



# GMO: geometric mean optimizer for solving engineering problems

Farshad Rezaei<sup>1</sup> · Hamid R. Safavi<sup>1</sup> · Mohamed Abd Elaziz<sup>2,3,4,5</sup>  · Seyedali Mirjalili<sup>6,7</sup>

Accepted: 30 March 2023 / Published online: 25 April 2023

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

## Abstract

This paper introduces a new meta-heuristic technique, named geometric mean optimizer (GMO) that emulates the unique properties of the geometric mean operator in mathematics. This operator can simultaneously evaluate the fitness and diversity of the search agents in the search space. In GMO, the geometric mean of the scaled objective values of a certain agent's opposites is assigned to that agent as its weight representing its overall eligibility to guide the other agents in the search process when solving an optimization problem. Furthermore, the GMO has no parameter to tune, contributing its results to be highly reliable. The competence of the GMO in solving optimization problems is verified via implementation on 52 standard benchmark test problems including 23 classical test functions, 29 CEC2017 test functions as well as nine constrained engineering problems. The results presented by the GMO are then compared with those offered by several newly proposed and popular meta-heuristic algorithms. The results demonstrate that the GMO significantly outperforms its competitors on a vast range of the problems. Source codes of GMO are publicly available at <https://github.com/farshad-rezaei1/GMO>.

**Keywords** Global optimization · Meta-heuristic technique · Geometric mean optimizer · Fuzzy logic

## 1 Introduction

Optimization techniques in the field of evolutionary computation can be categorized into two classes: individual-based versus population-based techniques. Individual-based techniques are set to begin the process of optimization by a single search agent usually generated randomly in a search space attempting to find the global optimum of the problem. This category of algorithms benefits from some advantages and suffers from some disadvantages. The main shortcoming is premature convergence emerging in solving problems with unimodal and multi-modal search landscapes. In unimodal problems, premature convergence usually occurs when the algorithm converges to the global optimum very slowly, while in multi-modal optimization problems, the local optima entrapment is known as the main reason for premature convergence (Mirjalili 2015). Indeed, individual-based algorithms begin the optimization from an initial random point in the search space and usually can only search for the optimum solution in the proximity of that initial random point, without being able of escaping from that region to search other promising regions in the search space. On the contrary, population-based techniques start the optimization process by generating a set of initial

---

✉ Hamid R. Safavi  
hasafavi@iut.ac.ir

✉ Mohamed Abd Elaziz  
abd\_el\_aziz\_m@yahoo.com

Farshad Rezaei  
f.rezaei@alumni.iut.ac.ir

Seyedali Mirjalili  
ali.mirjalili@laureate.edu.au

<sup>1</sup> Department of Civil Engineering, Isfahan University of Technology, Isfahan, Iran

<sup>2</sup> Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt

<sup>3</sup> Artificial Intelligence Research Center (AIRC), Ajman University, P.O. Box 346, Ajman, United Arab Emirates

<sup>4</sup> Faculty of Computer Science and Engineering, Galala University, Suez 435611, Egypt

<sup>5</sup> Department of Electrical and Computer Engineering, Lebanese American University, Byblos 13-5053, Lebanon

<sup>6</sup> Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Brisbane, Australia

<sup>7</sup> YFL (Yonsei Frontier Lab), Yonsei University, Seoul, Korea

random solutions and improving their quality by lapse of iterations. This category of the techniques is of less probability to be stuck in the local optima in multi-modal problems and also conducts the search process more rapidly than the first category when solving the unimodal problems. However, more computational cost as a result of evaluating much more objective functions are two substantial disadvantages this category suffers from as compared to the individual-based problems. Some of the popular individual-based algorithms include single candidate optimizer (Shami et al. 2022).

Moreover, the optimization approaches can also be laid in two different categories: stochastic versus deterministic approaches. Stochastic approaches do not need the derivation of the mathematical functions, as these algorithms frequently change the variables and evaluate the resulting objective functions to reach the best-fitted variables at the framework of a solution. Stochastic algorithms have also lower chances to fall in local optima than the deterministic algorithms (Paper 2015).

Evolutionary algorithms mainly tend to be stochastic and population-based, so they are considered to be the best options to solve any optimization problem with any degree of complexity in variable domain and in the objective function. Some of the most popular and widely used evolutionary algorithms are genetic algorithm (GA) (Goldberg and Holland 1988), and differential evolution (DE) (Storn and Price 1997). It is worth mentioning that the no free lunch (NFL) theorem (Wolpert and Macready 1997) let new evolutionary algorithms be created, since according to this theory, the performances of all algorithms are the same when implemented on several optimization problems. Thus, one algorithm may be highly effective when solving some optional series of problems but is absolutely ineffective on some other series. In the past decade, the researchers have proposed numerous new evolutionary algorithms as required by the NFL theorem. These algorithms can be broken down into four main categories:

- (1) Swarm-based algorithms: cuckoo search (CS) (Gandomi et al. 2013), bat algorithm (BA) (Yang 2010), grey wolf optimizer (GWO) (Mirjalili et al. 2014), moth-flame optimization (MFO) [1], Harris Hawks optimizer (HHO) (Heidari et al. 2019), whale optimization algorithm (WOA) (Mirjalili and Lewis 2016), gradient-based optimizer (GBO) (Ahmadianfar et al. 2020), and Aquila optimizer (AO) (Abualigah et al. 2021c).
- (2) Physics-based algorithms: gravitational search algorithm (GSA) (Rashedi et al. 2009), galaxy-based search algorithm (GbSA) (Hosseini 2011), black hole (BH) algorithm (Hatamlou 2013), colliding bodies optimization (CBO) (Kaveh and Mahdavi 2014),

equilibrium optimizer (EO) (Faramarzi et al. 2020), and flow direction algorithm (FDA) (Karami et al. 2021).

- (3) Human-based algorithms: group search optimizer (GSO) (He et al. 2009), interior search algorithm (ISA) (Gandomi 2014), nomadic people optimizer (NPO) (Salih and Alsewari 2020), and teaching–learning-based optimization (TLBO) (Rao et al. 2011).
- (4) Evolutionary techniques: gradient evolution (GE) (Kuo and Zulvia 2015), biogeography-based optimization (BBO) (Simon and Member 2008), and space transformation search (Wang et al. 2009).

Every optimization algorithm must accomplish two missions to solve an optimization problem: exploration and exploitation. At the exploration phase, the positions of the search agents are frequently varied while kept diversified to be able of searching as broadly as possible. When transiting from the exploration to the exploitation stage, the search agents must gradually avoid further moving in the space and intensify locally searching their possibly high-fitness neighborhood. Every evolutionary algorithm must be able to hold a balance between these two key phases of any optimization process to make this process be properly and accurately done without leaving any chance for the premature convergence to occur (El et al. 2017).

As it is known, in a variety of the other meta-heuristics, there are two separate mechanisms to do exploration and exploitation, and thus, these two phases may be inconsistent, as a solution may have high fitness but low diversity or is of high diversity but low fitness. In this case, deciding on which particle should be selected as a guide may be very difficult and even if be done, this selection may disrupt the exploration–exploitation balance at numerous iterations of the algorithm. So, there is a need to design an index that is able to simultaneously evaluate fitness and diversity of the solutions as two major factors to be considered during the optimization process. As is clear, the high fitness is a major player to choose the guide of the search agents in the exploitation phase of the optimization process, and the diversity is the major factor helping designate the guide of the search agents in the exploration step. The integration of the abilities to evaluate these two major factors can facilitate the process of selection of the guides such that the best possible guides can be selected by the optimizer. The importance of this issue shows itself better when knowing that it is impossible to delineate a certain boundary between the exploration process and the exploitation step over the course of iterations of an optimization process. Accordingly, the major motivation of the authors of this paper is to introduce a new optimization algorithm while

benefitting from an index evaluating fitness and diversity at the same time.

As mentioned above, this paper proposes a novel algorithm, called geometric mean optimizer (GMO). The geometric mean operator in mathematics is found so effective to enhance the abilities which an optimization algorithm should have. Utilizing this operator, the GMO algorithm can deal with both diversity of the search agents and the fitness of them at the same time by evaluating a single simple index, named dual-fitness index (DFI). This index is calculated for all agents in the population and used to define a local guide for each search agent during its search process. The introduction of the DFI in the proposed GMO can make knowing the phase of the optimization (exploration or exploitation) unnecessary and neutral during the optimization process that is being on its right way to be accomplished despite ambiguity in the boundary between the exploration and exploitation phases of an optimization process. Designing the DFI index in GMO is the major novelty of this algorithm against any other meta-heuristic. Among the other advantages of the GMO, there are multiple unique guides designated for the search agents at the proper distances from them in the GMO to firstly help them avoid getting trapped in local optima, and secondly not engaged in a drift causing to lose a large number of good solutions when fast moving toward their guides. Moreover, a Gaussian mutation is applied on the guides in the GMO to further give stochastic nature to the search process of this algorithm as a way to better explore the search space of any optimization problem. At last but not least, the GMO has no parameter to tune, contributing its results to be highly reliable.

The organization of the rest of this paper can be summarized as follows: Sect. 2 is dedicated to present the theory of the GMO, its unique properties helping the GMO to effectively solve the optimization problems, and the steps to implement GMO in detail. In Sect. 3, the results of testing the performance of the proposed algorithm and its rivals on numerous benchmark test functions are introduced. In this section, a statistical analysis is also presented to benchmark how significant the results the proposed algorithm presents are when become superior to its competitors. A qualitative analysis is also conducted in this section to further investigate the convergence behavior of the GMO. In Sect. 4, the results of solving several engineering problems by the GMO are compared to those offered by a large number of other optimizers. Section 5 highlights the strengths and weaknesses of the proposed GMO algorithm. Finally, in Sect. 6, the major conclusions are introduced.

## 2 Methodology

### 2.1 Geometric mean optimizer (GMO)

GMO is a meta-heuristic technique, utilizing the behavior of several search agents when socially interacting with each other. The way these agents should interact to yield the best result must be delineated in any optimization algorithm. Here, the unique mathematical properties of the GMO algorithm when handling optimization problems are first explained and then the detailed formulation by help of which this algorithm solves the problems is presented. Assume  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  and  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$  are the  $i$ th agent's position and velocity, respectively. According to the requirements mentioned in Sect. 1 for solving the optimization problems, the GMO can simultaneously evaluate the fitness and diversity of the search agents in the search space by a single index, named the dual-fitness index (DFI). The way to calculate the DFI is presented in the remainder of this section in detail. Referring to the fuzzy logic concepts, first introduced by Zadeh (Zadeh 1965), the similarity among several variables can be expressed by multiplying the fuzzy membership function (MF) values attributed to those variables. This multiplication is known as the product-based Larsen implication function, vastly applied in the fuzzy systems. In contrast, the geometric average of  $n$  membership degrees (MD) can be expressed by  $\sqrt[n]{\mu_1 \times \mu_2 \times \dots \times \mu_n}$ . As a result, the multiplication of several MDs, i.e.,  $(\mu_1 \times \mu_2 \times \dots \times \mu_n)$ , can be treated as the geometric average of them, while not having the  $n$ th root, in which  $\mu_i$  is the  $i$ th MD and  $i = 1, 2, \dots, n$ . The experimental results that are presented in Sect. 3 of this paper revealed that the  $n$ th root should be neglected in the geometric average operator employed in the GMO. Therefore, this form of the geometric average named the pseudo-geometric average and calculated over the MFs of several variables, can reflect the average and similarity, simultaneously, among them.

According to some mathematical underpinnings mentioned above, the structure of GMO is to be explained here. In GMO, the comprehensive fitness of a search agent can be calculated with regard to its opposites' fitnesses. Here, the "opposite" agents of a certain agent are referred to as the set of individuals forming the whole population of agents/elite agents excepting the certain agent itself. At each iteration, the personal best-so-far agent position (the position identified to be the best position that agent has experienced till that iteration) is selected. Then, for all opposite personal best-so-far agents of a certain agent, the multiplication of the objective values, transformed into the fuzzy MFs, is calculated. It is noteworthy that in the

implementation of GMO, the fuzzy membership functions must be directly proportional to the fuzzified variables and thus must be ascending in the domain. In a minimization process, the larger the pseudo-geometric average of the MFs obtained for the opposite agents of a focused agent, the more fitted that agent will be. This is because in this case, two tasks are simultaneously fulfilled for the focused agent:

- (1) The opposite agents' MF values are averagely larger, revealing that the focused agent is of relatively less MF and thus the less objective value.
- (2) The similarity among the opposite agents' MFs/objectives is at a higher level, meaning that these agents are gathered in a closed region and thus, the diversity of them is at a lower level. Consequently, the focused agent is located in a relatively more diversified region in the search space.

Both these highlight the capability of the focused agent to have a more suitable situation in terms of both fitness and diversity together in a minimization process, as compared to another agent having less pseudo-geometric average on its opposite agents. To define the MFs, we have used the following formula.

$$f(x) = \frac{1}{1 + e^{a(x-c)}}; a < 0 \tag{1}$$

In Eq. (1),  $a$  and  $c$  stand for the parameters of the sigmoidal MF. Indeed, this function assigns an MD to “high  $x$  values” set to the variable  $x$ . Since the parameters  $a$  and  $c$  of Eq. (1) are unknown in each fuzzification effort, there is a need to tune these parameters via a trial-and-error technique. The trial and error procedure has been demonstrated to be an inexact and time-consuming process for parameter tuning in fuzzy systems. Hence, it is better to determine these parameters via other efficient methods. One of these methods can be making the fuzzy membership functions' parameters such as  $a$  and  $c$  in Eq. (1) dependent on the statistical parameters. Considering this idea, Rezaei et al. (Rezaei et al. 2017) proved that in an ascending sigmoidal MF,  $c = \mu$  and  $a = \frac{-4}{\sigma\sqrt{e}}$ ; where  $\mu$  and  $\sigma$  represent the mean and standard deviation of the  $x$  values, respectively. Also,  $e$  is Napier's constant. Substituting  $x$  by the objective function value calculated for a personal best-so-far search agent, the fuzzy MF value of this agent can be calculated according to Eq. (2).

$$MF_j^t = \frac{1}{1 + \exp\left[-\frac{4}{\sigma^t\sqrt{e}} \times (Z_{best,j}^t - \mu^t)\right]}; j = 1, 2, \dots, N \tag{2}$$

where  $Z_{best,j}^t$  is the objective function value of the  $j$ th personal best-so-far agent at the  $t$ th iteration;  $\mu^t$  and  $\sigma^t$  are the

mean and standard deviation of the objective function values of all personal best-so-far agents at the  $t$ th iteration, respectively;  $MF_j^t$  is the MF value of the  $j$ th personal best-so-far agent, and  $N$  is the population size. Then, a novel index, we name the dual-fitness index (DFI), can be calculated for a search agent as follows:

$$DFI_i^t = MF_1^t \times \dots \times MF_{i-1}^t \times MF_{i+1}^t \times \dots \times MF_N^t \\ = \prod_{\substack{j=1 \\ j \neq i}}^N MF_j^t \tag{3}$$

where  $N$  is the population size and  $DFI_i^t$  is the dual-fitness index of the  $i$ th agent at the  $t$ th iteration. For making all personal best-so-far agents contribute to generating a unique global guide agent for each agent, a weighted average of all opposite personal best-so-far agents are defined whose weights are the DFI values that are assigned to them. Equation (4) indicates the mentioned relation:

$$Y_i^t = \frac{\sum_{j=1, j \neq i}^N DFI_j^t \times X_j^{best}}{\sum_{j=1}^N DFI_j^t + \varepsilon} \tag{4}$$

where  $Y_i^t$  is the position vector of the unique global guide agent calculated for the agent  $i$  at iteration  $t$ ,  $X_j^{best}$  stands for the personal best-so-far position vector of the  $j$ th search agent, and  $DFI_j^t$  is the dual-fitness index of the  $j$ th search agent at the  $t$ th iteration. Moreover,  $\varepsilon$  is a very small positive number added to the denominator of Eq. (4) to avoid singularity. It is noteworthy that the  $\varepsilon$  is necessary to be added here only when solving the simple problems especially the problems with the optimum at the center of their domains. However, adding the  $\varepsilon$  is unnecessary when solving the more complicated problems, especially the problems with the optimum at a position away from the center of the domain. If there is no prior information about the problem decided to be solved, it is recommended not to add the  $\varepsilon$  to the denominator of Eq. (4) and consequently Eq. (5). For making the algorithm search process more effective as well as alleviating the computational burden of performing the algorithm, it is recommended to involve only the elite personal best-so-far agents in the calculation of each guide agent. For this purpose, all personal best-so-far agents are sorted from the highest  $DFI$  to the lowest  $DFI$  and the first  $N_{best}$  personal best-so-far agents are selected as the elite agents. As a simple way to set the number of  $N_{best}$ , it can be linearly decreased over the course of iterations, such that it is equal to the population size at the first iteration and is equal to 2 at the final iteration. Considering the value of 2 for  $N_{best}$  at the final iteration mainly helps make the elite agents always change

their positions to further keep diversity. Thus, Eq. (4) can be transformed to Eq. (5) when elitism is addressed in finding the guide agents.

