



# An enhanced moth flame optimization with mutualism scheme for function optimization

Saroj Kumar Sahoo<sup>1</sup> · Apu Kumar Saha<sup>1</sup> · Sushmita Sharma<sup>1</sup> · Seyedali Mirjalili<sup>2,4</sup> · Sanjoy Chakraborty<sup>3</sup>

Accepted: 11 November 2021 / Published online: 11 January 2022

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

## Abstract

Nature-inspired meta-heuristics have demonstrated superior efficiency in the solution of complicated nonlinear optimization problems than conventional techniques. In this article, an enhanced moth flame optimization (EMFO) is designed using the mutualism phase from the symbiotic organism search (SOS) algorithm. The suggested approach is examined on 36 classical benchmark functions taken from literature. The outputs of EMFO are compared with the latest meta-heuristic algorithms and variants of the MFO algorithm. The comparison results indicate that our proposed method is competitive from the compared methods. Also, the Friedman rank test is used to evaluate the new algorithm's efficiency, and it is found that the rank of EMFO is superior. Finally, EMFO is being applied to solve seven real-world problems, and the outcomes of the proposed algorithm were found to be satisfactory.

**Keywords** Optimization · Moth flame optimization · Mutualism phase · Benchmark functions · Friedman rank test · Real-world problem · Algorithm · Particle swarm optimization · Genetic algorithm

## 1 Introduction

Optimization is one of the main factors in both industrial purposes and the scientific research world. Many numerical and computational processes have been invented to clear up optimization issues in the last twenty years. However, with the aid of numerical methods, it is very complicated to resolve the problems which are non-convex, highly non-linear, include a giant quantity of variables and constraints. To overcome the drawbacks, such as extra mathematical calculations, initial guess, and convergent problems in discrete optimization problems, a set of optimization algorithms known as meta-heuristics algorithms have been proposed in the latest decades, namely genetic algorithm

(GA) (Holand 1992), particle swarm optimization (PSO) (Kennedy and Eberhart 1995), differential evolution (DE) (Storn and Price 1997), artificial bee colony (ABC) (Yi and He 2014), firefly algorithm (FA) (Yang 2009), cuckoo search algorithm (CS) (Gandomi et al. 2013), symbiotic organisms search (SOS) (Cheng and Prayogo 2014), JAYA algorithm (Arora and Singh 2015), butterfly optimization algorithm (BOA) (Mirjalili 2015), moth flame optimization (MFO) (Wang et al. 2015), monarch butterfly optimization (MBO) (Rao 2016), etc.

Nowadays, meta-heuristic algorithms are extensively used to solve engineering applications for their accessible mathematical operators, uncomplicated executions, and fewer chances to stick at local optimum solutions. Usually, these algorithms start with a randomly taken set of the initial solution and then run the process until the globally optimal solutions of the objective functions are obtained. Broadly we divide meta-heuristic algorithms into two classes, viz., single solution-based (SSB) methods and population-based (PB) methods. The SSB methods perform the search by single search representatives, and a group of search representatives is used in PB methods. Depending on single and social information, each solution's position is renovated in PB methods. Moreover, various solutions can

✉ Apu Kumar Saha

<sup>1</sup> Department of Mathematics, National Institute of Technology Agartala, Tripura 799046, India

<sup>2</sup> Centre for Artificial Intelligence Research and Optimization, Torrens University Australia, Fortitude Valley, Brisbane, QLD 4006, Australia

<sup>3</sup> Department of Computer Science and Engineering, National Institute of Technology Agartala, Tripura 799046, India

<sup>4</sup> Yonsei Frontier Lab, Yonsei University, Seoul, South Korea

quickly check the whole search space; hence, better results are produced than SSB methods. The PB optimization methods are broadly grouped into four types: evolutionary algorithms, swarm intelligence (SI) algorithms, physical or chemical law-based algorithms and human-based algorithms. Apart from these algorithms, several algorithms have been proposed using mathematics concepts such as algebra and geometry. Some of such popular algorithms are sine cosine algorithm (SCA) (Mirjalili 2016), generalized convex approximation (GCA) (Chickermane and Gea 1996), nonlinear integer and discrete programming (NIDP) (Sandgren 1990), and the method of moving asymptotes (MMA) (Svanberg 1987).

The population-based optimization methods have two essential characteristics: diversification (exploration) and intensification (exploitation). Diversification refers to searching the entire space, representing the algorithm's capability in its global search competence. Exploitation refers to the quality of solutions in the course of iterations and shows the ability of a method in the local search and finding the best answer around an already seen potential solution. On the other hand, higher exploration causes a slow convergence rate for achieving a globally optimum solution. Higher exploitation causes convergence approaches to local optimum before key reaching global optima. Thus, a trade-off between these two factors is vital for an optimization algorithm to become efficient and robust.

In the last few decades, researchers have frequently modified several meta-heuristics (for example, improving algorithm formulae, parameters tuning of algorithms, hybridizing two or more algorithms, etc.). They have established other meta-heuristic methods to enhance the efficiency of the basic algorithms for solving complex and challenging scientific problems. They use commonly shared information among multiple agents, which is the main reason behind the popularity of meta-heuristic algorithms.

In the present article, MFO is considered to be thoroughly studied and analyzed. MFO is a swarm intelligence-based algorithm that was introduced in 2015 by Mirjalili (2015). MFO's inspiration came from the moth navigation technique in nature referred to as transverse orientation. The author of the MFO has proved that it has superior performance compared to the other popular meta-heuristic algorithms over twenty-nine benchmark functions. MFO has a solid ability to solve numerous challenging constrained and unknown search space problems, which is the main advantage of MFO among all other traditional algorithms.

Due to the advantages of MFO, such as faster convergence toward global solutions and a smaller number of algorithm-specific parameters, it has been enforced for several applications. Few of them are image segmentation

(Muangkote et al. 2016), PID parameter optimization (Bourouba et al. 2018), economic load dispatch (ELD) problems (Babar et al. 2016), manufacturing industry optimization (Yidiz and Yidiz 2017), power load forecasting (Li et al. 2016a), solar energy devices (Allam and A, Eteiba MB, 2016), multilayer perceptron (Yamany et al. 2015), medical diagnoses (Wang et al. 2017), feature selection problem (Zawbaa et al. 2016), robust routing problem (Khan et al. 2018), unit commitment problem (Reddy et al. 2018), and optical network unit placement (Singh and Prakash 2017).

As a new PB optimization algorithm, however, MFO's performance is still required to improve in some direction, such as convergence rate and global optimum solution. Therefore, a lot of methods were proposed to enhance MFO's efficiency. In (Li et al. 2016b), the author developed a Levy flight operator, which increases population diversity, disrupts each moth's role, and enhances MFO convergence efficiency. Emary and Zawbaa (2016) added chaos parameter in the spiral equation to change the location for moths to improve MFO's efficiency to boost the basic MFO's. In (Apinantanakon and Sunat 2017), Apinantanakon and Sunat have introduced a new strategy named as opposition-based learning (OBL) to enhance the convergence speed of MFO. Elaziz et al. (2020) introduced a new method called OMFODE, by the combination of opposition-based learning and differential evolution (DE) which overcome the above limitations of the basic MFO algorithm. In (Singh and Salgotra 2018), three new techniques such as iteration division, Cauchy distribution function, and best flame technique have been used by authors to achieve a good trade-off between intensification and diversification. In (Li et al. 2018), the authors used two new techniques; one is flame generation technique by DE, and the other is dynamic guidance of flame. The obtained results outperform six modified MFO algorithms on CEC 2013 test suits concerning convergence rate and better solution. Savsani and Tawhid (2017) have proposed a multi-objective version of MFO by sorting non-dominated outcomes and crowding distance approach, which generates actual Pareto front and helps maintain diversity among the set of optimal results. Nanda and Multi-objective moth flame optimization (2016) have proposed multi-objective moth flame optimization by exploring and exploiting MFO algorithm, grid techniques, and non-dominated solutions for generating the Pareto front and finding the solutions of a multi-objective problem.

In recent times, meta-heuristics and hybrid meta-heuristics have played a significant role in the research field. Hybridization is used to solve complex optimization problems due to the combination of two to three individual meta-heuristics algorithms. It is also helpful for improving the meta-heuristics algorithm with additional techniques

for better improvement in results, run-time, or both. Some of the hybrid methods of MFO have been developed by different authors, such as Bhesdadiya et al. (2017) proposed an algorithm by integrating PSO and MFO, which enhance the diversification search during solving high complex design problems and showed superiority in solving unconstrained optimization problems. In (Khalilpourazari and Khalilpourazary 2019), the author developed a modified algorithm of MFO by the mixture of water cycle algorithm (WCA) and MFO noted as WCMFO. Here MFO increases the exploitation, and WCA improves the diversification of WCMFO. Also, it has been used in solving constrained optimization problems. To overcome the premature convergence problem of MFO, Wu et al. (2018) proposed a new MFO algorithm which combines chaotic operator crisscross strategy in MFO. Kamalpathi et al. (2018) developed a new hybrid algorithm by the mixture of fuzzy logic control (FLC) and MFO algorithm and have been used to solve torque tap problem of the BLDC engine. In this literature, MFO has been used to control the minimum line stream harmonics and voltage in a motor system. In contrast, FLC has been used to increase the MFO's efficiency by improving the MFO update position. Sarma et al. (2017) combined two algorithms named MFO and GSA and applied them to determine the degree of food rottenness problem, minimizing monetary losses due to food and storage. Also, to disable the shortcomings of the original MFO algorithm like low-quality solutions and slow convergence, Sayed and Hassanien (2018) developed another new algorithm by combining simulated annealing (SA) MFO algorithm. Recently, Li et al. (2021) developed a high-quality improvement in the MFO algorithm named ODSMFO to overcome the demerits of the MFO algorithm. The author added the OBL mechanism and DE algorithm to obtain good quality solution and diversity enhancement, respectively, then used an enhanced local search technique based on shuffled frog leaping algorithm (SFLA) to improve global search ability and finally applied death mechanism to eliminate individuals with low fitness value. Shan et al. (2021) proposed double adaptive weight mechanism for stabilization of MFO algorithm (WEMFO). The main goal of the WEMFO is to enhance the search capability and maintain a good trade-off between diversification and intensification of the classic MFO algorithm. The authors used two weights to adjust the search strategy adaptively in a different phase of the algorithm. Then, the efficiency of the proposed WEMFO was measured by applying it to train kernel extreme learning machines (KELM) and several engineering problems. Khan et al. (2021) used the strength of the MFO algorithm in an integrated power plant system containing a stochastic window (SW). The authors combined active set algorithm (ASA), interior point algorithm (IPA), and sequential

quadratic programming (SQP) with MFO algorithm and developed three hybrid techniques named MFO-ASA, MFO-IPA, and MFO-SQP to solve economic load dispatch (ELD) problem and ELD including stochastic wind (ELD-SW). Pelosi et al. (2020) introduced an enhanced version of the basic MFO algorithm named improved moth flame optimization (IMFO) to reduce the shortcomings of the MFO algorithm, such as convergence speed and global searchability. They added a weight factor to the proposed algorithm to maintain equilibrium between local search and global search. Ma et al. (2021) introduced an improved version of the MFO algorithm developed to overcome the basic MFO algorithm's issues, such as slow convergence and convergence to a local minimum. The inertia weight of the diversity feedback control and the small probability mutation factor (added after the position update phase) is embedded in the proposed exploration balancing (and exploitation) and optimization performance algorithm. Zhao et al. (2020) proposed improved population-based techniques named improved MFO algorithm in which mutation operator and linear search techniques are used for position updating of original MFO algorithm and OBL strategy for a flame generation. Kigsisirsin and Miyauchi (2021) proposed a method that is derived from the MFO algorithm named alternative binary MFO (BAMFO) to solve unit commitment (UC) problems. The main disadvantage of the MFO algorithm is a prefixed flame strategy responsible for stocking at a local optimum. In this proposed BAMFO, the authors introduced four flame generation strategies instead of following the predetermined flame strategy of the basic MFO algorithm. After that, a repair strategy was submitted to solve the UC problem. Tumar et al. (2020) developed a modified MFO algorithm named EBMFO (enhanced binary moth flame optimization algorithm) for the prediction of software faults by using adaptive synthetic sampling (ADASYN). In this literature, the original MFO algorithm's transformation to binary MFO algorithm has been discussed and then proposed the upgraded version named EBMFO algorithm. Spare and Mini (2021a) offered a new version of the MFO algorithm, namely the Emulous mechanism based multi-objective MFO algorithm (EMMFO). In this literature, pairwise competitions between the moths have been used to update the position of the moth in each iteration. Zhang et al. (2020) invented a new novel called improved multi-objective optimization and have been applied to solve the cascade reservoir model. In short, the proposed method is called R-IMOMFO, and the formulation has three phases. In the first phase, the authors of the literature first improved the basic MFO algorithm by adding three mechanisms (position update method, flame population update strategy, and inspiration of the moth linear flight path) to overcome the local stagnation and enhance the MFO algorithm. In the

second phase, R-domination (with three reference points) technique has been used to distinguish the solution in Pareto domination. Finally, five new algorithms have been generated by integrating different evolutionary algorithms and multi-objective mechanisms to verify the performance of both IMFO and R-domination. Dabba et al. (2021) developed a modified version of the MFO algorithm named mutual information maximization-modified moth flame algorithm (MIM-MFA) and then applied it to solve gene selection in microarray data classification by the help of MIM. Kadry et al. (2021) proposed a new technique by using Kapur's threshold image segmentation and modified MFO algorithm and then applied it to remove the tumor section from the clinical-grade MRI slices recorded with Flair and T2 modalities. The suggested algorithm has experimented on benchmark images of BRAINIX and TCIA-GBM. The experimental results proved the superiority against the T2 modality. Spare and Mini embedded (Sapre and Mini 2021b) a new algorithm named differential moth flame optimization algorithm (DMFO) and applied it to reduce wireless sensor networks' problems (WSNs) with mobile sink. Dash et al. (2020) applied Jaya-based MFO (JMFO) and basic MFO algorithm in IEEE network to minimize the loss transmission through an efficient position of FACTS devices. Two compensators named TCSC and SVC are used in IEE 14 AND IEE 30 bus system as a fitness function for MFO and JMFO algorithm. Gupta et al. (2020) developed a higher version of the MFO algorithm named modified MFO (MMFO) and applied it for suitable feature selection problems. Suja (2021) developed a modified algorithm to improve the performance and reduce the power quality issues for smart grid (SG) systems. The name of the modified MFO algorithm is Leavy flight MFO (LMFO) algorithm. The suggested method helps in getting better optimal solutions with respect to the objectives of SG systems.

Apart from the above modifications on MFO algorithm, various researchers have worked on other efficient meta-heuristic algorithms to enhance their performances. Some of those works are given below:

Asghari et al. (2021) developed two-hybrid methods named chaotic-based hybrid whale and PSO by integrating PSO with WOA (in short CWP) and multiswarm version of CWP (in short MCWP) which integrate MFO with improved WOA by using various chaotic maps and roulette wheel selection operator. Asghari et al. (2021) introduced another hybrid method named chaotic GWO and WOA (CGWW) by combining GWO with WOA to overcome the shortcomings of the both algorithm and developed an efficient algorithm. At first, the authors modified WOA and then integrate it with GWO. After that, chaotic maps have been applied to the hybrid GWO and WOA for enhancement of diversification and intensification. Gharehchopogh

and Gholizadeh (2019) proposed a comprehensive survey on whale optimization algorithm and its application. The author provided an updated review version of the recently proposed whale optimization algorithm. They added various variants, improved and hybrid version of WOA with its application in various paths, including image processing, networks, engineering, task scheduling, and other engineering applications. Masdar and Zangakani introduced (Masdari and Zangakani 2020) an extensive survey on proactive virtual machine placement approaches, they split it into various categories such as simulation software, workload data, power management method, evaluation parameters, and prediction factor according to their applied forecasting methods. Masdar et al. (2017) developed another survey paper on PSO-based scheduling algorithms in cloud computing. They presented a deep analysis of the task and workflow scheduling scheme on PSO for the cloud environment and provided the classification based on the type of PSO algorithms. Masdar et al. (2019) introduced a new modified version of artificial bee colony algorithm named chaotic discrete version of an artificial bee colony (CDABC) with the help of chaotic maps. The author applied this algorithm to organize the lifetime of wireless sensor networks (WSNs) by choosing appropriate nodes. Nama et al. (2020a) proposed a novel hybrid meta-heuristic algorithm by combining teaching-learning-based optimization (TLBO) algorithm and quadratic approximation (QA), where the QA technique is applied to improve the global, as well as local search capability of the proposed algorithm. Nama et al. (2017a) proposed a new hybrid algorithm named hybrid symbiosis organisms search (HSOS) by integrating symbiosis organisms search (SOS) algorithm with simple quadratic interpolation (SQI), which helps in enhancing the robustness of the algorithm. Sharma et al. (2021) introduced a novel hybrid butterfly optimization algorithm named MPBOA, where the BOA is combined with mutualism and parasitism phases of the SOS algorithm to enhance the search behavior of BOA, which ensures a better trade-off between the global and local search of the proposed MPBOA algorithm. Sharma and Saha (2021) introduced a powerful hybrid algorithm named BOSCA by combining SCA with BOA, which helps stabilize the global exploration and local exploitation ability of the proposed algorithm. Chakraborty et al. (2021a) introduced a powerful hybrid meta-heuristic algorithm named WOAmM. The mutualism phase from the SOS algorithm is modified and integrated with WOA to alleviate premature convergence's inherent drawback. Chakraborty et al. (2021b) introduced a new hybrid method named SHADE-WOA by integrating modified WOA with success history-based adaptive differential evolution (SHADE). The main goal of this hybrid method is to reduce the shortcomings of both algorithms and guide both

algorithms to explore and exploit in the search space and helps obtain good quality of solutions.

