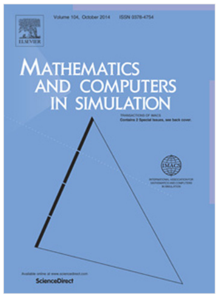
Journal Pre-proof



Dealing with multi-modality using synthesis of Moth-flame optimizer with sine cosine mechanisms

Chengcheng Chen, Xianchang Wang, Helong Yu, Mingjing Wang, Huiling Chen

PII: S0378-4754(21)00124-5
DOI: https://doi.org/10.1016/j.matcom.2021.04.006
Reference: MATCOM 5338

To appear in: *Mathematics and Computers in Simulation*

Received date : 31 May 2020
Revised date : 7 February 2021
Accepted date : 4 April 2021

Please cite this article as: C. Chen, X. Wang, H. Yu et al., Dealing with multi-modality using synthesis of Moth-flame optimizer with sine cosine mechanisms, *Mathematics and Computers in Simulation* (2021), doi: https://doi.org/10.1016/j.matcom.2021.04.006.

Dealing with Multi-modality using Synthesis of Moth-flame Optimizer with Sine Cosine Mechanisms

Chengcheng Chen[1,2], Xianchang Wang[1,2,3*#], Helong Yu[4*#], Mingjing Wang[5], Huiling Chen[6*#]

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China
chencc18@mails.jlu.edu.cn, xcwang89@jlu.edu.cn
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Changchun 130012, China
chencc18@mails.jlu.edu.cn, xcwang89@jlu.edu.cn
[3]Chengdu Kestrel Artificial Intelligence Institute, Chengdu 611730, China
xcwang89@jlu.edu.cn
[4]College of Information Technology, Jilin Agricultural University, Changchun 130118, China
yuhelong@aliyun.com
[5]Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam
mingjingwang@duytan.edu.vn
[6]College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, Zhejiang 325035, China
chenhuiling.jlu@gmail.com

 # These authors contributed equally to this work

⁎ Corresponding Author: Xianchang, Helong Yu and Huiling Chen

E-mail: xcwang89@jlu.edu.cn (Xianchang Wang), yuhelong@aliyun.com (Helong Yu), chenhuiling.jlu@gmail.com (Huiling Chen)

Abstract Evolutionary population-based methods have found their applications in dealing with many real-world simulation experiments and mathematical modeling problems. The Moth-flame optimization (MFO) algorithm is one of the swarm intelligence algorithms and it can be used with constrained and unknown search spaces. However, there are still some defects in its performance, such as low solution accuracy, slow convergence, and insufficient exploration capability. This study improves the basic MFO algorithm from the perspective of improving exploration capability and proposes a hybrid swarm-based algorithm called SMFO. The essential notion is to further explore and scan the feature space with taking advantages of the sine cosine strategy. We methodically investigated the efficacy, solutions, and optimization compensations of the developed SMFO using more than a few demonstrative benchmark tests, together with unimodal, multimodal, hybrid and composition tasks, and two widely applied engineering test problems. The simulations point toward this fact that the diversification and intensification inclinations of the original MFO and its convergence traits are fortunately upgraded. The findings and remarks show that the suggested SMFO is a favorable algorithm and it can show superior efficacy compared to other techniques.

Keywords: Moth-flame optimization algorithm; Global Optimization; Swarm intelligence; Sine

cosine algorithm

1 Introduction

With the advancement of technology and the increasing amount of data, big data, and the emergence of machine learning and deep learning [1, 2], the complexity of tackling optimization cases have been vividly enhanced. Searching for feasible variables of problems, which we may face in manufacturing, is highly demanded in many constrained and unconstrained applications such as operation optimization [3], supply chain planning [4], optimization of hydrothermal systems [5], RFID positioning [6], production and distribution problems [7], and order-picking systems [8]. Also, in some applications such as parameters optimization [9], deployment optimization [10], optimal resource allocation [11], and temperature optimization [12], the problem has several dimension that we should consider at the same time and minimize or maximize an explicit objective function regarding the specific boundary conditions and set of variables. Optimization is a kind of philosophy and way of thinking that can be translated into an algorithmic process and can have a real impact on the quality of life [13]. The base idea is that it tries to enhance the quality of solutions of a problem gradually or based on a prearranged set of conditions in different forms of multiobjective [14-17], many objectives [18-22], fuzzy optimization [23], large scale optimization [24, 25], robust optimization [26], and memetic optimization [27]. Traditional optimization methods do not meet the specific optimization needs of real life. The emergence of meta-heuristics (MAs) offers a new way to solve these complex optimization problems in different areas of neural networks and artificial intelligence and energy field [28-37]. Compared to traditional optimization methods that need to gradient info and slope of the surface [38-40], MAs are more efficient and flexible, and they can find an approximately optimal solution to most of the practical problems within a specific time and with a reasonable complexity [41-48]. Swarm intelligence (SI) algorithms are one of the famous class of MAs that has been applied to many fields such as parameter identification of photovoltaic systems [49-52], feature selection [53], neural networks and machine learning [44, 54-57], image segmentation [29, 58, 59], medical diagnosis [60-65], maximum satisfiability problem) [66, 67], PID optimization control [68-70], wind speed prediction [71], scheduling problem [72, 73], fault diagnosis of rolling bearings [74, 75], gate resource allocation [76, 77], prediction problems in educational field [78-82], bankruptcy prediction [83-88], engineering applications [89, 90], industrial applications [91, 92], etc. SI algorithms are usually inspired by the habits of natural populations [93-96]. The general idea is to distribute the individuals in the search space randomly and then evolve them until a required quality of results [97]. Each individual collects relevant information according to the perception of the surrounding situation, and then gathers these unintelligent individuals to express intelligence through the exchange of information, and adjusts the relevant convergence factors by themselves. Iterate this process until a termination criterion is satisfied and outputs an optimal solution eventually. Usually, such algorithms have the characteristics of fewer control parameters, self-organization, self-learning, and adaptability, which provides the possibility to solve some complex problems efficiently. A large number of SI algorithms have been proposed, which here we can list some of the practical studies on them such as differential search [98, 99], particle swarm optimization (PSO) [100, 101], differential evolution

(DE) [102], slime mould algorithm (SMA)[1] [103], Harris hawks optimization (HHO)[2] [29, 32, 60, 78, 104, 105], whale optimization algorithm (WOA) [37, 41, 106], extremal optimization algorithm [20, 107-109], monarch butterfly optimization (MBO) [110-112], moth search algorithm (MSA) [113].

Moth-Flame optimizer (MFO) is a new SI algorithm proposed in 2015 [114]. It is inspired by the "traverse navigation" mechanism in which moth navigates in space, and it mathematically models this behaviour to perform optimization. The performance of MFO is verified by multiple benchmark functions and engineering applications in [114]. Thus, the MFO has been widely used in many fields. Traditional distributed clustering of wireless sensor networks using K-Means tends to be easily confined to the local optimum. Kotary et al. [115] addressed this problem by using diffusion MFO (DMFO) with the ability to determine the global optimum solution to minimize the distance within the cluster and thus determine the optimal partitioning of each sensor node. Lei et al. [116] incorporated a protein complex prediction algorithm called MFOC based on MFO and applied it to a reliable weighted dynamic protein interaction network. The results show that MFOC outperforms other classical algorithms. Li et al. [117] used MFO to optimize support vector machines (SVM) with feature selection for the diagnosis of tuberculous pleural effusion (TPE). The model exhibited up to 95% accuracy. Elsakaan et al. [118] added Lévy-flight to the MFO to enhance population diversity and applied the model to non-convex economic scheduling problems with valve point effects and emissions. Sayed et al. [119] used a combination of MFO and neutrosophic sets (NS) in histopathological section imaging for automated mitotic detection. MFO was mainly used to select the best distinguishing features of mitotic cells, and the selected features were used for classification and regression tree (CART) prediction. Ng Shin Mei et al. [120] used MFO for optimizing reactive power dispatch (ORPD) problem, and the results show that MFO can produce more satisfactory power loss and voltage deviation than comparable techniques in the literature. The MFO utilized to train a neural net for building a neuroevolution model [121]. The authors in [122] developed an advanced orthogonal MFO with Broyden–Fletcher–Goldfarb–Shanno procedure to solve real-world tasks. Hassanien et al. [123] used MFO for the tomato disease detection problem. Specifically, features are selected through the high performance of MFO and rough set, and the selected features are used for data classification of the SVM. Zhang et al. [124] combined Firefly Algorithm (FA) and MFO to mitigate premature convergence of MFO. The hybrid model was also used for the facial expression recognition system. Li et al. [125] proposed a prediction model with hybrid MFO and least squares SVM and used the model for annual power compliance forecasting. Because of the need for a more accurate optimization algorithm to estimate the model parameters, Allam et al. [126] used MFO to estimate the parameters of a three-diode model for polycrystalline silicon solar cells.

Although MFO had been widely used, it still has its drawbacks [30, 52, 86, 127, 128]. Its ability to explore is not enough to form a good balance with the ability to exploit and prone to stagnation to local or deceptive optima (LO) during continuous iterations. In order to solve these defects, some modifications are proposed. Wang et al. [64] introduced a chaotic strategy in the original MFO and named it CMFO, and then used CMFO for parameter optimization and feature selection in KELM. Finally, the optimized model was used in medical predictive diagnostic problems. Zhang et al. [52] improved the integrative performance of the original MFO by introducing the Nelder-Mead Simplex

---

[1] The source codes of this SMA optimizer is provided at http://aliasgharheidari.com/SMA.html

[2] The source codes of HHO optimizer is provided at http://aliasgharheidari.com/HHO.html

(NMS) mechanism and orthogonal learning (OL) operator in the underlying MFO. The OL is used to create more distributed candidate positions for the search agents in the MFO, while the NMS mechanism explores the space around the optimal position with refinement to improve accuracy. Shehab et al. [129] introduced two steps to enhance the underlying MFO. The first step was to accelerate the search process by combining the Hill Climbing (HC) algorithm with the MFO. The second step is to increase the probability of finding a better solution by adding six crossover mechanisms. To overcome the weak convergence rate of MFO during the global search, Pelusi et al. [130] introduced adaptation-dependent weighting factors to update the position of moth individuals. To solve the problem of insufficient exploration capacity of the original MFO, Kaur et al. [131] improved the distribution of the algorithm by using Cauchy distribution function. At the same time, iterative division and adaptive step size are adopted to balance the distribution and accuracy of the algorithm, and it is named E-MFO. Among them, the differential evolution method is used to improve the accuracy of MFO, and opposite learning mechanism is used to improve the convergence speed of MFO. Xu et al. [132] added Lévy mutation (LM) to MFO to search for the distributed randomness of agents. Cauchy mutation (CM) and Gaussian mutation (GM) are used to increase the number of pilot sampling evaluations of the algorithm on a global scale. Sapre et al. [133] added OBL, CM and Evolutionary Boundary Constraint Processing (EBCH) to the MFO to cope with its premature convergence and local optimal trapping. Xu et al. [86] introduced GM and chaotic local search (CLS) into MFO to propose a new algorithm CLSGMFO and used it for the parameter optimization of kernel extreme learning machine. GM was used to enhance the diversity of the population. CLS was used to improve the refined search of search agents in specific areas. Xu et al. [134] combined the GM strategy and cultural learning mechanisms [135] into the MFO. GM was used to help the MFO escape the local optimum, and CL was used to help search individuals to remember historical experiences and improve exploration capabilities. Savsani et al. [136] proposed a non-dominated moth optimization algorithm (NS-MFO) using the MFO method. Based on the original MFO, the algorithm used non-dominated sorting and crowded distance method to obtain different non-dominated levels, thus ensuring the richness of the optimal solution.

In this study, a hybrid MFO algorithm called SMFO is proposed using the sine cosine strategy, which was derived from the sine cosine algorithm (SCA) that proposed in 2016 [137]. First, divide all individuals into search space randomly and assign a flame to each individual. This allows them to converge to the possible solution while exploring the global space fully. Then, as the iterative process progresses, reduce the number of flames, and the focus of the optimizer will gradually shift from global exploration to specific area exploitation. Finally, the sine cosine strategy is introduced into the MFO to improve exploration capabilities further.

In this paper, MFO is improved to solve the problem that MFO is easy to fall into local optimum due to the lack of exploration ability. By introducing SCA, SMFO improves the exploration ability of MFO and shortens the cycle of the exploration phase, which makes SMFO have more time to execute the exploitation phase, which greatly improves the accuracy of SMFO. Experiments show that the new algorithm outperforms many well-known MAs and improved algorithms in performance.

2 An overview of the MFO algorithm

The standard MFO proposed to be a population-based optimization algorithm. This metaphorbased algorithm was inspired by the navigation method of the moth. We usually observe that moths fly spirally around the lights. In fact, the moth is not attracted by the light but is trapped around the light source by a transverse orientation mechanism used in flight. The transverse orientation mechanism means that the moth maintains a fixed angle with the light source to ensure its straight flight. When the light source is far away, like the moon, the light emitted by it can be regarded as parallel light, and the mechanism can stably play a navigation role. However, when the light source becomes a point source with a relatively close distance, the emitted light is no longer parallel. Currently, the mechanism causes the moth to fall into a deadly spiral flight, and the moth converges toward the light source. The MFO algorithm makes use of this misleading phenomenon modelling to achieve the optimal value solution.

From the above, we can easily conclude that moths and flames are two key components. Moths are the search agents that fly in d-dimensional hyper plane (d = 1, 2, 3, …) and their location is stored in matrix M. the fitness value of each moth is stored in array OM. The flames are the best position that moth had attained so far and is stored in matrix F. The fitness values of flames are stored in array OF. Each moth updates its position according to its own flame using the following equation:

$$\overrightarrow{M_i} = \overrightarrow{D_i} \cdot e^{bt} \cdot \cos(2\pi t) + \overrightarrow{F_j} \qquad \text{Eq. (1)}$$

where $\overrightarrow{M_i}$ indicate the i-th moth, $\overrightarrow{F_j}$ is the j-th flame after sorting, and S is the spiral function. This spiral function must satisfy the following conditions:

1. The spiral should start from the moth.
2. Spiral's final point should be flame.
3. The fluctuation of range of spiral should be within given search space.

Considering these points, the equation is defined as follows:

where $\overrightarrow{D_i}$ indicates the distance between the i-th moth and j-th flame. The i,j is the same as i,j in Eq.(1) It can be calculated as:

$$\overrightarrow{D_i} = \left|\overrightarrow{F_j} - \overrightarrow{M_i}\right| \qquad \text{Eq. (2)}$$

Here, b is a constant for defining the shape of spiral and t is a random number between -1 and 1. The t parameter decides the step size of moth's next movement. After the moths' position updated, update their corresponding flames if any of the moths become fitter than it. Each moth is updated according to the corresponding flame, which may weaken the exploration of the beat promising solutions. Therefore, the following formula is used to adaptively reduce the number of flames to balance exploration and exploitation.

$$flame\_no = \text{round}\left(N - \text{l} * \frac{N-1}{T}\right) \qquad \text{Eq. (3)}$$

where l is the current number of iterations, N is the maximum number of flames, and T indicates the maximum number of iterations.

According to the previous description, the standard MFO algorithm working steps can be obtained as follows. The matrix $\vec{M}$ is sorted by the fitness of $\vec{M}$.

The Pseudo-code of the standard MFO

```
Initialize the position of moths
while (Iteration<=Max_iteration)
  Update flame no using Eq. ()
   OM=FitnessFunction(M);
   if iteration = = 1
```

$\vec{F}$=sort($\vec{M}$);
$\overrightarrow{OF}$=sort ($\overrightarrow{OM}$);
else
$\vec{F}$ = sort ($\vec{M}_{t-1}$, $\vec{M}_t$);
$\overrightarrow{OF}$= sort ($\overrightarrow{OM}_{t-1}$, $\overrightarrow{OM}_t$);
end
for i = 1: n
for j=1: d
Update r and t
Calculate D using Eq. () with respect to the corresponding moth
Update M (i, j) using Eq. (1) and Eq. (2) with respect to the corresponding moth
end
end
end

When the iteration end condition is satisfied, the best moth is returned as the best-obtained approximation optimum.

