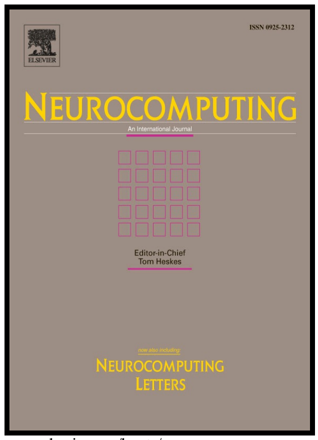
Author's Accepted Manuscript



www.elsevier.com/locate/neucom

Opposition-based krill herd algorithm with Cauchy mutation and position clamping

Gai-Ge Wang, Suash Deb, Amir H. Gandomi, Amir H. Alavi

PII: S0925-2312(15)01728-2
DOI: http://dx.doi.org/10.1016/j.neucom.2015.11.018
Reference: NEUCOM16355

To appear in: *Neurocomputing*

Received date: 31 August 2015
Revised date: 1 October 2015
Accepted date: 12 November 2015

Cite this article as: Gai-Ge Wang, Suash Deb, Amir H. Gandomi and Amir H. Alavi, Opposition-based krill herd algorithm with Cauchy mutation and positio clamping, *Neurocomputing,* http://dx.doi.org/10.1016/j.neucom.2015.11.018

TITLE PAGE

Title

"Opposition-based krill herd algorithm with Cauchy mutation and position clamping"

1st & Corresponding Author

Gai-Ge Wang*

School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, Jiangsu, 221116, China

Institute of Algorithm and Big Data Analysis, Northeast Normal University, Changchun, 130117, China

School of Computer Science and Information Technology, Northeast Normal University, Changchun, 130117, China

E-mail: gaigewang@163.com (G.-G. Wang); gaigewang@gmail.com (G.-G. Wang)

2nd Author

Suash Deb

Department of Computer Science & Engineering, Cambridge Institute of Technology, Cambridge Village, Tatisilwai, Ranchi 835103, Jharkhand, INDIA

Email: suashdeb@gmail.com

3rd Author

Amir H. Gandomi

Pioneer of Krill Herd Algorithm

BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, MI 48824, USA

E-mail: a.h.gandomi@gmail.com; ag72@uakron.edu

4th Author

Amir H. Alavi

Pioneer of Krill Herd Algorithm

Department of Civil and Environmental Engineering, Michigan State University, East Lansing, MI 48824, USA

E-mail: ah_alavi@hotmail.com, alavi@msu.edu

Number of Words: 6926

Number of Tables: 13

Number of Figures: 4

Number of Appendices: 0

Number of Supplementary Material: 0

Opposition-based krill herd algorithm with Cauchy mutation and position clamping

Abstract

Krill herd (KH) has been proven to be an efficient algorithm for function optimization. For some complex functions, this algorithm may have problems with convergence or being trapped in local minima. To cope with these issues, this paper presents an improved KH-based algorithm, called Opposition Krill Herd (OKH). The proposed approach utilizes opposition-based learning (OBL), position clamping (PC) and Cauchy mutation (CM) to enhance the performance of basic KH. OBL accelerates the convergence of the method while both PC and heavy-tailed CM help KH escape from local optima. Simulations are implemented on an array of benchmark functions and two engineering optimization problems. The results show that OKH has a good performance on majority of the considered functions and two engineering cases. The influence of each individual strategy (OBL, CM and PC) on KH is verified through twenty-five benchmarks. The results show that the KH with OBL, CM and PC operators, has the best performance among different variants of OKH.

Keywords

Krill herd; Opposition-based learning; Cauchy mutation; Position clamping; Engineering optimization

1. Introduction

Inspired by nature, a variety of modern intelligent algorithms have been developed and applied to solve optimization problems. Some of them, like cuckoo search (CS) [1-5], biogeography-based optimization (BBO) [6-13], artificial bee colony (ABC) [14, 15], genetic algorithm (GA) [16], genetic programming (GP) [17], stud GA (SGA) [18, 19], differential evolution (DE) [20-25], ant lion optimizer (ALO) [26], chicken swarm optimization (CSO) [27], wolf search algorithm (WSA) [28], multi-verse optimizer (MVO) [29], earthworm optimization algorithm (EWA) [30], grey wolf optimizer (GWO) [31, 32], firefly algorithm (FA) [33-35], dragonfly algorithm (DA) [36], harmony search (HS) [37-41], bird swarm algorithm (BSO) [42], moth-flame optimization (MFO) [43], animal migration optimization (AMO) [44], particle swarm optimization (PSO) [45-48], ant colony optimization (ACO) [49], bat algorithm (BA) [50-54], have solved several complicated challenging problems that are hard to deal with by traditional optimization techniques. Among these algorithms, krill herd (KH) method [55-57] has been studied extensively due to its promising performance for solving most complex problems. KH was first proposed by Gandomi and Alavi by the idealization of communicating and foraging behaviors of krill swarms [55]. KH performed well on various optimization problems [55]. However, in some cases, it might not be capable of escaping from local minima. In order to decrease the influence of this problem for KH, this paper proposes different variants of KH algorithm using opposition-based learning (OBL), position clamping (PC) and Cauchy mutation (CM). The main idea of OBL is to search for a better candidate solution through the simultaneous consideration of a solution and its opposite that is closer to the global optimum. OBL can successfully handle this task by updating the other half krill according to the previous ones following its basic theory. Consequently, a faster convergence can be provided for the KH method. The heavy-tailed CM and PC help the krill not trap into the local optima. Through different experiments, the KH together with OBL, CM and PC operators performs the best among various OKHs. Experimental simulations on twenty-five benchmark functions and two engineering optimization problems show that OKH performs well on the majority of benchmark functions and engineering problems.

The remainder of this paper is organized as follows. The next section introduces the main process of the basic KH and OBL theory. Section 3 proposes an improved OKH model by combination of KH, CM operator and PC operator. Then, in Section 4, a series of comparison experiments on various benchmarks and two engineering cases are conducted. The final section provides our concluding remarks and points out our future work orientation.

2. Preliminary

2.1 Krill herd

In computer science, KH [55] is a probabilistic technique for solving computational problems. It is a kind of swarm intelligence algorithms that take advantage of the evolving behaviors of krill individuals. It is based on the idealization of the krill swarms when hunting for food and communicating with each other. The KH method repeats the implementation of the three movements and takes search directions that proceed to the best solution. The behavior of krill is idealized into three actions as:

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

KH algorithm adopted the following Lagrangian model as **Eq.** (1).

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

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 motion direction, α_i, is primarily decided by the target effect, a local effect, and a repulsive effect. For a krill, it can be given as:

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

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

The second motion is mainly decided by the two factors: the food location and its previous experience. For the ith krill, it can be defined as:

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

where

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

and V_f is the foraging speed, ω_f is the inertia weight for the second motion, F_i^{old} is the last motion, β_i^{food} is the food attractive and β_i^{best} is the effect of the best fitness of the ith krill so far.

The third motion is essentially a random process. It contains two parts: a maximum diffusion speed and a random directional vector. It can be expressed as follows:

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

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

Based on the three above-mentioned movements, the position of a krill from t to $t+\Delta t$ is given as:

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

In standard KH, the process of movement influenced by other krill, foraging action and physical diffusion continue for a fixed number of generations or until a termination Condition is satisfied. More detailed description about KH algorithm can be referred as [55].

2.2 Opposition-based learning

OBL [58, 59] is a relatively novel technology in optimization field. It has been shown to be a promising approach to improve the performance of many metaheuristic algorithms, such as ACO [60], DE [61-63], population-based incremental learning (PBIL) [64, 65], PSO [66] and ABC [67]. Evidently, a simultaneous evaluation of a solution and its opposite can increases the chance of finding an individual close to the global best solution. The definition of the *Opposite number* (1-dimensional) and *opposite point* (*D*-dimensional) are given below.

Opposite number [66]: let $x \in [a,b]$ be a real number. Its opposite is defined by:

$$x^* = a + b - x \tag{7}$$

Opposite point [66]: let $X=(x_1, x_2, ..., x_D)$ be a point in a *D*-dimensional space, where $x_1, x_2, ..., x_D \in R$ and $x_i \in [a_i, b_i]$, $i \in 1, 2, ..., D$. The definition of the opposite point $X^* = (x_1^*, x_2^*, \cdots, x_D^*)$ is as follows:

$$x_i^* = a_i + b_i - x_i \tag{8}$$

The definition of opposition-based optimization can be given in terms of opposite point as follows.

Opposition-based optimization [66]: let $X=(x_1, x_2, ..., x_D)$ be a candidate solution in search space, and its fitness function is $f(X)$. As per Eq. (8), $X^* = (x_1^*, x_2^*, \cdots, x_D^*)$ is the opposite of $X=(x_1, x_2, ..., x_D)$. If $f(X^*)<f(X)$, then update X with X^*; otherwise X is unchanged. Therefore, simultaneously computing and evaluating the current point and its opposite makes the population proceed to the best solution.

3. Improving KH using OBL

This study is aimed at improving the performance of KH by combining it with OBL, CM, and PC operators. The OBL method forces krill to move toward the best solutions, while CM and PC operators are well capable of adding the diversity of the population. These two operators also provide an effective balance between exploration and exploitation.

3.1 The Opposition Krill Herd Method

In the improved OKH method, the OBL idea is combined with the traditional KH in order to improve its search ability. The main steps of the OKH method can be given below.

In the first step, the first half population (*P*) is generated by a random distribution. The rest half population (*OP*) is initialized as per the first half population (*P*) in terms of OBL as shown in Section 2.2.

After initialization, for the first half population (*P*), it is subjected to update the position of the krill as shown in Section 2.1. Simultaneously, for the rest half population (*OP*), it is subjected to update the position of the krill as per the first half population (*P*) as shown in Section 2.2.

After the updating of the solutions in the population, the two subpopulations are composed of one population. This operation can make the population size unchanged in all the optimization process. Furthermore, we can sort the population in terms of their fitness and locate the best individual for this generation. The procedure is then iterated.