$$Y_i^t = \frac{\sum_{j \in N_{best}, j \neq i} DFI_j^t \times X_j^{best}}{\sum_{j \in N_{best}} DFI_j^t + \varepsilon} \tag{5}$$

To make the  $Y_i^t$  more stochastic in nature which can help diversity of the guide agents be better preserved, these agents are mutated in GMO. The mutation scheme is considered to be a Gaussian mutation. The equation imposing this form of mutation on the guide agents is as follows:

$$Y_{i,mut}^t = Y_i^t + w \times randn \times (Std_{max}^t - Std^t) \tag{6}$$

where  $randn$  is a random vector generated from the standard normal distribution;  $Std^t$  is the standard deviation vector calculated for the personal best-so-far agents at the  $t$ th iteration;  $Std_{max}^t$  is a vector consisting of the maximum standard deviation values of the personal best-so-far agents' dimensions at the  $t$ th iteration, and  $w$  is a parameter decreasing the mutation step size by lapse of iterations and calculated through Eq. (9). Finally,  $Y_{i,mut}^t$  is the mutated  $Y_i^t$  used to guide the search agents. As can be seen, the more the standard deviation of a dimension in the search space, the smaller the mutation step of the guide agents to maintain the existing high diversity among the best agents, contributing to keep the whole population diversified in the search space. On the other hand, the less the standard deviation of a dimension in the search space, a larger step is taken in the mutation mechanism, to broaden the search space and enhance the diversity of the agents in the certain dimension.

Finally, the updating equations can be formulated for each search agent according to Eqs. (7) to (9):

$$\begin{aligned} V_i^{t+1} &= w \times V_i^t + \varphi \times (Y_{i,mut}^t - X_i^t); \varphi \\ &= 1 + (2 \times rand - 1) \times w \end{aligned} \tag{7}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{8}$$

$$w = 1 - \frac{t}{t_{max}} \tag{9}$$

where  $w$  is a control parameter,  $t$  is the number of current iteration, and  $t_{max}$  is the maximum number of iterations.  $V_i^t$  is the velocity vector of the  $i$ th search agent at the  $t$ th iteration,  $V_i^{t+1}$  is the velocity vector of the  $i$ th search agent at  $(t + 1)$  th iteration,  $Y_{i,mut}^t$  is the position vector of the unique global guide for the agent  $i$  at iteration  $t$  and  $X_i^t$  is the  $i$ th agent's position vector at iteration  $t$ . In addition,  $\varphi$  is a scaling parameter vector, delineating the steps the agent  $i$  takes toward its guide; and  $rand$  is a random number generated in  $[0, 1]$ . As can be seen, the length of the vector

of  $\varphi$  is decreasing, as it is varied in the ranges  $[0, 2]$ ,  $[0.1, 1.9]$ ,  $[0.2, 1.8]$ , ...,  $[0.8, 1.2]$ ,  $[0.9, 1.1]$ , and  $[1, 1] = \{1\}$ , as the iterations go on. This decreasing order considered for the  $\varphi$  ranges is able of enhancing the exploration capability of the GMO at the initial iterations as well as highly increasing the focus on the exploitation at the final iterations to help make the exploration–exploitation transition well-balanced. Figure 1 illustrates the flowchart of the GMO.

### 2.2 Differences between GMO and PSO

Within this section, we are willing to compare the strategies used by each of the GMO and PSO algorithms to solve the optimization problems to disambiguate the possible similarities between these two methods. To do so, we first explain the search process of the PSO and then mention several substantial differences there are between the two algorithms. Assume that  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  and  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$  are the vectors containing the position and velocity of the  $i$ th particle in each dimension, respectively, and  $D$  is the dimensionality of the search space. If  $Pbest_i^t = (p_{i1}, p_{i2}, \dots, p_{iD})$  is the  $i$ th personal best ( $Pbest$ ) position and  $Gbest^t = (p_{g1}, p_{g2}, \dots, p_{gD})$  represents the global best ( $Gbest$ ) position of the whole swarm, the velocity- and position-updating procedure of the particles in PSO are illustrated in Eqs. (10) and (11), respectively (Eberhart and Kennedy 1995).

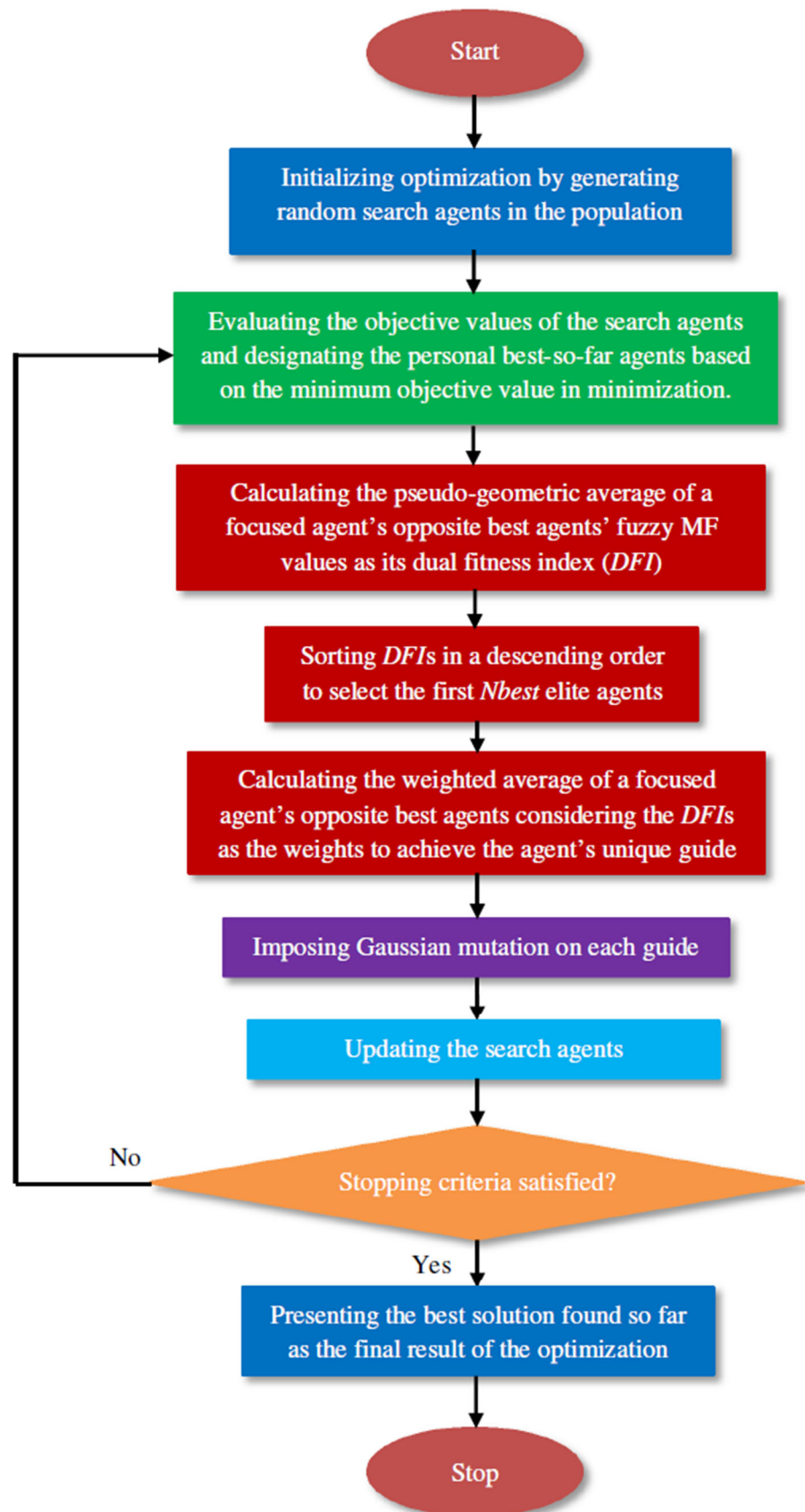
$$V_i^{t+1} = wV_i^t + c_1r_1(Pbest_i^t - X_i^t) + c_2r_2(Gbest^t - X_i^t) \tag{10}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{11}$$

where  $i \in \{1, 2, \dots, N\}$ ;  $N$  is the population size;  $t$  is the current number of iteration;  $w$  is the inertia weight;  $r_1$  and  $r_2$  are two random vectors in the interval  $[0, 1]$ , and  $c_1$  and  $c_2$  are two tunable parameters, usually set to 2. As can be inferred from the updating procedure of the PSO, each particle has some tendency to move toward both its  $Pbest$  and  $Gbest$ . The mission of the  $Pbest$ s is maintaining the particles diversified to do exploration, while the role of the  $Gbest$  is providing the particles with the access to the high-fitness regions to do exploitation. There are many substantial discrepancies between the GMO and PSO algorithms that are briefly listed below.

- (1) In GMO, a novel single index, named dual-fitness index (DFI), is defined to simultaneously evaluate the fitness and diversity of the agents. While, in a variety of the other meta-heuristics including the PSO, there are two separate mechanisms to do so, and thus, the exploration and exploitation phases may be inconsistent, as a particle may have high

Fig. 1 Steps of GMO



fitness but low diversity or is of high diversity but low fitness. In this case, deciding on which particle should be selected as a guide may be very difficult and even if be done, this selection may disrupt the exploration–exploitation balance at numerous iterations of the algorithm. Designing the DFI index in GMO is the major novelty of this developed algorithm against the PSO and any other meta-heuristic.

- (2) In PSO, each particle has two guides ( $Pbest$  and  $Gbest$ ), attracting it toward themselves. As these guides may be occasionally conflicting in attracting the particles, the particles' movements may be biased to one of these two guides and thus get unbalanced, contributing them to be stuck in local optima when the  $Gbest$  is too near to the particles or making them get into a drift if the  $Gbest$  is too far away from the particles helping numerous optima be missed by the particles. Both these behaviors can bring about the premature convergence in the PSO, while, in GMO, each search agent is assigned a single and unique global guide that is positioned at a proper distance from each agent, hindering premature convergence.
- (3) In PSO, there are two control parameters ( $c_1$  and  $c_2$ ) in the velocity updating procedure of the algorithm with two random numbers to be generated and assigned to each one, while in GMO, there is a single control parameter ( $\varphi$ ) which requires only one random number to be generated to calculate its value. As a result, the complexity of the GMO can be less than that of the PSO, making the GMO a faster algorithm than the PSO.
- (4) In PSO, there is no interaction between the particles, whereas, in GMO, the unique guide of each search agent is obtained as the average of several elite agents exchanging their information with that guide and thus can positively affect the search process of each search agent.
- (5) In PSO, there is a single guide ( $Gbest$ ) to impede the particles to be trapped in local optima especially when solving the multi-modal problems, while in GMO, multiple guides attempt to make the search agents escape from the local.
- (6) In GMO, a Gaussian mutation strategy is carefully set at the body of the algorithm to impart a more uncertain nature to the guides, especially at the initial stages of the optimization, while in PSO, there is no mutation imposed on the guide particles and thus, there is no enforcement for the particles to escape from the local optima, if trapped in them.

- (7) In GMO, a smooth transition is conducted from the exploration to the exploitation, as the search agents are gradually approaching their guides, while in PSO, the tendency of the particles to their  $Pbest$ s and  $Gbest$  is always the same over the course of iterations. This fact means that a particle may completely focus on its guides ( $Pbest$  and/or  $Gbest$ ) at the initial iterations of the PSO, while distancing itself from its guides and searching for the possible good positions within this distance at the final iterations. In this case, the exploration and exploitation procedures are not properly performed throughout the optimization process.

### 2.3 The potential merits of GMO and its computational complexity

The GMO has the ability to integrate the evaluation procedure for the fitness and diversity of the existing solutions in the search space to delineate the guide agents. This is a unique ability that discriminates this algorithm from any other meta-heuristic and is proposed to use in this algorithm for the first time. Since the measurement of the fitness and diversity follow two completely different and independent mechanisms in the meta-heuristics, there may be always a chance to choose inappropriate guide agents, leading the other agents in an unbalanced and ineffective way. If the algorithm is in the exploration phase that entails the search agents to be scattered as much as possible, the different mechanisms to lead the search agents may take effect in a way that agents are so far from the high-fitness area and their movement toward this area is retarded. In the multi-modal problems, this method to excessively stress the exploration process may prompt the agents to get trapped in local optima. Taking highly different mechanisms to lead the search agents may also gather these agents in a highly closed and densely populated region nearby the high-fitness area at the exploitation phase of the search process, speeding up the convergence but at the same time, leading to premature convergence.

The GMO attempts to change this commonly followed way to guide the search agents by defining a novel index representing the fitness and diversity of each search agent in an integrated way, thus impeding the possible disruptions in the exploration and exploitation phases of the algorithm to more effectively accomplish the optimization process. In GMO, there are multiple guide agents calculated at each of the iterations helping better keep the diversity among the solutions found during the search process. Although these guides are computed so precisely by an integrated index, their merits are still uncertain. As a solution, a Gaussian mutation procedure is applied to them

to promote their role as the guide agents. As the exploration and exploitation abilities of the search agents are made balanced in the GMO, these agents are set to gradually approach their guides, in contrast to some other meta-heuristics allowing the agents to always have a stochastic movement around their guide search agents, disregarding if the algorithm is at its early or late iterations. An elitism mechanism is also incorporated into the GMO to improve its performance and speed up its convergence to the optimum. These potential merits are quantitatively evaluated through conducting numerous comparisons between the GMO and a vast range of the other newly proposed and widely used meta-heuristic algorithms. A schematic way to conduct the optimization process in the GMO is illustrated in Fig. 2. It is worth mentioning that in this figure, the average fitness of the four separate regions is different such that:  $\overline{fitness}_{R1} \geq \overline{fitness}_{R2} \geq \overline{fitness}_{R3} \geq \overline{fitness}_{R4}$ , in which  $R$  denotes the *Region*. Accordingly, we have a relation as follows:  $DFI_1^t > DFI_3^t > DFI_5^t$ .

