Neural Computing and Applications (2020) 32:2315–2349
https://doi.org/10.1007/s00521-018-3821-6

ORIGINAL ARTICLE

An enhanced moth flame optimization

Komalpreet Kaur[1] · Urvinder Singh[1] · Rohit Salgotra[1]

Received: 4 July 2017 / Accepted: 15 October 2018 / Published online: 26 October 2018

Abstract

Moth flame optimization (MFO) is a recent nature-inspired algorithm, motivated from the transverse orientation of moths in nature. The transverse orientation is a special kind of navigation method, which demonstrates the movement of moths toward moon in a straight path. This algorithm has been successfully applied on various optimization problems. But, MFO suffers from the problem of poor exploration. So, in order to enhance the performance of MFO, some modifications are proposed. A Cauchy distribution function is added to enhance the exploration capability, influence of best flame has been added to improve the exploitation and adaptive step size and division of iterations is followed to maintain a balance between the exploration and exploitation. The proposed algorithm has been named as enhanced moth flame optimization (E-MFO) and to validate the applicability of E-MFO, and it has been applied to twenty benchmark functions. Also, comprehensive comparison of E-MFO with other meta-heuristic algorithms like bat algorithm, bat flower pollination, differential evolution, firefly algorithm, genetic algorithm, particle swarm optimization and flower pollination algorithm has been done. Further, the effect of population and dimension size on the performance of MFO and E-MFO has been discussed. The experimental analysis shows the superior performance of E-MFO over other algorithms in terms of convergence rate and solution quality. Also, statistical testing of E-MFO has been done to prove its significance.

Keywords Nature-inspired algorithms · Moth flame optimization · Cauchy function · Benchmark functions

1 Introduction

From the past few decades, classical methods have been successfully established for finding optimum solution for various optimization problems. Due to increase in number of parameters, the complexities of these problems keep on increasing. Therefore, while working with high-dimensional problems, classical methods fail in obtaining exact global minima or maxima and get trapped in local minima. So, it becomes a challenging task for classical methods to find exact solution of problems [1]. Nature-inspired algorithms were defined in the same context to deal with the problems of classical optimization algorithms. It is fact that nature has served as a great source of inspiration for human to solve real-world problems from millions of years. There are so many real-world processes which depict the nature-inspired computing like decision making, immune system, collective behavior, learning. Based on these phenomena, various nature-inspired optimization algorithms were designed. Presently, nature-inspired algorithms have become popular in almost every area of research because of their faster and flexible nature. The main reason for their popularity is that these algorithms are population based and do not require any gradient information for finding the global solution. Another reason for their popularity is that they do not require any initial guess as required by their classical counterparts. Till date, a large number of algorithms have been proposed in this context and are mainly grouped into two categories namely swarm intelligent algorithms (SIA) and evolutionary algorithms (EA).

Swarm intelligence (SI) is a branch of meta-heuristic algorithm which is based on the social behavior of swarm such as group of insects, ants, bees, birds. In this technique, swarm interacts with each other and with surroundings in search of food or prey. According to a study, this method is useful to solve nonlinear and non-differentiable engineering problems [2]. It has two main components namely self-organization and division of labor. Self-organization is the feature by which swarms interact with each other under the

✉ Urvinder Singh
urvinder@thapar.edu

[1] Department of ECE, Thapar University, Patiala, India

control of a central authority or external elements, whereas for division of labor work is divided into smaller tasks which are performed by specialized individuals [3]. One of the popular swarm-based algorithms is particle swarm optimization (PSO) [4] which mimics the flocking behavior of birds. Furthermore, this category includes ant colony optimization (ACO) [5] (based on collective behavior of ants to find shortest distance between their position and food), artificial bee colony (ABC) [6] (inspired from foraging patterns of honeybees), cuckoo search (CS) [7] (based on the breeding patterns of cuckoo species found in nature), bat algorithm (BA) [8] (inspired from the echolocation principle of micro-bats), firefly algorithm (FA) [9] (based on the flashing light patterns of fireflies found in nature), gray wolf optimization (GWO) [10] (based on the hunting mechanism of gray wolves) and whale optimization algorithm (WOA) [11] (based on natural behavior of whales).

Evolutionary algorithms follow the basic rules of evolution. The purpose of these algorithms is to find the fittest individual among all solutions and pass it to the next generation. The reason behind this is that best solutions from the previous generation have more chances to produce more optimized solutions. One of the popular evolutionary algorithms is genetic algorithm (GA) [12]. This algorithm was proposed by Holland in 1992 and is based on Darwin's theory of evolution. Later on, to overcome the problems of GA, differential evolution (DE) [13] algorithm was introduced. Other algorithms in this category include genetic programming (GP) [14], biogeography-based optimization (BBO) [15], ant lion optimization (ALO) [16], evolutionary strategy (ES) [17], moth flame optimization (MFO) [18] and others.

MFO is an evolutionary algorithm proposed in the recent past. This algorithm is inspired from the navigation behavior or transverse orientation of moths found in nature. According to transverse orientation, moths fly toward the moon in straight line by making fixed angle with respect to it and this phenomenon is only valid for those light sources which are far away from moth. So, any artificial lights in their path make them to follow the light and due to which they stuck around it and keep moving in deadly spiral path. This phenomenon was used to formulate the MFO algorithm and since its inception, it has been applied to solve various real-world tasks like image thresholding [19], optimal power flow [20], neural networks [21] and others. The extensive literature about the MFO is given in Sect. 2.

For any generalized algorithm, exploration and exploitation are the most important features. Here, exploration refers to exploring the search space or the global search, whereas exploitation defines the local search. Though MFO is a good algorithm as per the no free lunch theorem [22], no algorithm is best fit for all the optimization problems. So, in order to enhance the performance of MFO, some modifications are proposed which increase its exploitation and exploration capabilities and also maintain a proper balance between the two. Four modifications have been proposed in the basic MFO algorithm. Here, the concept of division of iterations, exponential step size, Cauchy-based global search and best flame has been used for updating the position. The newly proposed algorithm has been named as enhanced MFO (E-MFO), and it focuses on enhancing the searching abilities as well as convergence rate of MFO.

This paper is divided into following sections: Sect. 2 provides the illustration of basic MFO and gives the literature survey, whereas Sect. 3 describes the proposed modifications and Sect. 4 discusses about results and discussion. The final Sect. 5 discusses about the conclusion and future work.

2 Moth flame optimization algorithm

2.1 Inspiration

Moths are winged insects and belong to the family of butterflies. There are almost 160,000 species of moths present in nature. There are two stages in the lifetime of moths, namely larvae and adult. In initial stage, moth larvae live in cocoons and fully grow into adult moths. But some of them live under the earth for evolving process. During night, moths fly with the help of moonlight. For flying purpose, they use special navigation method called transverse orientation. According to this method, a moth maintains the fixed angle with respect to moon while flying as shown in Fig. 1. This method is helpful for traveling long distances and also ensures the straight flight toward moon. However, it has been observed that moths fly around

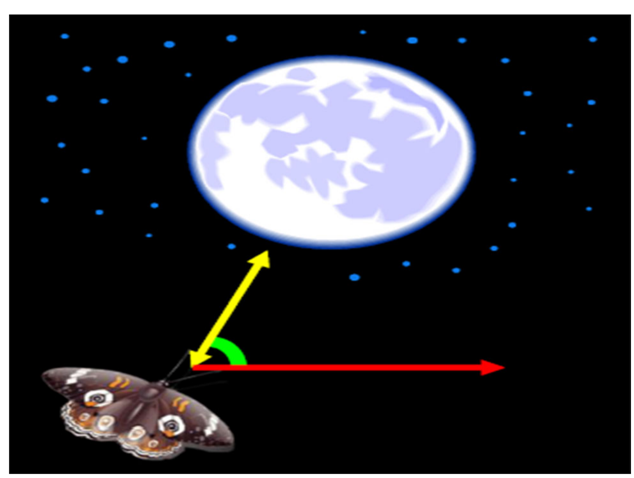


Fig. 1 Transverse orientation [18]

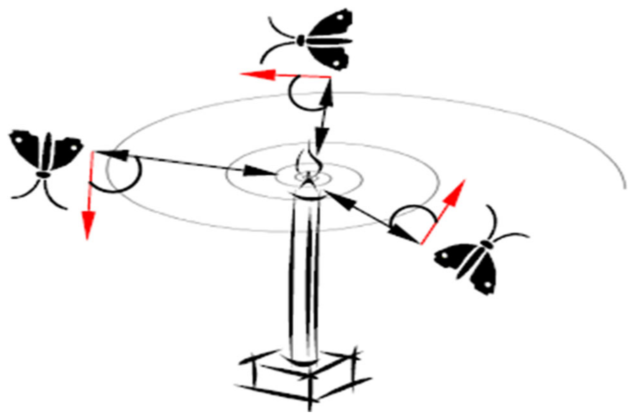


Fig. 2 Spiral flying path [18]

the light in spiral manner instead of transverse orientation. This is because transverse orientation is effective only when light source is very far away from the moth. Moths show same behavior around the artificial lights and try to make similar angle with light so that they can travel in straight line. Since the light source is not as far as moon so this straight line path is changed to spiral path which is depicted in Fig. 2. This behavior has been mathematically modeled, and the model is called moth flame optimization (MFO), which is described in subsequent subsection.

2.2 Mathematical model

The key components of MFO algorithm are given below:

Moth component The MFO algorithm consists of three parts which is used to find global minima for any given problem. It can be defined as follows:

$$\mathrm{MFO} = (I, P, T) \tag{1}$$

The I component defines the initialization of randomly distributed moths in search space, P function is the movement of moths toward the flame and T function returns true when the termination condition is satisfied and false otherwise.

In MFO algorithm, the number of solutions represents the moths (n) and dimension (d) of problem defines the position of moth in search space. As we know, MFO is a population-based algorithm so moths can be represented in form of matrix.

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,d} \\ m_{2,1} & m_{2,2} & m_{2,d} \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ m_{n,1} & m_{n,2} & m_{n,d} \end{bmatrix}$$

The fitness value of each solution is stored in array as shown below:

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ \cdot \\ \cdot \\ \cdot \\ OM_n \end{bmatrix}$$

Here, n represents the number of moths. Basically, moth's position vector is passed to fitness function which returns the output called fitness value.

Flame component To describe the flames, F matrix is considered same as moth matrix and its fitness value OF is also calculated in similar way. The moths and flames are both solutions, and the only difference lies in their updation mechanism. Moreover, we can say that moths are the search agents which explore the search space and the flames are best solution obtained so far. Flames can be considered as flags which are dropped by moths while searching.

Transverse orientation of moths To simulate the behavior of moth in mathematical model, the position of moth is updated around flame using Eq. (2)

$$M_i = S\left(M_i, F_j\right) \tag{2}$$

Here, M_i represents the ith moth, F_j represents the jth flame and S is spiral function which defines how moth updates its position "around" the flame not necessarily in the space between them. The logarithmic spiral is chosen for this paper. The chosen spiral must satisfy the below conditions:

1. The starting point of spiral must be a moth's position.
2. Spiral should end at flame.
3. Fluctuation of range of spiral should be within given search space.

The equation used for updating mechanism of MFO algorithm is given below:

$$S\left(M_i, F_j\right) = D_i.e^{bt}.\mathrm{Cos}(2\pi t) + F_j \tag{3}$$

where D_i indicates the distance between the ith moth and jth flame. It can be calculated as:

$$D_i = \left|F_j - M_i\right| \tag{4}$$

Here, b is any constant number that defines the shape of spiral and t is random number between 1 and − 1. The t parameter decides the step size of moth's movement toward flame. In other words, this parameter describes that how much moth's next position will be close to the flame. It is shown in Fig. 3 that when t = 1 moth is far from the flame, whereas t = − 1 indicates the closest position of moth around the flame.

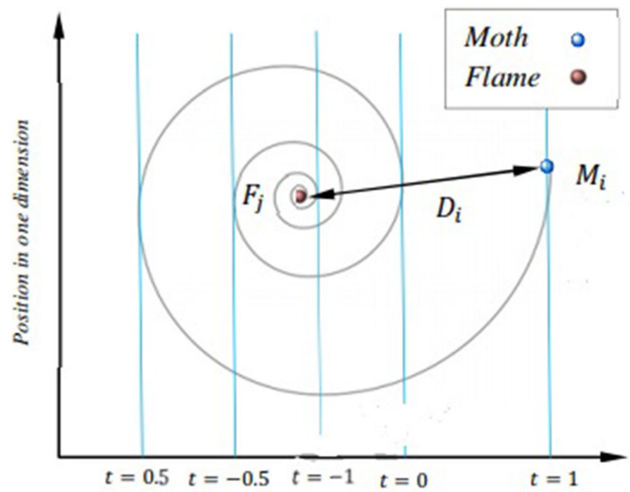


Fig. 3 Logarithmic spiral [18]

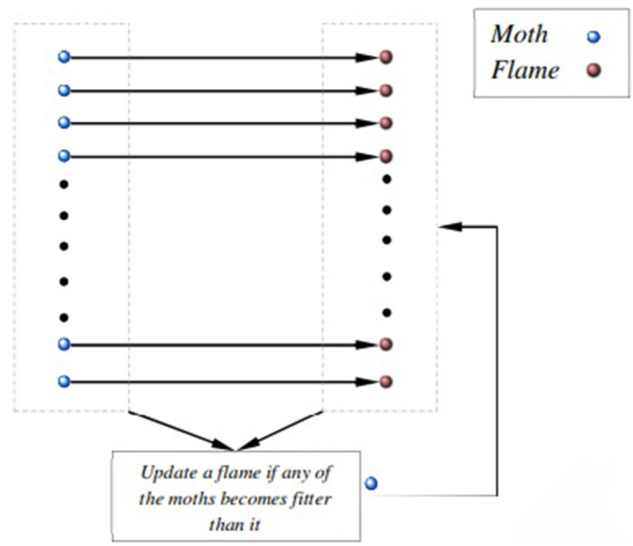


Fig. 4 Connection between moth and flame [18]

2.3 Some points about MFO

1. The t parameter is key index, by changing its value moth can converge to any position around the flame. As the value of t decreases, distance between moth and flame also decreases. Moreover, for proper exploitation, the parameter t is taken in the range of r to 1 where r is linearly decreasing function from -1 to -2 over course of iterations. This change will lead to more precise exploitation of corresponding flames by moths.
2. To avoid local minima, each moth is allowed to update its position toward only one flame. In every iteration, flames are updated and sorted based on the fitness value. Afterward, moths update its position with respect to their corresponding flames. If all moths are attracted toward only one flame, it will lead to stagnation. However, moving toward different flames leads to high exploration. First moth will update its position according to first flame, and last moth will update its position with respect to worst flame. Figure 4 shows the connection between the moths and flames. To ensure the best results, best solutions obtained so far are considered as flames and moths move toward the flame in hyper-sphere path.
3. Best solutions in each iteration get changed and so do the flames. So, moths are required to update their position toward newly updated flame. It will cause the moths to move around different flames. This rapid movement of moths in the search space promotes exploration.
4. The numbers of flames also keep on decreasing with increase in iterations. In initial stages, the moths update its position around assigned flame and start moving toward the best flame in the final stages. This adaptive mechanism makes balance between the exploration and exploitation. The following relation is used to generalize this statement:

$$\text{flameno} = \text{round}\left((N - L) * \frac{N-1}{T}\right) \tag{5}$$

where N is number of flames, L is current iteration and T is maximum iterations.

2.4 The literature survey

MFO is a recently introduced population-based algorithm inspired from moth's special navigation methods. Due to its good performance, it has been used for solving many engineering design problems. This section includes the discussion about its modified versions and number of fields in which it has been applied. Few of the articles related to MFO are discussed below:

Article 1 [23] In this article, authors proposed an improved version of MFO named as Levy flight-based MFO (LMFO). To improve the diversity of algorithm, Levy function is used to update the position at the end of iteration. This will lead to rise in convergence speed due to its higher capability of avoiding local minima. Combination of MFO with Levy flights makes an adequate balance between exploration and exploitation. The proposed algorithm is tested on 19 benchmark functions. According to numerical analysis, LMFO outperforms basic MFO. So, this approach helps MFO in obtaining global minima and help in avoiding local minima. Furthermore, this LMFO algorithm is also applied on two real-world problems namely beam welding design and speed reducer design. Just like benchmark functions, LMFO algorithm performs effectively for these engineering problems also.

Article 2 [24] This paper studies the effect of spirals on performance of MFO algorithm. In original paper, logarithmic spiral is used; however, in this article, two different spirals are used, one is reciprocal spiral and another is Archimedes' spiral. The new version of MFO is applied on feature selection problem to predict the terrorist groups. Data taken from Global Terrorism Database are preprocessed with help of MFO, and then its best feature set is chosen. The results are conducted on two classifier algorithms such as random forests (RF) ensemble and K-nearest neighbor algorithm. The results shown in paper conclude that new version of MFO gives competitive results to original MFO. Also, this algorithm provides more stability, validity and robustness.

Article 3 [21] The goal of this paper is to train the multilayer perceptron (MLP) with optimal values of weights and biases so that minimum error and high classification rate can be obtained. Here MFO is used to find the best possible values of weights which are used with training samples to train MLP. The mean square error is taken as key index to evaluate the performance. This problem is verified on five different datasets like XOR, heart, balloon, iris and breast cancer. Experimental analysis shows that MFO algorithm is very effective in training MLP and avoiding local minima problems. Moreover, it gives better results in comparison with other algorithms due to its high exploring and exploiting capabilities.

Article 4 [25] In this paper, authors describe the various mathematical models for the solar cells and proposed another model named as three diode model. Authors emphasized the need for extracting or estimating the modal parameters by using meta-heuristic algorithms. Here, MFO algorithm is used to achieve minimum root-mean-square error (RMSE) and mean bias error (MBS). It is clear from the results that MFO algorithm efficiently calculates the model parameter while optimizing the RMSE and MBS. In comparison with other algorithms, MFO has high convergence rate which means it reaches the global minima in less number of iterations.

Article 5 [20] Optimal power flow is challenging task in an electrical domain. In this paper, an interconnected power system is taken with various constraints. The objective of this problem is to minimize the fuel cost, rate of emission and improvement in voltage profile. MFO algorithm has been used to test the 59-bus test system which has 20 control variables. The results show the effectiveness and robustness of this algorithm in comparison with various other algorithms.

Article 6 [26] According to this paper, MFO algorithm is hybridized with rough set theory to detect tomato disease. This combination of MFO and rough set is used for feature selection phase. The fitness function of MFO depends on rough sets. The idea behind using rough set is to generate all possible feature reductions and then choose one with minimum cardinality. This proposed algorithm exploited the high exploration capability of MFO and high performance of rough sets for feature selection of real dataset of tomato disease. To verify the results, proposed algorithm is tested on six benchmark datasets and results showed that this algorithm is efficient in comparison with others in terms of accuracy, feature size and execution time.

Article 7 [27] This article deals with elimination and minimization of harmonic distortions in multilevel inverters by using latest developed meta-heuristic algorithm. To verify the effectiveness of MFO algorithm, this is tested on 7 and 11 multilevel inverter. MFO is used to calculate the switching angles at different values of modulation index. The results showed that MFO is successful in solving harmonic problem efficiently.

Article 8 [28] The objective of this paper is to find the shortest path between source and destination even if that shortest path may contain attacker node. In other words, we can say that it deals with open shortest path first problem (OSPF). This problem is optimized using MFO. The optimized OSPF can be used in huge networks for secure routing. Delay and consumption of energy are key parameters to evaluate the performance of optimized OSPF. For simulation, a large network area is considered with nine blocks and results show the superior performance of MFO with respect to others.

Article 9 [19] The new modification is proposed in this paper which helps to improve the exploration ability of MFO. Chaos theory and crossover processes are introduced in this algorithm which increases the randomness or diversity. Chaotic systems have properties like randomness, certainty and ergodicity which help the solution to jump out of local minima. The linear chaotic map is employed and is controlled by random parameter P whose value varies from 0 to 0.5. The updating equation of basic MFO is converted into:

$$M_i = S(M_i, F_i) + c * (2 * X_i - 1) \tag{6}$$

Here, c specifies the constant whose value is equal to 1.75 and X_i is chaotic vector. Furthermore, this algorithm employs crossover process with crossover probability equal to 0.5. Moreover, adaptive flame number mechanism is also changed using sinusoidal calculation; it will help algorithm to make proper balance between exploration and exploitation. This modified MFO algorithm is used to solve satellite image thresholding problems. Dataset of five satellite images is taken to evaluate the performance of modified MFO. Clearly, it is shown in results that proposed approach provides better results in comparison with others.

