Information Sciences 492 (2019) 181–203

Contents lists available at ScienceDirect

Information Sciences

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

Enhanced Moth-flame optimizer with mutation strategy for global optimization

Yueting Xu, Huiling Chen*, Jie Luo, Qian Zhang, Shan Jiao, Xiaoqin Zhang

Department of Computer Science, Wenzhou University, Wenzhou, Zhejiang 325035, China

ARTICLE INFO

Article history:
Received 14 June 2018
Revised 8 April 2019
Accepted 12 April 2019
Available online 12 April 2019

Keywords:
Moth-flame optimization
Gaussian mutation
Cauchy mutation
Lévy flight

ABSTRACT

Moth-flame optimization (MFO) is a widely used nature-inspired algorithm characterized by a simple structure with simple parameters. However, for some complex optimization tasks, especially the high dimensional and multimodal problems, MFO may have problems with convergence or tend to fall into local optima. To overcome these limitations, here a series of new variants of MFO are proposed by combining MFO with Gaussian mutation (GM), Cauchy mutation (CM), Lévy mutation (LM) or the combination of GM, CM and LM. Specifically, GM is introduced into the basic MFO to improve neighborhood-informed capability. Then, CM with a large mutation step is adopted to enhance global exploration ability. Finally, LM is embedded to increase the randomness of search agents' movement. The best variant of MFO was compared to 15 state-of-the-art algorithms and 4 well-known advanced optimization approaches on a comprehensive set of 23 benchmark problems and 30 CEC2017 benchmark tasks. The experimental results demonstrate that the three strategies can significantly boost exploration and exploitation capabilities of the basic MFO.

© 2019 Elsevier Inc. All rights reserved.

1. Introduction

Swarm intelligence optimizers have received increasing attention in recent years [1–8]. As a recently developed swarm intelligence algorithm, Moth-flame Optimization (MFO) [9] is known for its simple structure with simple parameters. Moreover, because MFO is robust, efficient and easy to implement in many computer languages, it has been used widely in many real-world optimization applications. For instance, MFO was applied to the parameter extraction process of tested models as proposed by Allam et al. [10]. Li et al. [11] used MFO to optimize LSSVM's parameters (LSSVM–MFO) and applied it to forecast the hybrid annual power load mode. A hybrid electricity consumption forecasting method was proposed in which MFO was used to tune the parameters in the grey model with a rolling mechanism [12]. These methods can empirically improve forecasting performance. In order to find the optimal threshold values, Aziz et al. [13] used the whale optimization algorithm (WOA) and MFO, and showed that the two algorithms outperformed the other alternative optimizers, with MFO obtaining better results than WOA and providing a good balance between exploration and exploitation at small and high threshold values. MFO was also used to solve the optimal reactive power dispatch problems [14]. An automatic mitosis detection approach using the mixed neutrosophic sets and MFO was proposed [15]. Li et al. [16] developed an artificial intelligence based on the diagnostic model, which employs MFO-based support vector machines (SVM) with feature selection. In 2018, Mohanty et al. [17] applied MFO to optimize a dual-mode controller for a multi-area hybrid interconnected power system and the experimental results demonstrated that this method can obtain better performance

* Corresponding author.
E-mail address: chenhuiling.jlu@gamil.com (H. Chen).

https://doi.org/10.1016/j.ins.2019.04.022
0020-0255/© 2019 Elsevier Inc. All rights reserved.

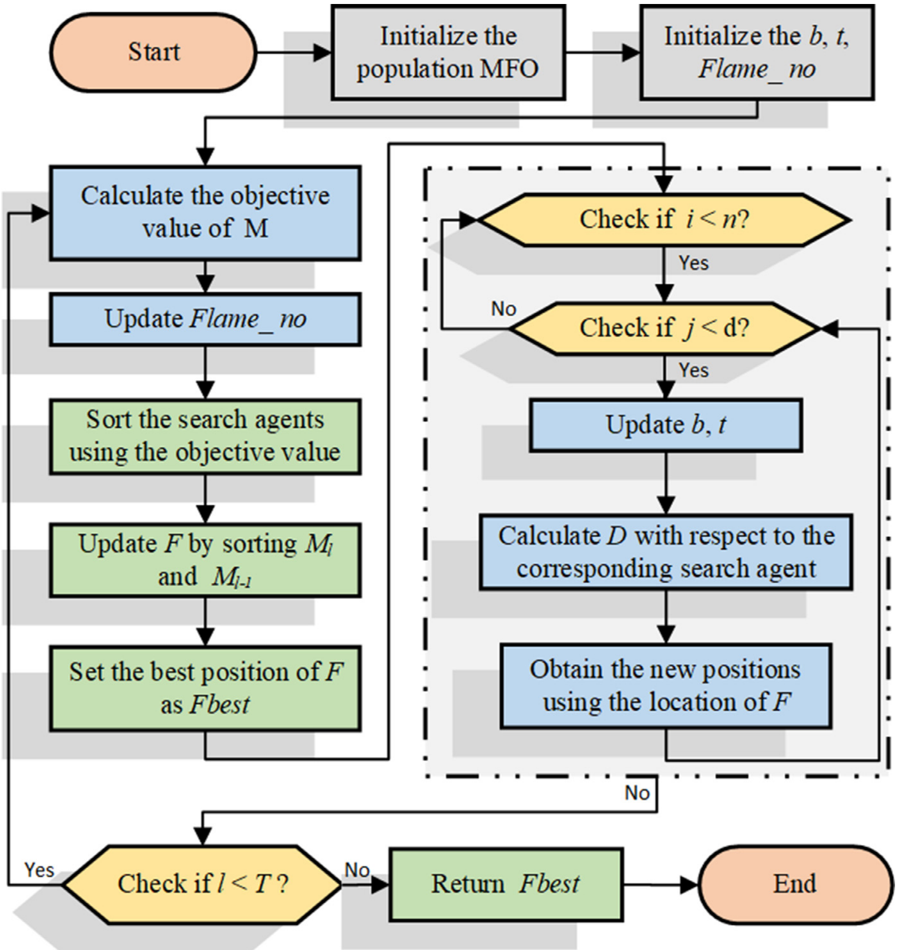

Fig. 1. The flowchart of MFO.

than genetic algorithm (GA), bacterial foraging optimization algorithm (BFO), differential evolution (DE), and hybrid bacterial foraging optimization algorithm particle swarm optimization for the same system. Ibrahim et al. [18] employed MFO as a local search operator to enhance the exploitation ability of brainstorm optimization (BSO). Mohanty [19] applied MFO to solve the automatic generation control problem. However, the original MFO may suffer from premature convergence when solving complex optimization tasks. Thus, a range of approaches have been introduced to cope with this problem. In 2016, Zhang et al. [20] proposed an evolutionary firefly algorithm by mixing MFO's spiral movement and Levy-flight firefly algorithm's attractiveness search actions for feature optimization. In 2017, Hassanien et al. [21] proposed an improved MFO to automatically detect tomato diseases. A hybrid algorithm was also presented to solve constrained engineering optimization problems [22]. In this method, MFO's spiral movement was introduced to the water cycle algorithm (WCA) to enhance its searching ability. To perform parameter optimization and feature selection for the kernel extreme learning machine (KELM), a novel MFO via chaotic strategy was proposed by Wang et al. [23] in 2017. After comparison with the traditional models such as the basic MFO, GA, and PSO, the method was found to have better performance and a smaller feature subset. In 2018, Xu et al. [24] proposed an enhanced MFO by combining the cultural learning and Gaussian mutation. The simulation results showed that the two operators were able to significantly boost the performance of MFO in terms of solution quality and robustness. Sapre et al. [25] integrated the opposition-based learning, Cauchy mutation and evolutionary boundary constraint handling technique into MFO. Although all the above-mentioned MFO variants enhance searching abilities or convergence speeds, they have difficulty in escaping from local optimal when applied to complex tasks with limited iterations. Therefore, the following conclusions can be drawn. Firstly, the limited searching ability of the basic MFO renders it fall into a local minimum easily and may result in premature convergence. Secondly, single mutation algorithms can hardly balance exploration and exploitation abilities. Finally, with limited iterations, the basic MFO has difficulty in obtaining the global optima for complex problems. The conclusions have motivated the following implementations:

1) The mixed mutation operators are introduced to mutate the population in a variety of ways.

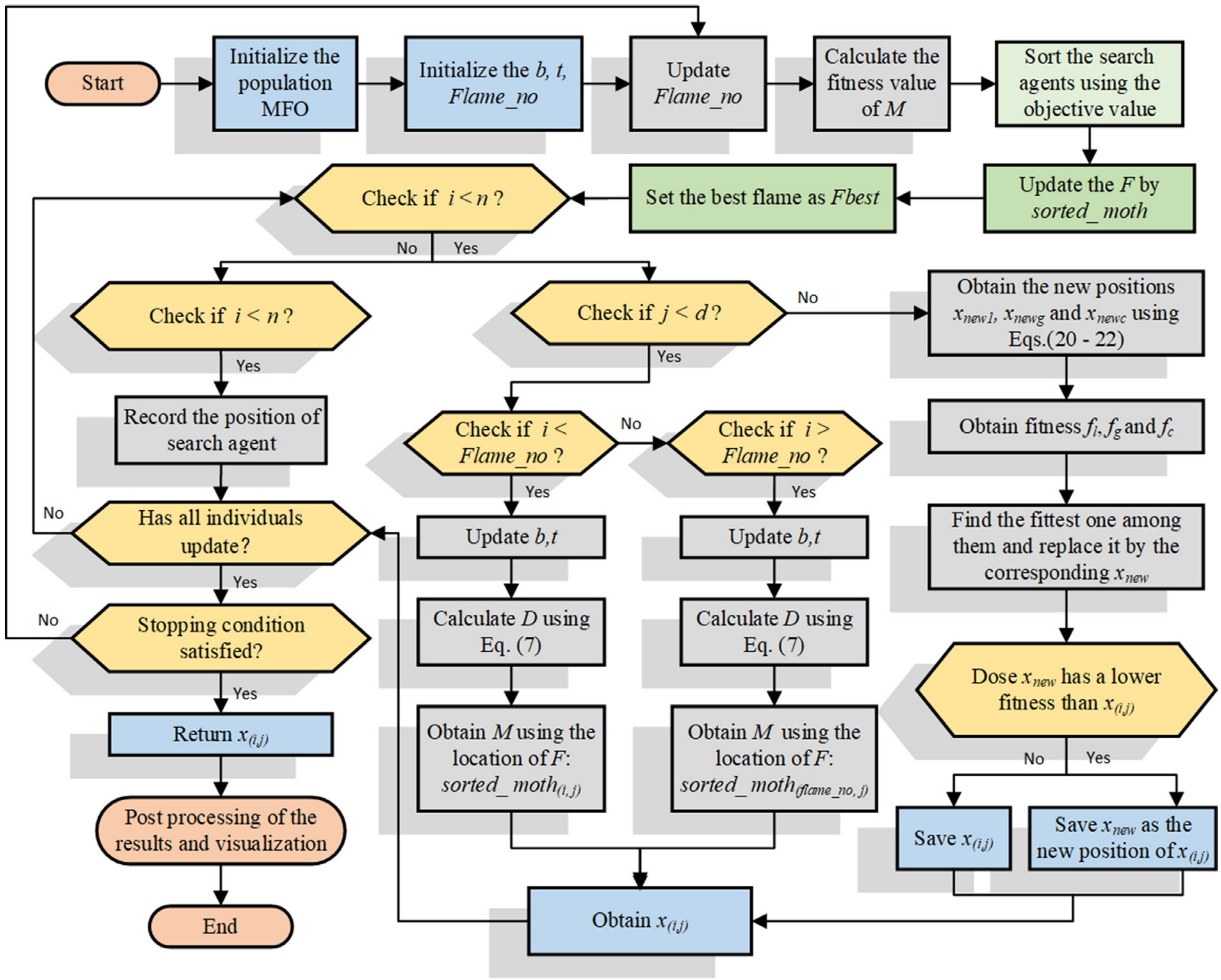

Fig. 2. The framework of LG-MFO.

2) Based on mutation, 7 new operators are mainly adapted from MFO's mutation mechanism. The first three variants exploit Lévy, Gaussian, or Cauchy distribution. The last four use Lévy, Gaussian, and Cauchy distribution combinations in four different ways, which utilize spiral movement's adaptive component.
3) Mutating operators are adopted to increase population diversity, making individuals jump out of local optima easily.

In this paper, in order to further balance the exploration and exploitation processes, we firstly propose a series of mutation-based MFO, namely 'LGCMFO', 'LGMFO', 'LCMFO', 'GCMFO', 'LMFO', 'GMFO', and 'CMFO', which are presented in Section 3. These MFO variants use mutation methods such as Gaussian mutation (GM), Cauchy mutation (CM), Lévy mutation (LM) and the combination of GM, CM and LM operators to improve the performance of the basic MFO. Then, a series of comparative experiments were conducted on CEC2017 and standard benchmark problems, and the best method among all the proposed variants of the basic MFO is analyzed in terms of different dimensions and statistical testing in Section 4. Finally, conclusions and future directions are given in Section 5.