3.2 The Cauchy Mutation Operator

Some experiments have proven that the krill in KH will change slightly between around the global best krill before it get final best solution [57]. It is possible to improve the search ability of krill by the addition of the neighbors of the global best krill in each generation. This can also help the whole krill individuals proceed to the better positions. The implementation of a CM on the global best krill results in reaching this goal. CM has been successfully combined with many metaheuristic algorithms, such as PSO [68, 69], DE [70, 71]. The 1-*D* Cauchy density function is defined by:

$$f(x) = \frac{1}{\pi} \frac{t}{\tau^2 + x^2}, x \in R \tag{9}$$

where τ>0 is a scale parameter [68]. The Cauchy distributed function is

$$F_\tau(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{\tau}\right) \tag{10}$$

This mutation operator is well capable of decreasing the possibility of trapping into a local optimum. The description of the CM operator in OKH is given below:

$$W(j) = \left(\sum_{i=1}^{NP} x_{i,j}\right) \Big/ NP \tag{11}$$

where $x_{i,j}$ is the *j*th position vector of the *i*th krill, *NP* is the population size, *W*(*j*) is a weight vector.

$$x'(j) = x(j) + W(j) * C \tag{12}$$

where *C* is a random number drawn from a Cauchy distribution with τ=1.

3.3 The Position Clamping Operator

In order to make good balance between exploration and exploitation, the PC operator is introduced into KH method. The clamping technology has been successfully together with several metaheuristic algorithms (PSO [72]).

In basic KH, the values of positions become very large, especially for the krill distant from the best positions. Accordingly, some krill may exceed the boundaries of the search space. In order to get over this problem, positions of krill are clamped to limit the global search. If a krill *i* moves out of the allowed position, it is set to the maximum position ($X_{max,j}$). Here, $X_{max,j}$ and $X_{min,j}$ represents the maximum and minimum positions of krill in *j*th dimension. The position of krill is updated using Eq. (13).

$$X_{i,j}(t+1)=\begin{cases} X'_{i,j}(t+1), \text{if } X_{ij}(t+1)<X_{\max,j} \\ X_{\max,j}, \text{otherwise} \end{cases} \tag{13}$$

In OKH, exploration and exploitation are adjusted by adjusting the value of $X_{max,j}$, and the maximum and minimum positions are initialized by Eqs. (14) and (15).

$$X_{\max,j} = \delta_1 (X_{\max,j} - X_{\min,j}) \tag{14}$$

$$X_{\min,j} = \delta_2 (X_{\min,j} - X_{\max,j}) \tag{15}$$

where δ_1 and δ_2 are constant factors and are respectively taken 0.4 and 0.6 in our OKH solution.

By combination of KH method and OBL, CM and PC operator, the detailed steps of the OKH algorithm are as follows:

Step 1: Initialization

The first half population (P) including $NP/2$ individuals is generated by a random distribution. The rest $NP/2$ individuals (OP) is initialized as per the first half population (P) in terms of OBL as shown in Section 2.2. In the present work, the population size NP is an even number.

Step 2: Evaluation

All the individuals in the population are evaluated according to its position.

Step 3: The **KH process**

For the $NP/2$ individuals in P, their positions are updated by the three motions in KH method as described in Section 2.1. The main step of KH process can be described as follows:

for i=1:NP/2 (all krill in P) **do**
  Perform the following motion calculation.
    Motion induced by other individuals
    Foraging motion
    Physical diffusion
  Update the krill individual position in the search space.
end for i

Step 4: The **OBL process**

Corresponding to P, for the last $NP/2$ individuals in OP, the position of individual is updated by the rules of OBL as described in Section 2.2. The main step of OBL process can be described as follows:

for i= NP/2+1:NP (all krill in OP) **do**
  Compute $x^*_{NP/2+i}$ as per x_i and Eq. (8).
  if $f(x^*_{NP/2+i}) > f(x_i)$
    Update $x_{NP/2+i}$ by using $x^*_{NP/2+i}$.
  end if
end for i

Step 5: Combination

After a population (P and OP) of all the individuals are updated in this way, the P and OP are combined into one population.

Step 6: The **CM operator**

For all the individuals in population (P and OP), the CM operator is performed as shown in

Section 3.2.

Step 7: The **PC operator**

For all the individuals in population (*P* and *OP*), the PC operator is performed as shown in Section 3.3.

Step 8: Finding the best solution

Find out the best solution ever found and evaluate the average performance of the population.

Step 9: Stop or not

If the stopping criterion is satisfied, the OKH algorithm stops, and output the best solution, else return to **Step 2**.

The description of the proposed OKH process is also given in Fig. 1.

Fig. 1. Flowchart of OKH algorithm.

4. Simulation results

In this section, the OKH method is evaluated from various aspects using a series of experiments on benchmark functions (see Table 1) and two engineering optimization problems. In order to obtain fair results, all the implementations are conducted under the same conditions as discussed in [73]. More detailed descriptions of all the benchmarks can be referred as [6, 74]. Note that, without special clarification, the dimensions of function is set to 20. Population size and maximum generation are set to 50. In order to decrease the influence of the randomness, we have run 200 times for every method on each function and engineering case. The optimal solution for each test problem is bolded in each table. In addition, the scales that are used to do normalizations in the tables are fully different with each other, therefore values from different tables are independent each other. The detailed normalization process can be found in [73]. For all the tables, the total number of functions on which the metaheuristic method performs the best is given in the last row.

Table 1. Benchmark functions

4.1 The influence of OBL, CM and PC

According to the Section 3, three search strategies (OBL, CM and PC) have been combined with the basic KH method. In order to investigate the influence of each strategy and its random combination, seven different OKHs are generated as shown in Table 2. Their performance is tested through twenty-five benchmarks (see Table 1).

Table 2. Various OKHs with three search strategies.

In Table 2, "OBL", "CM", and "PC" represent "Opposition-based learning", "Cauchy mutation operator", and "Position clamping operator", respectively. "1" represents KH with this strategy, while "0" means the method including KH instead of the responding strategy. We take OKH5 as an example. OKH is the combination of KH, OBL and PC operator while not CM operator. Note that, OKH0 is essentially a basic KH method, there we use KH directly in Table 2. In addition, Gandomi and Alavi have concluded that, the KH II (KH with crossover operator) has the best performance among four types of KHs [55]. In the present work, the KH II is therefore selected as standard KH algorithm. That is to say, here OKH0 is KH II indeed. The results are recorded in Tables 3-5.

From Tables 3-5, OKH7 is well capable of searching for the best function values for most cases. This indicates the combination of KH, OBL, CM and PC operators leads to the krill move toward the best solutions. For other OKHs, OKH6 (KH+OBL+CM) is only worse than OKH7. OKH4 (KH+OBL) and OKH5 (KH+OBL+PC) have the similar performance between each other. Further, considering OKH1, OKH2 and OKH4, i.e., KH together with only one strategy, OKH4 is the best, OKH2 is the second best, and OKH1 is the third best. It has proven that, OBL has played the greatest role among three different strategies. In general cases, the ranks from the best to the worst are approximately as follows: OKH7>OKH6>OKH5>OKH4>OKH2>OKH3>OKH1>OKH0 (KH/KH II). Comprehensively considering, OKH7 is selected as the best method to compare with other methods in next experiments.

Table 3. Mean function values obtained by various OKHs.
Table 4. Best function values obtained by various OKHs.
Table 5. Worst function values obtained by various OKHs.

4.2 Comparisons of OKH7 with other methods

In order to explore the benefits of OKH, its performance was compared with seven optimization methods on twenty-five optimization problems, which are ABC [14], ACO [49], BBO [6], DE [20], GA [16], KH [55], and SGA [18]. The same parameters for KH and OKH7 are set as shown in Section 3 and above mentions. For the parameters used in the other methods, their settings are the same as [6, 75]. The obtained function values are recorded in **Tables 6-9**.

From **Table 6**, we see that, on average, OKH7 has the best performance on five out of twenty-five benchmarks. Usually, BBO and SGA have the second best performance on five functions out of twenty-five functions. For best solutions, **Table 7** shows that OKH7 is well capable of finding the optimal solutions on seventeen out of twenty-five benchmarks. SGA and BBO can search for the best solutions on five and two out of twenty-five benchmarks. For the worst function values shown in **Table 8**, the function values obtained by OKH7 are better than BBO and SGA on fourteen out of twenty-five benchmarks. Generally, other methods perform slightly differently with each other. Furthermore, standard deviation (Std) of the eight methods (see **Table 9**) indicates that, OKH7 has the smallest Std on sixteen out of twenty-five benchmarks. In other words, for most benchmarks, OKH7 is able to find the optimal solutions within smaller range than other methods. And it is therefore more practical and feasible than others. From the Tables 6-9, for most high-dimensional benchmarks, OKH7 is the best method at searching for the optimal function values.

Table 6. Mean function values.
Table 7. Best function values.
Table 8. Worst function values.
Table 9. The Std of different methods.

Furthermore, to intuitively clearly show the superiority of the OKH7, evolution curves of BBO, OKH7 and SGA on some most typical benchmarks are also provided in this section (see Figs. 2-4).

Fig. 2. Convergent curves of the F01, F04, F06 and F07 function.

From Fig. 2, for F01 function, it can be seen that, OKH7 is far better than BBO and SGA, while BBO is little better than SGA. For F04, SGA has obtained the better final value than BBO though it is worse than OKH7. For F06, though BBO, OKH7 and SGA converge to the similar value finally, the function value of OKH7 is smaller than BBO and SGA all the time. For F07, OKH7 has the better function value than BBO and SGA that have similar trend and final value.

Fig. 3. Convergent curves of the F09-F12 function.

From Fig. 3, for F09 and F12, they have the similar convergent trend. OKH7 is far better than BBO and SGA that has the similar final values. For F10 and F11, they have similar convergent trend, too. OKH7 is far better than BBO and SGA.

Fig. 4. Convergent curves of the F18-F20 and F22 function.

From Fig. 2, it is clear that OKH7 is better than BBO and SGA for F18. For F19, though BBO has a relative good initial value, the three methods converge to the similar values finally. For F20 and F22, OKH7 is better than BBO and SGA that converge to the similar values. Especially, OKH7 converge sharply about generation 20. It is indicated that the OBL, CM operator and PC operator used in OKH7 have a good influence on the performance of OKH7.

From the above Tables 6-9 and Figs. 2-4, we can conclude that our proposed OKH7 algorithm is well capable of finding the best function minimum. In general, SGA and BBO are only inferior to OKH7 among eight methods.