3 Enhanced MFO with SCA

For any SI algorithm, we should focus on balancing the main traits of the exploration and exploitation [138]. Here, exploration emphasizes the global search capability of the algorithm, while exploitation focuses on the accuracy and quality of the solution. There is two way of thinking about how to deal with solving a practical problem. One way is to utilize a set of rules to extract knowledge from the mathematical info of the surface. Another way is to use the estimated info to approximate some satisfactory solutions for the problem during a reasonable time. Stochastic operations of an algorithm can assist it in showing more variety of patterns during carrying out [139]. The improved SMFO improves the global exploration capability of MFO by combining the sin cosine strategy, increasing the possibility of escaping the local optimal solution. At the same time, the adjustment parameters in sin cosine strategy also ensure the accuracy of the optimal solution, which has shown great potential in many problems [79, 89, 140-145].

The core of sine cosine strategy is to change the position of the initial state through changes in the mathematical function [28, 146, 147]. Individual position updates in the population rely on the increase or decrease of the function value to randomly update the state of each individual in each iteration, using multiple adjustment parameters to ensure the population maintains diversity in the early stage, and the individual tends to develop locally in the later stage, and finally converges to the optimal solution.

During each iteration, the following formula is used to update the state of the individual:

$$\vec{X}_i^{t+1} = \begin{cases} \vec{X}_i^t + r_1 \times sin(r_2) \times \left| r_3 \vec{P}_i^t - \vec{X}_i^t \right|, \ r_4 < 0.5 \\ \vec{X}_i^t + r_1 \times cos(r_2) \times \left| r_3 \vec{P}_i^t - \vec{X}_i^t \right|, \ r_4 \geq 0.5 \end{cases} \quad \text{Eq. (4)}$$

where $\vec{X}_i^t$ is the position of the current solution in i-th dimension at t-th iteration(solution), $\vec{P}_i^t$ is the position of the current optimal solution in i-th dimension at t-th iteration(destination), and $|\,|$ indicates the absolute value.

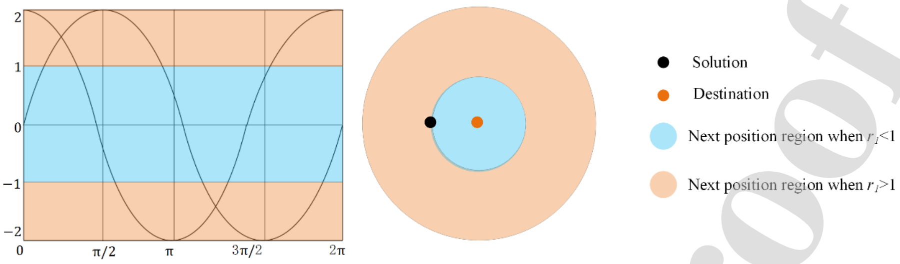


Fig. 1. The basic principle of sine cosine strategy

The effects of sine, cosine and parameters are illustrated in Fig. . The parameter r_1 determines whether the search range for the next location is between or outside solution and destination. This improves the global exploration capability of the MFO algorithm. Parameter r_2 defines the update step of the next position. r_3 is a random weight, the range of values determines the influence of destination on the current solution. r_4 is the random probability of sine and cosine mechanism switching. The cyclic pattern of sine and cosine function allows a solution to be re-positioned around another solution. This can guarantee exploitation of the space defined between two solutions. In order to further balance exploration and exploration, we introduced in $r1{=}a{-}\ t\frac{a}{T}$ Eq. ().

$$r_1 = a - t\frac{a}{T} \qquad \text{Eq. (5)}$$

where t is the current number of iterations, T is the maximum number of iterations, and a is a constant, usually set to 2. This formula adaptively adjusts the parameter size, so that the exploration finally converges gradually to the global optimal.

This paper adds the above mechanism to the spiral update process of MFO, which helps to escape from LO and further convergence towards the global minimum.

The flow chart of SMFO is shown in Fig. 2. It can be seen from the flow chart that SCA is added to SMFO on the basis of MFO, and has little effect on the time complexity.

The Pseudo-code of the SMFO

Initialize the position of moths
while (Iteration<=Max_iteration)
  Update *flame_no* using $flame\_no$=round$\left(N - l * \frac{N-1}{T}\right)$ Eq. ()
  $\overrightarrow{OM}$=FitnessFunction($\vec{M}$);
  if iteration = = 1
    $\vec{F}$=sort($\vec{M}$);
    $\overrightarrow{OF}$=sort ($\overrightarrow{OM}$);
  else
    $\vec{F}$ = sort ($\vec{M}_{t-1}$, $\vec{M}_t$);
    $\overrightarrow{OF}$= sort ($\overrightarrow{OM}_{t-1}$, $\overrightarrow{OM}_t$);
  end
  for i = 1:n

Calculate r_1, r_2, r_3, r_4;

Calculate D using $Di = \left|\vec{F_j} - \vec{M_i}\right|$ Eq. () with respect to the corresponding moth

Update $M(i, j)$ using $Mi = \vec{D_i} \cdot e^{bt} \cdot \cos(2\pi t) + \vec{F_j}$ Eq. (1) and Eq. (2) with respect to the corresponding moth

if (r_4<0.5)

Update $\vec{M}(i, j)$ using $Xit{+}1 = \begin{cases} \vec{X}_i^t + r_1 \times sin(r_2) \times \left|r_3 \vec{P}_i^t - \vec{X}_i^t\right|, & r_4 < 0.5 \\ \vec{X}_i^t + r_1 \times cos(r_2) \times \left|r_3 \vec{P}_i^t - \vec{X}_i^t\right|, & r_4 \geq 0.5 \end{cases}$

Eq. ()

else

Update $\vec{M}(i, j)$ using $Xit{+}1 = \begin{cases} \vec{X}_i^t + r_1 \times sin(r_2) \times \left|r_3 \vec{P}_i^t - \vec{X}_i^t\right|, & r_4 < 0.5 \\ \vec{X}_i^t + r_1 \times cos(r_2) \times \left|r_3 \vec{P}_i^t - \vec{X}_i^t\right|, & r_4 \geq 0.5 \end{cases}$

Eq. ()

end

end

end

end

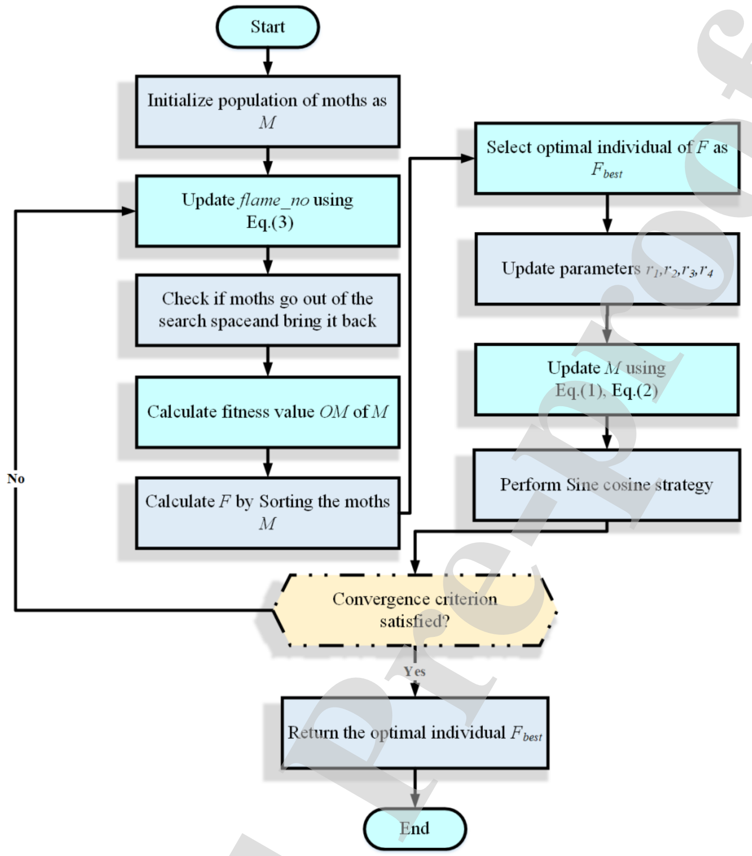


Fig. 2. The flowchart of SMFO

The time complexity of the improved SMFO algorithm relies on the number of algorithm iterations (t), the total number of moths (n), and the number of dimensions of the problems (d). Through analysis, the overall time complexity is O(SMFO)=$O(t(O(Quick\ sort)+O(position\ updating)))$. The time complexity of Quicksort is of $O(nlogn)$ and $O(n^2)$ in the best and worst case, respectively. Updating the position of the moth is $O(n\times d)$. Hence, the final time complexity of the improved SMFO is as follows: O(SMFO)=$O(t(n^2+n\times d))=O(tn^2+tnd)$.

4 Experimental results and discussions

In this section, we firstly introduce the involved benchmark functions and experimental setup, then present the balance and diversity analysis, and finally deliver the comparison results between the SMFO and other competitive peers on the benchmark problems.

4.1 Benchmark functions and experimental setup

In this experimental section, 25 classical functions were chosen to test the performance of the proposed SMFO. The equations of the functions are listed in Table 1, where *Dim* represents the function's dimension; *Dim* is uniformly set to 30. *Range* is the boundary of the feature space for the matching function, and f_{min} represents the optimal value. F1-F5 are unimodal benchmark functions, and F6-F10 are multimodal functions. The first ten benchmark functions are the classical functions utilized by many researchers. F11 is the hybrid function, and F12-F17 are composition functions. They are all taken from IEEE CEC2014. The remaining F18-F25 are functions selected from IEEE CEC2017. In order to ensure the validity of the experimental results, all tests are carried out under the same conditions, the population size and maximum evaluation times were set at 30 and 300,000, respectively. Also, each benchmark function is executed 30 times independently, to avoid the contingency of the experiment. To more intuitively evaluate test results, the Friedman test [148] was used. It is a non-parametric statistical comparison test to rank the average performance of all selected methods. For further statistical comparison, the paired Wilcoxon signed-rank test [149] was also utilized in this paper. If the p-value obtained by the test is less than 0.05, then SMFO is significantly different from other algorithms.

Table 1. Details of 25 benchmark functions

ID	Function Equation	Search Range	Optimum Value
Unimodal Functions			
F1	$f_1(x)=\sum_{i=1}^{n} x_i^2$	[-100,100]	$f_1\{X_{min}\}=0$
F2	$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	[-10,10]	$f_2\{X_{min}\}=0$
F3	$f_3(x)=\sum_{i=1}^{n}(\sum_{j-1}^{i} x_j)^2$	[-100,100]	$f_3\{X_{min}\}=0$
F4	$f_4(x)=max_i\{\lvert x_i\rvert, 1\le i\le n\}$	[-100,100]	$f_4\{X_{min}\}=0$
F5	$f_7(x)=\sum_{i=1}^{n} ix_i^4+random[0,1)$	[-1.28,1.28]	$f_7\{X_{min}\}=0$
Multimodal Functions			
F6	$f_8(x)=\sum_{i=1}^{n}-x_i\sin(\sqrt{\lvert x_i\rvert})$	[-500,500]	$f_8\{X_{min}\}=$ - 418.9829×5
F7	$f_9(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	[-5.12,5.12]	$f_9\{X_{min}\}=0$
F8	$f_{10}(x)=-20\exp\left\{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i}\right\}-exp\{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\}+20+e$	[-32,32]	$f_{10}\{X_{min}\}=0$
F9	$f_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[-600,600]	$f_{11}\{X_{min}\}=0$
F10	$f_{12}(x)=\frac{\pi}{n}\{10\sin(ay_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10sin^2(\pi y_{i+1})]+(y_n-1)^2+\sum_{i=1}^{n}\mu(x_i,10,100,4)\}$ $y_i=1+\frac{x_i+1}{4}$ $\mu(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & x_i>a\\ 0 & -a<x_i<a\\ k(-x_i-a)^m & x_i<-a\end{cases}$	[-50,50]	$f_{12}\{X_{min}\}=0$
CEC 2014 Hybrid Function			
F11	Hybrid function 6	[-100,100]	$f_{20}\{X_{min}\}=2200$
CEC 2014 Composition Functions			

F12	Composition function 1	[-100,100]	$\mathrm{f}_{24}\{X_{min}\} = 2300$
F13	Composition function 2	[-100,100]	$\mathrm{f}_{25}\{X_{min}\} = 2400$
F14	Composition function 4	[-100,100]	$\mathrm{f}_{26}\{X_{min}\} = 2600$
F15	Composition function 5	[-100,100]	$\mathrm{f}_{27}\{X_{min}\} = 2700$
F16	Composition function 6	[-100,100]	$\mathrm{f}_{28}\{X_{min}\} = 2800$
F17	Composition function 7	[-100,100]	$\mathrm{f}_{29}\{X_{min}\} = 2900$
CEC 2017 Composition Functions			
F18	Composition function 3	[-100,100]	$\mathrm{f}_{29}\{X_{min}\} = 2300$
F19	Composition function 4	[-100,100]	$\mathrm{f}_{29}\{X_{min}\} = 2400$
F20	Composition function 5	[-100,100]	$\mathrm{f}_{29}\{X_{min}\} = 2500$
F21	Composition function 6	[-100,100]	$\mathrm{f}_{29}\{X_{min}\} = 2600$
F22	Composition function 7	[-100,100]	$\mathrm{f}_{29}\{X_{min}\} = 2700$
F23	Composition function 8	[-100,100]	$\mathrm{f}_{29}\{X_{min}\} = 2800$
F24	Composition function 9	[-100,100]	$\mathrm{f}_{29}\{X_{min}\} = 2900$
F25	Composition function 10	[-100,100]	$\mathrm{f}_{29}\{X_{min}\} = 3000$

4.2 Diversity and balance analysis

We further analyze SMFO and MFO on the benchmark functions selected in this paper. Fig. 3 shows the results of the feasibility experimental analysis of SMFO and MFO on F3 and F6. (a) Show the distribution of SMFO search locations in three dimensions. (b) Show the distribution of SMFO search locations in two dimensions. We can see that most of the SMFO search locations in the figure are concentrated on a single line, and a small number of points are scattered throughout the space. (c) Show the trajectory of the SMFO. In the selected graphs, F5 remains to fluctuate for a long time, while the other functions have stabilized at the beginning, which shows that SMFO has excellent search capability and has quickly identified the position of the optimal solution. (d) Shows the convergence curves of the two tested algorithms. From the convergence curves, we can see more intuitively that the SMFO algorithm is not only better than MFO in terms of convergence speed, but also significantly better than MFO in terms of the quality of the solutions found. This is mainly due to the introduction of the SCA strategy, which allows MFO to find a right balance between global and local search.

