Applied Computing and Intelligence
https://www.aimspress.com/journal/aci

Applied Computing and Intelligence
5(2): 213–235.
DOI: 10.3934/aci.2025013
Received: 30 March 2025
Revised: 24 August 2025
Accepted: 05 September 2025
Published: 10 September 2025

Research article

Enhanced mountain gazelle optimizer for global performance

Abdul-Fatawu Seini Yussif[1,*], Bright Ayasu[2] and Toufic Seini[3]

[1] School of Engineering, University for Development Studies, Tamale, Ghana
[2] Department of Computer and Electrical Engineering, University of Energy and Natural Resources, Sunyani, Ghana
[3] Department of Mathematics, University for Development Studies, Tamale, Ghana

* **Correspondence:** Email: fatawu.seini@uds.edu.gh; Tel: +233(0)240809483.

Academic Editor: Chih-Cheng Hung

Abstract: In this paper, we present the Enhanced Mountain Gazelle Optimizer (EnMGO), a new variant of the mountain gazelle optimizer developed to address the persistent challenge of balancing exploration and exploitation in high-dimensional optimization. While earlier modifications of the algorithm focused on reformulating the control F-parameter to improve stability, they remained limited in adaptability and convergence efficiency. The proposed EnMGO introduces two modifications: An inertia weight based on a chaotic random technique to improve global exploration, and an exponentially decreasing formulation of the F-parameter to enhance local exploitation. These mechanisms create a more adaptive search process capable of efficiently navigating complex solution spaces. The algorithm was evaluated on a set of standard benchmark functions and engineering design problems, and the results demonstrated that EnMGO consistently outperforms previous variants, achieving faster convergence and higher-quality solutions. It achieved better results in approximately 86.96% of the total 23 benchmark functions considered in the study, thereby highlighting its robustness and effectiveness. Furthermore, in the application to engineering design optimization problems, the EnMGO consistently outperformed the comparative algorithms across both design cases, reaffirming its potential as a highly reliable and efficient optimization tool. Based on the performance, the potential of EnMGO can be explored for application in real engineering optimization problems.

Keywords: algorithm; inertia weight; mountain gazelle, metaheuristic; optimizer

1. Introduction

Optimization constitutes a critical aspect of problem-solving across a broad spectrum of scientific, engineering, and industrial domains [1]. The main goal is to identify the most efficient, cost-effective, or high-performing solution from a defined set of feasible alternatives, typically under the presence of multiple conflicting objectives and constraints [2]. Traditional mathematical optimization techniques, although foundational, often fall short of effectively addressing real-world problems that are non-linear, multi-modal, and high-dimensional in nature [3]. These challenges have led to the increasing adoption of metaheuristic algorithms, which are widely recognized for their robustness, adaptability, and capability to yield near-optimal solutions within reasonable computational timeframes [4].

Metaheuristic algorithms are generally inspired by natural phenomena, biological evolution, or collective intelligence observed in social or ecological systems [5]. Notable examples include *genetic algorithms (GA)*, which draw from principles of natural selection and genetics [6]; *particle swarm optimization (PSO)*, modeled on the collective motion of bird flocks and fish schools [7]; and *grey wolf optimizer (GWO)*, which simulates the leadership hierarchy and cooperative hunting behavior of grey wolves [8]. Over the years, a myriad of such algorithms have been proposed [9], each attempting to address specific performance limitations associated with exploration, exploitation, convergence rate, or stagnation in local optima [10,11].

Among these, the mountain gazelle optimizer (MGO) has emerged as a recent addition, inspired by the behavioral ecology of mountain gazelles [12]. The MGO algorithm models critical aspects of gazelle behavior such as territorial defense, group dynamics, and foraging patterns. Although MGO demonstrated commendable potential in solving various optimization problems, several performance issues remain inherent in its original formulation [13]. These include an insufficient balance between global exploration and local exploitation, vulnerability to premature convergence, and suboptimal performance when applied to high-dimensional search spaces [14].

In response to these challenges, a modified version of the MGO—termed the improved f-parameter mountain gazelle optimizer (IFMGO)—was introduced in a previous study [15]. The IFMGO replaced the original MGO's F-parameter with a new mathematical formulation that improved the dynamic balance between exploration and exploitation. Comprehensive testing on both high- and fixed-dimensional benchmark functions demonstrated that IFMGO significantly outperforms both the base MGO and PSO in terms of convergence accuracy, solution stability, and computational efficiency [15]. The enhancements introduced in IFMGO therefore laid a solid foundation for further developments in the MGO framework.

Building upon the foundation established by IFMGO, we propose a more advanced variant of the MGO, herein referred to as the enhanced mountain gazelle optimizer (EnMGO). This enhancement introduces two key modifications to further elevate the algorithm's global search performance and resilience against stagnation. The first is the incorporation of an adaptive inertia weight strategy, inspired by the PSO algorithm, which dynamically adjusts the momentum of search agents during the optimization process [16]. This mechanism facilitates a more effective trade-off between exploration and exploitation throughout the iterations, enabling the algorithm to escape local optima and converge more efficiently toward global solutions.

The second enhancement involves the integration of chaotic mapping-based sequences into the position update mechanism. Chaotic maps, known for their deterministic yet unpredictable nature, are employed to introduce controlled randomness into the search process [17]. This increases population diversity and mitigates premature convergence by enabling the algorithm to more thoroughly explore the search space, particularly in high-dimensional and complex landscapes.

Our primary objective of this research is to rigorously evaluate the effectiveness of these enhancements in improving the convergence behavior, solution accuracy, and robustness of the MGO framework. The performance of EnMGO is benchmarked against algorithms using a comprehensive set of standardized optimization functions. Through this investigation, we aim to contribute a more scalable and generalizable optimization tool capable of addressing a wider class of real-world problems across domains.

2. Mountain gazelle optimization

In this study, we provide a new version of the MGO called the EnMGO. It is an improvement to the original MGO, which is inspired by the social life of the mountain gazelles in wildlife. Various modifications are proposed and integrated into the MGO to form the EnMGO. A comprehensive test on the EnMGO is done, and the performance is compared to the MGO and another version to justify the improved performance of the proposed EnMGO on standard benchmark test functions.

The original MGO draws inspiration from the mountain gazelles [13]. They are a type of mammal that lives in the Arabian Peninsula and nearby territories [18]. Although they are spread across a wide coverage, they have a relatively low population density [10]. This species is commonly found in habitats where Robinia trees grow [19]. Mountain gazelles are highly territorial, maintaining territories at considerable distances from one another [19]. They typically form three (3) types of groups: The herds of mother gazelles and their young ones, the groups of young male gazelles, and the solitary adult male gazelles within their respective territories [10]. The male gazelles often engage in fights, primarily competing for resources than for mates [18]. In these encounters, younger males tend to use their horns more frequently than older or the dominant males do. These gazelles can migrate by traveling over 120 kilometers in search of good food [20]. They are also known for their exceptional speed, capable of running about 100 meters at an average speed of 80 km/h [18]. The social behavior of the mountain gazelles in the wild was mimicked to develop the original MGO. The key behavioral components were modeled mathematically to form the algorithm. The mathematical modeling is presented in the next section.

Mathematical Model of MGO: The MGO algorithm is modeled considering the concept of the social behavior of mountain gazelles presented earlier, and their natural habitat [21]. It captures the major aspects of their grouping dynamics, which consist of the behavior of young male gazelles, the herds of mother gazelles and babies (maternity), the solitary and territorial nature of male gazelles, as well as the migration patterns to search for good food [13,22]. The mathematical framework is structured as follows.

2.1. The Territorial Solitary Male (TSM) phase

The male adults among the gazelles protect their prestigious territories against strange intruders who attempt to take over their territory or pose a possible danger to the members of the territory. This ideology is mathematically modeled in Eq (1) [13].

$$TSM = male_{gazelle} - \left|\left(r_{i1} \times BH - r_{i2} \times X(t)\right) \times F\right| \times Cof_r, \tag{1}$$

where r_{i1} and r_{i2} are integers selected randomly to be either 1 or 2. $male_{gazelle}$ is the position vector representing the function value of the best male gazelle.