4.3 Comparisons using *t*-test

As per experimental results of 200 trials on each function, Table 10 presents the t values on every function of the two-tailed test with the 5% level of significance between the OKH7 and other optimization methods. In the table, the value of t with 398 degrees of freedom is significant at α=0.05 by a two-tailed test. **Boldface** means that OKH7 is better than the compared algorithm. For the last three rows, the "Better", "Equal", and "Worse" represent that OKH7 is better than, equal to, and worse than certain comparative method on certain benchmark. Here, we take the OKH7 and the BBO for instance. OKH7 performed better than BBO on thirteen functions, does as good as BBO on six functions, and is worse than the latter on six functions. In conclusion, this indicates that the OKH7 generally performs better than BBO on the solution accuracy. Furthermore, for KH, the number of "Better", "Equal", and "Worse" are 17, 3 and 5, respectively. These three numbers indicate that OKH7 has the better performance than KH on most functions. Though the OKH7 is outperformed on some functions, Table 10 still reveals that it performs better than the other seven methods for most functions.

Table 10. Comparisons between OKH7 and other methods at α = 0.05 on a two-tailed *t*-tests.

4.4 Analyze the number of Fitness evaluation

Here, in order to fully investigate the advantage of the OKH7 method, the number of fitness evaluations is also studied. We look at the number of fitness evaluations on twenty-five functions for each method to certain fixed function values. In our experiments, the fixed function values are set to *opt*+2. *opt* is the optimum for every function. The maximum of fitness evaluations is set to 50,000. That is to say, if a method is not able to find the values *opt*+2 within the maximum of fitness evaluations (50,000), it will stop, too, and return the 50,000. All the results are recorded in the Table 11. Table 11 shows that, for the sixteen functions (F01-F02, F04-F07, F12, F15-F16, and F18-F24), OKH7 can find the satisfactory results with the least fitness evaluations. SGA, DE and BBO have worse performance than OKH7, and they can converge to the fixed values with the least fitness evaluations on the six functions (F03, F08, F10-F11, F13, and F17), two functions (F09, F25) and one function (F14), respectively. In conclusion, OKH7 can find the satisfactory solution by using the least fitness evaluations on most function.

Table 11. The number of fitness evaluations for different methods.

4.5 Engineering Optimization Problems

Except the standard functions discussed in the section above, two engineering optimization problems are also used to validate the OKH7 method.

4.5.1 Test-Sheet Composition (TSC) problem

Examination has a great role in motivating teaching and student learning. Automatic TSC using computer is to find a combination of questions to satisfy constraints from item bank. In essence, TSC model is a multi-constraint multi-objective optimization problem.

In general, the error between eight constraints and evaluation requirements can be considered as objective function *f*. The mathematical model for the TSC problem is as follows:

$$\min f=\sum_{j=1}^{2}(d_j^{+}+d_j^{-})\times\omega_j+\sum_{i=1}^{h}(d_{3i}^{+}+d_{3i}^{-})\times\omega_3+\sum_{i=1}^{l}(d_{4i}^{+}+d_{4i}^{-})\times\omega_4+\sum_{i=1}^{p}(d_{5i}^{+}+d_{5i}^{-})\times\omega_5+$$
$$\sum_{i=1}^{q}(d_{6i}^{+}+d_{6i}^{-})\times\omega_6+\sum_{j=7}^{8}(d_j^{+}+d_j^{-})\times\omega_j \tag{16}$$

where, d_i^+, d_{ij}^+, d_{ij}^+, d_{ij}^+ and d_{ij}^+ are the positive deviation between the test-sheet property and constraints; d_i^-, d_{ij}^-, d_{ij}^-, d_{ij}^- and d_{ij}^- are the negative deviation between the test-sheet property and constraints. The produce of the positive deviation and negative deviation is 0, *i.e.*, $d^-\times d^+=0$; $\omega_j\,(j=1,2,\cdots,8)$ is weight for TSC problem, whose sum is 1.

In practice, the weight ω_j has an important impact on the TSC problem. In the present work, Analytic hierarchy process (AHP) [76] is used to address the weights. We get the goal weights for TSC model that are {0.3423, 0.1412, 0.0792, 0.0319, 0.2244, 0.0417, 0.0970, 0.0424} according to the AHP [77].

In OKHs, the standard continuous encoding of OKH is not proper solve TSC problem directly. In order to apply OKH to TSC problem, preprocessing and encoding should be implemented firstly. More details about AHP, preprocessing and encoding can be referred as [77]. More description of the mathematical model for TSC problem are can be referred as [77].

When using OKH to solve TSC problem, the status code for each krill individual represents a candidate solution. Krill *i* in population is evaluated by the objective function $f(X_i)$ in TSC model. Here, the eight algorithms are used to find the best combination of questions. The constraints of the TSC problem are set as shown in [77].

From Table 12, we see that OKH7 is the most effective method on the best, mean and worst values. Moreover, OKH7 has the least Std among eight methods. In sum, from Table 12, OKH7 is well capable of finding the most satisfactory solution for TSC problem. Since TSC problem is an essentially a discrete problem, therefore we can say, OKH7 is suitable for solving discrete optimization problem, too.

Table 12. Optimization results for the test-sheet composition problem.

4.5.2 Sensor Selection problem

The sensor selection problem [6] for aircraft engine health estimation can be considered as a test problem to validate the QKH method.

The Modular Aero Propulsion System Simulation (MAPSS) [6] is used as the engine simulation in sensor selection problem. Based on the analyses in [6], the object function of the health estimation problem can be expressed as

$$J = \sum_{i=4}^{13} \sqrt{\frac{\sum(i,i)}{\sum_0(i,i)}} + \frac{\alpha C}{C_0} \tag{17}$$

$\sum_0$ and C_0 are used for normalization. $\sum_0$ is the covariance, and C_0 is the cost of setting the aircraft engine with all 11 sensors. α is a scale factor that can balance the importance of financial cost and estimation accuracy.

It is clear that, the selection of sensors in order to generate the minimum for J is essentially an optimization problem. In fact, optimization methods can be used to solve the sensor selection problem. A population member contains a vector of integers, and its element is corresponding to a sensor number. Here suppose that a total of 20 sensors can be used form unique 11 sensors, and each sensor is only selected less than or equal to four times. The number of possible sensor sets is equal to the coefficient of x^{20} as

$$q(x) = (1 + x + x^2 + x^3 + x^4)^{11} \tag{18}$$

For x^{20} in Eq. (18), its coefficient is 3 755 070. In order to minimize J in Eq. **(17)**, the total number of sensor sets must be searched. $\sum$ for a sensor set should be solved to compute J for a single sensor set. In order to solve for $\sum$, if a discrete algebraic Riccati equation (DARE) [6] is used, this brute-force search would consume 21 hours [6].

Instead of a brute-force 21-h search, eight optimization methods are used to search for a sub-optimal sensor set. The results on the sensor selection problem are recorded in Table 13. We see that OKH7 performs the best in terms of both average performance and best performance. For worst function values, SGA performs the best among different methods, and OKH7 is only inferior to SGA. Furthermore, the Std of OKH7 is much smaller than other methods except SGA. To sum up, OKH7 has a relatively high possibility of finding the satisfactory sensor set.

Table 13. Optimization results for the Sensor Selection problem.

5. Conclusion

This paper presented various OKH methods for solving the continuous and discrete optimization problems. In OKH, first of all, a half population of candidate solutions is randomly initialized. And then, the rest half population is initialized as per the first half population based on the OBL theory. After initialization, new solutions are created by applying the KH and OBL process. By simultaneous consideration of the krill in the KH process and OBL process, OBL can provide a higher chance of finding solutions which are closer to the global optimum. The two subpopulations are combined into one population. In order to add the diversity of the krill population and balance the exploration and exploitation, the CM and (PC operators are introduced into the OKH method. After the implementation of the CM and PC operators, the fitness of the resulting solutions are evaluated and find the best solution for this generation. Several experiments are carried out on an array of well-known benchmark problems and two engineering optimization problems and the related results are compared with seven other methods. The experimental results verify that the proposed OKH model is a more efficient and practical method in solving optimization problems.

In addition, in order to verify the influence of the OBL, CM and PC strategies, KH is combined with different combinations of them. The results show that KH combined with the OBL, CM and PC operators, i.e., OKH7, has the best performance among different variants of OKH.

In future, our research highlights would be focused on the following points. Firstly, the influence of the OKH parameters on convergence and performance would be carefully analyzed and investigated. Secondly, from Table 3-5, we claimed that OKH7 is the best performer compared with other seven OKHs. But it is obvious that the computational complexity of OKH7 is the most high. Because it has more strategies than other OKHs, which can be seen from Table 2. Thus, OKH7 will spend more CPU time than other OKHS, in order to find optimal solutions. It should be noted that CPU time is a significant factor to the implementation of most optimization methods. The time used by OKH should be studied. Based on this time study, how to make the OKH consume less time is still worthy of further study. Thirdly, through OKH, we can see, an improved method significantly outperforms the original methods. In our future studies, the KH will be combined with other search strategies. Fourthly, we would use more benchmarks to test our method, such as constrained optimization and complex engineering optimization problems [78, 79]. Finally, future analysis of our OKH using dynamic system and Markov chain can give us theoretical insight into their advantages and disadvantages.

Acknowledgements

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

References

[1] X.S. Yang, S. Deb, Cuckoo search via Lévy flights, in: A. Abraham, A. Carvalho, F. Herrera, V. Pai (Eds.) Proceeding of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), IEEE Publications, Coimbatore, India, 2009, pp. 210-214.

[2] G.-G. Wang, A.H. Gandomi, X. Zhao, H.E. Chu, Hybridizing harmony search algorithm with cuckoo search for global numerical optimization, Soft Comput. (2014).

[3] G.-G. Wang, S. Deb, A.H. Gandomi, Z. Zhang, A.H. Alavi, Chaotic cuckoo search, Soft Comput. (2015).

[4] G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, J. Wang, A hybrid meta-heuristic DE/CS algorithm for UCAV path planning, J. Inform. Comput. Sci. 9 (2012) 4811-4818.

[5] X. Li, J. Wang, M. Yin, Enhancing the performance of cuckoo search algorithm using orthogonal learning method, Neural Comput. Appl. 24 (2013) 1233-1247.

[6] D. Simon, Biogeography-based optimization, IEEE Trans. Evolut. Comput. 12 (2008) 702-713.

[7] G.-G. Wang, A.H. Gandomi, A.H. Alavi, An effective krill herd algorithm with migration operator in biogeography-based optimization, Appl. Math. Model. 38 (2014) 2454-2462.

[8] S. Saremi, S. Mirjalili, A. Lewis, Biogeography-based optimisation with chaos, Neural Comput. Appl. 25 (2014) 1077-1097.

[9] S. Mirjalili, S.M. Mirjalili, A. Lewis, Let a biogeography-based optimizer train your Multi-Layer Perceptron, Inf. Sci. 269 (2014) 188-209.