Article 10 [29] Here, MFO is employed to solve the optimal power flow problem (OPF). OPF is one of the necessities of any electrical network. Higher convergence rate and capability of finding good solution make MFO superior than other algorithms. In this work, MFO is implemented on IEEE 30-bus test system. The aim of this work is to reduce the fuel cost, minimize active and reactive power loss. The results show the superior performance of MFO with respect to other popular algorithms.

Inference It is clear from the presented literature survey that MFO has been effectively applied on various optimization problems. Moreover, this is also true that original MFO suffers from local stagnation problem because of less diversity among search agents. The articles in the literature present that various methods have been applied to improve its exploration ability such as chaos theory, crossover and Levy distribution function. But, their work lacks explanation and experimentation analysis. So, these drawbacks have motivated the authors to design new enhanced version of MFO which has been discussed in the consecutive section.

3 Enhanced moth flame optimization

For any generalized optimization algorithm, exploration and exploitation are the two main aspects. Exploration refers to the global search which means searching whole of the search space, whereas exploitation is defined as local search corresponding to searching the small areas inside the large search space. Both these phenomena help the algorithm in avoiding local minima stagnation problem (from exploitation) and provide better convergence and diversity in the solutions (from exploration). Another important feature is the balance between these two phenomena. Any algorithm which is able to achieve these three properties can be considered in the family of state-of-art algorithms. MFO is a recently introduced algorithm, and it can be inferred from the literature that it suffers from the problem of poor global searching capability. Here, it can be said that exploration part of MFO need to be enhanced to make it fit for high-end problems. Also, improving only, the exploration and not the exploitation may not help the algorithm in maintaining a balance between the two. So, in the present work, a new enhanced version of MFO namely E-MFO has been proposed by improving the local as well as global search capability of the original MFO algorithm. The details of proposed modifications are given below:

3.1 Division of iterations

Since exploration and exploitation are the two main aspects of any optimization algorithm, there should be a proper balance between them to achieve the global solution. So during initial iterations, it is required that search agents must move around whole search area and try to explore it instead of getting trapped in some local minima. Hence, search agents gather the information about best solutions during exploration process and toward the later stages, exploit that information and try to reach global minima. Here, it should be noted that too much exploration may cause more randomness and degrade the performance of any algorithm. Similarly, too much exploitation may lead to loss of diversity causing local optima stagnation. A large number of articles have been published in this context, and it is still a matter of concern and more research is to be carried out to know how much exploration and exploitation is to be carried out. In the present work, we have considered 50% exploration and 50% exploitation as the basis and this has been achieved by dividing the total number of iterations into two equal parts. During first half, algorithm will tend to perform more exploration and in second half it will emphasize on exploitation. This is ensured by utilization of Cauchy function and influence of best flame as discussed in following subsections. Cauchy function is used in first half iterations to improve the exploration capability and in other half iterations, influence of best flame is taken in order to enhance exploitation capability. Thus, it ensures a proper balance between the exploration and exploitation.

3.2 Exponential step size

The parameter t describes the next position of moth with respect to flame. In basic MFO, the step size t, varies from r to 1 where r is a linearly decreasing function from -1 to -2 and is proportional to iterations. According to Fig. 3, when value of t is high, it means the position of moth is far from the flame so moth has to take large steps (exploration) to reach at the destination while when the value of t is less, step size is less (exploitation) and it will reach the destination in limited time. A proper t value should be there to ensure balanced exploration and exploitation. So, in proposed approach exponential function helps to increase the exploration rate with respect to iterations in the initial stages and also exploitation in the later stages. In other words, using exponential function helps to maintain a proper balance between exploration and exploitation. The value of r will change slowly. The equation used for this is as follows:

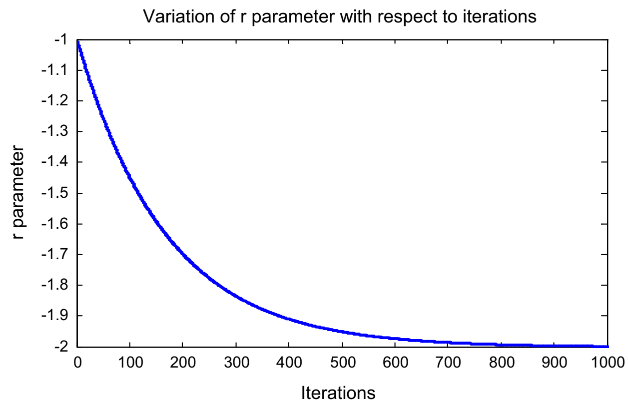


Fig. 5 Exponential curve for step size

$$r = \frac{-2}{1 + e^{\left(-6*\frac{L}{T}\right)}} \tag{7}$$

where L is the current iteration and T represents the maximum iterations. Here, the value of r varies from -1 to -2 as shown in Fig. 5.

3.3 Cauchy jump

This modification involves the use of Cauchy distribution to generate a new random number. Cauchy distribution allows the search agents to explore the search space in a much better way. The reason for superior performance of Cauchy-based random number is because of the heavy-tailed structure of Cauchy distribution. It helps the candidate solution to jump out of local minima and avoid the premature convergence problem. The Cauchy-based random number is utilized in each iteration after the position update by Eq. (3) using Eq. (8). This process is followed only during the first half of the maximum number of iterations, and this is because of the need for extensive exploration at the initial stage.

$$M_i = M_i + \text{sign}(\text{rand} - 0.5) * C(\delta) \tag{8}$$

Here, $\text{sign}(\text{rand} - 0.5)$ may take three values 1, 0 and -1. Combination of sign (rand $-$ 0.5) and Cauchy jump leads to do more random walk with large step sizes. Here, Cauchy operator will generate a random number using Cauchy distribution formula which is given as:

$$y = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{\delta}{g}\right) \tag{9}$$

The Cauchy density function calculated as:

$$f_{\text{Cauchy}(0,g)}(\delta) = \frac{1}{\pi}\frac{g}{g^2 + \delta^2} \tag{10}$$

The parameter g is set to 1 and y belongs to range of [0 1]. After solving Eq. (9), we get

$$\delta = \tan\left(\pi\left(y - \frac{1}{2}\right)\right) \tag{11}$$

This modification is only applied for first half iterations so that requirement of more exploration can be fulfilled.

3.4 Influence of best flame

The basic paper of MFO states that each moth updates its position with respect to its corresponding flame. There is sequence of flames, which are assigned to moths in each iteration. First moth will move for best flame and last will move toward worst flame. This is not the actual behavior of moths in environment. However, assigning each moth a flame promotes the exploration but at the same time, it degrades the exploitation capability.

In proposed approach, the influence of best flame is taken for all the moths. Here, all the moths will update their position according to their corresponding flame and the best flame. The moths will move toward average position of best and its corresponding flame. Since the moths are moving toward the best flame, so this modification promotes the exploitation and helps in improving the convergence speed. The effect of best flame is taken only during second half of iterations where more exploitation is the requirement. The newly updated equation will become:

$$S\left(M_i, F_j\right) = D_i.e^{bt}.\text{Cos}(2\pi t) + \frac{a * F_j + (1 - a) * \text{best flame}}{2} \tag{12}$$

where

$$a = 0.6 - 0.4 * \frac{L}{T} \tag{13}$$

Here, parameter "a" is linearly decreasing function and is used as controlling factor which controls the effect of best and corresponding flame. The value of "a" varies from 0.4 to 0.2 with respect to iterations. In initial stage, the value of a parameter is kept high which signifies that the effect of corresponding flame is more so that moth can explore the search space efficiently. After that in later stages, the value of a is decreased which helps the algorithm to move toward best flame and improve the convergence speed. The pseudo-code for proposed approach is given in Fig. 6.

4 Results and discussion

In this section, we analyze the effectiveness of proposed algorithm using benchmark functions. Also, the results of E-MFO are compared to basic MFO as well as other metaheuristic algorithms. The results are taken at different

```
Initialize the position of moths
While (iteration<= Maximum iteration)
    if (iteration <=Maximum iteration/2)
        Update flame Number using equation (5)
        OM=Fitness Function (M)
        If iteration==1
            F=sort (M)
            OF=sort (OM)
        Else
            F=sort (M_t − 1, M_t)
            OF=sort (M_{t−1}, M_t)
        End
        For i =1: n
            For j=1: d
                Change r and t
                Calculate D using (4)
                Evaluate M (i, j) using (3) and (1)
            End
        Update M (i, j) using (8)
    End
else
    Update flame Number using equation (5)
    OM=Fitness Function (M)
    If iteration==1
        F=sort (M)
        OF=sort (OM)
    Else
        F=sort (M_t − 1, M_t)
        OF=sort (M_{t−1}, M_t)
    End
    For i =1: n
        For j=1: d
            Change r and t
            Evaluate D using (4)
            Calculate M (i, j) using (12) and (1)
        end
    end
    end
end
```

Fig. 6 Pseudo-code for enhanced moth flame optimization (E-MFO)

population and dimension size to evaluate the effect of population size on algorithm.

4.1 PC configuration and parameter settings

For performance evaluation, results are taken using 32-bit Windows 7, Intel Core i3-4200U, CPU@ 1.6GHZ processor with 4 GB RAM and using MATLAB version R2007b. All the algorithms are run for 50 times and 500 iterations. The results of E-MFO are compared with firefly algorithm (FA), flower pollination algorithm (FPA) [30], bat algorithm (BA), bat flower pollination algorithm (BFP) [31], genetic algorithm (GA) [12], particle swarm optimization (PSO) [4] and differential evolution algorithm (DE). The various parameters of these algorithms are given in Table 1.

4.2 Test suite

To check the performance, this algorithm has been verified on twenty benchmark functions. These benchmark functions include unimodal functions, multimodal functions and multimodal functions with fixed dimension. Unimodal functions are those which have only one global minima on the other hand multimodal functions have a large number of local minima. Here, it should be noted that unimodal functions help to check the exploitative ability of an algorithm, whereas multimodal functions help in keeping a check on the explorative capabilities of any algorithm. The benchmark functions are discussed in more detail in Tables 2, 3 and 4.

4.3 Effect of each modification

As it is discussed in Sect. 3, four modifications have proposed to moth flame optimization to increase the performance of algorithm. In this section, the effect of individual modification is analyzed on MFO. Table 5 shows the comparison of MFO with different cases of E-MFO. In case 1, only one modification namely adaptive step size is considered. According to this modification, the step size is varied as an exponential curve and the results show that out of twenty functions, the proposed approach performed better than MFO for only seven functions and for six functions it provides comparable results with respect to MFO. This shows that E-MFO is able to achieve global minima and make balance between exploration and exploitation for only a few set of functions. For case 2, the use of Cauchy function and its effect on MFO is analyzed. Cauchy function has been employed to increase the exploration of algorithm. Table 5 shows that for eleven functions E-MFO performed better in

Table 1 Parameter settings for various algorithms

Algorithm	Parameter settings
BA	Loudness (A) = 0.5, pulse rate (r) = 0.5
FPA	Switch probability (p) = 0.8
DE	Control parameter (F) = 1.5, crossover rate (CR) = 0.8
BFP	Loudness (A) = 0.5, pulse rate(r) = 0.5, switch probability (p) = 0.8
FA	Randomization parameter(α) = 0.25, attractiveness (β_o) = 1, light absorption coefficient (γ) = 1
GA	Crossover rate = 0.7, mutation rate = 0.2
PSO	Acceleration coefficients c1, $c2$ = 1 and $c3$ = 0
MFO	b = 1
E-MFO	b = 1

Table 2 Unimodal functions

Equation	Global minima	Dim	Range
$f_1(x) = \sum_{i=1}^{D} x_i^2$	0	30	[– 100 100]
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	0	30	[– 10, 10]
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	0	30	[– 100, 100]
$f_4(x) = max_i\{\lvert x_i \rvert, 1 \le i \le n\}$	0	30	[– 100, 100]
$f_5(x) = \sum_{i=1}^{N-1} (100(x_{i+1} - x_i)^2 + (x_i - 1)^2$	0	30	[– 100 100]
$f_6(x) = \sum_{i=1}^{n} ix_i^4 + \text{random } [0, 1)$	0	30	[– 1.28, 1.28]
$f_7(x) = \left[\frac{1}{n-1} \sqrt{s_i} \cdot \left(\sin\left(50.0 s_i^{\frac{1}{5}}\right) + 1\right)\right]^2 = \sqrt{x_i^2 + x_{i+1}^2}$	0	30	[– 100 100]
$f_8(x) = x_0^2 + 10000 \sum_{i=1}^{D} x_i^2$	0	30	[– 10 10]
$f_9(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$	0	50	[– 100 100]
$f_{10}(x) = \sum_{i=1}^{D} x_i^{i+1}$	0	30	[– 1 1]
$f_{11}(x) = \sum_{i=1}^{D} ix_i^2$	0	30	[– 10 10]
$f_{12}(x) = \sum_{i=1}^{d} \left[(x_{4i-3} - 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^2\right]$	0	30	[– 4 5]

Table 3 Multimodal functions

Equation	Global minima	Dim (d)	Range
$f_{13}(x) = 10D + \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i)\right]$	0	30	[– 5.12 5.12]
$f_{14}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	0	30	[– 100 100]
$f_{15}(x) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0	30	[– 600 600]
$f_{16}(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^4$	0	30	[– 5 10]
$f_{17}(x) = sin^2(\pi w_1) + \sum_{i=1}^{d-1}(w_i - 1)^2[1 + 10sin^2(\pi w_i + 1)] + (w_d - 1)^2[1 + 10sin^2(2\pi w_d)]$	0	30	[– 10 10]
$f_{18}(x) = \sum_{i=1}^{d}(\lvert x_i \rvert)$	0	30	[– 100, 100]

comparison with MFO and for five functions E-MFO provides comparable results. The use of Cauchy operator provides diverse solutions that lead to improvement in exploration and hence global minima is achieved for more number of functions. In the last case, influence of best flame is considered to improve the exploitation ability and the results show that almost all the functions except two provide better average and standard deviation value. The fourth modification employed is division of iterations whose results are shown in Table 6 under the next section. The next section gives details of the effect of population size on the performance of MFO and the proposed approach. Apart from this, the bold values in the result tables corresponds to the best values for the particular algorithm.

4.4 Effect of population

To investigate the effect of population on algorithm, all the algorithms including E-MFO, MFO, BA, BFP, DE, FA, GA, PSO and FPA runs for 30, 60 and 100 population size.

Table 4 Multimodal functions with fixed dimensions

Equation	Global minima	Dim (d)	Range
$f_{19}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	− 1.0316	2	[− 5, 5]
$f_{20}(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] * \left[30 + (2x_1 - 3x_2)^2 * (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$	3	2	[− 2, 2]

Table 5 Effect of each modification on performance of E-MFO

Function	MFO		Case 1 (exponential step size)		Case 2 (use of cauchy function)		Case 3 (influence of best flame)	
	Average	SD	Average	SD	Average	SD	Average	SD
f_1	2.80E+03	5.72E+03	18.0247	38.80778	3.667178	1.879578	2.95E−27	9.34E−27
f_2	1.2000	3.854548	26.8356	22.23867	2.884945	0.479923	2.16E−15	5.70E−15
f_3	4.00E+02	1.37E+03	2.16E+04	1.28E+04	8.09E+03	3.20E+03	3.36E−15	1.43E−14
f_4	2.241167	4.5042567	71.3737	7.79915	41.76514	7.639836	1.95E−11	2.50E−11
f_5	3.99E+08	1.97E+09	1.46E+03	3.24E+03	9.72E+03	7.98E+03	28.05619	0.27598
f_6	0.0059	0.0266	5.42E+03	1.69E+04	0.3436	0.113586	2.96E−04	1.75E−04
f_7	5.04E−04	0.002345	5.87E−06	4.15E−05	9.47E−11	7.93E−11	0.00E+0	0.00E+0
f_8	2.07E+08	2.40E+07	8.16E+04	2.73E+05	4.08E+03	1.08e+03	5.84E−25	1.45E−24
f_9	3.65E+07	5.84E+07	3.54E+05	4.93E+05	2.10E+05	1.55E+05	2.23E−23	7.81E−23
f_{10}	7.23E−16	3.52E−15	1.98E+02	3.18E+02	3.07E−05	1.76E−05	1.5E−100	6.7E−100
f_{11}	6.44E+02	6.59E+02	2.42E+02	3.84E+02	7.63192	1.82359	2.50E−27	7.74−27
f_{12}	8.89E+02	1.12E+03	9.62E+03	1.15E+04	34.5514	17.1964	3.11E−24	1.17E−23
f_{13}	1.53E+02	37.32309	1.83E+02	59.78736	1.45E+02	24.5807	0.00E+0	0.00E+0
f_{14}	19.9999	5.36E−07	0.046788	2.656469	20.0184	0.01057	2.91124	7.28818
f_{15}	16.31648	39.52865	0.113356	0.23433	0.923867	0.17978	0.00E+0	0.00E+0
f_{16}	3.02E+02	1.18E+02	4.55E+02	1.95E+02	1.28E+02	31.688	2.21E−14	6.56E−14
f_{17}	30.11427	12.76732	33.29178	9.317677	2.30E−08	2.41E−08	1.86862	0.13461
f_{18}	9.21E+05	4.08E+04	8.02E+02	3.38E+03	0.472680	0.114508	3.74662	0.392689
f_{19}	− 1.03162	2.24E−16	− 1.031628	2.242E−16	− 1.03162	4.74E−07	− 1.03162	4.58E−06
f_{20}	3.000000	2.08E−15	3.000000	2.015E−15	3.00000	2.33E−05	3.0000	3.52E−06

The analysis of their results is shown in Tables 6, 7 and 8 for population size 30, 60 and 100, respectively.

Case 1: Population size 30

In the first case, population size and dimension both are considered to be equal to 30. From Table 6, it is clear that for functions f_1,f_2,f_3 and f_4 E-MFO outperforms other algorithms in terms of best, worst, average and standard deviation. For function f_5, E-MFO gives comparable results with respect of MFO in terms of best solution and better results in terms of standard deviation. Further in case of f_6, E-MFO provides better results in comparison with others. Although best value in case of f_7 is same for algorithms like MFO, E-MFO, DE and FPA, standard deviation of E-MFO is better. The functions f_8,f_9,f_{10},f_{11} and f_{12} come up with very good results in case of E-MFO. Their values are near to global optimum values and better than other algorithms. In case of f_{13} and f_{14}, E-MFO outperforms other algorithms and f_{13} provides exact optimum values. For function f_{15} provide exact global solutions in case of E-MFO. Function f_{16} offer better results in terms of best, worst, average and standard deviation for E-MFO algorithm. Function f_{17} gives comparable results for FA and E-MFO in terms of standard deviation, but FA performs better than other algorithms because its best solution is close to global minima. Moreover, for function f_{18}, only GA provides optimum minimum values. For function f_{19}, MFO outperforms other algorithms with its better standard deviation and for this particular function BA and BFP algorithm's performance is very poor. MFO, DE, FPA provides highly competitive results for function f_{20}; however, MFO algorithm considered as best due to its slightly higher standard deviation.

