optimization is the act of obtaining the best result under given circumstances [1]. The ultimate goal of this procedure is either to maximize the desired benefit or to minimize the required effort. Since the desired benefit or the required effort can be expressed as an objective function, optimization can be defined as the procedure of finding the suitable variables that give the minimum or maximum value of an objective function $f(x)$.

In the past few decades, due to the importance of optimization techniques, many new metaheuristic optimization algorithms have been developed. These algorithms do not have any limitations in inspiration resource. Many of them use nature as the principal source of inspiration. One of the main groups of the nature-inspired algorithms is biologically inspired algorithms. Evolutionary algorithms such as genetic algorithm (GA) [2], evolutionary programming (EP) [3, 4] and evolutionary strategies (ES) [5], which are inspired by biological evolution, are one of the main categories of biologically inspired algorithms. Swarm intelligence algorithms such as ant colony optimization (ACO) [6], artificial bee colony (ABC) [7], cuckoo search (CS) [8] and particle swarm optimization (PSO) [9], which are inspired by the social behavior of the animals, are the other category of biologically inspired algorithms.

Besides the biologically inspired algorithms, there is another group of nature-inspired algorithms such as simulated annealing (SA) [10], big bang-big crunch (BB-BC) [11] and charged system search (CSS) [12] that use physical phenomena as the main source of inspiration.

Regardless of inspiration resource, a powerful optimization algorithm can solve many important problems and is a vital need. Optimization algorithms have a wide application in many fields of science such as economics, business, computer science and aerospace, electrical and mechanical engineering [13–20].

In this paper, a new metaheuristic optimization algorithm for solving the global optimization problems is proposed, which uses human behavior as the main source of inspiration. The rest of this paper consists of three sections. In Sect. 2, human behavior will be discussed and HBBO will be proposed. Then, in Sect. 3, the experimental studies on the proposed HBBO will be presented, and finally, in Sect. 4, the paper will be concluded.

2 Human behavior-based optimization

In human society, everyone is moving toward his personal purposes, but may not attain it yet. A successful person is one who achieves all of his purposes. If a person desires to reach his goals, he has to attempt to be the best version of himself. As the viewpoints of people are different from each other, every individual finds success in one way and opts reaching some specific targets to achieve success. Because of that, people are working and studying in different fields and try to master in them. For example, one individual may find singing as the way of success, while the other finds it in sports. Among all people, which are working in a specific field, one individual is more expert than the others, so the others intend to learn from this expert individual and improve their skills in that field.

In addition, every individual of a society has many interests besides his professional field. For example, an electrical engineer may also be interested in painting besides his major interest. Our society consists of many people who have various viewpoints, but these viewpoints are not constant through their whole life. Everyone during his life contacts with a variety of people and uses their ideas and advices to improve his life. Each of these contacts can be considered as a meeting with an advisor, which can be effective or non-effective. In addition, each person may consult with other people who have different beliefs and they change his viewpoints. In some cases, that person may change his professional field because of that consultation and look for a better position in another field to improve himself.

Now, these simple behaviors of people, which were discussed, will be modeled and it will be shown that by this modeling, a powerful optimization algorithm will be achieved.

As mentioned before, every individual finds his success in one specific way, so in this algorithm, after generating the initial individuals, all of them spread in different fields. In each field, all individuals try to improve themselves by means of education process, in a way that will be described in Sect. 2.2, and after that, they find a random advisor from the whole society and start to consult with him. In addition, as it mentioned before that in the real society the beliefs of some people may alter and they change their job or educational field, in this algorithm, by considering a field changing probability, in some fields, an individual may find another way suitable and change his field. Finally, the stopping criteria will be checked, and if one of them reaches, the algorithm stops. This algorithm consists of the five steps as follows:

Step 1: Initialization
Step 2: Education
Step 3: Consultation
Step 4: Field changing probability
Step 5: Finalization

2.1 Initialization

This step devoted to generating and evaluating the initial individuals and spreading them among the fields in a way which is shown in Fig. 1. In an optimization problem with N_{var} variables, an individual is defined as follows:

$$\text{Individual} = [x_1, x_2, \ldots, x_{N_{\mathrm{var}}}] \tag{1}$$

The algorithm generates N_{pop} of initial individuals and randomly spreads them among N_{field} of initial fields. These individuals form the society. The number of initial individuals in each field is as follows:

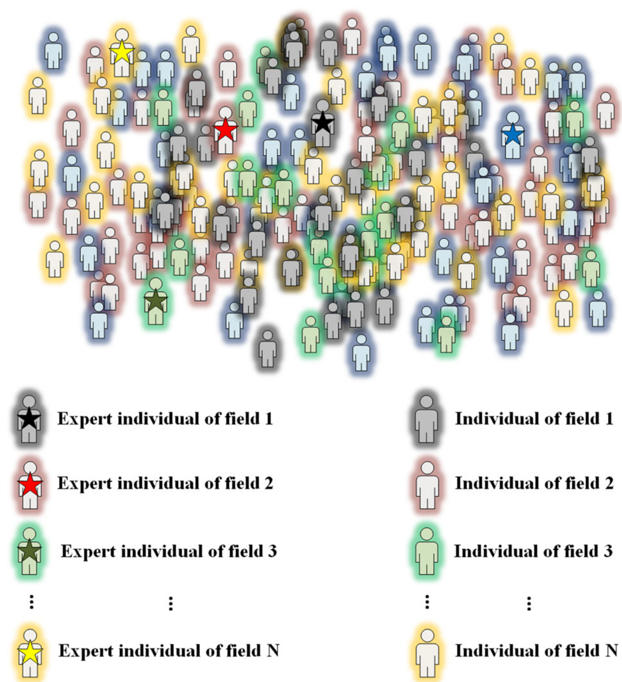


Fig. 1 Initialization: generating initial individuals and spreading them among the fields. Expert individuals are the best individuals of each field

$$N.\mathrm{Ind}_i = \mathrm{round}\left\{\frac{N_{\mathrm{pop}}}{N_{\mathrm{field}}}\right\} \quad (2)$$

where $N.\mathrm{Ind}_i$ is the number of initial individuals in i-th field. After generating the initial individuals, the function values of them will be calculated. The function value for an individual is defined as follows:

$$\text{function value} = f(x_1, x_2, \ldots, x_{N_{\mathrm{var}}}) \quad (3)$$

2.2 Education

In education process, every individual tries to learn and improve themselves by moving around the best individual of their field who is called expert individual. Expert individual is the one who has the best function value in each field (the best function value in a minimization problem is the minimum one and in a maximization problem is the maximum one). In order to model this procedure, coordinate system is implemented and the expert individual is the origin. This movement around expert individual for a three-dimensional problem is shown in Fig. 2 and will be performed by changing the coordinates of the individuals in spherical coordinate system. The movement space is limited by a sphere around the expert individual.