[10] G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, Dynamic deployment of wireless sensor networks by biogeography based optimization algorithm, J. Sens. Actuator Netw. 1 (2012) 86-96.

[11] X. Li, M. Yin, Multi-operator based biogeography based optimization with mutation for global numerical optimization, Comput. Math. Appl. 64 (2012) 2833-2844.

[12] X. Li, J. Wang, J. Zhou, M. Yin, A perturb biogeography based optimization with mutation for global numerical optimization, Appl. Math. Comput. 218 (2011) 598-609.

[13] X. Li, M. Yin, Multiobjective binary biogeography based optimization for feature selection using gene expression data, IEEE Trans. Nanobiosci. 12 (2013) 343-353.

[14] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global. Optim. 39 (2007) 459-471.

[15] X. Li, M. Yin, Self-adaptive constrained artificial bee colony for constrained numerical optimization, Neural Comput. Appl. 24 (2012) 723-734.

[16] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine learning, Addison-Wesley, New York, 1998.

[17] A.H. Gandomi, A.H. Alavi, Multi-stage genetic programming: A new strategy to nonlinear system modeling, Inf. Sci. 181 (2011) 5227-5239.

[18] 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.

[19] G.-G. Wang, A.H. Gandomi, A.H. Alavi, Stud krill herd algorithm, Neurocomputing 128 (2014) 363-370.

[20] R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341-359.

[21] 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. 25 (2014) 297-308.

[22] G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, M. Shao, Path planning for uninhabited combat aerial vehicle using hybrid meta-heuristic DE/BBO algorithm, Adv. Sci. Eng. Med. 4 (2012) 550-564.

[23] X. Li, M. Yin, Parameter estimation for chaotic systems by hybrid differential evolution algorithm and artificial bee colony algorithm, Nonlinear Dynam. 77 (2014) 61-71.

[24] X.Z. Gao, X. Wang, S.J. Ovaska, Fusion of clonal selection algorithm and differential evolution method in training cascade–correlation neural network, Neurocomputing 72 (2009) 2483-2490.

[25] D. Zou, H. Liu, L. Gao, S. Li, An improved differential evolution algorithm for the task assignment problem, Eng. Appl. Artif. Intel 24 (2011) 616-624.

[26] S. Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80-98.

[27] X. Meng, Y. Liu, X. Gao, H. Zhang, A new bio-inspired algorithm: chicken swarm optimization, in: Y. Tan, Y. Shi, C.C. Coello (Eds.) Advances in Swarm Intelligence, Springer International Publishing, 2014, pp. 86-94.

[28] S. Fong, S. Deb, X.-S. Yang, A heuristic optimization method inspired by wolf preying behavior, Neural Comput. Appl. (2015).

[29] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. (2015).

[30] G.-G. Wang, S. Deb, L.d.S. Coelho, Earthworm optimization algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, Int. J. of Bio-Inspired Computation (2015).

[31] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46-61.

[32] S. Saremi, S.Z. Mirjalili, S.M. Mirjalili, Evolutionary population dynamics and grey wolf optimizer, Neural Comput. Appl. 26 (2014) 1257-1263.

[33] A.H. Gandomi, X.-S. Yang, A.H. Alavi, Mixed variable structural optimization using firefly algorithm, Comput. Struct. 89 (2011) 2325-2336.

[34] X.S. Yang, Firefly algorithm, stochastic test functions and design optimisation, Int. J. of Bio-Inspired Computation 2 (2010) 78-84.

[35] G.-G. Wang, L. Guo, H. Duan, H. Wang, A new improved firefly algorithm for global numerical optimization, J. Comput. Theor. Nanos. 11 (2014) 477-485.

[36] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. (2015).

[37] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2001) 60-68.

[38] G. Wang, L. Guo, H. Duan, H. Wang, L. Liu, M. Shao, Hybridizing harmony search with biogeography based optimization for global numerical optimization, J. Comput. Theor. Nanos. 10 (2013) 2318-2328.

[39] D. Zou, L. Gao, S. Li, J. Wu, Solving 0-1 knapsack problem by a novel global harmony search algorithm, Appl. Soft Comput. 11 (2011) 1556-1564.

[40] D. Zou, L. Gao, J. Wu, S. Li, Y. Li, A novel global harmony search algorithm for reliability problems, Comput. Ind. Eng. 58 (2010) 307-316.

[41] D. Zou, L. Gao, S. Li, J. Wu, X. Wang, A novel global harmony search algorithm for task assignment problem, J. Syst. Software. 83 (2010) 1678-1688.

[42] X.-B. Meng, X.Z. Gao, L. Lu, Y. Liu, H. Zhang, A new bio-inspired optimisation algorithm: bird swarm algorithm, J. Exp. Theor. Artif. In (2015) 1-15.

[43] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic

paradigm, Knowl-based. Syst. (2015).

[44] X. Li, J. Zhang, M. Yin, Animal migration optimization: an optimization algorithm inspired by animal migration behavior, Neural Comput. Appl. 24 (2014) 1867-1877.

[45] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceeding of the IEEE International Conference on Neural Networks, IEEE, Perth, Australia, 1995, pp. 1942-1948.

[46] G.-G. Wang, A.H. Gandomi, X.-S. Yang, A.H. Alavi, A novel improved accelerated particle swarm optimization algorithm for global numerical optimization, Eng. Computation. 31 (2014) 1198-1220.

[47] S. Mirjalili, G.-G. Wang, L.d.S. Coelho, Binary optimization using hybrid particle swarm optimization and gravitational search algorithm, Neural Comput. Appl. 25 (2014) 1423-1435.

[48] J.j. Zhao, G.h. Ji, Y. Xia, X.l. Zhang, Cavitary nodule segmentation in computed tomography images based on self–generating neural networks and particle swarm optimisation, Int. J. of Bio-Inspired Computation 7 (2015) 62-67.

[49] M. Dorigo, T. Stutzle, Ant Colony Optimization, MIT Press, Cambridge, 2004.

[50] A.H. Gandomi, X.-S. Yang, A.H. Alavi, S. Talatahari, Bat algorithm for constrained optimization tasks, Neural. Comput. Appl. 22 (2013) 1239-1255.

[51] X.S. Yang, A.H. Gandomi, Bat algorithm: a novel approach for global engineering optimization, Eng. Computation. 29 (2012) 464-483.

[52] S. Mirjalili, S.M. Mirjalili, X.-S. Yang, Binary bat algorithm, Neural Comput. Appl. 25 (2013) 663-681.

[53] F. Xue, Y. Cai, Y. Cao, Z. Cui, F. Li, Optimal parameter settings for bat algorithm, Int. J. of Bio-Inspired Computation 7 (2015) 125-128.

[54] X. Cai, L. Wang, Q. Kang, Q. Wu, Bat algorithm with Gaussian walk, Int. J. of Bio-Inspired Computation 6 (2014) 166-174.

[55] A.H. Gandomi, A.H. Alavi, Krill herd: a new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simulat. 17 (2012) 4831-4845.

[56] G.-G. Wang, A.H. Gandomi, A.H. Alavi, S. Deb, A hybrid method based on krill herd and quantum-behaved particle swarm optimization, Neural Comput. Appl. (2015).

[57] G. Wang, L. Guo, H. Wang, H. Duan, L. Liu, J. Li, Incorporating mutation scheme into krill herd algorithm for global numerical optimization, Neural Comput. Appl. 24 (2014) 853-871.

[58] H.R. Tizhoosh, Opposition-based learning: A new scheme for machine intelligence, International Conference on Computational Intelligence for Modelling, Control and Automation, IEEE, Vienna, 2005, pp. 695-701.

[59] X.Z. Gao, X. Wang, S.J. Ovaska, K. Zenger, A hybrid optimization method of harmony search and opposition-based learning, Eng. Optimiz. 44 (2011) 895-914.

[60] A.R. Malisia, H.R. Tizhoosh, Applying Opposition-Based Ideas to the Ant Colony System, IEEE Swarm Intelligence Symposium 2007 (SIS 2007), IEEE, Honolulu, HI, 2007, pp. 182-189.

[61] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama, Opposition-based differential evolution,

IEEE Trans. Evolut. Comput. 12 (2008) 64-79.

[62] 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.

[63] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama, Quasi-oppositional Differential Evolution, IEEE Congress on Evolutionary Computation 2007 (CEC 2007), IEEE, Singapore, 2007, pp. 2229-2236.

[64] M. Ventresca, H.R. Tizhoosh, A diversity maintaining population-based incremental learning algorithm, Inf. Sci. 178 (2008) 4038-4056.

[65] X.Z. Gao, X. Wang, T. Jokinen, S.J. Ovaska, A. Arkkio, K. Zenger, A hybrid PBIL-based harmony search method, Neural Comput. Appl. 21 (2012) 1071-1083.

[66] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, M. Ventresca, Enhancing particle swarm optimization using generalized opposition-based learning, Inf. Sci. 181 (2011) 4699-4714.

[67] A. Rajasekhar, R. Kumar Jatoth, A. Abraham, Design of intelligent PID/PIλDμ speed controller for chopper fed DC motor drive using opposition based artificial bee colony algorithm, Eng. Appl. Artif. Intel 29 (2014) 13-32.

[68] H. Wang, H. Li, Y. Liu, C. Li, S. Zeng, Opposition-based particle swarm algorithm with Cauchy mutation, IEEE Congress on Evolutionary Computation 2007 (CEC 2007), IEEE2007, pp. 4750-4756.

[69] Q. Wu, Hybrid forecasting model based on support vector machine and particle swarm optimization with adaptive and Cauchy mutation, Expert. Syst. Appl. 38 (2011) 9070-9075.

[70] H. Qin, J. Zhou, Y. Lu, Y. Wang, Y. Zhang, Multi-objective differential evolution with adaptive Cauchy mutation for short-term multi-objective optimal hydro-thermal scheduling, Energ. Convers. Manage 51 (2010) 788-794.

[71] M. Ali, M. Pant, Improving the performance of differential evolution algorithm using Cauchy mutation, Soft Comput. 15 (2010) 991-1007.

[72] F. Shahzad, A.R. Baig, S. Masood, M. Kamran, N. Naveed, Opposition-Based Particle Swarm Optimization with Velocity Clamping (OVCPSO), in: W. Yu, E. Sanchez (Eds.) Advances in Computational Intelligence, Springer Berlin Heidelberg, 2009, pp. 339-348.

[73] G.-G. Wang, L. Guo, A.H. Gandomi, G.-S. Hao, H. Wang, Chaotic krill herd algorithm, Inf. Sci. 274 (2014) 17-34.