Table 6 Results of E-MFO compared with other algorithms for population size 30

Function	Performance measure	Algorithm								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_1	Best	**1.51E−73**	4.62E−06	6.09E+03	2.29E+04	4.80E+03	0.001627	3.45E+02	1.340433	0.244519
	Worst	**9.04E−69**	2.00E+04	3.99E+04	8.42E+04	8.38E+04	0.0055095	1.99E+03	5.649403	0.768875
	Average	**3.06E−70**	2.80E+03	1.95E+04	6.32E+04	5.49E+04	0.0029205	8.97E+02	2.300848	0.421184
	SD.	**1.29E−69**	5.72E+03	8.10E+03	1.21E+04	2.18E+04	7.93E−04	3.30E+02	0.689615	0.148242
f_2	Best	**4.23E−50**	3.23E−20	0.799159	12.89518	0.058768	0.058768	0.264359	0.007761	1.24E+04
	Worst	**1.01E−56**	20.0000	54.363152	50.763038	16.608305	0.0063620	3.7517812	5.7899750	1.25E+04
	Average	**1.41E−57**	1.20000	20.212749	33.154089	3.1239795	0.0027869	1.4495611	2.2054760	1.25E+04
	SD.	**2.05E−57**	3.85454	14.369427	7.8308621	3.3459719	7.27E−04	0.7239958	0.9888468	5.6077532
f_3	Best	**1.54E−98**	1.60E−11	2.45E+02	8.06E+03	9.198866	8.10E−05	0.031425	0.003559	5.47E−13
	Worst	**3.33E−92**	5.00E+03	1.53E+04	3.53E+04	7.69E+03	8.50E−04	0.6695819	3.8877083	5.28E−09
	Average	**3.19E−93**	4.00E+02	5.36E+03	1.85E+04	1.47E+03	2.70E−04	0.136197	2.226970	1.23E−09
	SD.	**7.71E−93**	1.37E+03	2.89E+03	5.65E+03	1.92E+03	1.60E−04	0.1103161	0.7797089	1.36E−09
f_4	Best	**2.00E−55**	4.39E−06	14.50311	48.09316	1.723137	0.002353	0.480584	2.738645	0.528199
	Worst	**1.20E−52**	21.81578	61.435179	83.318482	64.270224	0.0112001	2.6076867	8.9232292	0.530233
	Average	**1.33E−53**	2.241167	35.658991	65.699653	28.148822	0.0062217	1.3954512	7.8646029	0.5289957
	SD	**2.23E−53**	4.5042567	9.8208726	8.2987753	17.241827	0.0016856	0.4344367	0.8749623	5.13E−04
f_5	Best	**27.21687**	62.24768	3.83E+08	8.16E+09	2.68E+09	27.43123	5.15E+05	1.155150	51.09970
	Worst	**28.839765**	9.97E+09	1.36E+10	4.10E+10	4.22E+10	1.39E+04	3.80E+07	5.159018	1.33E+02
	Average	**28.068967**	3.99E+08	3.07E+09	2.78E+10	3.08E+10	1.45E+03	1.07E+07	2.402547	78.2156
	SD	**0.192879**	1.97E+09	2.28E+09	8.58E+09	6.90E+09	3.03E+03	8.91E+06	0.797356	18.94955
f_6	Best	**1.63E−06**	0.002000	0.054085	0.692697	0.009755	0.003236	0.004622	0.003567	0.359427
	Worst	**2.60E−04**	0.017547	4.6075454	11.896073	2.1225818	0.1490810	0.0302129	4.4471877	1.417900
	Average	**6.88E−05**	0.0059	0.7965411	5.5161543	0.226336	0.0298927	0.0298927	2.2318054	0.889035
	SD	**5.86E−05**	0.0266	0.8246265	2.2357915	0.4254076	0.0282967	0.0058381	0.754118	0.290415
f_7	Best	**0.00E+00**	**0.00E+00**	1.77E−13	0.004275	0.00E+00	2.29E−12	0.00E+00	3.07E−04	9.76E−15
	Worst	**0.00E+00**	0.0153561	0.3928647	0.4663736	0.2575325	5.66E−10	9.60E−13	0.090058	9.01E−09
	Average	**0.00E+00**	5.04E−04	0.1190811	0.2946212	0.0335426	1.20E−10	6.38E−14	0.023736	1.21E−09
	SD	**0.00E+00**	0.002345	0.1114218	0.128708	0.0510467	1.10E−10	1.69E−13	0.016997	2.37E−09
f_8	Best	**3.15E−69**	0.001179	8.55E+04	2.96E+06	4.80E+05	0.21042	3.83E+04	0.017600	2.41E+03
	Worst	**1.82E−64**	1.00E+08	1.80E+06	7.98E+06	7.75E+06	1.29E+02	2.11E+05	4.297074	6.87E+03
	Average	**6.52E−66**	2.07E+08	5.64E+05	5.90E+06	4.78E+06	31.256325	8.25E+04	2.144305	4.07E+03
	SD	**2.75E−65**	2.40E+07	3.77E+05	1.10E+06	2.29E+06	37.390463	3.34E+04	0.728043	1.12E+03
f_9	Best	**1.95E−68**	5.40E+05	1.38E+08	9.42E+08	6.66E+07	1.29E+06	8.18E+05	0.931311	1.55E+02
	Worst	**1.32E−62**	3.74E+08	2.31E+09	4.38E+09	1.86E+09	1.86E+07	9.68E+06	4.029094	1.04E+03
	Average	**3.84E−64**	3.65E+07	7.88E+08	2.37E+09	6.37E+08	6.24E+06	3.08E+06	2.116899	0.645965
	SD	**1.74E−64**	5.84E+07	4.89E+08	7.85E+08	5.02E+08	3.72E+06	1.88E+06	3.48E+02	1.89E+02

Table 6 (continued)

Function	Performance measure	Algorithm								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_{10}	Best	**1.27E−316**	2.96E−23	1.25E−08	0.020192	0.288252	2.89E−08	6.09E−10	0.93136	1.18E−04
	Worst	**2.02E−299**	2.29E−14	1.25E−07	0.6319118	1.5353542	1.18E−06	5.09E−07	4.074251	0.002562
	Average	**7.73E−301**	7.23E−16	5.00E−08	0.2592852	0.775902	2.82E−07	4.01E−08	2.044825	8.53E−04
	SD	**0.00E+00**	3.52E−15	2.66E−08	0.1612362	0.294377	2.45E−07	5.44E−07	0.626943	6.69E−04
f_{11}	Best	**4.01E−72**	1.63E−06	2.15E+02	5.10E+03	8.24E+02	0.007319	44.46101	0.932904	4.725698
	Worst	**3.36E−67**	2.40E+03	3.44E+03	1.15E+04	1.23E+04	1.8620043	2.72E+02	4.211492	17.37789
	Average	**1.11E−68**	6.44E+02	9.30E+02	8.99E+03	7.10E+03	0.273306	1.13E+02	2.097958	8.049821
	SD	**4.72E−68**	6.59E+02	5.48E+02	1.37E+03	3.63E+03	0.3845407	52.064633	0.6726278	2.5074356
f_{12}	Best	**1.98E−124**	0.02636	0.037852	4.34E+03	3.67E+02	0.1822	4.952664	0.003565	0.045471
	Worst	**1.59E−112**	3.85E+03	2.24E+02	2.32E+04	1.15E+04	6.5718834	77.383768	3.9343764	0.2085276
	Average	**7.59E−114**	8.89E+02	34.225633	1.11E+04	2.64E+03	2.1574333	24.34227	2.158556	0.114570
	SD	**2.64E−113**	1.12E+03	50.957114	4.00E+03	2.22E+03	1.4794017	13.517725	0.7758192	0.0452078
f_{13}	Best	**0.00E+00**	69.64709	39.80093	2.83E+02	1.56E+02	23.88061	98.87448	0.931328	84.90325
	Worst	**0.00E+00**	2.30E+02	2.27E+02	4.23E+02	4.93E+02	75.618224	1.70E+02	5.350435	1.64E+02
	Average	**0.00E+00**	1.53E+02	99.100651	3.66E+02	3.62E+02	47.779669	1.26E+02	2.2240769	1.28E+02
	SD	**0.00E+00**	37.32309	45.018746	32.738443	1.12E+02	13.571661	13.248671	0.7339709	20.530824
f_{14}	Best	**8.88E−16**	19.9999	19.9709	20.9921	20.1399	0.04561	20.9167	0.00312	1.12623
	Worst	**20.7710**	19.9999	20.0000	21.2619	20.5298	20.0058	21.1306	5.79009	2.99883
	Average	**12.78624**	19.9999	19.9999	21.1331	20.364	19.5999	21.0462	2.21025	1.83966
	SD	**9.95335**	5.36E−07	0.00425	0.07481	0.10511	2.82185	0.04743	5.79007	0.44622
f_{15}	Best	**0.00E+00**	1.68E−05	94.00835	3.26E+02	2.70E+02	0.002136	3.450933	1.079650	0.022135
	Worst	**0.00E+00**	1.81E+02	4.19E+02	7.47E+02	7.48E+02	0.0150722	17.305742	1.1435555	0.0752721
	Average	**0.00E+00**	16.31648	1.94E+02	5.84E+02	5.82E+02	0.0055429	8.8112189	1.1153061	0.0352287
	SD	**0.00E+00**	39.52865	65.96575	93.30520	1.06E+02	0.002858	2.953524	0.013529	0.010178
f_{16}	Best	**1.02E−43**	32.42624	18.90162	4.13E+02	7.97E+02	9.239959	1.45E+02	1.340509	0.038107
	Worst	**3.57E−39**	6.01E+02	4.69E+04	6.37E+08	6.06E+04	1.18E+02	8.33E+02	4.737258	0.103731
	Average	**2.18E−40**	3.02E+02	3.81E+03	3.48E+07	1.26E+04	46.929186	3.30E+02	2.3183725	0.0736421
	SD	**6.16E−40**	1.18E+02	9.12E+03	1.21E+08	1.12E+04	28.13372	1.91E+02	0.6226917	0.0162481
f_{17}	Best	1.356978	10.08956	7.054811	1.21E+02	31.31569	**7.86E−06**	2.573172	0.001522	1.206398
	Worst	1.956867	88.30436	46.566024	2.53E+02	2.38E+02	**0.9089115**	22.556051	3.9818473	4.3763577
	Average	1.666989	30.11427	17.888414	1.80E+02	92.044334	**0.0855539**	9.5783313	2.1327605	1.6611856
	SD	0.133898	12.76732	8.837656	30.08757	46.56949	**0.191229**	4.728633	0.724253	0.693433
f_{18}	Best	**1.15E+02**	8.02E+05	2.35E+05	6.83E+05	1.46E+06	3.21E+04	5.52E+04	0.004416	**0.00E+00**
	Worst	**1.19E+02**	1.00E+06	1.27E+06	1.56E+06	1.58E+06	8.28E+04	8.59E+04	3.734013	**0.00E+00**
	Average	**1.17E+02**	9.21E+05	4.16E+05	1.35E+06	1.53E+06	4.87E+04	7.12E+04	2.031640	**0.00E+00**
	SD	**0.814615**	4.08E+04	1.59E+05	2.85E+05	2.95E+04	1.14E+04	6.37E+03	0.637453	**0.00E+00**

Table 6 (continued)

Function	Performance measure	Algorithm								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_{19}	Best	− 1.031638	− 1.0316285	− 1.0316285	− 1.0316285	− 1.0316285	− 1.0316285	− 1.0316285	− 0.98413	− 1.0316285
	Worst	− 1.031628	− 1.0316285	− 1.0316285	− 1.0316285	− 1.0316285	− 1.0316285	− 1.0316285	9.5603431	− 1.0316285
	Average	− 1.031628	− 1.0316285	− 1.0316285	− 1.0316285	− 1.0316285	− 1.0316285	− 1.0316285	1.8330968	− 1.0316285
	SD	1.35E−09	**2.24E−16**	0.2236679	0.4709543	8.63E−09	1.39E−09	2.23E−10	2.297242	1.13E−05
f_{20}	Best	3.0000000	3.0000000	3.0000000	3.0000000	3.0000000	3.0000000	3.0000000	0.0129994	3.0000010
	Worst	3.0000000	3.0000000	84.000000	96.485886	3.0000000	3.0000000	3.0000000	4.029097	3.002571
	Average	3.0000000	3.0000000	10.560000	15.342473	3.0000000	3.0000000	3.0000000	2.083875	3.0005115
	SD	1.18E−05	**2.08E−15**	22.517259	19.364906	6.94E−15	1.45E−08	3.60E−15	0.722614	7.82E−04

Case 2: Population size 60

In this case, all the algorithms are run for 60 population size to evaluate the effect of change of population size. Starting from functions f_1,f_2, f_3 and f_4, it is observed from Table 7 that E-MFO performs better in comparison with other algorithms. For function f_5, PSO provides better standard deviation and best solution in comparison with others. The function f_6 gives better results in case of E-MFO. In case of f_7, the results provided by MFO and E-MFO are same as well as better than others. Also, both of these algorithms have achieved global optimum values. For functions f_8, f_9, f_{10}, f_{11} and f_{12} provides best results in case of E-MFO. Function f_{13} gives exact optimum results in E-MFO, whereas other algorithms get stuck in local minima. The results provided by functions f_{14} and f_{15} are very close to optimum values in case of E-MFO only. For f_{16}, E-MFO performs better in comparison with all other given algorithms. Although for f_{17}, E-MFO is better in comparison with MFO but FA and PSO give better results in this case. For function f_{18}, GA outperforms over all other algorithms. Further, for both f_{19} and f_{20}, MFO gives better results in terms of standard deviation and best solution provided by all the algorithms is same.

Case 3: Population size 100

As shown in Table 8, the E-MFO algorithm outperforms other algorithms for functions f_1,f_2,f_3 and f_4. The optimum values obtained are very close to global minima. For function f_5, best solution is provided by PSO. E-MFO again outperforms other algorithms in case of f_6. For f_7, MFO and E-MFO both give optimum solution. Further, in case of f_8, f_9, f_{10}, f_{11} and f_{12}, E-MFO is efficiently capable of avoiding local stagnation problem by providing results close to global optima. For function f_{13} only E-MFO provides exact optimum value. The best, worst and average value of f_{14} are same in case of E-MFO that is why its standard deviation is zero. In case of function f_{15}, E-MFO achieves exact global minima, whereas other algorithms get stuck in some local minima. For function f_{16}, E-MFO is able to reach global minima. Further, in case of f_{17}, FA and PSO provide comparable results in terms of best solution and standard deviation but overall PSO considered as best. For f_{18} GA outperforms other algorithms. For the functions f_{19} and f_{20}, all the algorithms except BFP achieve global minima. Also, for functions f_{19} and f_{20}, the results of E-MFO are superior than that of other algorithms.

Inference This test is carried out on 20 benchmark functions for three different population sizes 30, 60 and 100. In all first two cases, MFO is better in two functions, E-MFO is better in sixteen functions, GA and FA are in one function. But for third case, PSO outperforms in one function and rest are same as of previous case. The performance of E-MFO is almost same for all population sizes. On other hand, with increase in population size the

Table 7 E-MFO compared with other algorithms for population size 60

Function	Performance measure	Algorithm								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_1	Best	**7.39E−75**	5.86E−07	4.69E+03	3.30E+04	2.54E+04	0.0016	4.08E+02	1.340433	0.146653
	Worst	**6.18E−69**	2.00E+04	2.19E+04	7.30E+04	7.63E+04	0.005058	1.31E+03	5.649403	0.350927
	Average	**3.55E−70**	1.60E+03	1.11E+04	5.80E+04	5.96E+04	0.002978	8.26E+02	2.300848	0.228596
	SD	**1.10E−69**	4.22E+03	3.80E+03	9.96E+03	1.22E+04	8.77E−04	2.26E+02	0.689615	0.148242
f_2	Best	**8.68E−62**	2.47E−15	0.001855	18.55163	0.073608	0.001319	2.309649	7.74E−08	1.24E+04
	Worst	**1.66E−59**	10.00000	40.11894	49.07007	6.477327	0.01429	5.55647	3.40212	1.25E+04
	Average	**3.99E−60**	0.6	7.718678	32.22959	1.221067	0.002934	3.945466	1.294083	1.25E+04
	SD	**3.78E−60**	2.398979	10.88551	7.622916	1.192882	0.001788	0.918793	0.790401	4.402113
f_3	Best	**3.55E−102**	5.03E−09	2.99E+02	5.75E+03	16.13605	7.92E−05	0.245658	1.67E−07	1.29E−12
	Worst	**1.25E−97**	5.00E+02	1.10E+04	1.94E+04	4.54E+03	8.24E−04	2.33319	3.887708	6.10E−09
	Average	**6.25E−99**	5.00E+02	3.66E+03	1.34E+04	8.75E+02	2.91E−04	0.974603	1.288994	5.96E−10
	SD	**1.95E−98**	5.00E+02	2.12E+03	3.51E+03	9.75E+02	1.58E−04	0.452135	0.888014	1.36E−09
f_4	Best	**9.00E−58**	3.47E−07	7.264344	40.90572	5.097891	0.003689	1.795478	6.394326	0.528102
	Worst	**1.34E−55**	9.36E−05	44.34353	75.04778	37.07507	0.010373	5.38931	8.77383	0.53021
	Average	**1.71E−56**	1.23E−05	25.06516	59.20485	13.85543	0.00644	3.285244	7.871236	0.528932
	SD	**2.54E−56**	1.90E−05	8.340469	6.67291	5.956983	0.001577	0.718641	0.502084	4.66E−04
f_5	Best	27.22451	17.83941	2.56E+08	9.48E+09	1.17E+10	26.93427	1.40E+06	**2.01E−06**	41.32293
	Worst	28.86501	1.00E+06	3.64E+09	3.45E+10	3.79E+10	7.65E+03	2.49E+07	**2.738608**	92.80429
	Average	28.03732	2.41E+05	1.06E+09	2.54E+10	2.76E+10	1.07E+03	9.33E+06	**1.367266**	62.32602
	SD	0.292307	4.31E+05	7.18E+08	6.50E+09	6.02E+09	2.16E+03	4.54E+06	**0.775465**	13.91935
f_6	Best	**2.14E−06**	6.63E−04	0.047906	0.999477	0.007706	0.001542	0.006578	3.49E−08	0.218802
	Worst	**1.42E−04**	0.001835	0.868799	8.414058	0.552526	0.075147	0.033037	3.404158	0.897499
	Average	**3.65E−05**	6.63E−04	0.033037	4.55035	0.096287	0.018333	0.016295	1.235472	0.435527
	SD	**3.39E−05**	0.001835	0.19291	1.855118	0.104485	0.018002	0.006246	0.748848	0.157283
f_7	Best	**0.00E+00**	0.00E+00	2.69E−14	2.35E−04	0.00E+00	1.15E−12	4.44E−16	9.31E−05	6.79E−13
	Worst	**0.00E+00**	0.00E+00	0.297993	0.424208	0.037384	0.003127	9.37E−12	0.055792	3.70E−09
	Average	**0.00E+00**	0.00E+00	0.045054	0.225861	0.002755	2.50E−04	1.01E−12	0.011883	5.36E−10
	SD	**0.00E+00**	0.00E+00	0.061824	0.113423	0.006535	8.57E−04	1.95E−12	0.012097	7.75E−10
f_8	Best	**2.62E−71**	1.58E−04	0.136446	3.26E+06	1.49E+06	0.239096	3.44E+04	4.55E−07	1.15E+03
	Worst	**2.09E−64**	1.00E+06	5.35E+04	6.58E+06	7.74E+06	1.01E+02	1.08E+05	2.495401	3.54E+03
	Average	**1.02E−65**	1.00E+05	8.88E+03	5.02E+06	5.21E+06	15.41739	6.89E+04	1.275894	2.06E+03
	SD	**3.52E−65**	3.03E+05	1.64E+04	9.54E+05	1.79E+06	19.17616	1.70E+04	0.720003	6.13E+02
f_9	Best	**9.54E−126**	7.77E+05	4.34E+07	8.71E+08	4.93E+07	9.09E+05	4.20E+06	2.84E−07	74.94
	Worst	**2.95E−113**	1.53E+08	1.05E+09	3.05E+09	1.53E+09	9.44E+06	1.33E+07	2.957699	4.63E+02
	Average	**9.62E−115**	2.78E+07	3.78E+08	1.91E+09	3.67E+08	3.80E+06	7.14E+06	1.4579615	2.38E+02
	SD	**4.33E−114**	2.82E+07	2.04E+08	5.99E+08	3.20E+08	1.83E+06	2.04E+06	0.615723	92.29513

Table 7 (continued)