Another standard SI (Swarm Intelligence)-based algorithm is SOS, created in 2014 by Cheng and Prayogo. SOS comprises three, namely mutualism, commensalism, and parasitism process. Out of these three stages, the SOS phase of mutualism and commensalism describes enhancing the SOS algorithm's local exploitation potential. In contrast, the phase of parasitism deals with improving the algorithm's exploration capability. The results obtained from three versions of SOS with some combinations of benefit factors and adaptive benefit factors showed that it achieves a better trade-off between intensification and diversification in search space. From (Gandomi et al. 2013), it can be concluded that SOS is a powerful algorithm in terms of efficiency and solving numerous complicated real-life problems, which makes it a powerful algorithm as compared to other traditional algorithms. SOS has some advantages as it mimics common parameters like maximum number of feature evaluation and population size, has good capacity for exploitation with commensalism and mutualism processes, and has capacity for discovery by integrating mutation and cloning into the parasitism system.

Due to the above advantages, SOS has been applied in solving numerous engineering optimization problems. Few of them have been found in the literature which are given in Kavousi-Fard et al. (2015); Nama et al. (2016) and (Verma et al. 2017). Lots of modifications have been developed to improve SOS performance for various characteristics and structures of the problem. The details of these modifications have discussed in Nama et al. (2017b), Kumar et al. (2019) and Tejani et al. (2019), etc.

Though the MFO versions mentioned above have proved the most beneficial enhancements in their efficiency, they may also be caught at local optima and fail to generate excessive-performance flames and low convergence charge when engaged in solving a complex optimization problem. Recently, in m-MBOA (Sharma et al. 2020), the creator utilized the SOS mutualism step in the exploration section of BOA to decorate the overall performance of the original BOA algorithm. To tackle these problems and motivated by way of the above works, in this article, a novel expanded MFO (namely, EMFO) has been added in order to improve the performance of original MFO similarly by using adding the mutualism section of SOS with the simple MFO algorithm. The effectiveness of the proposed EMFO algorithm has been examined on a set of 36 benchmark take a look at functions from the literature, and also, the acquired outcomes have been in contrast with some ultra-modern optimization algorithms and determined that EMFO performs higher than the different meta-heuristic optimization algorithms.

The rest of the present article is designed as follows: A short summary on MFO algorithm is shown in Sect. 2. The mutualism phase is proposed in Sect. 3. The proposed EMFO algorithm and computational complexity of the proposed method are shown in Sect. 4 and Sect. 5, respectively. The simulation results and performance are presented in Sect. 6. Further in Sect. 6, a brief detail on benchmark functions, statistical test, convergence analysis, and comparison with variants of MFO algorithm are presented. The application of real-world problem is shown in Sect. 7 and at the end, conclusions are discussed in Sect. 8.

## 2 Moth flame optimization

Moths are insects and belong to the class of Arthropoda. The navigation techniques of moths are unique, which attracts researchers to think about it. Moths travel in the night with the help of the moonlight, and for navigation, moths utilize a transverse orientation mechanism. Through crosswise inclination, they fly using moonbeam by keeping a fixed tendency toward the moon for a long journey in a straight path. The efficiency of direction depends on the distance of flame, i.e., when the distance between them decreases, the moth moves in a helix path around the flame which connects the moth to the flame. Using these behaviors of moth and mathematical modeling, the MFO algorithm was developed by Mirjalili in 2015.

### 2.1 MFO algorithm

In basic MFO, all moths are expressed as a set of candidate's solutions and their position is expressed as a vector of decision variables. Let us consider the following matrix for moths

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \ddots & \cdots & \cdots & x_{2,n} \\ \vdots & \cdots & \ddots & \cdots & \vdots \\ x_{N-1,1} & \cdots & \cdots & \ddots & x_{N-1,n} \\ x_{N,1} & x_{N,2} & \cdots & x_{N,n-1} & x_{N,n} \end{bmatrix} \quad (1)$$

where  $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]$ ,  $i \in \{1, 2, \dots, N\}$ .  $N$  indicates moths' number at initial population and  $n$  as variable numbers.

Now the fitness vector of moth is shown in below

$$\text{Fit}[X] = \begin{bmatrix} \text{Fit}[X_1] \\ \text{Fit}[X_2] \\ \vdots \\ \text{Fit}[X_n] \end{bmatrix} \quad (2)$$

The second key point of the MFO is flame matrix. Here the size of both moth's matrix (X) and flame matrix (FM) are same as each moth flies around the corresponding flame.

$$\begin{aligned}
 FM &= \begin{bmatrix} FM_1 \\ FM_2 \\ \vdots \\ FM_N \end{bmatrix} \\
 &= \begin{bmatrix} Fm_{1,1} & Fm_{1,2} & \cdots & Fm_{1,n-1} & Fm_{1,n} \\ Fm_{2,1} & \ddots & \cdots & \cdots & Fm_{2,n} \\ \vdots & \cdots & \ddots & \cdots & \vdots \\ Fm_{N-1,1} & \cdots & \cdots & \ddots & Fm_{N-1,n} \\ Fm_{N,1} & Fm_{N,2} & \cdots & Fm_{N-1} & Fm_{N,n} \end{bmatrix}
 \end{aligned}
 \tag{3}$$

Also, the fitness vector of flame matrix is stored in following matrix, i.e.,

$$\text{Fit}[FM] = \begin{bmatrix} \text{Fit}[FM_1] \\ \text{Fit}[FM_2] \\ M \\ \text{Fit}[FM_n] \end{bmatrix}
 \tag{4}$$

Here  $\text{Fit}[*]$  is a candidate solution's fitness function. In fact, MFO has two important component one is moth and other is flame, where moth moves through the respective flame to achieve suitable outcomes and the best outcomes acquired by the moth is known as flame. As the moth moves in a spiral manner; therefore, the author of MFO has defined a spiral function called logarithmic spiral function in order to create a spiral path which is represent in following equation.

$$x_i^{K+1} = \begin{cases} \delta_i \cdot e^{bt} \cdot \cos(2\pi t) + Fm_i, & i \leq N \cdot FM \\ \delta_i \cdot e^{bt} \cdot \cos(2\pi t) + Fm_i, & i \geq N \cdot FM \end{cases}
 \tag{5}$$

where  $\delta_i = |x_i^K - Fm_i|$ , i.e., the distance between the  $i$ th moth  $M_i$  and its specific flame  $F_i$ ;  $b$  is a constant to recognize the shape of the search for helix flight shape;  $t$  be any random number between -1 and 1 referring to how much closer the moth is to its specific flame. Illustration 1 represents that a moth flies toward its flame in a helix manner, with a distinct value of  $t$  in a one-dimensional manner. An adjustable course of action has been suggested to lessen the variable ( $t$ ) value over the repetition, which enhance the effectiveness of both exploration and exploitation in first and last iterations, respectively.

$$a_1 = -1 + \text{current}_{\text{iter}} \left( \frac{-1}{\text{max}_{\text{iter}}} \right)
 \tag{6}$$

$$t = (a_1 - 1) \times r + 1
 \tag{7}$$

where  $\text{max}_{\text{iter}}$  represents the number of maximum

iterations,  $a_1$  be the convergence constant decreases from  $-1$  to  $-2$  linearly proving that both diversification and intensification occurs in MFO algorithm. Each moth must renovate its position to just one flame to bypass local minima. Flames are sorted and modified in each iteration according to the fitness value. First, moth changes its location according to the first flame and last moth as per the last. However, this positioning can harm the exploitation of best solutions for  $N$ -flame positions. To overcome this issue, the number of flames ( $N \cdot FM$ ) is reduced over the iteration can be obtained by the following formula

$$N \cdot FM = \text{round} \left( N \cdot FM_{Lst\text{it}} - \text{crnt} \cdot \text{it} \frac{(N \cdot FM_{Lst\text{it}} - 1)}{\text{max it}} \right)
 \tag{8}$$

where both the flame number and last iteration flame number are denoted by  $N \cdot FM$  and  $N \cdot FM_{Lst\text{it}}$ , respectively. Figures 1 and 2 represent the moths place corresponding to the first and last iterations of the method and spiral movement of moth around the flame, respectively. The details about MFO are described in Mirjalili (2015).

### 3 Symbiotic organisms search algorithm

SOS belongs to the SI-based design algorithm that has been invented by Cheng and Prayogo and can be used for a number of real problems. It copies the interdependency between various species to grow in a habitat as an individual solution in a community. The efficiency and sturdiness of SOS are standard and practically reasonable with respect to both benchmark and real-life problems. It haphazardly produces the first community by initializing  $N$

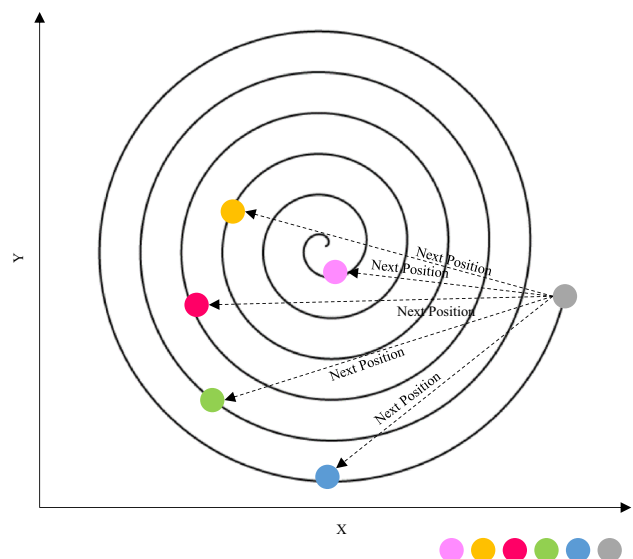


Fig. 1 Position of moths

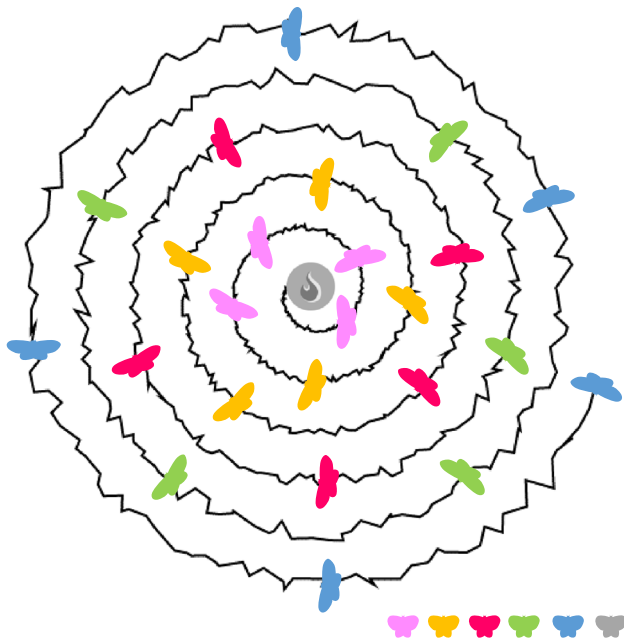


Fig. 2 Spiral movement of moth

organisms and then using the three-way techniques like mutualism, commensalism, and parasitism phases new solutions are modernized. We have only merged the mutualism step of SOS in our proposed algorithm which is summarized in below.

When in nature, two different species get benefit from their interaction then it is called mutualism. Let us take two species  $X_i$  and  $X_j$  from the ecosystem where  $X_i$  interacts with randomly selected species  $X_j$  to create new solutions by using Eqs. (7) and (8):

$$X_{i\text{new}} = X_i + r[0, 1] \times \left( X_b - \left( \frac{X_i + X_j}{2} \right) \times \text{Ben}_{F_1} \right) \quad (9)$$

$$X_{j\text{new}} = X_j + r[0, 1] \times \left( X_b - \left( \frac{X_i + X_j}{2} \right) \times \text{Ben}_{F_2} \right) \quad (10)$$

where  $r[0, 1]$  be a random number lying in the range  $[0, 1]$ ,  $X_b$  be the best ecosystem organism,  $\text{Ben}_{F_1}$  and  $\text{Ben}_{F_2}$  are the benefit factors whose value is either 1 or 2. These factors indicate the level of benefit to each organism, and mutual vector, i.e.,  $(X_i + X_j/2)$ , implies the characteristics relationship between  $X_i$  and  $X_j$ .  $X_{i\text{new}}$  and  $X_{j\text{new}}$  will then be compared to  $X_i$  and  $X_j$  to select the best solution for each pair. In mutualism, new solutions are created on the basis of  $X_b$ .

### 4 Proposed method

The main motto of any meta-heuristic algorithm is to handle the balance between exploration and exploitation. We know that excessive exploration is the reason for losing

optimal solutions because it spends more time searching the uninteresting regions. On the other hand, extreme exploitation is the main reason for premature convergence as the population rapidly lacks diversity. So, better performance of any algorithm is achieved when it maintains stability between diversification and intensification. In basic MFO, the parameter value  $t$  supports diversification and intensification. The power of the exponent factor ' $t$ ' gives a better clarification about exploration and exploitation. We know that the next position of the moth is obtained from the spiral Eq. 5. In this equation,  $t$  defines the next moth's location relative to the flame. When  $t = -1$ , the next position moth is nearest, and  $t = 1$  represents the farthest moth's position regarding the flame. Therefore, exploration and exploitation occur when any moth's next position belongs to the space to moth and flame and does not belong to the space, respectively. We observe that the sorting procedure of the best moth produces flames, and in other side position of moths are updated to the flame, leading to an imperfection in MFO and a reason for losing the promising individuals in the search space. Due to the more exploitation of basic MFO, some best solutions are stuck at local optima. To escape these solutions, we applied a simple step called the mutualism step of SOS in MFO, which is very effective in maintaining a balance between diversification and intensification in the search space and accelerates the speed of the solution. In mutualism phase, two candidate solutions update their position with the help of the best solution and thus they help themselves to find new solutions as the second candidate solution is chosen randomly and use of best solution confirms that the strategy also helps the solutions to find the nearby solutions of already existed best solution. For these benefits of mutualism strategy, Sharma et al. (2020), Chakraborty et al. (2021a), Wang et al. (2020), Tan et al. (2020), and Nama et al. (2017b) have applied mutualism scheme to various algorithms to enhance the performances of those algorithms to produce efficient algorithms. Our proposed methodology is called EMFO with the goal of maximizing population diversity against premature convergence and accelerating convergence speeds, i.e., this framework is useful to create a good balance between the diversification and intensification ability of MFO. We start the algorithm in the similar manner like MFO, and then, we apply mutualism phase (Eqs. 11 and 12) for position updating, i.e., in mutualism phase, we take two organisms (here organism means moths) from the population for updating the position of each moth in each iteration and share information with another randomly chosen moth to update their respective positions in the search space. The formulation of mutualism phase is presented below:

$$X_{i_{new}}^{K+1} = X_i^K + r[0, 1] \times \left( X_b^K - \left( \frac{X_i^K + X_j^K}{2} \right) \times \text{Ben}_{F_1} \right) \tag{11}$$

$$X_{j_{new}}^{K+1} = X_j^K + r[0, 1] \times \left( X_b^K - \left( \frac{X_i^K + X_j^K}{2} \right) \times \text{Ben}_{F_2} \right) \tag{12}$$

where  $X_j^K$  is randomly chosen another solution and  $X_{i_{new}}^{K+1}$ ,  $X_{j_{new}}^{K+1}$  are new updated populations.  $\text{Ben}_{F_1}$  and  $\text{Ben}_{F_2}$  are benefit factor w.r.t  $X_i^K$  and  $X_j^K$  whose value is randomly considered as 1 or 2. Also, we plot a graph between the values of t with respect to the iterations in Fig. 3 for EMFO which provides the better value for t as compared to MFO. Due to the above characteristics, the proposed approach could provide superior performance over MFO and maintains good balance between exploration and exploitation.