[74] X.-S. Yang, Z. Cui, R. Xiao, A.H. Gandomi, M. Karamanoglu, Swarm Intelligence and Bio-Inspired Computation, Elsevier, Waltham, MA, 2013.

[75] L. Guo, G.-G. Wang, A.H. Gandomi, A.H. Alavi, H. Duan, A new improved krill herd algorithm for global numerical optimization, Neurocomputing 138 (2014) 392-402.

[76] T.L. Saaty, The analytic hierarchy process: planning, priority setting, resource allocation, McGraw-Hill, New York, 1980.

[77] H. Duan, W. Zhao, G. Wang, X. Feng, Test-sheet composition using analytic hierarchy process and hybrid metaheuristic algorithm TS/BBO, Math. Probl. Eng. 2012 (2012) 1-22.

[78] X.S. Yang, A.H. Gandomi, S. Talatahari, A.H. Alavi, Metaheuristics in Water,

Geotechnical and Transport Engineering, Elsevier, Waltham, MA, 2013.

[79] A.H. Gandomi, X.S. Yang, S. Talatahari, A.H. Alavi, Metaheuristic Applications in Structures and Infrastructures, Elsevier, Waltham, MA, 2013.

LIST OF TABLES

Table 1. Benchmark functions.

No.	*Name*	*No.*	*Name*
F01	Ackley	F14	Quartic with noise
F02	Alpine	F15	Rastrigin
F03	Brown	F16	Rosenbrock
F04	Dixon & Price	F17	Schwefel 2.26
F05	Fletcher-Powell	F18	Schwefel 1.2
F06	Griewank	F19	Schwefel 2.22
F07	Holzman 2 function	F20	Schwefel 2.21
F08	Levy	F21	Sphere
F09	Pathological function	F22	Step
F10	Penalty #1	F23	Sum function
F11	Penalty #2	F24	Zakharov
F12	Perm #1	F25	Wavy1
F13	Powel		

Table 2. Various OKHs with three search strategies.

	OBL	CM	PC
KH	0	0	0
OKH1	0	0	1
OKH2	0	1	0
OKH3	0	1	1
OKH4	1	0	0
OKH5	1	0	1
OKH6	1	1	0
OKH7	1	1	1

Table 3. Mean function values obtained by various OKHs.

	KH	OKH1	OKH2	OKH3	OKH4	OKH5	OKH6	OKH7
F01	702.31	289.64	1.24	290.34	1.88	150.43	1.24	1.92
F02	2.4E4	67.34	1.81	52.59	1.32	4.7E3	1.81	2.36
F03	221.67	6.61	11.36	855.44	5.93	**1.00**	11.36	8.6E4
F04	2.5E3	2.18	153.86	1.53	**1.00**	2.0E3	153.86	10.70
F05	3.82	10.84	11.84	19.17	9.83	**1.00**	11.84	17.52
F06	6.21	1.02	**1.00**	1.02	**1.00**	2.16	**1.00**	**1.00**
F07	4.8E16	5.0E11	18.68	6.0E11	1.39	3.1E16	18.68	**1.00**
F08	10.78	**1.00**	13.83	1.04	2.80	4.13	13.83	21.71
F09	433.00	357.77	104.61	348.09	31.85	42.65	104.61	**1.00**
F10	1.6E5	**1.00**	27.31	1.25	2.71	2.8E6	27.31	18.43
F11	8.7E5	7.76	**1.00**	8.57	4.06	1.5E6	**1.00**	3.88
F12	6.9E17	6.6E18	684.60	1.1E19	4.4E18	9.1E13	684.60	**1.00**
F13	339.01	**1.00**	348.30	1.55	94.49	87.95	348.30	195.49
F14	8.2E14	7.9E9	**1.00**	6.0E9	1.07	4.4E14	**1.00**	3.11
F15	4.82	2.86	6.36	2.86	2.37	**1.00**	6.36	3.19
F16	6.88	1.01	5.78	1.01	**1.00**	1.45	5.78	**1.00**
F17	5.21	9.47	9.52	10.22	9.66	**1.00**	9.52	10.09
F18	2.9E8	3.8E6	**1.00**	3.3E6	1.01	8.6E7	**1.00**	10.39
F19	1.8E3	188.69	743.56	273.47	1.16	523.59	743.56	281.44
F20	1.1E3	244.80	1.94	2.3E3	**1.00**	207.91	1.94	2.17
F21	5.0E7	1.8E5	3.68	2.5E5	3.93	2.0E7	3.68	**1.00**
F22	2.5E18	8.5E16	**1.00**	7.8E16	**1.00**	1.1E18	**1.00**	**1.00**
F23	1.1E8	8.6E5	4.74	4.5E5	**1.00**	3.5E7	4.74	1.46
F24	3.33	58.26	7.15	9.0E5	9.10	**1.00**	7.15	1.0E5
F25	3.83	7.02	10.42	6.64	6.97	**1.00**	10.42	11.09
Total	0	3	5	0	6	6	5	7

Table 4. Best function values obtained by various OKHs.

	KH	OKH1	OKH2	OKH3	OKH4	OKH5	OKH6	OKH7
F01	4.7E4	6.2E3	16.40	1.1E4	23.36	**1.00**	1.1E4	22.00
F02	3.8E6	8.5E3	4.56	6.6E3	**1.00**	12.92	7.2E5	83.95
F03	94.08	5.33	8.72	983.94	5.67	1.8E3	**1.00**	3.4E3
F04	1.4E3	1.21	77.52	**1.00**	1.11	1.09	1.1E3	1.14
F05	5.63	12.32	10.31	27.31	9.76	24.78	**1.00**	23.45
F06	5.57	1.39	1.38	1.39	1.38	1.38	**1.00**	1.38
F07	7.2E17	9.2E12	**1.00**	4.4E12	**1.00**	**1.00**	5.0E17	**1.00**
F08	15.24	**1.00**	6.24	1.13	4.75	5.05	6.88	6.88
F09	1.3E9	1.1E9	111.84	1.0E9	**1.00**	15.60	1.2E8	64.86
F10	421.05	1.46	26.78	**1.00**	5.50	3.62	1.4E6	13.83
F11	2.5E5	7.56	**1.00**	7.84	9.11	9.31	1.0E6	7.11
F12	1.3E13	1.0E19	**1.00**	4.1E25	2.3E16	4.1E25	7.0E14	123.50
F13	919.40	**1.00**	1.4E3	2.66	551.74	562.23	185.86	1.2E3
F14	4.4E14	1.6E9	**1.00**	2.8E8	**1.00**	**1.00**	1.3E14	**1.00**
F15	5.6E7	3.7E7	4.24	3.6E7	2.6E7	3.0E7	1.3E7	**1.00**
F16	4.55	**1.00**	1.01	1.02	1.01	1.01	1.16	1.02
F17	4.79	9.09	9.53	9.27	8.98	10.79	**1.00**	9.53
F18	2.7E13	9.9E11	242.21	6.7E11	11.45	1.3E3	1.7E13	**1.00**
F19	1.6E6	1.9E5	28.23	2.8E5	32.15	50.75	4.1E5	**1.00**
F20	1.2E4	2.7E3	3.45	6.0E4	**1.00**	5.9E4	3.4E3	5.35
F21	1.1E15	9.1E11	5.4E3	3.0E12	322.74	1.5E4	4.0E14	**1.00**
F22	1.6E18	2.7E16	**1.00**	2.3E16	**1.00**	**1.00**	7.3E17	**1.00**
F23	8.6E14	4.1E12	6.0E3	3.4E12	6.2E4	2.8E4	3.4E14	**1.00**
F24	4.16	6.22	6.11	11.11	8.86	19.03	**1.00**	11.27
F25	3.48	6.81	8.73	6.01	5.68	6.27	**1.00**	10.75
Total	0	3	5	2	6	4	6	8

Table 5. Worst function values obtained by various OKHs.

	KH	OKH1	OKH2	OKH3	OKH4	OKH5	OKH6	OKH7
F01	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F02	1.02	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F03	**1.00**	1.36	1.36	9.90	1.36	24.96	1.36	40.55
F04	1.2E3	6.36	70.42	**1.00**	6.36	6.36	1.1E3	6.36
F05	8.32	13.60	31.83	26.09	17.67	22.55	**1.00**	21.20
F06	11.82	1.40	9.57	1.40	9.57	9.57	**1.00**	1.38
F07	1.3E18	9.2E12	3.1E16	6.0E13	3.1E16	**1.00**	8.8E17	**1.00**
F08	7.90	1.08	9.05	**1.00**	6.50	1.98	3.53	2.90
F09	1.14	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F10	616.52	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	2.5E4	**1.00**
F11	1.5E5	1.39	1.39	**1.00**	1.39	1.39	9.6E4	1.39
F12	2.1E18	1.5E20	1.1E7	1.7E23	1.4E25	3.8E25	9.2E16	**1.00**
F13	376.37	1.88	1.4E3	**1.00**	353.74	292.98	298.71	490.68
F14	8.6E14	8.6E9	3.1E16	9.2E8	3.1E16	3.1E16	5.7E14	**1.00**
F15	2.9E6	1.7E6	4.3E6	1.8E6	1.4E6	1.2E6	6.2E5	**1.00**
F16	4.47	1.01	**1.00**	1.02	**1.00**	**1.00**	1.45	**1.00**
F17	4.74	9.00	9.42	9.17	11.17	11.12	**1.00**	9.63
F18	146.93	1.04	**1.00**	**1.00**	**1.00**	**1.00**	33.72	**1.00**
F19	5.74	1.55	1.55	**1.00**	1.55	1.55	1.73	1.55
F20	6.59	3.30	3.30	9.56	3.30	9.46	**1.00**	3.30
F21	1.2E3	**1.00**	1.2E5	9.47	1.2E5	1.2E5	440.05	1.2E5
F22	2.5E18	8.1E16	5.7E17	2.3E16	5.7E17	5.7E17	8.4E17	**1.00**
F23	2.6E9	1.4E7	2.3E8	5.6E6	2.3E8	**1.00**	8.2E8	96.38
F24	3.48	11.09	5.01	1.3E6	9.22	4.0E3	**1.00**	62.50
F25	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Total	3	6	7	11	7	9	9	12

Table 6. Mean function values.