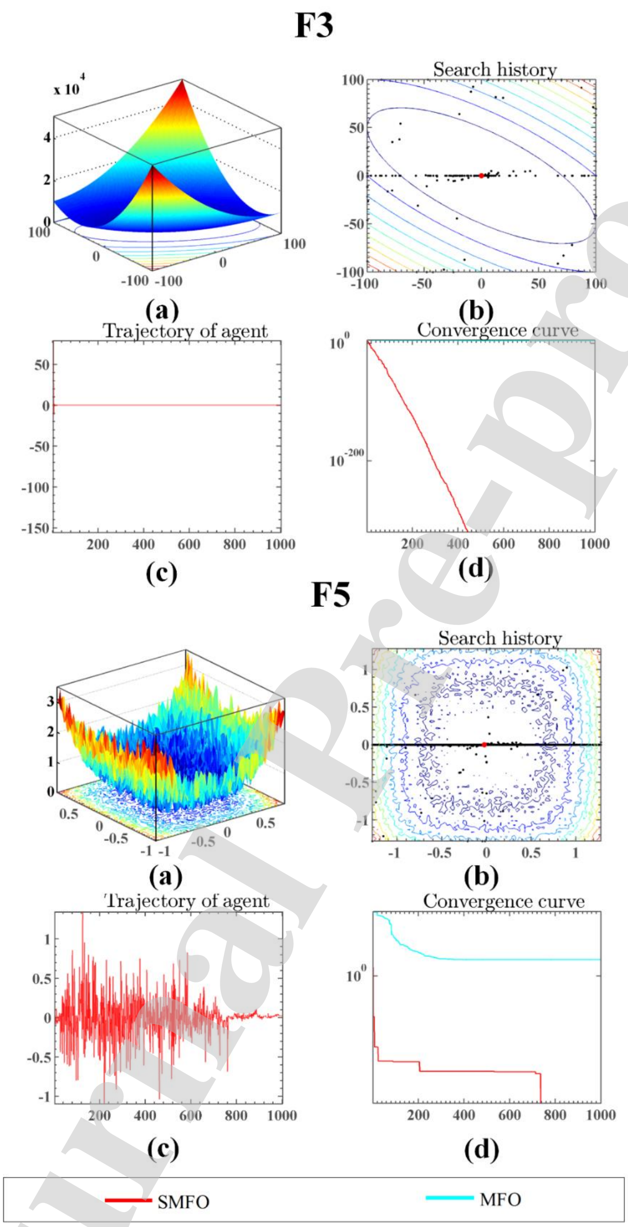


Fig. 3. (a) Three-dimensional location distribution of SMFO, (b) Two-dimensional location distribution of SMFO, (c) Trajectory of SMFO in the first dimension, (d) Convergence of SMFO

We next perform a balance and diversity analysis of SMFO to understand the characteristics of SMFO further. Fig. 4 shows the results of the balanced analysis of SMFO and MFO. We express the abscissa in the graph as an index, so as to better observe the change of its curve. There are three curves in the diagram of the equilibrium analysis, two of which represent search and exploitation, respectively. Which curve has a high value is indicative of which behaviour was dominant at the time. Moreover, in order to represent the relationship more visually between the two behaviours, a third incremental-decremental curve was added. When the curve is incremental, it means that the search effect is greater than the exploitation effect. The curve will reach its maximum when the effect of the two behaviours is the same.

From Fig. 4, we can see that the two algorithms are relatively similar in their equilibrium. For most of the time, they are performing local searches. Both algorithms ended the global search phase very quickly and then moved on to the local exploitation phase. However, compared with MFO, SMFO can enter the exploitation stage faster, which indicates that the exploitation ability of SMFO is stronger than MFO.

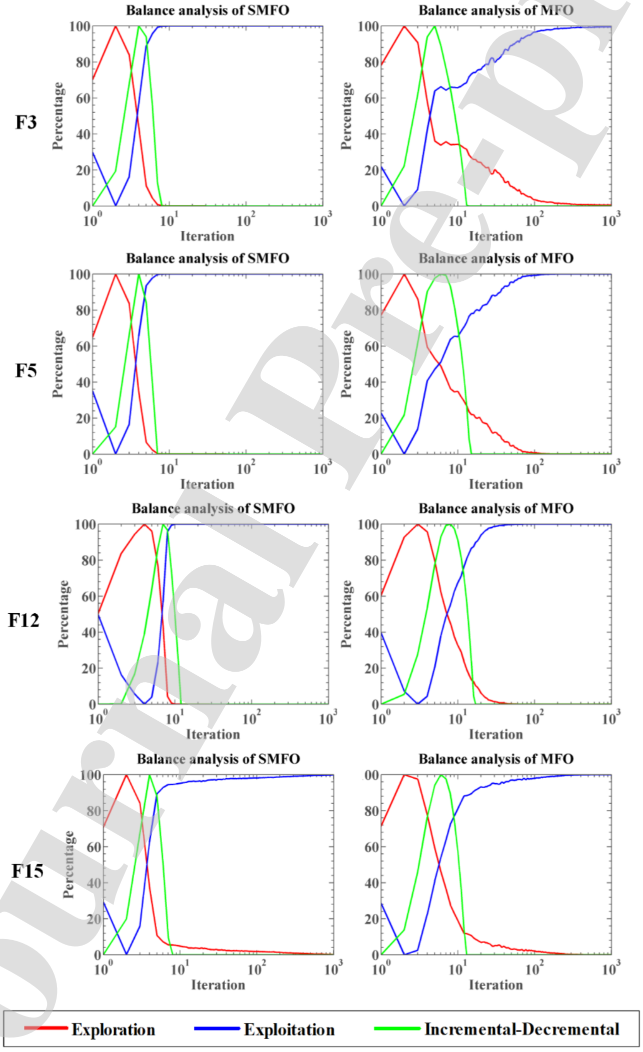


Fig. 4. Balance analysis of SMFO and MFO

Fig. 5 shows the results of the diversity analysis of SMFO. Population diversity is always a large value at the outset, as the position is randomly generated at the time of initialization of a population. From the figure, we can see that the diversity changes of SMFO and MFO are also relatively similar. Neither has a large fluctuation in diversity change, and both fall rapidly shortly after running, suggesting that both algorithms converge rapidly after starting to run. It can be seen from the graph that the diversity of SMFO declines faster than MFO, which reaches a small value almost immediately at the beginning. This indicates that the SMFO converges rapidly and the region where the optimal solution is located is identified in a short period of time.

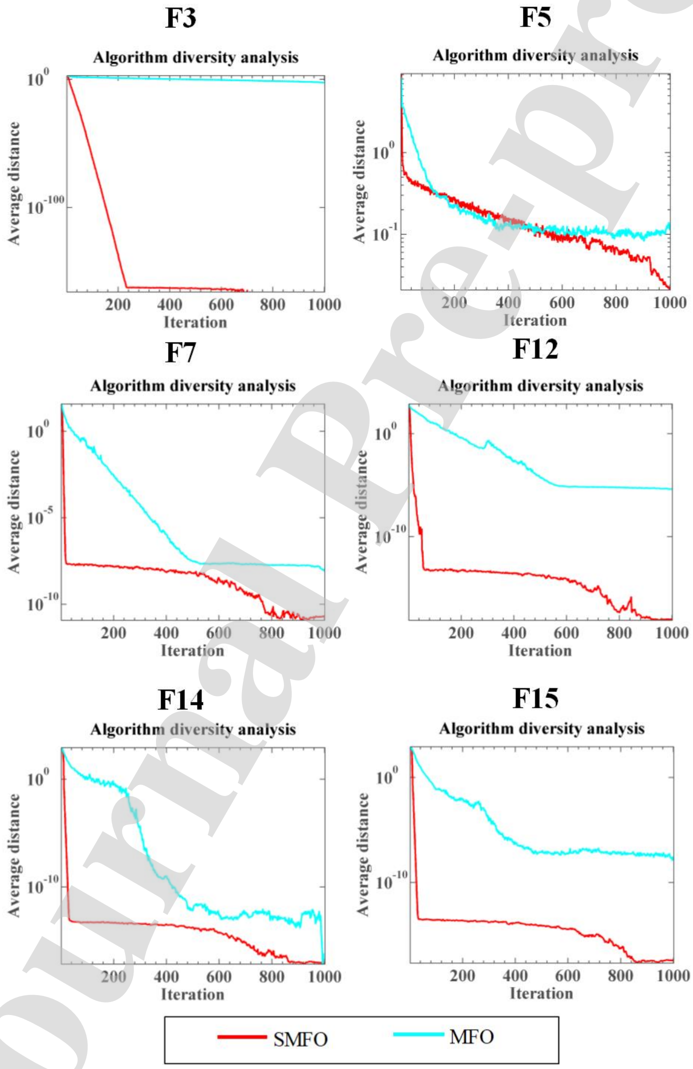


Fig. 5. Diversity analysis of SMFO and MFO

4.3 Comparison with other methods

In this section, we compare the improved SMFO with the original MFO [114] and other common

MAs including sine cosine algorithm (SCA) [137], grey wolf optimizer (GWO) [150], WOA, multi-verse optimizer (MVO) [151], PSO, DE, firefly algorithm (FA) [152], bat algorithm (BA) [153], cuckoo search [154]. For fairness of comparisons, as a logical rule in the optimization and neural networks community, we are constrained to establish the same setting and computing environment for the compared methods. Such a condition can make the comparisons more reliable and make it easier to compare the methods again and obtain the attained outcomes and patterns. All the methods involved in these tests are consistent with their common parameters, as labeled in section 4.1 and the parameters used by each algorithm are set to the recommended values of the original paper. Parameters are shown in Table 2. The experimental results are shown in Tables 3-5.

Table 3 shows that the performance of SMFO is not only significantly better than the original MFO but also superior to other algorithms on 22 out of 25 functions. In addition, the std value of SMFO is the lowest of most test functions, which means SMFO has the smallest deviations compared with other methods. This shows that SMFO has better stability and can find optimum values in a smaller range.

Table 4 shows the results of the Wilcoxon test for SMFO compare with other algorithms. 94% p-value is less than 0.05. As for statistical significance cases "+/-/=", where "+", "−" and "=" indicate that the performance of SMFO is better, worse, and equal to the corresponding algorithm, respectively. It shows that SMFO is better than each algorithm in most comparison functions.

Table 2 Parameter settings of the comparison algorithms

Algorithm	Parameter settings
SCA	$A = 2$
GWO	$a = [2,0]$
MFO	$b = 1;\ t = [-1,1];\ a \in [-1,-2]$
WOA	$a_1 = [2,0];\ a_2 = [-2,-1];\ b = 1$
MVO	$existence\ probability \in [0.2\ 1];\ travelling\ distance\ rate \in [0.6\ 1]$
PSO	$c_1 = 2;\ c_2 = 2;\ vMax = 6$
DE	$scaling\ factor = 0.5;\ crossover\ probability = 0.5$
FA	$\alpha = 0.5;\ \beta = 0.2;\ \gamma = 1$
BA	$A = 0.5;\ r = 0.5$
CS	$P_a \in [0,1]$

Table 3 Comparison results of SMFO on the 25 functions with traditional algorithms

	F1		F2		F3	
Algorithm	AVG	STD	AVG	STD	AVG	STD
SMFO	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
MFO	3.33E+03	6.06E+03	4.00E+01	1.97E+01	1.70E+04	1.32E+04
GWO	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	7.42E-178	0.00E+00
WOA	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	3.76E+01	6.64E+01
BA	7.01E-01	4.46E-01	3.42E+00	1.77E+00	2.40E-01	1.31E-01
SCA	9.21E-56	2.96E-55	4.44E-60	1.56E-59	1.38E+00	5.44E+00
CS	1.28E-30	2.08E-30	3.15E-14	2.49E-14	1.21E-03	8.19E-04
DE	2.49E-159	6.69E-159	1.94E-94	2.07E-94	1.34E+03	4.54E+02
MVO	3.13E-03	9.65E-04	3.89E-02	9.76E-03	3.66E-01	1.37E-01
FA	1.11E+04	1.05E+03	4.79E+01	2.88E+00	1.96E+04	2.86E+03
PSO	9.77E+01	1.26E+01	4.62E+01	3.33E+00	1.73E+02	2.36E+01

	F4		F5		F6	
Algorithm	AVG	STD	AVG	STD	AVG	STD
SMFO	**0.00E+00**	0.00E+00	**8.31E-06**	9.63E-06	**-1.26E+04**	2.04E-04
MFO	6.70E+01	7.55E+00	3.22E+00	7.65E+00	-8.47E+03	8.23E+02
GWO	5.41E-152	2.27E-151	6.33E-05	4.16E-05	-6.46E+03	6.53E+02
WOA	6.08E+00	1.11E+01	1.34E-04	1.32E-04	-1.24E+04	3.90E+02
BA	5.02E+00	4.92E+00	1.37E+01	8.05E+00	-7.14E+03	6.72E+02
SCA	4.28E-02	2.18E-01	3.06E-03	3.26E-03	-4.46E+03	2.76E+02
CS	1.90E+00	1.44E+00	1.18E-02	4.46E-03	-1.07E+04	3.12E+02
DE	3.43E-15	2.96E-15	2.64E-03	5.31E-04	-1.24E+04	1.25E+02
MVO	1.07E-01	3.59E-02	3.05E-03	1.37E-03	-8.12E+03	5.95E+02
FA	4.07E+01	1.44E+00	4.01E+00	7.29E-01	-4.19E+03	1.88E+02
PSO	3.75E+00	2.27E-01	1.15E+02	3.15E+01	-6.66E+03	1.08E+03
	F7		F8		F9	
Algorithm	AVG	STD	AVG	STD	AVG	STD
SMFO	**0.00E+00**	0.00E+00	**8.88E-16**	0.00E+00	**0.00E+00**	0.00E+00
MFO	1.45E+02	4.96E+01	1.32E+01	8.07E+00	1.51E+01	4.16E+01
GWO	**0.00E+00**	0.00E+00	7.40E-15	1.35E-15	**0.00E+00**	0.00E+00
WOA	**0.00E+00**	0.00E+00	3.26E-15	2.35E-15	7.96E-04	3.18E-03
BA	2.48E+02	2.16E+01	1.73E+00	8.24E-01	1.72E-02	2.01E-02
SCA	2.97E-03	1.63E-02	7.01E+00	9.23E+00	3.37E-11	1.85E-10
CS	2.74E+01	5.23E+00	6.21E-02	2.36E-01	3.27E-13	1.79E-12
DE	6.63E-02	2.52E-01	7.40E-15	1.35E-15	**0.00E+00**	0.00E+00
MVO	9.27E+01	2.09E+01	8.44E-02	3.85E-01	2.66E-02	1.31E-02
FA	2.28E+02	1.31E+01	1.60E+01	2.82E-01	1.01E+02	9.23E+00
PSO	3.39E+02	1.58E+01	7.70E+00	3.82E-01	1.02E+00	1.26E-02
	F10		F11		F12	
Algorithm	AVG	STD	AVG	STD	AVG	STD
SMFO	2.39E-07	4.00E-07	**2.50E+03**	0.00E+00	**2.60E+03**	8.88E-06
MFO	8.53E+06	4.67E+07	2.67E+03	5.03E+01	2.67E+03	3.10E+01
GWO	3.82E-02	2.54E-02	2.63E+03	7.87E+00	**2.60E+03**	5.55E-04
WOA	8.49E-07	4.29E-07	2.62E+03	3.50E+01	2.61E+03	4.41E+00
BA	6.93E+00	3.29E+00	2.62E+03	2.14E-03	2.66E+03	1.84E+01
SCA	3.42E-01	5.33E-02	2.67E+03	1.15E+01	**2.60E+03**	5.58E-02
CS	6.78E-15	3.71E-14	2.62E+03	1.39E-12	2.63E+03	1.05E+00
DE	**1.57E-32**	5.57E-48	2.62E+03	1.39E-12	2.63E+03	2.06E+00
MVO	6.38E-02	1.05E-01	2.62E+03	1.26E-01	2.62E+03	1.45E+01
FA	2.12E+06	8.79E+05	2.73E+03	2.22E+01	2.70E+03	4.18E+00
PSO	3.52E+00	5.30E-01	2.62E+03	4.31E-01	2.63E+03	5.65E+00
	F13		F14		F15	
Algorithm	AVG	STD	AVG	STD	AVG	STD
SMFO	**2.70E+03**	0.00E+00	**2.90E+03**	0.00E+00	**3.00E+03**	0.00E+00
MFO	2.71E+03	7.64E+00	3.64E+03	1.90E+02	4.01E+03	2.17E+02
GWO	2.71E+03	4.97E+00	3.33E+03	1.17E+02	3.95E+03	2.77E+02
WOA	2.71E+03	1.64E+01	3.82E+03	3.57E+02	4.99E+03	5.85E+02
BA	2.73E+03	1.45E+01	3.93E+03	3.43E+02	5.38E+03	8.43E+02
SCA	2.72E+03	8.77E+00	3.46E+03	3.16E+02	4.83E+03	3.49E+02
CS	2.71E+03	1.46E+00	3.11E+03	1.08E+01	3.78E+03	7.55E+01
DE	2.71E+03	1.10E+00	3.23E+03	8.71E+01	3.64E+03	2.59E+01
MVO	2.71E+03	1.65E+00	3.23E+03	1.50E+02	3.84E+03	2.44E+02
FA	2.73E+03	4.75E+00	3.79E+03	9.70E+01	4.22E+03	1.19E+02