The main EMFO steps can simply be shown in Algorithm 1 and summarized in below.

*Step 1:* Initialize all parameters such as number of populations, maximum iteration, and function evaluation randomly.

*Step 2:* Apply the sorting procedure to the both moth matrix and flame matrix w.r.t the fitness value and update number of flames using Eqs. 8.

*Step 3:* Update r and t using Eqs. 6 and 7.

*Step 4:* Update moths position w.r.t corresponding flame using Eq. 5.

*Step 5:* Update the new solution by using Eqs. 11 and 12 and then find the fitness value of the new solutions. Best fitness gives the optimum value.

*Step 6:* If it does not satisfy the stopping criteria then go to 2nd step until to get the best fitness value.

### 5 Computational complexity of EMFO

Complexity of any algorithm is a function which provides the running time or space with respect to input size. This is of two kinds: One is complexity of space and other is time complexity. The process of finding a formula for total space that will be required toward execution of the algorithm is referred as space complexity. Also, process of finding a formula for total time required for successful execution of algorithm is known as time complexity. The complexity of EMFO also depends on initialization of moth position ( $T_{IMP}$ ), evaluation of moth position ( $T_{EMP}$ ), searching of spiral flight ( $T_{SSF}$ ), flame generation ( $T_{FG}$ ), and position updating in mutualism phase  $T_{MP}$ . Let maximum iterate number, variable number, and moths' number are denoted by  $I$ ,  $D$ , and  $N$ , respectively. Here we will use time complexity for the comparison of both EMFO and MFO algorithm. According to the quicksort algorithm, computational complexity for sorting N-flame and N-moth are lying between  $3N\log 3NI$  and  $(3N)^2I$  toward worst and best case

$$\begin{aligned} T_{m-MFO} &= T_{IMP} + T_{EMP} + T_{SSF} + T_{FG} + T_{MP} \\ &= ND + NDI + NDI + (3N)^2I + NDI \\ &= ND + 3NDI + 9N^2I \end{aligned}$$

Hence, time complexity for EMFO with respect to worst case is  $O [NI (D + N)]$ . Also, from Mirjalili (2015), the time complexity of MFO for the worst case is  $O [NI (D + N)]$ . Therefore, both MFO and EMFO has same complexity.

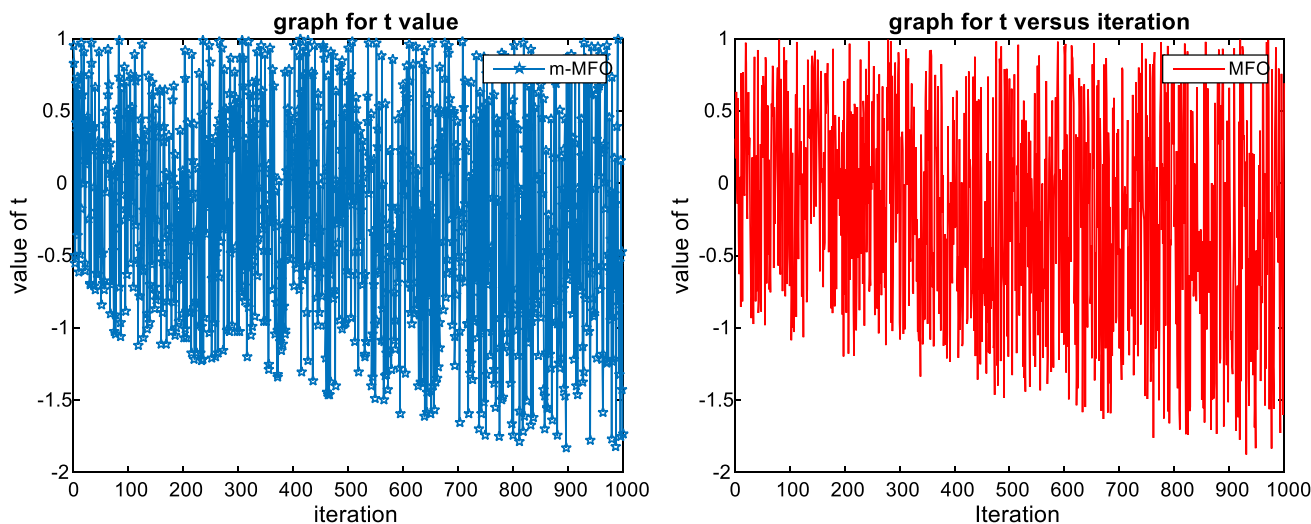


Fig. 3 Comparison between EMFO and MFO w.r.t the parameter value t



```

Objective function  $f(X)$ ,  $X = (x_1 x_2 \dots x_{dim})$ ;
for i= 1: N
    for j= 1: n
        Generate n organism solutions  $X_i$  ( $i = 1, 2, \dots, n$ ) using
equation
         $X(i, j) = L.B(i) + (U.B(i) - LB(i)) * rand$ 
    end
end
While iteration < Max_iteration
    if iteration==1
        Enter N.FM= N in initial population
    else
        Employ eqn.8
    end
    FM=Fitness Function  $f(X)$ ;
    if iteration==1
        Sort the moths according to FM
        Update the Flames
        Iteration=0;
    else
        Sort the moths based on FM from last iteration
        Update the Flames
    end
    Reduce the convergence constant
    for j = 1: N
        for k= 1: n
            Find r and t using eqn.6) & eqn.7)
            Update moths position as to their particular flame
        end
    end
    select one solution randomly ( $i \neq j$ );
    Update the solution  $X_i$  &  $X_j$  according to Eqn. (11) and (12);
    Find new solutions best value;
end while
Output: The best solution with the minimum fitness function
value in the ecosystem;

```

## 6 Results and discussion

This section includes a brief note on the benchmark functions and several discussions on the obtained results of both unimodal and multimodal benchmark functions. In Sect. 6.1, details of the benchmark functions are discussed. In Sect. 6.2, the experimental setup of our proposed method has been discussed. In Sect. 6.3, a comparison of EMFO with basic MFO and other evolutionary algorithms has been discussed. Friedmann rank test and convergence analysis of benchmark functions are presented in Sects. 6.4 and 6.5, respectively. In subsection 6.6, a comparison of EMFO with six variants of the MFO algorithm is presented with statistical measurement.

### 6.1 Benchmark functions

Every new meta-heuristic algorithm's performance must be validated and compared with other existing meta-heuristic algorithms over a good set of test functions. Thus, benchmark functions play an essential role in terms of reliability, verification, and efficiency of the algorithms. These test functions have been carefully selected from Jamil and Yang (2013) and are presented in Appendix 1. To validate our proposed EMFO algorithm, 36 benchmark functions have been selected and split into two parts, namely unimodal benchmark functions and multimodal benchmark functions.

The selected unimodal functions (from F1 to F15) include only one local optimum value called the global minimum value of the corresponding unimodal function.

**Table 1** Comparison of unimodal functions with MFO, SOS, PSO, JAYA, DE, and BOA

Function	EMFO	MFO	SOS	PSO	JAYA	DE	BOA	
F1	M	0	6.03E - 173	1.76E - 51	1.39E + 04	2.61E - 165	0	
	SD	0	0	1.12E - 50	3.54E + 03	0	0	
F2	M	0	0	3.10E - 53	1.86E + 12	1.40E - 172	0	
	SD	0	0	2.10E - 52	2.08E + 11	0	0	
F3	M	0	0	1.31E - 31	1.47E + 04	1.30E - 01	0	
	SD	0	0	6.65E - 31	2.91E + 03	4.85E - 01	0	
F4	M	2.69E + 01	2.89E + 001	- 1	2.57E + 02	2.83E + 07	2.35E + 01	2.88E + 01
	SD	1.28	6.44E - 002	0	3.73E + 02	1.20E + 07	4.98	3.12E - 02
F5	M	0	0	2.27E - 74	1.20E + 03	6.54E + 04	1.14E - 175	0
	SD	0	0	2.41E - 74	5.44E + 02	1.57E + 04	0	0
F6	M	2.03E - 197	0	1.24E - 86	1.05E - 01	6.56E + 01	1.02	6.99E - 153
	SD	0	0	1.26E - 86	6.53E - 02	6.69	4.59E - 01	1.47E - 152
F7	M	0	0	- 4.19E + 02	7.28E + 01	4.58E + 02	8.15E - 91	0
	SD	0	0	0	2.27E + 01	8.32E + 02	1.23E - 90	0
F8	M	1.59E - 05	1.04	3	0	6.22	0	0
	SD	2.23E - 05	1.78	1.94E - 15	0	6.18	0	0
F9	M	0	0	7.51E - 13	0	2.51E - 01	0	0
	SD	0	0	3.99E - 12	0	2.33E - 01	0	0
F10	M	1.31E - 12	0	4.62E - 03	3.36E + 03	1.02E + 02	1.30E + 02	0
	SD	4.94E - 12	0	6.45E - 03	9.52E + 02	7.95E + 01	1.85E + 02	0
F11	M	- 3.79E - 03	- 3.18E - 003	1.26E - 07	7.67E + 05	1.53E - 01	- 4.89E + 03	- 2.75E + 07
	SD	1.47E - 12	8.43E - 004	3.29E - 07	4.01E + 05	1.21E - 01	2.72E + 01	5.65E + 07
F12	M	7.05E - 06	8.63E - 002	9.98E - 01	- 3.79E - 03	4.51E - 01	- 3.79E - 03	- 3.79E - 03
	SD	6.47E - 06	1.02E - 001	2.92E - 16	0	4.12E - 01	4.57E - 19	0
F13	M	0	0	0	6.19E + 01	3.57E + 04	5.11E + 04	1.79E - 08
	SD	0	0	0	3.44E + 01	6.22E + 03	0	1.14E - 09
F14	M	0	0	0	8.40E + 05	1.00E + 06	1.00E + 06	2.08E - 08
	SD	0	0	0	3.23E + 05	8.60E + 01	1.36E + 01	4.70E - 09
F15	M	8.30	8.09	1.74	3.74	3.74	2.61E - 03	3.89E - 05
	SD	4.46E - 01	3.68	2.05E - 01	1.02	2.19	1.05E - 03	2.90E - 05

Unimodal functions are helped in validating the exploitation capability of the stochastic optimization algorithm. Therefore, these functions are optimized by those meta-heuristic algorithms which have better exploitation capability.

In the chosen multimodal functions (from F16 to F36), many local minimum values are associated with these functions and harder to solve than unimodal functions as sometimes solutions of these functions are stuck at local optima and are not escaped. Also, the difficulty level of multimodal functions rises with the number of dimensions, search area, and local optima value. Due to the ability to search new places, these functions test the exploration ability of meta-heuristic algorithms.

## 6.2 Experimental setup

The code of the proposed algorithm is written and implemented using MATLAB R2015a on a computer with an Intel i5 processor, 8 GB of RAM with Windows 2010 operating system. At most, 10,000 iterations are in use as a basis to stop our proposed algorithm. There are different ways to stop the algorithm, such as maximum number of iterations achieved, a fixed error tolerance value, maximum use of CPU time, and maximum number of iterations having zero improvements. Each function was repeated for 30 runs and rounded up two digits after the decimal to produce more minor statistical errors and statistically significant output.

We put down the mean ( $M$ ) and standard deviation ( $SD$ ) of EMFO with other algorithms for collation. One

**Table 2** Comparison of multimodal functions with MFO, SOS, PSO, JAYA, DE, and BOA

Function	EMFO	MFO	SOS	PSO	JAYA	DE	BOA	
F16	M	0	0	0	3.03E - 02	3.04E + 02	0	0
	SD	0	0	0	2.61E - 17	2.69E + 02	0	0
F17	M	<b>8.88E - 16</b>	8.88E - 016	1.00E - 15	1.81E + 01	1.96E + 01	1.71	1.71
	SD	<b>0</b>	0	6.38E - 16	5.48	6.58E - 01	0	0
F18	M	0	0	0	4.05E + 01	1.26E + 02	9.22E - 17	1.84E - 19
	SD	0	0	0	4.13E + 01	3.41E + 01	2.04E - 17	2.68E - 20
F19	M	0	0	0	1.44E + 02	1.19E + 02	1.24E + 01	0
	SD	0	0	0	3.79E + 01	1.26E + 01	3.30	0
F20	M	0	0	1.82E - 01	0	3.50E + 02	0	0
	SD	0	0	1.20E - 01	0	1.24E + 02	0	0
F21	M	1	- 4.11E - 001	8.84E + 01	- 1.00E + 00	- 7.24E - 14	- 1.00	- 1
	SD	2.79E - 05	3.69E - 001	1.22E - 03	0.00E + 00	4.03E - 13	0	0
F22	M	- 4.19E + 02	- 4.05E + 02	7.04E - 25	- 7.57E + 03	- 4.03E + 02	- 4.56E + 12	- 2.26E + 03
	SD	1.37E - 07	3.58E + 01	9.02E - 25	8.88E + 02	3.69E + 01	3.19E + 12	4.56E + 02
F23	M	1.32E - 02	1.53E - 002	1.09E - 117	1.98E - 04	4.50E + 01	6.27E - 04	2.84E - 02
	SD	8.93E - 03	1.23E - 002	3.94E - 117	1.65E - 04	7.40E + 01	1.3E - 03	1.31E - 02
F24	M	2.56E - 02	7.88E - 01	4.71E - 32	2.72E - 01	6.11E + 07	1.11E + 03	2.27E - 01
	SD	1.16E - 02	3.44E - 01	2.23E - 47	4.71E - 01	3.44E + 07	1.84E + 03	8.54E - 02
F25	M	1.06E + 00	6.10	9.98E - 01	1.25	1.10E + 02	9.98E - 01	2.84
	SD	3.62E - 01	3.62	2.92E - 16	5.71E - 01	1.30E + 02	2.50E - 16	2.55
F26	M	3.09E - 04	5.80E - 03	3.27E - 04	6.05E - 04	1.51E - 01	1.37E - 03	6.66E - 04
	SD	6.63E - 07	6.72E - 03	7.60E - 05	4.62E - 04	1.64E - 01	4.72E - 04	7.52E - 04
F27	M	3.98E - 01	5.13E - 01	3.98E - 01	3.98E - 01	1.50	3.98E - 01	4.98E - 01
	SD	1.53E - 05	1.36E - 01	3.17E - 07	5.84E - 09	1.05	4.44E - 16	1.47E - 01
F28	M	- 9.61E	- 7.72	- 9.63	- 6.21	- 6.11E - 01	- 8.02	- 2.54
	SD	3.07E - 01	1.03	1.53	2.17	2.77E - 01	1.88	2.08
F29	M	- 9.86	- 7.59	- 1.01E + 01	- 7.14	- 8.02E - 01	- 9.73	- 4.48
	SD	2.66E - 01	1.13	1.33	2.85	4.01E - 01	5.68E - 01	1.35E + 01
F30	M	- 9.92	- 7.78	- 1.05E + 01	- 6.87	- 8.99E - 01	- 1.03E + 01	- 5.93E + 01
	SD	2.39E - 01	1.20	2.83E - 15	2.57	2.57E - 01	3.12E - 01	3.75E + 02
F31	M	0	0	0	6.93E + 02	7.32E + 03	7.50E + 03	2.56E - 08
	SD	0	0	0	1.97E + 02	1.204E + 02	0	9.87E - 10
F32	M	7.79E - 02	2.19E + 01	0	3.24E + 04	1.95E + 06	2.30E + 06	1.02E + 01
	SD	2.37E - 01	1.19E + 01	0	2.91E + 04	8.82E + 04	2.14E + 01	5.33
F33	M	- 2.21E + 03	- 1.37E + 03	- 2.35E + 03	- 6.29E + 02	1.22E + 03	7.45E + 03	- 1.38E + 03
	SD	3.61E + 01	1.14E + 02	9.59E - 13	1.17E + 02	1.41E + 02	2.21E + 01	5.33E + 01
F34	M	0	0	0	1.57E + 02	6.49E + 02	2.56E + 03	1.62E - 11
	SD	0	0	0	1.86E + 01	3.48E + 01	9.83E - 02	9.74E - 13
F35	M	0	0	0	1.06E + 01	1.9719E + 01	3.34E + 01	2.92E - 08
	SD	0	0	0	7.62E - 01	4.1799E - 01	2.98E - 01	1.29E - 09
F36	M	3.66E - 02	0	9.98E - 02	2.81E + 01 1.35	4.2035E + 01	5.25E + 01	9.99E - 02
	SD	4.90E - 02	0	1.99E - 13		5.2080E - 01	4.53E - 01	1.37E - 07

particular union of variables was used for EMFO in the copy of both unimodal and multimodal benchmark functions to fulfill this criterion. The powers exponent constant

b is equal to 1 and  $t$  varies from - 1 to 1, and the population size is 50.