	ABC	ACO	BBO	DE	GA	KH	OKH7	SGA
F01	1.9E3	2.1E3	1.1E3	1.6E3	2.3E3	587.93	**1.00**	1.2E3
F02	3.17	5.37	**1.00**	6.00	4.69	3.43	1.08	1.23
F03	10.04	14.83	1.31	2.52	6.76	12.13	3.1E5	**1.00**
F04	8.3E3	1.2E4	388.89	2.2E3	5.3E3	782.05	**1.00**	180.68
F05	2.40	8.74	**1.00**	3.12	3.58	3.65	14.95	1.15
F06	41.55	12.82	10.52	21.82	37.95	6.45	**1.00**	8.75
F07	6.8E15	7.6E15	4.6E14	2.1E15	4.5E15	6.6E14	**1.00**	1.9E14
F08	5.27	10.65	1.28	7.53	10.04	3.86	7.79	**1.00**
F09	4.49	4.98	3.83	2.62	3.78	5.49	**1.00**	3.53
F10	7.0E5	5.5E7	1.8E4	1.4E5	2.5E5	1.2E4	**1.00**	51.32
F11	6.2E6	1.0E8	8.7E4	1.7E6	3.1E6	2.0E5	**1.00**	8.4E3
F12	5.0E19	5.0E19	2.5E20	5.0E14	9.3E18	2.5E19	**1.00**	4.8E18
F13	6.16	29.53	1.81	13.41	6.51	5.14	2.95	**1.00**
F14	5.7E15	4.8E15	4.3E14	2.2E15	6.8E15	6.2E14	**1.00**	1.5E14
F15	2.49	4.40	**1.00**	3.93	4.25	2.42	1.20	1.36
F16	3.05	17.11	**1.00**	2.64	4.83	1.27	1.37	1.18
F17	3.19	2.16	1.10	4.07	1.72	3.75	7.49	**1.00**
F18	1.7E8	1.7E8	9.9E7	2.6E8	2.0E8	1.2E8	**1.00**	1.5E8
F19	3.93	9.89	1.44	4.53	8.09	5.88	**1.00**	2.11
F20	1.4E3	967.18	1.0E3	1.2E3	1.3E3	244.53	**1.00**	913.96
F21	1.1E7	3.5E7	2.6E6	6.3E6	2.0E7	1.9E6	**1.00**	2.6E6
F22	2.1E19	1.0E19	4.5E18	1.1E19	2.2E19	2.9E18	**1.00**	3.5E18
F23	8.4E7	1.8E8	1.5E7	3.9E7	8.5E7	1.6E7	**1.00**	1.5E7
F24	1.74	1.24	**1.00**	2.03	2.13	1.24	7.2E4	1.50
F25	2.15	1.71	1.01	2.33	2.78	2.17	5.89	**1.00**
Total	0	0	5	0	0	0	15	5

Table 7. Best function values.

	ABC	ACO	BBO	DE	GA	KH	OKH7	SGA
F01	2.1E5	2.2E5	1.3E5	2.0E5	2.5E5	5.2E4	**1.00**	1.2E5
F02	2.3E6	3.0E6	4.3E5	3.9E6	2.6E6	2.3E6	**1.00**	7.5E5
F03	5.53	26.35	**1.00**	4.64	8.94	16.49	225.31	1.45
F04	4.0E3	9.5E3	214.44	3.5E3	1.9E3	654.57	**1.00**	148.45
F05	4.58	15.00	1.39	5.37	6.07	5.24	31.78	**1.00**
F06	17.75	6.38	6.03	15.33	19.58	3.81	**1.00**	4.11
F07	1.1E18	9.9E18	1.8E17	2.5E18	2.2E18	1.0E18	**1.00**	8.3E16
F08	8.64	12.71	1.22	14.21	7.46	4.27	3.09	**1.00**
F09	2.2E10	2.6E10	1.9E10	9.9E9	1.7E10	2.9E10	**1.00**	1.8E10
F10	1.1E4	**1.00**	17.65	2.0E4	31.02	114.48	1.23	2.91
F11	7.6E5	10.24	14.65	3.4E5	1.1E5	3.6E4	**1.00**	13.43
F12	5.4E24	6.8E26	6.8E26	5.4E16	6.8E26	1.0E15	**1.00**	6.8E26
F13	7.56	64.75	3.66	26.40	6.21	12.98	8.42	**1.00**
F14	6.4E14	1.2E15	2.5E13	1.1E15	1.6E15	3.6E14	**1.00**	2.0E13
F15	2.9E13	5.1E13	9.3E12	4.6E13	4.8E13	2.5E13	**1.00**	1.4E13
F16	8.04	57.30	2.71	8.26	11.62	4.29	**1.00**	2.52
F17	4.69	2.99	1.37	6.10	1.52	4.96	11.86	**1.00**
F18	6.9E15	5.4E15	4.1E15	1.4E16	8.0E15	6.0E15	**1.00**	4.9E15
F19	1.6E4	2.7E4	5.6E3	2.0E4	2.3E4	2.2E4	**1.00**	7.5E3
F20	3.0E4	1.7E4	2.1E4	2.9E4	2.5E4	3.5E3	**1.00**	1.6E4
F21	1.2E14	5.6E14	4.6E13	8.1E13	3.0E14	2.5E13	**1.00**	3.6E13
F22	6.3E18	4.8E18	2.0E18	6.7E18	6.1E18	1.4E18	**1.00**	1.3E18
F23	2.3E12	3.3E12	3.1E11	1.2E12	1.2E12	4.8E11	**1.00**	3.7E11
F24	2.96	1.20	**1.00**	3.04	2.32	1.52	6.40	1.60
F25	3.63	2.04	1.19	3.91	4.40	2.32	9.66	**1.00**
Total	0	1	2	0	0	0	17	5

Table 8. Worst function values.

	ABC	ACO	BBO	DE	GA	KH	OKH7	SGA
F01	**1.00**	**1.00**	**1.00**	**1.00**	1.01	**1.00**	**1.00**	**1.00**
F02	**1.00**	2.46	2.15	2.15	2.15	2.15	3.24	2.15
F03	**1.00**	3.61	2.14	2.14	2.14	2.14	5.4E3	2.14
F04	1.4E3	786.63	53.91	395.05	1.5E3	119.73	**1.00**	57.46
F05	1.28	6.83	1.12	3.25	4.72	3.71	10.95	**1.00**
F06	5.32	1.08	**1.00**	2.45	6.08	1.68	1.68	1.68
F07	122.29	167.42	4.39	100.14	90.58	37.08	**1.00**	13.78
F08	3.94	3.86	**1.00**	6.88	9.27	1.92	5.97	4.96
F09	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F10	2.3E5	3.0E3	407.50	7.1E3	7.1E4	5.5E3	**1.00**	**1.00**
F11	8.1E4	1.3E4	389.53	2.7E5	1.5E4	9.7E3	**1.00**	1.80
F12	5.4E21	8.5E22	2.8E21	5.5E16	6.4E20	1.4E21	**1.00**	6.4E20
F13	6.85	15.06	3.11	18.07	1.98	7.60	7.46	**1.00**
F14	11.19	13.08	**1.00**	3.61	13.21	105.62	105.62	105.62
F15	3.51	5.71	1.11	5.85	6.19	3.88	**1.00**	1.41
F16	8.02	87.59	2.70	8.23	55.23	12.18	**1.00**	7.02
F17	3.55	1.91	**1.00**	4.45	2.00	3.24	7.68	1.46
F18	333.46	273.78	121.74	454.71	227.65	236.25	**1.00**	211.91
F19	1.18	2.09	**1.00**	1.28	1.51	2.46	**1.00**	**1.00**
F20	4.01	1.63	2.00	3.12	4.00	**1.00**	**1.00**	1.56
F21	2.10	12.50	**1.00**	1.92	6.35	6.35	6.35	6.35
F22	310.78	188.02	87.62	105.13	467.56	34.20	**1.00**	51.46
F23	10.12	13.61	**1.00**	4.12	4.16	1.07	607.08	607.08
F24	2.06	1.44	**1.00**	2.11	3.53	1.08	3.75	1.55
F25	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Total	5	3	11	3	2	4	14	7

Table 9. The Std of different methods.

	ABC	ACO	BBO	DE	GA	KH	OKH7	SGA
F01	1.72	0.53	1.33	0.22	0.57	0.90	**0.01**	1.35
F02	1.58	1.54	0.84	0.75	1.40	6.36	**0.70**	10.01
F03	13.13	4.61	0.63	**0.30**	1.21	1.64	25.03	1.2E5
F04	1.6E4	2.8E4	699.79	**5.66**	2.4E3	5.5E3	1.8E3	26.91
F05	6.9E4	1.5E5	3.0E4	7.3E4	4.6E4	7.8E4	**2.9E4**	2.7E5
F06	19.95	8.21	1.95	0.11	5.38	20.18	**0.21**	0.25
F07	1.4E3	6.7E3	114.70	0.04	591.14	1.8E3	**0.01**	0.23
F08	4.28	2.02	1.28	**0.21**	3.23	7.03	0.95	27.99
F09	0.39	0.33	0.33	0.59	0.59	0.43	**0.31**	3.64
F10	1.2E6	1.0E8	2.3E3	**0.81**	6.9E4	1.5E5	3.9E4	3.68
F11	8.2E6	2.6E8	4.4E5	1.1E2	2.2E6	2.5E6	**0.08**	0.09
F12	1.2E50	5.1E49	7.3E50	2.8E47	1.0E44	5.1E49	6.1E32	**3.0E31**
F13	172.31	706.79	64.09	53.79	208.93	493.27	**40.13**	45.26
F14	0.48	0.33	0.04	16.09	0.19	0.77	**0.02**	0.03
F15	7.34	17.48	4.77	14.08	8.50	20.14	**0.19**	80.93
F16	79.47	469.31	52.07	**8.13**	63.31	332.45	51.18	94.50
F17	311.40	351.13	229.21	543.75	559.22	263.88	**52.15**	306.03
F18	1.4E3	2.1E3	3.3E3	1.1E3	2.1E3	2.1E3	**4.12**	4.61
F19	3.42	10.95	**0.98**	15.66	3.78	9.92	1.81	3.61
F20	7.22	7.97	16.73	0.11	4.89	9.92	4.03	**0.01**
F21	3.00	5.93	0.48	0.04	1.19	2.63	**0.28**	0.32
F22	1.9E3	486.19	321.41	0.02	538.12	2.7E3	**0.15**	0.51
F23	127.10	308.55	43.73	0.05	31.23	145.77	**0.36**	0.78
F24	42.71	3.5E7	15.01	38.11	47.85	40.68	**1.29**	2.3E7
F25	63.05	**22.41**	28.03	33.26	46.44	118.97	27.25	60.03
Total	0	1	1	5	0	0	16	2

Table 10. Comparisons between OKH7 and other methods at $\alpha = 0.05$ on a two-tailed *t*-tests.