PSO	2.71E+03	7.39E+00	3.45E+03	2.91E+02	6.91E+03	9.34E+02
	F16		F17		F18	
Algorithm	AVG	STD	AVG	STD	AVG	STD
SMFO	2.32E+06	9.49E+06	8.89E+05	7.15E+05	**2.50E+03**	0.00E+00
MFO	2.47E+06	3.45E+06	5.99E+04	4.63E+04	2.96E+03	3.58E+01
GWO	1.16E+06	2.16E+06	4.35E+04	2.85E+04	2.88E+03	5.76E+01
WOA	6.38E+06	4.64E+06	9.29E+04	6.22E+04	3.16E+03	1.24E+02
BA	3.48E+07	3.36E+07	1.98E+04	4.48E+04	3.56E+03	3.07E+02
SCA	1.63E+07	9.98E+06	2.39E+05	7.30E+04	3.26E+03	4.22E+01
CS	**3.91E+03**	1.19E+02	**5.00E+03**	4.68E+02	2.91E+03	2.60E+01
DE	3.00E+04	1.15E+05	7.13E+03	1.81E+03	2.88E+03	9.43E+00
MVO	1.01E+06	3.01E+06	8.18E+03	1.46E+03	2.87E+03	2.32E+01
FA	3.38E+06	1.03E+06	1.63E+05	4.09E+04	3.11E+03	1.53E+01
PSO	9.05E+04	1.79E+05	1.32E+04	5.69E+03	4.70E+03	5.74E+02
	F19		F20		F21	
Algorithm	AVG	STD	AVG	STD	AVG	STD
SMFO	**2.60E+03**	0.00E+00	**2.70E+03**	0.00E+00	**2.80E+03**	0.00E+00
MFO	3.50E+03	4.34E+01	3.53E+03	6.81E+02	6.62E+03	3.72E+02
GWO	2.97E+03	3.47E+02	3.19E+03	1.21E+02	5.12E+03	8.07E+02
WOA	2.68E+03	3.06E+02	2.71E+03	6.62E+01	4.38E+03	2.50E+03
BA	2.87E+03	5.29E+02	3.02E+03	3.80E+01	5.15E+03	3.49E+03
SCA	3.86E+03	7.51E+01	3.63E+03	1.34E+02	8.00E+03	4.21E+02
CS	2.74E+03	1.72E+02	2.91E+03	1.97E+01	3.98E+03	1.04E+03
DE	3.40E+03	7.58E+00	2.91E+03	5.93E+00	5.41E+03	9.44E+01
MVO	3.39E+03	1.50E+02	2.92E+03	2.77E+01	4.99E+03	8.57E+02
FA	3.69E+03	2.05E+01	4.12E+03	1.24E+02	7.29E+03	1.61E+02
PSO	2.67E+03	3.32E+00	2.96E+03	6.34E+01	3.40E+03	4.35E+01
	F22		F23		F24	
Algorithm	AVG	STD	AVG	STD	AVG	STD
SMFO	**2.90E+03**	0.00E+00	**3.00E+03**	0.00E+00	**3.10E+03**	0.00E+00
MFO	3.62E+03	1.06E+02	5.15E+03	3.64E+02	4.07E+03	2.53E+02
GWO	3.72E+03	1.99E+02	3.75E+03	4.14E+02	3.48E+03	1.62E+02
WOA	4.02E+03	1.92E+02	3.24E+03	5.57E+02	4.32E+03	4.10E+02
BA	3.91E+03	1.87E+02	3.47E+03	6.34E+02	4.65E+03	4.28E+02
SCA	4.04E+03	1.03E+02	5.81E+03	6.09E+02	4.16E+03	3.18E+02
CS	3.52E+03	5.79E+01	3.22E+03	3.14E+01	3.69E+03	1.00E+02
DE	3.43E+03	1.57E+01	4.07E+03	8.58E+02	3.48E+03	7.76E+01
MVO	3.61E+03	1.02E+02	3.41E+03	5.99E+02	3.69E+03	1.74E+02
FA	3.91E+03	8.92E+01	4.18E+03	4.35E+02	4.54E+03	1.71E+02
PSO	5.05E+03	7.79E+02	3.30E+03	3.59E+01	3.98E+03	2.45E+02
	F25					
Algorithm	AVG	STD				
SMFO	**3.20E+03**	0.00E+00				
MFO	1.65E+06	1.70E+06				
GWO	6.92E+05	6.84E+05				
WOA	3.06E+06	3.55E+06				
BA	1.36E+06	9.03E+05				
SCA	6.49E+06	1.26E+07				
CS	7.56E+03	1.29E+03				
DE	6.13E+04	2.31E+04				
MVO	1.55E+06	1.65E+06				

FA	1.01E+08	3.62E+07
PSO	2.79E+06	1.27E+06

Table 4 Comparison results of SMFO and conventional algorithms are compared on Wilcoxon test (p-values of the Wilcoxon test, lower is better. The worst p-values are shown in bold)

	MFO	GWO	WOA	BA	SCA	CS	DE	MVO	FA	PSO
F1	1.70E-06	**1.00E+00**	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F2	1.59E-06	**1.00E+00**	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F3	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F4	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F5	1.73E-06	3.52E-06	7.69E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F6	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.36E-05	1.73E-06	1.73E-06	1.73E-06
F7	1.73E-06	**1.00E+00**	**1.00E+00**	1.73E-06	**1.00E+00**	1.73E-06	**5.00E-01**	1.73E-06	1.73E-06	1.73E-06
F8	1.72E-06	2.57E-07	1.05E-04	1.73E-06	1.61E-06	1.73E-06	2.57E-07	1.73E-06	1.73E-06	1.73E-06
F9	2.67E-05	**1.00E+00**	**5.00E-01**	1.73E-06	**1.00E+00**	**5.00E-01**	1.00E+00	1.73E-06	1.73E-06	1.73E-06
F10	**2.06E-01**	1.92E-06	3.11E-05	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F11	1.73E-06	1.73E-06	3.79E-06	1.73E-06	1.73E-06	4.32E-08	4.32E-08	1.73E-06	1.73E-06	1.73E-06
F12	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F13	1.73E-06	1.23E-05	1.22E-04	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F14	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F15	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F16	3.59E-04	3.59E-04	3.06E-04	5.75E-06	9.71E-05	3.59E-04	3.59E-04	3.59E-04	3.59E-04	3.59E-04
F17	1.36E-05	1.36E-05	1.36E-05	1.36E-05	6.89E-05	1.36E-05	1.36E-05	1.36E-05	2.37E-05	1.36E-05
F18	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F19	1.73E-06	8.84E-05	**5.00E-01**	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F20	1.73E-06	2.56E-06	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F21	1.73E-06	2.56E-06	3.91E-03	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F22	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F23	1.71E-06	1.73E-06	9.77E-04	1.73E-06	1.73E-06	1.73E-06	1.71E-06	1.73E-06	1.73E-06	1.73E-06
F24	1.73E-06	1.73E-06	2.56E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F25	1.73E-06	1.73E-06	3.79E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
+/-/=	23/1/1	19/2/4	18/1/6	24/1/0	22/1/2	21/3/1	20/3/2	23/2/0	24/1/0	23/2/0

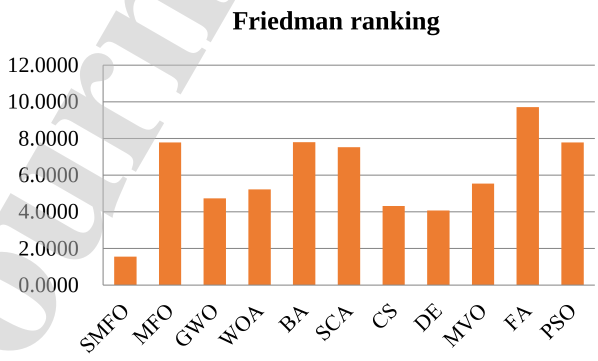


Fig. 6. Average ranking values of SMFO and other well-known algorithms

In order to compare SMFO with other algorithms more intuitively, the Friedman test is used, and the results are listed in Fig. 6. It is revealed that SMFO ranks first, indicating that SMFO has the best performance.

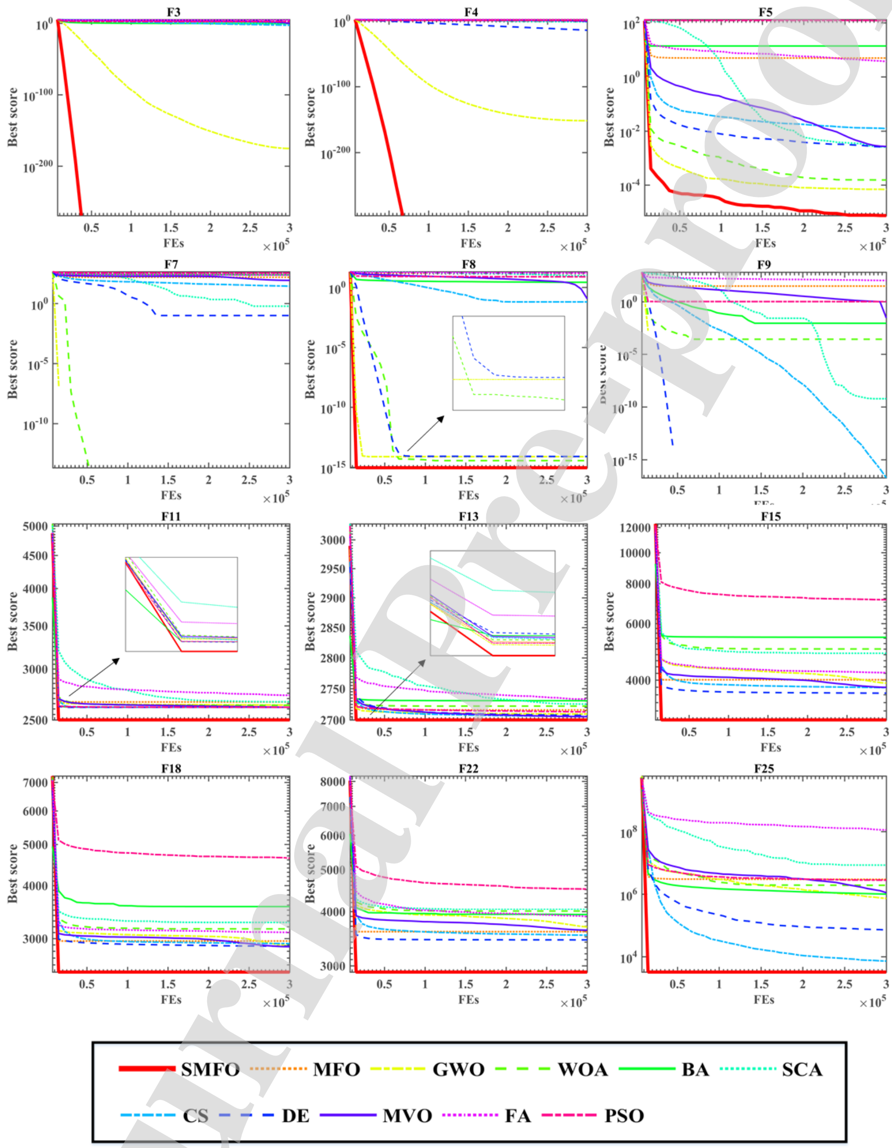


Fig. 7. Convergence curves of SMFO and other algorithms

Fig. 7 shows the convergence curves of SMFO and other original algorithms with 30 dimensions. We can use the diagram to evaluate the performance of the SMFO further. From F3 and F4, we can see that the convergence speed of SMFO is the fastest. But it does not mean that SMFO runs less times than other algorithms, which indicates that SMFO has found the global optimal solution in the early stage. Although GWO follows it, the precision of SMFO is far less than that of GWO. Moreover, some even fall into local optimum at the beginning. As for F7 and F9, other algorithms finally find the best value, but SMFO converges rapidly to global optimal solution at the beginning of the iteration. For F5, F8, F11, F13, F15 and F18, the convergence speed of each function is similar, but SMFO has better exploration ability, and it provides the best results. For F22 and F25, the convergence rate of SMFO is not the fastest in the early exploration, but it can find the solution much better than other algorithms. From the above analysis, we can see that SMFO has superior performance to find optimal values for a different type of functions.

4.4 Comparison with other well-established advanced algorithms

To further illustrate the performance of SMFO, we compare SMFO with CMA-ES [155], CMFO [156], CLSGMFO [86], CESCA [157], CSSA [158], FSTPSO [159], and OBSCA [160]. These algorithms have superior performance, even CLSGMFO surpasses SMFO in general, but SMFO is still irreplaceable.

It can be seen from Table 5 that although SMFO is inferior in F16-F25, SMFO has strong performance and stability on F1-F15. More than half of SMFO functions can reach the minimum value, which fully reflects the powerful performance of SMFO. CLSGMFO followed closely and even surpassed SMFO on F16-F25. However, SMFO has faster convergence speed than CLSGMFO. It can be seen from Fig. 8 that the convergence speed of SMFO on F1, F2 and F4 is faster than that of CLSGMFO, and even higher accuracy is obtained on F5 than CLSGMFO.

As can be seen from Table 6, SMFO goes beyond most of the comparison algorithms. CMA-ES and SMFO have their advantages and disadvantages. CMA-ES is better than SMFO in combination function, and SMFO surpasses CMA-ES in single-mode and multi-mode functions. However, on the whole, SMFO is better than CMA-EA. CLSGMFO is very close to SMFO in F1, F2, F3, F4, F7, F8, F9, F11, F13, F14, F15, F16, and surpasses SMFO in F17-F25. However, the convergence speed and time complexity of SMFO are lower than that of CLSGMFO.

Table 5 Comparison results of SMFO on the 25 functions with variants algorithms

	F1		F2		F3	
Algorithm	AVG	STD	AVG	STD	AVG	STD
SMFO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
CMA-ES	1.92E-29	1.54E-30	1.54E-02	6.08E-02	7.23E-28	9.95E-29
CMFO	2.60E+02	5.85E+02	1.98E-01	8.48E-01	3.78E+04	8.64E+03
CLSGMFO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
CESCA	9.84E+02	7.46E+02	8.45E+00	2.55E+00	4.40E+03	2.99E+03
CSSA	4.86E-03	3.65E-03	3.11E-02	1.01E-02	1.21E+00	6.60E-01
FSTPSO	2.43E+03	1.18E+03	2.91E+01	1.26E+01	1.05E+04	4.00E+03
OBSCA	1.29E-107	6.71E-107	7.72E-91	3.14E-90	1.55E-23	8.42E-23
	F4		F5		F6	
Algorithm	AVG	STD	AVG	STD	AVG	STD
SMFO	**0.00E+00**	**0.00E+00**	**6.65E-06**	**7.34E-06**	**-1.26E+04**	3.63E-04
CMA-ES	2.11E-15	9.90E-17	5.95E-02	1.59E-02	-7.17E+03	5.89E+02
CMFO	5.28E+01	8.51E+00	9.25E-01	4.43E-01	-9.78E+03	3.14E+03
CLSGMFO	**0.00E+00**	**0.00E+00**	3.64E-05	2.97E-05	**-1.26E+04**	**5.24E-08**
CESCA	1.99E+01	8.04E+00	4.96E-01	3.51E-01	-3.95E+03	2.11E+02
CSSA	2.51E-02	1.10E-02	2.56E-04	3.13E-04	**-1.26E+04**	1.30E-05
FSTPSO	3.02E+01	5.09E+00	5.14E-01	3.27E-01	-5.15E+03	8.90E+02
OBSCA	5.05E-21	2.73E-20	9.68E-04	5.75E-04	-4.15E+03	2.61E+02

Algorithm	F7 AVG	F7 STD	F8 AVG	F8 STD	F9 AVG	F9 STD
SMFO	**0.00E+00**	**0.00E+00**	**8.88E-16**	**0.00E+00**	**0.00E+00**	**0.00E+00**
CMA-ES	2.35E+02	4.67E+01	1.95E+01	1.81E-01	8.21E-04	3.38E-03
CMFO	6.88E+01	2.32E+01	1.89E+00	2.56E+00	1.85E+00	1.86E+00
CLSGMFO	**0.00E+00**	**0.00E+00**	**8.88E-16**	**0.00E+00**	**0.00E+00**	**0.00E+00**
CESCA	4.91E+01	1.69E+01	7.02E+00	1.87E+00	1.03E+01	5.49E+00
CSSA	1.08E+02	1.04E+02	1.71E-02	7.86E-03	1.68E-02	1.35E-02
FSTPSO	1.87E+02	3.25E+01	1.35E+01	1.12E+00	1.71E+01	7.71E+00
OBSCA	**0.00E+00**	**0.00E+00**	4.44E-15	**0.00E+00**	**0.00E+00**	**0.00E+00**