Function	Performance measure	Algorithm								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_{10}	Best	**1.02E−297**	4.61E−27	5.04E−09	4.15E−04	0.191838	9.06E−09	1.47E−08	3.10E−07	2.72E−05
	Worst	**2.88E−279**	1.24E−20	2.33E−08	0.159422	1.114176	3.40E−07	1.78E−05	2.579534	6.41E−04
	Average	**1.19E−280**	1.06E−21	1.21E−08	0.050544	0.697173	1.01E−07	2.29E−06	1.118009	1.90E−04
	SD	**0.00E+00**	2.78E−21	4.64E−09	0.039257	0.228109	7.03E−08	3.24E−06	0.859022	1.54E−04
f_{11}	Best	**2.43E−74**	4.17E−07	2.24E−04	2.91E+03	2.03E+03	0.001985	58.69568	3.78E−07	2.387478
	Worst	**1.20E−66**	2.10E+03	1.90E+02	1.03E+04	1.02E+04	0.692639	1.68E+02	4.074286	6.373484
	Average	**3.56E−68**	2.90E+02	18.54717	7.40E+03	6.52E+03	0.158907	1.03E+02	1.390485	4.194906
	SD	**1.74E−67**	4.63E+02	35.36918	1.52E+03	2.30E+03	0.185188	23.64044	0.913822	1.063098
f_{12}	Best	**3.38E−126**	0.010303	0.005755	2.34E+03	4.33E+02	0.090064	12.57818	6.72E−07	0.022752
	Worst	**7.06E−111**	3.71E+03	0.230388	1.29E+04	8.35E+03	6.315861	90.08843	3.46168	0.125734
	Average	**1.42E−112**	7.03E+02	0.017593	6.49E+03	3.27E+03	1.364417	33.40408	1.360676	0.06681
	SD	**9.99E−112**	9.75E+02	0.032417	2.32E+03	1.90E+03	1.149368	15.63032	0.787002	0.022375
f_{13}	Best	**0.00E+00**	55.7176	28.85563	1.91E+02	1.52E+02	19.90106	1.20E+02	1.01E−07	71.1471
	Worst	**0.00E+00**	2.26E+02	1.31E+02	3.69E+02	4.67E+02	70.64366	1.72E+02	2.957893	1.61E+02
	Average	**0.00E+00**	1.38E+02	74.90292	2.94E+02	3.82E+02	34.92485	1.42E+02	1.353922	1.12E+02
	SD	**0.00E+00**	37.70683	23.84936	35.09971	76.49085	10.60093	11.85017	0.826643	24.6819
f_{14}	Best	**8.88E−16**	20.0000	19.97242	20.54088	20.17702	0.032016	20.85079	8.65E−08	0.798293
	Worst	**4.44E−15**	20.00001	20.00003	21.00325	20.50007	20.00211	21.09232	4.584744	2.151323
	Average	**9.59E−16**	20.0000	19.99912	20.81227	20.31751	14.78981	21.01417	1.27446	1.357285
	SD	**5.02E−16**	6.98E−08	0.00393	0.096971	0.088286	8.829073	0.053304	0.876664	0.406447
f_{15}	Best	**0.00E+00**	3.28E−06	31.39266	3.02E+02	1.13E+02	0.002746	5.255169	1.083129	0.010554
	Worst	**0.00E+00**	1.80E+02	2.03E+02	6.90E+02	6.96E+02	0.012703	13.89611	1.132224	0.040282
	Average	**0.00E+00**	19.90595	1.17E+02	5.39E+02	5.39E+02	0.004988	8.428997	1.107867	0.021774
	SD	**0.00E+00**	41.97551	35.87848	90.43191	1.09E+02	0.001994	1.839533	0.012885	0.00706
f_{16}	Best	**2.72E−44**	4.50831	3.89E−05	3.50E+02	7.88E+02	7.42783	2.71E+02	5.04E−07	0.023689
	Worst	**2.58E−38**	5.00E+02	3.76E+03	6.10E+06	2.50E+04	80.46322	1.33E+03	2.738319	0.066056
	Average	**8.88E−40**	2.71E+02	5.68E+02	2.03E+05	5.55E+03	29.94175	6.62E+02	1.278075	0.041497
	SD	**3.72E−39**	1.21E+02	7.81E+02	8.87E+05	5.26E+03	16.61655	2.50E+02	0.804443	0.011497
f_{17}	Best	1.551233	11.72771	4.261282	89.35431	36.03825	**9.60E−06**	8.179657	2.64E−07	0.999288
	Worst	2.18124	98.66733	38.35814	1.85E+02	2.12E+02	**0.08954**	29.76227	2.814099	1.877566
	Average	1.90497	26.84348	11.37969	1.36E+02	94.43659	**0.004246**	18.87273	1.25461	1.368568
	SD	0.130732	13.80313	5.724067	25.96749	42.49181	**0.01782**	4.652033	0.80607	0.20947
f_{18}	Best	2.862444	0.08423	4.25E+03	2.27E+04	4.12E+04	0.006392	3.25E+03	3.77E−07	**0.00E+00**
	Worst	4.551398	1.01E+04	2.88E+04	7.43E+04	7.74E+04	0.026928	7.38E+03	3.093024	**0.00E+00**
	Average	3.864832	1.20E+03	1.25E+04	5.84E+04	6.35E+04	0.015004	4.78E+03	1.293157	**0.00E+00**
	SD	0.327505	3.30E+03	5.34E+03	1.08E+04	8.00E+03	0.00482	9.09E+02	0.799387	**0.00E+00**

Table 7 (continued)

Function	Performance measure	Algorithm								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_{19}	Best	− 1.03163	**− 1.03163**	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 0.94582	− 1.03162
	Worst	− 1.03163	**− 1.03163**	− 0.21546	− 0.21546	− 1.03163	− 1.03163	− 1.03163	8.472472	− 1.03159
	Average	− 1.03163	**− 1.03163**	− 1.01531	− 0.92824	− 1.03163	− 1.03163	− 1.03163	0.67846	− 1.03162
	SD	1.12E−07	**2.24E−16**	0.115423	0.268649	2.62E−16	1.24E−09	1.35E−10	1.730473	7.85E−06
f_{20}	Best	3.00000	**3.00000**	3.00000	3.00000	3.00000	3.00000	3.00000	1.36E−06	3.000001
	Worst	3.00000	**3.00000**	30.000	84.00001	3.00000	3.00000	3.00000	3.399826	3.006784
	Average	3.00000	**3.00000**	4.08	7.860002	3.00000	3.00000	3.00000	1.492235	3.000576
	SD	1.02E−05	**1.66E−15**	5.344614	14.10892	3.16E−15	1.63E−08	7.19E−15	0.620944	0.001335

results of MFO gets better but they are not as good as E-MFO. FA algorithm also provides good and comparable results to MFO as population size increase. BA fails to achieve global minima in all cases. Out of all given algorithms, BFP is proven to be a worst algorithm. After all this discussion, it is concluded that E-MFO gives almost similar results on all populations and also outperforms in comparison with other algorithms. The population size 30 is taken for further experimentation to avoid the increase in number evaluations.

4.5 Effect of dimension (Scalability Study)

The performance of proposed approach is also evaluated on different dimension sizes. Five different dimensions (30, 50, 100, 200 and 500) are considered to study its effect on MFO and E-MFO. The results of MFO and E-MFO are compared with other meta-heuristic algorithms like BA, BFP, DE, FA and FPA. The performance of MFO and E-MFO for different dimension size is evaluated only on unimodal and multimodal functions which are given in Tables 2 and 3. Out of twenty functions, there are two fixed dimension functions whose dimension cannot be changed. Hence, performance is evaluated on eighteen functions. During this experimentation, population size is taken as 30. The results for dimension size 30 are discussed in Sect. 4.3 under population size of 30. From Table 6, it is clear that E-MFO performs better in sixteen functions and FA and GA are good in only one function (f_{17}) for dimension size 30. In case of dimension size 50, E-MFO achieves global minima for all the functions except f_{17} and f_{18} as shown in Table 9. For function f_{17}, FA and for f_{18} GA provides better results. Table 10 shows simulation results for dimension size 100 and results show that E-MFO outperforms in all functions. Simulation results for dimension size 200 are given in Table 11. The results show that GA comes up with good results in f_{14} and for rest of functions, E-MFO gives superior performance. Table 12 depicts the results for dimension size 500. Again, E-MFO provides better results for all the functions except f_{14}. For f_{14} function, all the algorithms provide competitive results in terms of best, worst, average and standard deviation. But, GA considered as best due to better standard deviation.

Inference drawn The effect of dimension is evaluated on five different sizes which are 30, 50, 100, 200 and 500. It is concluded from results that E-MFO outperforms for all dimension sizes. In case of 30 and 50 dimension sizes, E-MFO performed better for sixteen functions and FA for only one. For 100-dimension size, E-MFO outperforms in all functions. In case of 200 and 500 dimension, E-MFO provides good results in all functions except one function. The results show that with increase in dimension size

Table 8 Results of E-MFO and other algorithms for population size 100

Function	Performance measures	Algorithms								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_1	Best	**1.09E−74**	1.35E−06	6.71E+02	3.45E+04	3.62E+04	0.001248	6.97E+02	3.35E−11	0.060652
	Worst	**9.75E−70**	1.00E+04	1.32E+04	6.73E+04	6.90E+04	0.006042	2.79E+03	1.899744	0.201207
	Average	**7.72E−71**	8.00E+02	5.87E+03	5.53E+04	5.91E+04	0.003129	1.46E+03	0.796319	0.129085
	SD	**1.65E−70**	2.74E+03	2.80E+03	7.65E+03	8.05E+03	0.001046	3.66E+02	0.724585	0.034571
f_2	Best	**6.67E−63**	5.52E−15	0.001311	13.82175	0.357783	0.00146	2.630346	1.98E−11	1.24E+04
	Worst	**6.78E−61**	10.00001	43.92926	51.00619	4.386435	0.00415	6.822717	2.495401	1.25E+04
	Average	**1.25E−61**	0.493673	2.575453	30.53602	1.828254	0.002642	5.131648	0.828225	1.24E+04
	SD	**1.40E−61**	1.979487	8.486373	7.196034	0.947588	5.48E−04	0.965264	0.748731	3.378996
f_3	Best	**5.08E−104**	5.13E−11	1.37E+02	4.50E+03	70.24189	1.03E−04	0.655378	1.03E−11	7.46E−14
	Worst	**9.49E−101**	5.00E+03	5.08E+03	1.86E+04	5.63E+03	8.86E−04	3.720408	2.2201127	8.45E−10
	Average	**1.03E−101**	2.00E+02	1.77E+03	1.06E+04	1.32E+03	2.91E−04	2.016671	0.5869963	1.60E−10
	SD	**1.88E−101**	9.90E+02	1.20E+03	3.23E+03	1.10E+03	1.61E−04	0.690654	0.7438574	2.59E−10
f_4	Best	**6.00E−59**	1.21E−07	3.826875	41.07376	6.48024	0.003826	2.769896	2.579534	0.528204
	Worst	**4.23E−57**	1.73E−05	32.38125	70.95115	25.90295	0.011335	6.735047	8.78193	0.529333
	Average	**5.66E−58**	2.68E−06	17.1303	55.61738	15.26117	0.006488	4.724891	7.48623	0.52879
	SD	**7.02E−58**	2.95E−06	6.8725	6.886787	5.192263	0.001515	0.866823	0.881869	3.03E−04
f_5	Best	27.20582	14.17206	1.84E+07	6.30E+09	1.05E+10	26.83713	6.46E+06	**8.60E−12**	36.1991
	Worst	28.83624	1.00E+06	5.73E+08	3.18E+10	3.46E+10	6.43E+03	6.12E+07	**2.738319**	62.17348
	Average	27.80225	1.82E+05	2.46E+08	2.15E+10	2.50E+10	5.52E+02	3.04E+07	**0.811444**	46.65917
	SD	0.444945	3.87E+05	1.47E+08	5.05E+09	5.80E+09	1.37E+03	1.33E+07	**0.822412**	7.330247
f_6	Best	**6.26E−07**	3.32E−04	0.02202	0.476259	0.011644	1.89E−04	0.003502	1.04E−11	0.107517
	Worst	**9.00E−05**	0.004967	0.597634	7.449452	0.19434	0.040244	0.040244	2.12005	0.488061
	Average	**2.27E−05**	0.002475	0.193514	3.448126	0.097394	0.011021	0.015715	0.871319	0.248712
	SD	**1.91E−05**	9.85E−04	0.124682	1.449928	0.042495	0.009171	0.007014	0.655417	0.1000932
f_7	Best	**0.00E+00**	0.00E+00	4.44E−16	3.62E−07	0.00E+00	8.73E−12	1.55E−15	2.13E−05	6.61E−13
	Worst	**0.00E+00**	0.00E+00	0.148222	0.381139	0.013342	0.00809	9.84E−12	0.025851	8.01E−10
	Average	**0.00E+00**	0.00E+00	0.019866	0.150564	4.28E−04	4.04E−04	6.30E−13	0.009894	1.08E−10
	SD	**0.00E+00**	0.00E+00	0.026131	0.105044	0.001943	0.001509	1.50E−12	0.007531	1.82E−10
f_8	Best	**3.16E−71**	21.56645	0.371805	3.63E+06	9.17E+06	4.519641	2.69E+05	1.09E−11	4.22E+02
	Worst	**8.27E−65**	2.00E+06	40.63813	1.08E+07	1.31E+07	70.57294	7.04E+05	3.284006	2.11E+03
	Average	**2.21E−66**	5.40E+05	5.995336	8.76E+06	1.14E+07	20.10809	4.91E+05	0.732694	1.21E+03
	SD	**1.17E−65**	7.06E+05	8.215169	1.53E+06	8.62E+05	11.73151	1.14E+05	0.770726	3.55E+02
f_9	Best	**8.36E−131**	2.53E+05	2.10E+07	3.38E+08	5.05E+07	5.04E+05	7.78E+06	7.33E−11	76.73029
	Worst	**6.56E−113**	1.60E+08	4.48E+08	2.35E+09	1.07E+09	6.49E+06	3.07E+07	2.579534	2.65E+02
	Average	**2.72E−114**	2.29E+07	1.23E+08	1.50E+09	3.04E+08	2.05E+06	1.70E+07	0.800176	1.56E+02
	SD	**1.22E−113**	3.21E+07	7.94E+07	4.19E+08	2.22E+08	1.27E+06	4.23E+06	0.785399	55.28248

Table 8 (continued)

Function	Performance measures	Algorithms								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_{10}	Best	**2.11E−303**	3.71E−17	1.50E−09	1.78E−04	0.305767	5.73E−09	3.19E−06	1.88E−11	1.82E−05
	Worst	**4.10E−283**	6.83E−11	1.07E−08	0.109786	1.213377	2.57E−07	2.18E−04	2.579535	2.83E−04
	Average	**9.35E−285**	2.85E−12	5.08E−09	0.009186	0.772623	5.63E−08	4.96E−05	0.703784	9.69E−05
	SD	**0.00E+00**	1.21E−11	2.14E−09	0.017004	0.200543	5.43E−08	4.74E−05	0.737577	6.52E−05
f_{11}	Best	**1.85E−73**	0.078573	0.001845	8.92E+03	2.10E+04	0.646657	6.56E+02	7.21E−11	1.341918
	Worst	**1.19E−67**	5.80E+03	4.06E+01	2.78E+04	3.17E+04	18.87371	1.88E+03	2.120053	4.080163
	Average	**3.90E−69**	2.25E+03	0.969132	2.06E+04	2.75E+04	4.048379	1.11E+03	0.621615	2.662254
	SD	**1.75E−68**	1.80E+03	5.753953	4.24E+03	2.28E+03	3.461136	2.47E+02	0.7352893	0.677056
f_{12}	Best	**3.01E−125**	2.926889	0.013429	5.32E+03	7.62E+03	0.908233	1.82E+02	3.24E−11	0.015116
	Worst	**6.10E−110**	9.41E+03	0.029649	2.04E+04	4.00E+04	13.16133	6.25E+02	1.777996	0.084356
	Average	**1.33E−111**	2.57E+03	0.021995	1.25E+04	2.03E+04	5.862109	3.80E+02	0.761918	0.042655
	SD	**8.62E−111**	2.41E+03	0.003142	3.49E+03	7.25E+03	3.072925	1.14E+02	0.684462	0.01647
f_{13}	Best	**0.00E+00**	1.34E+02	50.75047	3.27E+02	6.49E+02	28.87913	2.70E+02	3.33E−11	57.756
	Worst	**0.00E+00**	3.30E+02	1.93E+02	5.42E+02	7.92E+02	77.68683	3.62E+02	2.220968	1.53E+02
	Average	**0.00E+00**	2.45E+02	1.06E+02	4.66E+02	7.49E+02	48.77846	3.14E+02	0.665841	89.89446
	SD	**0.00E+00**	42.85861	33.18021	57.45631	32.70928	11.10025	20.98795	0.769335	19.90981
f_{14}	Best	**8.88E−16**	20.00001	19.95783	20.21452	20.13511	0.025757	20.86859	2.83E−11	0.468857
	Worst	**8.88E−16**	20.0001	20.00003	20.73213	20.65241	19.99658	21.08082	1.899744	1.896023
	Average	**8.88E−16**	20.00005	19.99831	20.47791	20.39644	4.823835	20.98318	0.683063	1.086094
	SD	**0.00E+00**	8.83E−08	0.00671	0.105519	0.133967	8.595871	0.048197	0.695375	0.439883
f_{15}	Best	**0.00E+00**	0.274703	82.67837	3.98E+02	8.38E+02	0.007461	25.53438	3.33E−11	0.0062753
	Worst	**0.00E+00**	2.71E+02	2.35E+02	1.15E+03	1.21E+03	0.019512	61.80807	2.220969	0.021375
	Average	**0.00E+00**	49.29168	1.48E+02	9.48E+02	1.03E+03	1.19E−02	43.94409	0.665841	0.0130239
	SD	**0.00E+00**	70.97532	33.98977	1.79E+02	85.27715	0.002474	8.659456	0.769335	0.0036082
f_{16}	Best	**3.47E−48**	2.71E+02	0.001276	9.31E+02	2.78E+03	61.22957	7.83E+02	2.62E−11	0.014639
	Worst	**1.02E−39**	8.65E+02	1.33E+04	9.97E+07	7.27E+04	1.95E+02	4.86E+03	1.899744	0.041701
	Average	**3.96E−41**	6.55E+02	1.35E+03	3.09E+06	1.70E+04	1.18E+02	2.58E+03	0.6240675	0.0249626
	SD	**1.50E−40**	1.35E+02	2.04E+03	1.44E+07	1.72E+04	30.76644	9.32E+02	0.7335713	0.007684
f_{17}	Best	1.672098	19.57391	6.973861	1.24E+02	2.19E+02	3.14E−04	34.21073	**3.23E−11**	0.957702
	Worst	2.060547	1.20E+02	35.13762	2.89E+02	4.37E+02	0.502529	78.65889	**2.316849**	3.374942
	Average	1.894835	55.74949	14.43146	2.20E+02	3.66E+02	0.02805	56.33423	**0.612219**	1.263067
	SD	0.100801	20.04668	5.508629	38.75242	49.72361	0.077813	9.203493	**0.715699**	0.429663
f_{18}	Best	0.644093	2.36E−07	2.13E+03	2.45E+04	1.88E+04	0.001321	9.19E+02	3.08E−11	**0.00E+00**
	Worst	2.460441	1.01E+04	1.27E+04	6.85E+04	7.08E+04	0.006833	2.02E+03	2.579534	**0.00E+00**
	Average	1.541961	4.04E+02	6.26E+03	5.45E+04	5.55E+04	0.002704	1.44E+03	0.828012	**0.00E+00**
	SD	0.41302	1.99E+03	2.36E+03	1.00E+04	1.19E+04	0.001026	2.69E+02	0.8061879	**0.00E+00**

Table 8 (continued)

Function	Performance measures	Algorithms								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_{19}	Best	− 1.03163	**− 1.03163**	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 1.02926	− 1.03162
	Worst	− 1.03163	**− 1.03163**	− 1.03163	− 0.21546	− 1.03163	− 1.03163	− 1.03163	10.2544	− 1.03161
	Average	− 1.03163	**− 1.03163**	− 1.03163	− 0.96634	− 1.03163	− 1.03163	− 1.03163	0.309509	− 1.03166
	SD	9.23E−07	**2.24E−16**	2.66E−10	0.223668	2.24E−16	9.53E−10	9.56E−11	1.624983	2.20E−06
f_{20}	Best	3.00000	**3.00000**	3.00000	3.00000	3.00000	3.00000	3.00000	2.42E−11	3.00000
	Worst	3.00000	**3.00000**	3.00000	84.00001	3.00000	3.00000	3.00000	7.59629	3.001396
	Average	3.00000	**3.00000**	3.00000	8.940001	3.00000	3.00000	3.00000	0.787694	3.000115
	SD	1.38E−06	**2.00E−15**	1.03E−08	14.72802	3.87E−15	9.23E−09	6.50E−15	1.223225	2.98E−04

results degrade but even then E-MFO manages to perform better than other algorithms.