	ABC	ACO	BBO	DE	GA	KH	SGA
F01	**41.51**	**42.09**	**39.77**	**79.05**	**50.41**	**21.26**	**29.23**
F02	**3.37**	**6.39**	-0.12	**7.68**	**5.40**	**3.70**	0.25
F03	-1.02	-1.02	-1.02	-1.02	-1.02	-1.02	-1.02
F04	**6.99**	**6.80**	**6.47**	**12.57**	**6.22**	**11.38**	**8.02**
F05	-17.82	-6.54	-20.02	-16.77	-15.62	-14.90	-19.73
F06	**12.67**	**11.48**	**12.35**	**19.79**	**10.82**	**11.11**	**10.01**
F07	**6.20**	**11.50**	**5.39**	**10.82**	**5.51**	**9.35**	**7.05**
F08	-1.31	1.37	-3.42	-0.13	1.03	-2.05	-3.57
F09	**8.22**	**9.41**	**6.67**	**3.76**	**6.44**	**10.62**	**5.95**
F10	**3.02**	**4.00**	1.45	**3.95**	**2.27**	**4.73**	1.06
F11	**4.61**	**3.30**	**3.91**	**7.48**	**4.05**	**5.15**	1.70
F12	**2.64**	1.84	**2.93**	**2.53**	**2.40**	1.58	**6.97**
F13	**4.96**	**11.90**	-2.80	**8.71**	**3.30**	**3.62**	-5.06
F14	**6.97**	**7.24**	**4.69**	**12.00**	**6.24**	**10.92**	**5.87**
F15	**3.03**	**7.36**	-0.47	**6.40**	**6.84**	**2.82**	0.38
F16	**3.48**	**15.69**	-0.86	**2.82**	**5.28**	-0.24	-0.43
F17	-39.44	-41.59	-61.04	-27.80	-36.68	-24.25	-49.75
F18	**18.03**	**15.84**	**14.53**	**26.38**	**8.40**	**16.44**	**12.82**
F19	**5.96**	**14.01**	0.91	**7.10**	**10.60**	**8.77**	**2.30**
F20	**39.06**	**19.83**	**21.01**	**43.67**	**23.76**	**15.62**	**17.27**
F21	**13.48**	**15.98**	**14.35**	**18.86**	**11.26**	**15.47**	**10.86**
F22	**14.03**	**10.96**	**14.69**	**19.21**	**10.83**	**11.05**	**7.85**
F23	**15.51**	**15.35**	**12.75**	**21.46**	**7.68**	**20.99**	**9.76**
F24	-2.42	-2.42	-2.42	-2.42	-2.42	-2.42	-2.42
F25	-32.52	-29.99	-43.78	-28.98	-23.55	-25.67	-42.53
Better	19	18	13	19	19	17	13
Equal	2	3	6	2	2	3	6
Worse	4	4	6	4	4	5	6

Table 11. The number of fitness evaluations for different methods.

	ABC	ACO	BBO	DE	GA	KH	OKH7	SGA
F01	39725	50000	18730	15900	50000	12865	**4681**	19210
F02	18480	50000	3600	22380	14870	10147	**3330**	4530
F03	16450	50000	2330	5660	9170	35938	49981	**1920**
F04	48160	50000	50000	21700	50000	23254	**7519**	42130
F05	50000	50000	50000	50000	50000	49986	**49981**	50000
F06	21070	18650	6230	11060	29040	10056	**5610**	5949
F07	21595	50000	7950	12900	18970	14979	**6100**	9845
F08	15820	31500	3180	11140	11320	10237	25458	**2160**
F09	50000	50000	41900	**8340**	50000	49981	49981	44370
F10	18375	47440	5560	15440	16740	26304	12502	**4570**
F11	23345	39260	11940	16920	30100	20687	12472	**6830**
F12	50000	50000	50000	50000	50000	49986	**49981**	50000
F13	47530	50000	18670	45820	44500	27512	49981	**11910**
F14	7210	1600	**440**	2800	1670	5345	3714	720
F15	50000	50000	22590	50000	50000	49981	**20022**	33180
F16	50000	50000	50000	50000	50000	49986	**49981**	50000
F17	50000	50000	50000	50000	50000	49981	49981	**31370**
F18	50000	50000	50000	50000	50000	49981	**7942**	50000
F19	21350	50000	8410	13200	46590	48652	**11320**	13378
F20	50000	50000	50000	39240	50000	14918	**4077**	50000
F21	15750	50000	2820	7020	13670	7912	**2340**	4862
F22	36785	50000	39690	18620	50000	17425	**5013**	20680
F23	24815	50000	14280	13220	43720	14798	**6885**	8660
F24	50000	50000	50000	50000	50000	49981	**49981**	50000
F25	46445	50000	50000	**26520**	50000	49981	49981	50000
Total	0	0	1	2	0	0	16	6

Table 12. Optimization results for the sensor selection problem.

	Best	Mean	Worst	Std
ABC	3.89	4.54	4.87	0.36
ACO	6.02	6.62	6.77	0.31
BBO	3.40	4.69	4.85	0.34
DE	5.44	6.13	6.15	0.28
GA	4.12	5.06	4.84	0.37
KH	3.96	5.52	6.12	0.52
OKH7	**3.35**	**4.47**	**4.79**	**0.27**
SGA	3.73	4.87	4.86	0.47

Table 13. Optimization results for the Sensor Selection problem.

	Best	Mean	Worst	Std
ABC	8.02	8.10	8.14	0.021
ACO	8.10	8.24	8.15	0.040
BBO	7.20	8.02	8.22	0.015
DE	7.70	8.09	8.11	0.021
GA	8.04	8.06	8.06	0.038
KH	8.03	8.08	8.11	0.034
OKH7	**7.16**	**8.00**	8.06	**0.010**
SGA	8.02	8.02	**8.02**	**0**

LIST OF FIGURES

Fig. 1

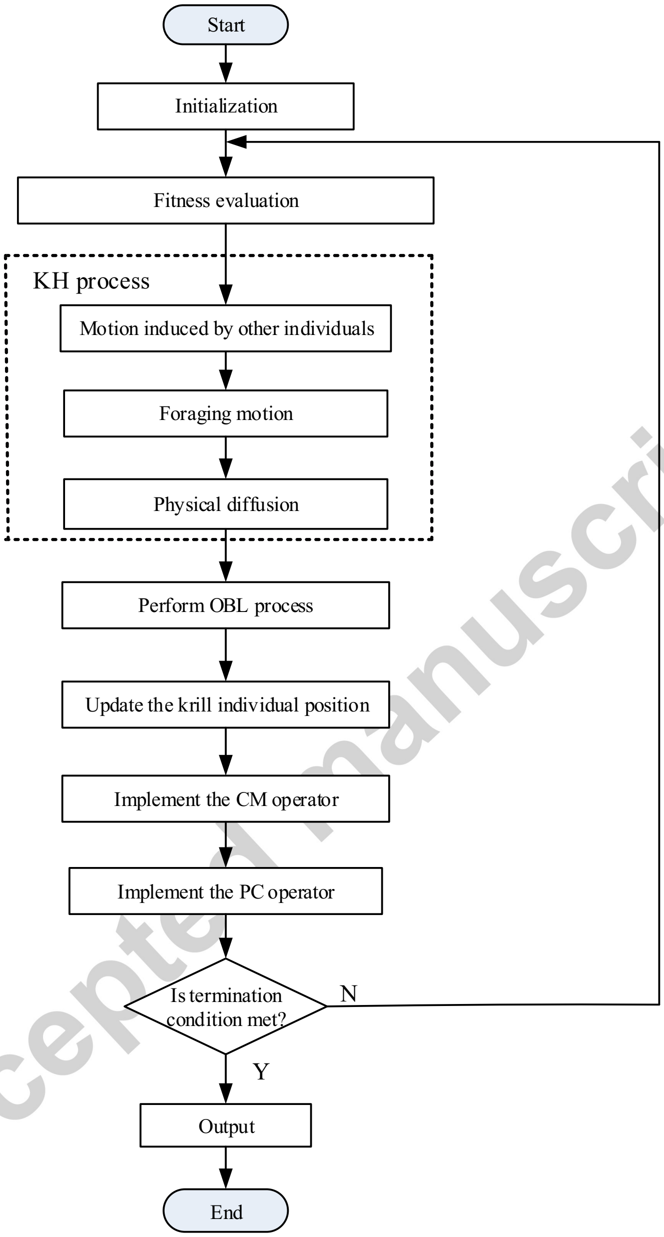

Start
Initialization
Fitness evaluation
KH process
Motion induced by other individuals
Foraging motion
Physical diffusion
Perform OBL process
Update the krill individual position
Implement the CM operator
Implement the PC operator
Is termination condition met?
N
Y
Output
End