2. Moth-flame optimization (MFO)

MFO is a nature-inspired optimization algorithm proposed by Mirjalili [9] in 2015, which simulates the behavior of individuals in a swarm of moths (search agents), which have special navigation methods in the night. In the MFO algorithm, an assumption is made that the candidate solutions are search agents. In order to model the spiral movement of individuals,

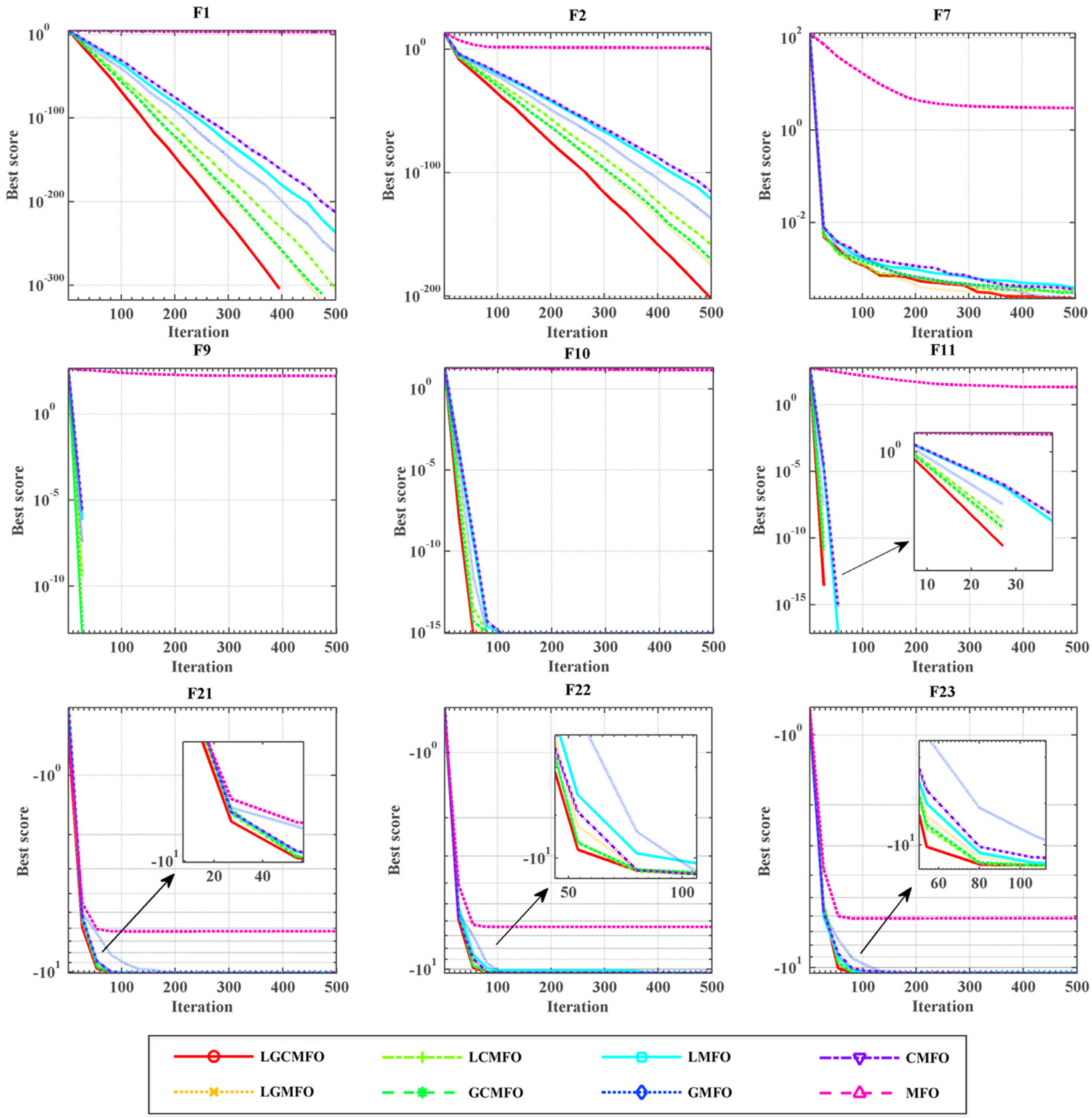

Fig. 3. Convergence curves on 9 benchmark functions.

the following matrix is utilized,

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,d} \\ m_{2,1} & \ddots & \ddots & m_{2,d} \\ \vdots & \ddots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,d} \end{bmatrix} \tag{1}$$

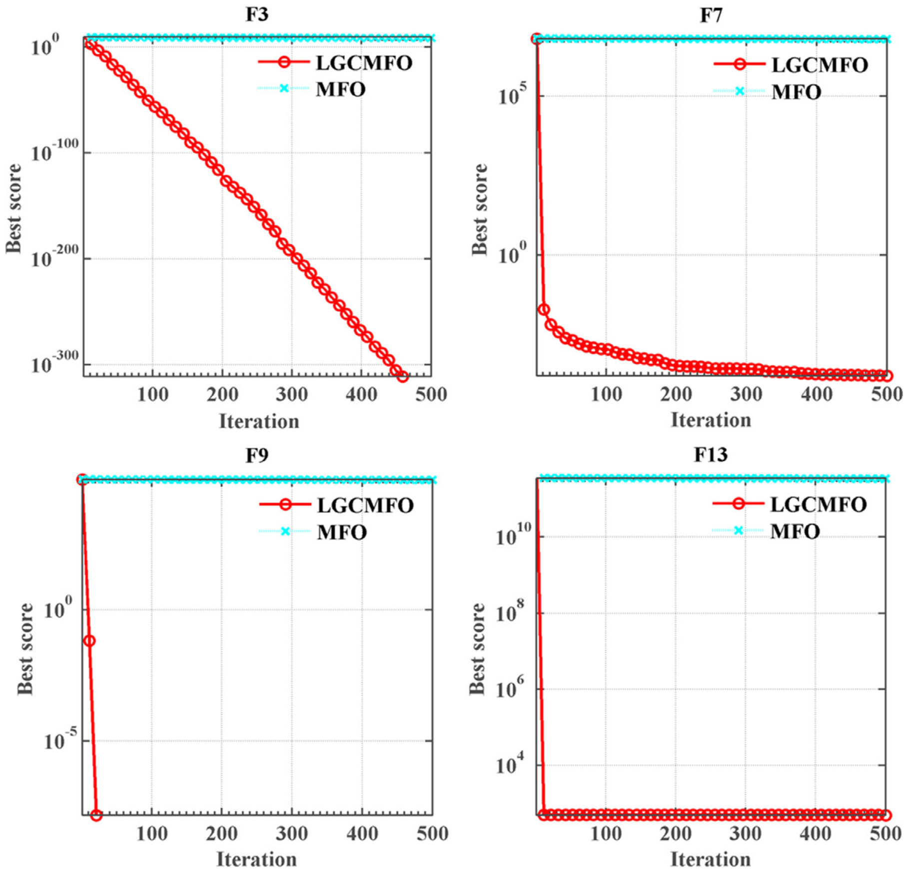

Fig. 4. Convergence curves on 4 benchmark functions in dimension 5000.

Table 1
Unimodal benchmark functions.

Function equation	Dim	Range	f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10,10]	0
$f_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	30	[−100,100]	0
$f_4(x) = max_i\{\lvert x_i \rvert, 1 \le i \le n\}$	30	[−100,100]	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[−30,30]	0
$f_6(x) = \sum_{i=1}^{n} [x_i + 0.5]^2$	30	[−100,100]	0
$f_7(x) = \sum_{i=1}^{n} ix_i^4 + random[0, 1)$	30	[−1.28,1.28]	0

where n is the number of search agents, and d indicates the number of dimensions. For all individuals, it is assumed that there is an array for storing the value of objective function as follows:

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ \vdots \\ OM_n \end{bmatrix}. \quad (2)$$

The other integral part of this algorithm is the flame which is represented in a matrix as in Eq. (3):

$$F = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,d} \\ f_{2,1} & \ddots & \ddots & f_{2,d} \\ \vdots & \ddots & \ddots & \vdots \\ f_{n,1} & f_{n,2} & \cdots & f_{n,d} \end{bmatrix} \quad (3)$$

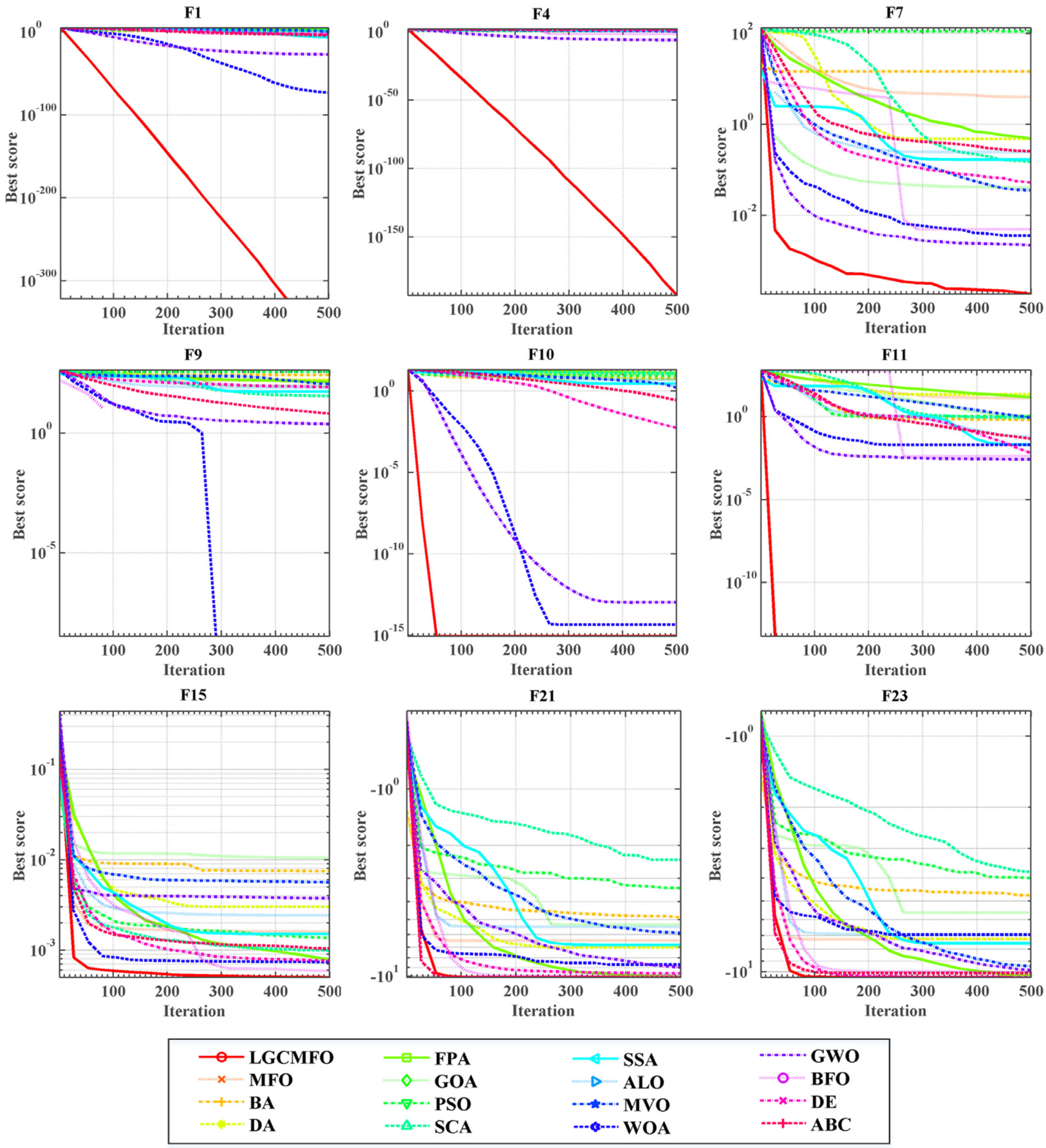

Fig. 5. Convergence curves on 9 benchmark functions.

An assumption is made that there exists an array for storing the fitness value of F as follows:

$$OF = \begin{bmatrix} OF_1 \\ OF_2 \\ \vdots \\ OF_n \end{bmatrix} \tag{4}$$

Table 2
Multimodal benchmark functions.

Function equation	Dim	Range	f_{min}
$f_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	[−500,500]	-418.9829×5
$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[−5.12,5.12]	0
$f_{10}(x) = -20\exp\{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i}\} - exp\{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\} + 20 + e$	30	[−32,32]	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[−600,600]	0
$f_{12}(x) = \frac{\pi}{n}\{10\sin(ay_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2$ $+ \sum_{i=1}^{n}\mu(x_i, 10, 100, 4)\} y_i = 1 + \frac{x_i+1}{4}\mu(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[−50,50]	0
$f_{13}(x) = 0.1\{sin^2(3\pi x_i) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]$ $+ \sum_{i=1}^{n}\mu(x_i, 5, 100, 4)$	30	[−50,50]	0

Table 3
Fixed-dimension multimodal benchmark functions.

Function equation	Dim	Range	f_{min}
$f_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	2	[−65,65]	1
$f_{15}(x) = \sum_{i=1}^{11}[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	[−5,5]	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_i^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5,5]	−1.0316
$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	[−5,5]	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times\ [30 + (2x_1 - 3x_2)^2 \times\ (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[−2,2]	3
$f_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_{j-}p_{ij})^2)$	3	[1,3]	−3.86
$f_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_{j-}p_{ij})^2)$	6	[0,1]	−3.32
$f_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.1532
$f_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.4028
$f_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.5363

In the MFO algorithm, F can be considered as the best position of M when searching the search space. To model this behavior mathematically, the position of each search agent is updated as follows:

$$M_i = S\left(M_i, F_j\right), \tag{5}$$

where M_i indicates the $i-th$ search agent, F_j is the $j-th$ best position obtained so far, and S indicates the logarithmic spiral function which is calculated as follows:

$$S\left(M_i, F_j\right) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j \tag{6}$$

where t is a random number in $[-1, 1]$, b is a constant that defines the shape of the logarithmic spiral, and D_i refers to the distance of the $i-th$ search agent for the $j-th$ flame, which is defined as follows:

$$D_i = \left|F_j - M_i\right| \tag{7}$$

In this algorithm, M is forced to use only one of the F to update its position and an adaptive mechanism for the number of F is proposed as in the following equation:

$$Flame\_no = \text{round}\left(n - l * \frac{n-1}{T}\right) \tag{8}$$

where l refers to the current number of iteration and T indicates the maximum number of iteration. The flowchart of MFO algorithm is shown in Fig. 1.

3. Proposed methodology

MFO is characterized by a fast convergence speed and a simple structure. However, for some complex problems, especially the high dimensional and multimodal functions, it may fall into local optima easily. The optimization performance of the basic MFO depends mainly on the interaction between individuals. Specifically, when a single individual traps into a local optimum, it can escape from it by an F. However, if most of search agents are trapped by the same local minimum, the whole algorithm will slow down and eventually stagnate.

Table 4
Results on 23 benchmark functions.