4.6 Effect of parameter *b* on performance of MFO and E-MFO

In basic MFO algorithm, b parameter defines the shape of spiral. The spiral chosen for this paper is logarithmic spiral whose equation is as follows:

$$r = e^{b\theta} \tag{13}$$

where r and θ are polar coordinates of curve which defines the distance from origin and angle from x-axis, respectively, b is any real positive constant. The rate of change of radius with respect to θ is given as:

$$\frac{\mathrm{d}r}{\mathrm{d}\theta} = b \cdot e^{b\theta} = b \cdot r \tag{14}$$

The angle is calculated between tangent and radial line at point (r, θ) and it is given as:

$$\Phi = \tan^{-1}\left(\frac{r}{\frac{\mathrm{d}r}{\mathrm{d}\theta}}\right) = \tan^{-1}\left(\frac{1}{b}\right) \tag{15}$$

So, from equations, it is observed that b is parameter which defines how tightly and in which direction spiral will spin. There are two extreme cases for the values of b that is for $b = 0$ the spiral is converted into circle and for $b = \infty$ spiral becomes a straight line. The value chosen in the original MFO article for this parameter is equal to 1, and no parametric study for its use has been given. So, in order to evaluate the effect of b, four different values are taken to evaluate the effect of this parameter on algorithm.

Experimental analysis The results are taken at different values of b such as 0.5, 1, 1.5 and 2. The population size and iterations are considered as 30 and 1000, respectively, during experimental analysis. This analysis is conducted on both MFO and E-MFO algorithm. Firstly, in MFO for $b = 0.5$, it achieves better results for eight functions (f_2,f_3,f_5,f_6,f_8,f_9,f_{11} and f_{15}) in terms of best, worst, average and standard deviation. When the results are taken at $b = 1$, four functions, i.e.,f_1, f_4, f_{10} and f_{16} come up with better results, whereas at $b = 1.5$ and 2 only one and two functions, respectively, give better results. There are five functions which are f_7, f_{14}, f_{18}, f_{19} and f_{20} comes up with highly competitive results for all values of b. Further, in case of E-MFO, it is concluded from Table 12 that nine functions offer highly competitive results for all values of b. The six functions $f_1, f_2, f_4, f_8, f_{11}$ and f_{16} are successful in obtaining global optima at $b = 1$ on the other hand for $b = 0.5$, out of 20 functions only for three functions, i.e.,f_9, f_{10} and f_{12} good results are obtained. Moreover, in case of

Table 9 Results of E-MFO compared with others in case of dimension size 50

Function	Performance measures	Algorithms								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_1	Best	**9.82E−51**	3.908591	1.71E+04	6.28E+04	1.05E+05	0.012525	1.80E+03	1.04E−05	0.01809
	Worst	**2.21E−45**	3.00E+04	5.73E+04	1.49E+05	1.44E+05	3.62E−02	5.03E+03	0.011275	0.035205
	Average	**2.08E−46**	6.77E+03	3.28E+04	1.13E+05	1.27E+05	0.021249	3.22E+03	0.001189	0.024277
	SD	**4.78E−46**	7.83E+03	9.33E+03	2.14E+04	1.02E+04	0.005907	7.93E+02	0.002387	0.004232
f_2	Best	**2.79E−33**	20.13997	28.56251	1.00E+10	88.01279	0.35757	29.58089	0.428324	2.09E+04
	Worst	**7.13E−30**	1.40E+02	1.34E+13	8.77E+19	3.60E+02	3.511229	67.71294	6.790059	2.09E+04
	Average	**4.61E−31**	70.54696	4.18E+11	2.58E+18	1.67E+02	0.908903	41.87777	1.903951	2.09E+04
	SD	**1.16E−30**	30.87216	2.14E+12	1.27E+19	63.16034	0.704378	7.817614	1.255481	1.373143
f_3	Best	**2.25E−46**	1.51E+04	4.02E+04	1.63E+05	5.54E+04	6.79E+03	2.29E+03	1.531931	1.25E+04
	Worst	**5.88E−42**	1.25E+05	3.32E+05	7.75E+05	3.05E+05	2.98E+04	7.71E+03	35.02898	1.25E+04
	Average	**4.00E−43**	4.84E+04	1.56E+05	3.72E+05	1.27E+05	1.32E+04	4.59E+03	9.873501	1.25E+04
	SD	**9.96E−43**	2.51E+04	6.56E+04	1.54E+05	5.19E+04	4.12E+03	1.27E+03	6.00081	0.758706
f_4	Best	**8.65E−28**	69.23615	34.85881	81.19559	85.35093	0.286266	14.99628	0.362565	0.695204
	Worst	**2.38E−25**	93.2834	86.54522	96.1857	95.74333	8.790009	27.35923	1.256614	0.634333
	Average	**3.33E−26**	82.31819	56.76027	90.47538	92.60194	2.68246	20.16288	0.845551	0.578
	SD	**4.64E−26**	6.015157	9.872651	3.569025	2.274813	1.774635	2.760949	0.175568	1.03E−04
f_5	Best	**47.54265**	6.52E+04	2.00E+09	2.25E+10	4.08E+10	48.23219	1.76E+07	40.26629	49.96849
	Worst	**48.81203**	9.98E+09	1.39E+10	7.35E+10	7.73E+10	2.05E+05	1.19E+08	1.98E+02	52.23669
	Average	**48.05179**	9.97E+08	5.53E+09	5.65E+10	6.33E+10	1.58E+04	5.41E+07	88.24595	50.9976
	SD	**0.312199**	3.01E+09	2.93E+09	1.06E+10	7.89E+09	4.51E+00	2.46E+07	36.95163	0.56013
f_6	Best	**1.56E−05**	0.247644	4.751723	1.51E+02	2.96E+02	0.022205	0.256331	0.477529	0.079034
	Worst	**3.65E−04**	85.65464	30.74124	3.96E+02	5.26E+02	0.221239	1.099411	8.54467	0.83629
	Average	**1.21E−04**	18.16018	14.02752	2.72E+02	4.04E+02	0.064656	0.625464	1.520455	0.310407
	SD	**8.46E−05**	22.83136	5.432146	60.57708	57.48182	0.03535	0.204166	1.163179	0.194695
f_7	Best	**0.00E+00**	0.00E+00	4.73E−14	0.015626	5.90E−06	7.04E−13	0.00E+00	0.00E+00	1.18E−13
	Worst	**0.00E+00**	0.001125	0.343255	0.475936	0.257498	3.31E−10	3.42E−12	0.00E+00	1.99E−10
	Average	**0.00E+00**	2.89E−05	0.077117	0.287786	0.042654	1.06E−10	4.08E−13	0.00E+00	2.92E−11
	SD	**0.00E+00**	1.65E−04	0.088457	0.125548	0.05665	9.15E−11	7.59E−13	0.00E+00	4.61E−11
f_8	Best	**1.95E−58**	1.65E+05	5.87E+05	6.25E+06	1.02E+07	7.223926	1.65E+05	0.148152	1.21E+02
	Worst	**3.86E−54**	4.00E+06	3.68E+06	1.37E+07	1.39E+07	2.20E+02	4.77E+05	1.88E+02	3.23E+02
	Average	**5.75E−55**	7.42E+05	1.68E+06	1.10E+07	1.23E+07	47.11834	3.13E+05	15.66314	2.46E+02
	SD	**9.82E−55**	9.84E+05	7.20E+05	1.88E+06	1.88E+06	46.6808	6.86E+04	33.4619	51.87049
f_9	Best	**4.10–126**	3.75E+05	9.82E−55	1.01E+09	3.72E+07	1.38E+06	5.96E+05	3.79E−05	4.23E+02
	Worst	**4.6E−113**	3.00E+08	1.82E+09	4.61E+09	1.85E+09	1.44E+07	1.15E+07	4.76E+03	1.97E+03
	Average	**1.0E−114**	4.21E+07	7.62E+08	2.53E+09	5.53E+08	6.14E+06	3.34E+06	2.25E+02	8.98E+02
	SD	**6.0E−114**	5.88E+07	4.20E+08	8.34E+08	4.19E+08	3.08E+06	2.08E+06	7.31E+02	3.33E+02

Table 9 (continued)

Function	Performance measures	Algorithms								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_{10}	Best	**4.56E−61**	1.00E+00	1.63E+03	1.53E+04	8.41E+03	0.227668	3.87E+02	6.84E−26	2.41E−08
	Worst	**1.25E−55**	7.40E+03	8.26E+03	3.36E+04	3.59E+04	37.98557	1.37E+03	6.01E−09	7.92E−06
	Average	**9.00E−57**	2.74E+03	4.04E+03	2.77E+04	2.96E+04	7.492614	7.19E+02	1.20E−10	1.72E−06
	SD	**2.56E−56**	1.94E+03	1.63E+03	3.92E+03	5.49E+03	7.84792	2.02E+02	8.50E−10	1.57E−06
f_{11}	Best	**0.00E+00**	1.10E−13	1.80E−08	0.034787	0.549786	5.59E−08	1.17E−09	4.10E−05	0.358152
	Worst	**0.00E+00**	3.00E−06	1.35E−07	0.783276	1.873582	8.93E−07	4.20E−07	0.22724	0.857122
	Average	**0.00E+00**	7.86E−08	6.16E−08	0.384903	1.09182	3.33E−07	6.07E−08	0.018494	0.628503
	SD	**0.00E+00**	4.31E−07	2.70E−08	0.199591	0.316775	2.22E−07	8.43E−08	0.037369	0.108809
f_{12}	Best	**2.4E−108**	84.99353	13.16207	7.41E+03	3.28E+03	3.98546	60.48988	0.020549	0.12762
	Worst	**1.67E−92**	8.08E+03	1.31E+03	4.59E+04	5.04E+04	31.06265	4.43E+02	10.04294	0.676306
	Average	**3.34E−94**	3.42E+03	2.54E+02	2.83E+04	1.91E+04	14.88969	2.09E+02	0.633531	0.3431752
	SD	**2.36E−93**	2.41E+03	2.87E+02	8.52E+03	1.28E+04	6.463814	88.57409	1.47761	0.133389
f_{13}	Best	**0.00E+00**	2.18E+02	58.71291	5.94E+02	5.95E+02	55.91358	2.27E+02	82.58822	3.583967
	Worst	**0.00E+00**	4.45E+02	3.02E+02	7.56E+02	8.47E+02	1.78E+02	3.23E+02	1.74E+02	23.12976
	Average	**0.00E+00**	3.05E+02	1.46E+02	6.79E+02	7.78E+02	96.22701	2.77E+02	1.18E+02	5.782618
	SD	**0.00E+00**	50.9097	50.342	33.99931	41.85033	23.74788	20.44704	23.4742	3.372481
f_{14}	Best	**8.88E−16**	20	19.99671	21.17698	20.36133	20.00797	21.05664	0.091249	0.084675
	Worst	**20.9473**	20	20.00007	21.3594	20.83655	20.03609	21.28213	2.676562	0.154943
	Average	**13.78734**	20	19.99983	21.28038	20.56707	20.02193	20.02193	1.950735	0.115497
	SD	**9.834276**	2.54E−07	5.12E−04	0.042162	0.086475	0.006991	0.053717	0.444956	0.016937
f_{15}	Best	**0.00E+00**	0.883172	1.45E+02	6.17E+02	9.08E+02	0.00539	18.49829	1.64E−07	6.36E−04
	Worst	**0.00E+00**	3.61E+02	5.80E+02	1.27E+03	1.36E+03	0.020282	39.19273	0.041029	0.001436
	Average	**0.00E+00**	68.30557	3.24E+02	1.04E+03	1.13E+03	0.011161	27.77052	0.001451	0.001086
	SD	**0.00E+00**	79.01692	1.00E+02	1.64E+02	85.53582	0.002371	5.2091	0.006057	1.67E−04
f_{16}	Best	**1.95E−35**	1.43E+02	1.94E+02	1.01E+03	8.62E+02	1.43E+02	1.43E+02	9.055072	0.132012
	Worst	**2.60E−28**	1.13E+03	4.58E+05	6.36E+10	4.13E+05	5.41E+02	7.83E+03	1.08E+02	0.276045
	Average	**6.41E−30**	8.12E+02	1.73E+04	3.11E+09	6.29E+04	3.16E+02	2.12E+03	39.20565	0.209589
	SD	**3.67E−29**	1.40E+02	6.58E+04	1.11E+10	8.22E+04	80.22963	1.16E+03	19.60644	0.035877
f_{17}	Best	3.212196	43.8235	10.08134	1.84E+02	1.35E+02	**5.37E−04**	5.974415	0.902346	3.892741
	Worst	3.825433	1.24E+02	97.85139	4.74E+02	5.26E+02	**2.729747**	2.729747	11.33528	4.408983
	Average	3.45907	70.74052	38.2227	3.56E+02	3.78E+02	**0.384985**	20.72206	4.048971	4.129676
	SD	0.146625	21.06246	17.55698	60.09763	92.99365	**0.503965**	7.788164	2.326723	0.1323023
f_{18}	Best	6.54355	0.18169	1.95E+04	1.95E+04	8.84E+04	0.010868	1.91E+03	1.39E−05	**0.00E+00**
	Worst	7.968962	3.03E+04	6.42E+04	1.41E+05	1.45E+05	0.031909	5326.976	0.013474	**0.00E+00**
	Average	7.202862	5.04E+03	3.24E+04	1.16E+05	1.25E+05	0.022032	3.29E+03	9.88E−04	**0.00E+00**
	SD	0.396199	7.38E+03	9.46E+03	1.59E+04	1.19E+04	0.004567	8.74E+02	0.002344	**0.00E+00**

Table 10 Results of E-MFO compared with others in case of dimension size 100

Function	Performance measures	Algorithms								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_1	Best	**8.73E−53**	5.24E+03	3.89E+04	1.36E+05	2.42E+05	0.130716	6.05E+03	0.378486	0.071222
	Worst	**2.08E−45**	7.91E+04	1.58E+05	2.95E+05	2.98E+05	2.98E+05	1.35E+04	6.665085	0.115094
	Average	**4.29E−47**	3.69E+04	7.41E+04	2.42E+05	2.71E+05	0.19235	9.80E+03	1.6917947	0.0951846
	SD	**2.93E−46**	1.69E+04	2.53E+04	4.47E+04	1.39E+04	0.031247	1.50E+03	1.308203	0.011968
f_2	Best	**6.74E−30**	76.70737	1.54E+02	2.87E+21	2.64E+02	4.601386	76.70737	9.795558	4.18E+04
	Worst	**1.79E−26**	2.89E+02	7.67E+33	2.54E+47	5.54E+50	50.97501	1.09E+02	24.68882	4.18E+04
	Average	**1.85E−27**	1.76E+02	1.53E+32	5.50E+45	1.20E+49	20.94172	90.12284	17.90471	4.18E+04
	SD	**3.94E−27**	54.33873	1.08E+33	3.59E+46	7.83E+49	10.98618	9.384296	4.004437	1.987223
f_3	Best	**1.39E−39**	1.00E+05	2.67E+05	6.75E+05	6.75E+05	5.14E+04	1.23E+04	54.73408	1.25E+04
	Worst	**1.05E−34**	3.30E+05	1.27E+06	3.58E+06	1.14E+06	1.06E+05	3.88E+04	1.76E+02	1.25E+04
	Average	**4.12E−36**	2.02E+05	6.00E+05	1.44E+06	5.17E+05	7.53E+04	2.12E+04	95.36729	1.25E+04
	SD	**1.49E−35**	5.85E+04	2.84E+05	5.34E+05	2.16E+05	1.40E+04	5.72E+03	26.7058	1.219841
f_4	Best	**7.41E−23**	87.53573	45.44762	86.47633	94.40296	94.40296	20.38937	1.378784	0.847582
	Worst	**9.52E−21**	96.59618	81.25122	97.31797	97.93031	33.05645	30.56896	2.222543	1.32837
	Average	**1.71E−21**	93.14778	64.15183	94.97631	96.33653	25.59123	25.05148	1.820295	0.83671
	SD	**1.78E−21**	2.081455	8.183842	2.055199	2.055199	4.133269	2.139599	0.205359	3.03E−05
f_5	Best	**97.83219**	6.56E+08	1.78E+08	3.24E+10	1.15E+11	1.24E+02	9.98E+07	2.97E+02	1.05E+02
	Worst	**98.81081**	2.53E+10	3.80E+10	1.65E+11	1.73E+11	1.85E+05	5.01E+08	1.31E+03	1.10E+02
	Average	**98.43629**	8.06E+09	1.24E+10	1.19E+11	1.47E+11	1.61E+04	2.74E+08	5.92E+02	1.07E+02
	SD	**0.327202**	6.03E+09	7.40E+09	3.84E+10	1.06E+10	3.62E+04	8.31E+07	1.78E+02	1.351756
f_6	Best	**4.04E−06**	16.41162	15.0506	3.65E+02	1.51E+03	0.089564	1.744613	14.73571	0.168376
	Worst	**6.10E−04**	4.54E+02	60.23618	1.93E+03	2.30E+03	0.293075	8.760175	1.31E+02	0.837434
	Average	**1.36E−04**	1.49E+02	30.2305	1.31E+03	1.91E+03	0.193614	3.8938	41.253046	0.4063992
	SD	**1.36E−04**	1.04E+02	9.421689	4.30E+02	1.64E+02	0.044121	1.365825	26.46412	0.182236
f_7	Best	**0.00E+00**	0.00E+00	2.71E−14	2.03E−11	1.15E−10	2.42E−13	0.00E+00	0.00E+00	4.21E−15
	Worst	**0.00E+00**	0.00E+00	0.345929	0.48589	0.269149	3.83E−10	2.04E−12	0.00E+00	1.02E−10
	Average	**0.00E+00**	0.00E+00	0.091339	0.280384	0.045804	9.44E−11	1.69E−13	0.00E+00	1.33E−11
	SD	**0.00E+00**	0.00E+00	0.086893	0.137033	0.055261	7.93E−11	4.08E−13	0.00E+00	2.28E−11
f_8	Best	**3.36E−49**	2.69E+05	1.40E+06	1.19E+07	2.13E+07	5.81E+02	6.13E+05	1.63E+03	6.71E+02
	Worst	**1.63E−43**	5.95E+06	1.30E+07	2.84E+07	2.84E+07	1.20E+03	1.34E+06	3.48E+04	1.31E+03
	Average	**1.06E−44**	3.06E+06	5.16E+06	2.39E+07	2.65E+07	8.14E+02	9.55E+05	1.27E+04	9.11E+02
	SD	**2.64E−44**	1.33E+06	2.26E+06	4.19E+06	1.62E+06	1.54E+02	1.78E+05	8.38E+03	1.65E+02
f_9	Best	**4.45E−123**	1.07E+06	8.90E+07	6.10E+08	4.19E+07	1.27E+06	5.21E+05	0.001591	1.71E+03
	Worst	**1.13E−112**	4.79E+08	1.69E+09	3.79E+09	2.11E+09	1.42E+07	1.03E+07	1.03E+03	4.39E+03
	Average	**2.52E−114**	4.51E+07	6.68E+08	2.24E+09	7.01E+08	6.21E+06	2.83E+06	1.01E+02	3.02E+03
	SD	**1.59E−113**	7.34E+07	3.50E+08	7.36E+08	5.31E+08	3.03E+06	1.79E+06	1.71E+02	7.28E+02

Table 10 (continued)