The determination of values of variables *BH, F,* and Cof_r is based on Eqs (2)–(4), respectively. The *BH* parameter represents a young male herd coefficient vector, Cof_r is a randomly selected coefficient vector (with the *r* signifying the randomness) updated in each iteration and used to increase the search capability, and $X(t)$ is the position of the gazelle vector.

$$BH = X_{ra} \times r_1 + M_{pr} \times r_2, \qquad ra = \left\{\frac{N}{3}, \dots, N\right\}. \tag{2}$$

Variable X_{ra} represents a randomly selected solution (young male) within the specified range *ra*, while M_{pr} denotes the average number of search agents. *N* refers to the total number of gazelles, r_1 and r_2 are randomly generated numbers in the range (0, 1) [18].

$$F = N_1(D) \times \left(2 - Iter \times \left(\frac{2}{MaxIter}\right)\right). \tag{3}$$

Here, variable N_1 consists of randomly generated values corresponding to the problem's dimension (*D*). It is calculated based on a standard distribution approach. The terms *Iter* and *MaxIter* depict the current iteration and the maximum number of iterations, respectively [18].

$$Cof_i = \begin{cases} (a+1) + r_3, \\ a \times N_2(D), \\ r_4(D), \\ N_3(D) \times N_4(D)^2 \times \cos\left((r_4 . 2) \times N_3(D)\right), \end{cases} \tag{4}$$

where r_3 and r_4 represent terms with random values (numbers) generated from the range (0, 1). Moreover, parameters N_2, N_3, and N_4 are sets of random numbers having the size of the problem function. The *i* represents the iteration, while *D* retains its purpose as the problem dimension [18].

Variable *a* is a value that is determined using Eq (5) at every iteration [18].

$$a = -1 + Iter \times \left(\frac{-1}{MaxIter}\right). \tag{5}$$

2.2. The Maternity Herd (MH) phase

Just like other mammals in the wildlife, the mother gazelles have a protective instinct to keep their offspring safe. The mechanism applied by the mother gazelles in this context is expressed by employing mathematical modeling presented in Eq (6) [19].

$$MH = \left(BH + Cof_{1,r}\right) + \left(r_{i3} \times male_{gazelle} - r_{i4} \times X_{rand}\right) \times Cof_{1,r}. \tag{6}$$

In Eq (6), X_{rand} is the position of a function value of a randomly selected gazelle from the population expressed as a vector [19], and variables r_{i3} and r_{i4} are integers randomly selected as either 1 or 2. The term $Cof_{1,r}$ is a coefficient randomly selected from the several formulations in Eq (4) during the first iteration, where subscript *1,r* indicates the first iteration and randomness, respectively. The term introduces randomness and scaling to enhance diversity in the solution space.

2.3. The Bachelor Male Herds (BMH) phase

Young male gazelles create their new territories and try to win over some female gazelles to join them in the newly created territories [19]. The strategy of creating new territories by these young adult male gazelles is modeled in Eq (7).

$$BMH = (X(t) - D) + \left(r_{i5} \times male_{gazelle} - r_{i6} \times BH\right) \times Cof_r \quad . \tag{7}$$

In Eq (7), variable *X(t)* represents the values in vector form of the gazelle in the current iteration. Variables r_{i5} and r_{i6} are random values chosen randomly from (1, 2). The r_6 denotes a value randomly chosen from the range (0, 1). Cof_r is a randomly selected coefficient vector updated in each iteration and used to increase the search capability. Finally, the value of variable D is calculated using Eq (8).

$$D = \left(|X(t)| + \left|male_{gazelle}\right|\right) \times (2 . r_6 - 6). \tag{8}$$

2.4. The Migration in Search of Food (MSF) phase

This phase involves the habit of roaming randomly in search of nutritious food varieties by the mountain gazelles, especially during periods of the year when food is relatively scarce. This mechanism of searching by a random movement is modeled in Eq (9) [19]. This model generates values randomly within the search space.

$$MSF = (ub - lb) \times r_7 + lb. \tag{9}$$

Parameters lb and ub indicate the search space or range, with the first variable representing the lower search bound, while the second variable represents the upper or maximum search bound. The variable r_7 represents a random number chosen such that its maximum cannot be greater than 1 and its possible smallest value cannot be less than 0 [19].

3. Enhanced mountain gazelle optimization

In this section, we present the modifications carried out for selected parts of the MGO algorithm that led to the development of the EnMGO algorithm.

3.1. The proposed modification

The proposed EnMGO is an improved version of the original MGO. It takes the same inspiration from the gazelles' social lifestyle, modeled mathematically in Equations from (1) to (9), with some modifications introduced to effectively boost its performance. The proposed modifications introduced are comprehensively presented below.

The TSM phase plays a major role in the population update operators of the algorithm. The mathematical representation is presented as in Eq (1) [13]. This update phase is one of the keys to determining the performance of the MGO algorithm. Hence, positive amendments in this phase can lead to enhanced performance.

Two major amendments are proposed in this modification to improve the TSM phase for better global performance. They include an alternative approach for determining the values of the factor F parameter in Eq (1) to improve the exploitation, and an introduction of an adaptive inertia weight to improve the exploration quality of the algorithm.

3.1.1. Alternative F parameter

In the original MGO algorithm, the value of the F parameter is determined using Eq (3) [13]. The values produced by this equation start with 7.3743 and exponentially decrease to the lowest value of 1. The decreasing characteristics of the value of the F parameter are illustrated below in Figure 1.

The values of the F parameter presented in Figure 1 greatly influence the convergence process of the MGO algorithm. It generates exponentially decreasing factors that significantly determine the amount of deduction that needs to be done on the best solution obtained so far for a possible better solution.

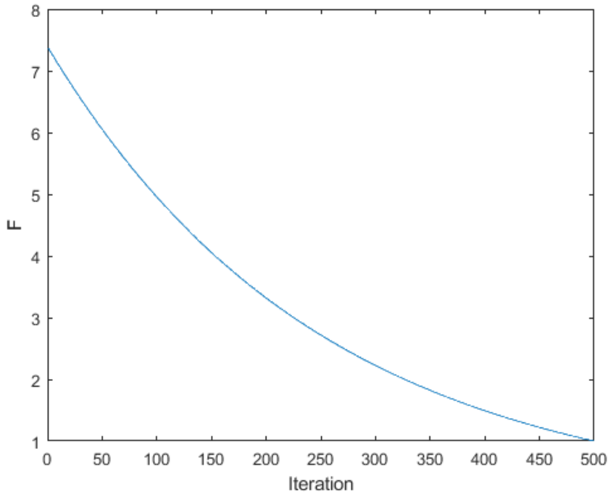

Figure 1. The original varying characteristics of the F parameter.

The range of the F parameter values indicates substantial deductions during the algorithm's execution [15]. However, it has been established that smaller update changes in the best solutions give better chances of obtaining the global solution to the optimization problems, but with slow convergence speed [20]. While larger update changes lead to faster convergence but very high tendency to skip the global solution of the optimization problem. The set of values for the F parameter, as presented in Figure 1, shows that the original MGO algorithm has larger update changes, which cause it to struggle to produce results very close to the global solutions.

An alternative approach for the determination of the F parameter values is proposed. This approach is based on the natural exponent inertia weight (NEIW) strategy [16]. The proposition leads to a new way of determining the value of the F parameter at each iteration, which is given in Eq (10).

$$F = w_1 + (w_2 - w_1) \times e^{-\left(\frac{Iter}{\left(\frac{MaxIter}{10}\right)}\right)}, \tag{10}$$

where the values of w_1, and w_2 are given as 0.00009 and 0.9, representing the possible lowest and highest that could be produced by the equation, respectively. The maximum iterations and the iteration count are also represented as *MaxIter* and *Iter,* respectively. The characteristics indicating how the value changes with respect to the iterations are illustrated below for 500 iterations in Figure 2.