In an N-dimensional optimization problem by using the definition of spherical coordinate system for N-dimensional Euclidean space [21], the algorithm will find a random radial coordinate (r) between $r_{\min} = k_1 d$ and $r_{\max} = k_2 d$, where d is the Euclidian distance between the origin and individual, and k_i, as an algorithm parameter, is the weighting factor. In addition, the algorithm will find $N-1$ random angular coordinates $(\theta_1, \theta_2, \ldots, \theta_{N-1})$, where θ_{N-1} will be found between 0 and 2π radians and the other angles will be selected between 0 and π radians.

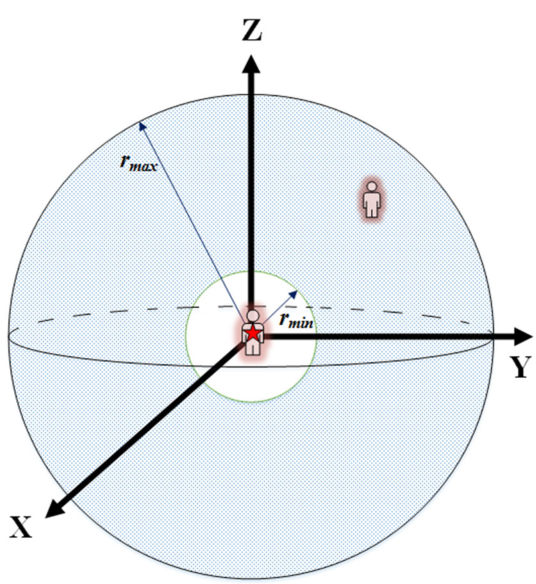


Fig. 2 Education: moving around the expert individual

2.3 Consultation

In this step, every individual (except the best individual of the society) finds a random advisor from the whole society and starts consulting with this advisor. In consultation process, the advisor will change some of the individual variables in a way that is shown in Fig. 3. Now if the new set of variables has a better function value, it means that their consultation is effective (a better function value in a minimization problem is a lower value and in a maximization problem is an upper value). In this case, the individual variables will be replaced with the new set of variables. Nevertheless, if the new set of variables has not a better function value, nothing will be changed. The number of random variables that will be changed is obtained as follows:

$$N_c = \mathrm{round}\{\sigma \times N_{\mathrm{var}}\} \quad (4)$$

where σ, as an algorithm parameter, is the consultation factor, which determines the number of random variables (N_c) that may be changed during the consultation process. Since checking function value for the new set of variables may need time in some cases (due to running the objective function), two different modes have been considered in this algorithm: advanced mode, which is just the same as the procedure depicted in Fig. 3 and explained before, and simple mode, which replaces the individual variables with the new set of variables without checking function value of them.

2.4 Field changing probability

As mentioned before, in each iteration, in some fields, an individual may change his field. The probability of this changing for each field is calculated using a rank probability method. In this method, every field is sorted according to their expert individual function value, as follows:

$$\text{sort fields} = [\mathrm{field}_1, \mathrm{field}_2, \ldots, \mathrm{field}_n] \quad (5)$$

where the expert individual of field_1 and field_n has the worst and the best function values among the others, respectively. After that, the changing probability for each field can be calculated as follows:

$$P_i = \frac{O_i}{N_{\mathrm{field}} + 1} \quad (6)$$

where P_i and O_i are the field changing probability and the sort order for the i-th field, respectively. By using this method, the field in which its expert individual has a better

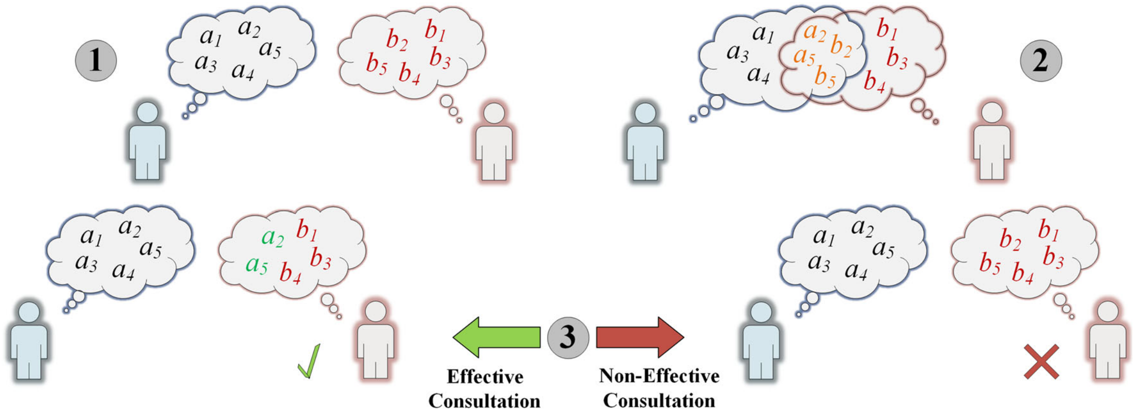


Fig. 3 Consultation in advanced mode: In this process, the advisor will change some of the individual variables, and if the new set of variables has a better function value, it means their consultation is effective