Function	Performance measures	Algorithms								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_{10}	Best	**0.00E+00**	9.91E−07	2.11E−08	0.013344	0.630326	3.62E−08	1.36E−09	1.61E−05	1.52E−07
	Worst	**0.00E+00**	0.002301	3.59E−08	0.300021	0.397031	3.95E−07	6.40E−07	1.24E+13	6.00E−06
	Average	**0.00E+00**	8.21E−04	6.33E−08	0.569071	1.498911	4.88E−07	1.74E−07	3.25E+11	1.98E−06
	SD	**0.00E+00**	0.002301	3.59E−08	0.300021	0.397031	3.95E−07	6.40E−07	1.82E+12	1.69E−06
f_{11}	Best	**1.18E−51**	4.44E+03	1.18E+04	6.65E+04	1.20E+05	19.98107	2.86E+03	7.981126	3.214133
	Worst	**1.47E−45**	4.62E+04	4.45E+04	1.48E+05	1.48E+05	2.35E+02	6.76E+03	2.41E+02	5.708358
	Average	**7.73E−47**	1.75E+04	2.26E+04	1.14E+05	1.34E+05	87.81589	4.46E+03	64.29493	4.302195
	SD	**2.39E−46**	8.26E+03	7.57E+03	2.21E+04	6.84E+03	49.89147	8.10E+02	46.5313	5.708358
f_{12}	Best	**2.03E−51**	1.31E+03	99.34773	3.66E+04	6.23E+04	32.68077	6.16E+02	12.522572	0.927053
	Worst	**4.12E−45**	3.00E+04	4.80E+03	1.25E+05	1.34E+05	1.38E+02	1.69E+03	1.65E+02	2.1165021
	Average	**1.55E−46**	1.23E+04	1.74E+03	8.45E+04	1.02E+05	83.28036	9.65E+02	51.255784	1.3459801
	SD	**6.10E−46**	6.64E+03	1.17E+03	2.02E+04	1.74E+04	25.51051	2.28E+02	31.53378	0.281337
f_{13}	Best	**0.00E+00**	5.43E+02	6.10E−46	1.23E+03	1.49E+03	1.93E+02	6.45E+02	1.60E+02	16.0387
	Worst	**0.00E+00**	9.39E+02	4.47E+02	1.58E+03	1.74E+03	4.24E+02	7.52E+02	3.71E+02	23.72978
	Average	**0.00E+00**	7.52E+02	2.65E+02	1.47E+03	1.64E+03	2.90E+02	6.93E+02	2.69E+02	19.656806
	SD	**0.00E+00**	86.87307	58.4988	88.92216	58.77398	51.20952	26.56881	43.43726	2.052437
f_{14}	Best	**4.44E−15**	20	19.9904	21.35306	20.68334	20.06513	21.32538	2.488935	0.13476
	Worst	**21.42653**	21.42653	20.00016	21.48368	21.0389	20.15414	21.42653	4.225948	0.2089347
	Average	**18.90539**	20.00001	19.9995	21.41729	20.85937	20.10049	21.38318	3.193036	0.167698
	SD	**6.366135**	3.78E−05	0.002006	0.030592	0.095717	0.019191	0.022966	0.371464	0.016806
f_{15}	Best	**0.00E+00**	62.48609	3.33E+02	1.19E+03	2.10E+03	0.026208	69.37639	0.007909	16.0387
	Worst	**0.00E+00**	6.01E+02	1.60E+03	2.66E+03	2.71E+03	0.079403	1.14E+02	0.135635	23.72978
	Average	**0.00E+00**	2.77E+02	7.16E+02	2.21E+03	2.47E+03	0.041701	91.97533	0.031285	19.6568
	SD	**0.00E+00**	12.73271	2.51E+02	4.00E+02	1.29E+02	0.010145	12.73271	0.024996	2.052437
f_{16}	Best	**2.36E−26**	4.31E+02	59.82885	4.46E+02	6.51E+02	2.06E+02	2.18E+02	1.21E+02	0.556069
	Worst	**6.42E−21**	1.04E+03	3.87E+04	1.42E+07	5.38E+05	3.82E+02	2.07E+03	3.30E+02	1.194154
	Average	**3.38E−22**	7.53E+02	1.68E+03	3.08E+05	3.89E+04	2.88E+02	7.47E+02	1.99E+02	0.777557
	SD	**1.53E+02**	1.53E+02	5.53E+03	2.01E+06	7.79E+04	33.10453	4.32E+02	52.68694	0.153958
f_{17}	Best	**7.546086**	1.42E+02	53.14262	4.50E+02	8.24E+02	2.05E+00	33.70335	2.942215	8.730477
	Worst	**8.420767**	4.19E+02	2.14E+02	1.00E+03	1.10E+03	18.36023	97.69666	15.74558	9.049739
	Average	**7.973306**	2.13E+02	94.81253	7.91E+02	9.63E+02	8.602651	61.72317	8.868537	8.933344
	SD	**0.213834**	53.5671	30.86755	1.43E+02	6.12E+01	4.377323	14.48923	2.664143	0.070157
f_{18}	Best	17.32299	4.00E+03	3.49E+04	1.17E+05	2.13E+05	0.137464	6.87E+03	0.182169	**0.00E+00**
	Worst	19.65245	5.32E+04	1.32E+05	2.97E+05	2.95E+05	0.259834	1.42E+04	6.52165	**0.00E+00**
	Average	18.74351	2.84E+04	7.19E+04	2.50E+05	2.71E+05	0.190151	1.01E+04	1.999252	**0.00E+00**
	SD	0.493934	1.20E+04	2.13E+04	4.60E+04	1.51E+04	0.026608	1.80E+03	1.313208	**0.00E+00**

Table 11 Results of E-MFO compared with other algorithms in case of dimension size 200

Function	Performance measures	Algorithms								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_1	Best	**1.45E−47**	1.37E+05	8.82E+04	2.63E+05	5.36E+05	0.763145	2.04E+04	15.85378	0.202338
	Worst	**1.73E−41**	2.71E+05	2.63E+05	6.04E+05	6.19E+05	1.97E+00	3.28E+04	44.656714	0.303891
	Average	**6.28E−43**	1.83E+05	1.50E+05	4.88E+05	5.79E+05	1.10E+00	2.48E+04	30.14362	0.252025
	SD	**3.01E+04**	3.01E+04	3.84E+04	9.79E+04	2.08E+04	0.235442	2.74E+03	7.838145	0.0276851
f_2	Best	**1.76E−29**	4.55E+02	1.99E+03	9.19E+46	1.81E+31	1.23E+02	1.64E+02	51.98113	8.37E+04
	Worst	**1.71E−24**	6.97E+02	9.03E+62	1.70E+101	2.17E+104	3.12E+02	2.32E+02	93.89323	8.38E+04
	Average	**1.45E−25**	5.49E+02	1.81E+61	9.26E+99	1.82E+103	1.96E+02	2.00E+02	71.24135	8.37E+04
	SD	**3.33E−25**	53.00042	1.28E+62	3.64E+100	5.18E+103	46.32443	16.43607	9.368079	1.999194
f_3	Best	**1.46E−35**	4.81E+05	7.11E+05	2.25E+06	7.77E+05	2.50E+05	4.92E+04	2.69E+02	1.25E+04
	Worst	**2.71E−30**	1.09E+06	5.26E+06	1.15E+07	4.08E+06	4.50E+05	1.37E+05	8.39E+02	1.25E+04
	Average	**8.85E−32**	7.12E+05	2.75E+06	5.92E+06	2.21E+06	3.48E+05	8.75E+04	4.69E+02	1.25E+04
	SD	**3.93E−31**	1.51E+05	1.07E+06	1.92E+06	8.10E+05	4.36E+04	1.86E+04	1.43E+02	0.656492
f_4	Best	**4.07E−21**	93.08323	54.03602	78.84859	96.04051	42.07106	23.74176	2.090823	0.528204
	Worst	**7.38E−10**	98.52615	79.41252	98.34887	98.97741	55.97332	36.10634	2.900365	0.526733
	Average	**2.82E−11**	96.96512	68.64042	96.44807	98.15073	48.4488	29.24717	2.515605	0.52879
	SD	**1.14E−10**	1.063848	6.496662	3.068331	0.570873	3.387015	2.63306	0.19626	5.23E−04
f_5	Best	**1.98E+02**	4.44E+10	1.20E+10	6.42E+10	2.76E+11	2.25E+03	5.47E+08	3.11E+03	2.19E+02
	Worst	**1.99E+02**	1.25E+11	9.91E+10	3.69E+11	3.49E+11	1.44E+06	1.32E+09	1.03E+04	2.29E+02
	Average	**1.99E+02**	7.91E+10	3.31E+10	2.53E+11	3.26E+11	1.44E+05	8.49E+08	6.30E+03	2.24E+02
	SD	**0.223952**	1.86E+10	1.73E+10	9.45E+10	1.56E+10	2.84E+05	1.97E+08	1.64E+03	2.546328
f_6	Best	**1.98E−05**	1.08E+03	38.92616	1.59E+03	7.09E+03	1.562429	9.256986	6.19E+02	0.283232
	Worst	**3.19E−04**	2.90E+03	1.17E+02	8.92E+03	9.14E+03	5.726941	36.29072	6.68E+03	1.009639
	Average	**1.15E−04**	1.86E+03	67.6422	6.65E+03	8.43E+03	3.016414	21.15751	1.81E+03	0.588733
	SD	**8.62E−05**	4.51E+02	20.08162	1.82E+03	5.41E+02	0.91736	5.207852	9.42E+02	0.1974908
f_7	Best	**0.00E+00**	0.00E+00	3.13E−13	0.00817	0.00E+00	1.16E−13	0.00E+00	0.00E+00	1.15E−14
	Worst	**0.00E+00**	0.003127	0.341554	0.473123	0.249703	4.15E−10	5.66E−11	0.00E+00	1.97E−10
	Average	**0.00E+00**	6.46E−05	0.101385	0.293907	0.061709	9.23E−11	1.38E−12	0.00E+00	1.98E−11
	SD	**0.00E+00**	4.42E−04	0.101006	0.129029	0.060307	9.15E−11	7.99E−12	0.00E+00	3.89E−11
f_8	Best	**9.95E−45**	1.26E+07	5.56E+06	2.60E+07	5.17E+07	4.33E+03	1.92E+06	1.37E+05	2.11E+03
	Worst	**6.48E−38**	2.28E+07	1.93E+07	6.11E+07	6.11E+07	1.05E+04	2.99E+06	5.45E+05	3.03E+03
	Average	**2.54E−39**	1.78E+07	1.11E+07	4.57E+07	5.76E+07	6.46E+03	2.45E+06	2.89E+05	2.63E+03
	SD	**1.12E−38**	2.60E+06	2.95E+06	1.13E+07	2.22E+06	1.28E+03	2.70E+05	8.11E+04	2.47E+02
f_9	Best	**6.8E−124**	2.47E+06	1.80E+08	6.70E+08	2.25E+07	1.53E+06	1.02E+06	1.912979	6.25E+03
	Worst	**1.7E−108**	2.45E+08	2.64E+09	3.83E+09	2.90E+09	1.62E+07	9.69E+06	4.16E+03	1.14E+04
	Average	**3.5E−110**	5.32E+07	8.23E+08	2.30E+09	9.50E+08	5.66E+06	3.34E+06	4.54E+02	8.42E+03
	SD	**2.4E−109**	5.67E+07	5.10E+08	8.01E+08	7.15E+08	2.89E+06	1.84E+06	6.95E+02	1.25E+03

Table 11 (continued)

Function	Performance measures	Algorithms								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_{10}	Best	**0.00E+00**	1.43E−04	1.70E−08	0.080006	1.059435	4.49E−08	7.24E−09	3.05E+21	1.46E−07
	Worst	**0.00E+00**	0.994845	1.38E−07	1.845873	2.630292	2.35E−06	2.73E−05	3.89E+49	6.09E−06
	Average	**0.00E+00**	0.059233	5.70E−08	0.770675	1.85E+00	7.90E−07	9.47E−07	7.81E+47	1.54E−06
	SD	**0.00E+00**	0.174417	2.94E−08	0.43964	0.419724	5.14E−07	3.96E−06	5.50E+48	1.32E−06
f_{11}	Best	**1.34E−46**	1.04E+05	4.51E+04	2.44E+05	4.90E+05	7.65E+02	1.67E+04	9.90E+02	18.70683
	Worst	**7.16E−40**	2.30E+05	2.16E+05	5.93E+05	6.21E+05	3.17E+03	2.97E+04	5.64E+03	29.81683
	Average	**4.23E−41**	1.63E+05	1.08E+05	5.15E+05	5.68E+05	1.58E+03	2.32E+04	2.31E+03	24.35727
	SD	**1.26E−40**	2.66E+04	3.54E+04	9.43E+04	2.89E+04	5.40E+02	3.37E+03	8.05E+02	2.529588
f_{12}	Best	**7.21E−47**	2.33E+04	1.70E+03	6.38E+04	1.93E+05	1.58E+02	2.04E+03	3.24E+02	2.7405821
	Worst	**5.28E−41**	6.19E+04	2.21E+04	2.77E+05	3.19E+05	5.08E+02	3.80E+03	1.61E+03	5.6576739
	Average	**4.35E−42**	3.93E+04	8.02E+03	2.05E+05	2.55E+05	3.31E+02	2.89E+03	7.07E+02	4.207254
	SD	**1.08E−41**	7.77E+03	4.25E+03	5.66E+04	2.40E+04	87.21924	3.83E+02	2.51E+02	0.749036
f_{13}	Best	**0.00E+00**	1.74E+03	3.15E+02	2.48E+03	3.21E+03	6.24E+02	1.51E+03	5.88E+02	40.65584
	Worst	**0.00E+00**	2.15E+03	9.29E+02	3.37E+03	3.53E+03	9.51E+02	1.68E+03	9.07E+02	59.10264
	Average	**0.00E+00**	1.95E+03	5.00E+02	3.13E+03	3.41E+03	7.79E+02	1.60E+03	7.09E+02	49.50283
	SD	**0.00E+00**	89.58739	1.45E+02	2.07E+02	69.91233	81.36572	40.86178	74.7027	4.697452
f_{14}	Best	21.10011	20	19.99425	21.46023	20.89783	20.16807	21.43279	3.683187	**0.164595**
	Worst	21.29661	20.09454	20.00086	21.55383	21.24759	20.28109	21.5232	4.9168	**0.23552**
	Average	21.20242	20.01554	20.00017	21.50665	21.07666	20.22424	21.49185	4.167273	**0.210279**
	SD	0.044325	0.023989	0.001098	0.020649	0.077417	0.026702	0.018635	0.211929	**0.014902**
f_{15}	Best	**0.00E+00**	1.27E+03	6.86E+02	1.81E+03	4.51E+03	0.14324	1.54E+02	0.078398	0.002081
	Worst	**0.00E+00**	2.16E+03	2.33E+03	5.56E+03	5.56E+03	0.306529	3.25E+02	0.31181	0.003802
	Average	**0.00E+00**	1.66E+03	1.41E+03	4.61E+03	5.16E+03	0.207294	2.18E+02	0.201604	0.002868
	SD	**0.00E+00**	2.13E+02	3.16E+02	1.07E+03	2.15E+02	0.031399	28.88742	0.057798	3.68E−04
f_{16}	Best	**5.90E−19**	5.63E+03	1.77E+04	6.40E+04	7.47E+04	3.07E+03	2.08E+04	3.54E+02	1.707246
	Worst	**1.31E−08**	7.16E+03	2.14E+08	4.17E+15	7.37E+07	6.30E+03	2.92E+05	8.56E+02	3.760735
	Average	**3.54E−10**	6.22E+03	5.74E+06	3.22E+14	9.55E+06	4.57E+03	8.67E+04	5.37E+02	2.471902
	SD	**1.85E−09**	3.78E+02	3.09E+07	8.44E+14	1.26E+07	6.68E+02	5.87E+04	1.33E+02	0.5280802
f_{17}	Best	**16.37915**	5.48E+02	1.28E+02	7.78E+02	1.87E+03	30.04823	1.05E+02	14.96406	17.95556
	Worst	**17.54155**	1.01E+03	5.63E+02	2.29E+03	2.32E+03	1.26E+02	1.96E+02	40.57207	18.21754
	Average	**17.05695**	7.60E+02	3.03E+02	1.83E+03	2.11E+03	69.30709	1.52E+02	23.85825	18.16062
	SD	**0.222976**	1.02E+02	93.40166	3.64E+02	1.08E+02	19.67717	22.08205	6.145443	0.045842
f_{18}	Best	41.34965	1.21E+05	9.21E+04	2.99E+05	5.53E+05	0.790574	1.90E+04	12.26393	**0.00E+00**
	Worst	44.31655	2.25E+05	2.41E+05	6.13E+05	6.20E+05	2.35185	2.98E+04	52.34478	**0.00E+00**
	Average	42.93578	1.61E+05	1.46E+05	5.24E+05	5.85E+05	1.120659	2.45E+04	33.14138	**0.00E+00**
	SD	0.711676	2.32+04	3.39E+04	8.90E+04	1.93E+04	0.250298	2.77E+03	8.971067	**0.00E+00**

Table 12 Results of E-MFO compared with others in case of dimension size 500

Function	Performance measures	Algorithms								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_1	Best	**3.20E−45**	8.62E+05	2.07E+05	7.65E+05	1.44E+06	3.13E+04	6.04E+04	1.58E+02	0.700708
	Worst	**6.53E−38**	1.03E+06	7.80E+05	1.59E+06	1.59E+06	8.22E+04	8.35E+04	3.23E+02	0.891416
	Average	**3.54E−39**	9.50E+05	3.82E+05	1.33E+06	1.53E+06	4.82E+04	6.95E+04	2.38E+02	0.805819
	SD	**1.15E−38**	3.53E+04	1.08E+05	2.68E+05	3.27E+04	1.07E+04	6.56E+03	38.06859	0.06021
f_2	Best	**3.74E−28**	2.01E+03	3.54E+08	9.20E+140	2.0E+148	9.44E+02	4.36E+02	2.46E+02	2.09E+05
	Worst	**7.29E−24**	2.48E+03	1.10E+208	5.60E+268	3.7E+270	1.45E+03	5.86E+02	3.63E+02	2.09E+05
	Average	**6.64E−25**	2.25E+03	2.20E+206	1.12E+267	1.4E+269	1.16E+03	5.13E+02	2.95E+02	2.09E+05
	SD	**3.73E−28**	97.77479	Inf	Inf	Inf	89.70169	34.11001	23.57923	3.949205
f_3	Best	**6.99E−33**	2.30E+06	7.58E+06	1.76E+07	7.87E+06	1.27E+06	6.62E+05	1.47E+03	1.25E+04
	Worst	**5.13E−25**	5.50E+06	4.22E+07	7.75E+07	5.29E+07	2.36E+06	2.91E+06	8.27E+03	1.25E+04
	Average	**1.05E−26**	3.66E+06	1.85E+07	3.62E+07	2.24E+07	1.81E+06	1.58E+06	3.70E+03	1.25E+04
	SD	**7.26E−26**	8.38E+05	7.54E+06	1.23E+07	9.73E+06	2.83E+05	4.32E+05	1.64E+03	0.829675
f_4	Best	**3.62E−18**	98.22465	6.30E+01	8.97E+01	9.87E+01	5.56E+01	39.6108	2.810859	0.528204
	Worst	**99.2946**	99.4224	9.03E+01	9.94E+01	9.96E+01	9.17E+01	6.61E+01	3.483475	0.529333
	Average	**37.55688**	98.95586	7.49E+01	9.80E+01	9.93E+01	6.56E+01	4.83E+01	3.102431	0.52879
	SD	**48.45998**	0.289448	6.40E+00	1.85E+00	2.00E−01	1.05E+01	5.29E+00	0.1720504	3.03E−04
f_5	Best	**4.98E+02**	4.34E+11	3.07E+10	2.02E+11	7.76E+11	1.11E+09	1.48E+09	5.17E+04	5.64E+02
	Worst	**4.99E+02**	5.59E+11	1.97E+11	9.28E+11	9.27E+11	7.73E+09	4.32E+09	1.19E+05	5.87E+02
	Average	**4.99E+02**	4.99E+11	8.51E+10	7.20E+11	8.76E+11	3.98E+09	2.85E+09	7.53E+04	5.77E+02
	SD	**0.106515**	2.83E+10	3.88E+10	2.42E+11	3.16E+10	1.44E+09	5.82E+08	1.31E+04	5.880869
f_6	Best	**3.53E−06**	2.50E+04	2.00E+02	1.59E+02	5.20E+04	4.24E+02	2.21E+02	4.14E+04	0.9388711
	Worst	**7.17E−04**	3.75E+04	3.11E+03	6.09E+04	6.34E+04	9.73E+02	6.83E+02	1.46E+05	2.048431
	Average	**2.06E−04**	3.06E+04	9.46E+02	7.07E+04	5.78E+04	7.00E+02	3.71E+02	8.92E+04	1.435729
	SD	**1.65E−04**	2.40E+03	5.95E+02	1.21E+04	2.44E+03	1.27E+02	1.08E+02	2.45E+04	0.254926
f_7	Best	**0.00E+00**	0.00E+00	5.13E−14	0.043544	0.00E+00	3.27E−12	0.00E+00	0.00E+00	0.00E+00
	Worst	**0.00E+00**	0.002964	0.342796	0.481134	0.355643	4.96E−10	6.13E−12	0.00E+00	1.19E−10
	Average	**0.00E+00**	5.93E−05	0.081967	0.305825	0.059003	7.87E−11	3.07E−13	0.00E+00	2.00E−11
	SD	**0.00E+00**	4.19E−04	0.094435	0.120609	0.074542	9.22E−11	9.32E−13	0.00E+00	2.46E−11
f_8	Best	**4.03E−43**	8.89E+07	1.80E+07	7.27E+07	1.45E+08	2.95E+06	5.86E+06	1.61E+06	7.12E+03
	Worst	**8.00E−34**	1.06E+08	6.24E+07	1.59E+08	1.58E+08	6.93E+06	8.71E+06	3.15E+06	1.00E+04
	Average	**4.72E−35**	9.59E+07	3.33E+07	1.37E+08	1.54E+08	4.80E+06	6.91E+06	2.47E+06	8.14E+03
	SD	**1.52E−34**	3.89E+06	1.04E+07	2.59E+07	2.83E+06	9.82E+05	6.09E+05	3.65E+05	6.17E+02
f_9	Best	**4.70E−126**	7.24E+05	1.73E+08	1.33E+09	9.27E+06	1.05E+06	6.82E+05	5.621493	2.67E+04
	Worst	**1.56E−111**	3.67E+08	1.86E+09	3.89E+09	3.59E+09	1.22E+07	7.11E+06	2.17E+03	4.67E+04
	Average	**4.52E−113**	4.41E+07	8.07E+08	2.53E+09	1.26E+09	5.58E+06	2.77E+06	5.59E+02	3.43E+04
	SD	**2.38E−112**	6.58E+07	4.72E+08	6.65E+08	1.03E+09	2.56E+06	1.48E+06	5.03E+02	4.80E+03