**Table 3** Performance assessment of EMFO compared to MFO, SOS, PSO, DE, BOA, and JAYA on 36 benchmark functions

Proposed algorithm	MFO	SOS	PSO	DE	BOA	JAYA
Superior to	16	20	27	26	21	33
Similar to	15	10	2	3	9	0
Inferior to	5	6	7	7	6	3

### 6.3 Experimental results of benchmark functions

In this subsection, the obtained simulation result of our proposed EMFO has been compared with other six meta-heuristics MFO, SOS, PSO, DE, JAYA, and BOA, on 36 benchmark functions including both unimodal and multi-modal functions and are presented in subsection 6.3.1 and

**Table 4** Comparison of EMFO with ABC, GA, FA, MBO, and CS

Function		EMFO	ABC	GA	FA	MBO	CS
F1	M	0	2.58E - 01	2.20E - 02	7.86E - 04	2.21E - 01	6.49E - 61
	SD	0	5.13E - 01	9.67E - 03	1.36E - 04	1.41E - 01	9.77E - 61
F2	M	0	8.43E - 03	2.19E - 04	1.08E - 05	3.40E - 04	5.53E - 63
	SD	0	2.30E - 02	1.09E - 04	1.45E - 06	4.78E - 04	8.92E - 63
F3	M	0	2.23E - 01	2.30E - 02	0	3.17E + 04	1.21
	SD	0	4.82E - 01	1.05E - 02	0	2.57E + 04	1.48
F4	M	2.69E + 01	1.25E + 02	3.78E + 01	1.83E + 01	<b>7.14</b>	3.95E + 01
	SD	1.28	8.77E + 01	2.52E + 01	4.24	<b>1.42E + 01</b>	3.33E + 01
F5	M	0	2.36E - 02	2.42E - 03	5.24E - 02	1.11E + 05	4.05E - 02
	SD	0	4.18E - 02	1.07E - 03	6.70E - 02	1.98E + 04	6.06E - 02
F6	M	<b>2.03E - 197</b>	1.00E - 01	4.66E - 02	1.43E - 04	3.54	3.67E - 03
	SD	<b>0</b>	6.60E - 01	1.08E - 02	1.48E - 05	3.99	1.59E - 03
F7	M	0	1.42E - 01	5.43E - 02	1.50E - 03	7.02E - 03	1.12E + 01
	SD	0	6.83E - 02	1.61E - 02	1.13E - 04	2.96E - 03	2.49E + 01
F8	M	1.59E - 05	1.32E - 11	0	0	8.88E - 03	1.81E - 05
	SD	2.23E - 05	2.42E - 11	0	0	1.51E - 02	1.40E - 05
F9	M	0	1.19E - 04	8.38E - 104	0	0	1.33E - 06
	SD	0	1.30E - 04	2.51E - 103	0	0	2.16E - 06
F10	M	<b>1.31E - 12</b>	1.51E + 01	3.27E - 02	2.39E - 04	3.43E + 03	6.63
	SD	<b>4.94E - 12</b>	1.10E + 01	1.11E - 02	4.76E - 05	2.63E + 03	6.99
F11	M	- 3.79E - 03	- 3.79E - 03	- 3.79E - 03	- 3.79E - 03	2.01E - 01	- 3.79E - 03
	SD	1.47E - 12	5.71E - 18	4.57E - 19	4.57E - 19	2.23E - 01	4.45E - 07
F12	M	0	0	0	0	0	4.92E - 04
	SD	0	0	0	0	0	8.66E - 04
F13	M	<b>8.88E - 16</b>	3.65E	4.66E - 02	7.73E - 04	1.31E + 01	1.70E + 01
	SD	<b>0.00E + 00</b>	6.93E - 01	1.68E - 02	4.80E - 05	9.41	5.46
F14	M	0	5.48E - 01	5.55E - 02	1.07E - 07	4.30E + 02	2.01E + 01
	SD	0	2.72E - 01	2.94E - 02	1.31E - 08	1.65E + 02	4.30E - 02
F15	M	0	2.36E + 01	1.64E + 01	3.47E + 01	6.37E + 01	1.08E + 02
	SD	0	5.48	4 .10	9.54	4.29E + 01	4.99E + 01
F16	M	0	8.62E - 16	1.69E - 17	1.43E - 02	6.68E - 04	9.81E - 01
	SD	0	8.54E - 16	1.68E - 16	5.35E - 03	8.09E - 04	7.87E - 01
F17	M	1	- 9.00E - 01	- 1	- 8.50E - 01	- 1.00	- 1.00
	SD	2.79E - 05	2.73E - 01	0	3.58E - 01	0	0
F18	M	- 4.19E + 02	1.05E + 04	- 1.18E + 04	- 7.14E + 03	- 3.94E + 04	- 1.08E + 02
	SD	1.37E - 07	2.93E + 02	2.95E + 02	9.09E + 02	5.70E + 04	1.08E + 01
F19	M	1.32E - 02	7.13E - 02	1.29E - 04	1.29E - 04	5.02E + 01	1.40E - 02
	SD	8.93E - 03	5.13E - 02	1.13E - 04	1.56E - 04	4.07E + 01	1.45E - 02
F20	M	<b>7.05E - 06</b>	8.01E - 03	9.56E - 06	4.82E - 05	3.26E - 01	1.77E - 04
	SD	<b>6.47E - 06</b>	9.61E - 03	3.02E - 05	6.47E - 05	1.40E - 01	1.55E - 04

**Table 5** Statistical analysis (Friedman rank test)

Algorithm	Mean rank	Rank
EMFO	2.57	1
SOS	3.15	2
MFO	3.35	3
BOA	3.40	4
DE	4.43	5
PSO	4.57	6
JAYA	6.53	7

6.3.2, respectively. Also, in subsection 5.3.3, it has been compared with other five popular and latest algorithm namely ABC, GA, FA, MBO, and CS.

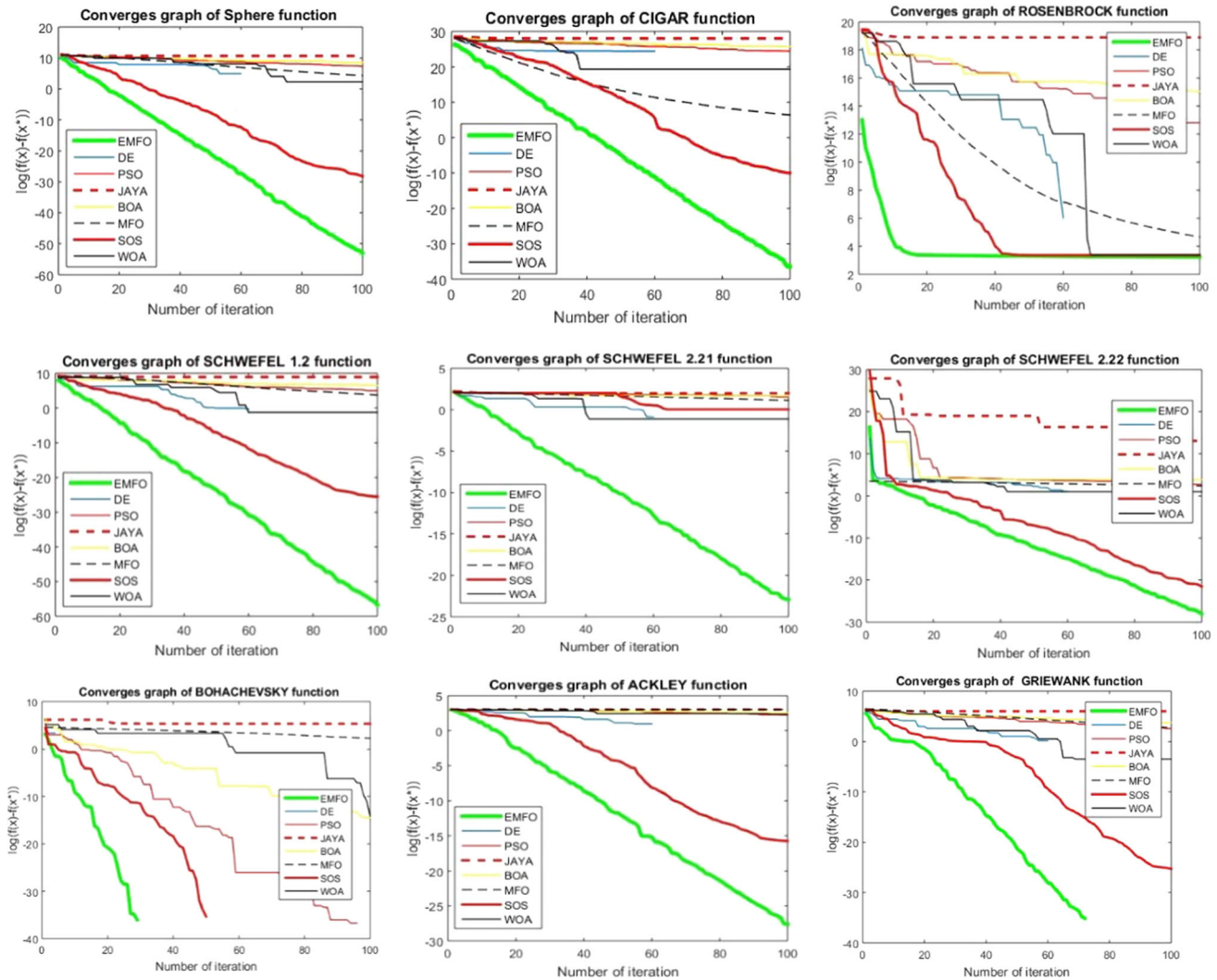
**6.3.1 Discussion on unimodal functions**

The means and the standard deviations for optimized unimodal functions along with EMFO and other six algorithms

are provided in Table 1. It is evident from the table that EMFO offered the least values compared to different algorithms. EMFO algorithm provides the best results for F1, F2, F3, F5, F7, F8, F9, F11, F13, and F14. It provides the second best results for functions F6 and F10 and inferior results for F4, F12, and F15. Therefore, our proposed approach can be assumed to be a superior algorithm to others.

**6.3.2 Discussion on multimodal functions**

In Table 2, functions F16 to F36 have been investigated under multimodal function optimization. It can be clear that for functions F16, F17, F18, F19, F20, F26, F27, F29, F30, F31, F32, F34, and F35, EMFO possess superior result than other algorithms. For functions F23, F28, and F36, EMFO offers the second highest results and for rest five functions it is inferior than other algorithms. Hence, it can be concluded that EMFO is a superior algorithm as



**Fig. 4** Convergence graph of benchmark functions for DE, PSO, JAYA, BOA, WOA, SOS, MFO, and EMFO

**Table 6** Comparison of EMFO with other variants of MFO algorithm

Sl. No		OMFO	LMFO	MMFO3	WCMFO	ODSMFO	WEMFO	EMFO
F1	M	1.39e - 77	8.93e - 03	5.64e - 66	1.29e - 40	1.09e + 04	1.51e - 142	0
	SD	7.62e - 77	2.01e - 03	3.09e - 65	2.27e - 40	2.88e + 03	8.27e - 142	0
F2	M	2.85e - 45	2.59e + 05	3.47e - 72	5.72e - 33	1.69e + 12	6.18e - 146	0
	SD	1.56e - 44	7.17e + 04	1.87e - 71	5.10e - 33	1.46e + 11	2.93e - 145	0
F3	M	0	0	0	3.05e - 01	5.79e + 04	0	0
	SD	0	0	0	0	5.67e + 03	0	0
F4	M	3.31e + 02	8.68e + 01	2.40e + 02	0	3.33e + 02	2.53e - 14	2.69E + 01 1.28
	SD	3.54e + 02	1.15e + 02	1.83e + 02	0	4.53e + 02	1.16e - 13	
F5	M	8.88e - 16	8.62e - 02	8.88e - 16	6.90	2.04e + 01	8.88e - 16	8.88E - 16
	SD	0	1.00e - 02	0	2.93e - 14	1.87e - 01	0	0
F6	M	0	4.40e - 04	0	1.4930e - 02	5.25e + 02	0	2.03E - 197
	SD	0	9.64e - 05	0	0	5.47e + 01	0	0
F7	M	0	1.91e + 00	0	3.9235e + 00	4.08e + 02	0	0
	SD	0	4.21e - 01	0	1.70e - 13	2.28e + 01	0	0
F8	M	2.63e + 02	1.78e + 02	2.25e + 02	3.30e - 10	2.39e + 02	1.04e - 04	1.59E - 05 2.23E - 05
	SD	9.70e + 01	7.62e + 01	1.03e + 02	5.57e - 10	8.42e + 01	2.47e - 04	
F9	M	2.89e + 01	2.97e + 01	2.89e + 01	3.67e + 00	1.94e + 08	2.88e + 01	0
	SD	6.56e - 02	1.98e - 01	5.25e - 02	1.47e + 01	3.69e + 07	3.68e - 02	0
F10	M	- 2.94e - 01	- 2.44e - 02	- 3.09e - 04	- 1.36e - 03	- 1.33e - 02	- 6.91e - 01	- 3.79E - 03
	SD	3.64e - 01	9.83e - 02	1.69e - 03	7.29e - 02	7.29e - 02	2.73e - 01	1.47E - 12
F11	M	6.48e - 47	1.25e - 01	1.49e - 83	1.11e - 20	8.36e + 03	2.52e - 149	1.31E - 12
	SD	3.39e - 46	2.37e - 02	8.17e - 83	1.58e - 22	9.92e + 02	1.36e - 148	4.94E - 12
F12	M	5.14e - 28	4.20e - 02	9.11e - 34	7.77e - 01	8.19e + 00	1.07e - 78	0
	SD	2.82e - 27	5.81e - 03	4.99e - 33	5.48e + 00	5.34e - 01	4.21e - 78	0
F13	M	1.24e - 35	4.04e - 01	3.47e - 42	2.12e - 10	4.27e + 11	2.62e - 75	0
	SD	6.56e - 35	3.51e - 02	1.90e - 41	5.80e - 12	1.65e + 12	1.43e - 74	0
F14	M	- 3.91e + 02	- 4.17e + 02	- 3.88e + 02	4.55e - 10	- 1.77e + 02	- 4.19e + 02	- 4.19E + 02
	SD	6.19e + 01	8.99e + 00	6.56e + 01	4.18e + 02	1.93e + 02	1.51e + 00	1.37E - 07
F15	M	1.05e + 01	8.77e - 01	3.70e + 00	4.12e - 05	3.25e + 00	3.06e - 01	0
	SD	1.01e + 01	1.23e + 00	4.75e + 00	2.36e - 07	3.16e + 00	3.08e - 01	0
F16	M	2.26e - 64	2.72e - 02	1.56e - 01	1.11e - 25	1.45e - 01	4.65e - 20	0
	SD	1.24e - 63	3.27e - 02	1.54e - 01	1.55e - 29	1.64e - 01	1.38e - 19	0
F17	M	6.70e - 78	2.80e - 05	2.58e - 78	3.98e - 07	6.89e + 01	2.90e - 161	1
	SD	3.67e - 77	2.31e - 05	1.41e - 77	2.56e - 10	5.10e + 01	1.59e - 160	2.79E - 05
F18	M	9.68e - 02	9.73e - 02	1.22e + 01	7.47e + 00	1.60e + 01	5.23e - 01	0
	SD	6.56e - 02	7.63e - 02	1.66e + 01	1.35e - 02	2.69e + 01	3.71e - 01	0
F19	M	1.24e - 01	6.55e - 03	1.18e - 01	3.48e - 10	8.79e - 02	3.79e - 01	1.32E - 02
	SD	1.25e - 01	1.74e - 02	1.59e - 01	3.79e - 03	1.03e - 01	5.29e - 06	8.93E - 03
F20	M	1.09e - 01	8.15e - 03	3.65e - 01	6.47e - 07	1.95e - 01	7.59e - 02	7.05E - 06
	SD	1.31e - 01	1.55e - 02	3.63e - 01	3.17e - 10	2.04e - 01	8.32e - 02	6.47E - 06

compared to the six other algorithms in respect of the optimization of multimodal functions.