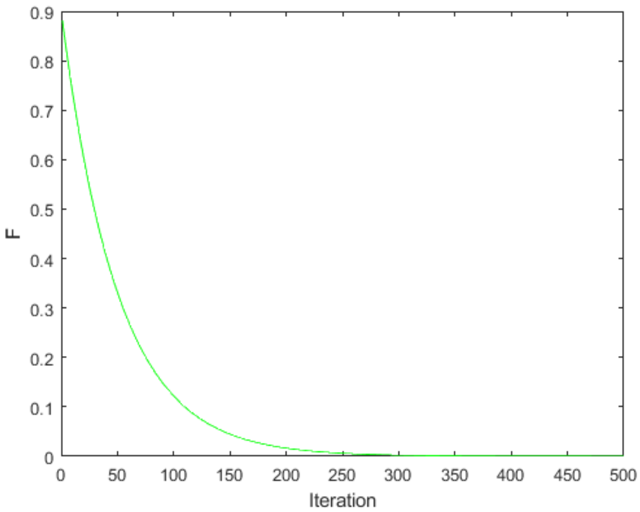

Figure 2. Varying characteristics of the proposed F parameter.

As shown in Figure 2, the proposed values range from a maximum of 0.9 and decrease exponentially to closer to 0. This ensures that a smaller magnitude of difference is made for the best result at every iteration to enhance the chances of obtaining global solutions. Hence, the exploitation by the algorithm is improved.

3.1.2. Introduction of Inertia Weight

The first modification proposed above enhances the algorithm's ability to obtain global solutions for optimization problems. However, it has the possibility of reducing the convergence speed, which means the exploration quality is poor. In this second modification, a chaotic random inertia weight (CRIW) is proposed to augment this possible negative impact of the first modification by introducing a chaotic random search [16]. The weight ω is integrated into Eq (1) and it modifies into Eq (11) below:

$$TSM = \omega \times male_{gazelle} - \left|\left(r_{i1} \times BH - r_{i2} \times X(t)\right) \times F\right| \times Cof_r, \quad (11)$$

where F is determined at every iteration based on Eq (10), and the value of ω is calculated based on the CRIW function given below in Eq (12). All other notations retain their meaning defined before.

$$\omega(t) = 0.5 \times rand + 0.5z, \quad (12)$$

where $\omega(t)$ is the weight at iteration t , $rand$ is a randomly generated value within the range of 0 to 1, and z is a constant integer.

By considering the value of z to be 1, and varying values of the proposed weight (ω), values for the 500 iterations are illustrated in Figure 3.

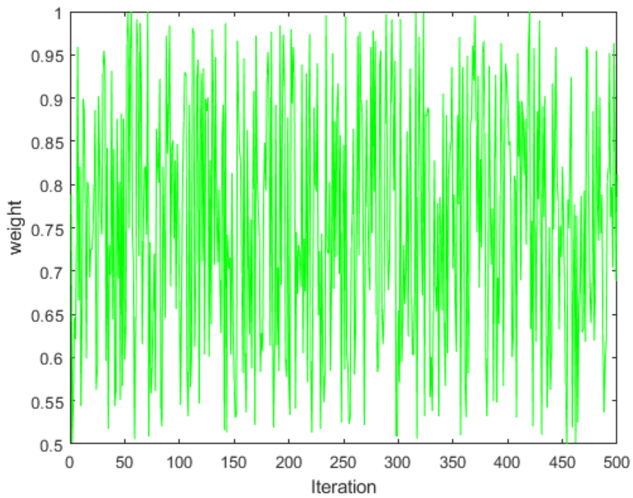

Figure 3. Varying characteristics of ω.

As illustrated in Figure 3, the proposed weight value changes randomly within a range of 0.5 to 1, ensuring a possible jump from one value to another. This improves the algorithm's exploration capability.

The two proposed modifications improve the algorithm's exploitation and exploration techniques, respectively, greatly enhancing its ability to solve different optimization problems [16]. The implementation guide/procedure of the EnMGO is presented in Figure 4 as a flowchart.

Figure 4 presents the systematic steps showing a comprehensive procedure for the implementation of the proposed EnMGO algorithm. Following these steps in any appropriate coding language will give the expected outcome.

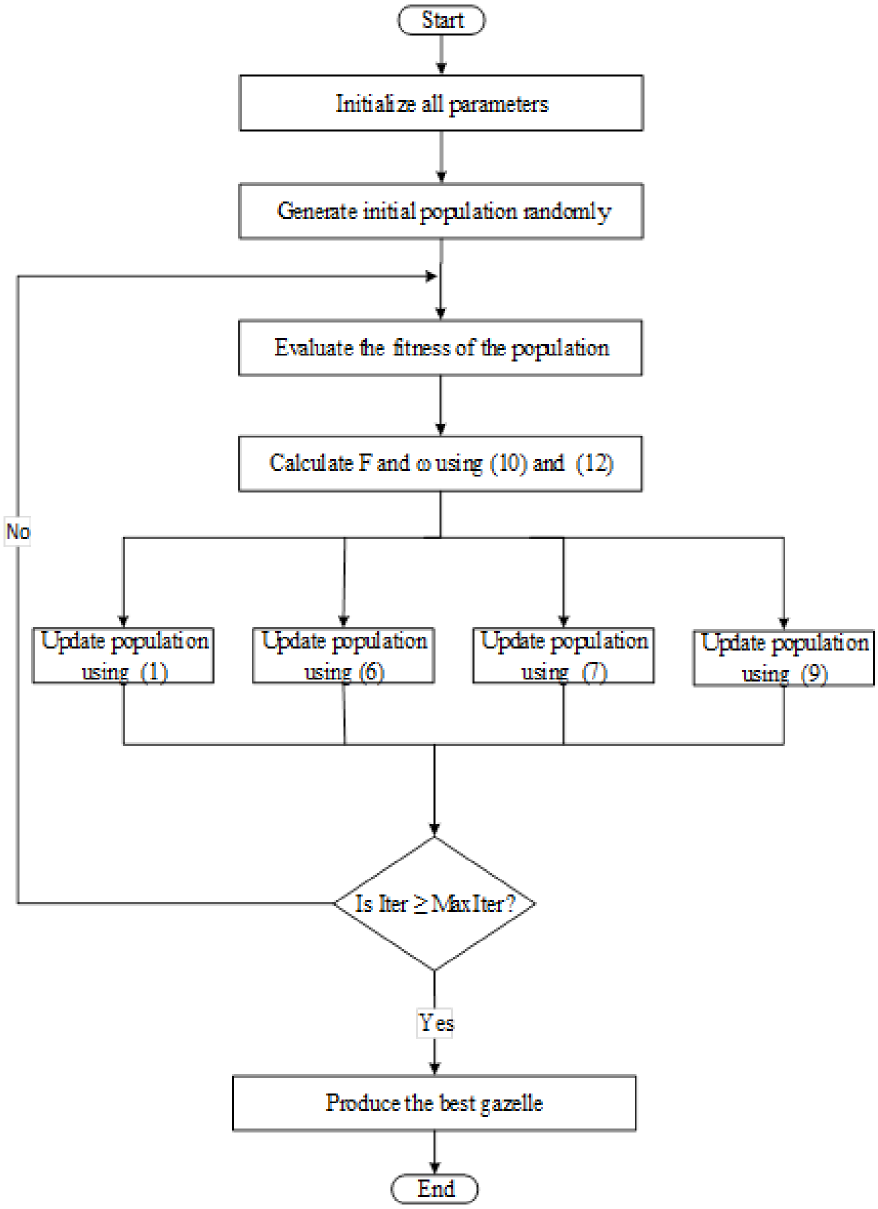

Figure 4. Implementation flowchart of the EnMGO algorithm.

3.2. Testing on standard benchmark functions

To demonstrate the enhanced performance of the EnMGO, it is tested using 23 standard benchmark test functions that were employed in the literature of the original MGO algorithm [23]. These functions are a reliable basis for assessing the algorithm's efficiency across optimization challenges. The first 7 test functions (F1–F7) are high-dimensional unimodal optimization problems. These functions are particularly useful for evaluating an algorithm's exploitation capability and its ability to scale effectively in problems with a single global optimum. The ability to efficiently converge to the best solution is a crucial aspect of optimization algorithms, and these functions provide insights into the EnMGO algorithm's strength in this regard.

The second category includes 6 benchmark test functions (F8–F13), which represent high-dimensional multimodal optimization problems. These test functions contain multiple local optimum solutions and are designed to assess how well an optimization algorithm can maintain a favorable exploration and exploitation balance [13]. An efficient algorithm should not only search broadly across the solution space but in addition also refine its search to converge toward the best possible solution. By evaluating EnMGO with these functions, its ability to avoid premature convergence and explore diverse regions of the search space can be examined.