Algorithm	F10 AVG	F10 STD	F11 AVG	F11 STD	F12 AVG	F12 STD
SMFO	6.30E-07	1.87E-06	**2.50E+03**	**0.00E+00**	**2.60E+03**	9.18E-06
CMA-ES	1.11E-30	1.59E-31	2.62E+03	1.83E-12	2.65E+03	6.70E+01
CMFO	3.34E+04	8.44E+04	2.63E+03	1.56E+01	2.66E+03	1.25E+01
CLSGMFO	**1.96E-28**	**5.18E-28**	**2.50E+03**	**0.00E+00**	**2.60E+03**	**0.00E+00**
CESCA	3.89E+04	1.63E+05	3.05E+03	1.63E+02	2.65E+03	1.81E+01
CSSA	3.32E-05	2.54E-05	2.63E+03	3.28E+02	**2.60E+03**	3.53E-01
FSTPSO	1.72E+05	3.90E+05	2.81E+03	6.69E+01	2.71E+03	1.42E+01
OBSCA	3.91E-01	4.00E-02	2.68E+03	1.55E+01	**2.60E+03**	2.74E-04

Algorithm	F13 AVG	F13 STD	F14 AVG	F14 STD	F15 AVG	F15 STD
SMFO	**2.70E+03**	**0.00E+00**	**2.90E+03**	**0.00E+00**	**3.00E+03**	**0.00E+00**
CMA-ES	**2.70E+03**	1.96E+00	3.12E+03	1.04E+02	7.42E+03	2.83E+03
CMFO	2.73E+03	8.27E+00	3.79E+03	3.88E+02	5.56E+03	7.01E+02
CLSGMFO	**2.70E+03**	**0.00E+00**	**2.90E+03**	**0.00E+00**	**3.00E+03**	**0.00E+00**
CESCA	2.72E+03	7.63E+00	4.04E+03	1.41E+02	5.43E+03	3.21E+02
CSSA	**2.70E+03**	1.05E-02	4.66E+03	3.57E+02	1.02E+04	1.55E+03
FSTPSO	2.75E+03	1.25E+01	3.99E+03	3.15E+02	8.24E+03	1.27E+03
OBSCA	**2.70E+03**	9.49E-10	3.25E+03	3.99E+01	5.56E+03	3.90E+02

Algorithm	F16 AVG	F16 STD	F17 AVG	F17 STD	F18 AVG	F18 STD
SMFO	1.28E+06	6.99E+06	7.78E+05	7.22E+05	3.34E+03	1.55E+02
CMA-ES	**3.69E+03**	**7.74E+01**	**5.42E+03**	**8.06E+02**	4.24E+03	9.08E+02
CMFO	4.63E+07	3.04E+07	4.65E+05	4.67E+05	2.96E+03	6.49E+01
CLSGMFO	5.17E+03	1.13E+04	6.35E+03	9.95E+03	**2.80E+03**	**3.24E+01**
CESCA	1.83E+07	3.53E+06	1.40E+06	2.88E+05	3.47E+03	4.29E+01
CSSA	4.13E+07	1.26E+08	1.07E+07	6.50E+06	3.87E+03	1.99E+02
FSTPSO	3.06E+07	3.03E+07	5.56E+05	3.04E+05	3.39E+03	2.15E+02
OBSCA	1.56E+07	8.66E+06	4.27E+05	1.39E+05	3.03E+03	3.39E+01

Algorithm	F19 AVG	F19 STD	F20 AVG	F20 STD	F21 AVG	F21 STD
SMFO	3.70E+03	2.05E+02	4.15E+03	5.41E+02	9.70E+03	1.07E+03
CMA-ES	**2.86E+03**	**1.39E+01**	**2.89E+03**	**6.06E-01**	**3.69E+03**	5.47E+02
CMFO	3.14E+03	1.08E+02	2.96E+03	5.29E+01	6.72E+03	8.80E+02
CLSGMFO	2.96E+03	3.50E+01	2.90E+03	1.59E+01	3.97E+03	1.32E+03
CESCA	3.47E+03	4.08E+01	5.63E+03	4.91E+02	1.13E+04	**5.28E+02**
CSSA	4.29E+03	2.14E+02	5.79E+03	7.01E+02	1.30E+04	8.42E+02
FSTPSO	3.50E+03	1.47E+02	4.20E+03	5.52E+02	9.30E+03	8.70E+02
OBSCA	3.19E+03	3.10E+01	3.36E+03	1.37E+02	7.01E+03	7.01E+02

	F22		F23		F24	

Algorithm	AVG	STD	AVG	STD	AVG	STD
SMFO	3.96E+03	3.27E+02	5.49E+03	6.03E+02	6.64E+03	2.77E+03
CMA-ES	3.46E+03	9.55E+02	**3.14E+03**	5.97E+01	**3.64E+03**	**1.54E+02**
CMFO	3.40E+03	9.69E+01	3.47E+03	4.49E+02	4.58E+03	4.31E+02
CLSGMFO	**3.31E+03**	6.87E+01	3.22E+03	**4.01E+01**	3.95E+03	1.85E+02
CESCA	3.70E+03	9.05E+01	7.10E+03	4.09E+02	6.07E+03	2.07E+02
CSSA	5.40E+03	5.03E+02	8.16E+03	6.80E+02	1.40E+04	5.21E+03
FSTPSO	3.92E+03	3.16E+02	5.29E+03	8.61E+02	5.73E+03	5.73E+02
OBSCA	3.45E+03	**4.14E+01**	4.20E+03	2.83E+02	5.00E+03	2.42E+02

	F25	
Algorithm	AVG	STD
SMFO	4.81E+08	6.35E+08
CMA-ES	**5.22E+03**	**2.11E+02**
CMFO	3.70E+05	7.62E+05
CLSGMFO	1.62E+05	2.94E+05
CESCA	2.20E+09	6.57E+08
CSSA	3.37E+09	1.56E+09
FSTPSO	7.77E+07	6.93E+07
OBSCA	1.20E+08	3.51E+07

Table 6 Comparison results of SMFO and variants algorithms are compared on Wilcoxon test

	CMAES	CMFO	CLSGM	CESCA	CSSA	FSTPSO	OBSCA
F1	1.73E-06	1.73E-06	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F2	1.73E-06	1.73E-06	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F3	1.73E-06	1.73E-06	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F4	1.73E-06	1.73E-06	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F5	1.73E-06	1.73E-06	2.16E-05	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F6	1.73E-06	1.73E-06	1.73E-06	1.73E-06	3.88E-06	1.73E-06	1.73E-06
F7	1.73E-06	1.73E-06	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	**1.00E+00**
F8	1.73E-06	1.73E-06	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	4.32E-08
F9	**5.00E-01**	1.73E-06	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	**1.00E+00**
F10	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F11	4.32E-08	1.73E-06	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F12	1.73E-06	1.73E-06	4.38E-04	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F13	1.73E-06	1.73E-06	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	1.56E-02
F14	1.73E-06	1.73E-06	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F15	1.73E-06	1.73E-06	**1.00E+00**	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F16	3.11E-05	1.73E-06	**7.50E-01**	1.80E-05	2.37E-05	4.73E-06	1.97E-05
F17	5.31E-05	**8.22E-02**	2.67E-05	9.63E-04	1.92E-06	**1.99E-01**	2.85E-02
F18	6.89E-05	1.73E-06	1.73E-06	1.11E-03	1.73E-06	**5.30E-01**	1.73E-06
F19	1.73E-06	1.73E-06	1.73E-06	1.49E-05	1.73E-06	7.16E-04	1.73E-06
F20	1.73E-06	1.73E-06	1.73E-06	1.92E-06	2.88E-06	**7.81E-01**	1.73E-06
F21	1.73E-06	1.73E-06	1.73E-06	4.73E-06	1.73E-06	**1.92E-01**	1.73E-06
F22	3.32E-04	1.73E-06	1.73E-06	9.63E-04	1.73E-06	**6.88E-01**	1.73E-06
F23	1.73E-06	1.92E-06	1.73E-06	1.73E-06	1.73E-06	**1.59E-01**	2.60E-06
F24	1.73E-06	1.92E-06	1.73E-06	**8.45E-01**	1.36E-05	2.18E-02	1.64E-05
F25	1.73E-06	1.73E-06	1.73E-06	2.88E-06	1.73E-06	6.98E-06	3.06E-04
+/-/=	14/10/1	16/8/1	1/12/12	222/1	24/1/0	16/3/0	14/9/2

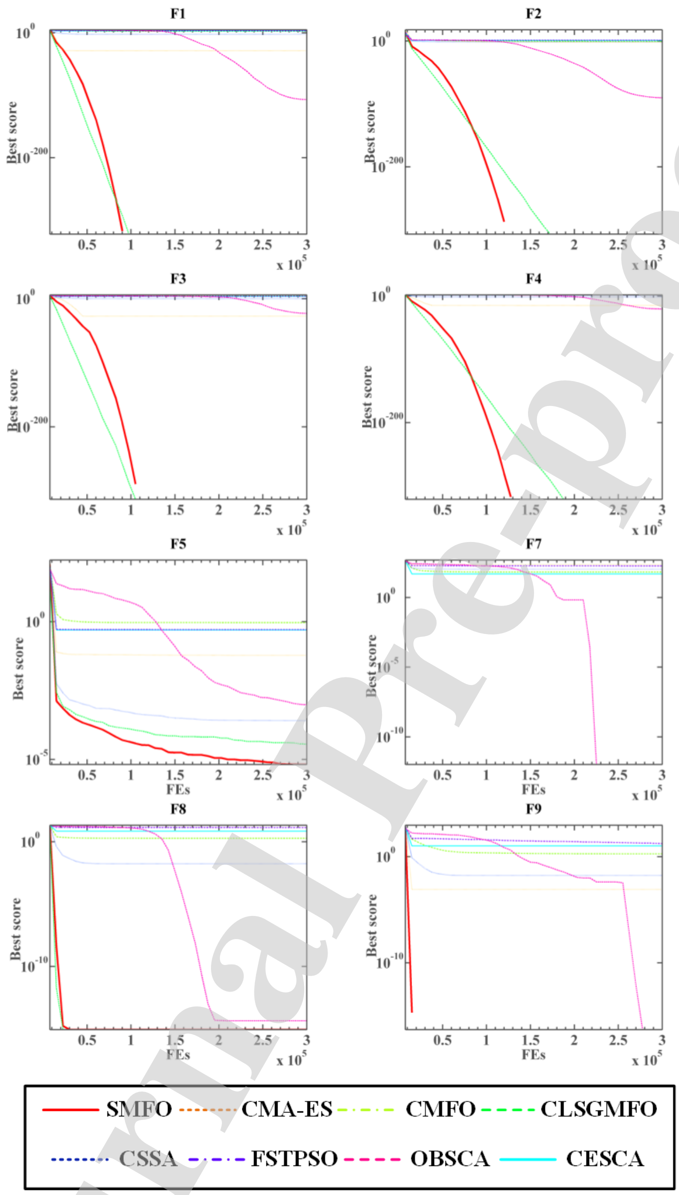


Fig. 8. Convergence curve of SMFO and other variants algorithms

4.5 Influence of parameter setting on SMFO's performance

We use F5 in benchmark functions as the test function when testing the influence of N and Max_FEs on SMFO algorithm. N is set to 10, 30, 50, 100 and 200, respectively. Max_FEs is initialized to 10000, 100000, 300000, 500000,1000000 and 2000000. The test results can be visually observed in Fig. 9. The increase of N and Max_FEs will improve the accuracy of SMFO, but when it reaches a certain level, the influence will become minimal. The experimenter can set it according to his own needs in the actual experiment, because if the value is too large, it will take a long time, and if the value is too small, the experimental results are not ideal.

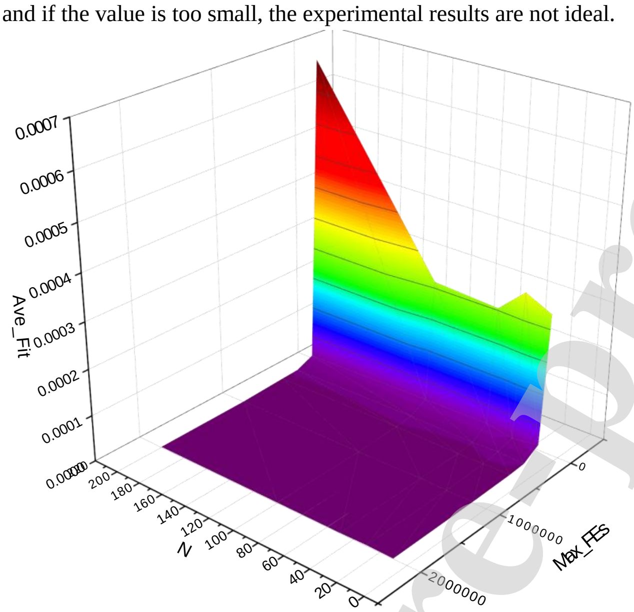


Fig. 9. The influence of N and Max_FEs

5 SMFO for the engineering benchmarks

5.1 I-beam design

The last structural optimization problem in this section is the design of I-beams, which aims to design I-beams to achieve minimum vertical deflection. At the same time, length, height and thickness are the structural parameters of the problem. The mathematical model of this problem can be described as follows:

Consider $\vec{x} = [x_1\ x_2\ x_3\ x_4] = [b\ h\ t_w\ t_f]$

Objective: $f(\vec{x})_{min} = \frac{5000}{\frac{t_w(h-2t_f)^3}{12}+\frac{bt_f^3}{6}+2bt_f(\frac{h-t_f}{2})^2}$

Subject to $g(\vec{x}) = 2bt_w + t_w(h-2t_f) \leq 0$

Variable range $10 \leq x_1 \leq 50,\ 10 \leq x_2 \leq 80,\ 0.9 \leq x_3 \leq 5,\ 0.9 \leq x_4 \leq 5$

We can use the mathematical method and MAs to solve the design problem of I-beam (IBD). MAs include BA, ARSM [161] , LARSM [161], CS [162], SOS [163], MFO and WOA. The experimental results between the above techniques and SMFO are shown in Table 8. In order to make a fair comparison, we apply a similar penalty function for SMFO.

Table 8 indicates that SMFO is superior to all other algorithms in dealing with IBD problems and ultimately provides the most effective design scheme.

Table 8. Comparison results of the I-beam design problem.

Algorithm	Optimal values for variables				Optimum weight
	b	***h***	t_w	t_f	
SMFO	**50.00000**	**80.00000**	**1.763301**	**5.000000**	**0.006626**
BA	43.91803	76.98691	2.613636	0.900000	0.006625959
ARSM	48.42000	79.99000	0.900000	2.400000	0.015700
IARSM	0.244200	6.223100	8.291500	0.243300	0.131000
CS	50.00000	80.00000	0.900000	2.321675	0.013075
SOS	50.00000	80.00000	0.900000	2.321790	0.013074
MFO	50.00000	80.00000	1.764700	5.000000	0.0066259
WOA	49.99799	80.00000	1.764748	5.000000	0.00662619

In summary, the experimental studies on these two classical engineering design problems reveal that the suggested SMFO has an excellent performance in the optimization of practical problems. Furthermore, the efficacy of the proposed algorithm is confirmed when dealing with constrained problems. The reason why SMFO outperforms other methods in the constrained problems is that SMFO can effectively assist the original MFO in coordinating the exploration and exploration propensity.