Computation complexity of the GMO depends on the complexity of four main sections in the algorithm: (1) fitness evaluations, (2) sorting mechanism of the elite solutions, (3) guide solution computation mechanism, and (4) updating procedure of the solutions. The computational complexity of the fitness evaluations is  $O(M \times N)$ , and the computational complexity of the sorting mechanism of the elite solutions and the guide solution computation mechanism is both  $O(M \times N^2)$ . Finally, the complexity of the updating procedure of the solutions is  $O(M \times N \times D)$ . Consequently, the computational complexity of the GMO

is  $O(M \times N \times (N + D))$ . It is worth mentioning that  $M$  is the maximum number of iterations,  $N$  is the number of search agents, and  $D$  is the number of decision variables.

### 3 Results and discussion

#### 3.1 Test problems

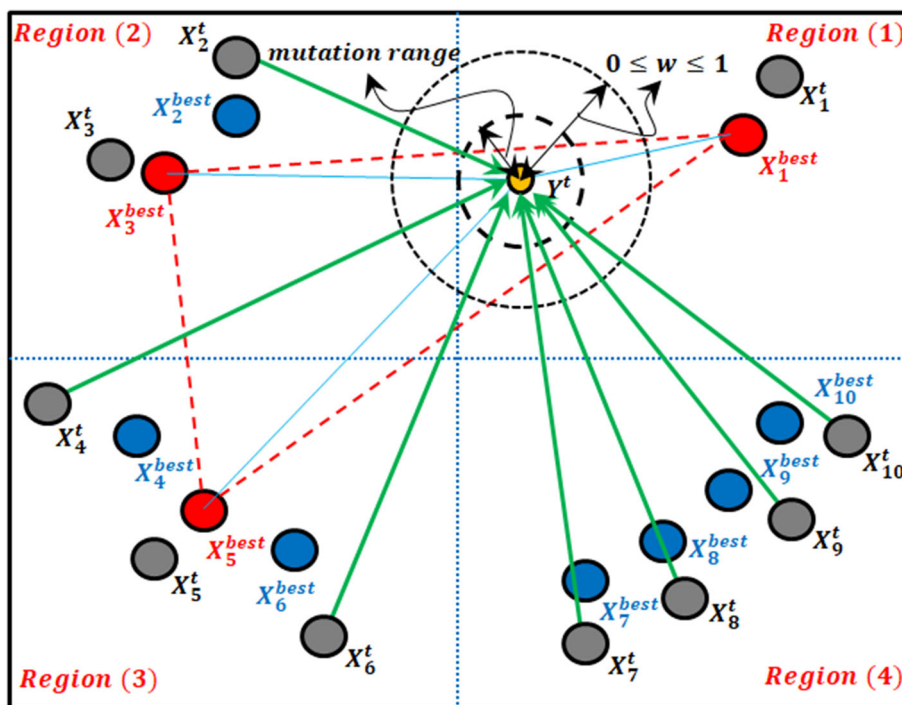
To investigate how eligible the GMO algorithm is when solving the optimization problems, it is implemented on 52 benchmark functions including 23 widely used classical benchmark test problems, and 29 CEC2017 test function suite (Yao et al. 1999; Wu et al. 2017). The aim is to minimize these benchmark functions.  $D$  is set to 30 for the classical benchmark functions and is assumed to 50 for all the functions included in the CEC2017 test suite.

#### 3.2 Comparison with other well-known algorithms

##### 3.2.1 Preparation of the algorithms

To investigate the competence of the GMO to tackle a variety of difficulties included in the optimization problems, the GMO was compared with six popular and newly proposed algorithms as the first set of the examinees. These methods include: (1) Harris Hawks optimizer (HHO) (Bui et al. 2019); (2) arithmetic optimization algorithm (AOA) (Abualigah et al. 2021a); (3) Aquila optimizer (AO)

Fig. 2 Schematic updating procedure of the search agents at the  $t$ th iteration of GMO





**Table 1** Parameter settings of the GMO and the first set of its competitive algorithms

Algorithm	Parameter settings
HHO	No parameter to tune
AOA	$\alpha = 5; \mu = 0.5$
AO	$r_1 \in [1, 20]; U = 0.00565;$ $D_1 = D; \omega = 0.005; \alpha = \delta = 0.1;$ $G_2 = [2, 0]; D = \text{number of dimensions}$
GBO	$\beta_{min} = 0.2; \beta_{max} = 1.2; pr = 0.5$
FDA	$\beta = 1$
EO	$a_1 = 2; a_2 = 1; GP = 0.5$
GMO	No parameter to tune

(Abualigah et al. 2021c), (4) gradient-based optimizer (GBO) (Ahmadianfar et al. 2020); (5) flow direction algorithm (FDA) (Karami et al. 2021); and (6) equilibrium optimizer (EO) (Faramarzi et al. 2020). The parameter settings of these algorithms as well as the proposed GMO are presented in Table 1. The population size and the maximum number of iterations of all algorithms implemented on first 23 test problems were set to 50 and 1000, except for the fixed-dimensional multi-modal problems (F14-F23), the maximum number of iterations of which was set to 500. Moreover, the number of solutions and maximum iterations was assumed to be 50 and 1000 for all algorithms when applied to the CEC2017 functions. All methods were run for 30 independent times. The final experimental results are shown in Tables 2, 3 and 4. The results corresponding to the best-performing algorithms are emboldened in these tables. The abbreviations ‘‘Ave’’ and ‘‘Std’’ in these tables stand for the average and standard deviation of the final results achieved during all runs. The convergence curves of the algorithms plotted for the classical benchmark functions are shown in Figs. 3 and 4.

### 3.2.2 Comparative results on the unimodal functions

The experimental results obtained from running algorithms on the unimodal problems indicate that GMO significantly outperforms all the comparative algorithms on 10 out of 14 (71%) of all the performance criteria. This desirable performance indicates the supreme ability of GMO to do exploitation, as having proper exploitation ability is the most important challenge in solving unimodal functions. Since strong exploration ability can also speed up the convergence to the optimal point when an algorithm attempts to solve the unimodal functions, the remarkable performance of GMO to solve this category of the benchmark functions also represents its ability to perfectly explore the search space of these functions. The assignment

of the unique and diversified guide agents to each agent and taking the diversity of the solutions into account as a major player to delineate the competence of the guide agents are among the other major factors holding diversity among the search agents and enhance the exploration capability of the proposed GMO algorithm.

The convergence curves depicted for the competitive algorithms are shown in Fig. 3. As can be inferred, the GMO algorithm is significantly faster when converging to the optimal point of all of the functions falling in this category. On F1, F3, and F4, the GMO is still willing to converge to the optimal point even after the stopping criterion is fulfilled, while its competitors seem to be stagnated at the late iterations. Furthermore, the GWO shows the fastest convergence behavior on the F6 and is the only algorithm reaching the optimal objective.

### 3.2.3 Comparative results on the multi-modal functions

The results received upon applying the algorithms to these problems indicate that the GMO is superior to the other competing methods on 7 out of 12 (58%) of the quality criteria, behaving similar to the GBO and slightly better than the AO and HHO each of which outperforms the other algorithms on 7, 6 and 6 out of 12 (58, 50, and 50%) of the criteria. It is clear that making the diversity be integrated into the main fitness of the solutions to enable the algorithm to assign the appropriate weights to each agent during the calculation of the unique guide agents could be very effective to help premature convergence have a little chance to occur when solving the multi-modal functions comprising a large number of local optima in their search space. In addition, imposing the mutation on the guide agents could further contribute the search agents to avoid local optima. The results of solving these functions are included in Table 2. The convergence curves are plotted in Fig. 4. This figure indicates the significantly faster convergence of the GMO to the optimal solution on both F10 and F13 functions, on which the GMO reaches the best solution at the end of the search process. On F9 and F11, the GMO reaches the zero value as the analytical optimal objective value of these problems, just like its competitors except for FDA and AOA on F11 and FDA on F9. The convergence rate of the GMO seems to be slightly less than the other algorithms on all the unimodal and multi-modal test problems at the early iteration; however, this rate is gradually boosted as the optimization process approaches its final stages. This behavior of the GMO is related to its high ability to diversify the search agents at the early iterations, which in turn can provide the bed to more accurately converge to the optimal point at the end of the optimization process.

**Table 2** Minimization results of the first set of the algorithms implemented on the classical benchmark functions

Criteria	HHO	AOA	AO	GBO	FDA	EO	GMO
F1	Ave 9.2006E - 193 Std <b>0.0000E + 00</b>	Ave 7.8895E - 182 Std <b>0.0000E + 00</b>	Ave 1.5675E - 204 Std <b>0.0000E + 00</b>	Ave 5.7916E - 267 Std <b>0.0000E + 00</b>	Ave 9.1388E - 12 Std 9.7219E - 12	Ave 3.8547E - 101 Std 1.8688E - 100	Ave <b>3.0138e - 322</b> Std <b>0.0000E + 00</b>
F2	Ave 5.4605E - 102 Std 1.3951E - 101	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>	Ave 3.1414E - 149 Std 1.2478E - 148	Ave 1.7127E - 134 Std 9.2323E - 134	Ave 5.3877E - 09 Std 4.6955E - 09	Ave 1.2494E - 57 Std 1.9448E - 57	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>
F3	Ave 2.8841E - 167 Std <b>0.0000E + 00</b>	Ave 8.2976E - 04 Std 3.3646E - 03	Ave 2.2004E - 207 Std <b>0.0000E + 00</b>	Ave 2.2148E - 223 Std <b>0.0000E + 00</b>	Ave 4.0472E - 01 Std 2.2927E - 01	Ave 9.4264E - 27 Std 4.9964E - 26	Ave <b>4.9407e - 324</b> Std <b>0.0000E + 00</b>
F4	Ave 2.3132E - 97 Std 6.7674E - 97	Ave 1.7737E - 02 Std 1.9819E - 02	Ave 4.2317E - 147 Std 1.6275E - 146	Ave 9.0252E - 123 Std 3.3185E - 122	Ave 1.3682E + 01 Std 2.9451E + 00	Ave 1.9210E - 25 Std 4.0403E - 25	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>
F5	Ave 6.4186E - 04 Std 7.4683E - 04	Ave 2.7855E + 01 Std 5.9786E - 01	Ave <b>3.8747E - 04</b> Std <b>6.0410E - 04</b>	Ave 1.6817E + 01 Std 2.8013E + 00	Ave 3.7711E + 01 Std 2.7420E + 01	Ave 2.3855E + 01 Std 1.8736E - 01	Ave 2.4767E + 01 Std 2.4519E - 01
F6	Ave 1.4291E - 05 Std 1.9068E - 05	Ave 2.5305E + 00 Std 2.2793E - 01	Ave 9.1910E - 06 Std 1.6930E - 05	Ave 3.1451E - 14 Std 1.2886E - 13	Ave 1.4526E - 11 Std 3.2988E - 11	Ave 7.2632E - 14 Std 1.9410E - 13	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>
F7	Ave 5.1291E - 05 Std 6.9175E - 05	Ave <b>1.7121E - 05</b> Std <b>1.3939E - 05</b>	Ave 3.1915E - 05 Std 3.0715E - 05	Ave 2.4346E - 04 Std 1.7123E - 04	Ave 5.7458E - 02 Std 2.3636E - 02	Ave 4.1963E - 04 Std 2.5443E - 04	Ave 2.1509E - 05 Std 2.5513E - 05
F8	Ave <b>-1.2569E + 04</b> Std <b>5.0935E - 02</b>	Ave -6.1649E + 03 Std 3.6264E + 02	Ave -9.7598E + 03 Std 3.5197E + 03	Ave -8.9161E + 03 Std 9.1036E + 02	Ave -8.8589E + 03 Std 6.2972E + 02	Ave -8.9007E + 03 Std 5.9972E + 02	Ave -6.4524E + 03 Std 1.2763E + 03
F9	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>	Ave <b>0.0000E + 00</b> Std 1.7723E + 01	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>
F10	Ave <b>8.8818E - 16</b> Std <b>0.0000E + 00</b>	Ave <b>8.8818E - 16</b> Std <b>0.0000E + 00</b>	Ave <b>8.8818E - 16</b> Std <b>0.0000E + 00</b>	Ave <b>8.8818E - 16</b> Std <b>0.0000E + 00</b>	Ave 3.0893E + 00 Std 1.0342E + 00	Ave 4.5593E - 15 Std 6.4863E - 16	Ave 4.4409E - 15 Std <b>0.0000E + 00</b>
F11	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>	Ave 8.9483E - 02 Std 7.0152E - 02	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>	Ave 2.2411E - 02 Std 2.6768E - 02	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>	Ave <b>0.0000E + 00</b> Std <b>0.0000E + 00</b>
F12	Ave 5.1780E - 07 Std 6.2749E - 07	Ave 3.0851E - 01 Std 4.9384E - 02	Ave 4.5872E - 07 Std 6.9013E - 07	Ave <b>1.5522E - 15</b> Std 6.8646E - 15	Ave 5.6195E - 01 Std 8.0939E - 01	Ave 2.7434E - 15 Std <b>4.9638E - 15</b>	Ave 5.2353E - 11 Std 3.2351E-11
F13	Ave 8.4075E - 06 Std 1.7001E - 05	Ave 2.7668E + 00 Std 1.3359E - 01	Ave 3.0672E - 06 Std 4.8375E - 06	Ave 6.5554E - 03 Std 1.3038E - 02	Ave 5.7681E - 02 Std 1.4114E - 01	Ave 1.4498E - 02 Std 3.3424E - 02	Ave <b>2.2873E - 10</b> Std <b>1.2323E - 10</b>
F14	Ave 1.1967E + 00 Std 4.8084E - 01	Ave 1.1029E + 01 Std 3.3093E + 00	Ave 2.1374E + 00 Std 3.1783E + 00	Ave <b>9.9800E - 01</b> Std 4.1233E - 17	Ave <b>9.9800E - 01</b> Std <b>0.0000E + 00</b>	Ave <b>9.9800E - 01</b> Std 1.3039E - 16	Ave 3.9936E + 00 Std 2.7699E + 00
F15	Ave 3.6063E - 04 Std 1.6866E - 04	Ave 1.0170E - 02 Std 1.1417E - 02	Ave 4.3820E - 04 Std 8.5052E - 05	Ave 4.6010E - 04 Std 3.4709E - 04	Ave 6.1041E - 04 Std 4.2494E - 04	Ave 3.6525E - 03 Std 7.6011E - 03	Ave <b>3.5231E - 04</b> Std <b>1.7746E - 05</b>
F16	Ave <b>-1.0316E+00</b> Std 6.3393E - 11	Ave <b>-1.0316E+00</b> Std 7.9937E - 08	Ave -1.0314E+00 Std 2.0443E - 04	Ave <b>-1.0316E+00</b> Std 6.7752E - 16	Ave <b>-1.0316E+00</b> Std 6.7752E - 16	Ave <b>-1.0316E+00</b> Std <b>6.2532E - 16</b>	Ave -1.0312E+00 Std 7.9623E - 04
F17	Ave <b>3.9789E - 01</b> Std 2.2608E - 06	Ave 4.0606E - 01 Std 6.8229E - 03	Ave 3.9801E - 01 Std 2.1139E - 04	Ave <b>3.9789E - 01</b> Std <b>0.0000E + 00</b>	Ave <b>3.9789E - 01</b> Std <b>0.0000E + 00</b>	Ave <b>3.9789E - 01</b> Std <b>0.0000E + 00</b>	Ave 4.5038E - 01 Std 1.6137E - 01
F18	Ave <b>3.0000E + 00</b> Std 2.7520E - 08	Ave 6.6006E + 00 Std 9.3349E + 00	Ave 3.0217E + 00 Std 2.4593E - 02	Ave <b>3.0000E + 00</b> Std <b>3.0000E + 00</b>	Ave <b>3.0000E + 00</b> Std <b>3.0000E + 00</b>	Ave <b>3.0000E + 00</b> Std <b>3.0000E + 00</b>	Ave <b>3.0000E + 00</b> Std 4.6526E - 11