The number of cases where EMFO's mean performance is better than, similar to, and worse than the other six algorithms is shown in Table 3. From Table 3, we noticed

that EMFO works better than MFO, SOS, PSO, DE, BOA, and JAYA in 16, 20, 27, 26, 21, and 33 benchmark functions, respectively, similar results can be seen in 15, 10, 2, 3, 9, and 0 occasions, respectively, and worse values are achieved in 5, 6, 7, 6, and 3 benchmark functions,

**Table 7** Performance assessment of EMFO compared to MFO, SOS, PSO, DE, BOA, and JAYA on 20 benchmark functions

Proposed algorithm	OMFO	LMFO	MMFO3	WCMFO	ODSMFO	WEMFO
Superior to	15	17	14	14	20	12
Similar to	3	1	3	0	0	3
Inferior to	2	2	3	6	0	5

**Table 8** Friedman rank test on variants of MFO algorithm

Algorithm	Mean rank	Rank	<i>P</i> -value
OMFO	3.93	3	<i>P</i> -value (0.000 < 0.01) indicates that Ho is rejected at 1% level of significance, i.e., there is a significant difference between the performance of different methods at a 1% level of significance
LMFO	4.55	6	
MMFO3	4.18	5	
WCMFO	4.13	4	
ODSMFO	6.10	7	
WEMFO	2.83	2	
EMFO	<b>2.30</b>	<b>1</b>	

respectively. The mathematical formulation of the 36 benchmark functions with dimension, range of the variables, and optimum value are shown in Appendix 1.

**6.3.3 Discussion on other evolutionary algorithms**

The simulation results including mean and standard deviations of EMFO, ABC, GA, FA, MBO, and CS are presented in Table 4 for validating the performance of EMFO algorithm over 20 benchmark functions containing both unimodal (F1 to F12) and multimodal functions (F16 to F23). These benchmark functions are taken from Appendix 1. Except EMFO all results of other algorithms are taken from the reference (Kigsirisin and Miyauchi 2021). From Table 4, it is clear that for functions F1, F2, F3, F5, F6, F7, F9, F10, F12, F13, F14, F16, F17, and F21, EMFO has the best results as compared to other algorithms and for rest six functions EMFO offers second, third, and fourth best results. Therefore, it can be concluded that EMFO is a competitive algorithm as compared to other meta-heuristic algorithms.

**6.4 Friedman rank test**

Milton (1937) developed a variable (non-parametric) statistical test called the Friedman test. It is used for the treatment of differences across numerous trials of the test. It ranks each row (or block) together and considers the value by columns. In this paper, for each benchmark function, the Friedman test is used from the average performance of algorithms. We use IBM-SPSS software for finding the middle rank. From Table 5, it is concluded that

rank of EMFO can be seen that the rank of EMFO is least which indicates it is best among the rest methods.

**6.5 Convergence analysis**

To collate the convergence rate of EMFO with other algorithms, some of the convergence graphs compared with DE, PSO, JAYA, BOA, WOA, SOS, and MFO for few benchmark functions such as F1 (Sphere), F2 (Cigar function), F4 (Rosenbrock function), F5 (Schwefel1.2 function), F6 (Schwefel2.21 function), F7 (Schwefel2.22 function), F16 (Bohachevsky function), F17 (Ackley function), and F18 (Griewank Function) are presented in Fig. 4. In these figures, both the number of iterations and fitness evaluations are presented in horizontal and vertical axis, respectively, and the curves were plotted with dimensional size 100. EMFO has rapid convergence as compared to the other methods concerning the worldwide optima. On the other hand, the convergence rate of other compared optimization algorithms is moderate due to being stuck in neighborhood optima. For F4, converging to the global optima is tough for optimization problems as it is placed in a small canyon. However, the proposed EMFO achieves this global optimum within hundred dimensions. Therefore, the proposed EMFO reveals its high convergence rate equipped with other optimization methods.

**6.6 Comparison with some of the variants of MFO algorithm**

In this subsection, comparison evaluation has been done in with six MFO variants such as LMFO (Li et al. 2016b),

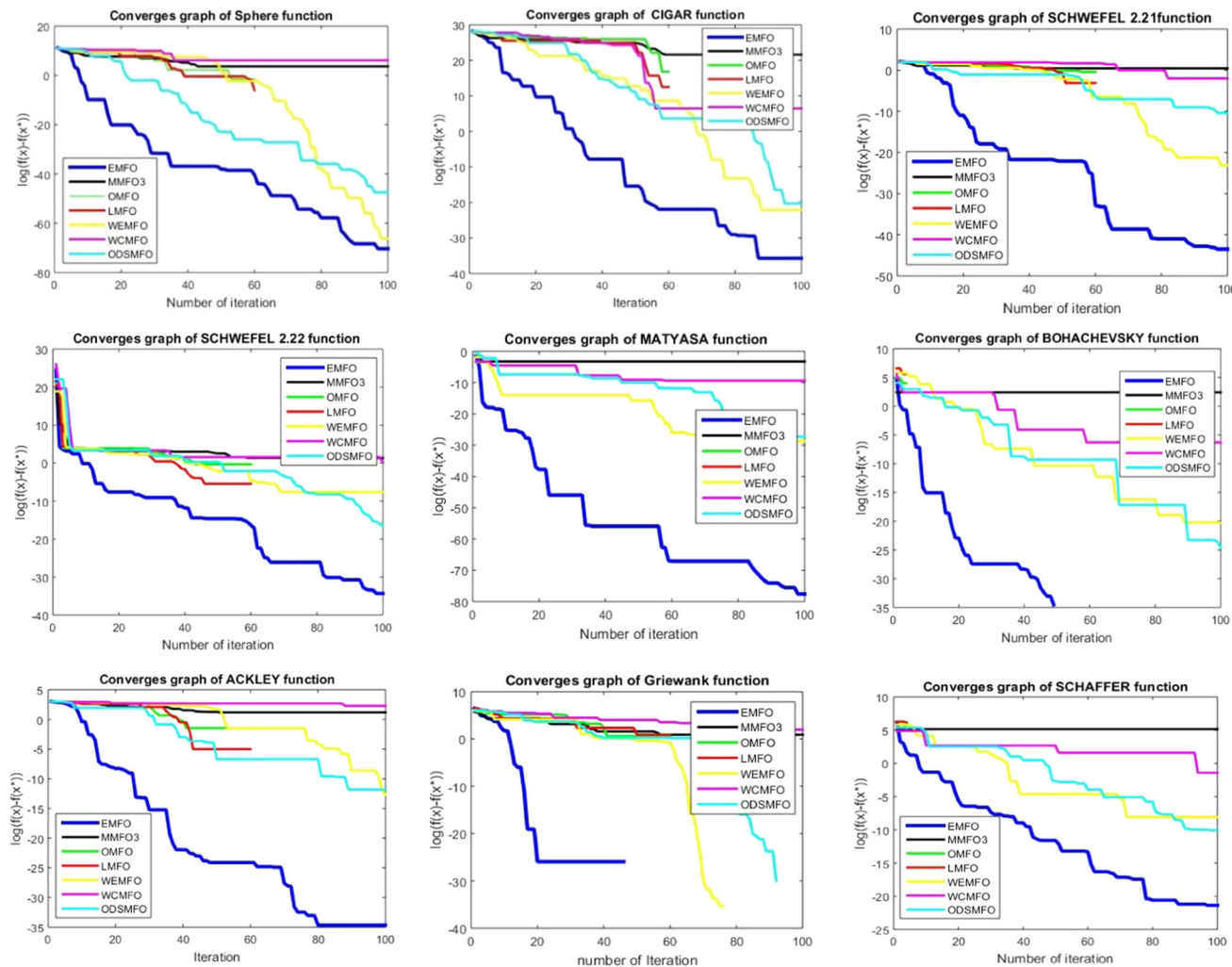


Fig. 5 Convergence graph of benchmark functions for variants of MFO and proposed EMFO

Table 9 Comparison performance of EMFO with MFO, DE, gravitation search algorithm (GSA), and DE-GSA for gas production problem

Item	DE	GSA	DE-GSA	MFO	EMFO
$x_1$	17.5	17.5	17.5	17.5	17.5
$x_2$	600	600	600	600	600
$f(x)$	169.844	169.844	169.844	71.4495	<b>71.4468</b>

OMFO (Apinantanakon and Sunat 2017), MFO3 (Soliman et al. 2016), WCMFO (Khalilpourazari and Khalilpourazary 2019), ODSMFO (Li et al. 2021), and WEMFO (Shan et al. 2021). The simulation outcomes of EMFO together with six MFO variants for 20 benchmark functions including both unimodal (F1 to F12) and multimodal functions (F16 to F23) are presented in Table 6. These benchmark functions are taken from Appendix 1. All the results are evaluated using Matlab 2015(a). The mean

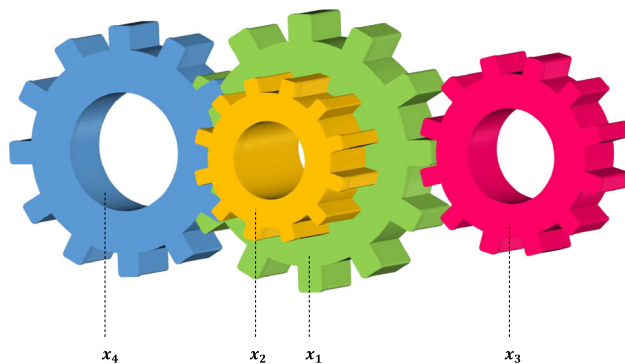


Fig. 6 Gear train design problem

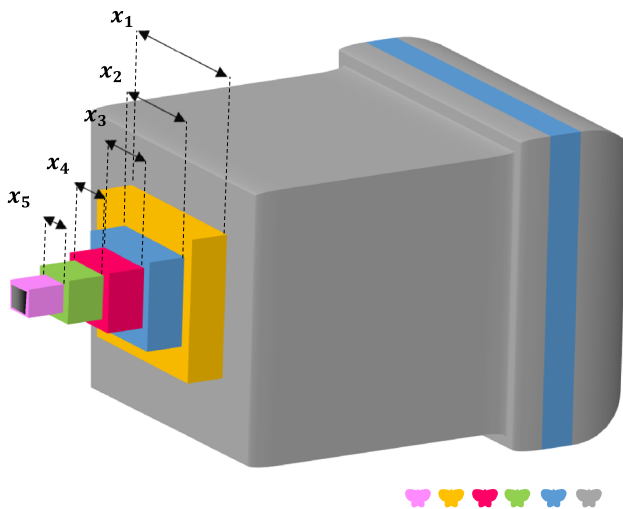
(M) and standard deviation of EMFO with other variants of the MFO algorithm are presented in Table 6.

From Table 6, it can be observed that our proposed EMFO algorithm achieved more than 85% best results for all groups of benchmark problems as compared to the



**Table 10** Comparison results of EMFO with different algorithms on gear train design problem

Algorithm	$x_1$	$x_2$	$x_3$	$x_4$	$f(\vec{x})$
EMFO	49	19	16	43	2.7008E - 12
BA	57.4517	19.48176	18.58749	43.68715	1.5310E - 11
DA	52.44017	17.00267	22.99426	51.67159	3.0210e - 11
FPA	51.15054	22.45769	17.97921	55.90958	4.8310e - 11
ACO	51.41824	21.35921	15.83209	47.34128	2.8710e - 05
PSO	51.32931	21.02347	14.79391	47.82131	3.0810e - 04
SA	51.34512	21.32743	14.98321	47.43271	1.7110e - 04
GA	52.57125	23.19834	16.93254	48.24512	1.1310e - 04
SA	51.34512	21.32743	14.98321	47.43271	1.7110e - 04
WWO	55.27473	24.3159	15.03104	45.82914	5.6710e - 12



**Fig. 7** Cantilever beam design

variants of MFO algorithms but it provides seventy percent best results when compared with WCMFO and WEMFO algorithm. Also, the number of occasions of superiority, similarity, and inferiority is presented in Table 7. From Table 7, we noticed that EMFO works better than OMFO, LMFO, MMFO3, WCMFO, ODSMFO, and WEMFO in 15, 17, 14, 14, 20, and 12 benchmark functions, respectively, similar results can be seen in 3, 1, 3, 0, 0, and 3 occasions, respectively, and worse values are achieved in 2, 2, 3, 6, 0, and 5 benchmark functions, respectively.

**Table 11** Comparison results of EMFO with different algorithms on cantilever beam design problem

Algorithm	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$f(\vec{x})$
EMFO	<b>6.33214</b>	<b>4.91592</b>	<b>4.65422</b>	<b>3.03527</b>	<b>1.81594</b>	<b>1.29169</b>
GCA-II	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
MMA	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA-I	6.0100	5.30400	4.4900	3.4980	2.1500	1.3400
SOS	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
CS	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
ALO	6.01812	5.31142	4.48836	3.49751	2.158329	1.33995

**6.6.1 Statistical analysis**

Friedman is used to analyze the performance of proposed EMFO algorithm. In this paper, for each benchmark function Friedman test is used from the average performance of algorithms. We use IBM-SPSS software for finding the average rank. The outcomes of the Friedman rank test between EMFO with OMFO, LMFO, MMFO3, WCMFO, ODSMFO, and WEMFO for 20 benchmark functions is shown in Table 8. From Table 8, it is clearly visible that EMFO obtains least rank among other algorithms at 1% relevant.

**6.6.2 Convergence analysis**

To collate the convergence rate of EMFO with other algorithms, some of the convergence graphs compared with OMFO, LMFO, MMFO3, WCMFO, ODSMFO, and WEMFO for few benchmark functions such as F1 (Sphere), F2 (Cigar function), F6 (Schwefel 2.21 function), F7 (Schwefel2.22 function), F9 (Matyasa function), F16 (Bohachevsky function), F17 (Ackley function), F18 (Griewank Function), and F20 (Schaffer function) are presented in Fig. 5. In these figures, both the iteration and objective function value are presented in horizontal and vertical axes, respectively, and the curves were plotted with dimensional size 100. EMFO has rapid convergence as compared to the other methods concerning the worldwide optima. On the

Fig. 8 I-beam deflection

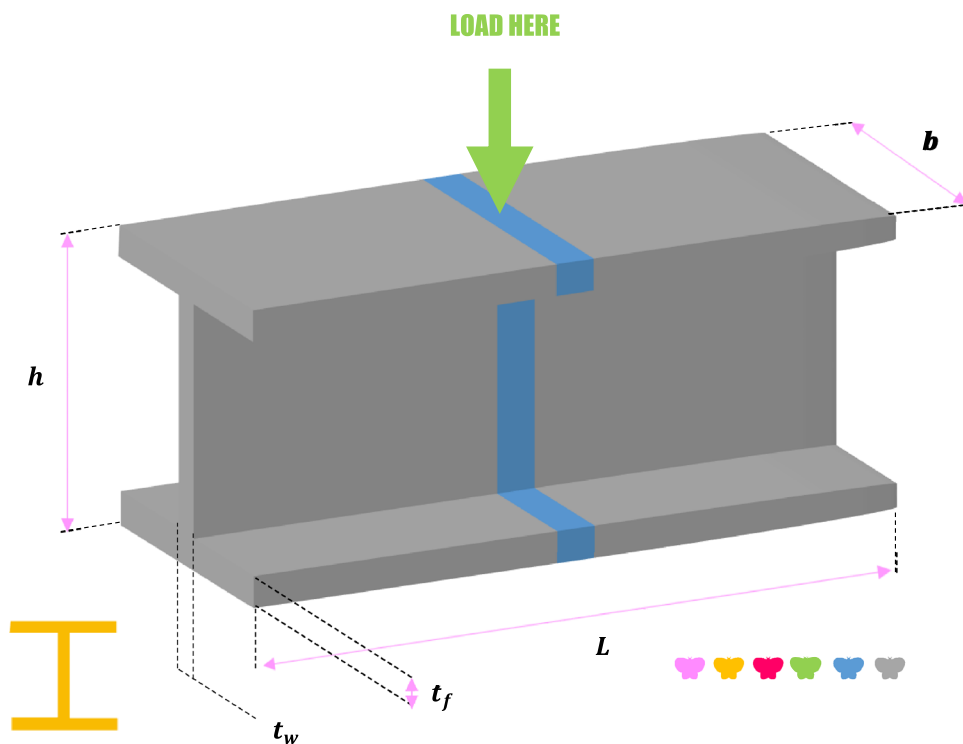


Table 12 Comparison of EMFO with ARSM, Improved ARSM, SOS, MFO, SSA, and SAISOS on I-beam vertical deflection problem

Algorithms	$h$ (cm)	$b$ (cm)	$t_w$ (cm)	$t_f$ (cm)	$g_1$	$g_2$	Opt. value	Max. eval
ARSM	80	37.05	1.71	2.31	N/A	N/A	0.0157	
Improved ARSM	79.99	48.72	0.90	2.40	N/A	N/A	0.0131	
CS	80	50	0.90	0.013	N/A	N/A	0.0130747	
SOS	80	50	0.90	2.3217	N/A	N/A	0.0130741	
MFO	80	50	1.7647	5.0000	N/A	N/A	0.0066259	
SSA	80	50	1.7647058	5.0000				
SaISOS	15.860770	49.999542	2.7882569	4.9995440	193.65761	15.89813	0.00197	10,000
EMFO	14.159935	36.035722	4.9347589	3.193023750	N/A	N/A	0.00196	10,000

other hand, the convergence rate of other compared optimization algorithms is moderate due to being stuck in neighborhood optima. However, the proposed EMFO achieves this global optimum within hundred dimensions. Therefore, the proposed EMFO reveals its high convergence rate as compared to other variants of MFO.