Function	Metric	LGCMFO	LGMFO	LCMFO	GCMFO	LMFO	GMFO	CMFO	MFO
F1	Avg	0.00E+00	0.00E+00	1.02E−305	0.00E+00	1.54E−237	1.66E−261	1.51E−213	1.03E+03
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.04E+03
F2	Avg	1.29E−202	1.36E−175	4.43E−159	3.11E−171	3.96E−122	9.28E−138	5.60E−116	3.42E+01
	Std.	0.00E+00	0.00E+00	2.21E−158	0.00E+00	1.52E−121	3.74E−137	2.76E−115	2.14E+01
F3	Avg	0.00E+00	0.00E+00	7.90E−292	0.00E+00	4.02E−207	8.01E−243	7.05E−190	2.00E+04
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.43E+04
F4	Avg	4.45E−193	1.01E−170	2.59E−151	1.60E−164	2.71E−112	4.56E−130	2.26E−104	7.17E+01
	Std.	0.00E+00	0.00E+00	1.22E−150	0.00E+00	1.37E−111	1.64E−129	1.20E−103	6.89E+00
F5	Avg	2.50E+01	2.52E+01	2.52E+01	2.52E+01	2.54E+01	2.53E+01	2.54E+01	1.66E+04
	Std.	1.99E−01	2.09E−01	2.20E−01	1.99E−01	1.65E−01	1.97E−01	1.96E−01	3.38E+04
F6	Avg	6.82E−06	6.79E−06	5.56E−06	7.28E−06	1.28E−05	1.21E−05	1.21E−05	1.35E+03
	Std.	6.26E−06	7.40E−06	4.33E−06	7.83E−06	1.51E−05	2.60E−05	9.34E−06	4.35E+03
F7	Avg	2.24E−04	2.29E−04	3.30E−04	3.00E−04	3.78E−04	2.30E−04	3.53E−04	3.02E+00
	Std.	1.50E−04	1.88E−04	4.55E−04	2.34E−04	3.13E−04	1.43E−04	2.73E−04	6.95E+00
F8	Avg	−1.00E+04	−9.96E+03	−1.03E+04	−1.02E+04	−1.00E+04	−9.70E+03	−1.00E+04	−8.70E+03
	Std.	9.67E+02	9.94E+02	8.19E+02	9.16E+02	8.13E+02	9.10E+02	1.10E+03	9.67E+02
F9	Avg	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.62E+02
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.22E+01
F10	Avg	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	8.88E−16	1.47E+01
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.77E+00
F11	Avg	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.20E+01
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.53E+01
F12	Avg	1.88E−07	1.75E−07	3.06E−07	1.99E−07	6.47E−07	2.53E−07	3.04E−07	2.47E+01
	Std.	1.72E−07	1.39E−07	7.63E−07	1.94E−07	1.19E−06	2.71E−07	3.14E−07	8.67E+01
F13	Avg	3.97E−01	4.13E−01	6.99E−01	5.65E−01	2.91E−01	6.15E−01	3.95E−01	1.37E+07
	Std.	4.65E−01	4.86E−01	7.57E−01	7.18E−01	3.62E−01	8.79E−01	5.22E−01	7.49E+07
F14	Avg	2.18E+00	1.06E+00	1.59E+00	1.62E+00	1.13E+00	1.06E+00	9.98E−01	1.99E+00
	Std.	2.49E+00	3.62E−01	1.86E+00	1.86E+00	5.03E−01	3.62E−01	0.00E+00	1.66E+00
F15	Avg	4.57E−04	4.24E−04	5.38E−04	5.60E−04	1.23E−03	5.80E−04	6.73E−04	9.08E−04
	Std.	2.84E−04	1.35E−04	4.18E−04	3.19E−04	3.62E−03	3.24E−04	4.04E−04	3.03E−04
F16	Avg	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00
	Std.	6.71E−16	6.65E−16	6.71E−16	6.65E−16	6.52E−16	6.65E−16	6.65E−16	6.78E−16
F17	Avg	3.98E−01	3.98E−01	3.98E−01	3.98E−01	3.98E−01	3.98E−01	3.98E−01	3.98E−01
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F18	Avg	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	Std.	1.25E−15	1.96E−15	1.41E−15	1.64E−15	8.69E−16	1.11E−15	1.87E−15	1.39E−15
F19	Avg	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00
	Std.	2.61E−15	2.65E−15	2.60E−15	2.65E−15	2.64E−15	2.63E−15	2.64E−15	2.71E−15
F20	Avg	−3.29E+00	−3.28E+00	−3.30E+00	−3.29E+00	−3.26E+00	−3.28E+00	−3.29E+00	−3.21E+00
	Std.	5.54E−02	5.85E−02	4.84E−02	5.35E−02	6.05E−02	5.70E−02	5.35E−02	8.73E−02
F21	Avg	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−6.23E+00
	Std.	1.94E−04	1.90E−05	2.41E−07	4.77E−09	1.39E−07	4.51E−07	6.35E−07	3.39E+00
F22	Avg	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−6.37E+00
	Std.	4.37E−09	2.26E−05	3.04E−10	7.67E−06	9.90E−16	9.80E−15	9.90E−16	3.44E+00
F23	Avg	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01	−6.15E+00
	Std.	6.89E−11	6.15E−08	2.29E−15	1.03E−13	4.33E−12	1.51E−15	1.44E−15	3.70E+00
Overall Rank	Rank	1	3	4	2	7	5	6	8
	+/=/−	~	3/18/2	5/18/0	3/20/0	6/15/2	5/17/1	8/13/2	18/5/0
	ARV	**3.663043**	3.85	3.976087	3.96087	4.638406	4.392029	4.715217	6.804348

In order to search the solution space more effectively for complex optimization problems, here we introduce a method called mutation strategy. In the mutation strategy, there are three innovations embedded into the basic MFO, which are Lévy, Gaussian, and Cauchy mutation operation. Mixed mutation operators are introduced to mutate the population in a variety of ways. Hence, this strategy forces search agents to move toward the best solution and the mixed mutation operators are highly capable of increasing the diversity of the population. In this work, we have introduced several mutation mechanisms into MFO. Different variants are based upon the following basic ideas: Mutation operators based on a new concept are proposed to replace the original ones presented in this section. Based on mutation, 7 new operators are mainly adapted for MFO: (1) The first, the second and the third variants exploit the concept of Lévy, Gaussian and Cauchy distribution, respectively. (2) The fourth, fifth, sixth and seventh variants use a combination of Lévy, Gaussian and Cauchy distributions in four different ways, which use the adaptive component in the spiral movement.

Prior to the local search of the population, the individuals after this mutation are compared with the parent ones. If the individual's fitness during the two evolutionary periods is not improved, the original individual should be retained, and the mutant individual should be discarded to improve the overall quality. This rule is used in all the proposed variants of MFO and will not be specifically mentioned in every subsection.

Table 5
Scalability results on 13 benchmark functions.

F1	Avg		Std		F2	Avg		Std	
	LGCMFO	MFO	LGCMFO	MFO		LGCMFO	MFO	LGCMFO	MFO
10	**0.00E+00**	1.41E−13	**0.00E+00**	4.48E−13		**2.42E−232**	4.31E−09	**0.00E+00**	5.77E−09
1000	**0.00E+00**	2.70E+06	**0.00E+00**	4.61E+04		**1.57E−196**	6.55E+04	**0.00E+00**	NAN
5000	**0.00E+00**	1.57E+07	**0.00E+00**	9.81E+04		**5.71E−197**	6.55E+04	**0.00E+00**	NAN
F3	Avg		Std		F4	Avg		Std	
	LGCMFO	MFO	LGCMFO	MFO		LGCMFO	MFO	LGCMFO	MFO
10	**0.00E+00**	1.67E+02	**0.00E+00**	9.13E+02		**8.03E−220**	2.85E−03	**0.00E+00**	4.50E−03
1000	**0.00E+00**	1.69E+07	**0.00E+00**	2.94E+06		**1.66E−190**	9.94E+01	**0.00E+00**	1.57E−01
5000	**0.00E+00**	4.30E+08	**0.00E+00**	8.76E+07		**4.36E−186**	9.99E+01	**0.00E+00**	1.98E−02
F5	Avg		Std		F6	Avg		Std	
	LGCMFO	MFO	LGCMFO	MFO		LGCMFO	MFO	LGCMFO	MFO
10	**4.75E+00**	2.07E+02	**1.64E−01**	5.65E+02		**5.71E−17**	2.92E−14	**2.52E−16**	3.72E−14
1000	**9.97E+02**	1.23E+10	**2.61E−01**	3.00E+08		**2.08E+02**	2.70E+06	**2.07E+00**	4.75E+04
5000	**5.00E+03**	7.50E+10	**1.56E−01**	6.84E+08		**1.21E+03**	1.57E+07	**3.19E+00**	1.15E+05
F7	Avg		Std		F8	Avg		Std	
	LGCMFO	MFO	LGCMFO	MFO		LGCMFO	MFO	LGCMFO	MFO
10	**8.61E−05**	5.35E−03	**7.23E−05**	2.57E−03		**−1.69E+20**	−3.27E+03	4.91E+20	**3.02E+02**
1000	**1.74E−04**	1.96E+05	**1.29E−04**	4.85E+03		−7.65E+04	**−9.48E+04**	1.62E+04	**5.88E+03**
5000	**1.66E−04**	6.11E+06	**1.09E−04**	6.62E+04		−1.65E+05	**−2.14E+05**	2.97E+04	**1.59E+04**
F9	Avg		Std		F10	Avg		Std	
	LGCMFO	MFO	LGCMFO	MFO		LGCMFO	MFO	LGCMFO	MFO
10	**0.00E+00**	1.81E+01	**0.00E+00**	1.19E+01		**8.88E−16**	8.77E−08	**0.00E+00**	8.19E−08
1000	**0.00E+00**	1.52E+04	**0.00E+00**	2.19E+02		**8.88E−16**	2.03E+01	**0.00E+00**	1.99E−01
5000	**0.00E+00**	8.67E+04	**0.00E+00**	4.68E+02		**8.88E−16**	2.02E+01	**0.00E+00**	6.20E−02
F11	Avg		Std		F12	Avg		Std	
	LGCMFO	MFO	LGCMFO	MFO		LGCMFO	MFO	LGCMFO	MFO
10	**0.00E+00**	1.45E−01	**0.00E+00**	7.69E−02		**1.24E−14**	4.15E−02	**5.51E−14**	1.35E−01
1000	**0.00E+00**	2.43E+04	**0.00E+00**	3.95E+02		**8.09E−01**	3.00E+10	**2.14E−02**	9.33E+08
5000	**0.00E+00**	1.42E+05	**0.00E+00**	8.39E+02		**1.10E+00**	1.87E+11	**5.49E−03**	2.09E+09
F13	Avg		Std			Overall rank		Rank	
	LGCMFO	MFO	LGCMFO	MFO		LGCMFO	MFO	LGCMFO	MFO
10	**7.69E−03**	2.56E−03	2.65E−02	**4.73E−03**		**13**	0	**1**	2
1000	**9.97E+01**	5.52E+10	**7.32E−02**	1.96E+09		**12**	1	**1**	2
5000	**5.00E+02**	7.50E+10	**1.56E−01**	6.84E+08		**12**	1	**1**	2

3.1. Gaussian-MFO (GMFO)

Like other evolutionary algorithms, the basic MFO tends to fall into a local optimal solution . Thefore, we should not follow the process by which every individual of MFO moves to another position inside the search space, but instead we leave a certain ambiguity in the transition to the next generation by means of the mutation strategy. By using the mutation strategy, the proposed enhanced MFO is integrated with a randomized mutation. In this study, there are n search agents mutated in every iteration. It can be described as follows,

$$x_i' = x_i \times (1 + \delta) \tag{9}$$

$$\mathrm{i} \in \{1, \cdots, n\} \tag{10}$$

where x_i and x_i' denote the current and mutated positions of a search agent, respectively, and δ is a mutation operator. The proposed GMFO technique utilizes a mutation operator, called Gaussian mutation(GM) [26]. We integrate GM into the basic MFO with the aim of coping with the loss of diversity in the search procedure. The density function of Gaussian is shown as follows,

$$f_{gaussian(0,\sigma^2)}(\alpha) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\alpha^2}{2\sigma^2}} \tag{11}$$

Table 6
Parameters setting for algorithms used.

Method	Population size	Maximum generation	Other parameters
LGCMFO	30	500, 1000	$b = 1$; $t = [-1\ 1]$; $a \in [-1\ -2]$
MFO	30	500, 1000	$b = 1$; $t = [-1\ 1]$; $a \in [-1\ -2]$
BA	30	500, 1000	$A = 0.5$; $r = 0.5$;
DA	30	500, 1000	$w \in [0.9\ 0.2]$; $s = 0.1$; $a = 0.1$; $c = 0.7$; $f = 1$; $e = 1$
PSO	30	500, 1000	$c_1 = 2$; $c_2 = 2$; $vMax = 6$
SCA	30	500, 1000	$A = 2$
WOA	30	500, 1000	$a_1 = [2\ 0]$; $a_2 = [-2\ -1]$; $b = 1$
BFO	30	500, 1000	$\Delta \in [-1, 1]$
FPA	30	500	$CR = 0.8$
GOA	30	500	$cMax = 1$; $cMin = 0.00004$
SSA	30	500	$c_1 \in [0\ 1]$; $c_2 \in [0\ 1]$
ALO	30	500	$k = 500$
MVO	30	500	*Existence probability*$\in [0.2\ 1]$; *Travelling distance rate*$\in [0.6\ 1]$
GWO	30	500	$a = [2,0]$
DE	30	500	$betaMin = 0.2$; $betaMax = 0.8$; $pCR = 0.2$
ABC	30	500	$Limit = 300$
BLPSO	30	1000	$w = [0.2\ 0.9]$; $c = 1.496$; $I = 1$; $E = 1$
CLPSO	30	1000	$w = [0.2\ 0.9]$; $c = 1.496$
HGWO	30	1000	$betaMin = 0.2$; $betaMax = 0.8$; $pCR = 0.2$
PSOBFO	30	1000	$c_1 = 1.2$; $c_2 = 0.5$

where σ^2 is the variance for each search agent. This equation is reduced to generating a random number according to Gaussian distribution with a value of mean to 0 and standard deviation to 1. The generated random number is applied to the spiral movement phase of the basic MFO algorithm and is shown as follows:

$$x_i' = x_i \times \ (1 + G(\alpha)) \tag{12}$$

where $G(\alpha)$ is a uniformly distributed random number drawn from Gaussian distribution. Thus, the implementation of GMFO can be expressed in detail as follows.

Pseudo code of GMFO

```
Set the parameters of MFO such as the population size, the maximum number of iterations, the boundary of search and the dimensionality of the space.
Create the initial population M
Calculate the objective function value of M: OM
l=1;
while l < T
    Update Flame_no using Eq. (8)
    if l == 1
        F=sort(M); OF=sort(OM)
    else
        F=sort(M_{l-1}, M_l);OF=sort(M_{l-1}, M_l)
    end if
    for i=1: n
        for j=1: d
            Update b and t
            Calculate D using Eq. (7) with respect to the corresponding search agent
            Perform the spiral movement by Eq. (6)
        end for
        Create the new position x_new using Eq. (12)
        if x_new better than x_(i,j)
            x_(i,j)= x_new
        end if
    end for
Update the positions of search agents and the Fbest.
end while
```

3.2. Cauchy-MFO (CMFO)

In this section, we use the CM operator instead of the GM operator proposed above. The experimental results showed that the CM operator also has a strong search capability [27]. The Cauchy distribution function can be described as follows:

$$y = \frac{1}{2} + \frac{1}{\pi} arctan\left(\frac{\gamma}{g}\right) \tag{13}$$

Table 7
Results on 23 benchmark functions.