Table 2 (continued)

	Criteria	HHO	AOA	AO	GBO	FDA	EO	GMO
F19	Ave	- 3.8619E+00	- 3.8539E+00	- 3.8584E+00	- 3.8628E+00	- 3.8628E+00	- 3.8628E+00	- <b>3.8794E+00</b>
	Std	1.2915E - 03	2.3635E - 03	3.1826E - 03	2.6962E - 15	2.7101E - 15	<b>2.5391E - 15</b>	2.0881E - 05
F20	Ave	- 3.1367E + 00	- 3.0914E + 00	- 3.2276E + 00	- 3.2665E + 00	- <b>3.3102E + 00</b>	- 3.2467E + 00	- 3.2322E + 00
	Std	1.0382E - 01	7.2463E - 02	6.6068E - 02	6.0328E - 02	<b>3.2733E - 02</b>	5.8273E - 02	1.8617E - 01
F21	Ave	- 5.2199E + 00	- 3.7338E + 00	- <b>1.0147E + 01</b>	- 7.7741E + 00	- 9.9833E + 00	- 8.7144E + 00	- 8.4704E + 00
	Std	9.0694E - 01	1.1264E + 00	<b>1.3926E - 02</b>	2.5868E + 00	9.3076E - 01	2.4632E + 00	2.9088E + 00
F22	Ave	- 5.1821E + 00	- 4.5136E + 00	- <b>1.0399E + 01</b>	- 8.6396E + 00	- 1.0003E + 01	- 9.3488E + 00	- 9.8629E + 00
	Std	1.0536E + 00	1.6562E + 00	<b>4.4476E - 03</b>	2.6167E + 00	1.5320E + 00	2.4261E + 00	2.0590E + 00
F23	Ave	- 5.1275E + 00	- 4.0556E + 00	- <b>1.0531E + 01</b>	- 8.7338E + 00	- 1.0313E + 01	- 1.0133E + 01	- 8.2832E + 00
	Std	<b>1.5118E - 03</b>	1.4526E + 00	7.5564E - 03	2.5929E + 00	1.2234E + 00	1.5454E + 00	3.5562E + 00

### 3.2.4 Comparative results on the fixed-dimension multi-modal functions

The results of applying the comparative algorithms to this category of the classical benchmark functions suggest that the proposed GMO is superior to its rivals only on 3 out of 10 (30%) of the benchmark functions. While the FDA, GBO, and EO show better performance on these problems, outperforming their competitors on 5, 4, and 4 out of 10 (50, 40, and 40%) of the test functions. The HHO and AO behave similar to the GMO, and AOA shows the worst performance among the other algorithms. As can be inferred, the proposed GMO algorithm is not so suitable for solving simple and/or low-dimensional optimization problems. Accordingly, it is recommended to employ this algorithm to solve complex, high-dimensional, and practical problems to receive the most desirable results that the other optimizers are unable to offer.

### 3.2.5 Comparative results on the CEC2017 benchmark functions

This set of the functions is a very challenging and hard-to-handle one for any optimization algorithm. The functions included in this set seem more realistic as compared to those included in the 23 classical benchmark functions, as the CEC2017 test suite comprises a variety of the composition, hybrid, unimodal, and multi-modal functions. Among these functions, the composite ones are the combination of various shifted, rotated, and biased multi-modal functions and provide a challenging environment for investigating the eligibility of the proposed algorithm in solving optimization problems. Here, all the functions included in the CEC2017 test suite are set to be 50-dimensional to further challenge the capabilities of the proposal to handle such large-scale and complex optimization problems. As the results suggest, the proposed algorithm performs far better than the other competitors on this test bed, such that the GMO can achieve the best performance criteria on 23 out of 58 (40%) of all cases, along with reaching the best averaged final results on 12 out of 29 (41%) of all cases. In detail, the GMO outperforms half of the composite functions, 2 out of 10 hybrid functions, and half of the unimodal and multi-modal functions. While only FDA has better performance than that of the GMO on the hybrid problems, outperforming the other competitors on 4 out of 10 hybrid functions. The HHO, AOA, and AO show the worst performance among the others, being able to outperform none of their rivals on the CEC2017 test suite.

The major characteristic making the proposed GMO the leading algorithm to solve such a challenging test suite is hidden in its highly enhanced exploration capability.

**Table 3** Minimization results of the first set of the algorithms implemented on the CEC2017 test suite

	Criteria	HHO	AOA	AO	GBO	FDA	EO	GMO
F1	Ave	1.2471E + 08	1.1127E + 11	1.6838E + 09	7.2920E + 03	3.8818E + 04	5.9034E + 03	<b>4.2020E + 03</b>
	Std	2.9145E + 07	8.9943E + 09	6.2457E + 08	8.4348E + 03	5.6460E + 04	<b>4.0853E + 03</b>	4.1389E + 03
F3	Ave	9.8478E + 04	1.7460E + 05	2.0248E + 05	3.9352E + 04	<b>2.7778E + 04</b>	8.1382E + 04	1.3609E + 05
	Std	2.2600E + 04	1.7459E + 04	4.4288E + 04	9.7975E + 03	<b>9.1622E + 03</b>	1.2685E + 04	1.7457E + 04
F4	Ave	7.6510E + 02	3.3253E + 04	1.0630E + 03	5.5786E + 02	5.3390E + 02	<b>5.3319E + 02</b>	5.8965E + 02
	Std	8.6911E + 01	5.8561E + 03	1.9549E + 02	5.3041E + 01	4.8907E + 01	4.4751E + 01	<b>4.3998E + 01</b>
F5	Ave	9.0007E + 02	1.1490E + 03	8.6665E + 02	8.1004E + 02	8.3679E + 02	6.7857E + 02	<b>6.2615E + 02</b>
	Std	3.2695E + 01	4.4583E + 01	3.3186E + 01	4.9971E + 01	5.2911E + 01	3.2609E + 01	<b>2.9495E + 01</b>
F6	Ave	6.7481E + 02	6.9431E + 02	6.6537E + 02	6.3725E + 02	6.5672E + 02	<b>6.0158E + 02</b>	6.1310E + 02
	Std	4.5326E + 00	5.6010E + 00	5.3658E + 00	9.7671E + 00	8.1625E + 00	<b>1.1725E + 00</b>	6.8630E + 00
F7	Ave	1.8318E + 03	1.9330E + 03	1.5161E + 03	1.2644E + 03	1.3949E + 03	9.7215E + 02	<b>8.8476E + 02</b>
	Std	9.1764E + 01	7.2198E + 01	1.2247E + 02	9.4996E + 01	1.0256E + 02	7.8489E + 01	<b>3.7289E + 01</b>
F8	Ave	1.1852E + 03	1.4682E + 03	1.1804E + 03	1.1110E + 03	1.1360E + 03	9.8236E + 02	<b>9.5240E + 02</b>
	Std	3.5456E + 01	4.7037E + 01	<b>3.4016E + 01</b>	5.6324E + 01	5.4639E + 01	4.1611E + 01	8.4595E + 01
F9	Ave	2.6187E + 04	2.9297E + 04	2.1934E + 04	7.4002E + 03	1.0338E + 04	2.4907E + 03	<b>1.5896E + 03</b>
	Std	2.7992E + 03	3.5386E + 03	3.8379E + 03	2.4743E + 03	2.2436E + 03	2.1668E + 03	<b>1.4550E + 03</b>
F10	Ave	9.2152E + 03	1.3223E + 04	9.1459E + 03	<b>7.8672E + 03</b>	8.7429E + 03	8.2564E + 03	8.7748E + 03
	Std	9.3959E + 02	<b>8.2921E + 02</b>	9.7917E + 02	9.4913E + 02	1.0974E + 03	1.1778E + 03	2.5236E + 03
F11	Ave	1.5752E + 03	2.2166E + 04	2.2437E + 03	1.3879E + 03	1.3733E + 03	<b>1.3187E + 03</b>	1.3630E + 03
	Std	7.4192E + 01	3.4432E + 03	2.7895E + 02	8.3613E + 01	8.8692E + 01	6.5704E + 01	<b>4.3463E + 01</b>
F12	Ave	1.3152E + 08	6.7197E + 10	6.1213E + 08	2.6664E + 06	<b>2.3629E + 06</b>	3.4670E + 06	3.3511E + 07
	Std	7.8725E + 07	1.0985E + 10	4.0105E + 08	2.2857E + 06	<b>1.5430E + 06</b>	1.9766E + 06	2.1415E + 07
F13	Ave	2.9929E + 06	3.4176E + 10	2.3863E + 07	1.2570E + 04	<b>7.0098E + 03</b>	8.5108E + 03	5.8217E + 04
	Std	8.5038E + 05	1.0233E + 10	3.8111E + 07	9.9819E + 03	6.4160E + 03	<b>5.4644E + 03</b>	3.0336E + 04
F14	Ave	1.6473E + 06	3.7757E + 07	4.8743E + 06	3.9591E + 04	<b>3.0465E + 04</b>	1.3926E + 05	1.1122E + 05
	Std	1.0895E + 06	3.2667E + 07	4.3827E + 06	3.9226E + 04	<b>3.1819E + 04</b>	7.8283E + 04	7.8991E + 04
F15	Ave	6.4801E + 05	3.4101E + 09	6.1620E + 05	1.2206E + 04	<b>1.0467E + 04</b>	1.2769E + 04	2.9686E + 04
	Std	2.7817E + 05	2.1087E + 09	3.6634E + 05	7.4912E + 03	<b>6.8212E + 03</b>	7.2382E + 03	2.3133E + 04
F16	Ave	4.5310E + 03	7.6573E + 03	4.3612E + 03	3.4870E + 03	3.4046E + 03	<b>2.9975E + 03</b>	3.0996E + 03
	Std	6.4335E + 02	1.4310E + 03	5.3030E + 02	4.9718E + 02	<b>3.4684E + 02</b>	4.5459E + 02	4.4990E + 02
F17	Ave	3.7987E + 03	9.0067E + 03	3.6432E + 03	3.0819E + 03	3.4706E + 03	2.8709E + 03	<b>2.7786E + 03</b>
	Std	4.5146E + 02	2.0948E + 03	3.8545E + 02	3.4783E + 02	3.1650E + 02	3.3586E + 02	<b>2.6692E + 02</b>
F18	Ave	4.3723E + 06	7.9914E + 07	9.1232E + 06	<b>2.1870E + 05</b>	2.9449E + 05	1.9514E + 06	1.2999E + 06
	Std	3.4563E + 06	4.7896E + 07	6.4256E + 06	<b>1.3210E + 05</b>	2.0203E + 05	1.4504E + 06	9.5764E + 05
F19	Ave	1.1264E + 06	3.1135E + 09	2.2803E + 06	<b>1.8302E + 04</b>	2.0434E + 04	1.9484E + 04	9.0972E + 05
	Std	8.5603E + 05	1.6281E + 09	2.1570E + 06	1.1745E + 04	1.1714E + 04	<b>1.1313E + 04</b>	9.4769E + 05
F20	Ave	3.5328E + 03	3.6188E + 03	3.2709E + 03	3.2025E + 03	3.4861E + 03	2.9766E + 03	<b>2.9560E + 03</b>
	Std	2.8422E + 02	3.1895E + 02	<b>2.6175E + 02</b>	4.0586E + 02	3.5025E + 02	2.9144E + 02	2.8389E + 02
	Criteria	HHO	AOA	AO	GBO	FDA	EO	GMO
F21	Ave	2.8744E + 03	3.0610E + 03	2.7043E + 03	2.5662E + 03	2.6266E + 03	2.4414E + 03	<b>2.4294E + 03</b>
	Std	8.0664E + 01	7.1074E + 01	6.3075E + 01	5.1954E + 01	6.4578E + 01	<b>3.2306E + 01</b>	4.3009E + 01
F22	Ave	1.1291E + 04	1.5943E + 04	1.0965E + 04	<b>9.4638E + 03</b>	1.0150E + 04	9.5673E + 03	1.0416E + 04
	Std	1.0318E + 03	<b>6.4563E + 02</b>	1.6841E + 03	1.6273E + 03	1.0895E + 03	1.8416E + 03	3.2104E + 03
F23	Ave	3.8310E + 03	4.4274E + 03	3.4393E + 03	3.0558E + 03	3.1479E + 03	<b>2.8742E + 03</b>	2.8888E + 03
	Std	2.0071E + 02	2.4540E + 02	9.2377E + 01	7.8350E + 01	1.1024E + 02	<b>4.6342E + 01</b>	6.8604E + 01
F24	Ave	4.2735E + 03	4.9461E + 03	3.5218E + 03	3.1724E + 03	3.2978E + 03	3.0284E + 03	<b>3.0248E + 03</b>
	Std	2.5064E + 02	2.3884E + 02	1.2135E + 02	6.0068E + 01	8.3941E + 01	3.8189E + 01	<b>3.0351E + 01</b>

**Table 3** (continued)

	Criteria	HHO	AOA	AO	GBO	FDA	EO	GMO
F25	Ave	3.2218E + 03	1.4965E + 04	3.4327E + 03	3.0855E + 03	3.0746E + 03	3.0896E + 03	<b>3.0490E + 03</b>
	Std	4.7421E + 01	1.6769E + 03	8.8058E + 01	2.3553E + 01	2.9128E + 01	2.5254E + 01	<b>2.2310E + 01</b>
F26	Ave	1.0559E + 04	1.6721E + 04	8.6249E + 03	7.0615E + 03	8.4098E + 03	5.4083E + 03	<b>5.2625E + 03</b>
	Std	1.9144E + 03	1.4623E + 03	2.4721E + 03	2.4977E + 03	2.4109E + 03	5.4784E + 02	<b>4.5503E + 02</b>
F27	Ave	4.3910E + 03	6.6280E + 03	4.0141E + 03	3.5950E + 03	3.5422E + 03	<b>3.3839E + 03</b>	3.4468E + 03
	Std	4.2369E + 02	6.2810E + 02	1.9720E + 02	1.3302E + 02	1.0081E + 02	6.8784E + 01	<b>5.7254E + 01</b>
F28	Ave	3.6322E + 03	1.1991E + 04	4.2899E + 03	3.3326E + 03	3.3361E + 03	3.3539E + 03	<b>3.3184E + 03</b>
	Std	1.0426E + 02	1.5119E + 03	2.7492E + 02	2.7941E + 01	3.7231E + 01	3.3683E + 01	<b>2.3001E + 01</b>
F29	Ave	5.9634E + 03	3.3009E + 04	6.1689E + 03	4.6781E + 03	4.7818E + 03	<b>4.0845E + 03</b>	4.5354E + 03
	Std	5.4130E + 02	1.6772E + 04	6.9454E + 02	3.7906E + 02	4.8072E + 02	<b>2.8174E + 02</b>	3.5989E + 02
F30	Ave	4.8890E + 07	6.1577E + 09	1.2248E + 08	1.0718E + 06	<b>1.0221E + 06</b>	1.1758E + 06	6.4005E + 07
	Std	1.7314E + 07	2.5496E + 09	4.9406E + 07	<b>2.1165E + 05</b>	2.3146E + 05	3.2019E + 05	1.5360E + 07

**Table 4** Parameter settings of the GMO and the second set of its competitive algorithms

Algorithm	Parameter settings
GA	$p r_{\text{crossover}} = 0.9; p r_{\text{mutation}} = \frac{1}{D};$ $D = \text{number of dimensions}$
PSO	$c_1 = c_2 = 2; k = [0.9, 0.4]; v_{\text{max}} = 0.1(UB - LB)$
GWO	$a = [2, 0]$
GSA	$G = 100; \beta = 50$
TLCO	No parameter to tune
KO	Threshold = 0.5
GMO	No parameter to tune