5.2 Multiple disk clutch brake problem

This is a minimization problem categorized as a discrete optimization problem. Its objective is to use five discrete design variables to minimize the quality of multidisc clutch brakes. The five variables are actuating force, inner and outer radius, number of 27 friction surfaces, and thickness of discs. Figure 15 shows the configuration of this problem. The mathematical model for this problem is as follows:

$f(x) = \Pi\left(r_0^2 - r_i^2\right)t(Z+1)\rho$

subject to:

$g_1(x) = r_0 - r_i - \Delta r \geq 0$

$g_2(x) = l_{max} - (Z+1)(t+\delta) \geq 0$

$g_3(x) = P_{max} - P_{rz} \geq 0$

$g_4(x) = P_{max} v_{sr_{max}} - P_{rz} v_{sr} \geq 0$

$g_5(x) = v_{sr_{max}} - v_{sr} \geq 0$

$g_6 = T_{max} - T \geq 0$

$g_7(x) = M_h - sM_s \geq 0$

$g_8(x) = T \geq 0$

$M_h = \frac{2}{3}\mu F Z \frac{r_0^3 - r_i^2}{r_0^2 - r_i^3} \quad P_{rz} = \frac{F}{\Pi\left(r_0^2 - r_i^2\right)} \quad v_{rz} = \frac{2\Pi n\left(r_0^3 - r_i^3\right)}{90\left(r_0^2 - r_i^2\right)} \quad T = \frac{I_z \Pi n}{30\left(M_h + M_f\right)}$

$\Delta r = 20\,mm \;\; I_z = 55\,kgmm^2 \;\; P_{max} = 1\,MPa \;\; F_{max} = 1000\,N \;\; T_{max} = 15\,s \;\; \mu = 0.5 \;\; s = 1.5$
$M_s = 40\,Nm \quad M_f = 3\,Nm \quad \mathrm{n} = 250\,rpm \quad v_{sr_{max}} = 10\,m/s \quad l_{max} = 30\,mm \quad r_{i_{min}} = 60$
$r_{i_{max}} = 80 \quad r_{0_{min}} = 90 \quad r_{0_{max}} = 110 \quad t_{min} = 1.5 \quad t_{max} = 3 \quad F_{min} = 600 \quad F_{max} = 1000$
$Z_{min} = 2 \;\; Z_{max} = 9$

This paper compares SMFO with WCA [164], PVS [165], and TLBO [166] to minimize the quality of multidisc clutch brakes. Table 9 shows the details of the comparison. We can find from

Table 14 that the quality of the SMFO algorithm is far less than that of other algorithms, reaching 0.313656. This shows that the algorithm has a stronger optimization ability and can find more high-quality problem solutions.

Table 9. Results of Multiple disk clutch brake compared with other methods

Algorithm	r_i	r_0	t	F	Z	Optimal cost
SFMO	60.00000	90.00000	1	600.0000	2.000000	0.313656
WCA	70.00000	90.00000	1	910.0000	3.000000	0.313656
PVS	70.00000	90.00000	1	980.0000	3.000000	0.313660
TLBO	70.00000	90.00000	1	810.0000	3.000000	0.313656

6 Conclusions and future works

In this paper, a hybrid MFO algorithm called SMFO is proposed by introducing the sine cosine strategy into the original MFO. The proposed SMFO has further balanced the exploration and exploitation of MFO and substantially enhanced the diversity of moth populations. The numerical results on representative benchmark functions including unimodal, multimodal, hybrid and composition functions show that the proposed method is significantly better than the original MFO. Furthermore, we also compared SMFO with other well-known MAs and improved variants of MAs to validate its superiority. It shows that SMFO is good at solving function optimization problems, and can effectively alleviate the premature convergence of MFO. Its ability to jump out of local optima is enhanced, and the accuracy of the solution is also improved to a great extent as well. Compared with the original MFO, SMFO improves the exploration ability of MFO and shortens the exploration cycle of SMFO. Therefore, the accuracy and convergence speed of SMFO are improved indirectly. However, shortening the exploration period means that on some complex multi-mode functions, it may be easier to fall into local optimization than other algorithms. For example, the gap can be seen in F7 and F9 functions.

For future work, how to improve the exploration ability of SMFO in the middle and late-stage under the condition of ensuring the exploitation ability is a problem needed to solve. Moreover, how to successfully apply the proposed SMFO to dealing with multiobjective and dynamic landscapes is also a popular research direction. Secondly, the application of SMFO to brain disease diagnosis [167], fault diagnosis [168], clustering analysis [169], face recognition and micro-expression recognition [170], social evolution modelling [171] and image recognition [172] is also an exciting topic for us, and we are researching to apply SMFO for them.

Acknowledgements

This research is supported by National Natural Science Foundation of China (U19A2061, U1809209, 62076185), Science and Technology Development Project of Jilin Province (20190301024NY), Jilin Provincial Industrial Innovation Special Fund Project (2018C039-3). We acknowledge the comments of editor and anonymous reviewers that enhanced this research significantly.

References

[1] K. Shi, J. Wang, S. Zhong, Y. Tang, and J. Cheng, "Non-fragile memory filtering of TS fuzzy delayed neural networks based on switched fuzzy sampled-data control," *Fuzzy Sets and Systems,* vol. 394, pp. 40-64, 2020.

[2] L. Xiong, H. Zhang, Y. Li, and Z. Liu, "Improved stability and H∞ performance for neutral systems with uncertain Markovian jump," *Nonlinear Analysis: Hybrid Systems,* vol. 19, pp. 13-25, 2016.

[3] E. Liu, L. Lv, Y. Yi, and P. Xie, "Research on the steady operation optimization model of natural gas pipeline considering the combined operation of air coolers and compressors," *IEEE Access,* vol. 7, pp. 83251-83265, 2019.

[4] Y. Chen, L. He, Y. Guan, H. Lu, and J. Li, "Life cycle assessment of greenhouse gas emissions and water-energy optimization for shale gas supply chain planning based on multi-level approach: Case study in Barnett, Marcellus, Fayetteville, and Haynesville shales," *Energy conversion and management,* vol. 134, pp. 382-398, 2017.

[5] Y. Deng, T. Zhang, B. K. Sharma, and H. Nie, "Optimization and mechanism studies on cell disruption and phosphorus recovery from microalgae with magnesium modified hydrochar in assisted hydrothermal system," *Science of The Total Environment,* vol. 646, pp. 1140-1154, 2019.

[6] C. Wu, X. Wang, M. Chen, and M. J. Kim, "Differential received signal strength based RFID positioning for construction equipment tracking," *Advanced Engineering Informatics,* vol. 42, p. 100960, 2019.

[7] L. Liu and S. Liu, "Integrated Production and Distribution Problem of Perishable Products with a Minimum Total Order Weighted Delivery Time," *Mathematics,* vol. 8, p. 146, 2020.

[8] W. Ran, S. Liu, and Z. Zhang, "A Polling-Based Dynamic Order-Picking System considering Priority Orders," *Complexity,* vol. 2020, p. 4595316, 2020/07/24 2020.

[9] C. Cai, X. Wu, W. Liu, W. Zhu, H. Chen, J. C. D. Qiu*, et al.*, "Selective laser melting of near-α titanium alloy Ti-6Al-2Zr-1Mo-1V: Parameter optimization, heat treatment and mechanical performance," *Journal of Materials Science & Technology,* 2020.

[10] B. Cao, J. Zhao, Y. Gu, S. Fan, and P. Yang, "Security-aware industrial wireless sensor network deployment optimization," *IEEE Transactions on Industrial Informatics,* vol. 16, pp. 5309-5316, 2019.

[11] J. Yan, W. Pu, S. Zhou, H. Liu, and M. S. Greco, "Optimal Resource Allocation for Asynchronous Multiple Targets Tracking in Heterogeneous Radar Networks," *IEEE Transactions on Signal Processing,* vol. 68, pp. 4055-4068, 2020.

[12] C. Cai, X. Gao, Q. Teng, R. Kiran, J. Liu, Q. Wei*, et al.*, "Hot isostatic pressing of a near α-Ti alloy: Temperature optimization, microstructural evolution and mechanical performance evaluation," *Materials Science and Engineering: A,* p. 140426, 2020.

[13] Q. Lv, H. Liu, J. Wang, H. Liu, and Y. Shang, "Multiscale analysis on spatiotemporal dynamics of energy consumption CO2 emissions in China: Utilizing the integrated of DMSP-OLS and NPP-VIIRS nighttime light datasets," *Science of the Total Environment,* vol. 703, p. 134394, 2020.

[14] B. Cao, J. Zhao, P. Yang, Y. Gu, K. Muhammad, J. J. Rodrigues*, et al.*, "Multiobjective 3-D Topology Optimization of Next-Generation Wireless Data Center Network," *IEEE Transactions on Industrial Informatics,* vol. 16, pp. 3597-3605, 2019.

[15] G.-G. Wang, X. Cai, Z. Cui, G. Min, and J. Chen, "High performance computing for cyber physical

social systems by using evolutionary multi-objective optimization algorithm," *IEEE Transactions on Emerging Topics in Computing,* vol. 8, pp. 20-30, 2020.

[16] J. Sun, Z. Miao, D. Gong, X.-J. Zeng, J. Li, and G.-G. Wang, "Interval multi-objective optimization with memetic algorithms," *IEEE Transactions on Cybernetics,* vol. 50, pp. 3444-3457, 2020.

[17] H. M. Ridha, C. Gomes, H. Hizam, M. Ahmadipour, A. A. Heidari, and H. Chen, "Multi-objective optimization and multi-criteria decision-making methods for optimal design of standalone photovoltaic system: A comprehensive review," *Renewable and Sustainable Energy Reviews,* vol. 135, p. 110202 (https://doi.org/10.1016/j.rser.2020.110202), 2021/01/01/ 2021.

[18] B. Cao, W. Dong, Z. Lv, Y. Gu, S. Singh, and P. Kumar, "Hybrid Microgrid Many-Objective Sizing Optimization with Fuzzy Decision," *IEEE Transactions on Fuzzy Systems,* 2020.

[19] B. Cao, X. Wang, W. Zhang, H. Song, and Z. Lv, "A Many-Objective Optimization Model of Industrial Internet of Things Based on Private Blockchain," *IEEE Network,* vol. 34, pp. 78-83, 2020.

[20] M.-R. Chen, G.-Q. Zeng, and K.-D. Lu, "A many-objective population extremal optimization algorithm with an adaptive hybrid mutation operation," *Information Sciences,* vol. 498, pp. 62-90, 2019/09/01/ 2019.

[21] Z.-M. Gu and G.-G. Wang, "Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization," *Future Generation Computer Systems,* vol. 107, pp. 49-69, 2020.

[22] Y. Zhang, G.-G. Wang, K. Li, W.-C. Yeh, M. Jian, and J. Dong, "Enhancing MOEA/D with information feedback models for large-scale many-objective optimization," *Information Sciences,* vol. 522, pp. 1-16, 2020.

[23] H. Huang, X. a. Feng, S. Zhou, J. Jiang, H. Chen, Y. Li*, et al.*, "A new fruit fly optimization algorithm enhanced support vector machine for diagnosis of breast cancer based on high-level features," *Bmc Bioinformatics,* vol. 20, pp. 10.1186/s12859-019-2771-z, Jun 10 2019.

[24] B. Cao, J. Zhao, Y. Gu, Y. Ling, and X. Ma, "Applying graph-based differential grouping for multiobjective large-scale optimization," *Swarm and Evolutionary Computation,* vol. 53, p. 100626, 2020.

[25] B. Cao, S. Fan, J. Zhao, P. Yang, K. Muhammad, and M. Tanveer, "Quantum-enhanced multiobjective large-scale optimization via parallelism," *Swarm and Evolutionary Computation,* vol. 57, p. 100697, 2020/09/01/ 2020.

[26] S. Qu, Y. Han, Z. Wu, and H. Raza, "Consensus Modeling with Asymmetric Cost Based on Data-Driven Robust Optimization," *Group Decision and Negotiation,* pp. 1-38, 2020.

[27] X. Fu, P. Pace, G. Aloi, L. Yang, and G. Fortino, "Topology Optimization Against Cascading Failures on Wireless Sensor Networks Using a Memetic Algorithm," *Computer Networks,* p. 107327, 2020.

[28] H. Chen, A. A. Heidari, X. Zhao, L. Zhang, and H. Chen, "Advanced orthogonal learning-driven multi-swarm sine cosine optimization: Framework and case studies," *Expert Systems with Applications,* vol. 144, p. 113113, 2020.

[29] M. Abd Elaziz, A. A. Heidari, H. Fujita, and H. Moayedi, "A competitive chain-based Harris Hawks Optimizer for global optimization and multi-level image thresholding problems," *Applied Soft Computing,* p. 106347, 2020.

[30] C. Yu, A. A. Heidari, and H. Chen, "A Quantum-behaved Simulated Annealing Enhanced Moth-flame Optimization Method," *Applied Mathematical Modelling,* 2020.

[31] S. Jiao, G. Chong, C. Huang, H. Hu, M. Wang, A. A. Heidari, *et al.*, "Orthogonally adapted Harris Hawk Optimization for parameter estimation of photovoltaic models," *Energy,* p. 117804, 2020.

[32] S. Gupta, K. Deep, A. A. Heidari, H. Moayedi, and M. Wang, "Opposition-based Learning Harris Hawks Optimization with Advanced Transition Rules: Principles and Analysis," *Expert Systems with Applications,* p. 113510, 2020.

[33] M. Wang, A. A. Heidari, M. Chen, H. Chen, X. Zhao, and X. Cai, "Exploratory Differential Ant Lion-based Optimization," *Expert Systems with Applications,* p. 113548 (https://doi.org/10.1016/j.eswa.2020.113548), 2020/05/19/ 2020.

[34] I. Aljarah, M. Habib, H. Faris, N. Al-Madi, A. A. Heidari, M. Mafarja, *et al.*, "A Dynamic Locality Multi-Objective Salp Swarm Algorithm for Feature Selection," *Computers & Industrial Engineering,* p. 106628, 2020.

[35] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Generation Computer Systems,* vol. 111, pp. 300-323, 2020/10/01/ 2020.

[36] Z. Xu, Z. Hu, A. A. Heidari, M. Wang, X. Zhao, H. Chen, *et al.*, "Orthogonally-designed adapted grasshopper optimization: A comprehensive analysis," *Expert Systems with Applications,* vol. 150, p. 113282, 2020.

[37] H. Chen, C. Yang, A. A. Heidari, and X. Zhao, "An Efficient Double Adaptive Random Spare Reinforced Whale Optimization Algorithm," *Expert Systems with Applications,* p. 113018, 2019/10/13/ 2019.

[38] N. Zhang, "Semistability of Steepest Descent with Momentum for Quadratic Functions," *Neural Computation,* vol. 25, pp. 1277-1301, May 2013.

[39] N. Zhang, "A Study on the Optimal Double Parameters for Steepest Descent with Momentum," *Neural Computation,* vol. 27, pp. 982-1004, Apr 2015.

[40] X. Zhang, D. Wang, Z. Zhou, Y. J. I. t. o. p. a. Ma, and m. intelligence, "Robust low-rank tensor recovery with rectification and alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 43, pp. 238-255, 2019.

[41] A. A. Heidari, I. Aljarah, H. Faris, H. Chen, J. Luo, and S. Mirjalili, "An enhanced associative learning-based exploratory whale optimizer for global optimization," *Neural Computing and Applications,* January 29 2019.

[42] A. A. Heidari, R. A. Abbaspour, and A. R. Jordehi, "An efficient chaotic water cycle algorithm for optimization tasks," *Neural Computing and Applications,* vol. 28, pp. 57-85, 2017.

[43] A. A. Heidari, H. Faris, I. Aljarah, and S. Mirjalili, "An efficient hybrid multilayer perceptron neural network with grasshopper optimization," *Soft Computing,* pp. 1-18, 2018.

[44] H. Faris, A.-Z. Ala'M, A. A. Heidari, I. Aljarah, M. Mafarja, M. A. Hassonah, *et al.*, "An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks," *Information Fusion,* vol. 48, pp. 67-83, 2019.

[45] R. Abbassi, A. Abbassi, A. A. Heidari, and S. Mirjalili, "An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models," *Energy Conversion and Management,* vol. 179, pp. 362-372, 2019.

[46] A. A. Heidari, R. A. Abbaspour, and A. R. Jordehi, "Gaussian bare-bones water cycle algorithm for optimal reactive power dispatch in electrical power systems," *Applied Soft Computing,* vol. 57, pp. 657-671, 2017.