The final category consists of 10 fixed-dimensional benchmark test functions (F14–F23), which are suitable for testing an algorithm's exploration ability and its capacity to handle problems with multiple local optima solutions [13]. These test functions are essential for analyzing the diversity of solutions generated by the EnMGO algorithm and determining its effectiveness in overcoming complex optimization challenges.

By considering all three categories of standard benchmark test functions, we provide a comprehensive assessment of the EnMGO's performance as compared to that of the original MGO algorithm and an existing variant.

The proposed EnMGO algorithm is implemented by coding and executed in a MATLAB environment using an HP Pavilion laptop (HP EliteBook). The implementation involves coding and running simulation tests on the standard benchmark test functions to evaluate the algorithm's performance on each of the functions. On each standard benchmark test function, the algorithm is tested 30 times independently. The specific parameter settings of the pre-requisite variables used for the simulations are detailed in Table 1 for effective comprehension.

Table 1. Settings of parameters

Parameter	**Value**
Size of population	30
Maximum number of Iterations	500
Number of Simulations	30

The detailed information on the standard benchmark test functions used in the test is accurately presented in Table 2 [13]. It presents the mathematical expressions of the functions for clarity. The range is also presented to indicate the numerical values search space within which the algorithm operates to find optimal solutions for the test functions. The dimensions are also captured accordingly to show the problem dimensions under each function, which can be interpreted as the number of variables in the given optimization problem. Finally, the global solutions (F_{min}) of the various test functions are determined. They are the best possible solutions for the test functions, and they are the benchmark for assessing any solution produced by an algorithm on the said test function. The relatively closer the results produced by the algorithm are to the global solution, the better the performance.

Table 2. Details of benchmark functions.

No.	**Equations**	**Range/Dimension**	**Fmin**
	High-dimensional unimodal benchmark functions		
F1	$f(x) = \sum_{i=1}^{d} x_i^2$	[-100, 100]/ 30	0
F2	$f(x) = \sum_{i=1}^{d} \lvert x_i \rvert + \prod_{i=1}^{d} \lvert x_i \rvert$	[-10, 10]/ 30	0
F3	$f(x) = \sum_{i=1}^{d} \left(\sum_{j=1}^{i} x_j \right)^2$	[-100, 100]/ 30	0
F4	$f(x) = max_i\{\lvert x_i \rvert, 1 \leq i \leq d\}$	[-100, 100]/ 30	0

Continued on next page

No.	Equations	Range/Dimension	Fmin
F5	$f(x)=\sum_{i=1}^{d-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	[-30, 30]/ 30	0
F6	$f(x)=\sum_{i=1}^{d}(\lvert x_i+0.5\rvert)^2$	[-100, 100]/ 30	0
F7	$f(x)=\sum_{i=1}^{d}(ix_i^4+random0,1)$	[-1.28, 1.28]/ 30	0
	High-dimensional multimodal benchmark functions		
F8	$f(x)=-\sum_{i=1}^{d}\left(x_i\, sin\left(\sqrt{\lvert x_i\rvert}\right)\right)$	[-500, 500]/ 30	-12,569
F9	$f(x)=10d+\sum_{i=1}^{d}\left[x_i^d-10\,cos(2\pi x_i)\right]$	[-5.12, 5.12]/ 30	0
F10	$f(x)=-20\,exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right)-exp\left(\frac{1}{d}\sum_{i=1}^{d}cos\,2\,\pi x_i\right)+20+e$	[-32, 32]/ 30	0
F11	$f(x)=\frac{1}{4000}\sum_{i=1}^{d}x_i^2-\prod_{i=1}^{d}cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[-600, 600]/ 30	0
F12	$f(x)=\frac{\pi}{d}\left\{10\,sin(\pi y_1)+\sum_{i=1}^{d-1}(y_i-1)^2[1+10\,sin^2(\pi y_{i+1})]+(y_d-1)^2\right\}$ $y_i=1+\frac{x_i+1}{4}$ $U(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, x_i>a\\ 0,-a<x_i<a\\ k(-x_i-a)^m, x<-a\end{cases}$	[-50, 50]/ 30	0
F13	$f(x)=0.1\left\{sin^2(3\pi x_i)+\sum_{i=1}^{d}(x_i-1)^2[1+sin^2(3\pi x_i+1)]+(x_d-1)^2\right\}$ $+\sum_{i=1}^{d}U(x_i,5,100,4)$	[-50, 50]/ 30	0
	Fixed-dimensional multimodal benchmark functions		
F14	$f(x)=\left[\frac{1}{500}+\sum_{i=1}^{25}\frac{1}{i+\sum_{j=1}^{2}\left(x_j-a_{j,i}\right)^6}\right]^{-1}$	[-65.53, 65.53]/ 2	0.998
F15	$f(x)=\sum_{i=1}^{d}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	[-5, 5]/ 4	0.00030
F16	$f(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	[-5, 5]/ 2	-1.0316
F17	$f(x)=\left(x_2-\frac{5.1}{4\pi^2}x^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)cos\,x_1+10$	[-5, 0] [10, 15]/ 2	0.398
F18	$f(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)].$ $[30+(2x_1-3x_2)^2.(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	[-5, 5]/ 2	3
F20	$f(x)=-\sum_{i=1}^{4}a_i\,exp\left(-\sum_{j=1}^{6}b_{ij}\left(x_j-p_{ij}\right)^2\right)$	[0, 1]/ 6	-3.32

Continued on next page

No.	**Equations**	**Range/Dimension**	**Fmin**
F21	$f(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]/ 4	-10.1532
F22	$f(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]/ 4	-10.4028
F23	$f(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]/ 4	-10.5363

3.3. Testing on engineering design problems

The EnMGO is tested on some engineering design problems, and the performances on these problems are compared to the performance of the original MGO and IFMGO reported in [19] to further establish the effectiveness of the EnMGO in solving real-world optimization problems. The engineering designs include: The pressure vessel design problem (PVDP) and the three-bar truss design problem (TTDP).

3.3.1. The pressure vessel design problem (PVDP)

The pressure vessel design problem is a widely used benchmark for testing optimization algorithms, aiming to minimize the manufacturing cost of a cylindrical pressure vessel with hemispherical heads. This involves optimizing four key design variables: The discrete thicknesses of the shell (x_1) and head (x_2), and the continuous inner radius (x_3) and cylindrical length (x_4). The problem is challenging due to its non-linear objective function and constraints (related to stress, minimum thickness, volume, and length), mixed discrete and continuous variables, and the presence of multiple local optima, making it an excellent test for an algorithm's ability to handle complexity, constraints, and find a globally optimal solution in a practical engineering context. The optimization problem is expressed mathematically in (13).

$$f(x) = 0.622x_1x_3x_4 + 1.778x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3, \tag{13}$$

where x_1, x_2, x_3, and x_4 represent the design variables.

Constraints:

$$\begin{cases} g_1(x) = -x_1 + 0.0193x_3 \le 0, \\ g_2(x) = -x_2 + 0.00954x_3 \le 0, \\ g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0, \\ g_4(x) = x_4 - 240 \le 0. \end{cases} \tag{14}$$

Variables bound:

$$\begin{cases} 0.0625 \le x_1, x_2 \le 6.1875, \\ 10 \le x_3, x_4 \le 200. \end{cases} \tag{15}$$

3.3.2. The three-bar truss design problem (TTDP)

The three-bar truss design problem involves minimizing the total weight or volume of a simple three-bar truss (Figure 5) by optimizing the cross-sectional areas of its bars while adhering to critical constraints like maximum allowable stress in each bar and potential displacement limits. This problem is popular because its continuous, constrained, and often non-linear nature provides a realistic yet well-understood challenge for evaluating the effectiveness of metaheuristics in finding optimal solutions within a complex search space. The objective function is given in Eq (16) below:

$$\text{Minimize } f(x) = \left(2\sqrt{2}x_1 + x\right).l. \tag{16}$$

Constraints:

$$\begin{cases} g_1(x) = \frac{\sqrt{2}x_1+x_2}{\sqrt{2}x_1^2+2x_1x_2}P - \sigma \leq 0, \\ g_2(x) = \frac{x_2}{\sqrt{2}x_1^2+2x_1x_2}P - \sigma \leq 0, \\ g_{31}(x) = \frac{1}{\sqrt{2}x_2+x_1}P - \sigma \leq 0, \end{cases} \tag{17}$$

where $0 \leq x_1, x_2 \leq 1, l = 100cm, P = 2KNcm^{-2}, \sigma = 2KNcm^{-2}$.