### 7 Description of real-world problems

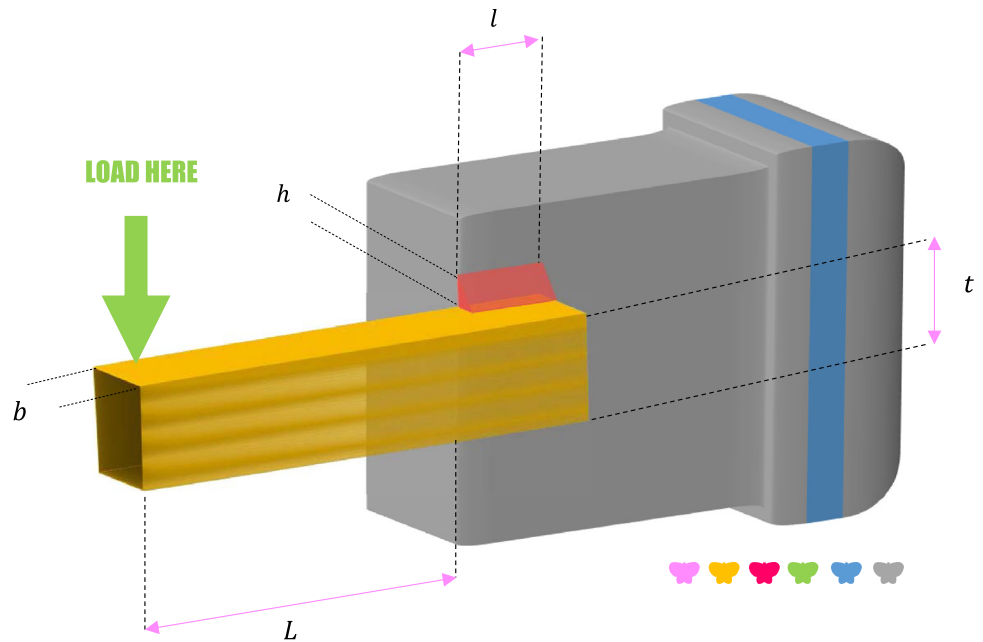
Seven real-world problems (RWP) such as optimal capacity of gas production problem, gear train design problem, cantilever beam design problem, minimize I-beam vertical deflection problem, welded beam design

problem, three-bar truss design problem, and car-side crash design problem have been considered and solved by our proposed EMFO algorithm and discussed in following section. Appendix 2 shows the mathematical formulation of all the above problem.

#### 7.1 RWP-1: optimal capacity of gas production problem

This problem is taken from Gandomi (2014). To measure the efficiency, EMFO has been tested by solving gas production problem. The experimental result of this problem is presented in Table 9 and compared with other algorithms, namely DE, GSA, and hybrid DE-GSA. It is clear that, the

**Fig. 9** Welded beam design problem



efficiency of our proposed work superior than other algorithms which is indicated by bold face in Table 9.

**7.2 RWP-2: gear train design problem**

It belongs to the branch of mechanical engineering and its target is to minimize the gear ratio for a set of four train gears (Połap 2017). It has four variables and has no constraint. The variables range are taken as constraints. The layout of the above problem is presented in Fig. 6.

The efficiency of EMFO has been tested by solving the above problem and the obtained result compared with other eight meta-heuristic algorithms which is taken from Nama

et al. 2020b and presented in Table 10. Table 10 confirms that EMFO algorithm produces superior results compared to the other algorithms.

**7.3 RWP-3: cantilever beam design problem**

It comprises five hollow elements with square-shaped cross section. The beam is constant in thickness and each element has a single variable, so that five structural parameters are linked to it, as shown in Fig. 7. Figure 7 usually requires a vertical load on the free end of a beam (node 6) and a stable beam side (node 1) support. There is also one constraint of vertical displacement that does not violate the

**Table 13** Comparison results of EMFO with different algorithms

Researchers	Algorithm	$h$	$l$	$t$	$b$	Cost	Function evaluation
Bernardino et al	AIS-GA	0.2444	6.2183	8.2912	0.2444	2.3812	320,000
Lemonge and Barbosa	GA	0.2443	6.2117	8.3015	0.2443	2.3816	320,000
Deb	GA	0.2489	6.1730	8.1789	0.2533	2.4331	320,080
Parsopoulos and Vrahatis	PSO	NA	NA	NA	NA	1.9220	100,000
Hedar and Fukushima	SA-DS	0.2444	6.2158	8.2939	0.2444	2.3811	56,243
Liu	SA	0.2444	6.2175	8.2915	0.2444	2.3810	NA
Atiqullah and Rao	SA	0.2471	6.1451	8.2721	0.2495	2.4148	NA
Zhang et al.	DE	0.2444	6.2175	8.2915	0.2444	2.3810	24,000
Lee and Geem	HS	0.2442	6.2231	8.2915	0.2443	2.381	110,000
Coello	EA	NA	NA	NA	NA	1.8245	NA
Leite and Topping	GA	0.2489	6.1097	8.2484	0.2485	2.4000	6273
Hwang and He	SA-GA	0.2231	1.5815	12.8468	0.2245	2.2500	26,466
Present study	EMFO	0.1623	4.6717	9.0893	0.2039	<b>1.8011</b>	50,000

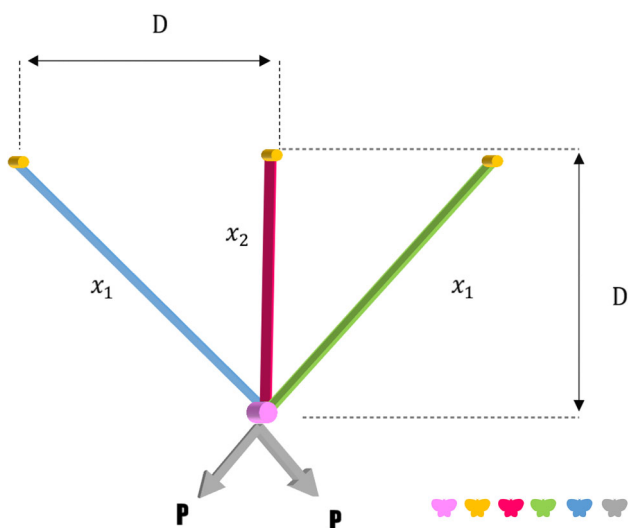


Fig. 10 Three-bar truss design problem

final optimal setup. The efficiency of EMFO has been tested by EMFO and compared with the other methods found in the literature (Kigsirisin and Miyauchi 2021), namely GCA-II, MMA, GCA-I, SOS, CS, and ALO. In Table 11, the bold face indicates that EMFO shows better performance for this problem.

**7.4 RWP-4: minimize I-beam vertical deflection problem:**

Out of the various structural problems, one of the important problems is I-beam vertical deflection. To minimize the vertical deflection of an I-beam is the main objective of this problem also stress constraints and cross-sectional area simultaneously satisfied under given loads. The figure of the problem is shown in Fig. 8.

In Table 12, the optimum value obtained by MFO, SOS, SSA, ARSM, improved ARSM, SaISOS, CS, and EMFO for I-beam vertical deflection problem are presented and except EMFO, all the results are taken from Gandomi et al. (2011). It can be seen that EMFO outperforms than other algorithms. Therefore, we can conclude that EMFO has

better efficiency, searching capability, and robustness in terms of solving real-world problems.

**7.5 RWP-5: welded beam design problem**

The problem of WB design is an important problem among all structural design issues, and it has been solved by different researchers. Fig. 9 shows the WB’s schematic figure. From Fig. 9 and Appendix 2, it is clear that the beam has seven constraints and four variables. The slogan of this issue is to maximize WB’s total cost with respect to the constraints of bending stress, shear stress, end deflection, and overhang load, respectively (P).

The optimization results of EMFO with other algorithms found in the literature are shown in Table 13. Our proposed EMFO algorithm found global optimum of 1.8011 within 50,000 function evaluations, i.e., it requires 1000 iterations. In Table 13 except EMFO all the other results are taken from Sapre and Mini 2021b and from Table 13, the bold face indicates that the obtained result of EMFO is superior to other algorithms found by the researchers.

**7.6 RWP-6: three-bar truss design problem**

Three-bar truss design problem is a structural optimization problem in civil engineering field. This problem is used due to its complex constrained search space (Gandomi et al. 2013; Sadollah et al. 2013). In addition to achieve minimum weight, two parameters of this design problem have been manipulated with respect to the constraints, namely buckling, stress, and deflection. The mathematical formulation and various components of the three-bar truss design problem are presented in Appendix 2 and Fig. 10, respectively.

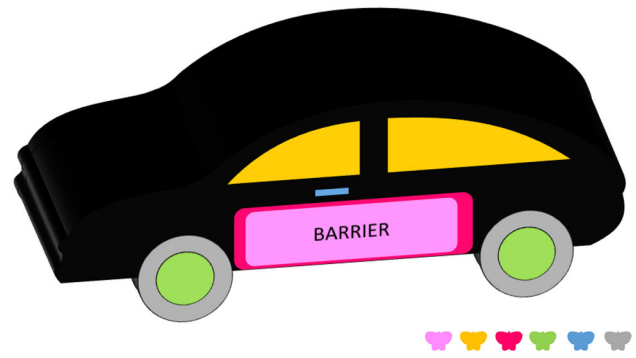
This design problem is solved by using our proposed EMFO algorithm and compared with DEDS, MBA, Tsa, PSO-DE and CS algorithms in the literature (Mirjalili 2015) and the compared results including optimal variable and optimal weights are shown in Table 14. From Table 14, it can be concluded that our proposed EMFO algorithm is superior as compared to other three algorithms.

Table 14 Comparison performance of EMFO with other algorithms for three-bar truss problem

Algorithm	Optimal variables		Optimal weight
	$x_1$	$x_2$	
EMFO	<b>0.398482921061734</b>	<b>0.311902386277691</b>	<b>174.2786484250945</b>
MFO	0.788244770931922	0.788244770931922	263.895979682
DEDS	0.78867513	0.40824828	263.8958434
MBA	0.7885650	0.4085597	263.8958522
Tsa	0.788	0.408	263.68
PSO-DE	0.7886751	0.4082482	263.8958433
CS	0.78867	0.40902	263.9716

**Table 15** Comparison performance of EMFO with other algorithms for car-side crash problem

Variables	Algorithm										
	EMFO	MFO	ABC	PSO	ALO	ER-WCA	GWO	WCA	MBA	SSA	WOA
$x_1$	0.533252	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000	0.500000
$x_2$	1.007717	1.116539	1.0624205	1.1165737	1.115960	1.118688	1.111484	1.1155932	1.1172701	1.1093195	1.108001
$x_3$	0.689264	0.500000	0.5148211	0.5000000	0.500000	0.500000	0.500000	0.5000000	0.5000030	0.5000000	0.534477
$x_4$	0.567134	1.301908	1.4491503	1.3018547	1.302860	1.298407	1.312203	1.3034919	1.3008438	1.3148010	1.305770
$x_5$	0.708759	0.500000	0.5000000	0.5000000	0.5000000	0.5000000	0.501214	0.5000146	0.5000000	0.5000000	0.500000
$x_6$	1.320214	1.500000	1.500000	1.5000000	1.5000000	1.500000	1.500000	1.5000000	1.4999867	1.4999998	1.473844
$x_7$	0.500000	0.500000	0.5000000	0.5000000	0.500000	0.500000	0.500034	0.5000000	0.5000000	0.5000000	0.500000
$x_8$	0.307892	0.345000	0.3450000	0.3450000	0.345000	0.345000	0.345000	0.3450000	0.3450000	0.3450000	0.345000
$x_9$	0.255958	0.345000	0.1920000	0.3450000	0.192000	0.192000	0.192000	0.1920000	0.3450000	0.1920000	0.192000
$x_{10}$	- 29.801677	- 19.5304	- 29.34755	- 19.52470	- 19.6330	- 19.1461	- 20.6057	- 19.69967	- 19.40045	- 20.821793	- 19.69924
$x_{11}$	- 1.323665	- 0.000006	0.7410998	- 0.019297	0.023649	- 0.01527	- 0.25531	- 0.023854	- 0.379205	0.4412962	3.4816923
$f_{min}$	21.592501	22.84297	23.175889	22.842984	22.84298	22.84298	22.85279	22.843596	22.846514	23.042162	23.042162



**Fig. 11** Car-side Image

**7.7 RWP-7: car-side impact design problem**

This problem was initially proposed by Gu and Wang (2016). The car is exposed to a side impact on the foundation of the European Enhanced Vehicle Safety Committee (EEVC) procedures. The objective is to minimize the car’s total weight using eleven mixed variables while maintaining safety performance according to the standard. These variables represent the thickness and material of critical parts of the vehicle. The 8th and 9th variables are discrete, and these are material design variables, while the rest are continuous and represent thickness design variables.

The symbols are used here to represent the variables thickness of B-pillar inner, the thickness of B-pillar reinforcement, the thickness of floor side inner, the thickness of cross members, the thickness of door beam, the thickness of door beltline reinforcement, consistency of roof rail, the material of B-pillar inner, the material of floor side inner, barrier height, and barrier hitting position, respectively. The problem is subjected to ten inequality constraints. The car-side impact design is considered a real case of a mechanical optimization problem with mixed discrete and continuous design variables.

This problem is solved by the proposed EMFO algorithm and compared with the other algorithms such as ABC, PSO, MFO, ALO, ER-WCA, GWO, WCA, MBA, SSA, and WOA found in the literature (Yildiz et al. 2020). The compared results are presented in Table 15. From Table 15, it can be concluded that the proposed EMFO has achieved the superior best solution compared to the other algorithms. The mathematical formulation and schematic diagram are presented in Appendix 2 and Fig. 11, respectively.

### 8 Conclusion

In order to the above discussions, we can conclude that the intensification ability of EMFO is remarkable as can be observed from the results of unimodal functions optimization. It provides the satisfactory results for 90% of the benchmark functions. The diversification ability of EMFO is also impressive as can be observed from its better performance than other meta-heuristic algorithms. The EMFO can handle better trade-offs between global search and local search. Thus, it has vast scope for modification and future work.

This paper acquaints with the upgraded version of MFO called EMFO, which improves the MFO algorithm by the help of the interdependency step. To examine the performance of EMFO, several demonstrations have applied to the benchmark functions. The obtained results have been compared with the basic algorithms such as MFO, SOS,

PSO, DE, BOA, JAYA, and with variants of the MFO algorithm. Also, validating the proposed EMFO was used to solve seven engineering design problems, which yield superior results than other popular algorithms. The results proved that the EMFO algorithm could use the better global optimum in the optimization phase, leading to rapid convergence. Further, the mutualism phase also allows the EMFO to avoid premature convergence and local optimum. Hence, we conclude that the proposed EMFO has convinced itself to be an encouraging method for solving engineering applications and real-life problems.