[47] Y. Zhang, R. Liu, A. Asghar Heidari, X. Wang, Y. Chen, M. Wang, *et al.*, "Towards Augmented

Kernel Extreme Learning Models for Bankruptcy Prediction: Algorithmic Behavior and Comprehensive Analysis," *Neurocomputing,* p. https://doi.org/10.1016/j.neucom.2020.10.038, 2020/10/22/ 2020.

[48] X. Wang, H. Chen, A. A. Heidari, X. Zhang, J. Xu, Y. Xu*, et al.*, "Multi-population following behavior-driven fruit fly optimization: A Markov chain convergence proof and comprehensive analysis," *Knowledge-Based Systems,* vol. 210, p. 106437 (https://doi.org/10.1016/j.knosys.2020.106437), 2020/12/27/ 2020.

[49] M. Wang, X. Zhao, A. A. Heidari, and H. Chen, "Evaluation of constraint in photovoltaic models by exploiting an enhanced ant lion optimizer," *Solar Energy,* vol. 211, pp. 503-521, 2020/11/15/ 2020.

[50] H. M. Ridha, C. Gomes, H. Hizam, M. Ahmadipour, A. A. Heidari, and H. Chen, "Multi-objective optimization and multi-criteria decision-making methods for optimal design of standalone photovoltaic system: A comprehensive review," *Renewable and Sustainable Energy Reviews,* vol. 135, p. 110202.

[51] A. Abbassi, R. Abbassi, A. A. Heidari, D. Oliva, H. Chen, A. Habib*, et al.*, "Parameters identification of photovoltaic cell models using enhanced exploratory salp chains-based approach," *Energy,* p. 117333, 2020.

[52] H. Zhang, A. A. Heidari, M. Wang, L. Zhang, H. Chen, and C. Li, "Orthogonal Nelder-Mead moth flame method for parameters identification of photovoltaic modules," *Energy Conversion and Management,* vol. 211, p. 112764, 2020.

[53] X. Zhang, Y. Xu, C. Yu, A. A. Heidari, S. Li, H. Chen*, et al.*, "Gaussian mutational chaotic fruit fly-built optimization and feature selection," *Expert Systems with Applications,* vol. 141, p. 112976, 2020/03/01/ 2020.

[54] M. Eshtay, H. Faris, A. A. Heidari, A.-Z. Ala'M, and I. Aljarah, "AutoRWN: automatic construction and training of random weight networks using competitive swarm of agents," *Neural Computing and Applications,* pp. 1-18, 2020.

[55] R. I. Yaghi, H. Faris, I. Aljarah, A.-Z. Ala'M, A. A. Heidari, and S. Mirjalili, "Link Prediction Using Evolutionary Neural Network Models," in *Evolutionary Machine Learning Techniques*, ed: Springer, 2020, pp. 85-111.

[56] M.-L. Sun, Y. Liu, G. Liu, D. Cui, A. A. Heidari, W.-Y. Jia*, et al.*, "Application of Machine Learning to Stomatology: A Comprehensive Review," *IEEE Access,* vol. 8, pp. 184360-184374, 2020.

[57] K. Shi, J. Wang, Y. Tang, and S. Zhong, "Reliable asynchronous sampled-data filtering of T–S fuzzy uncertain delayed neural networks with stochastic switched topologies," *Fuzzy Sets and Systems,* vol. 381, pp. 1-25, 2020.

[58] D. Zhao, L. Liu, F. Yu, A. A. Heidari, M. Wang, G. Liang*, et al.*, "Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy," *Knowledge-Based Systems,* p. 106510 (https://doi.org/10.1016/j.knosys.2020.106510), 2020/10/22/ 2020.

[59] D. Zhao, L. Liu, F. Yu, A. A. Heidari, M. Wang, D. Oliva*, et al.*, "Ant Colony Optimization with Horizontal and Vertical Crossover Search: Fundamental Visions for Multi-threshold Image Segmentation," *Expert Systems with Applications,* p. 114122, 2020.

[60] B. Shi, A. A. Heidari, C. Chen, M. Wang, C. Huang, H. Chen*, et al.*, "Predicting Di-2-Ethylhexyl Phthalate Toxicity: Hybrid Integrated Harris Hawks Optimization With Support Vector Machines," *IEEE Access,* vol. 8, pp. 161188-161202, 2020.

[61] X. Zhao, X. Zhang, Z. Cai, X. Tian, X. Wang, Y. Huang*, et al.*, "Chaos enhanced grey wolf

optimization wrapped ELM for diagnosis of paraquat-poisoned patients," *Computational Biology and Chemistry,* vol. 78, pp. 481-490, Feb 2019.

[62] L. Hu, H. Li, Z. Cai, F. Lin, G. Hong, H. Chen*, et al.*, "A new machine-learning method to prognosticate paraquat poisoned patients by combining coagulation, liver, and kidney indices," *Plos One,* vol. 12, p. e0186427, 2017.

[63] M. Wang and H. Chen, "Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis," *Applied Soft Computing,* vol. 88, p. 105946 (https://doi.org/10.1016/j.asoc.2019.105946), 2020/03/01/ 2020.

[64] M. Wang, H. Chen, B. Yang, X. Zhao, L. Hu, Z. Cai*, et al.*, "Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses," *Neurocomputing,* vol. 267, pp. 69-84, Dec 6 2017.

[65] L. Shen, H. Chen, Z. Yu, W. Kang, B. Zhang, H. Li*, et al.*, "Evolving support vector machines using fruit fly optimization for medical data classification," *Knowledge-Based Systems,* vol. 96, pp. 61-75, 3/15/ 2016.

[66] G.-q. Zeng, Y.-z. Lu, and W.-J. J. o. Z. U. S. C. Mao, "Modified extremal optimization for the hard maximum satisfiability problem," *Journal of Zhejiang University SCIENCE C,* vol. 12, pp. 589-596, 2011.

[67] G. Zeng, Y. Lu, Y. Dai, Z. Wu, W. Mao, Z. Zhang*, et al.*, "Backbone guided extremal optimization for the hard maximum satisfiability problem," *International Journal of Innovative Computing Information and Control,* vol. 8, pp. 8355-8366, 2012.

[68] G.-Q. Zeng, K.-D. Lu, Y.-X. Dai, Z.-J. Zhang, M.-R. Chen, C.-W. Zheng*, et al.*, "Binary-coded extremal optimization for the design of PID controllers," *Neurocomputing,* vol. 138, pp. 180-188, 2014.

[69] G.-Q. Zeng, J. Chen, Y.-X. Dai, L.-M. Li, C.-W. Zheng, and M.-R. J. N. Chen, "Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization," *Neurocomputing* vol. 160, pp. 173-184, 2015.

[70] G.-Q. Zeng, X.-Q. Xie, M.-R. Chen, and J. Weng, "Adaptive population extremal optimization-based PID neural network for multivariable nonlinear control systems," *Swarm and Evolutionary Computation,* vol. 44, pp. 320-334, 2019/02/01/ 2019.

[71] M. Chen, G. Zeng, K. Lu, and J. Weng, "A Two-Layer Nonlinear Combination Method for Short-Term Wind Speed Prediction Based on ELM, ENN, and LSTM," *IEEE Internet of Things Journal,* vol. 6, pp. 6997-7010, 2019.

[72] J. Pang, H. Zhou, Y.-C. Tsai, and F.-D. Chou, "A scatter simulated annealing algorithm for the bi-objective scheduling problem for the wet station of semiconductor manufacturing," *Computers & Industrial Engineering,* vol. 123, pp. 54-66, 2018/09/01/ 2018.

[73] H. Zhou, J. Pang, P.-K. Chen, and F.-D. Chou, "A modified particle swarm optimization algorithm for a batch-processing machine scheduling problem with arbitrary release times and non-identical job sizes," *Computers & Industrial Engineering,* vol. 123, pp. 67-81, 2018/09/01/ 2018.

[74] W. Deng, H. Liu, J. Xu, H. Zhao, Y. J. I. T. o. I. Song, and Measurement, "An improved quantum-inspired differential evolution algorithm for deep belief network," *IEEE Transactions on Instrumentation & Measurement,* p. https://doi.org/10.1109/TIM.2020.2983233, 2020.

[75] H. Zhao, H. Liu, J. Xu, W. J. I. T. o. i. Deng, and measurement, "Performance prediction using high-order differential mathematical morphology gradient spectrum entropy and extreme learning machine," *IEEE Transactions on Instrumentation & Measurement,* p.

https://doi.org/10.1109/TIM.2019.2948414, 2019.

[76] W. Deng, J. Xu, H. Zhao, and Y. Song, "A Novel Gate Resource Allocation Method Using Improved PSO-Based QEA," *IEEE Transactions on Intelligent Transportation Systems,* p. http://doi.org/10.1109/TITS.2020.3025796 2020.

[77] D. W, X. JJ, S. YJ, and Z. HM, "An Effective Improved Co-evolution Ant Colony Optimization Algorithm with Multi-Strategies and Its Application," *International Journal of Bio-Inspired Computation,* pp. 16(3): 158–170, 05/31 2020.

[78] Y. Wei, H. Lv, M. Chen, M. Wang, A. A. Heidari, H. Chen*, et al.*, "Predicting entrepreneurial intention of students: An extreme learning machine with Gaussian barebone Harris hawks optimizer," *IEEE Access,* vol. 8, pp. 76841-76855, 2020.

[79] W. Zhu, C. Ma, X. Zhao, M. Wang, A. A. Heidari, H. Chen*, et al.*, "Evaluation of sino foreign cooperative education project using orthogonal sine cosine optimized kernel extreme learning machine," *IEEE Access,* vol. 8, pp. 61107-61123, 2020.

[80] A. Lin, Q. Wu, A. A. Heidari, Y. Xu, H. Chen, W. Geng*, et al.*, "Predicting intentions of students for master programs using a chaos-induced sine cosine-based fuzzy K-nearest neighbor classifier," *Ieee Access,* vol. 7, pp. 67235-67248, 2019.

[81] J. Tu, A. Lin, H. Chen, Y. Li, and C. Li, "Predict the entrepreneurial intention of fresh graduate students based on an adaptive support vector machine framework," *Mathematical Problems in Engineering,* vol. 2019, pp. 1-16, 2019.

[82] Y. Wei, N. Ni, D. Liu, H. Chen, M. Wang, Q. Li*, et al.*, "An improved grey wolf optimization strategy enhanced SVM and its application in predicting the second major," *Mathematical Problems in Engineering,* vol. 2017, pp. 1-12, 2017.

[83] Y. Zhang, R. Liu, A. A. Heidari, X. Wang, Y. Chen, M. Wang*, et al.*, "Towards augmented kernel extreme learning models for bankruptcy prediction: Algorithmic behavior and comprehensive analysis," *Neurocomputing,* p. https://doi.org/10.1016/j.neucom.2020.10.038, 2020/10/22/ 2020.

[84] C. Yu, M. Chen, K. Cheng, X. Zhao, C. Ma, F. Kuang*, et al.*, "SGOA: annealing-behaved grasshopper optimizer for global tasks," *Engineering with Computers,* pp. 10.1007/s00366-020-01234-1, 2021/01/03 2021.

[85] Z. Cai, J. Gu, J. Luo, Q. Zhang, H. Chen, Z. Pan*, et al.*, "Evolving an optimal kernel extreme learning machine by using an enhanced grey wolf optimization strategy," *Expert Systems with Applications,* vol. 138, p. 112814, 2019/12/30/ 2019.

[86] Y. Xu, H. Chen, A. A. Heidari, J. Luo, Q. Zhang, X. Zhao*, et al.*, "An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks," *Expert Systems with Applications,* vol. 129, pp. 135-155, 2019/09/01/ 2019.

[87] J. Luo, H. Chen, Y. Xu, H. Huang, and X. Zhao, "An improved grasshopper optimization algorithm with application to financial stress prediction," *Applied Mathematical Modelling,* vol. 64, pp. 654-668, 2018.

[88] M. Wang, H. Chen, H. Li, Z. Cai, X. Zhao, C. Tong*, et al.*, "Grey wolf optimization evolving kernel extreme learning machine: Application to bankruptcy prediction," *Engineering Applications of Artificial Intelligence,* vol. 63, pp. 54-68, 2017.

[89] X. Liang, Z. Cai, M. Wang, X. Zhao, H. Chen, and C. Li, "Chaotic oppositional sine–cosine method for solving global optimization problems," *Engineering with Computers,* pp. 10.1007/s00366-020-01083-y, 2020/07/12 2020.

[90] A. F. Ba, H. Huang, M. Wang, X. Ye, Z. Gu, H. Chen*, et al.*, "Levy-based antlion-inspired optimizers with orthogonal learning scheme," *Engineering with Computers,* pp. 10.1007/s00366-020-01042-7, 2020/06/03 2020.

[91] X. Zhao, D. Li, B. Yang, C. Ma, Y. Zhu, and H. Chen, "Feature selection based on improved ant colony optimization for online detection of foreign fiber in cotton," *Applied Soft Computing,* vol. 24, pp. 585-596, 2014.

[92] Z. Chen, M. Lu, Y. Zhou, and C. Chen, "Information synergy entropy based multi-feature information fusion for the operating condition identification in aluminium electrolysis," *Information Sciences,* vol. 548, pp. 275-294, 2021/02/16/ 2021.

[93] A. A. Heidari and P. Pahlavani, "An efficient modified grey wolf optimizer with Lévy flight for optimization tasks," *Applied Soft Computing Journal,* vol. 60, pp. 115-134, 2017.

[94] M. Mafarja, I. Aljarah, A. A. Heidari, H. Faris, P. Fournier-Viger, X. Li*, et al.*, "Binary dragonfly optimization for feature selection using time-varying transfer functions," *Knowledge-Based Systems,* vol. 161, pp. 185-204, 2018.

[95] M. Mafarja, I. Aljarah, A. A. Heidari, A. I. Hammouri, H. Faris, A. M. Al-Zoubi*, et al.*, "Evolutionary Population Dynamics and Grasshopper Optimization approaches for feature selection problems," *Knowledge-Based Systems,* vol. 145, pp. 25-45, 2018.

[96] H. Faris, A. M. Al-Zoubi, A. A. Heidari, I. Aljarah, M. Mafarja, M. A. Hassonah*, et al.*, "An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks," *Information Fusion,* vol. 48, pp. 67-83, 2019.

[97] Y. Cao, Y. Li, G. Zhang, K. Jermsittiparsert, and M. Nasseri, "An efficient terminal voltage control for PEMFC based on an improved version of whale optimization algorithm," *Energy Reports,* vol. 6, pp. 530-542, 2020.

[98] M. A. Tawhid and K. B. Dsouza, "Hybrid Binary Bat Enhanced Particle Swarm Optimization Algorithm for solving feature selection problems," *Applied Computing and Informatics,* 2018/04/11/ 2018.

[99] J. Liu, C. Wu, G. Wu, and X. Wang, "A novel differential search algorithm and applications for structure design," *Applied Mathematics and Computation,* vol. 268, pp. 246-269, 2015.

[100] L. Yang and H. Chen, "Fault diagnosis of gearbox based on RBF-PF and particle swarm optimization wavelet neural network," *Neural computing and applications,* vol. 31, pp. 4463-4478, 2019.

[101] H. Chen, D. L. Fan, L. Fang, W. Huang, J. Huang, C. Cao*, et al.*, "Particle swarm optimization algorithm with mutation operator for particle filter noise reduction in mechanical fault diagnosis," *International Journal of Pattern Recognition and Artificial Intelligence,* p. 2058012, 2020.

[102] G. Sun, B. Yang, Z. Yang, and G. Xu, "An adaptive differential evolution with combined strategy for global numerical optimization," *Soft Computing,* pp. 1-20, 2019.

[103] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Generation Computer Systems,* vol. https://doi.org/10.1016/j.future.2020.03.055, 2020/04/03/ 2020.

[104] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems-the International Journal of Escience,* vol. 97, pp. 849-872, Aug 2019.

[105] S. Song, P. Wang, A. A. Heidari, M. Wang, X. Zhao, H. Chen*, et al.*, "Dimension decided Harris

hawks optimization with Gaussian mutation: Balance analysis and diversity patterns," *Knowledge-Based Systems,* p. 106425 (https://doi.org/10.1016/j.knosys.2020.106425), 2020/10/13/ 2020.