	F1		F2		F3	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**0.00E+00**	**0.00E+00**	**1.32E−200**	**0.00E+00**	**0.00E+00**	**0.00E+00**
MFO	1.34E+03	3.47E+03	3.74E+01	2.48E+01	2.14E+04	1.18E+04
BA	1.68E+01	2.71E+00	2.04E+03	8.49E+03	1.35E+02	4.07E+01
DA	2.02E+03	9.72E+02	1.74E+01	6.74E+00	1.30E+04	8.31E+03
FPA	1.57E+03	3.48E+02	4.35E+01	9.56E+00	1.82E+03	5.11E+02
GOA	5.09E+01	4.12E+01	1.64E+01	1.92E+01	3.21E+03	1.42E+03
PSO	1.39E+02	1.31E+01	1.04E+02	8.66E+01	6.00E+02	1.24E+02
SCA	1.54E+01	2.35E+01	3.00E−02	4.49E−02	8.40E+03	6.52E+03
SSA	2.18E−07	3.08E−07	1.72E+00	1.49E+00	1.52E+03	9.72E+02
ALO	1.98E−03	3.73E−03	5.00E+01	4.80E+01	4.05E+03	1.46E+03
MVO	1.22E+00	3.64E−01	5.67E+00	1.87E+01	2.32E+02	1.08E+02
WOA	8.21E−74	3.84E−73	2.45E−50	1.06E−49	4.08E+04	1.34E+04
GWO	7.08E−28	1.42E−27	8.61E−17	5.02E−17	1.07E−05	1.73E−05
BFO	8.77E−03	3.06E−03	3.65E−01	6.30E−02	2.48E−12	5.32E−12
DE	5.04E−04	1.80E−04	2.11E−03	3.85E−04	3.31E+04	5.14E+03
ABC	5.91E−05	7.65E−05	5.77E−03	2.87E−03	1.87E+04	2.76E+03
	F4		F5		F6	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**1.90E−193**	**0.00E+00**	**2.50E+01**	**2.33E−01**	**5.40E−06**	**2.89E−06**
MFO	6.96E+01	7.18E+00	2.68E+06	1.46E+07	6.71E+02	2.51E+03
BA	3.41E+00	1.35E+00	5.07E+03	1.47E+03	1.73E+01	1.99E+00
DA	3.15E+01	6.80E+00	5.61E+05	6.40E+05	1.96E+03	1.19E+03
FPA	2.62E+01	3.89E+00	2.26E+05	1.16E+05	1.62E+03	4.27E+02
GOA	1.52E+01	4.15E+00	9.20E+03	1.50E+04	3.85E+01	3.29E+01
PSO	5.03E+00	3.53E−01	1.71E+05	3.76E+04	1.43E+02	1.37E+01
SCA	3.57E+01	1.08E+01	2.34E+04	4.84E+04	2.25E+01	3.30E+01
SSA	1.11E+01	3.81E+00	1.52E+02	1.18E+02	**1.50E−07**	1.63E−07
ALO	1.64E+01	5.36E+00	2.36E+02	2.79E+02	1.23E−03	1.01E−03
MVO	2.09E+00	9.60E−01	4.80E+02	8.07E+02	1.30E+00	3.56E−01
WOA	5.21E+01	2.86E+01	2.78E+01	3.95E−01	4.40E−01	2.32E−01
GWO	1.00E−06	1.42E−06	2.71E+01	6.91E−01	7.30E−01	3.89E−01
BFO	3.22E−02	4.35E−03	6.55E+04	NAN	3.60E−01	8.84E−02
DE	1.30E+01	2.05E+00	1.59E+02	5.27E+01	4.52E−04	1.95E−04
ABC	4.29E+01	5.85E+00	6.58E+01	4.73E+01	9.96E−05	1.14E−04
	F7		F8		F9	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**1.88E−04**	**1.53E−04**	**−9.98E+03**	**1.10E+03**	**0.00E+00**	**0.00E+00**
MFO	4.00E+00	6.54E+00	−8.56E+03	6.55E+02	1.57E+02	3.53E+01
BA	1.46E+01	9.45E+00	−7.25E+03	6.45E+02	2.74E+02	3.06E+01
DA	4.81E−01	2.60E−01	−5.47E+03	6.27E+02	1.74E+02	3.80E+01
FPA	4.90E−01	1.93E−01	−7.47E+03	2.19E+02	1.44E+02	1.80E+01
GOA	4.15E−02	1.33E−02	−7.45E+03	7.68E+02	9.73E+01	3.78E+01
PSO	1.11E+02	2.21E+01	−7.07E+03	8.78E+02	3.78E+02	2.33E+01
SCA	1.50E−01	1.82E−01	−3.80E+03	4.15E+02	3.62E+01	2.61E+01
SSA	1.69E−01	6.19E−02	−7.62E+03	7.78E+02	5.56E+01	2.03E+01
ALO	2.48E−01	1.30E−01	−5.89E+03	1.40E+03	8.09E+01	2.21E+01
MVO	3.58E−02	1.54E−02	−7.71E+03	7.82E+02	1.15E+02	3.00E+01
WOA	3.61E−03	4.74E−03	−1.03E+04	1.72E+03	0.00E+00	0.00E+00
GWO	2.22E−03	1.39E−03	−5.92E+03	8.90E+02	2.43E+00	3.34E+00
BFO	4.99E−03	3.14E−03	−2.39E+03	5.75E+02	**−2.88E+02**	7.50E−01
DE	5.29E−02	1.48E−02	−9.87E+03	5.64E+02	8.65E+01	7.77E+00
ABC	2.58E−01	5.92E−02	**−1.15E+04**	2.66E+02	6.59E+00	2.05E+00
	F10		F11		F12	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**8.88E−16**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.88E−07**	**4.85E−07**
MFO	1.33E+01	8.02E+00	1.30E+01	3.11E+01	1.53E+02	7.78E+02
BA	6.04E+00	4.51E+00	6.47E−01	5.79E−02	1.47E+01	5.16E+00

(*continued on next page*)

Table 7 (*continued*)

DA	1.15E+01	2.07E+00	2.17E+01	9.93E+00	5.09E+03	1.17E+04
FPA	1.31E+01	1.12E+00	1.52E+01	3.34E+00	7.10E+01	2.40E+02
GOA	5.30E+00	1.18E+00	1.13E+00	1.03E−01	1.15E+01	1.05E+01
PSO	8.74E+00	3.22E−01	1.03E+00	5.08E−03	5.52E+00	7.66E−01
SCA	1.25E+01	9.52E+00	9.00E−01	2.80E−01	1.76E+05	6.73E+05
SSA	2.83E+00	6.85E−01	2.07E−02	1.32E−02	8.02E+00	2.87E+00
ALO	4.93E+00	2.89E+00	6.57E−02	2.41E−02	1.19E+01	4.02E+00
MVO	1.63E+00	5.03E−01	8.43E−01	8.98E−02	2.07E+00	1.77E+00
WOA	4.56E−15	2.38E−15	2.02E−02	6.98E−02	3.97E−02	5.82E−02
GWO	1.04E−13	1.89E−14	2.71E−03	6.34E−03	6.38E−02	1.06E−01
BFO	**−8.86E+12**	5.53E+11	4.21E−03	2.08E−03	**1.41E−10**	1.96E−10
DE	5.51E−03	1.26E−03	6.33E−03	8.05E−03	6.70E−05	5.12E−05
ABC	2.71E−01	2.81E−01	4.66E−02	3.62E−02	2.15E−05	3.92E−05
	F13		F14		F15	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**6.20E−01**	**6.55E−01**	**1.20E+00**	**6.05E−01**	**5.00E−04**	**2.65E−04**
MFO	1.37E+07	7.49E+07	1.95E+00	1.61E+00	1.61E−03	3.56E−03
BA	2.71E+00	3.74E−01	5.34E+00	3.96E+00	7.52E−03	9.25E−03
DA	4.99E+05	7.84E+05	1.39E+00	8.86E−01	3.03E−03	5.21E−03
FPA	5.37E+04	6.47E+04	1.04E+00	1.55E−01	7.80E−04	1.64E−04
GOA	3.47E+01	2.10E+01	9.98E−01	5.30E−16	1.05E−02	9.47E−03
PSO	3.32E+01	2.19E+01	3.50E+00	2.25E+00	1.37E−03	3.29E−04
SCA	5.57E+05	2.15E+06	1.60E+00	9.23E−01	9.97E−04	3.55E−04
SSA	1.52E+01	1.55E+01	1.16E+00	4.58E−01	1.51E−03	3.57E−03
ALO	2.96E+01	1.95E+01	2.71E+00	2.30E+00	2.43E−03	4.95E−03
MVO	1.83E−01	1.31E−01	9.98E−01	4.31E−11	5.66E−03	8.28E−03
WOA	5.89E−01	2.96E−01	2.44E+00	2.96E+00	7.42E−04	4.65E−04
GWO	6.34E−01	2.26E−01	3.85E+00	3.45E+00	3.76E−03	7.56E−03
BFO	9.92E−02	1.80E−10	2.98E+00	2.79E+00	5.91E−04	1.84E−04
DE	**2.97E−04**	1.26E−04	**9.98E−01**	0.00E+00	7.66E−04	2.87E−04
ABC	4.66E−04	2.00E−03	9.98E−01	2.08E−16	1.04E−03	4.05E−04
	F16		F17		F18	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**−1.03E+00**	**6.65E−16**	**3.98E−01**	**0.00E+00**	**3.00E+00**	**1.03E−15**
MFO	−1.03E+00	6.78E−16	3.98E−01	0.00E+00	3.00E+00	1.59E−15
BA	−1.03E+00	1.69E−03	3.98E−01	5.30E−04	3.10E+00	9.00E−02
DA	−1.03E+00	5.41E−07	3.98E−01	6.35E−09	3.00E+00	2.02E−06
FPA	−1.03E+00	3.72E−08	3.98E−01	6.68E−09	3.00E+00	1.20E−06
GOA	−1.03E+00	2.91E−13	3.98E−01	1.99E−13	3.00E+00	2.48E−12
PSO	−1.03E+00	2.60E−03	4.00E−01	1.81E−03	3.21E+00	2.32E−01
SCA	−1.03E+00	1.02E−04	4.00E−01	1.65E−03	3.00E+00	1.13E−04
SSA	−1.03E+00	3.93E−14	3.98E−01	9.21E−15	3.00E+00	3.02E−13
ALO	−1.03E+00	1.10E−13	3.98E−01	2.01E−14	3.00E+00	5.59E−13
MVO	−1.03E+00	4.17E−07	3.98E−01	1.55E−07	3.00E+00	3.33E−06
WOA	−1.03E+00	3.57E−09	3.98E−01	6.58E−06	3.00E+00	1.05E−04
GWO	−1.03E+00	2.75E−08	3.98E−01	6.24E−07	3.00E+00	6.79E−05
BFO	−1.03E+00	4.49E−06	3.98E−01	2.23E−06	3.00E+00	3.34E−04
DE	−1.03E+00	6.71E−16	3.98E−01	0.00E+00	3.00E+00	**1.29E−15**
ABC	−1.03E+00	5.22E−16	3.98E−01	0.00E+00	3.00E+00	9.42E−05
	F19		F20		F21	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**−3.86E+00**	**2.67E−15**	**−3.29E+00**	**5.11E−02**	**−1.02E+01**	**3.39E−08**
MFO	−3.86E+00	1.44E−03	−3.24E+00	6.46E−02	−6.47E+00	3.21E+00
BA	−3.83E+00	2.83E−02	−2.83E+00	1.97E−01	−4.85E+00	2.44E+00
DA	−3.86E+00	4.57E−04	−3.26E+00	9.29E−02	−7.03E+00	2.85E+00
FPA	−3.86E+00	7.32E−07	−3.32E+00	5.44E−03	−1.01E+01	5.73E−02
GOA	−3.73E+00	2.65E−01	−3.28E+00	6.13E−02	−5.30E+00	2.93E+00
PSO	−3.84E+00	1.35E−02	−2.77E+00	2.23E−01	−3.38E+00	9.65E−01
SCA	−3.85E+00	2.26E−03	−2.92E+00	3.37E−01	−2.39E+00	1.96E+00
SSA	−3.86E+00	1.81E−11	−3.22E+00	5.71E−02	−6.80E+00	3.31E+00
ALO	−3.86E+00	4.77E−13	−3.29E+00	5.56E−02	−5.46E+00	2.84E+00
MVO	−3.86E+00	4.05E−06	−3.27E+00	6.13E−02	−5.87E+00	3.24E+00
WOA	−3.86E+00	1.17E−02	−3.22E+00	1.04E−01	−8.68E+00	2.47E+00
GWO	−3.86E+00	2.81E−03	−3.27E+00	6.65E−02	−8.97E+00	2.18E+00
BFO	−3.86E+00	7.62E−04	−3.28E+00	1.89E−02	−9.96E+00	9.18E−01

(*continued on next page*)

Table 7 (*continued*)

DE	−3.86E+00	2.71E−15	−3.32E+00	2.17E−02	−9.68E+00		1.49E+00
ABC	−3.86E+00	1.23E−08	−3.32E+00	**1.40E−10**	−1.02E+01		4.20E−05
	F22		F23		Overall Rank		
	Avg	Std	Avg	Std	Rank	+/=/−	ARV
LGCMFO	**−1.04E+01**	**3.16E−06**	**−1.05E+01**	**7.75E−09**	**1**	**~**	**2.221739**
MFO	−8.08E+00	3.39E+00	−7.27E+00	3.62E+00	12	19/4/0	8.955797
BA	−5.95E+00	2.84E+00	−4.73E+00	2.95E+00	15	23/0/0	12.03768
DA	−8.20E+00	2.76E+00	−7.23E+00	2.77E+00	13	22/1/0	11.20652
FPA	−1.02E+01	4.36E−01	−1.04E+01	1.46E−01	10	21/1/1	11.3087
GOA	−6.36E+00	3.69E+00	−5.60E+00	3.67E+00	11	22/0/1	9.967391
PSO	−3.89E+00	1.59E+00	−4.00E+00	1.26E+00	16	23/0/0	12.99855
SCA	−3.04E+00	1.75E+00	−3.77E+00	1.71E+00	14	23/0/0	11.79275
SSA	−8.54E+00	2.96E+00	−7.56E+00	3.76E+00	6	21/0/2	7.131159
ALO	−6.35E+00	3.30E+00	−6.91E+00	3.56E+00	9	23/0/0	8.798551
MVO	−8.14E+00	3.34E+00	−9.46E+00	2.19E+00	8	21/0/2	8.204348
WOA	−8.44E+00	2.59E+00	−6.94E+00	3.53E+00	7	19/4/0	7.283333
GWO	−1.02E+01	9.70E−01	−9.81E+00	2.24E+00	5	20/3/0	6.527536
BFO	−1.00E+01	1.33E+00	−9.98E+00	1.64E+00	4	18/1/4	7.362319
DE	−1.01E+01	1.40E+00	−1.05E+01	4.11E−04	2	15/7/1	4.926812
ABC	−1.04E+01	2.85E−02	−1.01E+01	1.54E+00	3	15/6/2	5.276812

Table 8
CEC2017 benchmark tests.