Referring to the structure of this algorithm, the abilities to evaluate the fitness and diversity of the search agents are integrated into a single measure which in turn, can help both these major features of the agents be consistently and interactively evaluated. This property the GMO benefits from is a unique one, not allowing the diversity among the solutions to be lost in favor of achieving the high-fitness regions at the late iterations, and also not letting the search agents remain far away from the high-fitness region and get trapped in local optima at the early iterations. Among the other strengths of the GMO leading to this performance on CEC2017 test functions, the mutation mechanism imposed on the guide agents may be of high importance, since the guide agents are always uncertain to rely as the best existing agents in terms of both fitness and diversity. The elitism mechanism incorporated into the GMO is another factor helping the convergence of the algorithm get accelerated. The latter point is especially important when dealing with the unimodal functions or the small- and middle-scale optimization problems. As the final point,

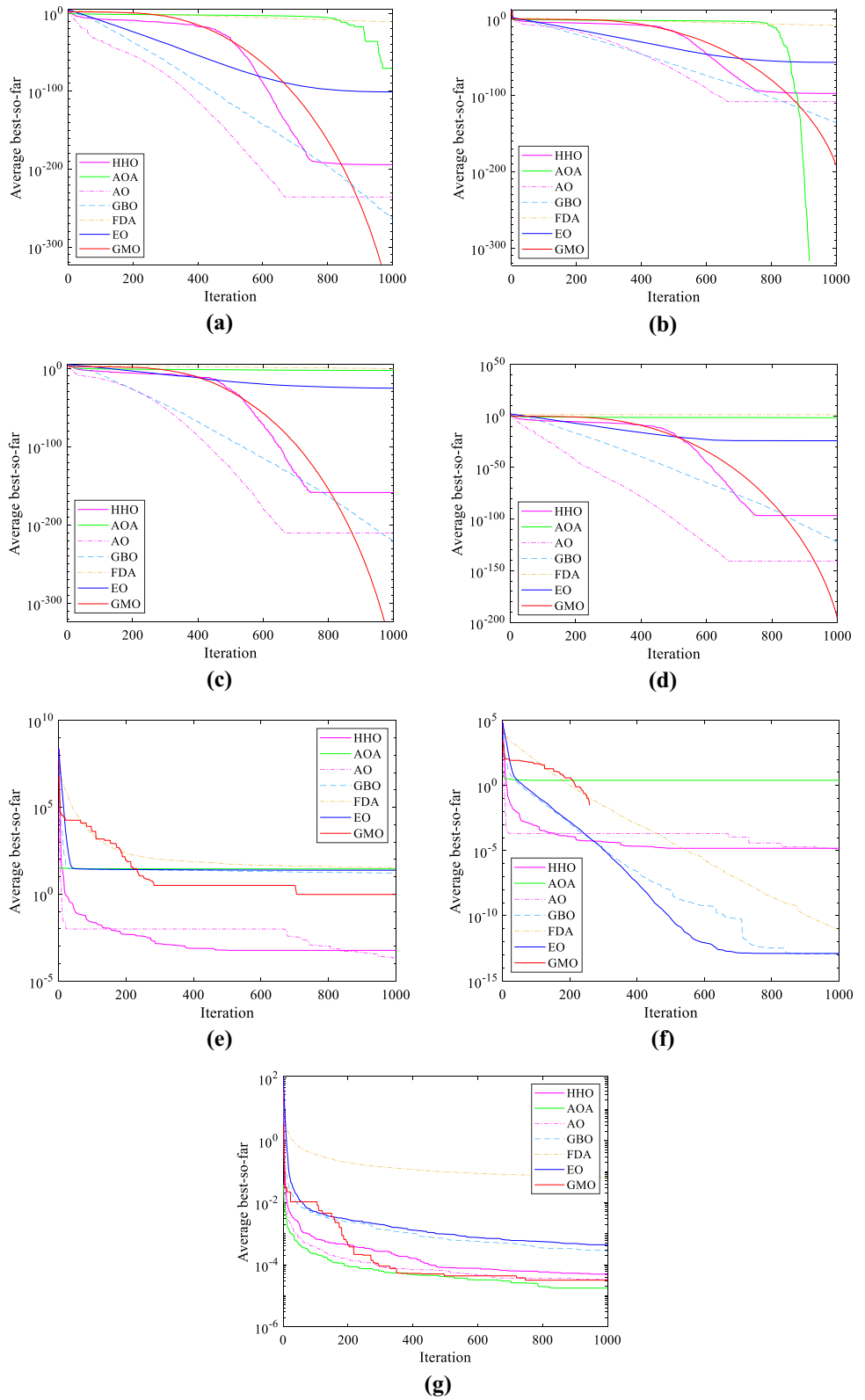
having multiple guide agents that are averaged in the search space can considerably boost the exploratory ability of the proposal and help the better interactions and information exchanges among the search agents during the optimization process.

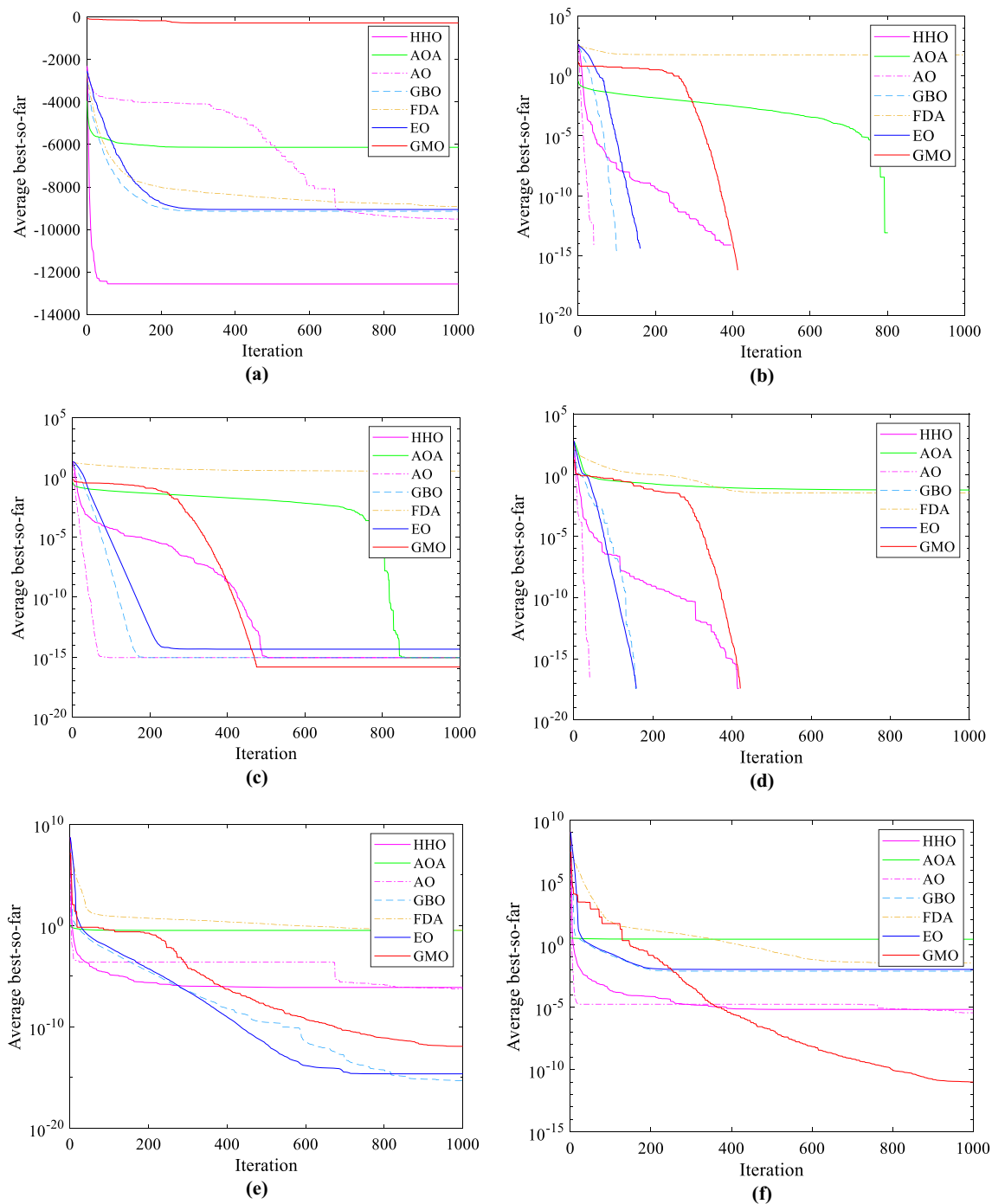
### 3.3 Comparison with other popular algorithms

#### 3.3.1 Preparation of the algorithms

To further investigate the merits of the GMO to compete a series of classical and popular optimizers as well as a couple of recently proposed and efficient optimization algorithms, the GMO was compared with six other algorithms as the second set of the examinees including: (1) genetic algorithm (GA); (2) particle swarm optimization (PSO); (3) grey wolf optimizer (GWO), (4) gravitational search algorithm (GSA); (5) termite life cycle optimizer (TLCO) (Minh et al. 2023); and (6) K-means optimizer (KO) (Minh et al. 2022). The parameter settings of these algorithms are delineated in Table 4. Just like the first category of the examined algorithms, the population size and the maximum number of iterations of these algorithms were set to 50 and 1000, when implemented on the classical 23 test problems, except for the fixed-dimension multi-modal problems (F14-F23), the maximum number of iterations of which was set to 500. In addition, number of search agents and maximum number of iterations were assumed to be 50 and 1000 for this set of the algorithms when applied to the CEC2017 functions. All optimizers were run for 30 independent times. The final experimental results are shown in Tables 5, 6 and 7. The results corresponding to the best-performing algorithms are shown in bold in these tables. The abbreviations ‘‘Ave’’ and ‘‘Std’’ in

**Fig. 3** Convergence curves plotted for **a** F1; **b** F2; **c** F3; **d** F4; **e** F5; **f** F6; **g** F7





**Fig. 4** Convergence curves plotted for **a** F8; **b** F9; **c** F10; **d** F11; **e** F12; **f** F13

these tables denote the average and standard deviation of the final results achieved during 30 runs.

### 3.3.2 Comparative results on the classical benchmark functions

The results obtained from implementing the above-mentioned algorithms on the unimodal problems indicate that

GMO significantly outperforms the GA on 18 out of 23 (78%) of all the performance criteria. The GMO also dominates PSO on 18 out of 23 (78%) of the criteria over the entire 23 classical functions. When the GMO is compared with two more recently proposed and well-known algorithms such as GWO and GSA, the outperformance rates of the GMO are still extended. The proposal performs much better than the GWO and GSA on 17 out of 23 (74%)

**Table 5** Minimization results of the second set of the algorithms implemented on the classical benchmark functions

	Criteria	GA	PSO	GWO	GSA	TLCO	KO	GMO
F1	Ave	7.1705E - 01	9.6480E - 05	3.8366E - 70	2.3354E - 17	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>	3.0138e - 322
	Std	2.8691E - 01	1.2475E - 04	1.5370E - 69	6.3780E - 18	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>
F2	Ave	1.9848E - 01	2.5910E - 02	4.2237E - 41	2.4600E - 08	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>
	Std	5.8153E - 02	3.3716E - 02	3.6410E - 41	4.4629E - 09	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>
F3	Ave	1.5016E + 03	6.4014E + 01	3.6016E - 20	2.3381E + 02	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>	4.9407e - 324
	Std	4.6711E + 02	4.5875E + 01	1.1549E - 19	9.5143E + 01	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>
F4	Ave	4.9110E + 00	1.5277E + 00	1.5246E - 17	3.3434E - 09	2.4125E - 302	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>
	Std	1.1726E + 00	5.3433E - 01	1.7004E - 17	6.4245E - 10	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>
F5	Ave	2.8289E + 02	6.8080E + 01	2.6564E + 01	4.2821E + 01	<b>2.3521E + 01</b>	2.5378E + 01	2.4767E + 01
	Std	3.2350E + 02	4.2962E + 01	5.2060E - 01	4.3274E + 01	1.1213E + 00	5.3859E - 01	<b>2.4519E - 01</b>
F6	Ave	9.6667E - 01	6.3333E - 01	3.6764E - 01	<b>0.0000E + 00</b>	1.8063E - 03	2.2670E - 03	<b>0.0000E + 00</b>
	Std	9.4688E - 01	1.0981E + 00	2.6586E - 01	<b>0.0000E + 00</b>	8.6808E - 03	6.4514E - 03	<b>0.0000E + 00</b>
F7	Ave	6.2401E - 02	6.8077E - 03	5.2159E - 04	1.6740E - 02	1.9834E - 05	<b>9.5733E - 06</b>	2.1509E - 05
	Std	2.3887E - 02	2.7754E - 03	2.2086E - 04	6.5938E - 03	3.5038E - 05	<b>1.2604E - 05</b>	2.5513E - 05
F8	Ave	- 6.7481E+03	- 6.9484E+03	- 6.3362E+03	- 2.7873E+03	<b>- 1.2406E+04</b>	- 1.0403E+04	- 6.4524E+03
	Std	6.4045E + 02	6.0376E + 02	7.1359E + 02	5.2943E + 02	<b>2.0414E + 02</b>	1.5041E + 03	1.2763E + 03
F9	Ave	1.9591E + 00	4.3745E + 01	4.3368E - 01	1.4759E + 01	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>
	Std	1.4974E + 00	9.0870E + 00	1.4376E + 00	3.4672E + 00	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>
F10	Ave	2.2595E - 01	5.3027E - 01	1.3441E - 14	3.9012E - 09	<b>8.8818E - 16</b>	<b>8.8818E - 16</b>	4.4409E - 15
	Std	6.9092E - 02	6.2152E - 01	2.7572E - 15	5.3337E - 10	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>	0.0000E + 00
F11	Ave	6.0048E - 01	1.4275E - 02	1.6645E - 03	4.1483E + 00	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>
	Std	1.9630E - 01	1.5124E - 02	4.3715E - 03	1.6543E + 00	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>	<b>0.0000E + 00</b>
F12	Ave	3.3586E - 03	2.4194E - 02	2.4746E - 02	2.4193E - 02	1.5800E - 04	5.0374E - 04	<b>5.2353E - 11</b>
	Std	4.7632E - 03	5.2254E - 02	1.4209E - 02	5.2253E - 02	8.0337E - 04	2.1388E - 03	<b>3.2351E - 11</b>
F13	Ave	4.4173E - 02	1.3824E - 02	3.3646E - 01	1.1000E - 03	1.0544E - 04	2.5777E - 03	<b>2.2873E - 10</b>
	Std	2.5687E - 02	4.5007E - 02	1.8879E - 01	3.4785E - 03	5.6802E - 04	8.5506E - 03	<b>1.2323E - 10</b>
F14	Ave	5.2932E + 00	1.3621E + 00	4.2328E + 00	6.3420E + 00	<b>9.9800E - 01</b>	<b>9.9800E - 01</b>	3.9936E + 00
	Std	2.8606E + 00	6.6368E - 01	3.9416E + 00	4.6754E + 00	<b>0.0000E + 00</b>	8.4322E - 13	2.7699E + 00
F15	Ave	1.2821E - 03	4.6099E - 04	3.6848E - 03	3.9154E - 03	<b>3.0751E - 04</b>	3.1835E - 04	3.5231E - 04
	Std	1.4095E - 03	3.5073E - 04	7.5880E - 03	2.7063E - 03	<b>1.4466E - 07</b>	5.1683E - 05	1.7746E - 05
F16	Ave	- 1.0316E + 00	- 1.0316E + 00	- 1.0316E + 00	- 1.0316E + 00	<b>- 1.0316E + 00</b>	<b>- 1.0316E + 00</b>	- 1.0312E + 00
	Std	<b>6.6613E - 16</b>	6.7752E - 16	6.7752E - 16	6.7752E - 16	6.7752E - 16	3.7918E - 07	7.9623E - 04
F17	Ave	3.9789E - 01	3.9789E - 01	3.9791E - 01	3.9789E - 01	<b>3.9789E - 01</b>	<b>3.9789E - 01</b>	4.5038E - 01
	Std	1.6653E - 16	1.6938E - 16	1.0589E - 04	1.6938E - 16	<b>0.0000E + 00</b>	6.0576E - 07	1.6137E - 01
F18	Ave	3.0002E + 00	<b>3.0000E + 00</b>	<b>3.0000E + 00</b>	<b>3.0000E + 00</b>	<b>3.0000E + 00</b>	<b>3.0000E + 00</b>	<b>3.0000E + 00</b>
	Std	5.7982E - 04	<b>0.0000E + 00</b>	1.8257E - 05	<b>0.0000E + 00</b>	6.9178E - 14	3.1173E - 06	4.6526E - 11