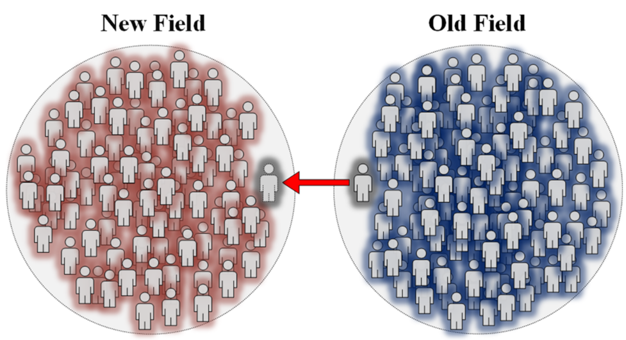


Fig. 4 Field changing: going to a random different field

function value is less probable, and the field in which its expert individual has a worse function value is more probable to see a field changing. After that, by generating a random number between 0 and 1, the following expression is checked, and if the expression is satisfied, the field changing for one of the individuals in this field occurs:

$$\text{if rand} \leq P_i \rightarrow \text{filed changing occurs} \tag{7}$$

In field changing process, according to the function value, a selection probability for each individual will be defined as follows:

$$P.S_j = \left| \frac{f(\text{Individual}_j)}{\sum_{k=1}^{N_{\text{ind}}} f(\text{Individual}_k)} \right| \tag{8}$$

where $P.S_j$ is the selection probability for the j-th individual and N_{ind} is the number of individuals in the selected field. After that, by using the roulette wheel selection method [22], an individual will be selected and will change his field by going to a random different field as shown in Fig. 4.

2.5 Finalization

By performing consultation and education process, the position of the individuals changes. Therefore, in this step, function values of the individuals will be calculated, and if one of the stopping criteria is met, the algorithm will be terminated; otherwise, the algorithm will go to step 2. The stopping criteria are as follows:

(a) The number of iterations reaches to maximum iterations.
(b) The maximum number of function evaluations is reached.
(c) The average relative change in the objective function value over stall iterations becomes less than function tolerance.

In the next section, the proposed algorithm will be tested on a set of benchmark functions.

3 Experimental studies

As the no free lunch theorem expresses, “for any algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class” [23]. To completely evaluate the HBBO performance, a set of 14 well-known benchmark functions is used in the experimental studies. These functions, which are given in Table 1, can be grouped into 30-dimensions unimodal functions (f_1–f_7), low-dimensional multimodal functions (f_8–f_9) and 30-dimensions multimodal functions (f_{10}–f_{14}).

The parameter setting of the HBBO is listed in Table 2. The initial population is generated in the range specified in Table 1. In all experiments, the same parameters are used.

Table 1 Benchmark functions, where n is the dimension of the function, S is the search space, and $f_{\min}$ is the minimum value of the function

Test function	n	S	$f_{\min}$
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]^n$	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30	$[-10, 10]^n$	0
$f_3(x) = \sum_{i=1}^{n} \lvert x_i\rvert^{i+1}$	30	$[-5, 5]^n$	0
$f_4(x) = \max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	30	$[-100, 100]^n$	0
$f_5(x) = \sum_{i=1}^{n-1} \left(100\left(x_{i+1} - x_i^2\right)\right)^2$	30	$[-30, 30]^n$	0
$f_6(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5\rfloor)^2$	30	$[-100, 100]^n$	0
$f_7(x) = \sum_{i=1}^{n} ix_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]^n$	0
$f_8(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]^n$	-1.0316285
$f_9(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	$[-5, 10] \times [0, 15]$	0.3978
$f_{10}(x) = -\sum_{i=1}^{n} \left(x_i \sin(\sqrt{\lvert x_i\rvert})\right)$	30	$[-500, 500]^n$	-12569.5
$f_{11}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right) + 20 + e$	30	$[-32, 32]^n$	0
$f_{12}(x) = \frac{1}{4000}\sum_{i=1}^{n} (x_i - 100)^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^n$	0
$f_{13} = \frac{\pi}{n}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a \le x_i \le a \\ k(-x_i - a)^m, x_i < a \end{cases}$	30	$[-50, 50]^n$	0
$f_{14}(x) = 0.2\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2\left[1 + \sin^2(3\pi x_{i+1})\right] + (x_n - 1)^2\left[1 + \sin^2(2\pi x_{30})\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	$[-50, 50]^n$	0

Table 2 HBBO parameters

Parameter	Value
N_{pop}	150
N_{field}	30
k_1	0
k_2	2.5
σ	0.2

In order to compare the accuracy and convergence rate of HBBO with other optimization algorithms, six different algorithms are selected:

1. Genetic algorithm (GA)
2. Particle swarm optimization (PSO)
3. Conventional evolutionary strategies (CES)
4. Conventional evolutionary programming (CEP)
5. Fast evolutionary strategies (FES)
6. Fast evolutionary programming (FEP)

For ES and EP, as there are no publically available toolboxes, the test results for f_1, f_3 and f_4–f_{14} are directly adopted from [24]. Function f_3 is not in their experiments. Yao and Liu proposed fast ES (FES) and fast EP (FEP), in which the Gaussian mutations in conventional ES (CES) and conventional EP (CEP) are replaced with Cauchy mutation [25, 26]. For GA, MATLAB optimization toolbox is employed which uses the adaptive feasible mutation and scattered crossover. The selection function using in it is the stochastic uniform. For PSO, a constriction coefficient parameter that is introduced in [27] is used. In [27], Clerc and Kennedy proposed a set of parameters that insure the convergence of PSO. The constriction coefficient parameter PSO has much better performance among the other versions of PSO algorithms. For all of the algorithms, the same population size of 150 is used. In order to have a fair comparison among algorithms, the same number of function evaluations is used. For each benchmark functions, the maximum number of function evaluations is listed in Table 3. As it can be seen from this table, the maximum number of function evaluations is equal or less than those which were used in [24].

To further assess the HBBO performance and statistically compare it with GA and PSO, a set of two-tailed t test

Table 3 Number of function evaluations

Function	HBBO/GA/PSO	CES/FES and CEP/FEP
f_1	150000	150000
f_2	200000	200000
f_3	100000	–
f_4	250000	500000
f_5	250000	2000000
f_6	50000	150000
f_7	150000	300000
f_8	3500	10000
f_9	3500	10000
f_{10}	100000	900000
f_{11}	150000	150000
f_{12}	150000	200000
f_{13}	150000	150000
f_{14}	150000	150000

is used [28]. In t test, the critical value of t is set to be ± 1.98 with 98 degrees of freedom at $\alpha = 0.05$. All of the experimental results are obtained based on 50 independent runs.