Table 12 (continued)

Function	Performance measures	Algorithms								
		E-MFO	MFO	BA	BFP	DE	FA	FPA	PSO	GA
f_{10}	Best	**0.00E+00**	0.038239	1.24E−08	0.121382	1.525914	6.10E−07	5.43E−09	2.3E+103	6.02E−08
	Worst	**2.373591**	1.363739	1.67E−07	1.916926	3.777572	0.94647	6.24E−06	6.4E+183	1.22E−05
	Average	**0.271064**	0.336057	6.22E−08	0.967956	2.575341	0.106643	7.76E−07	1.2E+182	2.33E−06
	SD	**0.602188**	0.290076	3.99E−08	0.533964	0.529162	0.159758	1.21E−06	Inf	2.01E−06
f_{11}	Best	**1.57E−42**	1.93E+06	3.37E+05	1.12E+06	3.56E+06	9.43E+04	1.37E+05	2.42E+04	1.66E+02
	Worst	**2.59E−34**	2.48E+06	1.24E+06	3.93E+06	3.99E+06	1.77E+05	2.33E+05	7.10E+04	2.24E+02
	Average	**6.28E−36**	2.19E+06	8.14E+05	3.31E+06	3.80E+06	1.31E+05	1.73E+05	5.09E+04	1.90E+02
	SD	**3.67E−35**	1.17E+05	2.02E+05	6.95E+05	9.44E+04	2.19E+04	1.93E+04	9.53E+03	15.07457
f_{12}	Best	**7.11E−44**	1.61E+05	1.05E+04	1.41E+05	6.33E+05	3.12E+03	6.67E+03	5.17E+03	12.22409
	Worst	**4.45E−36**	2.85E+05	1.19E+05	7.78E+05	8.38E+05	9.48E+03	1.13E+04	2.11E+04	19.81351
	Average	**1.01E−37**	1.95E+05	3.83E+04	5.59E+05	7.48E+05	5.81E+03	9.05E+03	1.00E+04	16.39981
	SD	**6.29E−37**	2.26E+04	2.19E+04	2.09E+05	4.59E+04	1.29E+03	1.12E+03	2.91E+03	1.933193
f_{13}	Best	**0.00E+00**	6.03E+03	1.14E+03	6.62E+03	8.49E+03	2.78E+03	4.42E+03	2.27E+03	0.221598
	Worst	**0.00E+00**	6.80E+03	2.49E+03	8.65E+03	8.98E+03	3.37E+03	4.78E+03	3.50E+03	0.263261
	Average	**0.00E+00**	6.47E+03	1.65E+03	8.14E+03	8.78E+03	3.09E+03	4.58E+03	2.70E+03	0.240703
	SD	**0.00E+00**	1.68E+02	2.84E+02	6.29E+02	1.16E+02	1.47E+02	77.79587	2.64E+02	0.009468
f_{14}	Best	21.25429	20	20.11317	21.54223	21.24674	20.48476	21.55964	4.477382	**0.009468**
	Worst	21.37776	21.59477	20.25831	21.61217	21.38999	20.62969	21.59476	5.319343	**0.263261**
	Average	21.32418	20.13839	20.18865	21.58754	21.32366	20.54637	21.58079	4.898799	**0.240703**
	SD	0.027639	0.151432	0.038652	0.013446	0.034576	0.029272	0.008826	0.174587	**0.009468**
f_{15}	Best	**0.00E+00**	7.69E+03	2.03E+03	6.06E+03	1.32E+04	2.33E+02	5.19E+02	0.473706	1.46E+02
	Worst	**0.00E+00**	9.56E+03	6.78E+03	1.43E+04	1.42E+04	6.16E+02	8.14E+02	0.786017	1.86E+02
	Average	**0.00E+00**	8.67E+03	3.82E+03	1.27E+04	1.38E+04	4.32E+02	6.40E+02	0.617803	1.60E+02
	SD	**0.00E+00**	3.79E+02	9.95E+02	2.24E+03	2.31E+02	87.71168	63.67595	0.066913	10.01437
f_{16}	Best	**7.28E−09**	1.55E+04	3.00E+04	1.03E+10	5.50E+06	1.25E+04	1.50E+05	1.02E+03	6.350059
	Worst	**0.003233**	7.31E+02	1.22E+08	1.95E+18	6.34E+08	1.66E+03	1.27E+06	3.44E+03	22.09129
	Average	**0.001177**	1.75E+04	3.19E+07	1.15E+18	4.42E+08	1.66E+04	1.22E+06	1.75E+03	12.09336
	SD	**0.003233**	7.31E+02	1.22E+08	1.95E+18	6.34E+08	1.66E+03	1.27E+06	4.77E+02	3.563376
f_{17}	Best	**43.87916**	3.10E+03	6.74E+02	2.52E+03	5.40E+03	3.51E+02	3.45E+02	71.06688	45.47434
	Worst	**44.92273**	3.89E+03	2.05E+03	5.81E+03	6.11E+03	6.76E+02	5.82E+02	1.27E+02	45.61268
	Average	**44.46996**	3.47E+03	1.11E+03	4.89E+03	5.76E+03	5.21E+02	4.62E+02	93.78572	45.544987
	SD	**0.221558**	1.80E+02	3.05E+02	9.82E+02	1.55E+02	79.57762	49.42061	14.68623	0.030731
f_{18}	Best	1.15E+02	8.32E+05	2.16E+05	6.31E+05	1.45E+06	2.72E+04	5.66E+04	1.71E+02	**0.00E+00**
	Worst	1.18E+02	9.76E+05	7.37E+05	1.58+06	1.58E+06	7.78E+04	1.01E+05	3.58E+02	**0.00E+00**
	Average	1.17E+02	9.14E+05	3.87E+05	1.33E+06	1.53E+06	4.91E+04	7.14E+04	2.72E+02	**0.00E+00**
	SD	0.757443	3.69E+04	9.98E+04	2.58E+05	3.18E+04	1.01E+04	8.42E+03	48.38942	**0.00E+00**

Table 13 Experimental analysis for different values of b

Function	Performance measures	$b = 0.5$		$b = 1$		$b = 1.5$		$b = 2$	
		MFO	E-MFO	MFO	E-MFO	MFO	E-MFO	MFO	E-MFO
f_1	Best	1.01E−06	3.79E−72	2.79E−08	1.74E−73	2.94E−04	9.09E−65	0.011986	1.21E−52
	Worst	1.00E+04	1.42E−67	26.2144	1.28E−65	2.00E+04	2.48E−60	3.00E+04	6.57E−47
	Average	1.04E+03	9.63E−69	3.67867	5.20E−67	3.40E+03	2.48E−60	2.80E+03	5.24E−48
	SD	3.51E+03	2.39E−68	9.1884	2.41E−66	6.58E+03	8.59E−60	6.40E+03	1.39E−47
f_2	Best	4.14E−23	2.42E−52	4.25E−15	5.09E−60	1.71E−17	1.01E−59	7.76E−15	9.54E−57
	Worst	10.0001	1.16E−49	20	1.49E−56	10.0001	1.14E−56	20	1.19E−53
	Average	0.45435	9.09E−51	3.46868	3.89E−57	0.46886	9.21E−58	2.88768	7.48E−55
	SD	1.414213	1.75E−50	5.9281411	1.49E−56	1.9794866	1.72E−57	4.9651849	1.78E−54
f_3	Best	2.37E−11	2.71E−84	1.24E−06	1.81E−98	1.84E−09	3.81E−100	3.80E−07	2.44E−94
	Worst	5.00E+03	2.85E−78	2.25E+03	2.22E−92	10000	1.04E−92	6.67E+03	1.62E−87
	Average	2.00E+02	7.91E−80	1.10E+03	1.81E−98	7.33E+02	3.89E−94	1.13E+03	3.46E−89
	SD	9.90E+02	4.04E−79	2.25E+03	3.85E−93	2.11E+03	1.62E−93	2.17E+03	2.28E−88
f_4	Best	0.003113	1.76E−49	3.36E−07	1.57E−55	6.07E−07	5.92E−55	4.15E−06	2.65E−53
	Worst	33.232336	2.66E−46	6.3989103	6.05E−52	0.0011902	4.12E−52	0.0029319	7.73E−49
	Average	8.8273498	2.11E−47	2.62E+00	1.57E−55	8.45E−05	8.45E−05	4.12E−04	6.21E−50
	SD	10.48673	4.48E−47	6.3989103	8.78E−53	1.82E−04	6.11E−53	6.82E−04	1.40E−49
f_5	Best	27.25042	27.41142	31.07914	27.43202	53.93414	27.18364	2.16E+02	27.30733
	Worst	1.00E+06	28.840794	1.00E+06	28.859687	9.98E+09	28.636438	9.98E+09	0.2508971
	Average	1.81E+05	28.141834	2.62E+05	28.186317	2.00E+08	27.915359	2.16E+02	27.914274
	SD	3.87E+05	0.2231016	4.42E+05	0.3011082	1.41E+09	0.2451642	2.16E+02	0.2508971
f_6	Best	4.42E−04	9.29E−06	7.44E−04	6.39891	5.96E−04	4.26E−06	9.14E−04	2.67E−06
	Worst	0.0211365	3.69E−04	0.0266772	2.19E−04	0.0158486	3.35E−04	0.0075774	5.07E−04
	Average	0.0056088	8.26E−05	7.44E−04	7.45E−05	0.0061108	7.76E−05	0.004491	1.23E−04
	SD	0.0035787	7.32E−05	0.0050839	5.86E−05	0.0061108	6.90E−05	0.0203794	9.77E−05
f_7	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Worst	0.0123332	0.00E+00	0.0126562	0.00E+00	0.0040419	0.00E+00	0.0036138	0.00E+00
	Average	0.0211365	0.00E+00	4.76E−04	0.00E+00	1.29E−04	0.00E+00	7.23E−05	0.00E+00
	SD	0.0021633	0.00E+00	2.06E−03	0.00E+00	6.20E−04	0.00E+00	5.11E−04	0.00E+00
f_8	Best	1.27E−04	1.46E−67	0.001861	2.13E−70	0.131189	5.08E−60	1.43994	2.68E−48
	Worst	1.00E+06	1.44E−63	1.00E+06	6.09E−63	2.00E+06	5.08E−60	2.00E+06	2.50E−42
	Average	1.20E+05	1.31E−64	1.40E+05	2.33E−64	0.1311886	1.53E−56	1.80E+05	7.13E−44
	SD	3.28E+05	3.07E−64	3.51E+05	8.88E−64	5.44E+05	4.78E−56	4.38E+05	3.55E−43
f_9	Best	2.33E+05	9.54E−134	9.80E+05	1.14E−125	2.69E+06	1.83E−101	3.48E+06	1.08E−79
	Worst	1.39E+08	1.72E−116	2.33E+08	4.91E−110	2.82E+08	5.59E−92	6.01E+08	9.46E−67
	Average	2.13E+07	3.64E−118	4.44E+07	1.04E−111	6.67E+07	1.20E−93	1.36E+08	3.31E−68
	SD	2.63E+07	2.43E−117	6.41E+07	6.95E−111	6.10E+07	7.89E−93	1.20E+08	3.31E−68

Table 13 (continued)

Function	Performance measures	$b = 0.5$		$b = 1$		$b = 1.5$		$b = 2$	
		MFO	E-MFO	MFO	E-MFO	MFO	E-MFO	MFO	E-MFO
f_{10}	Best	6.76E−22	4.80E−299	3.52E−24	6.65E−290	8.65E−22	4.86E−284	2.80E−20	1.15E−227
	Worst	5.78E−07	3.97E−283	1.02E−11	1.73E−266	6.93E−17	2.95E−260	3.47E−14	1.27E−205
	Average	2.15E−08	8.19E−285	2.32E−13	3.51E−268	6.93E−17	8.82E−262	1.44E−15	2.55E−207
	SD	9.40E−08	0.00E+00	1.45E−12	0.00E+00	6.93E−17	0.00E+00	1.44E−15	0.00E+00
f_{11}	Best	1.97E−07	1.74E−71	3.05E−06	4.05E−72	1.05E−04	2.06E−63	7.38E−04	2.03E−50
	Worst	3.00E+03	2.50E−66	3.60E+03	9.68E−65	2.70E+03	1.37E−57	3.60E+03	5.28E−45
	Average	2.84E+02	1.76E−67	576.0118	3.62E−66	6.26E+02	6.47E−59	7.68E+02	2.37E−46
	SD	5.75E+02	4.02E−67	687.3588	1.67E−65	6.91E+02	2.04E−59	7.53E+02	8.01E−46
f_{12}	Best	0.01696	2.63E−127	0.033	1.67E−65	8.93E−02	6.17E−104	0.091684	2.83E−80
	Worst	4.48E+03	6.11E−110	6.98E+03	1.71E−105	5.19E+03	6.17E−104	8.10E+03	1.87E−68
	Average	7.88E+02	1.27E−111	823.4958	3.42E−107	5.19E+03	4.64E−94	2.01E+03	4.09E−70
	SD	1.16E+03	8.64E−111	1.35E+03	2.42E−106	5.19E+03	3.08E−93	2.02E+03	2.65E−69
f_{13}	Best	1.07E+02	0.00E+00	58.70244	0.00E+00	4.58E+01	0.00E+00	12.97674	0.00E+00
	Worst	2.79E+02	0.00E+00	2.63E+02	0.00E+00	1.97E+02	0.00E+00	2.00E+02	0.00E+00
	Average	1.85E+02	0.00E+00	1.60E+02	0.00E+00	1.19E+02	0.00E+00	92.610892	0.00E+00
	SD	37.85178	0.00E+00	41.1879	0.00E+00	3.70E+01	0.00E+00	42.132758	0.00E+00
f_{14}	Best	20.00000	8.88E−16	19.9999	8.88E−16	20.00000	8.88E−16	20.00000	2.75E−07
	Worst	20.000006	20.872233	19.9999	4.44E−15	20.00000	20.668428	20.00000	20.565125
	Average	20.000006	6.1627047	19.9999	9.59E−16	2.00E+01	19.622287	20.00000	19.953637
	SD	9.96E−07	9.4989608	1.03E−07	5.02E−16	4.36E−08	4.047453	2.02E−08	2.8812868
f_{15}	Best	6.05E−07	0.00E+00	2.45E−05	0.00E+00	8.89E−04	0.00E+00	0.012923	0.00E+00
	Worst	90.524563	0.00E+00	1.81E+02	0.00E+00	8.89E−04	0.00E+00	1.81E+02	0.0884466
	Average	7.2774382	0.00E+00	28.97855	0.00E+00	25.296997	0.00E+00	25.40164	0.0017689
	SD	24.75227	0.00E+00	56.0204	0.00E+00	51.697514	0.00E+00	48.434818	0.0125082
f_{16}	Best	93.02829	4.41E−43	8.6625	1.43E−43	21.05645	4.96E−39	1.06E+02	2.91E−31
	Worst	4.79E+02	8.49E−38	5.76E+02	9.47E−37	6.59E+02	4.43E−33	7.25E+02	2.73E−27
	Average	2.41E+02	3.55E−39	2.74E+02	2.87E−38	3.52E+02	1.54E−34	4.67E+02	2.91E−31
	SD	92.364316	1.25E−38	1.42E+02	1.39E−37	1.45E+02	6.70E−34	4.67E+02	2.73E−27
f_{17}	Best	6.635761	1.395583	5.4543	1.652566	2.817941	1.470879	1.24E−04	1.391942
	Worst	97.7317	1.921921	78.3013	2.4243151	62.482561	1.9128414	68.516783	1.934818
	Average	34.002886	1.6899052	31.4418	1.9865545	16.438271	1.6862765	4.9868757	1.673846
	SD	16.16599	0.1337694	11.4949	0.1439446	10.112424	0.1002134	12.379324	1.673846
f_{18}	Best	4.32E−04	3.07E−04	5.12E−04	3.07E−04	4.41E−04	3.07E−04	5.50E−04	3.08E−04
	Worst	0.0203633	4.87E−04	0.002252	0.0017217	0.0203633	0.0018439	0.0203633	0.001666
	Average	0.001478	3.38E−04	0.0011584	3.59E−04	2.14E−03	3.78E−04	0.001591	4.34E−04
	SD	0.001478	4.03E−05	0.0011584	3.59E−04	4.05E−03	3.78E−04	0.0027534	4.34E−04

Table 13 (continued)

Function	Performance measures	$b = 0.5$		$b = 1$		$b = 1.5$		$b = 2$	
		MFO	E-MFO	MFO	E-MFO	MFO	E-MFO	MFO	E-MFO
f_{19}	Best	− 1.03162	− 1.031628	− 1.031628	− 1.031628	− 1.031628	− 1.031628	− 1.031628	− 1.031628
	Worst	− 1.031628	− 1.031628	− 1.031628	− 1.031628	− 1.031628	− 1.031628	− 1.031628	− 1.031628
	Average	− 1.031628	− 1.031628	− 1.031628	− 1.031628	− 1.031628	− 1.031628	− 1.031628	− 1.031628
	SD	2.24E−16	7.11E−10	2.24E−16	2.36E−10	2.24E−16	2.43E−10	2.24E−16	3.29E−09
f_{20}	Best	3.00000	3.00000	3.00000	3.00000	3.00000	3.00000	3.00000	3.00000
	Worst	3.00000	3.00000	3.00000	3.00000	3.00000	3.00000	3.00000	3.00000
	Average	3.00000	3.00000	3.00000	3.00000	3.00000	3.00000	3.00000	3.00000
	SD	1.88E−15	3.82E−06	2.46E−15	1.98E−06	2.05E−15	1.06E−06	2.30E−15	2.51E−07

$b = 1.5and2$ only one function is successful in getting optimum results (Table 13).

Inference In basic MFO, the b parameter basically defines the shape of spiral and its value is considered as one. Author does not provide any explanation about the chosen value. So, experimental analysis is conducted to analyze its effect on algorithm. The different values of b are chosen which are 0.5, 1, 1.5 and 2. It is clear from the above analysis that b = 1 is best suited value for E-MFO. In this case, most of the functions either provide better or comparable values in comparison with other values of b. However, in case of MFO best solutions are obtained in case of $b = 0.5$.

4.7 Statistical testing

The statistical testing includes the Wilcoxon's rank-sum test [32] which tests the performance of two different algorithms. Here, the performance of E-MFO is checked with respect to BA, BFP, DE, FA, FPA, PSO, GA and MFO. Basically, this test is used to find the difference between two algorithms and at the end of this test we get p value as an output. This p value signifies the impact factor of algorithm under test. If p value is < 0.05, then an algorithm is considered as statically significant. The results show in Table 9 that E-MFO performs better in seventeen functions, FA and GA in one and MFO in two functions. The NA value of any algorithm shows the algorithm's superiority over others and other algorithms finds its p value with respect to that algorithm. Even statistical results prove that E-MFO is better algorithm than others (Table 14).