Table 5 (continued)

Criteria	GA	PSO	GWO	GSA	TLCO	KO	GMO
F19	Ave Std	- 3.8794E + 00 1.3550E - 15	- 3.8617E + 00 2.2996E - 03	- 3.8628E + 00 3.1618E - 15	- 3.8628E + 00 2.6569E - 15	- 3.8628E + 00 5.8886E - 05	- 3.8794E + 00 2.0881E - 05
F20	Ave Std	- 3.3027E + 00 4.5135E - 02	- 3.2409E + 00 6.4483E - 02	- 3.3220E + 00 1.8067E - 15	- 3.2031E + 00 1.7190E - 09	- 3.3050E + 00 4.3311E - 02	- 3.2322E + 00 1.8617E - 01
F21	Ave Std	- 6.9962E + 00 3.6054E + 00	- 8.7194E + 00 2.4536E + 00	- 5.0403E + 00 3.4367E + 00	- 1.0153E + 01 1.2659E - 11	- 1.0153E + 01 3.4680E - 04	- 8.4704E + 00 2.9088E + 00
F22	Ave Std	- 8.0287E + 00 3.3446E + 00	- 1.0049E + 01 1.3430E + 00	- 1.0088E + 01 1.2687E + 00	- 1.0385E + 01 9.6397E - 02	- 1.0402E + 01 3.3878E - 03	- 9.8629E + 00 2.0590E + 00
F23	Ave Std	- 9.5312E + 00 2.6052E + 00	- 9.8137E + 00 2.2382E + 00	- 9.9987E + 00 2.0464E + 00	- 1.0535E + 01 8.4077E - 03	- 1.0536E + 01 1.4605E - 04	- 8.2832E + 00 3.5562E + 00

and 18 out of 23 (78%) of all the performance criteria, respectively. The noticeable point is that the GMO absolutely dominates the GWO and GSA on all the unimodal and multi-modal functions and slightly performs worse than these algorithms on the fixed-dimension functions. This performance is a sign of the robustness of the GMO in solving any type of the global optimization problems, especially when they are high-dimensional in domain. Also, the results suggest the strong exploration and exploitation abilities of the GMO when facing the classical problems, as the high-dimensional problems over which the GMO is fully dominant, are quite desirable examples to test these two abilities any user expects any optimizer to be equipped with. As per the experimental results, the GMO is also superior to two newly proposed algorithms, namely TLCO and KO on 9 out of 13 (69%) of all the unimodal and multi-modal functions included in the examined classical benchmark functions, while each of these two competitors slightly outperforms the GMO on 8 out of 13 (62%) of the unimodal and multi-modal functions. The KO is rated as the best-performing algorithm on 10 fixed-dimension functions with a dominance rate of 70%, a rate very close to that the TLCO offers on this test bed. These results indicate that the GMO is specifically designed to handle large-scale and so complicated problems. The blunt reason for this inference is its brilliant performance when facing the first 13 high-dimensional problems and its slight weakness in solving the 13 fixed-dimension problems in the classical test functions. This point also clarifies that the GMO is of higher convergence accuracy than the convergence speed, contributing such a proposed optimizer to better tackle the problems with large dimensions, complicated domains, and constraints. It is noteworthy that the results of solving engineering design problems mentioned at the end of this paper clearly demonstrate the abilities asserted that the GMO benefits from when solving complicated problems.

### 3.3.3 Comparative results on the CEC2017 benchmark functions

To further examine the real abilities of the proposed algorithm in solving optimization problems, the performance of the GMO and the second set of the optimizers is all tested on the CEC2017 test suite. Here, the dimensions of all the CEC2017 benchmark functions are set to be 50 to challenge the abilities of the proposal to handle such large-scale and complex optimization problems. As the results suggest, the proposed algorithm performs far better than the other competitors on this test bed, such that the GMO can reach the best performance criteria on 29 out of 58 (50%) of all the problems, while achieving the best average optimal objective function values on 19 out of 29 (66%) of

**Table 6** Minimization results of the second set of the algorithms implemented on the CEC2017 test suite

	Criteria	GA	PSO	GWO	GSA	TLCO	KO	GMO
F1	Ave	8.0867E + 07	4.1183E + 07	4.5500E + 09	1.9300E + 10	3.6093E + 09	5.1536E + 06	<b>4.2020E + 03</b>
	Std	2.6283E + 07	1.8656E + 08	1.0000E + 09	2.8500E + 09	8.3084E + 08	1.5872E + 06	<b>4.1389E + 03</b>
F3	Ave	1.8044E + 05	1.7274E + 05	1.2400E + 05	1.8900E + 05	<b>9.9179E + 04</b>	1.1948E + 05	1.3609E + 05
	Std	2.6345E + 04	2.9773E + 04	2.1000E + 04	<b>5.3300E + 03</b>	1.4610E + 04	2.0013E + 04	1.7457E + 04
F4	Ave	6.6553E + 02	7.6603E + 02	1.3500E + 03	3.8400E + 03	1.1118E + 03	6.2783E + 02	<b>5.8965E + 02</b>
	Std	4.9498E + 01	9.5012E + 01	1.7700E + 02	6.6000E + 02	1.4690E + 02	4.9009E + 01	<b>4.3998E + 01</b>
F5	Ave	7.4790E + 02	6.7990E + 02	7.3000E + 02	8.4400E + 02	8.4861E + 02	8.0643E + 02	<b>6.2615E + 02</b>
	Std	2.7413E + 01	6.3260E + 01	6.1200E + 01	<b>2.4800E + 01</b>	5.5404E + 01	3.2886E + 01	2.9495E + 01
F6	Ave	6.2983E + 02	<b>6.0654E + 02</b>	6.1900E + 02	6.6300E + 02	6.2837E + 02	6.5902E + 02	6.1310E + 02
	Std	5.0762E + 00	<b>2.7202E + 00</b>	3.4700E + 00	3.2300E + 00	4.8356E + 00	7.2183E + 00	6.8630E + 00
F7	Ave	1.2561E + 03	1.0670E + 03	1.0900E + 03	1.3900E + 03	1.2716E + 03	1.1664E + 03	<b>8.8476E + 02</b>
	Std	6.3194E + 01	8.4239E + 01	8.4200E + 01	6.7600E + 01	7.6647E + 01	1.2719E + 02	<b>3.7289E + 01</b>
F8	Ave	1.0439E + 03	1.0106E + 03	1.0300E + 03	1.1600E + 03	1.1239E + 03	1.1134E + 03	<b>9.5240E + 02</b>
	Std	2.6439E + 01	9.1183E + 01	3.8700E + 01	<b>2.3800E + 01</b>	4.2109E + 01	3.0389E + 01	8.4595E + 01
F9	Ave	9.7046E + 03	2.7450E + 03	7.2200E + 03	1.1900E + 04	1.4020E + 04	2.4002E + 04	<b>1.5896E + 03</b>
	Std	1.6566E + 03	1.0967E + 03	2.5900E + 03	<b>5.9100E + 02</b>	4.4702E + 03	6.0152E + 03	1.4550E + 03
F10	Ave	7.9006E + 03	1.1106E + 04	<b>6.5400E + 03</b>	8.7900E + 03	8.2371E + 03	8.8409E + 03	8.7748E + 03
	Std	8.2855E + 02	2.1536E + 03	<b>5.2900E + 02</b>	1.1200E + 03	6.3289E + 02	2.6356E + 03	2.5236E + 03
F11	Ave	4.7913E + 03	1.5600E + 03	6.2300E + 03	1.8000E + 04	2.4557E + 03	1.4591E + 03	<b>1.3630E + 03</b>
	Std	1.4843E + 03	1.2287E + 02	4.8100E + 03	2.5900E + 03	7.8861E + 02	8.7615E + 01	<b>4.3463E + 01</b>
F12	Ave	2.7093E + 07	<b>1.1365E + 07</b>	5.4100E + 08	4.8800E + 09	1.9559E + 08	3.8175E + 07	3.3511E + 07
	Std	1.0509E + 07	<b>8.4228E + 06</b>	8.2400E + 08	3.9900E + 09	9.6834E + 07	1.4414E + 07	2.1415E + 07
F13	Ave	6.9021E + 06	<b>1.1554E + 04</b>	3.7800E + 08	2.1300E + 06	1.8269E + 06	1.0759E + 05	5.8217E + 04
	Std	3.1719E + 06	<b>9.4238E + 03</b>	4.2500E + 08	3.6100E + 06	1.2563E + 06	3.3385E + 04	3.0336E + 04
F14	Ave	6.4092E + 06	2.4573E + 05	5.7400E + 05	5.4100E + 06	1.5204E + 06	7.8068E + 05	<b>1.1122E + 05</b>
	Std	4.5298E + 06	1.6811E + 05	3.4300E + 05	5.1600E + 06	1.4510E + 06	6.6120E + 05	<b>7.8991E + 04</b>
F15	Ave	3.1225E + 06	3.9651E + 04	1.0600E + 05	3.1100E + 08	9.7781E + 04	3.4908E + 04	<b>2.9686E + 04</b>
	Std	1.8305E + 06	1.4600E + 05	5.2900E + 04	2.7600E + 08	3.6877E + 05	<b>1.5644E + 04</b>	2.3133E + 04
F16	Ave	3.2954E + 03	3.4038E + 03	3.1100E + 03	4.2700E + 03	3.5440E + 03	3.5798E + 03	<b>3.0996E + 03</b>
	Std	4.0332E + 02	6.9952E + 02	<b>3.6400E + 02</b>	5.7000E + 02	3.7673E + 02	4.9055E + 02	4.4990E + 02
F17	Ave	3.3643E + 03	2.9902E + 03	2.8700E + 03	3.7700E + 03	3.1667E + 03	3.4327E + 03	<b>2.7786E + 03</b>
	Std	3.7884E + 02	3.3747E + 02	3.7400E + 02	3.8400E + 02	<b>2.6359E + 02</b>	3.6965E + 02	2.6692E + 02
F18	Ave	5.2873E + 06	5.2811E + 06	6.8800E + 06	5.0500E + 06	5.1842E + 06	3.7095E + 06	<b>1.2999E + 06</b>
	Std	2.5971E + 06	5.1315E + 06	1.3000E + 06	1.1000E + 06	4.8820E + 06	2.0253E + 06	<b>9.5764E + 05</b>
F19	Ave	2.7074E + 05	<b>9.8542E + 03</b>	1.3300E + 07	3.2600E + 05	2.3925E + 04	5.4487E + 05	9.0972E + 05
	Std	2.4735E + 05	1.2081E + 04	4.5500E + 07	1.8700E + 05	<b>1.0485E + 04</b>	4.3619E + 05	9.4769E + 05
F20	Ave	3.1329E + 03	3.0089E + 03	<b>2.8900E + 03</b>	3.6800E + 03	2.9847E + 03	3.0221E + 03	2.9560E + 03
	Std	3.7542E + 02	4.0958E + 02	2.8600E + 02	<b>2.7800E + 02</b>	3.6910E + 02	3.0330E + 02	2.8389E + 02
	Criteria	GA	PSO	GWO	GSA	TLCO	KO	GMO
F21	Ave	2.5731E + 03	2.4751E + 03	2.5200E + 03	2.8600E + 03	2.6032E + 03	2.5801E + 03	<b>2.4294E + 03</b>
	Std	3.6186E + 01	5.8576E + 01	6.5800E + 01	4.1900E + 01	<b>2.9039E + 01</b>	3.9696E + 01	4.3009E + 01
F22	Ave	1.0224E + 04	1.4072E + 04	<b>9.2400E + 03</b>	1.1600E + 04	9.8609E + 03	9.7846E + 03	1.0416E + 04
	Std	9.5225E + 02	2.0110E + 03	2.6000E + 03	<b>6.1200E + 02</b>	2.0060E + 03	2.4253E + 03	3.2104E + 03
F23	Ave	3.5242E + 03	2.9730E + 03	2.9700E + 03	4.8600E + 03	3.1162E + 03	3.1334E + 03	<b>2.8888E + 03</b>
	Std	1.0030E + 02	9.9559E + 01	7.2700E + 01	2.0300E + 02	<b>4.2177E + 01</b>	1.0472E + 02	6.8604E + 01
F24	Ave	3.9360E + 03	3.2535E + 03	3.1800E + 03	4.4900E + 03	3.3881E + 03	3.2464E + 03	<b>3.0248E + 03</b>
	Std	9.9701E + 01	7.6166E + 01	1.1900E + 02	1.1400E + 02	7.9707E + 01	1.4563E + 02	<b>3.0351E + 01</b>

**Table 6** (continued)

	Criteria	GA	PSO	GWO	GSA	TLCO	KO	GMO
F25	Ave	3.2161E + 03	3.2043E + 03	3.5800E + 03	4.7400E + 03	3.5552E + 03	3.1568E + 03	<b>3.0490E + 03</b>
	Std	2.7174E + 01	4.5934E + 01	2.4700E + 02	3.8800E + 02	1.2299E + 02	3.1484E + 01	<b>2.2310E + 01</b>
F26	Ave	1.0906E + 04	5.7138E + 03	6.3700E + 03	1.2300E + 04	6.5533E + 03	8.1369E + 03	<b>5.2625E + 03</b>
	Std	7.9498E + 02	6.4885E + 02	<b>4.5400E + 02</b>	8.1000E + 02	2.2941E + 03	3.3522E + 03	4.5503E + 02
F27	Ave	4.7515E + 03	3.5528E + 03	3.6300E + 03	8.2900E + 03	3.8153E + 03	3.8439E + 03	<b>3.4468E + 03</b>
	Std	2.1918E + 02	8.1081E + 01	8.7700E + 01	6.6900E + 02	1.2050E + 02	1.2874E + 02	<b>5.7254E + 01</b>
F28	Ave	3.5963E + 03	3.4180E + 03	4.1200E + 03	5.5200E + 03	4.0776E + 03	3.4275E + 03	<b>3.3184E + 03</b>
	Std	6.2885E + 01	4.9966E + 01	2.3600E + 02	1.2700E + 02	2.0709E + 02	4.0071E + 01	<b>2.3001E + 01</b>
F29	Ave	5.2031E + 03	<b>4.2722E + 03</b>	4.5800E + 03	1.0500E + 04	4.4246E + 03	5.0434E + 03	4.5354E + 03
	Std	3.1439E + 02	3.3670E + 02	<b>2.8200E + 02</b>	5.2300E + 03	3.4678E + 02	4.0952E + 02	3.5989E + 02
F30	Ave	5.7944E + 06	<b>4.5049E + 06</b>	1.0900E + 08	2.3100E + 08	1.2469E + 07	3.5560E + 07	6.4005E + 07
	Std	2.4707E + 06	<b>1.5678E + 06</b>	4.9900E + 07	6.5300E + 06	3.2603E + 06	3.8659E + 06	1.5360E + 07