The experiments are done on a PC with Intel Core i5 2.50-GH processor and 8.0-GB RAM. All of the programs are implemented in MATLAB 2013. The operation system is Microsoft Windows 7.

3.1 Unimodal functions

Functions (f_1–f_7) are unimodal functions, and the 3D view for some of the functions is shown in "Appendix 2." Function f_6 is the step function, which is discontinuous and has one minimum. Function f_7 is a noisy quartic function where $\mathrm{random}[0, 1)$ is a uniformly distributed random variable in [0, 1) [29]. It should be noted that unimodal functions are not very challenging problems and they can be solved by many optimization algorithms. The aim of using them is to assess the convergence rate of the optimization algorithms [29]. Table 4 shows the mean best values and the standard deviations of the founded solutions. The best results are in boldface. In addition, the average CPU times in seconds and the t test results between HBBO and GA, and also, HBBO and PSO are given in Table 4 in front of the GA and PSO results, respectively. All results have been averaged over 50 independent runs. The results adopted from [24] for FEP, CEP, FES and CES are listed in Table 5 in comparison with the HBBO results.

In order to compare the convergence rate of HBBO with GA, and PSO, the evolutionary process of these algorithms on unimodal functions is shown in Fig. 5. This figure shows the mean best solutions found by HBBO, GA and PSO over 50 independent runs. As it can be observed from Table 4 and Fig. 5, HBBO performs better than GA

Table 4 Comparison of HBBO with GA and PSO on benchmark functions f_1–f_7. All results have been averaged over 50 runs

Function	Algorithms	Mean best	SD	t test	CPU
f_1	GA	1.6436×10^{-8}	6.8224×10^{-9}	$-17.03^{\dagger}$	9.1
	PSO	8.3840×10^{-22}	3.0437×10^{-21}	-1.94	8.2
	HBBO	$\mathbf{1.2809 \times 10^{-43}}$	4.3157×10^{-43}	–	5.0
f_2	GA	2.9392×10^{-4}	6.3322×10^{-5}	$-32.82^{\dagger}$	12.5
	PSO	1.6533×10^{-22}	8.4294×10^{-22}	-1.38	11.7
	HBBO	$\mathbf{4.6940 \times 10^{-34}}$	5.5754×10^{-34}	–	6.9
f_3	GA	2.5094×10^{-11}	7.6304×10^{-11}	$-2.32^{\dagger}$	6.6
	PSO	2.3814×10^{-13}	9.4588×10^{-13}	-1.78	6.1
	HBBO	$\mathbf{7.5439 \times 10^{-65}}$	3.5074×10^{-64}	–	3.7
f_4	GA	0.5153	5.9166×10^{-2}	$-61.06^{\dagger}$	15.5
	PSO	1.9678×10^{-2}	1.1508×10^{-2}	$-9.59^{\dagger}$	14.7
	HBBO	$\mathbf{3.1110 \times 10^{-3}}$	4.0750×10^{-3}	–	8.4
f_5	GA	65.7257	171.22456	$-2.71^{\dagger}$	15.6
	PSO	1.8185×10^{-9}	1.0069×10^{-8}	-1.27	14.8
	HBBO	$\mathbf{4.0296 \times 10^{-24}}$	2.5485×10^{-23}	–	8.6
f_6	GA	16.0601	4.7808	$-23.75^{\dagger}$	3.1
	PSO	0.7600	0.9707	$-5.53^{\dagger}$	2.8
	HBBO	**0**	0	–	1.7
f_7	GA	0.2819	9.2215×10^{-2}	$-19.19^{\dagger}$	10.3
	PSO	8.3645×10^{-2}	2.6705×10^{-2}	$-13.10^{\dagger}$	9.1
	HBBO	$\mathbf{2.9566 \times 10^{-2}}$	1.1764×10^{-2}	–	6.1

Table 5 Comparison of HBBO with FEP, CEP, FES, and CES on benchmark functions f_1–f_7. All results have been averaged over 50 runs

Function	Mean best function value (rank) SD				
	HBBO	FEP	CEP	FES	CES
f_1	1.2809×10^{-47} (**1**)	5.7×10^{-4} (**5**)	2.2×10^{-4} (**3**)	2.5×10^{-4} (**4**)	3.4×10^{-5} (**2**)
	$(4.31578 \times 10^{-47})$	(1.3×10^{-4})	(5.9×10^{-4})	(6.8×10^{-4})	(8.6×10^{-6})
f_2	4.6940×10^{-34} (**1**)	8.1×10^{-3} (**3**)	2.6×10^{-3} (**2**)	6.0×10^{-2} (**5**)	2.1×10^{-2} (**4**)
	(5.5754×10^{-34})	(7.7×10^{-4})	(1.7×10^{-4})	(9.6×10^{-3})	(2.2×10^{-3})
f_4	3.1110×10^{-3} (**1**)	0.3 (**3**)	2.0 (**5**)	5.5×10^{-3} (**2**)	0.35 (**4**)
	(4.0750×10^{-3})	(0.5)	(1.2)	(6.5×10^{-4})	(0.42)
f_5	4.0296×10^{-24} (**1**)	5.06 (**2**)	6.17 (**3**)	33.28 (**5**)	6.69 (**4**)
	(2.5485×10^{-23})	(5.87)	(13.61)	(43.13)	(14.45)
f_6	0 (**1**)	0 (**1**)	577.76 (**5**)	0 (**1**)	411.16 (**4**)
	(0)	(0)	(1125.76)	(0)	(695.35)
f_7	2.9566×10^{-2} (**4**)	7.6×10^{-3} (**1**)	1.8×10^{-2} (**3**)	1.2×10^{-2} (**2**)	3.0×10^{-2} (**5**)
	(1.1764×10^{-2})	(2.6×10^{-3})	(6.4×10^{-3})	(5.8×10^{-3})	(1.5×10^{-2})
Average rank	1.50	2.50	3.50	3.16	3.83
Final rank	1	2	4	3	5

and PSO in terms of final result and convergence speed. In addition, by comparing the CPU time of HBBO with GA and PSO, it can be found that HBBO requires less CPU time than GA and PSO. Also, it can be seen from Table 5 that HBBO is ranked the first and outperforms FEP, CEP, FES and CES. However, HBBO requires fewer function evaluations.