Function	Name of the function	Class	Optimum
C01	Shifted and Rotated Bent Cigar Function	Unimodal	100
C02	Shifted and Rotated Sum of Different Power Function	Unimodal	200
C03	Shifted and Rotated Zakharov Function	Unimodal	300
C04	Shifted and Rotated Rosenbrock’s Function	Multimodal	400
C05	Shifted and Rotated Rastrigin’s Function	Multimodal	500
C06	Shifted and Rotated Expanded Scaffer’s F6 Function	Multimodal	600
C07	Shifted and Rotated Lunacek Bi-Rastrigin Function	Multimodal	700
C08	Shifted and Rotated Non-Continuous Rastrigin’s Function	Multimodal	800
C09	Shifted and Rotated Lévy Function	Multimodal	900
C10	Shifted and Rotated Schwefel’s Function	Multimodal	1000
C11	Hybrid Function 1 ($N = 3$)	Hybrid	1100
C12	Hybrid Function 2 ($N=3$)	Hybrid	1200
C13	Hybrid Function 3 ($N=3$)	Hybrid	1300
C14	Hybrid Function 4 ($N=4$)	Hybrid	1400
C15	Hybrid Function 5 ($N=4$)	Hybrid	1500
C16	Hybrid Function 6 ($N=4$)	Hybrid	1600
C17	Hybrid Function 6 ($N=5$)	Hybrid	1700
C18	Hybrid Function 6 ($N=5$)	Hybrid	1800
C19	Hybrid Function 6 ($N=5$)	Hybrid	1900
C20	Hybrid Function 6 ($N=6$)	Hybrid	2000
C21	Composition Function 1 ($N=3$)	Composition	2100
C22	Composition Function 2 ($N=3$)	Composition	2200
C23	Composition Function 3 ($N=4$)	Composition	2300
C24	Composition Function 4 ($N=4$)	Composition	2400
C25	Composition Function 5 ($N=5$)	Composition	2500
C26	Composition Function 6 ($N=5$)	Composition	2600
C27	Composition Function 7 ($N=6$)	Composition	2700
C28	Composition Function 8 ($N=6$)	Composition	2800
C29	Composition Function 9 ($N=3$)	Composition	2900
C30	Composition Function 10 ($N=3$)	Composition	3000

The corresponding density function is given as,

$$f_{Cauchy(0,g)}(\gamma) = \frac{1}{\pi}\frac{g}{g^2+\gamma^2} \tag{14}$$

where $g=1$ is the proportion parameter [28], y is a uniformly distributed number between [0, 1], and $\gamma = \tan(\pi(y-1/2))$. The density function of Cauchy distribution is close to Gaussian distribution. However, there are some differences between them. The main differences are that Cauchy distribution is smaller than Gaussian distribution on the vertical direction, while on the horizontal direction, Cauchy distribution is broader than Gaussian distribution. It is possible to improve the search ability of search agents by the addition of the neighbors in each generation. It can be ensured that individuals can improve themselves in major scope, and then discard local optima readily. Hence, Cauchy distribution is adopted as a mutation operator. The version adopting the CM operator in the spiral movement phase of the basic MFO algorithm based on Eqs. (13) and (14) can be represented as follows:

$$x_i' = x_i \times \ (1 + C(\gamma)) \tag{15}$$

where C is a randomly distributed number drawn from Cauchy distribution. For the spiral movement phase, the introduction of the Cauchy mutation mechanism enables the enhanced MFO to exploit the promising space in a much better way. Hence, the quality of solutions can be enhanced with the Cauchy operator throughout the simulation. The pseudo-code is the same as that of GMFO except for the change in the spiral movement phase where Eq. (12) is changed to Eq. (15).

3.3. Lévy-MFO (LMFO)

In order to improve the search ability of search agents, the Lévy flight [6,29] operator (LM) is introduced into MFO. The LM operator (controlling mutation steps) confers a deeper search ability by helping all the search agents to proceed to better promising positions. In this way, LMFO can handle global search more efficiently. A simple version of Lévy distribution can be mathematically defined by:

$$Levy(\beta) \sim u = t^{-1-\beta}, 0 < \beta \le 2 \tag{16}$$

where β is an index of stability. The Lévy random number can be described by the following formula:

$$Levy(\beta) \sim \frac{\varphi \times \mu}{|\nu|^{1/\beta}} \tag{17}$$

where μ and ν are both standard normal distributions, Γ is known as a standard Gamma function, $\beta = 1.5$, and φ denotes as follows:

$$\varphi = \left[\frac{\Gamma(1+\beta) \ \times \sin(\pi \times \beta/2)}{\Gamma\left(\left(\frac{1+\beta}{2}\right) \ \times \beta \times 2^{\frac{\beta-1}{2}}\right)} \right]^{1/\beta} \tag{18}$$

The version adopting Lévy operator in the spiral movement phase of the original MFO algorithm based on Eqs. (16)–(18) can be represented as followed:

$$x_i' = x_i \times \ (1 + L(\beta)) \tag{19}$$

where $L(\beta)$ is a randomly distributed number drawn from Lévy distribution. The LM operator is likely to generate a different offspring because of its heavy-tailed distribution. Hence, it can help all the individuals to jump out of local optima easily. The pseudo-code is the same as that of GMFO except for the change in the spiral movement phase where Eq. (12) is changed to Eq. (19).

3.4. Combined mutated-MFO

3.4.1. LGMFO, LCMFO, GCMFO

Despite its strong local search capability, GM fails to handle global convergence more efficiently. Hence, the application of CM allows for larger mutation than Gaussian and improves the global search capability of the population in each generation. In addition, the Lévy flight that uses the LM operator to control mutation steps confers a deeper search ability. The mutation algorithm is known to be a common method used by evolutionary algorithms for generating new individuals. However, a single mutation algorithm is difficult to achieve a balance between exploration and exploitation abilities effectively. Hence, in order to further improve the performance, four major changes are proposed. Here we use the combinations of the GM, CM and LM operators, convolve them into the spiral movement phase of the basic MFO algorithm and then generate a new solution. The proposed method is achieved by deriving the fittest distributions among the candidate pool of seven different distributions. The three equations are formulated as:

$$x_{newl} = x_i \times \ (1 + L(\beta)) \tag{20}$$

$$x_{newg} = x_i \times \ (1 + G(\alpha)) \tag{21}$$

$$x_{newc} = x_i \times \ (1 + C(\gamma)) \tag{22}$$

Here we use the combination of the LM and GM operators, convolve them into the updating phase of the basic MFO algorithm and thus create a new solution. The combined mutation operator formulated by this process is called the LG operator. The LC operator is the combination of the LM and CM operators. In addition, the GC operator is the combination of the GM and CM operators.

For LG-MFO, the two equations are formulated by Eqs. (20) and (21). For LC-MFO, the two equations are formulated by Eqs. (20) and (22). For GC-MFO, the two equations are formulated by Eqs. (21) and (22). All these two equations are used

Table 9
Results on CEC2017 problems.

	C1		C2		C3	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**9.23E+03**	**7.92E+03**	**2.67E+13**	**3.84E+13**	**1.35E+04**	**3.78E+03**
MFO	8.45E+09	7.08E+09	2.33E+37	1.21E+38	1.35E+05	5.16E+04
BA	1.84E+07	2.44E+06	1.29E+07	3.01E+07	2.92E+03	1.58E+03
DA	3.51E+09	1.75E+09	8.50E+37	3.96E+38	1.15E+05	2.83E+04
SCA	1.73E+10	2.66E+09	1.88E+36	6.26E+36	6.27E+04	1.14E+04
WOA	7.47E+08	3.99E+08	5.09E+33	2.40E+34	2.56E+05	8.50E+04
PSO	1.67E+08	2.53E+07	2.39E+15	2.46E+15	9.93E+03	4.57E+03
BFO	7.66E+10	1.91E+09	2.25E+53	7.45E+53	2.05E+05	3.04E+04
BLPSO	2.14E+09	3.93E+08	1.55E+30	5.18E+30	8.26E+04	1.38E+04
CLPSO	1.16E+09	2.43E+08	8.41E+31	1.41E+32	1.22E+05	1.78E+04
HGWO	4.65E+09	1.65E+09	2.76E+33	4.76E+33	7.87E+04	7.65E+03
PSOBFO	7.67E+10	1.63E+09	2.06E+55	7.40E+55	2.04E+05	4.08E+04

	C4		C5		C6	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**5.00E+02**	**2.24E+01**	**6.28E+02**	**2.84E+01**	**6.10E+02**	**8.05E+00**
MFO	1.24E+03	7.45E+02	7.08E+02	4.54E+01	6.31E+02	1.10E+01
BA	5.02E+02	1.98E+01	8.39E+02	7.79E+01	6.78E+02	1.02E+01
DA	1.07E+03	4.31E+02	8.52E+02	6.35E+01	6.79E+02	1.54E+01
SCA	2.23E+03	4.66E+02	8.14E+02	2.24E+01	6.57E+02	6.64E+00
WOA	7.50E+02	1.05E+02	8.39E+02	5.29E+01	6.78E+02	1.18E+01
PSO	5.04E+02	3.03E+01	7.52E+02	3.80E+01	6.58E+02	9.87E+00
BFO	2.94E+04	3.30E+03	9.57E+02	4.59E+01	6.80E+02	7.98E+00
BLPSO	8.32E+02	4.86E+01	7.43E+02	1.59E+01	6.25E+02	2.73E+00
CLPSO	8.68E+02	8.04E+01	7.26E+02	1.50E+01	6.22E+02	3.69E+00
HGWO	7.42E+02	9.58E+01	7.23E+02	2.33E+01	6.22E+02	4.02E+00
PSOBFO	2.95E+04	1.91E+03	9.63E+02	3.52E+01	6.80E+02	8.70E+00

	C7		C8		C9	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**8.65E+02**	**4.21E+01**	**9.21E+02**	**2.65E+01**	**2.69E+03**	**8.54E+02**
MFO	1.06E+03	1.44E+02	1.00E+03	4.85E+01	6.32E+03	1.95E+03
BA	1.72E+03	2.08E+02	1.02E+03	4.12E+01	1.63E+04	4.84E+03
DA	1.12E+03	9.09E+01	1.10E+03	5.76E+01	1.18E+04	3.65E+03
SCA	1.21E+03	4.67E+01	1.08E+03	2.14E+01	7.16E+03	1.53E+03
WOA	1.31E+03	7.94E+01	1.04E+03	5.00E+01	1.06E+04	4.56E+03
PSO	9.52E+02	1.65E+01	9.99E+02	2.51E+01	6.77E+03	1.72E+03
BFO	1.36E+03	6.46E+00	1.17E+03	1.81E+01	8.83E+03	1.24E+03
BLPSO	1.07E+03	1.91E+01	1.04E+03	1.51E+01	2.68E+03	3.89E+02
CLPSO	1.00E+03	2.54E+01	1.03E+03	1.44E+01	5.73E+03	1.21E+03
HGWO	9.60E+02	3.72E+01	9.79E+02	1.66E+01	2.34E+03	5.53E+02
PSOBFO	1.36E+03	5.94E+00	1.18E+03	2.39E+01	9.31E+03	1.21E+03

	C10		**C11**		**C12**	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**4.92E+03**	**6.08E+02**	**1.24E+03**	**6.69E+01**	**1.18E+06**	**1.23E+06**
MFO	5.62E+03	9.26E+02	4.83E+03	3.73E+03	2.97E+08	4.83E+08
BA	6.06E+03	6.65E+02	1.34E+03	6.49E+01	1.44E+07	8.82E+06
DA	7.53E+03	6.50E+02	2.88E+03	1.29E+03	3.27E+08	1.98E+08
SCA	8.64E+03	2.70E+02	3.06E+03	8.32E+02	1.82E+09	5.32E+08
WOA	6.74E+03	9.32E+02	5.09E+03	1.92E+03	2.19E+08	1.47E+08
PSO	6.57E+03	5.96E+02	1.32E+03	4.90E+01	4.07E+07	1.89E+07
BFO	6.26E+03	4.47E+02	2.49E+04	8.83E+03	2.25E+10	3.68E+09
BLPSO	8.56E+03	2.37E+02	1.86E+03	1.63E+02	1.65E+08	4.98E+07
CLPSO	7.07E+03	3.85E+02	2.62E+03	3.98E+02	1.70E+08	6.50E+07
HGWO	6.41E+03	5.95E+02	4.01E+03	1.36E+03	3.64E+08	1.71E+08
PSOBFO	6.30E+03	4.11E+02	2.77E+04	9.71E+03	2.24E+10	4.79E+09

	C13		**C14**		**C15**	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**3.48E+05**	**1.13E+06**	**3.91E+04**	**4.50E+04**	**7.41E+03**	**6.45E+03**
MFO	5.95E+06	1.80E+07	2.04E+05	3.33E+05	5.44E+04	5.24E+04
BA	8.96E+05	2.67E+05	3.22E+04	2.57E+04	1.78E+05	7.89E+04
DA	5.87E+07	8.63E+07	7.72E+05	1.12E+06	1.04E+06	2.50E+06

(*continued on next page*)

Table 9 (*continued*)