Fig. 2

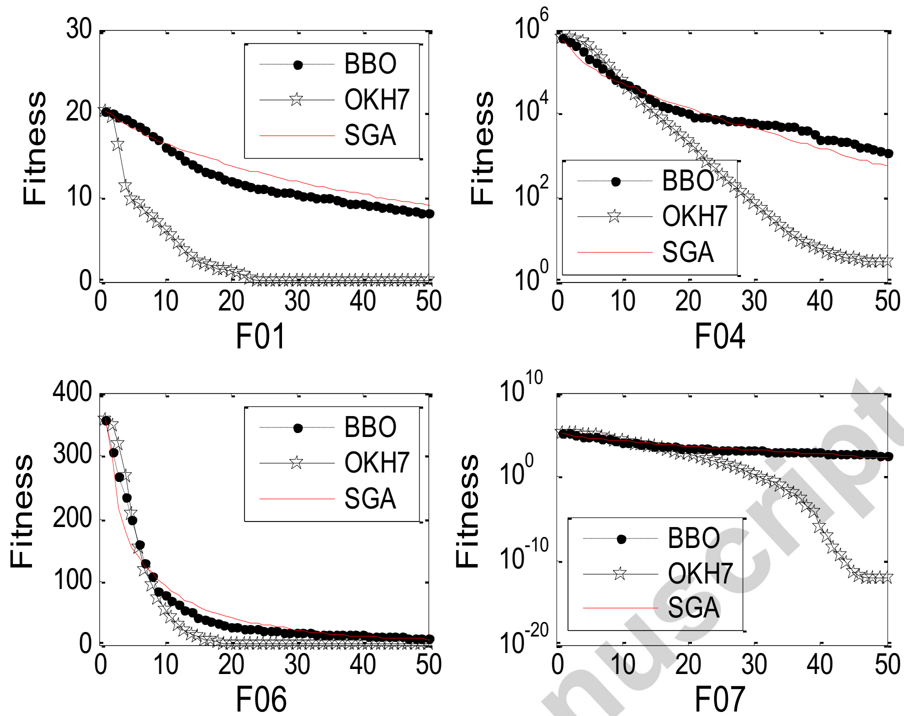


Fig. 3

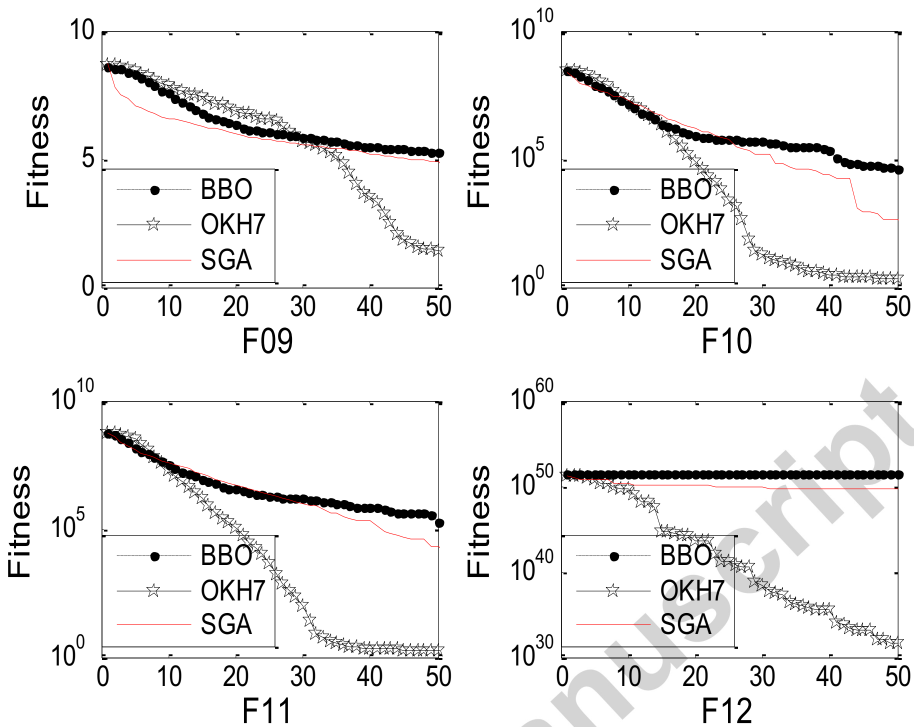
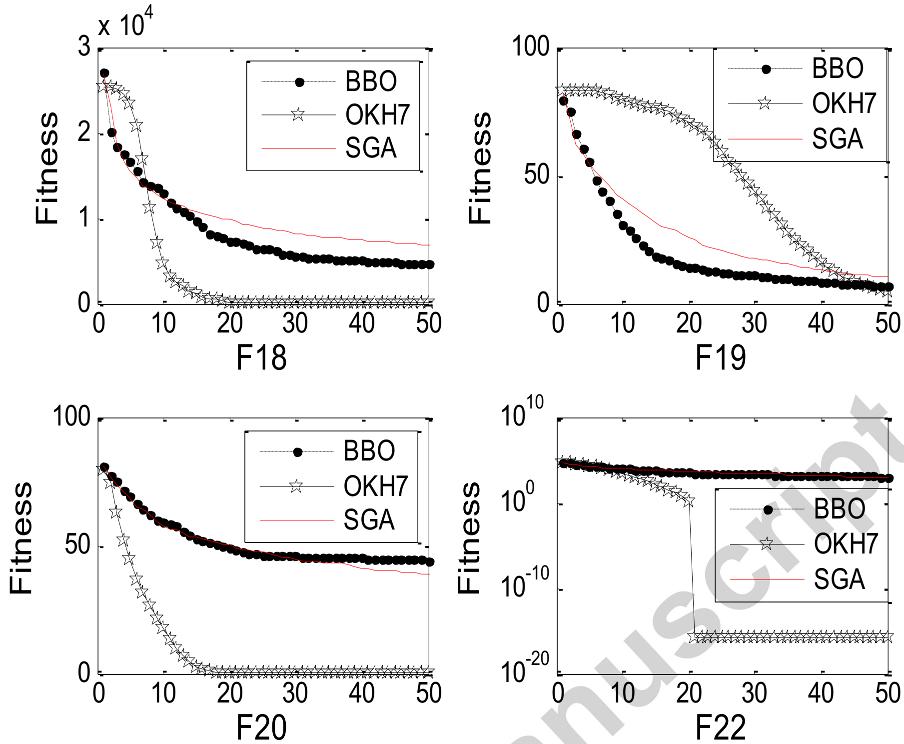


Fig. 4



Biography of the Authors

1st Author

Gai-Ge Wang

Gai-Ge Wang obtained his bachelor degree in computer science and technology from Yili Normal University, Yining, Xinjiang, China, in 2007. His master degree was in the field of intelligent planning and planning recognition at Northeast Normal University, Changchun, China. In 2010 he began working on his Ph.D for computational intelligence and its applications at Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China. He is currently an associate professor in School of Computer Science and Technology at Jiangsu Normal University, Xuzhou, China. Gai-Ge Wang has published over 50 journal papers and conference papers, and more than 30 papers are indexed by SCI/EI. Furthermore, he is referee of other 50 international journals (Elsevier, IEEE, and Springer). He is a member of the International Society for Metaheuristic Optimization in Science and Technology (ISMOST), Publications Chairs of ISCMI 2015, and International Program Committee Member in more than 30 conferences. His research interests are swarm intelligence, soft computing, evolutionary computation, metaheuristic optimization and its applications in engineering, such as scheduling, path planning.

2nd Author

Suash Deb

Suash Deb specializes in Soft Computing, Nanocomputing, Artificial Intelligence, Bioinformatics & the related fields & has published extensively in these areas on reputed SCI-indexed journals in these areas. In the year 2013, his two manuscripts entitled i) "*Multiobjective Cuckoo Search for Design Optimization*", published in Elsevier Publications "Computers & Operations Research" & ii) "*Coupled Eagle Strategy and Differential Evolution for Unconstrained and Constrained Global Optimization*" published in Elsevier Publications "Computers & Mathematics with Applications" were awarded one of the Top 25 Hottest (Most Downloaded) *Articles (Engineering Category* by **Elsevier.** Apart from journals. His research efforts were also found mention in "Wikipedia, the online Encyclopedia".

He has been the recipient of Bharat Excellence Award, Albert Einstein International Award for Scientific Excellence and also Rajiv Gandhi Education Excellence Award (from Intl. Business

Council). Indian Solidarity Council bestowed upon him the Global Education Excellence Award & Certificate of Education Excellence. In 2015, he was awarded the "Pride of International Education Excellence Award" by Intellectual People and Economic Growth Association at Indo-Nepal Friendship Summit, held in Kathmandu.

He also held a number of prestigious fellowships, including i) UNDP Fellowship for visiting Stanford University, USA, ii) CIMPA-INRIA-UNESCO Fellowship for Visiting Intl. Centre for Pure & Applied Mathematics, Nice, France, iii) ICTP Fellowship for visiting Intl. Centre for Theoretical Physics, Trieste, Italy etc.

He has served as the Asian Expert of Advanced Research Project Agency (ARPA), Dept. of Defense, Federal Govt. of USA. He is currently on the Editorial Board of Numerous Intl. journals. His experience consists of academics as well as industry with more emphasis on the former. Currently belonging to CIT, Ranchi, he served reputed institutions like Natl. Centre for Knowledge Based Computing, Kolkata, National Inst. of Science & Technology as well as C.V. Raman College of Engineering, Orissa. He is the Editor-in-Chief of International Journal of Soft Computing & Bioinformatics, Regional Editor of Neural Computer & Applications & Advisory Editor of a no. of other journals. He was the Regional Editor of IEEE Robotics & Automation.

He was elected the President of the International Neural Network Society (INNS)-India Regional Chapter and also served as the Secretary of the IEEE Computational Intelligence Society, Calcutta Chapter. He had also been the Chair of the Task Force of Business Intelligence & Knowledge Management, IEEE Computational Intelligence Society. He is the General Chair of Intl. Symposium on Computational & Business Intelligence (ISCBI15), the flagship event of INNS-India Regional Chapter, to be held in Bali, Indonesia this year. He is also the General Chair of ISCMI15, the event to commemorate the 5th anniversary of INNS-India, to be held in Hong Kong this year. He had been the General Chairs of ISCBI12 & ISCBI13 as well as of ISCMI14, held in New Delhi. Apart from these, he has been the General Chair of many intl. conferences in the field of artificial intelligence, computational intelligence, nanocomputing e.g. Intl. Conference on Intelligent Network & Computing (ICINC13), held in Dubai, Intl. Conference on NanoScience, Technology & Societal Implications (NSTSI12), held in Bhubaneswar etc. He was also the Technical Chair of numerous international events & travelled widely across the globe to deliver keynote address/plenary talk/tutorial talk etc. He is listed in a no. of Who'sWho.

3rd Author

Amir H. Gandomi

Amir H. Gandomi is the pioneer of Krill Herd Algorithm. He was selected as an elite in 2008 by Iranian National Institute of Elites. He used to be a lecturer in Tafresh University and serve as a researcher in National Elites Foundation. He is currently a researcher in Department of Civil

Engineering at the University of Akron, OH. Amir Hossein Gandomi has published over 70 journal papers and several discussion papers, conference papers and book chapters. He has two patents and has published three books in Elsevier. His research interests are Metaheuristics modeling and optimization.

4th Author

Amir H. Alavi

Amir H. Alavi is a senior researcher at the Department of Civil & Environmental Engineering, Michigan State University (MSU), MI, USA. His area of expertise includes energy harvesting, sensor technology, big data analysis, artificial intelligence, statistical and probabilistic methods, and metaheuristic modeling and optimization. He has published three books and over one hundred research papers in book chapters, indexed journals, and conference proceedings, along with two patents. He is on the editorial board of several journals. He is the pioneer of Krill Herd (KH) optimization algorithm.

Photos of the Authors

1st Author

Gai-Ge Wang

2nd Author

Suash deb

3rd Author

Amir H. Gandomi

4th Author

Amir H. Alavi