**Table 7** *p*-values calculated for the classical benchmark functions when applying the first set of the optimizers

Algorithms	HHO	AOA	AO	GBO	FDA	EO	GMO
HHO	<u>N/A</u>						
AOA	1.4125E - 04	<u>N/A</u>					
AO	<u>9.9683E - 01</u>	2.1185E - 04	<u>N/A</u>				
GBO	<u>9.7156E - 01</u>	5.1634E - 04	<u>7.7212E - 01</u>	<u>N/A</u>			
FDA	<u>8.2451E - 05</u>	<u>7.7098E - 01</u>	8.4419E - 05	1.0075E - 04	<u>N/A</u>		
EO	<u>7.5472E - 01</u>	7.1607E - 04	<u>5.9561E - 01</u>	<u>8.0019E - 01</u>	9.3845E - 05	<u>N/A</u>	
GMO	<u>8.8818E - 01</u>	4.4399E - 03	<u>8.4035E - 01</u>	<u>8.5337E - 01</u>	2.0510E - 03	<u>8.9161E - 01</u>	<u>N/A</u>

all the problems. In detail, the GMO is superior to its competitors on 7 out of 10 (70%) of the composite functions, 6 out of 10 (60%) of the hybrid functions, and 6 out of 9 (67%) of the unimodal and multi-modal functions, while the PSO is the closest rival of the GMO, achieving the best performance criteria on 10 out of 58 (17%), and the best average results on 6 out of 29 (21%) of the entire test problems of the CEC2017 suite. The GWO performs slightly better than the GSA, reaching the best performance criteria on 7 versus 6 out of 29 criteria. However, the GWO offers the best average results on 3 problems as well, while the GSA yields no best average results against the other examinees. After the GA that is rated as the worst optimizer, absolutely losing the competition to the other algorithms, the KO and TLCO stand on the second and third places as the optimizers not performing desirably on this test suite. As these two optimizers performed desirably on the 23 classical benchmark functions, the undesirable performance of them on the CEC2017 suite clearly indicate that the convergence speed of these optimizers is much higher than their convergence accuracy, at least compared

to the other four optimizers examined on this test bed. In other words, the strength of the TLCO and KO is in their exploitation ability, while their exploration ability can be enhanced.

### 3.4 Statistical analysis

In statistical analysis, the Wilcoxon rank-sum test is used as a nonparametric test to delineate whether a couple of sets of solutions are significantly different [43]. When assuming the results of a series of algorithms as the solutions, the best-performing algorithm is usually compared to its competitors to benchmark the significance of its dominances over them. The output of this test is a number named *p*-value. If the test results in a *p*-value < 0.05 for an algorithm, it is assumed to be statistically significantly different compared to another algorithm taking part in this test.

The test results are presented in Tables 8, 9, 10 and 11. In these tables, the expression N/A mentioned in each row represents that the algorithm ahead of the corresponding

row is “not applicable” in the test, meaning that this algorithm cannot be compared to itself. Also, the  $p$ -values exceeding 0.05 are underlined in these tables. On the classical benchmark functions, the GMO outperforms the first set of its competitors on 11 out of 23 (48%) of the benchmark functions, while GBO outperforms the others on 8 out of 23 (35%) and the HHO and AO both outperform their competitors on 7 out of 23 benchmarks (30%). The other algorithms show worse performance than the mentioned algorithms as the leading ones. On the CEC2017 functions, the GMO dominates its rivals on 12 out of 29 (41%), trailed by the EO and FDA each of which outperforms the other algorithms on 7, 6, and 4 out of 29 functions, meaning 24, 21, and 14% of all cases. As can be seen, the GMO shows much better performance on such a hard-to-solve set of the benchmark functions. However, as the Wilcoxon test suggests, there is a close competition among the examined algorithms in this paper on the classical 23 benchmark test problems, such that the rate of the significant outperformance is low among the comparative algorithms. In detail, among the first set of the examinees, the GMO, GBO, HHO, and AO that were identified to be the best-performing algorithms among the others to solve the classical benchmark functions, can only be significantly superior to two other algorithms among their rivals. While on the CEC2017 test problems, the GMO and FDA are significantly superior to all the other algorithms except for one, while the EO shows significant superiority to all other algorithms.

On the classical test problems, the GMO outperforms the second set of its competitors on 9 out of 23 benchmarks (39%), while the TLCO and KO are superior to the other competitive algorithms on 14 and 15 out of 23 benchmarks (61) and (65%), respectively. The other examinees perform far worse than the GMO, TLCO, and KO, outperforming their rivals on maximum two test functions included in the classical functions. On the CEC2017 test suite, the GMO is superior to the other optimizers on 19 out of 29 (66%) of all the problems, followed by the PSO as its closest rival

outperforming the other algorithms on 6 out of 29 (21%) of all the test functions of this suite. As mentioned before, the other optimizers such as TLCO and KO perform not desirably on the CEC2017 suite, at least when compared to their performance on the classical functions and also the performance of their competitors on the CEC2017 test bed. As the Wilcoxon test suggests, the TLCO and KO that were identified to be the best-performing algorithms to solve the classical problems cannot significantly outperform their rivals on these functions, such that the TLCO has a significant dominance over the all its competitors except for the GMO, and the KO is significantly superior to the other algorithms except for the TLCO and GMO. The GMO is also significantly superior to all its rivals from the second set of the examined optimizers except for the GSA, TLCO, and KO. While on the CEC2017 test suite, the GMO is not only the best-performing algorithm among the other six ones, but also shows significant superiority to all its competitors, the characteristics that none of the other examinees in the second set benefit from.

The statistical test reveals that the CEC2017 may be a more appropriate bed for benchmarking the effectiveness of each of the examined algorithms when evaluated individually or as compared to the other examinees, especially considering the CEC2017 test suite as a more challenging and more realistic test bed for evaluating the efficiency of each of the optimizers. Overall, the proposed GMO shows the best performance whether evaluated in terms of the dominance over its rivals or examined in terms of the significance of its dominances when solving both classical benchmarks and the CEC2017 test bed.

### 3.5 Qualitative analysis for the convergence of GMO

In this subsection, we qualitatively analyze the convergence behavior of the proposed GMO algorithm using five criteria illustrated for several benchmark functions selected among the 23 classical functions. These functions include a

**Table 8**  $p$ -values calculated for the CEC2017 benchmark functions when applying the first set of the optimizers

Algorithms	HHO	AOA	AO	GBO	FDA	EO	GMO
HHO	<u>N/A</u>						
AOA	3.0349E – 15	<u>N/A</u>					
AO	<u>7.8459E – 01</u>	5.8696E – 15	<u>N/A</u>				
GBO	5.4284E – 03	2.4333E – 16	4.1141E – 03	<u>N/A</u>			
FDA	1.0965E – 02	5.4227E – 17	1.2798E – 02	<u>1.0000E + 00</u>	<u>N/A</u>		
EO	2.9622E – 07	4.1926E – 19	5.0703E – 08	1.7423E – 02	2.7781E – 02	<u>N/A</u>	
GMO	5.8183E – 08	1.9799E – 17	2.3970E – 08	1.9715E – 03	4.1544E – 03	<u>1.9960E – 01</u>	<u>N/A</u>

**Table 9** *p*-values calculated for the classical benchmark functions when applying the second set of the optimizers

Algorithms	GA	PSO	GWO	GSA	TLCO	KO	GMO
GA	<u>N/A</u>						
PSO	3.8076E – 02	<u>N/A</u>					
GWO	5.2655E – 03	<u>2.6641E – 01</u>	<u>N/A</u>				
GSA	5.7096E – 03	<u>2.2302E – 01</u>	<u>7.2190E – 01</u>	<u>N/A</u>			
TLCO	4.0344E – 11	2.1482E – 09	5.2322E – 08	6.2618E – 06	<u>N/A</u>		
KO	3.7388E – 10	7.7013E – 08	3.2000E – 06	1.3993E – 04	<u>3.2189E – 01</u>	<u>N/A</u>	
GMO	5.2676E – 05	3.0422E – 03	1.2608E – 02	<u>5.6492E – 02</u>	<u>5.8613E – 02</u>	<u>1.8962E – 01</u>	<u>N/A</u>

**Table 10** *p*-values calculated for the CEC2017 benchmark functions when applying the second set of the optimizers

Algorithms	GA	PSO	GWO	GSA	TLCO	KO	GMO
GA	<u>N/A</u>						
PSO	<u>1.5485E – 01</u>	<u>N/A</u>					
GWO	<u>5.3249E – 01</u>	<u>8.3459E – 02</u>	<u>N/A</u>				
GSA	<u>9.3703E – 06</u>	1.2014E – 05	2.6732E – 04	<u>N/A</u>			
TLCO	<u>7.7824E – 01</u>	<u>2.6053E – 01</u>	<u>4.3098E – 01</u>	3.5899E – 06	<u>N/A</u>		
KO	<u>8.6407E – 01</u>	<u>1.7635E – 01</u>	<u>5.8597E – 01</u>	1.0244E – 04	<u>8.2301E – 01</u>	<u>N/A</u>	
GMO	1.1348E – 07	3.2394E – 04	1.0824E – 06	2.4957E – 11	2.7416E – 06	4.0468E – 07	<u>N/A</u>

**Table 11** Results of solving welded beam design problem

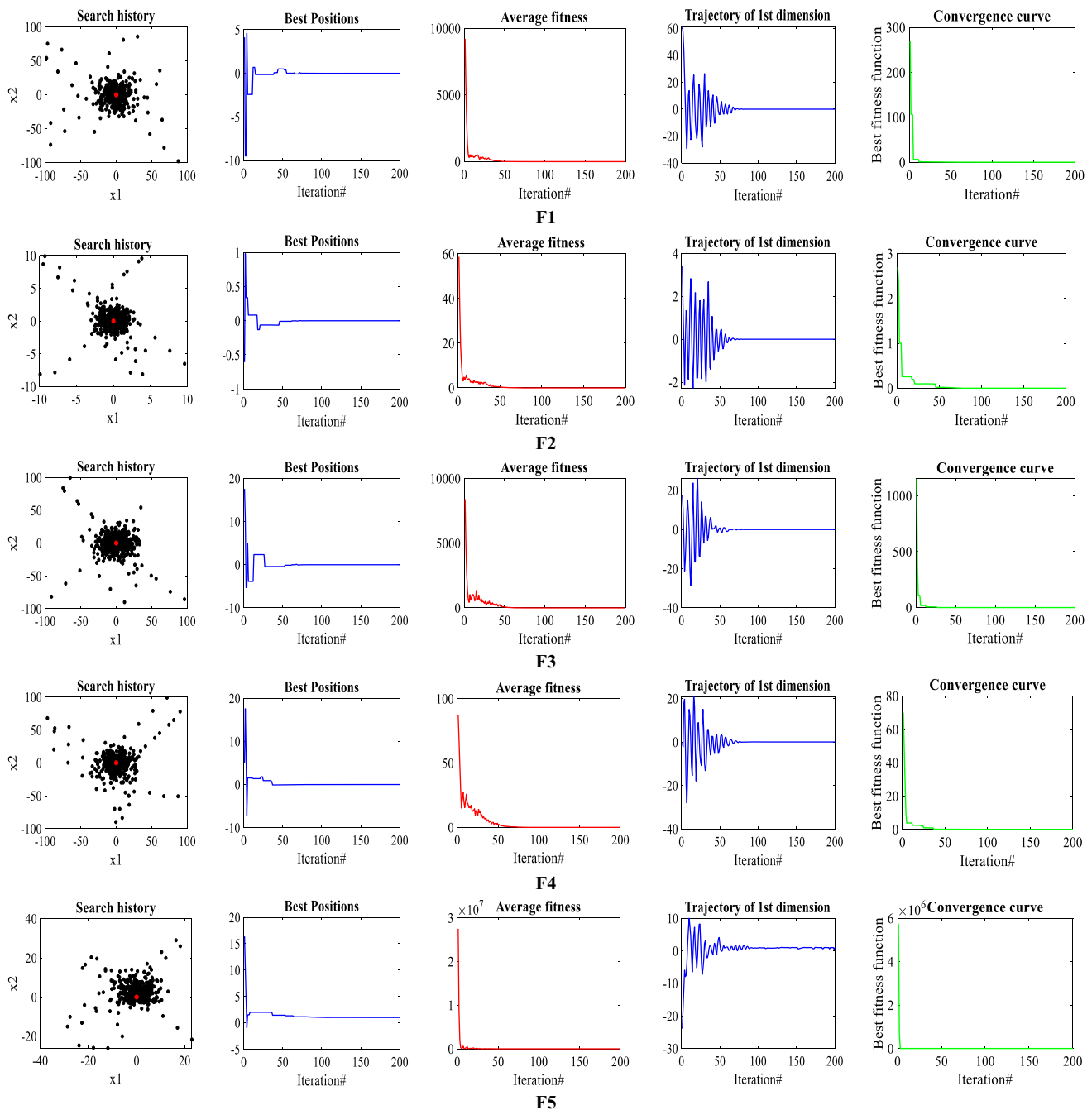
Algorithm	$x_1$	$x_2$	$x_3$	$x_4$	$f_{min}$
SIMPLEX (Ragsdell and Phillips 1976)	0.279200	5.625600	7.751200	0.279600	2.530700
DAVID (Ragsdell and Phillips 1976)	0.243400	6.255200	8.291500	0.244400	2.384100
APPROX (Ragsdell and Phillips 1976)	0.244400	6.218900	8.291500	0.244400	2.381500
GA (Deb 1991)	0.248900	6.173000	8.178900	0.253300	2.430000
HS (Lee and Geem 2005)	0.244200	6.223100	8.291500	0.240000	2.380700
CSCA (Huang et al. 2007)	0.203137	3.542998	9.033498	0.206179	1.733461
CPSO (He and Wang 2007a)	0.202369	3.544214	9.048210	0.205723	1.728020
RO (Kaveh and Khayatazad 2012)	0.203687	3.528467	9.004233	0.207241	1.735344
WOA (Mirjalili et al. 2016)	0.205396	3.484293	9.037426	0.206276	1.730499
GSA (Abd Elaziz et al. 2017)	0.182129	3.856979	10.000000	0.202376	1.879950
MVO (Mirjalili et al. 2016)	0.205463	3.473193	9.044502	0.205695	1.726450
OBSCA (Abd Elaziz et al. 2017)	0.230824	3.069152	8.988479	0.208795	1.722315
AOA (Abualigah et al. 2021a)	0.194475	2.570920	10.000000	0.201827	1.716400
<b>GMO</b>	<b>0.205420</b>	<b>3.258644</b>	<b>9.036624</b>	<b>0.205730</b>	<b>1.695541</b>

variety of unimodal and multi-modal problems adopted in a 2D domain. The detailed analytical results are presented in Figs. 5 and 6, depicted for unimodal and multi-modal functions, respectively.

The first column indicates the search history of the GMO when optimizing each test problem, illustrating the individual and interactional behavior of the search agents during the optimization process. As can be seen in Figs. 5 and 6, the search agents are all scattered across the full

domain of all problems, showing the diversity of the solutions are properly maintained during the search process. In addition, the search agents are able to efficiently explore the search space and find the fittest region of the search space to focus on this region sufficiently to finally exploit the optimal or near-optimal solution of each problem.