SCA	7.94E+08	5.50E+08	4.95E+05	3.48E+05	4.38E+07	3.25E+07
WOA	1.30E+06	1.48E+06	1.83E+06	1.56E+06	7.36E+05	7.91E+05
PSO	8.19E+06	2.31E+06	3.20E+04	3.10E+04	9.72E+05	3.95E+05
BFO	2.15E+10	9.09E+09	1.45E+07	1.73E+07	2.90E+09	1.51E+09
BLPSO	2.74E+07	1.32E+07	1.97E+05	1.08E+05	2.81E+06	2.50E+06
CLPSO	7.33E+07	4.26E+07	2.42E+05	1.53E+05	4.09E+06	3.35E+06
HGWO	1.86E+08	1.48E+08	9.51E+05	5.60E+05	1.71E+07	2.60E+07
PSOBFO	2.26E+10	8.37E+09	1.48E+07	1.67E+07	3.76E+09	9.62E+08
	C16		**C17**		**C18**	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**2.59E+03**	**2.79E+02**	**2.12E+03**	**2.01E+02**	**2.18E+05**	**1.80E+05**
MFO	3.14E+03	3.61E+02	2.50E+03	2.49E+02	5.29E+06	7.33E+06
BA	3.74E+03	4.86E+02	2.86E+03	4.04E+02	4.73E+05	2.86E+05
DA	4.08E+03	5.12E+02	2.80E+03	2.50E+02	1.20E+07	1.52E+07
SCA	3.86E+03	2.60E+02	2.66E+03	1.81E+02	8.20E+06	9.27E+06
WOA	3.96E+03	5.31E+02	2.72E+03	2.96E+02	1.09E+07	1.20E+07
PSO	3.08E+03	2.45E+02	2.40E+03	2.26E+02	6.02E+05	4.07E+05
BFO	6.69E+03	1.74E+03	1.53E+04	1.56E+04	3.00E+08	2.85E+08
BLPSO	3.53E+03	2.51E+02	2.25E+03	1.57E+02	3.99E+06	2.26E+06
CLPSO	3.16E+03	2.07E+02	2.27E+03	1.02E+02	1.84E+06	1.42E+06
HGWO	3.22E+03	2.95E+02	2.28E+03	1.42E+02	3.97E+06	4.85E+06
PSOBFO	6.92E+03	2.00E+03	9.56E+03	8.29E+03	2.01E+08	2.15E+08
	C19		**C20**		**C21**	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**5.20E+03**	**2.98E+03**	**2.36E+03**	**1.79E+02**	**2.41E+03**	**2.93E+01**
MFO	3.17E+06	1.36E+07	2.65E+03	2.00E+02	2.50E+03	4.49E+01
BA	1.55E+06	8.39E+05	3.01E+03	2.10E+02	2.65E+03	6.73E+01
DA	4.51E+07	6.04E+07	2.88E+03	1.72E+02	2.65E+03	6.27E+01
SCA	5.53E+07	3.16E+07	2.84E+03	1.45E+02	2.59E+03	2.64E+01
WOA	1.08E+07	9.61E+06	2.85E+03	1.99E+02	2.60E+03	6.82E+01
PSO	2.98E+06	1.34E+06	2.66E+03	1.61E+02	2.54E+03	3.61E+01
BFO	3.56E+09	1.29E+09	2.90E+03	2.15E+02	2.76E+03	5.00E+01
BLPSO	4.53E+06	3.79E+06	2.59E+03	1.17E+02	2.53E+03	1.42E+01
CLPSO	4.28E+06	2.51E+06	2.57E+03	1.08E+02	2.51E+03	5.37E+01
HGWO	1.12E+07	1.75E+07	2.63E+03	1.07E+02	2.46E+03	2.80E+01
PSOBFO	3.68E+09	1.20E+09	2.89E+03	1.81E+02	2.73E+03	6.63E+01
	C22		**C23**		**C24**	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**2.30E+03**	**1.26E+00**	**2.76E+03**	**3.20E+01**	**2.92E+03**	**2.89E+01**
MFO	6.03E+03	1.64E+03	2.83E+03	4.72E+01	2.98E+03	2.99E+01
BA	7.47E+03	1.33E+03	3.35E+03	1.19E+02	3.39E+03	1.11E+02
DA	7.74E+03	2.18E+03	3.23E+03	1.04E+02	3.45E+03	1.12E+02
SCA	9.01E+03	2.33E+03	3.04E+03	3.96E+01	3.21E+03	3.54E+01
WOA	7.74E+03	1.49E+03	3.10E+03	1.07E+02	3.22E+03	9.87E+01
PSO	4.71E+03	2.76E+03	3.15E+03	1.79E+02	3.26E+03	8.28E+01
BFO	8.20E+03	4.44E+02	3.63E+03	1.88E+02	4.07E+03	2.66E+02
BLPSO	2.64E+03	4.45E+01	2.91E+03	2.40E+01	3.08E+03	1.79E+01
CLPSO	4.92E+03	1.50E+03	2.90E+03	2.29E+01	3.09E+03	2.07E+01
HGWO	3.21E+03	1.03E+03	2.83E+03	3.30E+01	2.98E+03	2.56E+01
PSOBFO	8.22E+03	5.47E+02	3.66E+03	1.90E+02	4.10E+03	2.97E+02
	C25		C26		C27	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**2.89E+03**	**1.50E+01**	**3.64E+03**	**1.28E+03**	**3.30E+03**	**3.10E+01**
MFO	3.31E+03	3.70E+02	5.65E+03	5.27E+02	3.24E+03	2.45E+01
BA	2.91E+03	2.59E+01	8.45E+03	2.28E+03	3.38E+03	9.27E+01
DA	3.25E+03	1.59E+02	8.51E+03	1.53E+03	3.53E+03	1.47E+02
SCA	3.40E+03	1.29E+02	7.57E+03	3.60E+02	3.51E+03	7.70E+01
WOA	3.06E+03	4.41E+01	8.13E+03	1.13E+03	3.45E+03	1.12E+02
PSO	2.93E+03	2.27E+01	5.41E+03	2.29E+03	3.21E+03	8.81E+01
BFO	7.90E+03	4.03E+01	1.38E+04	1.23E+03	4.91E+03	3.80E+02
BLPSO	3.07E+03	3.08E+01	6.17E+03	4.36E+02	3.36E+03	2.03E+01
CLPSO	3.09E+03	3.88E+01	6.01E+03	5.72E+02	3.33E+03	1.67E+01
HGWO	3.04E+03	4.96E+01	5.35E+03	3.94E+02	3.28E+03	2.70E+01
PSOBFO	7.89E+03	4.16E+01	1.37E+04	1.44E+03	5.05E+03	4.66E+02

(*continued on next page*)

Table 9 (*continued*)

	C28		C29		C30	
	Avg	Std	Avg	Std	Avg	Std
LGCMFO	**3.23E+03**	**2.15E+01**	**3.87E+03**	**2.66E+02**	**3.39E+04**	**5.16E+04**
MFO	4.58E+03	9.86E+02	4.12E+03	3.03E+02	5.17E+05	8.65E+05
BA	3.23E+03	2.29E+01	4.90E+03	3.70E+02	3.73E+06	2.02E+06
DA	3.86E+03	6.45E+02	5.33E+03	4.49E+02	5.44E+07	7.48E+07
SCA	4.15E+03	2.42E+02	5.03E+03	1.83E+02	1.44E+08	5.36E+07
WOA	3.52E+03	9.55E+01	5.13E+03	5.21E+02	3.33E+07	1.96E+07
PSO	3.27E+03	1.93E+01	4.38E+03	2.05E+02	5.77E+06	2.39E+06
BFO	9.32E+03	3.88E+02	1.88E+04	2.00E+04	2.95E+09	1.72E+09
BLPSO	3.48E+03	3.71E+01	4.42E+03	1.48E+02	8.55E+06	3.38E+06
CLPSO	3.73E+03	1.09E+02	4.26E+03	1.83E+02	8.40E+06	4.73E+06
HGWO	3.53E+03	4.60E+01	4.35E+03	1.86E+02	6.08E+07	3.64E+07
PSOBFO	9.41E+03	1.73E+02	1.75E+04	1.38E+04	3.51E+09	2.08E+09
	Overall rank					
	Rank	+/=/−	ARV			
LGCMFO	**1**	~	**1.561111**			
MFO	4	29/0/1	4.817778			
BA	7	25/3/2	5.614444			
DA	10	30/0/0	8.195556			
SCA	9	30/0/0	8.437778			
WOA	8	30/0/0	7.357778			
PSO	2	26/2/2	4.313333			
BFO	11	30/0/0	10.84333			
BLPSO	6	29/1/0	5.478889			
CLPSO	4	30/0/0	5.46			
HGWO	3	28/1/1	5.054444			
PSOBFO	12	30/0/0	10.86556			

simultaneously to create new solutions, among which the fittest one is selected. The framework of LG-MFO is described in Fig. 2. It should be noted that the framework of LG-MFO is the same as those of LC-MFO and GC-MFO except for the change in the spiral movement where Eqs. (20) and (21) is replaced with Eqs. (20) and (22) for LCMFO and Eqs. (21) and (22) for GC-MFO.

3.4.2. LGCMFO

According to Section 3.4.1, three mutation strategies (LM, GM, and CM) are combined with the basic MFO. Use of the Lévy operator allows for larger mutation than Gaussian and Cauchy distributions and hence improves the global search capability of the population in each generation. With a wider tail, Lévy distribution is able to create a larger step, enabling it to avoid premature convergence. In summary, we adopt a combined mutation approach to search more space. For LGC-MFO, three equations are formulated by Eqs. (20)–(22). All three equations are used simultaneously to create three new solutions, among which the fittest one is selected. The framework is the same as that of LG-MFO except for a change in the spiral movement phase where Eqs. (19) and (20) is replaced with Eqs. (19)–(21).

3.4.3. Computational complexity

The computational complexity of the proposed MFO variants is determined mainly by the following seven procedures: initialization, fitness evaluation, sorting mechanism, population updating, the LM, GM and CM strategies. Since the time cost of fitness evaluation is dependent on specific optimization problems, the following analysis focuses on the other six procedures. Considering that the quicksort algorithm is utilized, the computational complexity of sorting procedure in each iteration is $O(T \times n^2)$ in the worst case. The computational complexity of initialization procedure is $O(n \times d)$. Updating the positions of all search agents is $O(T \times n \times d)$. LM, GM and CM are the strategies designed to update the positions of individuals, hence the computational complexity can be formulated as $O(4T \times (n \times d + 4))$. Therefore, the computational complexity of the basic MFO is $O(n \times d) + O(T \times (n^2 + n \times d))$ and the final complexity of LGCMFO is $O(n \times d) + O(T \times (n^2 + 5n \times d + 16))$. The complexity of all the proposed seven variants and the basic MFO is $O(n^2)$.

4. Experimental results

The experiments of the proposed MFO variants were conducted in three phases. Firstly, the performance of all the proposed variants was compared with that of the basic MFO on 23 benchmark functions. Then, the scalability test was performed to test the performance of the enhanced MFO variants and the basic MFO more comprehensively. Thirdly, the best proposed MFO variant was compared to other meta-heuristic algorithms on a comprehensive set of benchmarks including 23 benchmark functions and 30 CEC2017 benchmark problems. The numerical results of these methods in terms of the Average results (Avg.) and standard deviation (Std.) of the optimal function value were selected to evaluate the potential, and

Table 10
Wall-Clock Time Costs on 23 benchmark problems.

F	LGCMFO	MFO	BA	DA	FPA	GOA	PSO	SCA	SSA	ALO	MVO	WOA	GWO	BFO	DE	ABC
F1	**105.1**	8.4	11.6	501.9	23.3	3473.9	4.5	7.4	6.7	849.8	67.4	27.8	9.3	62.8	34.4	12.4
F2	**108.6**	8.0	11.0	707.0	24.2	3363.8	4.6	7.2	7.1	842.2	60.9	29.1	9.6	58.4	33.9	13.4
F3	**276.2**	34.9	39.0	531.5	50.5	3367.7	31.6	35.2	34.8	853.0	88.7	55.6	35.9	70.7	62.1	38.9
F4	**106.9**	9.0	12.0	495.0	23.8	3299.0	5.5	8.0	8.1	822.2	66.3	27.5	9.5	55.9	31.6	11.5
F5	**112.1**	9.5	14.3	494.0	26.5	3315.1	8.0	10.0	9.2	807.8	66.0	32.2	12.4	32.4	36.1	14.8
F6	**113.8**	9.4	14.1	517.3	26.0	3327.6	6.7	9.7	9.3	823.4	67.5	31.8	12.0	74.9	34.6	13.8
F7	**134.1**	14.1	18.6	511.4	28.7	3305.7	10.0	13.5	12.5	819.1	68.2	32.7	16.0	55.5	37.0	15.3
F8	**121.7**	10.7	15.7	526.0	28.1	3338.0	7.7	10.3	9.9	837.9	54.4	30.2	11.7	84.5	33.4	17.5
F9	**120.0**	10.2	15.6	488.4	28.1	3265.1	6.7	9.5	9.3	841.6	72.6	31.6	12.7	53.8	34.1	17.1
F10	**118.4**	11.3	15.7	533.4	28.8	3301.0	8.1	11.1	11.0	840.2	69.9	30.1	13.2	59.1	35.1	16.0
F11	**126.0**	13.0	16.4	510.5	30.5	3364.9	8.3	11.1	12.4	847.0	73.5	33.9	14.5	74.7	39.7	19.1
F12	**203.9**	23.9	30.0	531.8	44.6	3490.4	20.6	24.4	24.3	884.7	81.7	47.4	26.3	213.9	54.2	35.4
F13	**190.9**	21.5	27.4	499.7	40.8	3283.8	20.3	22.5	23.2	839.7	75.1	42.5	23.9	185.9	48.9	28.0
F14	**300.9**	41.2	50.0	231.3	64.5	283.9	40.8	41.8	44.2	120.5	47.2	47.4	45.7	172.1	70.0	57.2
F15	**87.0**	7.3	14.0	221.0	25.0	444.5	4.9	7.3	8.7	142.5	17.7	11.3	9.6	56.1	34.2	17.6
F16	**71.9**	5.7	11.8	184.3	20.0	216.9	3.3	5.2	6.5	73.7	10.6	6.8	5.9	40.3	28.2	12.3
F17	**74.7**	5.4	11.9	197.8	22.6	249.4	2.9	5.7	7.0	84.1	12.4	7.5	6.5	45.8	31.9	15.4
F18	**84.6**	6.3	13.8	217.8	22.4	250.6	3.1	5.3	7.4	89.4	12.7	8.2	7.0	47.4	31.1	16.0
F19	**120.2**	11.9	20.6	228.5	30.0	490.2	8.8	10.2	12.3	122.8	17.7	13.2	12.7	62.2	39.7	21.2
F20	**113.1**	11.7	19.1	238.7	28.5	659.9	7.8	10.6	12.6	197.1	21.5	15.7	11.3	55.4	36.6	19.5
F21	**117.1**	12.0	19.0	216.9	30.0	444.6	9.0	10.9	12.9	139.9	19.9	16.3	12.9	59.6	35.4	20.2
F22	**120.1**	13.4	19.2	212.6	30.9	440.2	10.6	12.2	13.7	139.0	20.4	15.6	14.8	67.6	39.6	24.7
F23	**137.9**	14.1	21.4	225.5	33.4	439.4	13.9	15.0	16.3	140.6	22.3	19.4	18.8	80.0	42.0	24.6

the best result of each problem is marked in bold. Furthermore, the non-parametric statistical test Wilcoxon sign rank test [30] at 0.05 significance level was employed to determine whether the improvement achieved by the proposed method is statistically significant. It should be noted that symbols of "$+/=/-$" indicate that the best MFO variant is superior to, equal to, and inferior to other competitors, respectively. Besides, the Friedman test [31] was carried out to assess the average performance of all the competitors for further statistical comparison, and the average ranking value (ARV) is offered in these comparative results.

4.1. The influence of LM, GM, CM

In this section, 23 classical benchmark functions are listed in Tables 1–3, which are used to evaluate the performance of the proposed MFO variants. These benchmark functions (unimodal, multimodal, and fixed-dimension multimodal) are widely adopted in numerical optimization methods. In this experiment, the dimension of Unimodal and Multimodal functions, the population size and the maximum number of iterations were set to 30, 30 and 500, respectively.

As can be seen from Table 4, all the proposed MFO variants significantly outperform the basic MFO on approximately 95% of the benchmark functions, with slight differences between the proposed variants. Notably, LGCMFO has the smallest Avg. on 10 out of 23 benchmarks. Moreover, for the most benchmark functions, LGCMFO is the best method for obtaining the optimal function value within a small range among all the proposed methods. Therefore, LGCMFO was selected for further research. To show the superiority of LGCMFO intuitively, convergence curves of LGCMFO, MFO, LMFO, GMFO, CMFO, LGMFO, LCMFO and GCMFO on some most typical benchmarks are provided (see Fig. 3).

As can be seen from Fig. 3, for F1 and F2, LGCMFO takes the first place, followed successively by LGMFO, GCMFO, LCMFO, GMFO, LMFO, CMFO and MFO. It should be noted that the ranking is based on the Avg. index over 30 independent runs. Obviously, the faster the convergence speeds, the better the effect of the mutation operator. This rule is used in all experiments in this paper and is not discussed in this section. For F7, all the MFO variants have similar convergent tendency(see Fig. 3), with LGCMFO being the best (see Table 4). This trend can also be seen in dealing with F10, with LGCMFO converging extremely fast during generation 20–60.Therefore, it can be inferred that the Lévy, Gaussian, Cauchy operators have a positive influence on the performance of LGCMFO. For F11 all the proposed MFO variants provide exact global optimum solutions, with LGMFO achieving the fastest convergence speed. For F21, F22 and F23, although LMFO, GMFO, CMFO, LGMFO, LCMFO, GCMFO and LGCMFO converge eventually to similar values, LGCMFO converges extremely fast throughout generation 0–100. Taken together, these results demonstrate that LGCMFO is the best method among all the proposed variants.