### Appendix 1

See Tables 16, 17 and 18.

**Table 16** Mathematical formulation of unimodal functions

Sl. no	Functions	Formulation of objective functions	<i>d</i>	<i>F</i> <sub>min</sub>	Search space
F1	Sphere	$f(x) = \sum_{j=1}^d x_j^2$	30	0	[− 100, 100]
F2	Cigar	$f(x) = 10^6 \sum_{j=1}^d x_j^2$	30	0	[− 100, 100]
F3	Step	$f(x) = \sum_{j=1}^d (x_j + 0.5)^2$	30	0	[− 100, 100]
F4	Rosenbrock	$f(x) = \sum_{j=1}^d [100(x_{j+1} - x_j^2) + (x_j - 1)^2]$	30	0	[− 2.048, 2.048]
F5	Schwefel 1.2	$f(x) = \sum_{j=1}^d \sum_{k=1}^j x_j^2$	30	0	[− 100, 100]
F6	Schwefel 2.21	$f(x) = \max\{ x_j , 1 \leq j \leq d\}$	30	0	[− 100, 100]
F7	Schwefel 2.22	$f(x) = \sum_{i=1}^d  x_i  + \prod_{i=1}^d  x_i $	30	0	[− 10, 10]
F8	Booth	$f(x) = (2x_1 + x_2 - 5)^2 + (x_1 + 2x_2 - 7)^2$	2	0	[− 10, 10]
F9	Matyas	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	2	0	[− 10, 10]
F10	Powell	$f(x) = \sum_{j=1}^{d/4} (x_{4j-3} + 10x_{4j-2})^2 + 5(x_{4j-3} + 10x_{4j-2})^2 + (x_{4j-2} + 2x_{4j-1})^4 + 10(x_{4j-3} - x_{4j})^4$	32	0	[− 4, 5]
F11	Zettl	$f(x) = (x - 1)^2 + x - 2^2 - 2x_1)^2 + 0.25x_1$	2	− 0.00379	[− 1, 5]
F12	Leon	$f(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2$	2	0	[− 1.2, 1.2]
F13	Zakhrov	$f(x) = \sum_{j=1}^d x_j^2 + \left(0.5 \sum_{j=1}^d jx_j\right)^2 + \left(0.5 \sum_{j=1}^d jx_j\right)^4$	2	0	[− 5, 10]
F14	Tablet	$f(x) = 10^6 \times x_1^2 + \sum_{j=1}^d x_j^6$	30	0	[− 1, 1]
F15	Quartic	$f(x) = \sum_{j=1}^d jx_j^4 + \text{random}(0, 1)$	30	0	[− 1.28, 1.28]

**Table 17** Mathematical formulation of multimodal functions

Sl. No	Functions	Formulation of objective functions	D	FMIN	SEARCH SPACE
F16	Bohachevsky	$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$	2	0	[- 100, 100]
F17	Ackley	$f(x) = 20 + e - 20e^{-\frac{1}{4} \left( \sqrt{\frac{1}{d} \sum_{j=1}^d x_j^2} \right)} - e^{d \left( \sum_{j=1}^d \cos(2\pi x_j) \right)}$	30	0	[- 32.768, 32.768]
F18	Griewank	$f(x) = \sum_{j=1}^d \frac{x_j^2}{4000} - \prod_{j=1}^d \cos\left(\frac{x_j}{\sqrt{j}}\right) - 1$	30	0	[- 600, 600]
F19	Rastrigin	$f(x) = 10D + \sum_{j=1}^d [x_j^2 - 10 \cos(2\pi x_j)]$	30	0	[- 5.12, 5.12]
F20	Schaffers	$f(x) = 0.5 + \frac{\sin^2(x_1^2 + x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	2	0	[- 100, 100]
F21	Easom	$f(x) = -\cos(x_1) \cos(x_2) \exp[-(x_1 - \pi)^2 \times 9(x_2 - \pi)^2]$	2	0	[- 100, 100]
F22	Schwefel 2.26	$f(x) = -\sum_{j=1}^d x_j \sin(\sqrt{ x_j })$	30	- 418.982	[- 500, 500]
F23	Powersum	$f(x) = \sum_{i=1}^d \left[ \left( \sum_{k=1}^d (x_k^i - b_i) \right)^2 \right]$	4	0	[0, 4]
F24	Penalized I.1	$f(x) = \frac{\pi}{d} \left\{ 10 \sin^2(\pi y_i) + \sum_{j=1}^{d-1} (y_j - 1)^2 \left[ 1 + 10 \sin^2(3\pi y_{j+1}) + (y_d - 1)^2 \right] + \sum_{j=1}^d u(x_j, 10, 100, 4) \right\}$ Where $y_i = 1 + \frac{1}{4}(x_j + 1)$ , $u(x_j, a, k, m) = \begin{cases} k(x_j - a)^m & x_j > a \\ 0 & x_j = a \\ k(-x_j - a)^m & x_j < a \end{cases}$	30	0	[- 50, 50]
F25	Foxholes	$f(x) = \left[ \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j} + \sum_{i=1}^D (x_i - a_{ij})^6 \right]^{-1}$	2	3	[- 65, 65]
F26	Kowalik	$f(x) = \sum_{j=1}^{11} \left[ a_j - \frac{x_j (b_j^2 + b_j x_j)}{(b_j^2 - b_j x_j - x_j)} \right]^2$	4	0.0003075	[- 5, 5]

**Table 18** Mathematical formulation of multimodal functions

Sl. No.	Functions	Formulation of objective functions	D	SEARCH SPACE	FMIN
F27	Branin	$f(x) = \left(x_2^2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}(x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\cos x_1\right)\right)$	2	0	[- 5, 5]
F28	Shekel 5	$f(x) = -\sum_{j=1}^5 [(x - a_j)(x - a_j)^T + c_j]^{-1}$	4	- 10.1499	[0, 10]
F29	Shekel- 7	$f(x) = -\sum_{j=1}^7 [(x - a_j)(x - a_j)^T + c_j]^{-1}$	4	[0, 10]	- 10.3999
F30	Shekel- 10	$f(x) = -\sum_{j=1}^{10} [(x - a_j)(x - a_j)^T + c_j]^{-1}$	4	[0, 10]	- 10.5319
F31	Bohachevsky 3	$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.3$	2	[- 50, 50]	0
F32	Colville	$f(x) = 100(x_1 - x_2^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2) + (x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1)$	4	[- 10, 10]	0
F33	Himmelblau	$f(x) = 1/30 \left[ \sum_{j=1}^d x_j^4 - 16x_j^2 + 5x_j \right]$	30	[- 5, 5]	- 78.33236
F34	Csendes	$f(x) = \sum_{j=1}^d x_j^6 (2 + \sin 1/x_j)$	30	[- 1, 1]	0
F35	Inverted cosine mixture	$f(x) = 0.1 \times 30 - \left[ 0.1 \times \sum_{j=1}^d 5\pi x_j - \sum_{j=1}^d x_j^2 \right]$	30	[- 1, 1]	0
F36	Salomon	$f(x) = 1 - \cos \left( 2\pi \sqrt{\sum_{j=1}^d x_j^2} \right) + 0.1 \sqrt{\sum_{j=1}^d x_j^2}$	30	[- 100, 100]	0

**Appendix 2**

a. Optimal capacity of gas production problem

$$\begin{aligned} \text{Min } f(x) &= 61.8 + 5.72 \times x_1 \times 0.2623 \\ &\times \left[ (40 - x_1) \times \ln \frac{x_2}{200} \right]^{-0.85} + 0.087 \\ &\times (40 - x_1) \times \ln \frac{x_2}{200} + 700.23 \times x_2^{-0.75} \end{aligned}$$

s.t.  $x_1 \geq 17.5, x_2 \geq 200, 17.5 \leq x_1 \leq 40, 300 \leq x_2 \leq 600;$

b. Gear train design problem

$$\begin{aligned} \text{Minimize } f(\vec{x}) &= \left[ \frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4} \right]^2, \\ \text{Subjected to } &12 \leq x_1, x_2, x_3, x_4 \leq 60, \\ \text{Where } \vec{x} &= [x_1 x_2 x_3 x_4] = [n\alpha \ n\beta \ n\gamma \ n\delta]. \end{aligned}$$

c. Cantilever beam design problem

$$\begin{aligned} \text{Minimize } f(\vec{x}) &= 0.06224(x_1 + x_2 + x_3 + x_4 + x_5), \\ \text{Subjected to : } &h(\vec{x}) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} \leq 1, \\ \text{Where } &0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100. \end{aligned}$$

d. I- beam vertical deflection problem

$$\text{Minimize } f(\vec{x}) = \frac{5000}{\frac{2t_f(h-t_f)^2}{2} + \frac{t_w*(h-2t_f)}{12} + \frac{bt_f^3}{6}}$$

Subjected to the cross-sectional area ( $< 300 \text{ cm}^2$ ).

$$x_1 = t_w * (h - 2t_f) + 2 * b * t_w \leq 300,$$

The stress constraint (where allowable bending stress of the beam is  $56 \text{ KN/CM}^2$ )

$$\begin{aligned} x_2 &= \frac{15 * b * 10^3}{(h - 2t_f) * t_w^3 + 2t_w * b^3} \\ &+ \frac{18 * h * 10^4}{(h - t_f)^3 * t_w + 2b(4t_f^2 + 3h(h - 2t_f))t_w * b^3} \leq 56 \end{aligned}$$

Provided  $0.9 \leq t_f, t_w \leq 5, 10 \leq b \leq 50$  and  $10 \leq h \leq 80$

e. Welded beam design problem

$$\begin{aligned} \text{Minimize } f(\vec{x}) &= f(h, l, t, b) = f(x_1, x_2, x_3, x_4) \\ &= Ax_2x_1^2 + Bx_4x_3(C + x_2). \end{aligned}$$

Subjected to  $h_1(x) = \tau_{\max} - \tau(x) \geq 0$

$$h_2(x) = \sigma_{\max} - \sigma(x) \geq 0$$

$$h_3(x) = \delta_{\max} - \delta(x) \geq 0$$

$$h_4(x) = x_4 - x_1 \geq 0$$



$$h_5(x) = P_c(x) - P \geq 0$$

where  $A = 1.10471$ ;  $B = 0.04811$ ;  $C = 14.0$   
 $\tau_{\max} = 13600$  psi;  $\sigma_{\max} = 30000$ psi;  $\delta_{\max} = 0.25$ in;

$$\tau(x) = \sqrt{(\tau'(x))^2 + \left( \frac{l\tau''(x)}{\sqrt{0.25(x_2^2 + (x_1 + x_3)^2)}} + (\tau''(x))^2 \right)}$$

$$\sigma(x) = \frac{50400}{x_4 x_3^2}$$

$$P_c(x) = 64746x_3(1 - 0.0282346x_3)x_4^3$$

$$\delta(x) = \frac{2.1952}{x_4 x_3^3}$$

$$\tau'(x) = \frac{6000}{\sqrt{2}x_2 x_1}$$

$$\tau''(x) = \frac{6000(0.5x_2 + 14)\sqrt{0.25(x_2^2 + (x_1 + x_3)^2)}}{2\left\{0.707x_1 x_2 \left(\frac{x_2^2}{12} + 0.25(x_1 + x_3)^2\right)\right\}}$$

Range of variables  $0.125 \leq x_1 \leq 5$

$0.1 \leq x_2, x_3 \leq 10$

$0.1 \leq x_4 \leq 5$

Parameters  $A = 1.10471$ ;  $B = 0.04811$ ;  $C = 14.0$   
 $\tau_{\max} = 13600$  psi;  $\sigma_{\max} = 30000$ psi;  $\delta_{\max} = 0.25$ in;  
 $E = 30 \times 10^6$  psi  $G = 12 \times 10^6$  psi.  
 $P = 6000$  lb.  
 $L = 14$ in.

f. Three-bar truss problem

$$\vec{k} = \{k_1, k_2, \}$$

Objective function

$$\text{Min} \cdot f(k) = L\{k_2 + 2\sqrt{2}k_1\}$$

Subject to

$$h_1(k) = \frac{k_2}{2k_2 k_1 + \sqrt{2}k_1^2} P - \sigma \leq 0,$$

$$h_2(k) = \frac{k_2 + \sqrt{2}k_1}{2k_2 k_1 + \sqrt{2}k_1^2} P - \sigma \leq 0,$$

$$h_3(k) = \frac{1}{k_1 + \sqrt{2}k_2} P - \sigma \leq 0,$$

where

$0 \leq k_1, k_2 \leq 1$ , and

$P = 2, L = 100 \& \sigma = 2$

g. Car-side crash design problem

$$\vec{a} = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}\}$$

Objective function

$$\text{Min} f(a) = 1.98 + 4.90a_1 + 6.67a_2 + 6.98a_3 + 4.01a_4 + 1.78a_5 + 2.73a_7,$$

Subject to

$$h_1(a) = 1.16 - 0.3717a_2 a_4 - 0.00931a_2 a_{10} - 0.484a_3 a_9 + 0.01343a_6 a_{10} \leq 1,$$

$$h_2(a) = 0.261 - 0.0159a_1 a_2 - 0.188a_1 a_8 - 0.019a_2 a_7 + 0.0144a_3 a_5 + 0.0008757a_5 a_{10} + 0.080405a_6 a_9 + 0.00139a_8 a_{11} + 0.00001575a_{10} a_{11} \leq 0.32,$$

$$h_3(a) = 0.214 + 0.00817a_5 - 0.131a_1 a_8 - 0.0704a_1 a_9 + 0.03099a_2 a_6 - 0.018a_2 a_7 + 0.0208a_3 a_8 + 0.121a_3 a_9 - 0.00364a_5 a_6 + 0.0007715a_5 a_{10} - 0.0005354a_6 a_{10} + 0.00121a_8 a_{11} \leq 0.32,$$

$$h_4(a) = 0.074 - 0.061a_2 - 0.163a_3 a_8 + 0.001232a_3 a_{10} - 0.166a_7 a_9 + 0.227a_2^2 \leq 0.32,$$

$$h_5(a) = 28.98 + 3.818a_3 - 4.2a_1 a_2 + 0.0207a_5 a_{10} + 6.63a_6 a_9 - 7.7a_7 a_8 + 0.32a_9 a_{10} \leq 32,$$

$$h_6(a) = 33.86 + 2.95a_3 + 0.1792a_{10} - 5.05a_1 a_2 - 11.0a_2 a_8 - 0.0215a_5 a_{10} - 9.98a_7 a_8 + 22.0a_8 a_9 \leq 32,$$

$$h_7(a) = 46.36 - 9.9a_2 - 12.9a_1 a_8 + 0.1107a_3 a_{10} \leq 32,$$

$$h_8(a) = 4.72 - 0.5a_4 - 0.19a_2 a_3 - 0.0122a_4 a_{10} + 0.009325a_6 a_{10} + 0.000191a_{11}^2 \leq 4,$$

$$h_9(a) = 10.58 - 0.674a_1 a_2 - 1.95a_2 a_8 + 0.02054a_3 a_{10} - 0.0198a_4 a_{10} + 0.028a(6)a(10) \leq 9.9,$$

$$h_{10}(a) = 16.45 - 0.489a_3 a_7 - 0.843a_5 a_6 + 0.0432a_9 a_{10} - 0.0556a_9 a_{11} - 0.000786a_{11}^2 \leq 15.7,$$

where

$0.5 \leq a_i \leq 1.5, i = 1, 2, 3, 4, 5, 6, 7$

$a_8, a_9 \in (0.192, 0.345),$

$-30 \leq a_{10}, a_{11} \leq 30.$

**Data availability** The manuscript does not contain any associate data.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest and the present article does not comprise any studies performed by any of the authors with human participants or animals.