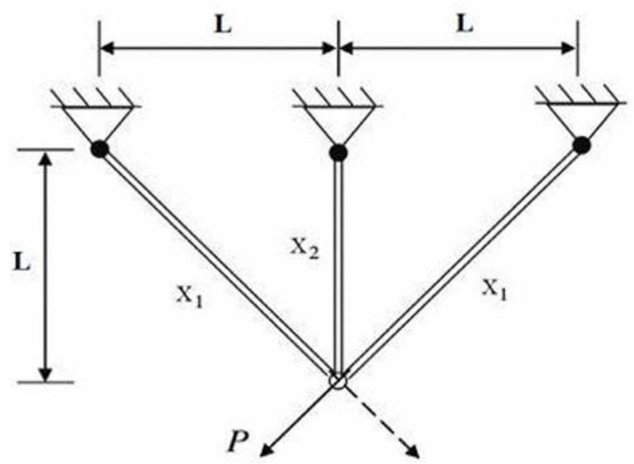

Figure 5. The three-bar truss design.

4. Results and discussion

4.1. Performance on standard benchmark test functions

The EnMGO was executed 30 times for each of the 23 test functions, and key statistical metrics (best, mean, worst, and standard deviation (STD)) were recorded [18]. The performances of the EnMGO on the benchmark functions are compared with those of two other algorithms reported in the literature [13]. The comparative analysis of EnMGO, MGO, and IFMGO is presented based on various types of test functions.

For the performance on high-dimensional unimodal functions, the simulation outcome of the proposed EnMGO is compared to the original MGO [13] and another variant of the MGO called the IFMGO [15] on the unimodal high-dimensional functions in Table 3. The comparison is presented in Table 4, where the best result under each function for the various statistical indicators is bolded.

The results presented in Table 3 indicate that the proposed EnMGO outperformed the MGO and the IFMGO exceptionally by producing the global optimal solutions on F1–F4. It also performed better than the MGO and the IFMGO for function F7. However, the IFMGO produced better results for functions F5 and F6 by outperforming the MGO and the proposed EnMGO. The proposed EnMGO performed relatively better by outperforming the other algorithms for five (5) benchmark functions of the seven (7) tested, representing about 71.4% good performance. This justifies the proposed EnMGO's exploitation power in solving optimization problems with one global solution.

Table 3. Results on unimodal functions.

Function	Parameter	EnMGO	MGO [13]	IFMGO 15]
	Best	**0**	2.415E-81	5.567E-273
F1	Worst	**0**	4.949E-71	4.268E-236
	Mean	**0**	4.7455E-72	1.429E-237
	STD	**0**	1.340E-71	**0**
	Best	**0**	1.676E-46	2.379E-154
F2	Worst	**0**	6.078E-41	1.628E-136
	Mean	**0**	3.907E-42	5.426E-138
	STD	**0**	1.189E-41	2.972E-137
	Best	**0**	3.531E-14	3.993E-67
F3	Worst	**0**	1.637E-7	2.079E-37
	Mean	**0**	6.822E-9	6.958E-39
	STD	**0**	2.979E-8	3.795E-38
	Best	**0**	5.25E-30	2.287E-117
F4	Worst	**0**	4.142E-22	5.137E-93
	Mean	**0**	1.591E-23	1.722E-94
	STD	**0**	7.543E-23	9.378E-94
	Best	2.4386E-9	**0**	**0**
F5	Worst	2.2368E-6	2.556E-22	**3.212E-29**
	Mean	1.8680E-7	1.195E-23	**2.718E-30**
	STD	4.1333E-7	4.959E-23	**7.541E-30**
	Best	1.0672E-7	4.809E-12	**1.243E-13**
F6	Worst	3.7268E-3	3.510E-8	**3.170E-10**
	Mean	2.8979E-4	4.540E-9	**2.135E-11**
	STD	7.1283E-4	7.654E-9	**6.922E-9**
	Best	**3.7252E-6**	3.245E-5	3.239E-5
F7	Worst	**3.6953E-4**	1.534E-3	9.906E-4
	Mean	**1.1618E-4**	5.596E-4	2.379E-4
	STD	**9.8718E-5**	3.889E-4	2.164E-4

The convergence characteristics of the EnMGO, MGO, and IFMGO are presented in the following curves (Figures 6–12) to illustrate the convergence processes.

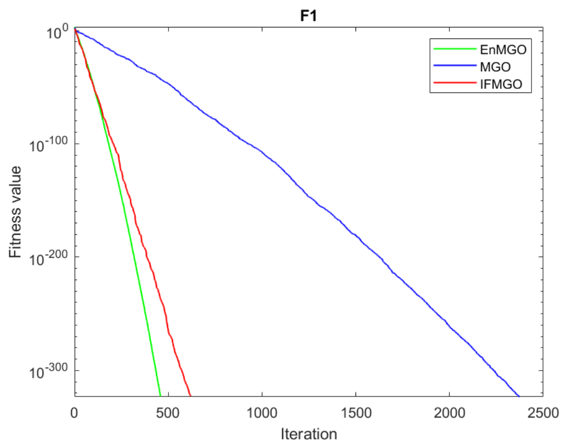

Figure 6. Convergence characteristics of function F1.

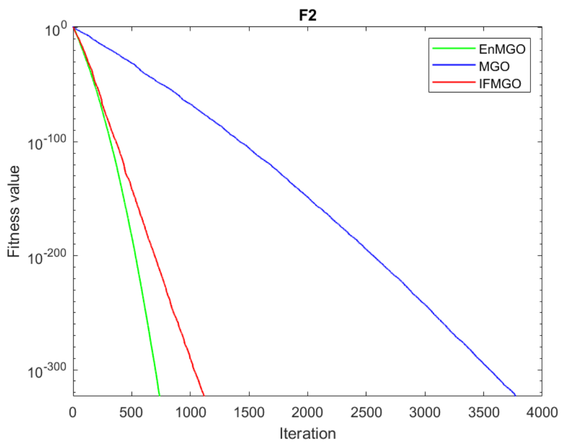

Figure 7. Convergence characteristics of function F2.

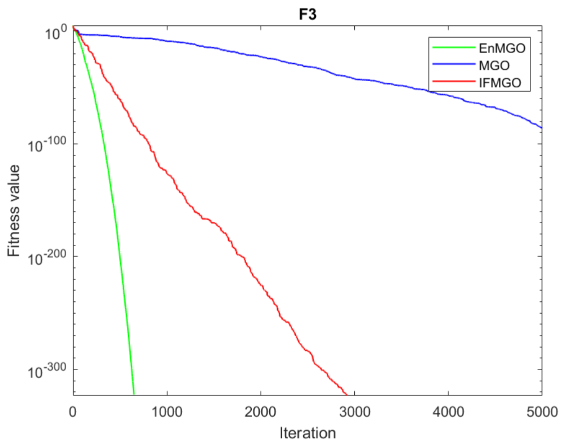

Figure 8. Convergence characteristics of function F3.

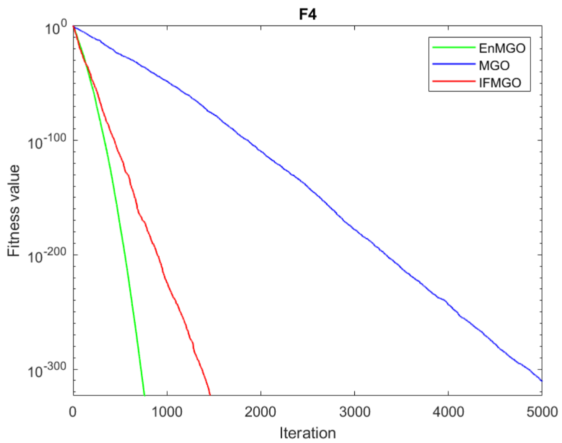

Figure 9. Convergence characteristics of function F4.