In addition, the Wilcoxon sign rank test was used to determine whether LGCMFO is significantly better than the other MFO variants. Statistically significant differences between the comparative results of LGCMFO and other variants are indicated by the symbols of "$+/=/-$". According to the results of the Wilcoxon's sign rank test, the best MFO variant LGCMFO is found to be significantly better than the basic MFO on 18 out of the 23 functions, and equal to the basic MFO on 5 functions. Similarly, LGCMFO are significantly better than or equal to LCMFO and GCMFO in dealing with all functions. Moreover, of the 23 functions, LGCMFO is found to be superior to LGMFO, LMFO, GMFO and CMFO on 3, 6, 5 and 8 functions, respectively, and equal to them on 18, 15, 17 and 13 functions, respectively, and inferior to them on 2, 2, 1, and 2 functions,

Table 11
Wall-Clock Time Costs on 30 CEC2017 benchmark problems.

F	LGCMFO	MFO	BA	DA	SCA	WOA	PSO	BFO	BLPSO	CLPSO	HGWO	PSOBFO
C01	**258.3**	23.7	33.2	1037.5	23.2	72.5	20.9	118.9	127.6	37.3	83.2	129.6
C02	**258.9**	22.3	32.1	1097.9	24.4	75.5	22.8	53.5	125.1	31.6	82.9	64.6
C03	**248.3**	22.4	33.8	1051.1	23.6	69.1	18.8	103.4	121.5	27.9	79.0	112.4
C04	**249.1**	21.7	31.8	1049.1	23.3	69.7	19.8	106.1	114.1	29.5	76.4	126.6
C05	**256.6**	22.2	30.9	1058.8	24.2	69.9	21.4	122.0	118.6	33.1	82.5	131.9
C06	**306.1**	30.1	37.4	1023.4	31.4	74.6	26.8	139.9	124.9	37.1	91.7	155.4
C07	**267.7**	23.3	32.1	1031.3	26.1	71.6	23.0	114.4	126.6	39.8	81.3	125.7
C08	**265.5**	24.5	32.0	1038.0	24.7	72.2	20.7	110.4	119.8	32.8	80.2	132.8
C09	**254.7**	24.8	32.3	1025.7	25.9	72.7	21.3	94.1	116.8	30.8	81.4	106.3
C10	**260.2**	26.2	33.8	1016.5	26.4	72.4	23.2	128.2	121.9	29.0	82.9	134.2
C11	**251.5**	23.2	31.6	1017.6	22.9	69.6	20.9	114.4	121.1	30.6	79.8	133.8
C12	**260.1**	24.7	30.4	1010.4	25.2	68.7	21.5	124.7	118.1	29.1	81.6	132.0
C13	**255.4**	23.5	31.2	1035.7	24.7	67.6	20.4	119.8	117.1	30.9	76.3	131.2
C14	**271.0**	24.4	34.5	1025.3	25.1	74.7	24.1	133.0	122.3	30.9	86.9	153.9
C15	**257.7**	24.7	31.5	1060.8	23.8	71.8	20.1	116.3	116.5	31.8	79.3	135.4
C16	**248.9**	22.5	30.7	1028.9	24.0	70.9	21.5	120.7	115.4	27.6	82.6	133.4
C17	**295.0**	28.1	39.4	1027.0	28.4	73.3	26.0	140.7	120.3	33.2	92.6	156.6
C18	**261.2**	25.8	32.4	1026.5	25.9	71.6	21.8	119.0	121.2	30.9	86.3	137.7
C19	**467.1**	57.5	63.2	1072.5	57.9	108.6	56.3	266.0	130.4	48.8	150.1	264.6
C20	**297.2**	29.8	39.0	1038.6	31.5	79.5	25.5	141.3	123.5	33.4	95.9	156.5
C21	**322.2**	34.3	45.2	1051.0	34.1	80.3	30.4	153.9	116.8	37.3	105.1	177.2
C22	**316.1**	34.0	47.4	1056.7	35.7	81.8	32.0	165.3	117.8	33.7	101.5	179.8
C23	**352.4**	37.9	50.1	1067.9	40.3	91.0	36.8	184.5	118.3	35.0	116.4	202.7
C24	**369.4**	40.7	51.0	1078.6	40.8	87.0	37.9	197.7	121.3	35.5	113.7	214.1
C25	**357.0**	38.8	48.6	1075.8	39.3	82.8	38.4	190.5	128.1	45.6	113.5	198.4
C26	**376.2**	43.3	51.1	1030.5	42.9	88.9	38.7	201.7	120.0	39.8	114.4	221.0
C27	**442.3**	54.8	61.7	1234.9	53.0	101.6	51.1	237.0	122.8	39.0	139.0	264.3
C28	**361.1**	41.7	52.2	1041.8	45.0	90.6	38.4	193.3	123.0	43.2	118.7	216.4
C29	**338.5**	37.0	45.7	1053.7	41.3	86.6	35.2	180.0	121.2	33.8	110.0	198.6
C30	**543.8**	71.7	79.0	1107.9	66.6	114.0	66.3	305.9	140.8	54.8	175.6	318.7

respectively. Using the Friedman test, the average performance of LGCMFO and the other MFO variants is also ranked. As can be seen from Table 4, LGCMFO obtains the lowest average ranking for the 23 benchmark tasks in terms of ARV index, followed successively by LGMFO, GCMFO, LCMFO, GMFO, LMFO, CMFO, and the basic MFO. Based on the Friedman test, it can be concluded that LGCMFO is still the best scheme for handling the three different kinds of benchmark functions.

To sum up, according to almost all metrics, the performance of all the MFO variants is improved compared to the basic MFO, with LGCMFO being the best method among all the proposed variants.

4.2. The scalability test for LGCMFO

In order to compare the performance of LGCMFO and the basic MFO more comprehensively, the scalability test was performed. This experiment focuses on the effect of dimensions on the quality of experimental results. In other words, the performance of LGCMFO and MFO are evaluated in the context of increasing dimensions. Therefore, the influence of dimensions on solution quality can be measured thoroughly. For this purpose, a total of three dimensions in F1–F13 functions were tested: 10, 1000 and 5000. In addition, in this experiment, the number of search agents and the maximum number of iterations were set to 30 and 500, respectively. The Avg. index and Std. index of the optimal value are recorded in Table 5.

Both the Unimodal benchmark problems and the Multimodal functions are more challenging to be addressed as the dimension of search space increases. For F8, MFO is better than LGCMFO when the dimension is 1000 or 5000. However, LGCMFO outperforms MFO in other ranges of dimension on many problems. Furthermore, LGCMFO has a faster convergence toward better results than MFO in higher dimensions (see Fig. 4). The basic MFO obtains the worst solutions in high dimensions because it is easily trapped in the local optimum, whereas LGCMFO is able to find the global optimal solutions within a few iterations not only in low dimensions but also in high dimensions. This may be because the combination of LM, GM and CM increases population diversity, enabling LGCMFO to jump out of the local optimum and obtain a better solution more likely. It was shown that the proposed three strategies can significantly improve the search ability of search agents in different ranges of dimension on the majority of the tasks. Taken together, it can be concluded that LGCMFO outperforms the basic MFO in different ranges of dimension.

4.3. Comparison of LGCMFO with other methods

From the above results, it can be seen that LGCMFO is better than the other MFO variants . In this section, 23 classical benchmark functions and CEC2017 [32] were used to evaluate the performance of LGCMFO. Then, LGCMFO were compared with a wide range of meta-heuristic algorithms. In all the next experiments, the parameter settings for these algorithms are listed in Table 6. For fair comparison, all algorithms were implemented in the same testing environment. The meta-heuristic algorithms used are as follows: MFO, BA [33], DA [34], Flower pollination algorithm (FPA) [35], Grasshopper optimization algorithm (GOA) [36], PSO [37], Sine cosine algorithm (SCA) [38], Salp swarm algorithm (SSA) [39], ALO [40], Multi-verse optimization algorithm (MVO) [41], WOA [42], Grey wolf optimization (GWO) [43], BFO [44], DE [45], ABC [46], Biogeography-based learning particle swarm optimization (BLPSO) (X. [47]), Comprehensive learning particle swarm optimizer (CLPSO) [48], Hybridizing grey wolf optimization with differential evolution (HGWO) [49], PSOBFO [50].

4.3.1. Comparative results on the 23 benchmark functions

To verify the optimization performance of LGCMFO, it was compared with a comprehensive set of classical optimization methods including BA, DA, FPA, GOA, PSO, SCA, SSA, ALO, MVO, WOA, GWO, BFO, DE, ABC, and the basic MFO. All tests were implemented based on the recommendations for the 23 benchmark functions, and the simulation results are presented in Table 7, which shows the Avg. index and Std. index of the best solution found by each algorithm over 30 independent runs. And in this experiment, the dimension of unimodal and multimodal functions, the population size and the maximum number of iterations were set to 30, 30 and 500, respectively. As can be seen from Table 7, LGCMFO obtains the best results on 14 out of the 23 test functions and it ranks first in three different kinds of benchmark functions. Therefore, LGCMFO outperforms the other algorithms on most of the test problems.

For F1, F2, F3, and F4, LGCMFO obtains the global optimum while all the other algorithms are stuck at some non-optimal solutions. For F5 and F7, all the basic algorithms are competitive, but LGCMFO provides the best results, indicating that the exploitation capability of LGCMFO on unimodal functions has been improved considerably. Furthermore, the function values of LGCMFO on F1–F7 are all smaller than those obtained by the other methods. The well-known F1–F7 problems, which are known to have no local optima and only one global optimum, are suitable for benchmarking the exploitation capability of algorithms. GM is advantageous for exploiting new space near the newly explored solutions. Consistent with this, it can be noted that LGCMFO has improved exploitation capability in dealing with unimodal cases, compared to the basic MFO. For F9, only LGCMFO and WOA obtain the global optimum. For F10 and F11, LGCMFO yields the best results. The multi-modal functions are useful for evaluating exploration capability since they have a large number of local optima. The results obtained by LGCMFO on F8–F13 show that LGCMFO has an outstanding exploration capability. This is because LGCMFO is combined with the GM, CM and LM operators, which can enhance both exploration and exploitation capabilities effectively. According to the convergence curves shown in Fig 5, it can be seen that LGCMFO is the best algorithm in terms of convergence speed and it outperforms all the other methods in optimizing the F15, F21, and F23 problems. According to all metrics, the performance of LGCMFO is better than the basic MFO. For the F18 problem, it can be seen that LGCMFO yields very competitive solutions compared to DE. With regard to the Std. index, DE explores more precise solutions. However, as can be seen from the convergence curves, LGCMFO has a faster convergence speed and superior performance on the F1, F4, F7, F9, F10, F11, F15, F21, and F23 problems.

As are indicated by the symbols of "+/=/−", of the 23 functions, LGCMFO is significantly better than the basic MFO, DA, WOA, and GWO on 19, 22, 19 and 20 functions, respectively, and equal to them on 4, 1, 4 and 3 functions, respectively. Similarly, LGCMFO is superior to FPA, GOA, SSA, MVO, BFO, DE and ABC on 21, 22, 21, 21, 18, 15 and 15 functions, respectively, and inferior to them on 1, 1, 2, 2, 4, 1 and 2 functions, respectively, and equal to them on 1, 0, 0, 0, 1, 7 and 6 functions, respectively. Moreover, the results yielded by LGCMFO are significantly better than those obtained by BA, PSO, SCA, and ALO on all functions. The Wilcoxon sign rank test shows that LGCMFO is statistically better the other optimizers on the majority of three different kinds of test functions. Also, LGCMFO provides the best ARV of 2.221739. Accordingly, LGCMFO has the best performance among all these well-known competitors from a statistical point of view.

4.3.2. Comparative results on CEC2017 problems

To further evaluate the performance of LGCMFO, four famous advanced algorithms were also included, namely, BLPSO, CLPSO, HGWO, and PSOBFO (See Table 8). The 30 CEC2017 benchmark tests can be categorized into 4 groups: Unimodal (C01–C03), Multimodal (C04–C10), Hybrid (C11–C20) and Composition (C21–C30) functions. For a fair comparison, all the algorithms were implemented in the same testing environment. And in this experiment, the dimension of CEC2017 benchmark tests, the population size and the maximum number of iteration were set to 30, 30 and 1000, respectively. The results obtained LGCMFO and the other algorithms on the C1–C30 problems are presented in Table 9.

As can be seen from Table 9, LGCMFO is the top three techniques in dealing with the unimodal C2 and C3 problems. For the C1 problem, it is superior to the original MFO and the other popular algorithms. It can be noted that the exploitation ability of the proposed technique on unimodal problems is enhanced considerably. The reason is that LGCMFO is able to put more emphasis on exploitative patterns during the whole iterative process due to the embedded GM-based operator that can enhance exploitation ability.

For the multimodal C9 problem, BLPSO, HGWO, and LGCMFO are comparable, with HGWO being the best algorithm. For the C4–C8 and C10 problems, LGCMFO obtains the best solutions. In addition, it can be noted that, although the results yielded by the other popular algorithms are comparable, that of LGCMFO is much better. This shows that LGCMFO outperforms the other competitors and the basic MFO for most multimodal tasks. The main reason is that LGCMFO can achieve a balance between exploration and exploitation in dealing with multimodal problems. For C14, PSO, BA, and LGCMFO are comparable, with PSO being the best algorithm. For the C11–C13 and C15–C20 problems, LGCMFO obtains the best results, indicating that LGCMFO can establish a better balance between exploration and exploitation on hybrid tests due to the