[106] M. Mafarja, A. A. Heidari, M. Habib, H. Faris, T. Thaher, and I. Aljarah, "Augmented whale feature selection for IoT attacks: Structure, analysis and applications," *Future Generation Computer Systems,* vol. 112, pp. 18-40, 2020/11/01/ 2020.

[107] L.-M. Li, K.-D. Lu, G.-Q. Zeng, L. Wu, and M.-R. Chen, "A novel real-coded population-based extremal optimization algorithm with polynomial mutation: A non-parametric statistical study on continuous optimization problems," *Neurocomputing,* vol. 174, pp. 577-587, 2016/01/22/ 2016.

[108] G.-Q. Zeng, J. Chen, L.-M. Li, M.-R. Chen, L. Wu, Y.-X. Dai*, et al.*, "An improved multi-objective population-based extremal optimization algorithm with polynomial mutation," *Information Sciences,* vol. 330, pp. 49-73, 2016/02/10/ 2016.

[109] M.-R. Chen, J.-H. Chen, G.-Q. Zeng, K.-D. Lu, and X.-F. Jiang, "An improved artificial bee colony algorithm combined with extremal optimization and Boltzmann Selection probability," *Swarm and Evolutionary Computation,* vol. 49, pp. 158-177, 2019/09/01/ 2019.

[110] G.-G. Wang, S. Deb, X. Zhao, and Z. Cui, "A new monarch butterfly optimization with an improved crossover operator," *Operational Research: An International Journal,* vol. 18, pp. 731-755, 2018.

[111] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing and Applications,* vol. 31, pp. 1995-2014, 2019.

[112] Y. Feng, S. Deb, G.-G. Wang, and A. H. Alavi, "Monarch butterfly optimization: a comprehensive review," *Expert Systems with Applications,* vol. 168, p. 114418, 2021.

[113] G.-G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing,* vol. 10, pp. 151-164, 2018.

[114] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Systems,* vol. 89, pp. 228-249, 2015/11/01/ 2015.

[115] D. K. Kotary and S. J. Nanda, "Distributed robust data clustering in wireless sensor networks using diffusion moth flame optimization," *Engineering Applications of Artificial Intelligence,* vol. 87, Jan 2020.

[116] X. Lei, M. Fang, and H. Fujita, "Moth-flame optimization-based algorithm with synthetic dynamic PPI networks for discovering protein complexes," *Knowledge-Based Systems,* vol. 172, pp. 76-85, May 15 2019.

[117] C. Li, L. Hou, B. Y. Sharma, H. Li, C. Chen, Y. Li*, et al.*, "Developing a new intelligent system for the diagnosis of tuberculous pleural effusion," *Computer Methods and Programs in Biomedicine,* vol. 153, pp. 211-225, 2018.

[118] A. A. Elsakaan, R. A. El-Sehiemy, S. S. Kaddah, and M. I. Elsaid, "An enhanced moth-flame optimizer for solving non-smooth economic dispatch problems with emissions," *Energy,* vol. 157, pp. 1063-1078, 2018.

[119] G. I. Sayed and A. E. Hassanien, "Moth-flame swarm optimization with neutrosophic sets for automatic mitosis detection in breast cancer histology images," *Applied Intelligence,* vol. 47, pp. 397-408, 2017.

[120] R. Ng Shin Mei, M. H. Sulaiman, Z. Mustaffa, and H. Daniyal, "Optimal reactive power dispatch solution by loss minimization using moth-flame optimization technique," *Applied Soft*

Computing Journal, vol. 59, pp. 210-222, 2017.

[121] A. A. Heidari, Y. Yin, M. Mafarja, S. M. J. Jalali, J. S. Dong, and S. Mirjalili, "Efficient Moth-Flame-Based Neuroevolution Models," in *Evolutionary Machine Learning Techniques*, ed: Springer, 2020, pp. 51-66.

[122] H. Zhang, R. Li, Z. Cai, Z. Gu, A. A. Heidari, M. Wang*, et al.*, "Advanced Orthogonal Moth Flame Optimization with Broyden–Fletcher–Goldfarb–Shanno Algorithm: Framework and Real-world Problems," *Expert Systems with Applications,* p. 113617, 2020.

[123] A. E. Hassanien, T. Gaber, U. Mokhtar, and H. Hefny, "An improved moth flame optimization algorithm based on rough sets for tomato diseases detection," *Computers and Electronics in Agriculture,* vol. 136, pp. 86-96, 2017.

[124] L. Zhang, K. Mistry, S. C. Neoh, and C. P. Lim, "Intelligent facial emotion recognition using moth-firefly optimization," *Knowledge-Based Systems,* vol. 111, pp. 248-267, 2016.

[125] C. Li, S. Li, and Y. Liu, "A least squares support vector machine model optimized by moth-flame optimization algorithm for annual power load forecasting," *Applied Intelligence,* vol. 45, pp. 1166-1178, 2016.

[126] D. Allam, D. A. Yousri, and M. B. Eteiba, "Parameters extraction of the three diode model for the multi-crystalline solar cell/module using Moth-Flame Optimization Algorithm," *Energy Conversion and Management,* vol. 123, pp. 535-548, 2016.

[127] W. Shan, Z. Qiao, A. A. Heidari, H. Chen, H. Turabieh, and Y. Teng, "Double adaptive weights for stabilization of moth flame optimizer: Balance analysis, engineering cases, and medical diagnosis," *Knowledge-Based Systems,* vol. 214, p. 106728, 2021/02/28/ 2021.

[128] J. Luo, H. Chen, Z. Hu, H. Huang, P. Wang, X. Wang*, et al.*, "A new kernel extreme learning machine framework for somatization disorder diagnosis," *IEEE Access,* vol. 7, pp. 45512-45525, 2019.

[129] M. Shehab, H. Alshawabkah, L. Abualigah, and N. Al-Madi, "Enhanced a hybrid moth-flame optimization algorithm using new selection schemes," *Engineering with Computers,* 2020.

[130] D. Pelusi, R. Mascella, L. Tallini, J. Nayak, B. Naik, and Y. Deng, "An Improved Moth-Flame Optimization algorithm with hybrid search phase," *Knowledge-Based Systems,* vol. 191, Mar 5 2020.

[131] K. Kaur, U. Singh, and R. Salgotra, "An enhanced moth flame optimization," *Neural Computing & Applications,* vol. 32, pp. 2315-2349, Apr 2020.

[132] Y. Xu, H. Chen, J. Luo, Q. Zhang, S. Jiao, and X. Zhang, "Enhanced Moth-flame optimizer with mutation strategy for global optimization," *Information Sciences,* vol. 492, pp. 181-203, 2019.

[133] S. Sapre and S. Mini, "Opposition-based moth flame optimization with Cauchy mutation and evolutionary boundary constraint handling for global optimization," *Soft Computing,* vol. 23, pp. 6023-6041, Aug 2019.

[134] L. Xu, Y. Li, K. Li, G. H. Beng, Z. Jiang, C. Wang*, et al.*, "Enhanced Moth-flame Optimization Based on Cultural Learning and Gaussian Mutation," *Journal of Bionic Engineering,* vol. 15, pp. 751-763, 2018.

[135] R. Santos, G. Borges, A. Santos, M. s. Silva, C. Sales, and J. o. C. W. A. Costa, "A rotationally invariant semi-autonomous particle swarm optimizer with directional diversity," *Swarm and Evolutionary Computation,* vol. 56, p. 100700, 2020/08/01/ 2020.

[136] V. Savsani and M. A. Tawhid, "Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems," *Engineering Applications of Artificial Intelligence,* vol. 63, pp. 20-32,

2017.

[137] S. Mirjalili, "SCA: A Sine Cosine Algorithm for solving optimization problems," *Knowledge-Based Systems,* vol. 96, pp. 120-133, 2016/03/15/ 2016.

[138] M. Črepinšek, S.-h. Liu, and M. Mernik, *Exploration and Exploitation in Evolutionary Algorithms: A Survey* vol. 45, 2013.

[139] Q. Dong, L. Cui, and S. Si, "Reliability and availability analysis of stochastic degradation systems based on bivariate Wiener processes," *Applied Mathematical Modelling,* vol. 79, pp. 414-433, 2020.

[140] H. Chen, M. Wang, and X. Zhao, "A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems," *Applied Mathematics and Computation,* vol. 369, p. 124872, 2020/03/15/ 2020.

[141] J. Tu, A. Lin, H. Chen, Y. Li, and C. Li, "Predict the Entrepreneurial Intention of Fresh Graduate Students Based on an Adaptive Support Vector Machine Framework," *Mathematical Problems in Engineering,* vol. 2019 pp. 1-16, 2019.

[142] H. Huang, X. Feng, A. A. Heidari, Y. Xu, M. Wang, G. Liang*, et al.*, "Rationalized Sine Cosine Optimization With Efficient Searching Patterns," *IEEE Access,* vol. 8, pp. 61471-61490, 2020.

[143] Y. Fan, P. Wang, A. Asghar Heidari, M. Wang, X. Zhao, H. Chen*, et al.*, "Rationalized Fruit Fly Optimization with Sine Cosine Algorithm: A Comprehensive Analysis," *Expert Systems with Applications,* pp. 113486, https://doi.org/10.1016/j.eswa.2020.113486, 2020/04/29/ 2020.

[144] Y. Ji, J. Tu, H. Zhou, W. Gui, G. Liang, H. Chen*, et al.*, "An Adaptive Chaotic Sine Cosine Algorithm for Constrained and Unconstrained Optimization," *Complexity,* vol. 2020, p. 6084917 (https://doi.org/10.1155/2020/6084917), 2020/10/31 2020.

[145] W. Zhou, P. Wang, A. A. Heidari, M. Wang, X. Zhao, and H. Chen, "Multi-core sine cosine optimization: Methods and inclusive analysis," *Expert Systems with Applications,* vol. 164, p. 113974, 2021/02/01/ 2021.

[146] H. Chen, S. Jiao, A. A. Heidari, M. Wang, X. Chen, and X. Zhao, "An opposition-based sine cosine approach with local search for parameter estimation of photovoltaic models," *Energy Conversion and Management,* vol. 195, pp. 927-942, 2019.

[147] G. Liu, W. Jia, M. Wang, A. A. Heidari, H. Chen, Y. Luo*, et al.*, "Predicting Cervical Hyperextension Injury: A Covariance Guided Sine Cosine Support Vector Machine," *IEEE Access,* vol. 8, pp. 46895-46908, 2020.

[148] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonprametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithm," *Swarm and Evolutionary Computation,* vol. 1, pp. 3-18, 03/01 2011.

[149] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences,* vol. 180, pp. 2044-2064, 05/01 2010.

[150] S. Mirjalili, I. Aljarah, M. Mafarja, A. A. Heidari, and H. Faris, "Grey Wolf Optimizer: Theory, Literature Review, and Application in Computational Fluid Dynamics Problems," in *Nature-Inspired Optimizers: Theories, Literature Reviews and Applications*, S. Mirjalili, J. Song Dong, and A. Lewis, Eds., ed Cham: Springer International Publishing, 2020, pp. 87-105.

[151] I. Aljarah, M. Mafarja, A. A. Heidari, H. Faris, and S. Mirjalili, "Multi-verse Optimizer: Theory, Literature Review, and Application in Data Clustering," ed, 2020, pp. 123-141.

[152] X.-S. Yang, *Firefly Algorithms for Multimodal Optimization* vol. 5792, 2010.

[153] X.-S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," vol. 284, 04/23 2010.

[154] A. Gandomi, X.-S. Yang, and A. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering With Computers,* vol. 29, pp. 1-19, 04/01 2013.

[155] N. Hansen and A. Ostermeier, "Completely Derandomized Self-Adaptation in Evolution Strategies," *Evolutionary Computation,* vol. 9, pp. 159-195, 2001.

[156] L. Hongwei, L. Jianyong, C. Liang, B. Jingbo, S. Yangyang, and L. Kai, "Chaos-enhanced moth-flame optimization algorithm for global optimization," *Journal of Systems Engineering and Electronics,* vol. 30, pp. 1144-1159, 2019.

[157] A. Lin, Q. Wu, A. A. Heidari, Y. Xu, H. Chen, W. Geng*, et al.*, "Predicting Intentions of Students for Master Programs Using a Chaos-Induced Sine Cosine-Based Fuzzy K-Nearest Neighbor Classifier," *IEEE Access,* vol. 7, pp. 67235-67248, 2019.

[158] G. I. Sayed, G. Khoriba, and M. H. Haggag, "A novel chaotic salp swarm algorithm for global optimization and feature selection," *Applied Intelligence,* vol. 48, pp. 3462-3481, Oct 2018.

[159] M. S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, and G. Pasi, "Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization," *Swarm and Evolutionary Computation,* vol. 39, pp. 70-85, Apr 2018.

[160] M. Elsayed abd el aziz, D. Oliva, and S. Xiong, "An Improved Opposition-Based Sine Cosine Algorithm for Global Optimization," *Expert Systems with Applications,* vol. 90, 08/15 2017.

[161] G. G. Wang, "Adaptive Response Surface Method Using Inherited Latin Hypercube Design Points," *Journal of Mechanical Design,* vol. 125, pp. 210-220, 2003.

[162] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers,* vol. 29, pp. 17-35, 2013/01/01 2013.

[163] M.-Y. Cheng and D. Prayogo, "Symbiotic Organisms Search: A new metaheuristic optimization algorithm," *Computers & Structures,* vol. 139, pp. 98-112, 2014/07/15/ 2014.

[164] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers & Structures,* vol. 110-111, pp. 151-166, 2012/11/01/ 2012.

[165] P. Savsani and V. Savsani, "Passing vehicle search (PVS): A novel metaheuristic algorithm," *Applied Mathematical Modelling,* vol. 40, pp. 3951-3978, 2016/03/01/ 2016.

[166] R. Venkata Rao, V. Savsani, and D. Vakharia, "Teaching-Learning-Based Optimization: A novel method for constrained mechanical design optimization problems," *Computer-Aided Design,* vol. 43, pp. 303-315, 03/01 2011.

[167] X. Fei, J. Wang, S. Ying, Z. Hu, and J. Shi, "Projective parameter transfer based sparse multiple empirical kernel learning Machine for diagnosis of brain disease," *Neurocomputing,* vol. 413, pp. 271-283, 2020/11/06/ 2020.

[168] S. Wang, J. Xiang, Y. Zhong, and Y. Zhou, "Convolutional neural network-based hidden Markov models for rolling element bearing fault identification," *Knowledge-Based Systems,* vol. 144, pp. 65-76, 2018/03/15/ 2018.

[169] S. Y. Wei, T. C. Chiao, and X. W. Ye, "The clustering analysis of corporate ownership and control contestability based on Shapley value," *Cluster Computing-the Journal of Networks Software Tools and Applications,* vol. 20, pp. 2703-2723, Sep 2017.

[170] Y. Wang, J. See, Y.-H. Oh, R. C. W. Phan, Y. Rahulamathavan, H.-C. Ling*, et al.*, "Effective

recognition of facial micro-expressions with video motion magnification," *Multimedia Tools and Applications,* vol. 76, pp. 21665-21690, 2017/10/01 2017.

[171] X. Xue, S. F. Wang, L. J. Zhan, Z. Y. Feng, and Y. D. Guo, "Social Learning Evolution (SLE): Computational Experiment-Based Modeling Framework of Social Manufacturing," *Ieee Transactions on Industrial Informatics,* vol. 15, pp. 3343-3355, Jun 2019.

[172] X. Zhang, T. Wang, W. Luo, P. J. I. T. o. C. Huang, and S. f. V. Technology, "Multi-level Fusion and Attention-guided CNN for Image Dehazing," *IEEE Transactions on Circuits and Systems for Video Technology,* p. DOI: 10.1109/TCSVT.2020.3046625, 2020.

Highlights (for review)

1. Sine cosine strategy is introduced into moth-flame optimization algorithm
2. The proposed method is compared with representative algorithms on lots of functions
3. The proposed method can solve engineering design problems efficiently