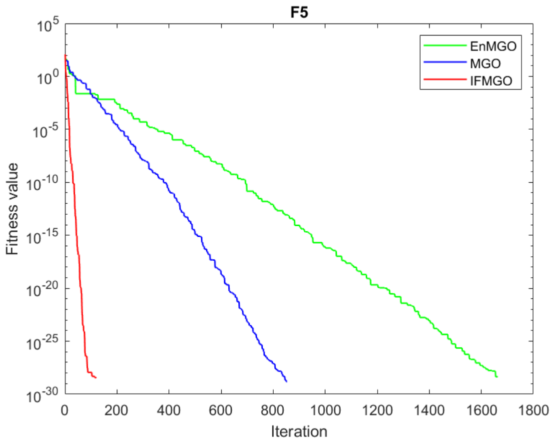

Figure 10. Convergence characteristics of function F5.

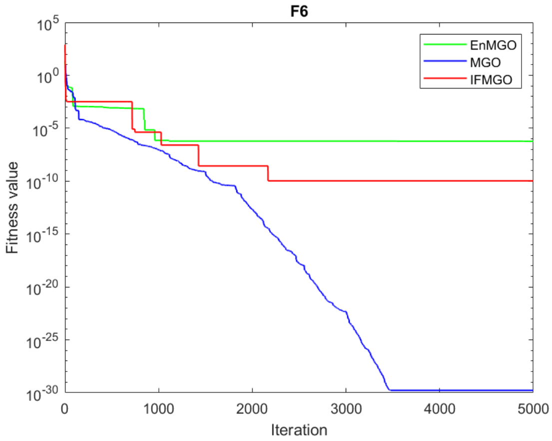

Figure 11. Convergence characteristics of function F6.

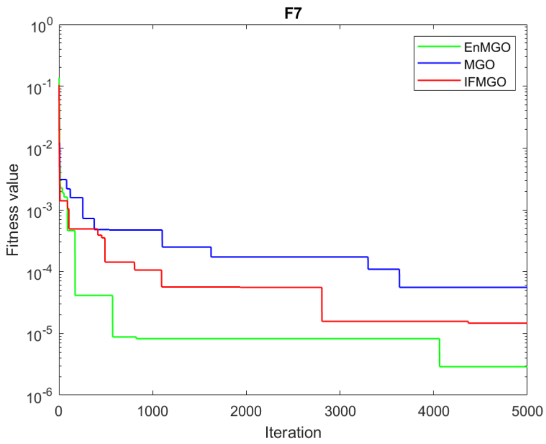

Figure 12. Convergence characteristics of function F7.

The convergence characteristics presented above in F1–F4, and F7 show the superior performance of the EnMGO, where it converges effectively towards the global solutions. In F5 and F6, the MGO shows better convergence characteristics, which agree with the results presented in Table 3.

4.1.1. Performance on high-dimensional multimodal functions

Table 4 presents the performance comparison of the proposed EnMGO, the original MGO, and the IFMGO on the high-dimensional multimodal test functions. This shows the algorithms' capabilities of solving optimization problems that require effective balancing of the exploration and exploitation mechanisms.

The simulation outcomes of the three (3) algorithms for the multimodal functions presented in Table 4 indicate very competitive performances among the algorithms. In the case of F8, the EnMGO, MGO, and IFMGO all produced the global solution with the best results. However, the EnMGO slightly outperformed the MGO and the IFMGO in terms of the standard deviation (STD). All the algorithms produced the global solutions in the cases of F9 and F11, while in F10 and F13, the EnMGO outperformed the MGO and the IFMGO by exhibiting better performances. In the case of F12, the IFMGO outperformed the EnMGO and the original MGO. In all, the proposed EnMGO has shown good performance on all the functions but F12, indicating its potential in solving high-dimensional multimodal optimization problems. It is relevant to acknowledge that the MGO and IFMGO also produced very competitive results across the 6 multimodal functions.

Table 4. Results on multimodal functions (high-dimensional).

Function	**Parameter**	**EnMGO**	**MGO** [13]	**IFMGO** [15]
F8	Best	**-12570**	**-12570**	**-12570**
	Worst	**-12570**	**-12570**	**-12570**
	Mean	**-12570**	**-12570**	**-12570**
	STD	**1.0984E-11**	3.9992E-8	1.7527E-8
F9	Best	**0**	**0**	**0**
	Worst	**0**	**0**	**0**
	Mean	**0**	**0**	**0**
	STD	**0**	**0**	**0**
F10	Best	**8.882E-16**	**8.882E-16**	**8.882E-16**
	Worst	**8.882E-16**	4.441E-15	4.441E-15
	Mean	**8.882E-16**	1.717E-15	1.007E-15
	STD	**0**	1.528E-15	6.486E-16
F11	Best	**0**	**0**	**0**
	Worst	**0**	**0**	**0**
	Mean	**0**	**0**	**0**
	STD	**0**	**0**	**0**
F12	Best	1.7480E-32	**1.571E-32**	**1.5705E-32**
	Worst	7.3642E-27	2.196E-25	**1.6916E-32**
	Mean	3.6821E-27	1.697E-26	**1.6313E-32**
	STD	**2.0407E-37**	4.538E-26	4.417E-34
F13	Best	**1.3498E-32**	**1.3498E-32**	**1.3498E-32**
	Worst	**1.3175E-32**	6.403E-32	3.569E-32
	Mean	**1.3531E-32**	1.814E-32	1.562E-32
	STD	**7.3617E-35**	9.954E-33	5.319E-33

The three algorithms (EnMGO, MGO, and IFMGO) are further compared in terms of their convergence processes in the following curves (Figures 13–18), illustrated. The maximum iteration is set to 5000 to provide enough room for the algorithms to converge towards the global solutions of the benchmark functions (F8 to F13).

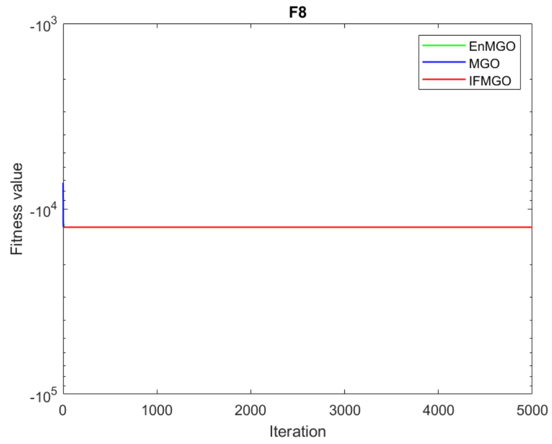

Figure 13. Convergence characteristics of function F8.

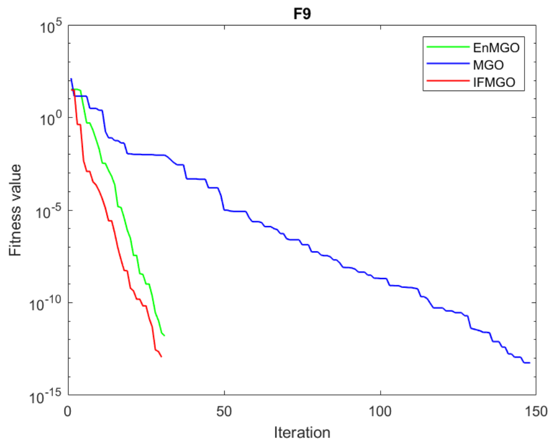

Figure 14. Convergence characteristics of function F9.

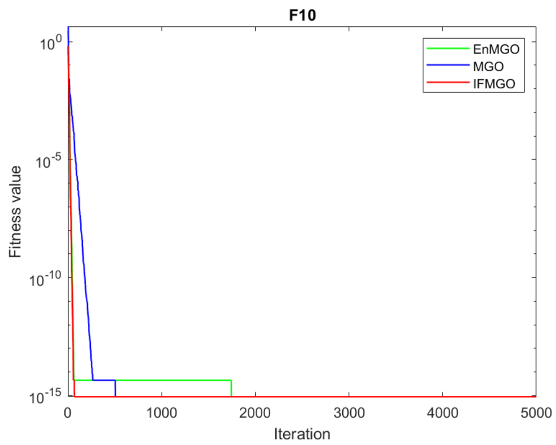

Figure 15. Convergence characteristics of function F10.