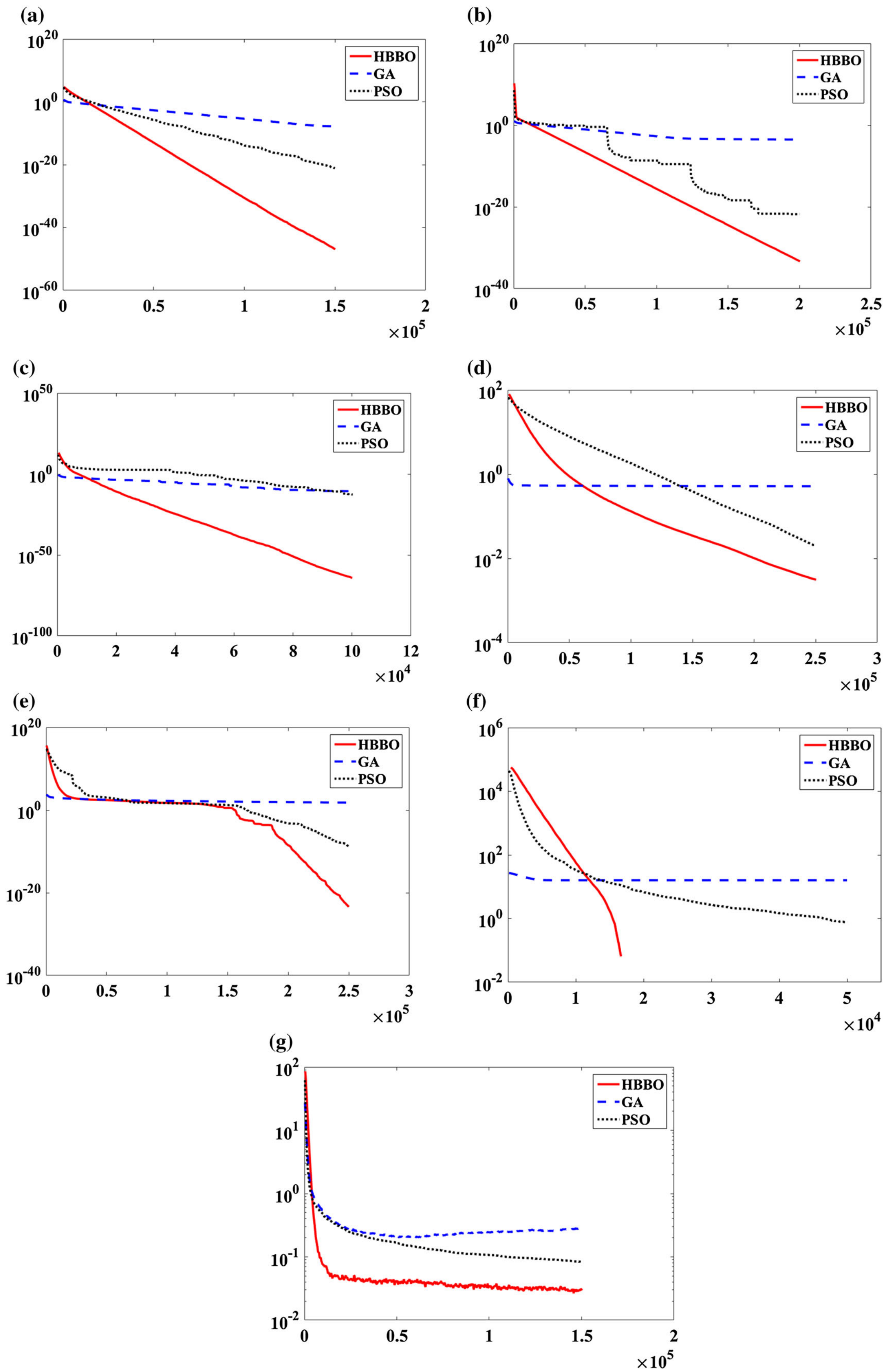
3.2 Multimodal functions

Functions f_8–f_{14} are multimodal functions in which the number of their local minima increases exponentially with the dimension of the function [29]. The 3D view for some of the functions is shown in "Appendix 2." In optimization problems, high-dimensional multimodal functions are among the most difficult class of problems. Functions f_8 and f_9 are two dimensions, and f_{10}–f_{14} are 30-dimensions multimodal functions. The mean best values, standard deviations, t test results and CPU times are given in Table 6. For low-dimensional multimodal functions, the t test results indicate that there is no statistical difference between HBBO and PSO, but HBBO is statistically better than GA. For 30-dimension multimodal functions, HBBO performs significantly better than GA and PSO.

For example, on function f_{10}, HBBO finds an acceptable minimum with only 100,000 function evaluations, while this number of function evaluations is not enough for GA to find an acceptable minimum. Also on function f_{12}, HBBO finds the global minimum in every run, while GA and PSO trapped in poor local minima. From Table 6, it can be seen that HBBO outperforms GA and PSO in terms of algorithm reliability, final result and convergence speed. The average CPU time indicates that HBBO needs less CPU time than GA and PSO. The result adopted from [24], in comparison with the HBBO results, is listed in Table 7. As it can be seen from this table, HBBO is ranked the first and is significantly better than these four algorithms. For six out of seven functions, HBBO generates better results and the only exception is function f_{10}. The reason of this exception is considering fewer function evaluations. Therefore, HBBO outperforms FEP, CEP, FES and CES with fewer function evaluations.

Additionally, another comparison test study has been conducted between HBBO and other newly introduced optimization algorithms containing animal migration optimization (AMO) [30], Multi-Verse Optimizer (MVO) [31] and biogeography-based optimization with chaos (CBBO) [32] which are recently published in Neural Computing and Application Journal. The simulation results proof the privilege of the HBBO. In order not to extend the paper, these results are not shown in the manuscript.