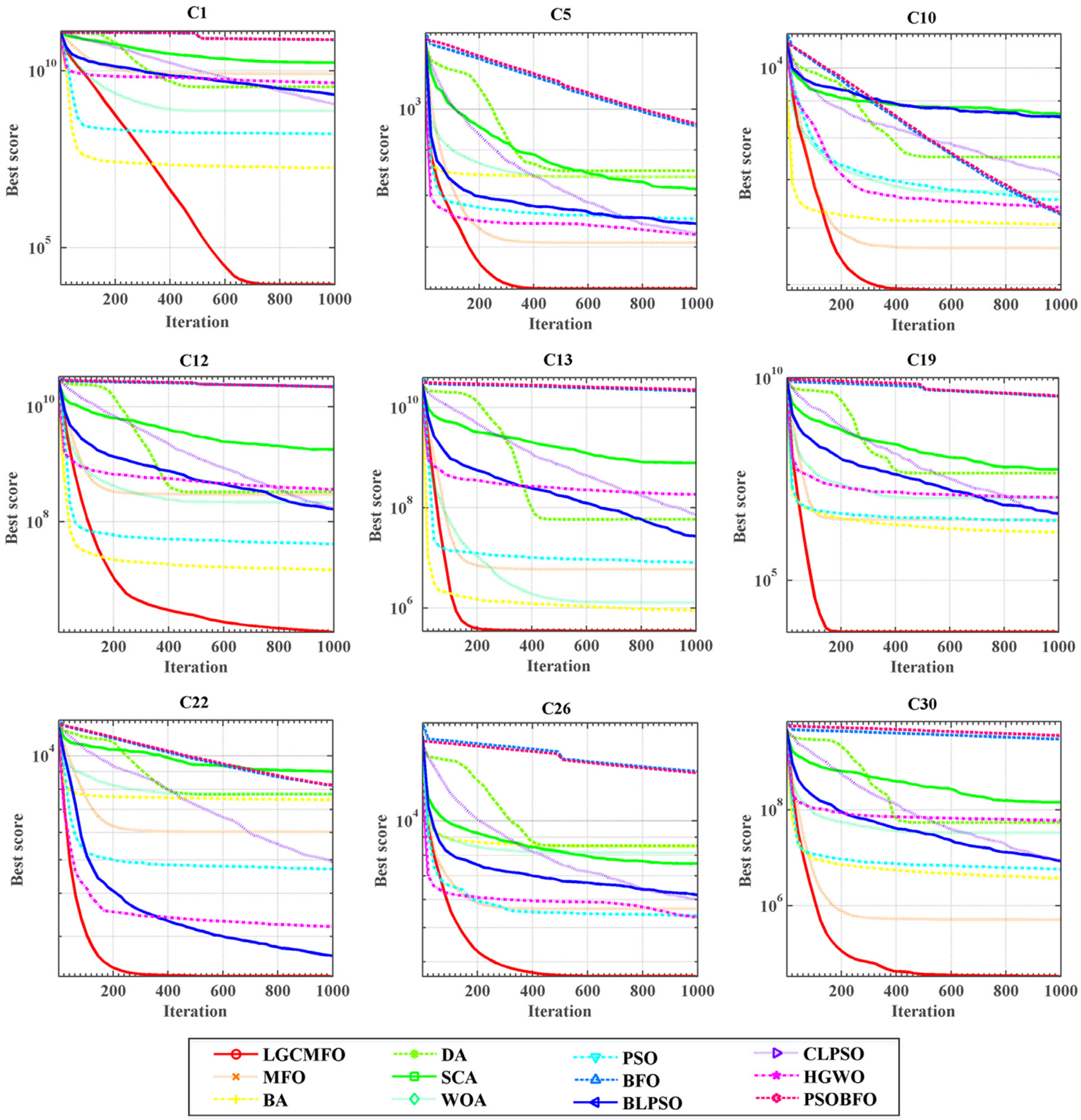

Fig. 6. Convergence curves on some selected functions from CEC2017.

ability of LM to increase the randomness of search agents' movement. Moreover, CM with a large mutation step enhances the global exploration ability of LGCMFO while GM enables LGCMFO to escape from local optima more likely. Notably, for the composition C21–C26, C28–C30 problems, LGCMFO outperforms BA, DA, SCA, WOA, PSO, BFO, BLPSO, CLPSO, HGWO, PSOBFO, and the basic MFO in terms of all metrics. For C25 and C29, LGCMFO, BA and SSA have similar convergence speeds and obtain similar values, with LGCMFO yielding a better solution. In addition, the faster convergence speed of LGCMFO in dealing with composition problems can be noted in Fig. 6. The results demonstrate that the GM, CM, and LM operators introduced into LGCMFO have improved the performance considerably.

As are indicated by the symbols "+/=/−", the optimal function values of LGCMFO are significantly better than those obtained by 7 state-of-the-art algorithms and 4 advanced methods on the majority of four different test functions. The ranking results for the 30 functions are presented in Table 9, with LGCMFO being the best, followed successively by PSO,

HGWO, CLPSO, MFO, BLPSO, BA, WOA, SCA, DA and BFO. Hence, LGCMFO is shown to be the best algorithm for CEC2017 problems.

4.3.3. Wall-clock time cost

The wall-clock time consumed by LGCMFO and other 19 competitors on 23 benchmark problems and 30 CEC2017 benchmark problems is shown in Tables 10 and 11. It should be noted that the wall-clock time is recorded in seconds over 30 independent runs. Based on wall-clock time on all 23 functions and 30 problems, it can be seen that the proposed LGCMFO consumes more time than the basic MFO. The main reason is that three strategies (GM, CM, and LM) are introduced into the basic MFO to achieve a better balance between exploration and exploitation abilities. In addition, it can be noted that the wall-clock time cost of the classical DA, GOA, and ALO is much greater than that of LGCMFO on 23 benchmark functions. Similarly, the time cost of DA over 30 independent runs is also much longer than that of LGCMFO on 30 CEC2017 problems. Although the time cost of LGCMFO is greater than that of MFO and other competitors, it can be seen from the experimental results that LGCMFO is significantly better on most cases. Therefore, it is of great value to introduce the three synchronizing strategies into the basic MFO.

5. Conclusions and future directions

In this paper, the performance of the recently proposed MFO algorithm is enhanced using a new updating strategy for search agents. Specifically, to further balance the exploration and exploitation processes and increase the diversity of the population, the CM, GM and LM operators are introduced into the MFO algorithm. Accordingly, a series of MFO variants are proposed, namely, ‘LGCMFO’, ‘LGMFO’, ‘LCMFO’, ‘GCMFO’, ‘LMFO’, ‘GMFO’, and ‘CMFO’. The combination operator is achieved by deriving the fittest distribution among the candidate pool of seven different distributions. The MFO variants are all better than the basic MFO algorithm, with LGCMFO being the best among all the variants. LGCMFO was also compared with a variety of popular algorithms, including BA, DA, FPA, GOA, PSO, SCA, SSA, ALO, MVO, WOA, GWO, BFO, DE and ABC, on an array of well-known benchmark tests and CEC2017 problems. In addition, the MFO variants were also analyzed in terms of different dimensions and statistical tests in Section 4. To measure convergence speed, we generated the convergence curves. Experimentally, convergence speed, mean and standard deviation were utilized to analyze the results. It can be seen that the mutation operation enables LGCMFO to jump out of local minima and increases convergence speed. Moreover, LGCMFO has sufficient solutions to explore the search space. Taken together, it can be concluded that the exploration and exploitation capabilities of LGCMFO are considerably improved, compared to the basic MFO.

Although LGCMFO has been shown to be superior to the basic MFO, it can be seen that the time complexity of LGCMFO is increased greatly. Therefore, the surrogate model can be applied to LGCMFO to reduce the computational burden in future. In addition, LGCMFO can also be extended to the discrete version to solve discrete optimization tasks. Furthermore, applying the proposed method to more practical problems such as engineering optimization control cases is a worthwhile direction. Combining the proposed LGCMFO with other popular algorithms is also interesting.

Conflict of interest

The authors declare that there is no conflict of interests regarding the publication of article.

Acknowledgments

This study was financially supported by Zhejiang Provincial Natural Science Foundation of China (LR17F030001, LY17F020012), the Science and Technology Plan Project of Wenzhou, China (ZG2017019, G20150017, ZG2017016, C20170008), the Graduate Scientific Research Foundation of Wenzhou University(3162018024).

References

[1] R. Abbassi, A. Abbassi, A.A. Heidari, S. Mirjalili, An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models, *Energy Convers. Manage.* 179 (2019) 362–372.
[2] H. Chen, Y. Xu, M. Wang, X. Zhao, A balanced whale optimization algorithm for constrained engineering design problems, *Appl. Math. Modell.* 71 (2019) 45–59.
[3] W. Deng, J. Xu, H. Zhao, An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem, IEEE Access 7 (2019) 20281–20292.
[4] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: algorithm and applications, *Future Gener. Comput. Syst.* (2019). https://doi.org/10.1016/j.future.2019.02.028.
[5] A.A. Heidari, P. Pahlavani, An efficient modified grey wolf optimizer with Lévy flight for optimization tasks, *Appl. Soft Comput. J.* 60 (2017) 115–134.
[6] J. Luo, H. Chen, Q. zhang, Y. Xu, H. Huang, X. Zhao, An improved grasshopper optimization algorithm with application to financial stress prediction, *Appl. Math. Model.* 64 (2018) 654–668.
[7] X. Zhang, W. Hu, W. Qu, S.J. Maybank, Multiple object tracking via species-based particle swarm optimization, *IEEE Trans. Circuits Syst. Video Technol.* 20 (2010) 1590–1602.
[8] X. Zhang, W. Hu, N. Xie, H. Bao, S. Maybank, A Robust tracking system for low frame rate video, *Int. J. Comput. Vision* 115 (2015) 279–304.
[9] S. Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, *Knowl. -Based Syst.* 89 (2015) 228–249.
[10] D. Allam, D.A. Yousri, M.B. Eteiba, Parameters extraction of the three diode model for the multi-crystalline solar cell/module using Moth-Flame Optimization Algorithm, *Energy Convers. Manage.* 123 (2016) 535–548.

[11] C. Li, S. Li, Y. Liu, A least squares support vector machine model optimized by moth-flame optimization algorithm for annual power load forecasting, *Appl. Intell.* 45 (2016) 1166–1178.
[12] H. Zhao, H. Zhao, S. Guo, Using GM (1,1) optimized by MFO with rolling mechanism to forecast the electricity consumption of inner mongolia, *Appl. Sci. (Switzerland)* 6 (2016) 1–18.
[13] M.A.E. Aziz, A.A. Ewees, A.E. Hassanien, Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation, *Expert Syst. Appl.* 83 (2017) 242–256.
[14] R. Ng Shin Mei, M.H. Sulaiman, Z. Mustaffa, H. Daniyal, Optimal reactive power dispatch solution by loss minimization using moth-flame optimization technique, *Appl. Soft Comput. J.* 59 (2017) 210–222.
[15] G.I. Sayed, A.E. Hassanien, Moth-flame swarm optimization with neutrosophic sets for automatic mitosis detection in breast cancer histology images, *Appl. Intell.* 47 (2017) 397–408.
[16] C. Li, L. Hou, B.Y. Sharma, H. Li, C. Chen, Y. Li, X. Zhao, H. Huang, Z. Cai, H. Chen, Developing a new intelligent system for the diagnosis of tuberculous pleural effusion, *Comput. Methods Programs Biomed.* 153 (2018) 211–225.
[17] B. Mohanty, B.V.S. Acharyulu, P.K. Hota, Moth-flame optimization algorithm optimized dual-mode controller for multiarea hybrid sources AGC system, *Opt. Control Appl. Methods* 39 (2018) 720–734.
[18] R.A. Ibrahim, M.A. Elaziz, A.A. Ewees, I.M. Selim, S. Lu, Galaxy images classification using hybrid brain storm optimization with moth flame optimization, *J. Astron. Telesc., Instrum. Syst.* 4 (2018).
[19] B. Mohanty, Performance analysis of moth flame optimization algorithm for AGC system, *Int. J. Model. Simul.* (2018) 1–15.
[20] F. Wang, H. Lu, B. Gou, X. Han, Q. Zhang, Y. Qin, Modeling of shield-ground interaction using an adaptive relevance vector machine, *Appl. Math. Modell.* 40 (2016) 5171–5182.
[21] A.E. Hassanien, T. Gaber, U. Mokhtar, H. Hefny, An improved moth flame optimization algorithm based on rough sets for tomato diseases detection, *Comput. Electron. Agric.* 136 (2017) 86–96.
[22] S. Khalilpourazari, S. Khalilpourazary, An efficient hybrid algorithm based on Water Cycle and Moth-Flame Optimization algorithms for solving numerical and constrained engineering optimization problems, *Soft Comput.* (2017) 1–24.
[23] M. Wang, H. Chen, B. Yang, X. Zhao, L. Hu, Z. Cai, H. Huang, C. Tong, Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses, Neurocomputing 267 (2017) 69–84.
[24] L. Xu, Y. Li, K. Li, G.H. Beng, Z. Jiang, C. Wang, N. Liu, Enhanced Moth-flame optimization based on cultural learning and Gaussian mutation, *J. Bionic Eng.* 15 (2018) 751–763.
[25] S. Sapre, S. Mini, Opposition-based moth flame optimization with Cauchy mutation and evolutionary boundary constraint handling for global optimization, *Soft Comput.* (2018).
[26] J. Lin, Y. Zhong, Accelerated shuffled frog-leaping algorithm with Gaussian mutation, *Inf. Technol. J.* 12 (2013) 7391–7395.
[27] R. Salgotra, U. Singh, Application of mutation operators to flower pollination algorithm, *Expert Syst. Appl.* 79 (2017) 112–129.
[28] H. Wang, H. Li, Y. Liu, C. Li, S. Zeng, Opposition-based particle swarm algorithm with Cauchy mutation, in: 2007 IEEE Congress on Evolutionary Computation, CEC 2007, 2007, pp. 4750–4756.
[29] X.S. Yang, Firefly algorithm, Lévy flights and global optimization, in: Research and Development in Intelligent Systems XXVI: Incorporating Applications and Innovations in Intelligent Systems XVII, 2010, pp. 209–218.
[30] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, *Inf. Sci.* 180 (2010) 2044–2064.
[31] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, KEEL: a software tool to assess evolutionary algorithms for data mining problems, *Soft Comput.* 13 (2009) 307–318.
[32] A. LaTorre, J.M. Pena, A comparison of three large-scale global optimizers on the CEC 2017 single objective real parameter numerical optimization benchmark, in: 2017 IEEE Congress on Evolutionary Computation, CEC 2017 - Proceedings, 2017, pp. 1063–1070.
[33] X.S. Yang, A new metaheuristic Bat-inspired Algorithm, in: Studies in Computational Intelligence, 284, 2010, pp. 65–74.
[34] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, *Neural Comput. Appl.* 27 (2016) 1053–1073.
[35] X.S. Yang, M. Karamanoglu, X. He, Flower pollination algorithm: a novel approach for multiobjective optimization, *Eng. Optim.* 46 (2014) 1222–1237.
[36] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, *Adv. Eng. Software* 105 (2017) 30–47.
[37] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks - Conference Proceedings, 4, 1995, pp. 1942–1948.
[38] S. Mirjalili, SCA: a Sine Cosine Algorithm for solving optimization problems, *Knowl. -Based Syst.* 96 (2016) 120–133.
[39] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems, *Adv. Eng. Software* 114 (2017) 163–191.
[40] S. Mirjalili, The ant lion optimizer, *Adv. Eng. Software* 83 (2015) 80–98.
[41] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-Verse Optimizer: a nature-inspired algorithm for global optimization, *Neural Comput. Appl.* 27 (2016) 495–513.
[42] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Software* 95 (2016) 51–67.
[43] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software* 69 (2014) 46–61.
[44] N.K. Jhankal, D. Adhyaru, Bacterial foraging optimization algorithm: a derivative free technique, in: 2011 Nirma University International Conference on Engineering: Current Trends in Technology, NUiCONE 2011 - Conference Proceedings, 2011.
[45] R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.* 11 (1997) 341–359.
[46] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *J. Global Optim.* 39 (2007) 459–471.
[47] X. Chen, H. Tianfield, C. Mei, W. Du, G. Liu, Biogeography-based learning particle swarm optimization, *Soft Comput.* 21 (2017) 7519–7541.
[48] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE Trans. Evol. Comput.* 10 (2006) 281–295.
[49] A. Zhu, C. Xu, Z. Li, J. Wu, Z. Liu, Hybridizing grey Wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC, *J. Syst. Eng. Electr.* 26 (2015) 317–328.
[50] W. Korani, Bacterial foraging oriented by particle swarm optimization strategy for PID tuning, in: GECCO'08: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation 2008, 2008, pp. 1823–1826.