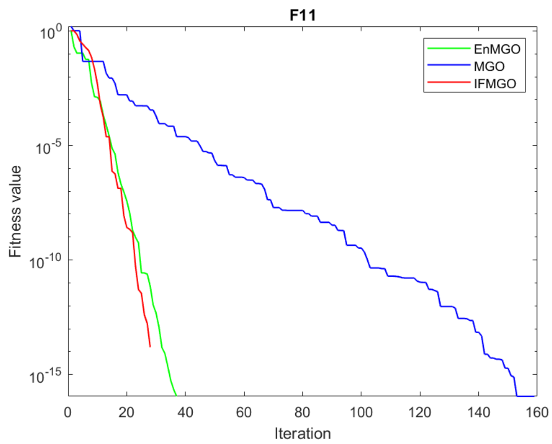

Figure 16. Convergence characteristics of function F11.

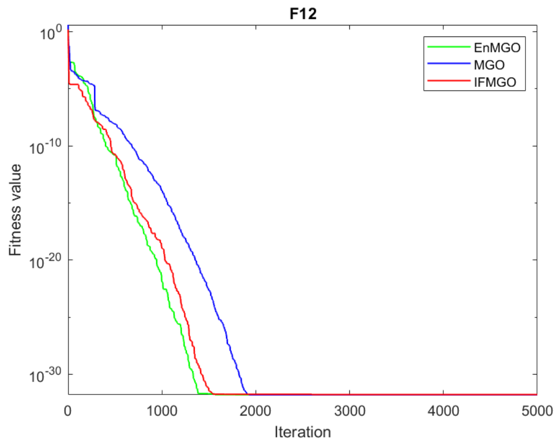

Figure 17. Convergence characteristics of function F12.

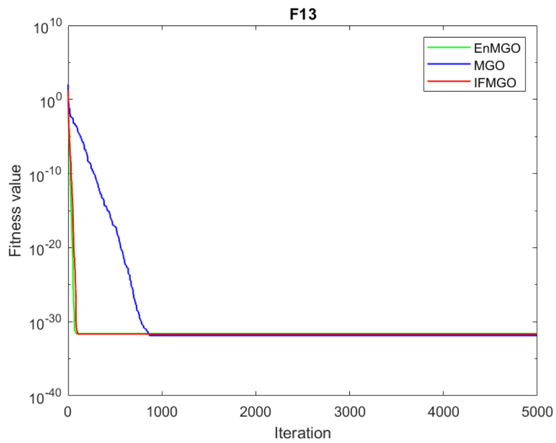

Figure 18. Convergence characteristics of function F13.

All the algorithms showed very close convergence characteristics, almost 100% the same, leading to overlapping curves in F8. In F9–F13, the EnMGO and IFMGO algorithms exhibit a closer convergence process than that of the MGO. However, all three algorithms converged to the global solutions within reasonable iterations.

4.1.2. Performance on fixed-dimensional multimodal functions

In this section, we present the performance of the proposed EnMGO, MGO, and IFMGO in Table 5. These algorithms are tested on 10 standard fixed-high-dimensional multimodal functions, and their simulation results are compared.

Table 5. Results on multimodal functions (fixed-dimensional).

Function	Parameter	EnMGO	MGO [13]	IFMGO [15]
F14	Best	**0.998**	**0.998**	**0.998**
	Worst	**0.998**	**0.998**	**0.998**
	Mean	**0.998**	**0.998**	**0.998**
	STD	**2.4236E-22**	1.8440E-16	5.9168E-17
F15	Best	**3.0561E-4**	3.0749E-4	3.0605E-4
	Worst	**7.1565E-4**	1.2232E-3	1.2343E-3
	Mean	**3.7585E-4**	3.7059E-4	3.0779E-4
	STD	**1.2405E-4**	2.3182E-4	1.9014E-4
F16	Best	**-1.03160**	**-1.03160**	**-1.03160**
	Worst	**-1.03160**	**-1.03160**	**-1.03160**
	Mean	**-1.03160**	**-1.03160**	**-1.03160**
	STD	**1.2211E-18**	4.7908E-16	6.9914E-17
F17	Best	**0.3980**	**0.39789**	**0.39789**
	Worst	**0.3980**	**0.39789**	**0.39789**
	Mean	**0.3980**	**0.39789**	**0.39789**
	STD	**0**	**0**	**0**
F18	Best	**3**	**3**	**3**
	Worst	**3**	**3**	**3**
	Mean	**3**	**3**	**3**
	STD	**1.7973E-25**	1.4092E-15	1.1019E-15
F19	Best	**-3.8621**	-3.8628	-3.8628
	Worst	**-3.8615**	-3.8628	-3.8628
	Mean	**-3.8627**	-3.8628	-3.8628
	STD	**1.8912E-24**	2.2584E-15	1.8436E-15
F20	Best	**-3.3218**	-3.3220	-3.3220
	Worst	**-3.1824**	-3.3220	-3.3220
	Mean	**-3.2417**	-3.3220	-3.3220
	STD	5.9552E-2	6.033E-2	**0**
F21	Best	**-10.15**	**-10.15**	**-10.15**
	Worst	**-10.15**	**-10.15**	**-10.15**
	Mean	**-10.15**	**-10.15**	**-10.15**
	STD	**0**	**0**	**0**
F22	Best	**-1.04028E+01**	**-**10.403	-10.403
	Worst	**-1.04027E+01**	-10.403	-10.403
	Mean	**-1.04028E+01**	-10.403	-10.403
	STD	3.9324E-5	**0**	**0**
F23	Best	**-1.05363E+1**	-10.536	-10.536
	Worst	**-1.05361E+01**	-10.536	-10.536
	Mean	**-1.05363E+01**	-10.536	-10.536
	STD	4.7387E-05	**0**	**0**

The results for functions F14, F16–F18, and F22 show that all the algorithms produced global solutions, which indicates that can solve optimization problems with similar characteristics to the functions. However, in the cases of functions F15, F19, F20, and F22, none of the algorithms produced the global solutions, but the proposed EnMGO produced relatively better results compared to the MGO and IFMGO. The EnMGO further exhibited superiority by producing the global solution of function F23. It is worth noting that though EnMGO generally outperformed the other algorithms, they all performed closely with very competitive results. The EnMGO has generally produced very good performance on all 10 fixed-dimensional multimodal test functions.

The convergence characteristics of the EnMGO, MGO, and IFMGO for F14–F23 exhibited very close convergence behavior toward the global solutions, producing overlapping curves. Hence, it was impossible to display them clearly. These common qualities of the three algorithms for these 10 functions agree with the very competitive or close results presented in Table 5. Therefore, the 3 algorithms are potential optimization tools for solving fixed-dimensional multimodal problems.

The holistic comparison of the 3 algorithms for the 23 standard benchmark test functions, as presented in Tables 3 and 5, shows that the proposed modified algorithms (EnMGO) produced very good performance for all the test functions except functions F5, F6, and F12. This performance represents 20 out of the 23 test functions, a very good indicator of about 86.96%. For the situations of functions F5, F6, and F12, the proposed EnMGO was outperformed by its counterparts. However, the performance of the EnMGO on these 3 test functions is close to that of the others. Hence, the overall performance of the proposed EnMGO is superior compared to the original MGO and IFMGO.

4.2. Performance on engineering design problems

The performance of the EnMGO on the 2 engineering design problems, the pressure vessel design problem, and the three-bar truss design problem, is compared to that of the MGO and IFMGO presented in reference [19]. The performances of the three algorithms on the engineering design problems are compared in Tables 6 and 7, respectively. The EnMGO was tested 30 times for each of the design problems, and the best results were recorded for the comparison.

Table 6. Performances on the pressure vessel design problem.

Algorithm	X1	X2	X3	X4	F(x)
MGO	0.8947789	0.4404352	46.3616	130.1125	**6108.9319**
IFMGO	0.7805281	0.386001	40.4248	198.799	**5897.7704**
EnMGO	0.7822144	0.3851635	40.52622	197.1715	**5888.9498**

Table 6 presents the performance of the 3 algorithms, where the function values (f(x)) represent the manufacturing cost of the pressure vessel produced by each corresponding algorithm. The proposed EnMGO produced the lowest cost value of 5888.998, while the original MGO produced the highest cost value of 6108.9319. The IFMGO produced a very competitive cost value of 5897.770 compared to that of the proposed EnMGO. However, the EnMGO outperformed the other two algorithms.