From these studies, it can be concluded that HBBO is a reliable and fast optimization algorithm and is significantly better than the other mentioned optimization algorithms.


(a)
HBBO
GA
PSO
(b)
HBBO
GA
PSO
(c)
HBBO
GA
PSO
(d)
HBBO
GA
PSO
(e)
HBBO
GA
PSO
(f)
HBBO
GA
PSO
(g)
HBBO
GA
PSO

◀ **Fig. 5** Evolutionary process of HBBO, GA and PSO on unimodal functions. The vertical axis is the function value, and the horizontal axis is the number of function evaluations. The figures show the average best results over 50 runs. **a** f_1 (sphere). **b** f_2 (Schwefel's problem 2.22). **c** f_3 (sum power). **d** f_4 (Schwefel's problem 2.21). **e** f_5 (generalized Rosenbrock's function). **f** f_6 (step function). **g** f_7 (quartic function with noise)

4 Conclusion

"Almost every problem in engineering (and in life) can be interpreted as an optimization problem" [33, 34]. In this paper, by modeling human behavior, a novel meta-heuristic optimization algorithm, namely HBBO, has been

Table 6 Comparison of HBBO with GA and PSO on benchmark functions f_8–f_{14}

Function	Algorithms	Mean best	SD	t test	CPU
f_8	GA	−1.029358	7.0929×10^{-3}	$-2.26^{\dagger}$	0.2
	PSO	−1.031628	4.5721×10^{-16}	0	0.2
	HBBO	**−1.031628**	3.5108×10^{-16}	–	0.1
f_9	GA	0.4008	5.6355×10^{-3}	$-3.76^{\dagger}$	0.2
	PSO	0.3978	1.8522×10^{-15}	0	0.2
	HBBO	**0.3978**	3.3306×10^{-16}	–	0.1
f_{10}	GA	−118.3590	9.3498×10^{-7}	$-984.08^{\dagger}$	6.3
	PSO	−10629.6512	499.4823	$-26.07^{\dagger}$	5.8
	HBBO	**−12500.7619**	88.9731	–	3.2
f_{11}	GA	8.7963×10^{-5}	1.4343×10^{-5}	$-43.36^{\dagger}$	9.6
	PSO	1.6671×10^{-7}	3.8286×10^{-7}	$-3.07^{\dagger}$	8.8
	HBBO	$\mathbf{1.0693 \times 10^{-14}}$	4.0544×10^{-15}	–	5.4
f_{12}	GA	1.4450×10^{-2}	1.8280×10^{-2}	$-5.58^{\dagger}$	9.5
	PSO	2.8107×10^{-2}	2.6100×10^{-2}	$-7.61^{\dagger}$	8.8
	HBBO	**0**	0	–	5.2
f_{13}	GA	1.0375×10^{-2}	3.1126×10^{-2}	$-2.35^{\dagger}$	26.1
	PSO	3.2642×10^{-12}	6.7063×10^{-12}	$-3.44^{\dagger}$	25.4
	HBBO	$\mathbf{5.5093 \times 10^{-26}}$	7.2111×10^{-26}	–	21.0
f_{14}	GA	1.4221×10^{-8}	7.4348×10^{-9}	$-13.52^{\dagger}$	25.0
	PSO	8.1011×10^{-16}	1.0732×10^{-15}	$-5.33^{\dagger}$	24.7
	HBBO	$\mathbf{3.2975 \times 10^{-32}}$	4.1491×10^{-32}	–	20.9

All results have been averaged over 50 runs

Table 7 Comparison of HBBO with FEP, CEP, FES, and CES on benchmark functions f_8–f_{14}

Function	Mean best function value (rank) (SD)				
	HBBO	FEP	CEP	FES	CES
f_8	−1.031628 (**1**)	−1.03 (**2**)	−1.03 (**2**)	−1.0316 (**4**)	−1.0316 (**4**)
	(3.5108×10^{-16})	(4.9×10^{-4})	(4.9×10^{-4})	(6.0×10^{-7})	(6.0×10^{-7})
f_9	0.3978 (**1**)	0.398 (**4**)	0.398 (**4**)	0.398 (**2**)	0.398 (**2**)
	(3.3306×10^{-16})	(1.5×10^{-7})	(1.5×10^{-7})	(6.0×10^{-8})	(6.0×10^{-8})
f_{10}	−12500.7619 (**3**)	−12554.5 (**2**)	−7917.1 (**4**)	−12556.4 (**1**)	−7549.9 (**5**)
	(88.9731)	(52.6)	(634.5)	(32.53)	(631.39)
f_{11}	1.0693×10^{-14} (**1**)	1.8×10^{-2} (**3**)	9.2 (**5**)	1.2×10^{-2} (**2**)	9.07 (**4**)
	(4.0544×10^{-15})	(2.1×10^{-2})	(2.8)	(1.8×10^{-3})	(2.84)
f_{12}	0 (**1**)	1.6×10^{-2} (**2**)	8.6×10^{-2} (**4**)	3.7×10^{-2} (**3**)	0.38 (**5**)
	(0)	(2.2×10^{-2})	(0.12)	(5.0×10^{-2})	(0.77)
f_{13}	5.5093×10^{-26} (**1**)	9.2×10^{-6} (**2**)	1.76 (**5**)	2.8×10^{-2} (**3**)	1.18 (**4**)
	(7.2111×10^{-26})	(6.1395×10^{-5})	(2.4)	(8.1×10^{-11})	(1.87)
f_{14}	3.2975×10^{-32} (**1**)	1.6×10^{-4} (**3**)	1.4 (**5**)	4.7×10^{-5} (**2**)	1.39 (**4**)
	(4.1491×10^{-32})	(7.3×10^{-5})	(3.7)	(1.5×10^{-5})	(3.33)
Average rank	1.28	2.57	4.14	2.42	4.00
Final rank	1	3	5	2	4

All results have been averaged over 50 runs proposed. Despite many of the optimization algorithms that are nature inspired, HBBO uses human behavior as the main source of inspiration. This algorithm proves that optimization does not have any limitation in inspiration resource, and with a closer look, two scientific areas that seems to be irrelevant can combine and make stunning results.