## References

- Allam DY, Eteiba A (2016) Parameter's extraction of the three-diode model for the multi-crystalline solar cell/module using moth flame optimization algorithm. *Energy Convers Manag* 123:535–548
- Apinantanakon W, Sunat K (2017) Omfo: a new opposition-based moth-flame optimization algorithm for solving unconstrained optimization problems. In: *International conference on computing and information technology*. Springer: Cham 22–31. [https://doi.org/10.1007/978-3-319-60663-7\\_3](https://doi.org/10.1007/978-3-319-60663-7_3)
- Arora S, Singh S (2015) Butterfly algorithm with levy flights for global optimization. In: *International conference on signal processing, computing and control*. IEEE, Solan, 220–224. <https://doi.org/10.1109/ISPC.2015.7375029>
- Asghari K, Masdari M, Gharehchopogh FS, Saneifard R (2021) A chaotic and hybrid gray wolf-whale algorithm for solving continuous optimization problems. *Progress Artif Intell*. <https://doi.org/10.1007/s13748-021-00244-4>
- Asghari K, Masdari M, Gharehchopogh FS, and Saneifard R Multi-swarm and chaotic whale-particle swarm optimization algorithm with a selection method based on roulette wheel. *Expert Syst* e12779. <https://doi.org/10.1111/exsy.12779>
- Babar S, Mohammad I, Mohammad A, Zahoor R (2016) Novel application of moth flame optimization algorithm for solving economic load dispatch problems with emission and valve point loading effect. In: *4th international conference on energy, environment and sustainable development*
- Bhesdadiya R, Trivedi IN, Jangir P, Kumar A, Jangir N, Totlani R (2017) A novel hybrid approach particle swarm optimizer with moth-flame optimizer algorithm. In: *Advances in computer and computational sciences*. Springer, pp 569–577. [https://doi.org/10.1007/978-981-10-3770-2\\_53](https://doi.org/10.1007/978-981-10-3770-2_53)
- Bourouba B, Ladaci S, Chaabi A (2018) Moth flame optimisation algorithm-based fractional order PID controller with MRAC tuning configuration. *Int J Syst Control Commun* 9:148–171
- Chakraborty S, Saha AK, Sharma S, Mirjalili S, Chakraborty R (2021a) A novel enhanced whale optimization algorithm for global optimization. *Comput Ind Eng* 153:107086. <https://doi.org/10.1007/s12652-021-03304-8>
- Chakraborty S, Sharma S, Saha AK, Chakraborty S (2021b) SHADE-WOA: a metaheuristic algorithm for global optimization. *Appl Soft Comput*. <https://doi.org/10.1016/j.asoc.2021.107866>
- Cheng MY, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput Struct* 139:98–112
- Chickermane H, Gea H (1996) Structural optimization using a new local approximation method. *Int J Numer Methods Eng* 39:829–846
- Dabba A, Tari A, Meftali S, Mokhtari R (2021) Gene selection and classification of microarray data method based on mutual information and moth flame algorithm. *Expert Syst Appl* 166:114012. <https://doi.org/10.1016/j.eswa.2020.114012>
- Dash SP, Subhashini KR, Satapathy JK (2020) Optimal location and parametric settings of FACTS devices based on JAYA blended moth flame optimization for transmission loss minimization in power systems. *Microsyst Technol* 26(5):1543–1552
- Elaziz MA, Ewees AA, Ibrahim RA, Lu S (2020) Opposition-based moth-flame optimization improved by differential evolution for feature selection. *Math Comput Simul* 168:48–75
- Emary E, Zawbaa HM (2016) Impact of chaos functions on modern swarm optimizers. *PLoS ONE* 11:27410691
- Gandomi AH (2014) Interior search algorithm (ISA) a novel approach for global optimization. *ISA Trans* 53:1168–1183
- Gandomi AH, Yang XS, Alavi AH (2011) Mixed variable structural optimization using firefly algorithm. *Comput Struct* 89:2325–2336
- Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng Comput* 29:17–35
- Gharehchopogh FS, Gholizadeh H (2019) A comprehensive survey: whale optimization algorithm and its applications. *Swarm Evol Comput* 48:1–24
- Gu H, Wang X (2016) Application of NSGA-II algorithm in the design of car body lateral crashworthiness. *DEStech Trans Mater Sci Eng (icmeat)*. <https://doi.org/10.12783/dtmse/icmeat2016/6083>
- Gupta D, Ahlawat AK, Sharma A, Rodrigues JJPC (2020) Feature selection and evaluation for software usability model using modified moth-flame optimization. *Computing* 102(6):1503–1520
- Holland JH (1992) Genetic algorithms. *Sci Am* 267:66–72
- Jamil M, Yang XS (2013) A literature survey of benchmark functions for global optimization problems. *Int J Math Model Numer Optim* 4:150–194
- Kadry S, Rajinikanth V, Raja NSM, Hemanth DJ, Hannon NM, Raj ANJ (2021) Evaluation of brain tumor using brain MRI with modified-moth-flame algorithm and Kapur's thresholding: a study. *Evol Intel* 14(2):1053–1063
- Kamalpathi K, Priyadarshi N, Padmanaban S, Holm-Nielsen J, Azam F, Umayal C, Ramachandara murthy V (2018) A hybrid moth-flame fuzzy logic controller based integrated cuk converter fed brushless dc motor for power factor correction. *Electronics* 7:288. <https://doi.org/10.3390/electronics7110288>
- Kavousi-Fard A, Rostami MA, Niknam T (2015) Reliability-oriented reconfiguration of vehicle-to-grid networks. *IEEE Trans Industr Inf* 11:682–691
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of ICNN'95*. In: *International conference on neural networks*, 4:1942–1948
- Khalilpourazari S, Khalilpourazary S (2019) An efficient hybrid algorithm based on water cycle and moth-flame optimization algorithms for solving numerical and constrained engineering optimization problems. *Soft Comput* 23(5):1699–1722
- Khan MF, Aadil F, Maqsood M, Bukhari SH, Hussain M, Nam Y (2018) Moth flame clustering algorithm for internet of vehicle (MFCA-IoV). *IEEE Access* 7:11613–11629
- Khan BS, Raja MAZ, Qamar A, Chaudhary NI (2021) Design of moth flame optimization heuristics for integrated power plant system containing stochastic wind. *Appl Soft Comput* 104:107193
- Kigisirin S, Miyauchi H (2021) Short-term operational scheduling of unit commitment using binary alternative moth-flame optimization. *IEEE Access* 9:12267–12281
- Kumar S, Tejani GG, Mirjalili S (2019) Modified symbiotic organisms search for structural optimization. *Eng Comput* 35:1269–1296
- Li C, Li S, Liu Y (2016a) A least squares support vector machine model optimized by moth flame optimization algorithm for annual power load forecasting. *Appl Intell* 45:1166–1178

- Li Z, Zhou Y, Zhang S, Song J (2016b) Levy-flight moth-flame algorithm for function optimization and engineering design problems. *Math Probl Eng* 2016:1–22
- Li C, Niu Z, Song Z, Li B, Fan J, Liu PX (2018) A double evolutionary learning moth-flame optimization for real-parameter global optimization problems. *IEEE Access* 6:76700–76727
- Li Z, Zeng J, Chen Y, Ma G, Liu G (2021) Death mechanism-based moth-flame optimization with improved flame generation mechanism for global optimization tasks. *Expert Syst Appl*. <https://doi.org/10.1016/j.eswa.2021.115436>
- Ma L, Wang C, Xie NG, Shi M, Ye Y, Wang L (2021) Moth-flame optimization algorithm based on diversity and mutation strategy. *Appl Intell*. <https://doi.org/10.1007/s10489-020-02081-9>
- Masdari M, Zangakani M (2020) Green cloud computing using proactive virtual machine placement: challenges and issues. *Journal of Grid Computing* 18(4):727–759
- Masdari M, Salehi F, Jalali M, Bidaki M (2017) A survey of PSO-based scheduling algorithms in cloud computing. *J Netw Syst Manag* 25(1):122–158
- Masdari M, Barshande S, Ozdemir S (2019) CDABC: chaotic discrete artificial bee colony algorithm for multi-level clustering in large-scale WSNs. *J Supercomput* 75(11):7174–7208
- Milton F (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J Am Stat Assoc* 32:675–701
- Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowled Based Syst* 89:228–249
- Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. *Knowl Based Syst* 96:120–133
- Muangkote N, Sunat K, Chiewchanwattana S (2016) Multilevel thresholding for satellite image segmentation with moth-flame based optimization. In: *Proceedings of 13th international Joint conferences on computer science and software engineering (JCSSE)*, Khon Kaen, Thailand, 1–6. <https://doi.org/10.1109/JCSSE.2016.7748919>
- Nama S, Saha AK, Ghosh S (2016) Improved symbiotic organisms search algorithm for solving unconstrained function optimization”. *Decis Sci Lett* 5:361–380. <https://doi.org/10.5267/j.dsl.2016.2.004>
- Nama S, Saha AK, Ghosh S (2017a) A hybrid symbiosis organisms search algorithm and its application to real world problems. *Memetic Comput* 9(3):261–280
- Nama S, Saha AK, Ghosh S (2017b) A hybrid symbiosis organisms search algorithm and its application to real world problems. *Memetic Comput* 9:261–280
- Nama S, Saha AK, Sharma S (2020b) A novel improved symbiotic organisms search algorithm. *Comput Intell*. <https://doi.org/10.1111/coin.12290>
- Nama S, Saha AK, Sharma S (2020) A hybrid TLBO algorithm by quadratic approximation for function optimization and its application. *Recent trends and advances in artificial intelligence and internet of things*. Springer: Cham, pp 291–341. [https://doi.org/10.1007/978-3-030-32644-9\\_30](https://doi.org/10.1007/978-3-030-32644-9_30)
- Nanda SJ (2016) Multi-objective moth flame optimization. In: *International conference on advances in computing, communications and informatics (ICACCI)*. IEEE 2470–2476. <https://doi.org/10.1109/ICACCI.2016.7732428>
- Pelusi D, Mascella R, Tallini L, Nayak J, Naik B, Deng Y (2020) An Improved Moth-Flame Optimization algorithm with hybrid search phase. *Knowl Based Syst* 191:105277
- Poľap D (2017) Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism. *Symmetry* 9:203. <https://doi.org/10.3390/sym9100203>
- Rao R (2016) Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int J Ind Eng Comput* 7:19–34
- Reddy S, Panwar LK, Panigrahi BK, Kumar R (2018) Solution to unit commitment in power system operation planning using binary coded modified moth flame optimization algorithm (BMMFOA), a flame selection based computational technique. *J Comput Sci* 25:298–317
- Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population-based algorithm for solving constrained engineering optimization problems. *Appl Soft Comput* 13(5):2592–2612
- Sandgren E (1990) Nonlinear integer and discrete programming in mechanical design optimization. *J Mech Des* 112:223–229
- Sapre S, Mini S (2021a) Emulous mechanism based multi-objective moth-flame optimization algorithm. *J Parallel Distribut Comput* 150:15–33
- Sapre S, Mini S (2021b) A differential moth flame optimization algorithm for mobile sink trajectory. *Peer-to-Peer Network Appl* 14(1):44–57
- Sarma A, Bhutani A, Goel L (2017) Hybridization of moth flame optimization and gravitational search algorithm and its application to detection of food quality. In: *2017 intelligent systems conference (IntelliSys)*. IEEE 52–60. <https://doi.org/10.1109/IntelliSys.2017.8324318>
- Savsani V, Tawhid MA (2017) Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems. *Eng Appl Artif Intell* 63:20–32
- Sayed GI, Hassanien AE (2018) A hybrid SA-MFO algorithm for function optimization and engineering design problems. *Complex Intell Syst* 4:195–212
- Shan W, Qiao Z, Heidari AA, Chen H, Turabieh H, Teng Y (2021) Double adaptive weights for stabilization of moth flame optimizer: balance analysis, engineering cases, and medical diagnosis. *Knowl-Based Syst* 214:106728. <https://doi.org/10.1016/j.knosys.2020.106728>
- Sharma S, Saha AK, Majumder A, Nama S (2021) MPBOA-A novel hybrid butterfly optimization algorithm with symbiosis organisms search for global optimization and image segmentation. *Multimed Tools Appl* 80(8):12035–12076
- Sharma S, Saha AK (2021) Bosca—a hybrid butterfly optimization algorithm modified with sine cosine algorithm. *Progress in advanced computing and intelligent engineering*. Springer: Singapore, pp 360–372. [https://doi.org/10.1007/978-981-15-6584-7\\_35](https://doi.org/10.1007/978-981-15-6584-7_35)
- Sharma S, Saha AK. m-MBOA (2020) A novel butterfly optimization algorithm enhanced with mutualism scheme. *Soft Comput* 24:4809–27
- Singh P, Prakash S (2017) Optical network unit placement in fiber-wireless (FiWi) access network by moth-flame optimization algorithm. *Opt Fiber Technol* 36:403–411
- Singh U, Salgotra R (2018) An enhanced moth flame optimization. *Neural Comput Appl* 26:1–35
- Soliman GMA, Khorshid MMH, Abou-El-Enien THM (2016) Modified moth-flame optimization algorithms for terrorism prediction 5(7):2
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11:341–359
- Suja KR (2021) Mitigation of power quality issues in smart grid using levy flight-based moth flame optimization algorithm. *J Ambient Intell Humaniz Comput*. <https://doi.org/10.1007/s12652-020-02626-3>
- Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. *Int J Numer Methods Eng* 24:359–373

- Tan LS, Zainuddin Z, Ong P (2020) Wavelet neural networks-based solutions for elliptic partial differential equations with improved butterfly optimization algorithm training. *Appl Soft Comput* 95:106518. <https://doi.org/10.1016/j.asoc.2020.106518>
- Tejani GG, Pholdee N, Bureerat S, Prayogo D, Gandomi AH (2019) Structural optimization using multi-objective modified adaptive symbiotic organisms search. *Expert Syst Appl* 125:425–441
- Tumar I, Hassouneh Y, Turabieh H, Thaher T (2020) Enhanced binary moth flame optimization as a feature selection algorithm to predict software fault prediction. *IEEE Access* 8:8041–8055
- Verma S, Saha S, Mukherjee V (2017) A novel symbiotic organisms search algorithm for congestion management in deregulated environment. *J Exp Theor Artif Intell* 29:59–79
- Wang GG, Deb S, Cui Z (2015) Monarch Butterfly Optim Neural Comput Appl 31:1995–2014
- Wang M et al (2017) Toward an optimal kernel extreme learning machine using a chaotic moth flame optimization strategy with applications in medical diagnoses. *Neurocomputing* 267:69–84
- Wang Z, Luo Q, Zhou Y (2020) Hybrid metaheuristic algorithm using butterfly and flower pollination base on mutualism mechanism for global optimization problems. *Eng Comput*. <https://doi.org/10.1007/s00366-020-01025-8>
- Wu W, Li Z, Lin Z, Wu W, Fang D (2018) Moth-flame optimization algorithm based on chaotic crisscross operator. *Comput Eng Appl* 2018:22
- Yamany W, Fawzy M, Tharwat A, Hassanien AE (2015) Moth flame optimization for training multi-layer perceptrons. In: *Proceedings of 11th international computer engineering conference (ICENCO)*, Cairo, Egypt, 267–272. <https://doi.org/10.1109/ICENCO.2015.7416360>
- Yang XS (2009) Firefly algorithms for multimodal optimization. In: *International symposium on stochastic algorithms*. Springer 169–178. DOI: [https://doi.org/10.1007/978-3-642-04944-6\\_14](https://doi.org/10.1007/978-3-642-04944-6_14)
- Yi Y, He R (2014) A novel artificial bee colony algorithm. In: *2014 sixth international conference on intelligent human–machine systems and cybernetics*, 1:271–274.
- Yidiz BS, Yidiz AR (2017) Moth flame optimization algorithm to determine optimal machining parameters in manufacturing processes. *Mater Test* 59:425–429
- Yildiz AR, Abderazek H, Mirjalili S (2020) A comparative study of recent non-traditional methods for mechanical design optimization. *Arch Comput Methods Eng* 27(4):1031–1048
- Zawbaa HM, Emary E, Parv B, Sharawi M (2016) Feature selection approach based on moth flame optimization algorithm. *Proc IEEE Congr Evol Comput (CEC)*. <https://doi.org/10.1109/CEC.2016.7744378>
- Zhang Z, Qin H, Yao L, Liu Y, Jiang Z, Feng Z, Ouyang S (2020) Improved multi-objective moth-flame optimization algorithm based on r-domination for cascade reservoirs operation. *J Hydrol* 581:124431
- Zhao X, Fang Y, Liu L, Li J, Xu M (2020) An improved moth-flame optimization algorithm with orthogonal opposition-based learning and modified position updating mechanism of moths for global optimization problems. *Appl Intell* 50(12):4434–4458

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.