4.8 Convergence graphs

The convergence graphs for all algorithms are drawn and compared to E-MFO in order to check the speed of algorithm. It is observed from Fig. 7 that E-MFO is able to achieve global minima in less number of iterations and has fast convergence rate in comparison with other algorithms.

4.9 Main findings of results

- MFO is recent addition to the field of evolutionary algorithms and has been successfully applied on various real-world problems with different domains of research. But this algorithm has poor exploration and has imbalance between exploration and exploitation.
- So, to improve its performance some modifications are proposed. The modifications include division of iterations and exponential step size to balance exploration and exploitation ability. Cauchy step size is used to further improve the searching capability during initial

Table 14 Statistical testing

Function	BA	BFP	DE	FA	FPA	PSO	GA	MFO	E-MFO
f_1	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	NA
f_2	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.05E−18	NA
f_3	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	NA
f_4	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	NA
f_5	7.06E−18	7.06E−18	7.06E−18	4.65E−13	7.06E−18	2.21E−07	7.50E−18	7.06E−18	NA
f_6	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	NA
f_7	3.31E−20	3.31E−20	1.25E−19	3.31E−20	5.96E−18	7.06E−18	7.06E−18	0.0065	NA
f_8	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	NA
f_9	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	NA
f_{10}	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	NA
f_{11}	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	NA
f_{12}	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−018	7.06E−18	7.06E−18	7.06E−18	NA
f_{13}	3.31E−20	3.31E−20	3.31E−20	3.31E−20	3.31E−20	3.31E−20	3.31E−20	3.31E−20	NA
f_{14}	0.0386	6.05E−18	0.8983	0.0379	6.05E−18	0.0158	0.0158	0.0366	NA
f_{15}	3.31E−20	3.31E−20	3.31E−20	3.31E−20	3.31E−20	3.31E−20	3.31E−20	3.31E−20	NA
f_{16}	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18	NA
f_{17}	7.06E−18	7.06E−18	7.06E−18	NA	7.06E−18	7.06E−18	7.06E−18	7.06E−18	7.06E−18
f_{18}	4.49E−16	7.06E−18	3.52E−17	1.80E−16	1.34E−16	1.71E−14	NA	5.62E−17	4.71E−35
f_{19}	3.31E−20	3.31E−20	4.24E−09	3.31E−20	3.31E−20	2.62E−23	3.31E−20	NA	3.31E−20
f_{20}	5.52E−18	5.52E−18	0.1540	5.52E−18	4.64E−06	0.4082	5.52E−18	NA	5.52E−18

stages and helps in avoiding local minima and it will enhance the exploration. Moreover, influence of best flame is taken for all moths while updating its position.

- The proposed approach is verified on twenty benchmark functions and compared with DE, FA, FPA, BA, GA, PSO and BFP. It is clear from the results that this approach is successful in getting optimum results and avoiding local minima.
- In this paper, effect of population is also analyzed on MFO, E-MFO and other algorithms. The results are taken at three different population sizes namely 30, 60 and 100. Moreover, E-MFO has been tested on the different size problems including high-dimensional problems, i.e., of dimension size of 50,100, 200, 500. And it has been found that for variable population sizes and dimension sizes, the proposed E-MFO algorithm gives highly competitive results for most of the test functions under test.
- Apart from this, there is wide discussion about the parameter b whose explanation is not given in the original paper and this paper clearly describes the importance of this parameter.

4.10 Limitations of proposed approach

E-MFO remains unsuccessful in getting good results for multimodal functions with fixed dimensions. In this case, the results are not equal to expected ones and other nature-inspired algorithms performed better than this algorithm. So, this can be concluded from this limitation that solutions are not able to reach the global minima because of getting trapped in local minima. The solutions provided by these functions are not consistent and hence degrades the performance efficiency of algorithm. Moreover, E-MFO algorithm is stochastic optimization algorithm so their global minimum is not sure.

5 Conclusion and future work

In this work, four modifications are proposed to enhance the performance of MFO. These modifications ensure the balance of exploration and exploitation, increase in diversity and improvement in convergence rate of E-MFO. Division of iterations, exponential step size, Cauchy global search and influence of best flame has been added to improve the performance of MFO. Basically, in E-MFO, iterations are divided into two groups, first group emphasizes exploration and second group on exploitation. Further, instead of using linear decreasing step size, E-MFO used exponential step size to make proper balance between exploration and exploitation. Afterward, Cauchy distribution function is employed which increase the diversity of algorithm and helps the solutions to jump out of local minima effectively.

Fig. 7 Convergence graphs

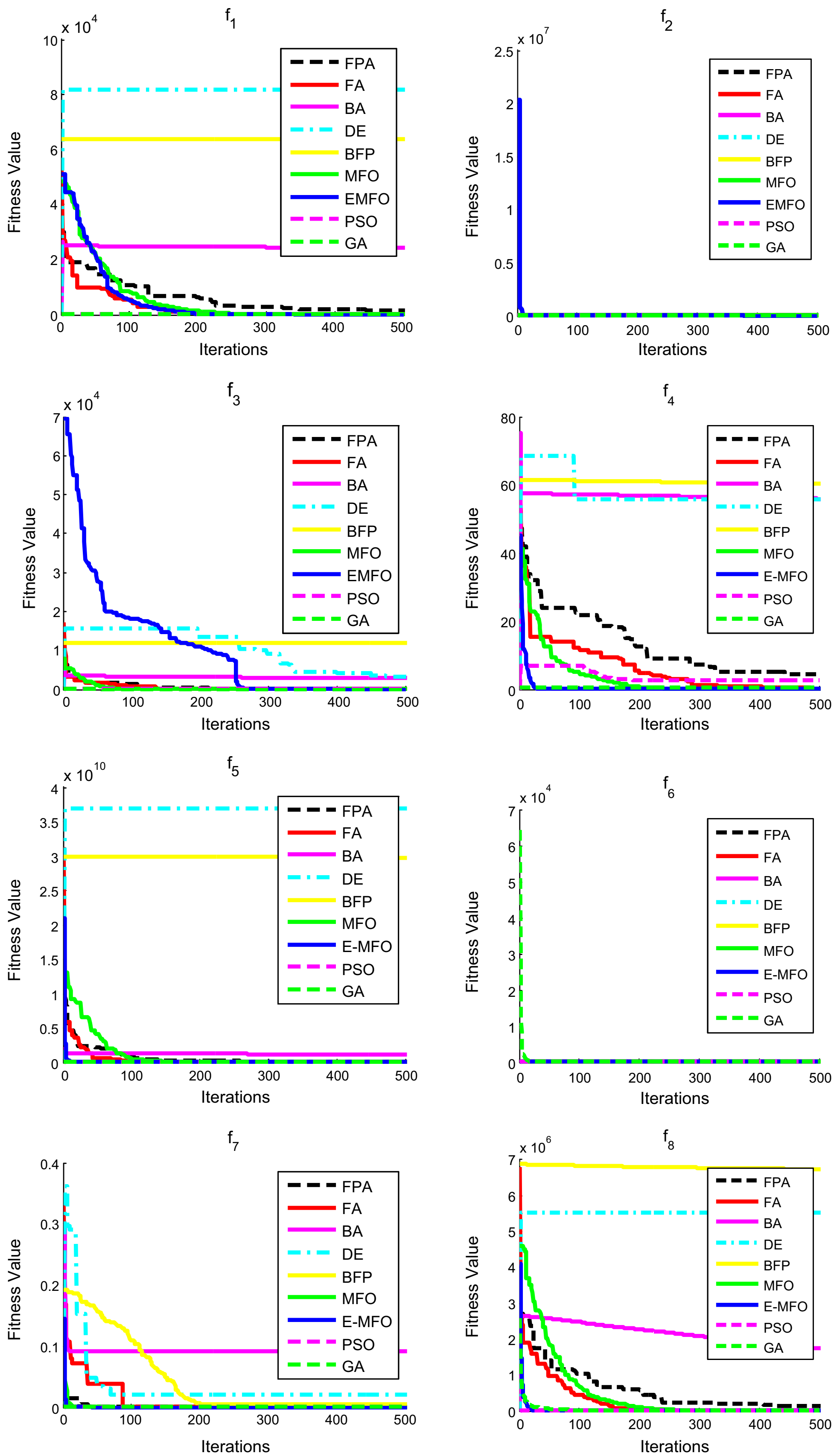


Fig. 7 continued

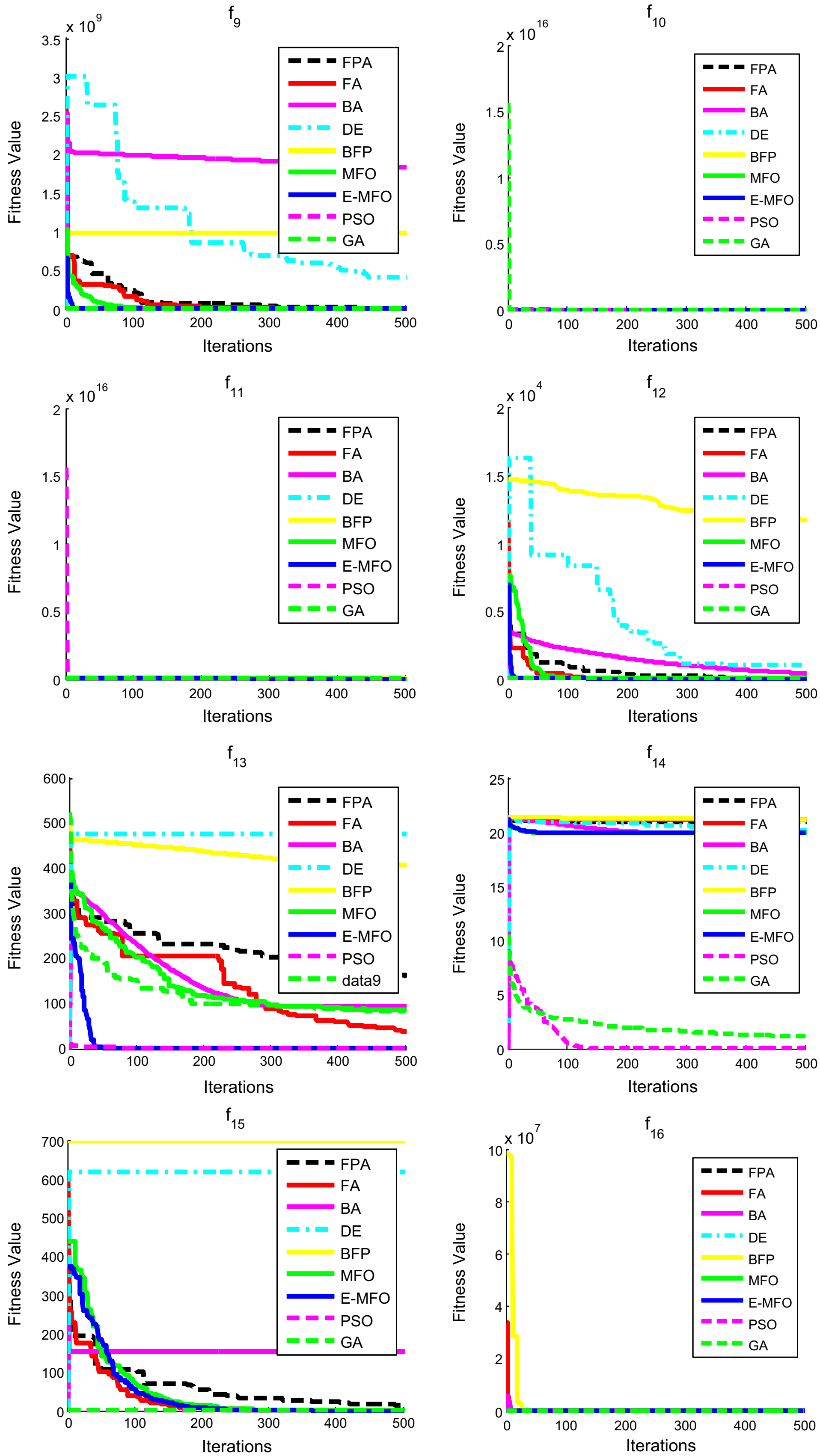
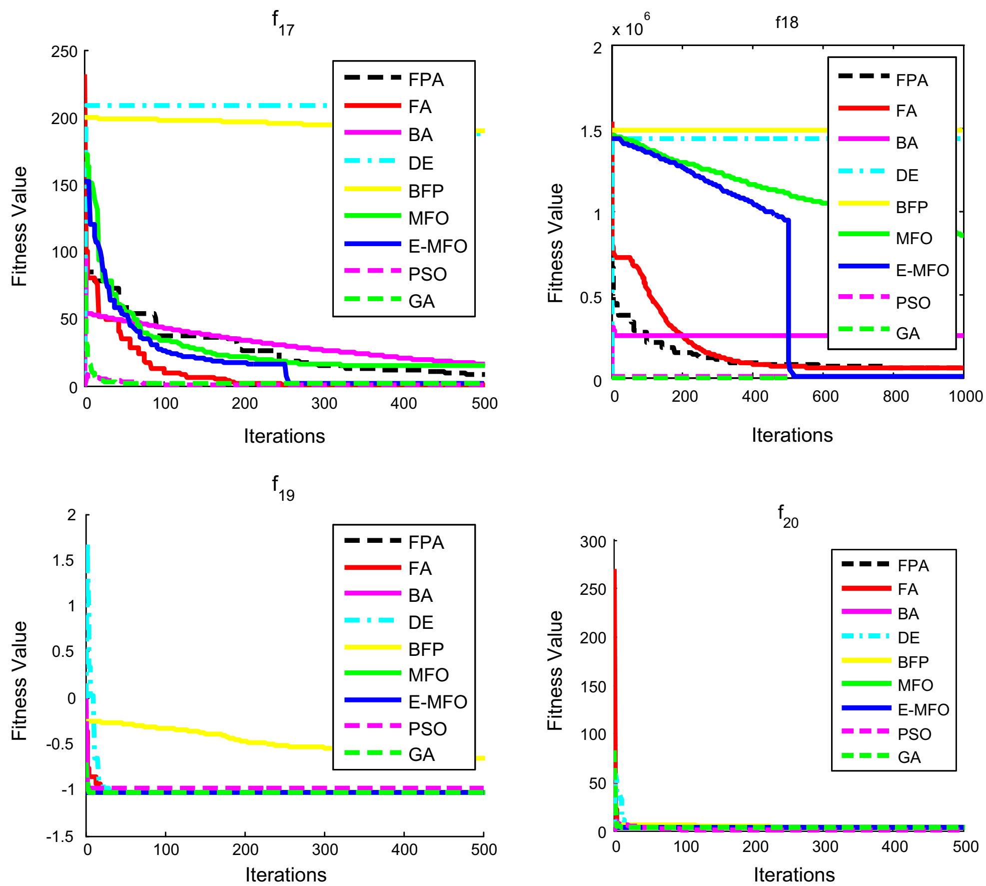


Fig. 7 continued



This property makes E-MFO faster and robust in comparison with MFO. E-MFO includes one controlling parameter which controls the effect of best flame over corresponding flame. This modification leads to more exploitation in algorithm. The effectiveness of E-MFO is checked on twenty benchmark functions and it is clear from the results that it provides better results in comparison with other state-of-the-art algorithms like BA, BFP, DE, FA and FPA. The convergence curves are also drawn which shows that E-MFO has capability to reach the global minima in less number of iterations. To analyze the effect of population on MFO and E-MFO, three different population sizes, i.e., 30,60 and 100 are considered and results show that there is no significant effect of population on the performance of E-MFO but MFO performs better as population size increases. The five different dimension sizes, which are 30, 50, 100, 200 and 500, have been discussed and it is clear from the results that E-MFO performed better at all dimension sizes. Subsequently, four values (0.5, 1, 1.5, 2) of "b" parameter is considered and from result section it is inferred that $b = 1$ is best suited value for E-MFO.

For future work, there are so many directions in which MFO can be efficiently recommended. This algorithm can be applied in various engineering problems like digital filter design, antenna design, wireless sensor network, knapsack problems and others. Also, binary and multi-objective versions can be developed to solve practical optimization problems. Further, to enhance its performance it can be hybridized with other meta-heuristic algorithms. More research can be done by exploiting the parameters of MFO like number of flames and step size. The algorithm can be extended to the field of mechanical, data clustering, big data analytics, VLSI floor planning, etc.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interests regarding the publication of this paper.

References

1. Gutjahr WJ (2009) Convergence analysis of metaheuristics. In: Matheuristics. Springer, Boston, pp 159–187
2. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems (No. 1). Oxford University Press, New York

3. Bansal JC, Sharma H, Jadon SS, Clerc M (2014) Spider monkey optimization algorithm for numerical optimization. Memet Comput 6(1):31–47
4. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, 1995, MHS'95. IEEE, pp 39–43
5. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39
6. Karaboga D, Basturk B (2007) Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In: International fuzzy systems association world congress. Springer, Berlin, pp 789–798
7. Yang XS, Deb S (2009) Cuckoo search via Lévy flights. In: World congress on Nature & Biologically Inspired Computing, 2009. NaBIC 2009. IEEE, pp 210–214
8. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: Gonzalez JR et al (eds) Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, Berlin, pp 65–74
9. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimization. Int J Bio-Inspired Comput 2(2):78–84
10. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
11. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
12. Holland JH (1992) Genetic algorithms. Sci Am 267(1):66–72
13. Storn R, Price K (1995) Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces, vol 3. ICSI, Berkeley
14. Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection, vol 1. MIT Press, Cambridge
15. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713
16. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
17. Rechenberg I (1978) Evolutions strategien. In: Schneider B, Ranft U (eds) Simulationsm ethoden in der Medizin und Biologie. Springer, Berlin, pp 83–114
18. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249
19. Muangkote N, Sunat K, Chiewchanwattana S (2016) Multilevel thresholding for satellite image segmentation with moth-flame based optimization. In: 2016 13th international joint conference on computer science and software engineering (JCSSE). IEEE, pp 1–6
20. Bentouati Bachir, Chaib Lakhdar, Chettih Saliha (2016) Optimal power flow using the moth flam optimizer: a case study of the algerian power system. Indones J Electr Eng Comput Sci 1(3):431–445
21. Yamany W, Fawzy M, Tharwat A, Hassanien AE (2015) Moth-flame optimization for training multi-layer perceptrons. In: 2015 11th international computer engineering conference (ICENCO). IEEE, pp 267–272
22. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
23. Li Z, Zhou Y, Zhang S, Song J (2016) Lévy-flight moth-flame algorithm for function optimization and engineering design problems. Math Probl Eng. https://doi.org/10.1155/2016/1423930
24. Soliman GM, Khorshid MM, Abou-El-Enien TH (2016) Modified moth-flame optimization algorithms for terrorism prediction. Int J Appl Innov Eng Manag 5(7):47–58
25. Allam Dalia, Yousri DA, Eteiba MB (2016) Parameters extraction of the three diode model for the multi-crystalline solar cell/module using moth-flame optimization algorithm. Energy Convers Manag 123:535–548
26. Hassanien Aboul Ella, Gaber Tarek, Mokhtar Usama, Hefny Hesham (2017) An improved moth flame optimization algorithm based on rough sets for tomato diseases detection. Comput Electron Agric 136:86–96
27. Ceylan O (2016) Harmonic elimination of multilevel inverters by moth-flame optimization algorithm. In: International symposium on industrial electronics (INDEL). IEEE, pp 1–5
28. Garg P, Gupta A (2016) Optimized open shortest path first algorithm based on moth flame optimization. Indian J Sci Technol. https://doi.org/10.17485/ijst/2016/v9i48/104255
29. Parmar SA, Pandya MH, Bhoye M, Trivedi IN, Jangir P, Ladumor D (2016) Optimal active and reactive power dispatch problem solution using moth-flame optimizer algorithm. In: 2016 international conference on energy efficient technologies for sustainability (ICEETS). IEEE, pp 491–496
30. Yang XS (2012) Flower pollination algorithm for global optimization. In: International conference on unconventional computing and natural computation. Springer, Berlin, pp 240–249
31. Salgotra R, Singh U (2016) A novel bat flower pollination algorithm for synthesis of linear antenna arrays. Neural Comput Appl. https://doi.org/10.1007/s00521-016-2833-3
32. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing Evolutionary and swarm intelligence algorithms. Swarm Evolut Comput 1(1):3–18