Comprehensive experimental investigations have been conducted on 14 benchmark functions including unimodal, low- and high-dimensional multimodal functions. The experimental results indicate that HBBO outperforms other optimization algorithms in terms of algorithm reliability, result accuracy and convergence speed. The average CPU time is an important factor for comparing optimization algorithms. The results show that HBBO requires less CPU time and is the fastest optimization algorithm among the others. HBBO is easy to implement and able to solve a variety of complex real-world optimization problems. The future works include the studies on how to use HBBO to solve these optimization problems.

Compliance with ethical standards

Conflict of interest The author declares that he has no conflict of interest.

Appendix 1

The software was used to generate the results of HBBO in this paper, and the implementation guides will be publicly available at http://a-ahmadi.com/hbbo.

Appendix 2

See Figs. 6, 7, 8, 9, 10, 11 and 12.

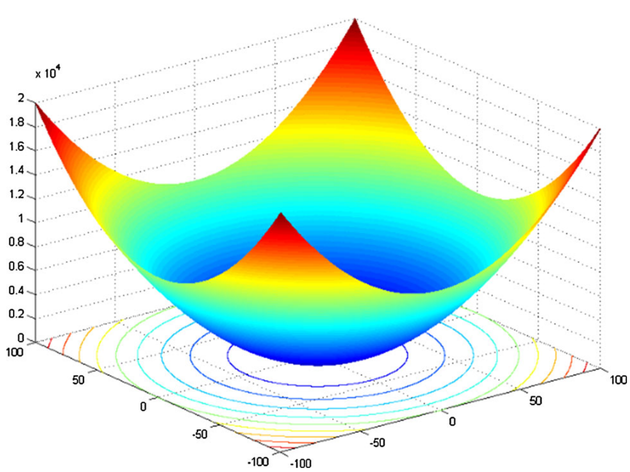


Fig. 6 3D view for function f_1 (sphere)

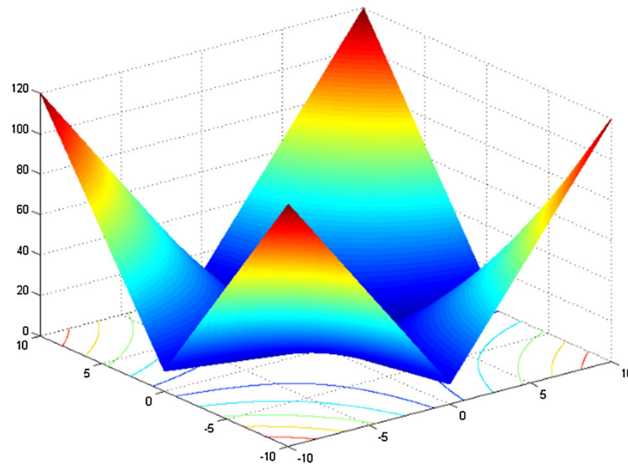


Fig. 7 3D view for function f_2 (Schwefel's problem 2.22)

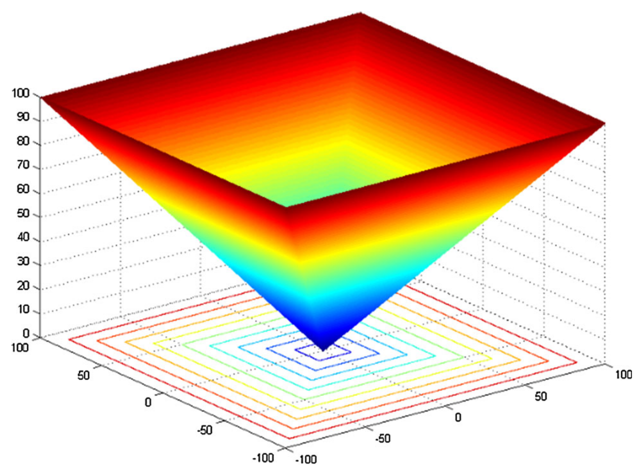


Fig. 8 3D view for function f_4 (Schwefel's problem)

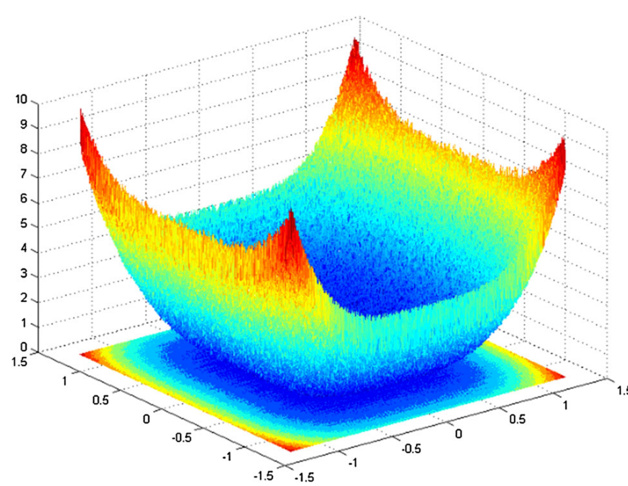


Fig. 9 3D view for function f_7 (quartic function with noise)

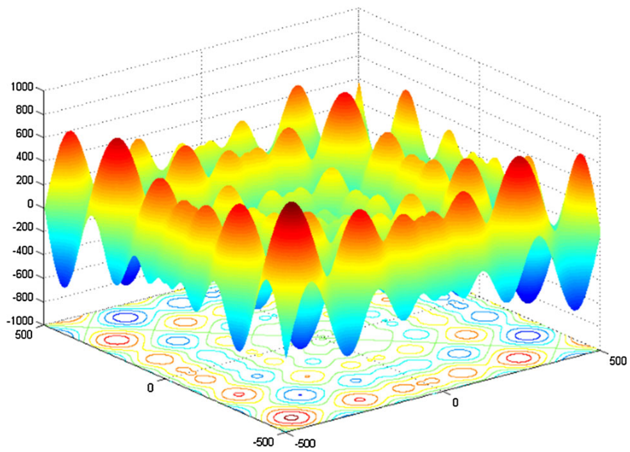


Fig. 10 3D view for function f_{10} (Schwefel)