The test result of the EnMGO on the three-bar truss design problem is also compared to that of the MGO and FIMGO in Table 7 below.

Table 7. Performance on the three-bar truss design problem.

Algorithm	X1	X2	F(x)
MGO	0.78534	0.41775	**263.9041**
IFMGO	0.78845	0.4089	**263.8959**
EnMGO	0.78918	0.40682	**263.8957**

The results in Table 7 show very close performances among the 3 algorithms. Each produced optimal function had values around 263. This indicates that they are all good candidate tools for solving optimization problems similar to the three-bar truss design problem. However, a closer look at the function values indicates minute differences among them, with the EnMGO producing the least, followed by the IFMGO and the MGO.

5. Conclusions

A new variant of the MGO called the EnMGO has been developed for better performance on optimization problems. Two key modifications were introduced to address relevant weaknesses that substantially enhance the performance of the algorithm. First, a NEIW strategy is introduced to determine the values of the F parameter for the sole purpose of improving exploitation for better performance. The second modification introduced a CRIW to improve the exploration capability of the algorithm. The proposed EnMGO was tested on standard benchmark functions for testing purposes, and the outcome was benchmarked against those of the original MGO and the IFMGO reported in the literature for the same functions. The new EnMGO showed very good performance on 20 test functions of the 23 standard test functions used, and this represents a significant performance of about 86.96%. The EnMGO was further tested on 2 engineering design problems (the pressure vessel design problem and the three-bar truss design problem). The outcome was compared to that of the MGO and IFMGO, and the performance was relatively good. Based on the remarkable performances exhibited by the proposed EnMGO on the standard benchmark test functions and the engineering design problems, we conclude that the modifications introduced have led to significant improvement in the algorithm. The EnMGO is therefore recommended to be adopted for solving engineering optimization problems, such as the optimization of power distribution systems compensations, and optimal renewable energy integration for effective power delivery. Its potential application can also be extended to other fields such as agriculture and mechanical engineering designs. An example of such potential applications is the truss structures optimization [24] in which the design of truss structures is optimized for maximum performance.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare no conflict of interest.

References

1. E. Osaba, E. Villar-Rodriguez, J. Del Ser, A. Nebro, D. Molina, A. LaTorre, et al., A tutorial on the design, experimentation and application of metaheuristic algorithms to real-World optimization problems, *Swarm Evol. Comput.*, **64** (2021), 100888. https://doi.org/10.1016/j.swevo.2021.100888
2. V. S. Prakash, V. Vinothina, K. Kalaiselvi, K. Velusamy, An improved bacterial colony optimization using opposition-based learning for data clustering, *Cluster Comput.*, **25** (2022), 4009–4025. https://doi.org/10.1007/s10586-022-03633-z

3. S. M. Almufti, A. Ahmad Shaban, Z. Arif Ali, R. Ismael Ali, J. A. Dela Fuente, R. R. Asaad, Overview of metaheuristic algorithms, *Global Journal of Scholarly Research and Trends*, **2** (2023), 10–32. https://doi.org/10.58429/pgjsrt.v2n2a144
4. B. A. Kumar, B. Jyothi, A. Singh, M. Bajaj, R. Rathore, M. Tuka, Hybrid genetic algorithm-simulated annealing based electric vehicle charging station placement for optimizing distribution network resilience, *Sci. Rep.*, **14** (2024), 7637. https://doi.org/10.1038/s41598-024-58024-8
5. T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, A. Cosar, A survey on new generation metaheuristic algorithms, *Comput. Ind. Eng.*, **137** (2019), 106040. https://doi.org/10.1016/j.cie.2019.106040
6. S. Katoch, S. S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future, *Multimed. Tools Appl.*, **80** (2021), 8091–8126. https://doi.org/10.1007/s11042-020-10139-6
7. A. G. Gad, Particle swarm optimization algorithm and its applications: a systematic review, *Arch. Computat. Methods Eng.*, **29** (2022), 2531–2561. https://doi.org/10.1007/s11831-021-09694-4
8. S. Makhadmeh, M. Al-Betar, I. Doush, M. Awadallah, S. Kassaymeh, S. Mirjalili, et al., Recent advances in Grey Wolf Optimizer, its versions and applications: review, *IEEE Access*, **12** (2023), 22991–23028. https://doi.org/10.1109/ACCESS.2023.3304889
9. E. Kaya, B. Gorkemli, B. Akay, D. Karaboga, A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems, *Eng. Appl. Artif. Intel.*, **115** (2022), 105311. https://doi.org/10.1016/j.engappai.2022.105311
10. A. Yussif, S. Adjei, B. Wilson, Sigmoid-function-based adaptive pelican optimization algorithm for global optimization, *IJEEAS*, **7** (2024), 15–25. https://doi.org/10.54554/ijeeas.2024.7.02.003
11. S. Syama, J. Ramprabhakar, R. Anand, J. M. Guerrero, A hybrid extreme learning machine model with lévy flight chaotic whale optimization algorithm for wind speed forecasting, *Results in Engineering*, **19** (2023), 101274. https://doi.org/10.1016/j.rineng.2023.101274
12. V. Tomar, M. Bansal, P. Singh, Metaheuristic algorithms for optimization: a brief review, *Eng. Proc.*, **59** (2023), 238. https://doi.org/10.3390/engproc2023059238
13. B. Abdollahzadeh, F. S. Gharehchopogh, N. Khodadadi, S. Mirjalili, Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems, *Adv. Eng. Softw.*, **174** (2022), 103282. https://doi.org/10.1016/j.advengsoft.2022.103282
14. G. Nayeem, M. Fan, Y. Akhter, A time-varying adaptive inertia weight based modified PSO algorithm for UAV path planning, *Proceedings of the 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, 2021, 573–576. https://doi.org/10.1109/ICREST51555.2021.9331101
15. T. Seini, A. Yussif, I. M. Katali, Enhancing mountain gazelle optimizer (MGO) with an improved F-parameter for global optimization, *Int. Res. J. Eng. Technol*, **10** (2023), 921–930.
16. J. Bansal, P. Singh, M. Saraswat, A. Verma, S. Jadon, A. Abraham, Inertia weight strategies in particle swarm optimization, *Proceedings of the Third World Congress on Nature and Biologically Inspired Computing*, 2011, 640–647. https://doi.org/10.1109/NaBIC.2011.6089659
17. R. Naik, U. Singh, A review on applications of chaotic maps in pseudo-random number generators and encryption, *Ann. Data Sci.*, **11** (2024), 25–50. https://doi.org/10.1007/s40745-021-00364-7
18. A. Seini Yussif, E. Twumasi, E. Asuming Frimpong, Modified mountain gazelle optimizer based on logistic chaotic mapping and truncation selection, *Int. Res. J. Eng. Technol*, **10** (2023), 1769–1776.
19. A. Seini Yussif, T. Seini, Improved F-parameter mountain gazelle optimizer (IFMGO): a comparative analysis on engineering design problems, *Int. Res. J. Eng. Technol.*, **10** (2023), 810–816.

20. M. Jain, V. Saihjpal, N. Singh, S. B. Singh, An overview of variants and advancements of PSO algorithm, *Appl. Sci.*, **12** (2022), 8392. https://doi.org/10.3390/app12178392
21. W. Aribowo, L. Abualigah, D. Oliva, A. Prapanca, A novel modified mountain gazelle optimizer for tuning parameter proportional integral derivative of DC motor, *BEEI*, **13** (2024), 745–752. https://doi.org/10.11591/eei.v13i2.5575
22. P. Sarangi, P. Mohapatra, Evolved opposition-based mountain gazelle optimizer to solve optimization problems, *J. King Saud Univ.-Com.*, **35** (2023), 101812. https://doi.org/10.1016/j.jksuci.2023.101812
23. P. Trojovský, M. Dehghani, Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications, *Sensors*, **22** (2022), 855. https://doi.org/10.3390/s22030855
24. N. Khodadadi, E. S. M. El-Kenawy, F. De Caso, A. H. Alharbi, D. S. Khafaga, A. Nanni, The mountain gazelle optimizer for truss structures optimization, *Applied Computing and Intelligence*, **3** (2023), 116–144. https://doi.org/10.3934/aci.2023007