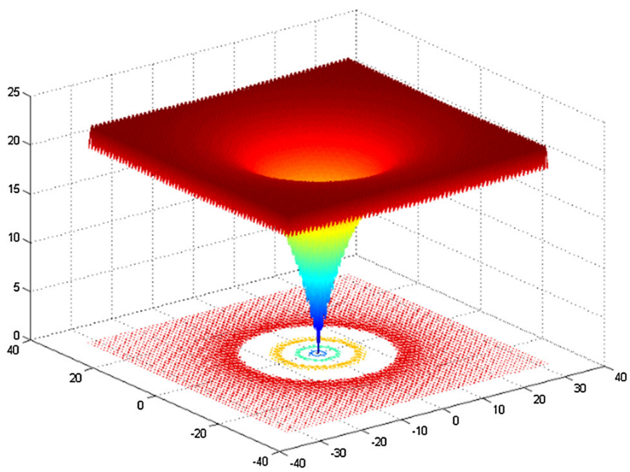


Fig. 11 3D view for function f_{11} (rotated Ackley)

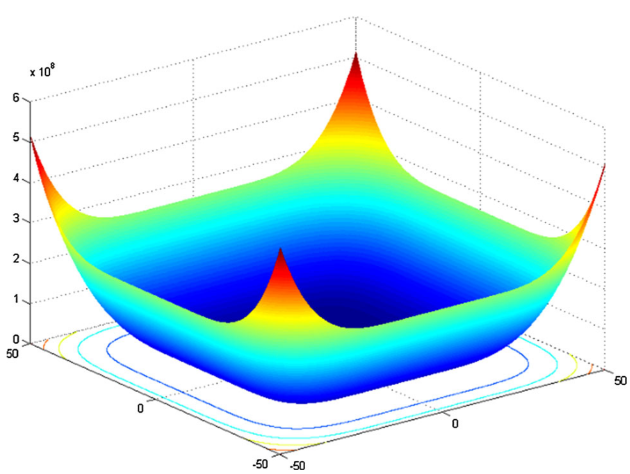


Fig. 12 3D view for function f_{13} (generalized penalized)

References

1. Rao SS (2009) Engineering optimization: theory and practice, 4th edn. Wiley, Hoboken
2. Holland JH (1975) Adaption in natural and artificial systems. Ann Arbor, MI
3. Fogel DB (1995) Evolutionary computation: toward a new philosophy of machine intelligence. IEEE Press, New York
4. Fogel LJ, Owens AJ, Walsh MJ (1965) Artificial intelligence through a simulation of evolution. In: Proceedings of 2nd Cybern. Sci. Symp. Biophysics Cybern. Syst. Spartan Books, Washington, pp 131–155
5. Schwefel HP (1995) Evolution and optimum seeking. Wiley, New York
6. Dorigo M, Maniezzo V, Colorni A (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B 26(1):29–41
7. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. J Global Optim 39(3):459471
8. Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29(1):17–35
9. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural network, vol 4. IEEE Press, Piscataway, pp 1942–1948
10. Kirkpatrick S, Gelatt C, Vecchi M (1983) Optimization by simulated annealing. Science 220(4598):671–680
11. Erol OK, Eksin I (2006) A new optimization method: big bangbig crunch. Adv Eng Softw 37(2)
12. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. Acta Mech 213:267–289
13. Kalamkarov AL, Kolpakov AG (1997) Analysis, design and optimization of composite structures, 2nd edn. Wiley, New York
14. Gurdal Z, Haftka RT, Hajela P (1998) Design and optimization of laminated composite materials. Wiley, New York
15. Jaluria Y (2007) Design and optimization of thermal systems, 2nd edn. CRC Press, Boca Raton
16. Edgar TF, Himmelblau DM (1988) Optimization of chemical processes. McGraw-Hill, New York
17. Micheli GD (1994) Synthesis and optimization of digital circuits. McGraw-Hill, New York
18. Johnson RC (1980) Optimum design of mechanical elements. Wiley, New York
19. Ravindran A, Ragsdell KM, Reklaitis GV (2006) Engineering optimization: methods and applications, 2nd edn. Wiley, New York
20. Brown JJ, Chen DZ, Greenwood GW, Hu RWTXS (1997) Scheduling for power reduction in a real-time system. In: International symposium on low power electronics and design. IEEE, Monterey, pp 84–87
21. Surhone LM, Timpledon MT, Marseken SF (2010) Spherical coordinate system: vector fields in cylindrical and spherical coordinates, sphere, N-sphere, Cartesian coordinate system. Cylindrical Coordinate System, Betascript
22. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading
23. Wolpert DH, Macready WG (1997) No free lunch theorems for search. IEEE Trans Evol Comput 1(1):67–82
24. He S, Wu QH, Saunders JR (2009) Group search optimizer: an optimization algorithm inspired by animal searching behavior. IEEE Trans Evol Comput 13(5):973–990

25. Yao X, Liu Y, Liu G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3(2):82–102
26. Yao X, Liu Y (1997) Fast evolution strategies. In: Proceedings of evolution programming VI. Springer, Berlin, pp 151–161
27. Clerc M, Kennedy J (2002) The particle swarmexplosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73
28. Montgomery DC (2012) Statistical quality control. Wiley, New York
29. Torn A, Zilinskas A (1989) Global optimisation. Lecture Notes in Computer Science. Springer, Berlin
30. Li X, Zhang J, Yin M (2014) Animal migration optimization: an optimization algorithm inspired by animal migration behavior. Neural Comput Appl 24(7–8):1867–1877. doi:10.1007/s00521-013-1433-8
31. Mirjalili S, Mirjalili SM, Hatamlou A (2015) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl. doi:10.1007/s00521-015-1870-7
32. Saremi S, Mirjalili S, Lewis A (2014) Biogeography-based optimisation with chaos. Neural Comput Appl 25:1077–1097. doi:10.1007/s00521-014-1597-x
33. Bernstein D (2006) Optimization r us. IEEE Control Syst Mag 26:6–7
34. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713