The second column illustrates the position of the optimal solution at its first dimension. As can be seen, the optimal



**Fig. 5** Qualitative results for the unimodal test problems

solution fluctuates at the early iterations of the optimization process, but soon converges to its optimal position in the search space. As shown in the second column for all exemplary test problems, the fluctuation of the optimal solution's first position is continued up to 50th iteration and then, the curve of the best (optimal) solution is flattened. This behavior of the best (optimal) solution indicates the fast convergence of the proposed algorithm on all the test functions.

The third column illustrates the average fitness of the population depicted iteration-by-iteration. As can be seen, the average fitness is high at the early iterations, while slightly fluctuating up to 50th iteration and then converges to the optimal fitness of each problem. Among the problems examined in this analysis, F9 has somehow different behavior, such that the average fitness of each iteration experiences severe fluctuations at the early iterations, being continued to the nearby of the final iterations and suddenly

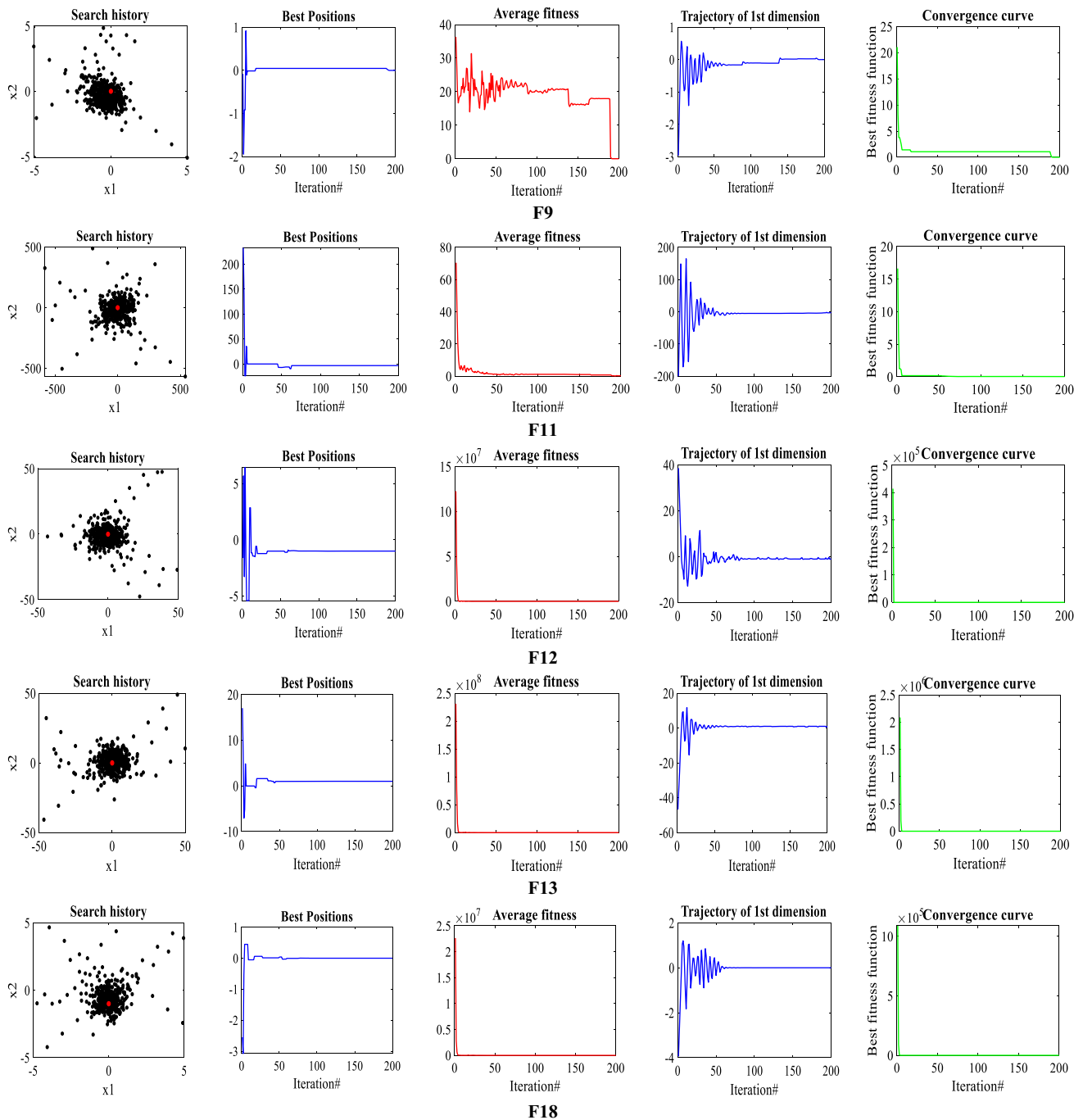


Fig. 6 Qualitative results for the multi-modal test problems

converges to the optimal fitness of the problem. The main reason explaining this behavior may be hidden in the nature of this function. The F9 is a highly multi-modal function, and reaching the optimal solution of this problem is a difficult task and cannot be soon accomplished. Generally, the third column again evidences the high convergence rate of the GMO when solving a variety of the unimodal and multi-modal optimization problems.

The fourth column illustrates the trajectory of the position of the first solution at the first dimension of the test functions. As this column suggests, the fluctuations of the first solution's position are high at the early iterations and are gradually decreased over the course of iterations to converge to its optimal position as a task fulfilled after 50th iteration on all problems. It is worth noticing that on the unimodal functions, the position of the first solution fluctuates just around the optimal solution that is located at the

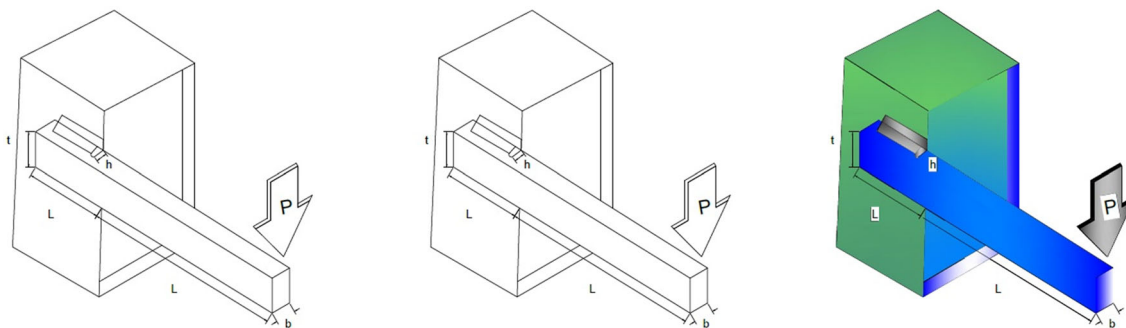


Fig. 7 Welded beam scheme (Mahdavi et al. 2007)

center of the domain; however, when dealing with the multi-modal functions, this position begins the search from one corner of the domain and gradually finds its right path to reach its optimal position in the search space. This is another evidence of difficulty of the multi-modal functions to be solved as compared to the unimodal ones.

The fifth column illustrates the convergence curve of the GMO when handling the exemplary test functions. As this curve displays the best fitness function found iteration-by-iteration, the convergence behavior of the algorithm can be easily identified. As can be seen in this column when depicted for all problems, the high convergence rate of the proposed algorithm is clearly inferred. Finally, this qualitative analysis stresses the abilities of the GMO to both maintain the diversity of the solutions and helping them rapidly converge to the optimal point on the optimization problems.

### 4 Comparative results on engineering problems

To validate the proposal to tackle other types of difficulties an optimization problem may be engaged with, the GMO is implemented on nine constrained engineering problems. The penalty functions are added to the cost functions of these problems for constraint handling. Furthermore, 50 search agents, and 1000 iterations, are set to be used in GMO. The algorithm is run 30 times, and the best results achieved are presented versus those reached by its competitors. It is worth noticing that the results corresponding to the best-performing algorithm are emboldened in the tables presented for all problems.

#### 4.1 Welded beam design problem

Coello (Coello Coello 2000) first proposed the welded beam problem which aims to minimize the construction cost. The main objective function is subject to constraints on shear stress ( $\tau$ ), bending stress ( $\sigma$ ) in the beam, buckling

load on the bar ( $P$ ), end deflection of the beam ( $\delta$ ), and side constraints. The problem includes four design variables consisting of  $h(x_1)$ ,  $l(x_2)$ ,  $t(x_3)$ , and  $b(x_4)$ , as shown in Fig. 7. The problem is formulated as follows.

$$\text{Minimize } f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \tag{12}$$

Subject to:

$$g_1(\vec{x}) = \tau(x) - \tau_{max} \leq 0 \tag{13}$$

$$g_2(\vec{x}) = \sigma(x) - \sigma_{max} \leq 0 \tag{14}$$

$$g_3(\vec{x}) = x_1 - x_4 \leq 0 \tag{15}$$

$$g_4(\vec{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0 \tag{16}$$

$$g_5(\vec{x}) = 0.125 - x_1 \leq 0 \tag{17}$$

$$g_6(\vec{x}) = \delta(x) - \delta_{max} \leq 0 \tag{18}$$

$$g_7(\vec{x}) = P - P_c(x) \leq 0 \tag{19}$$

$$0.1 \leq x_i \leq 2, \quad i = 1, 4 \tag{20}$$

$$0.1 \leq x_i \leq 10, \quad i = 2, 3 \tag{21}$$

where

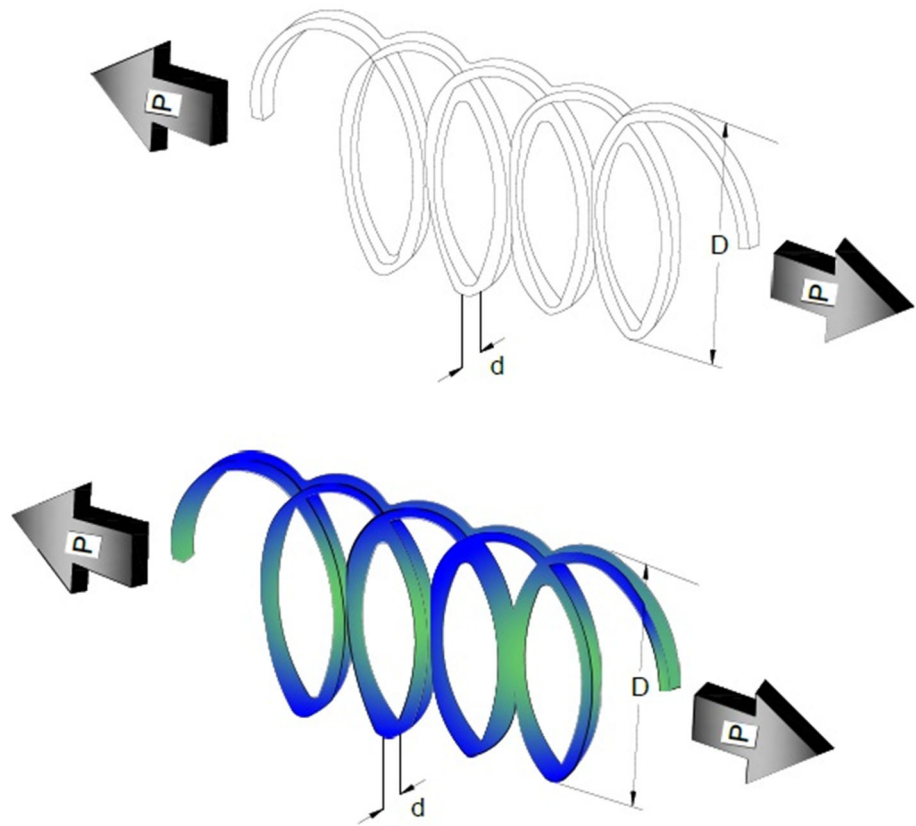
$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \tau' = \frac{P}{\sqrt{2x_1x_2}}, \tau'' = \frac{MR}{J} \tag{22}$$

$$M = P\left(L + \frac{x_2}{2}\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, J = 2\left\{\sqrt{2x_1x_2}\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \tag{23}$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2}, \delta(x) = \frac{4PL^3}{Ex_3^3x_4}, P_c(x) = \frac{4.013E\sqrt{(x_3^2x_4^6/36)}}{L^2} \left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{24}$$



**Fig. 8** Structure of tension/compression spring (Mahdavi et al. 2007)



$$P = 6000lb, L = 14in, E = 30 \times 10^6psi, G = 12 \times 10^6psi \tag{25}$$

$$\tau_{max} = 13,600psi, \sigma_{max} = 30,000psi, \delta_{max} = 0.25in \tag{26}$$

Table 11 shows the optimal cost function and design variables computed by the GMO and the other comparative algorithms upon solving the problem. As the results suggest, the GMO can reach the minimum cost among the other algorithms on such a highly constrained and complex real life problem.

### 4.2 Tension/compression spring design problem

Arora (Arora 2004) first introduced this problem. This problem aims to minimize the weight of a tension/compression spring (TCS), shown in Fig. 8, subject to some constraints. The design variables are wire diameter  $d(x_1)$ , mean coil diameter  $D(x_2)$ , and number of active coils  $N(x_3)$ . The problem is formulated as follows.

$$\text{Minimize } f(\vec{x}) = (x_3 + 2)x_2x_1^2 \tag{27}$$

Subject to:

$$g_1(\vec{x}) = 1 - \frac{x_2^3x_3}{71,785x_1^4} \leq 0 \tag{28}$$

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12,566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0 \tag{29}$$

$$g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \tag{30}$$

$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \tag{31}$$

$$0.05 \leq x_1 \leq 2.00 \tag{32}$$

$$0.25 \leq x_2 \leq 1.30 \tag{33}$$

$$2.00 \leq x_3 \leq 15.0 \tag{34}$$

Table 12 indicates the final results the GMO and its rivals can reach when the problem is solved. As can be seen, the GMO reaches the minimum weight of the spring, while not violating any of the constraints. However, the optimal results of the HS, RO, GSA, and AOA, violate the constraint  $g_2$ , and the results of the OBSCA violate the constraint  $g_1$ . The constraints violated are underlined in the table. As can be seen, the optimal objective value achieved by the OBSCA and AOA are less than that of the proposed GMO algorithm, however, these values cannot be accepted as their corresponding optimal variables violate one of the constraints of the problem. As a result, the GMO outperforms all its competitors on this engineering problem as it

could calculate the least weight for the beam while fully satisfying the constraints of the problem.

### 4.3 Speed reducer design problem

This problem targets to minimizing the weight of a speed reducer, subject to numerous constraints (Mezura-Montes and Coello 2005). A graphical image of the speed reducer is displayed in Fig. 9. The variables  $x_1$  to  $x_7$  represent  $b, m, z, l_1, l_2, d_1$  and  $d_2$ , respectively, as depicted in Fig. 9. Having 11 constraints in its structure, this problem is provided with a high difficulty which can make it much harder to solve for any algorithm. This problem is formulated as follows.

$$\begin{aligned} \text{Minimize } f(\vec{x}) = & 0.7854x_1x_2^2 \\ & (3.3333x_3^2 + 14.9334x_3 - 43.0934 - 1.508x_1 \\ & (x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)) \end{aligned} \tag{35}$$

Subject to:

$$g_1(\vec{x}) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0 \tag{36}$$

$$g_2(\vec{x}) = \frac{397.5}{71,785x_1^4} - 1 \leq 0 \tag{37}$$

$$g_3(\vec{x}) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leq 0 \tag{38}$$

$$g_4(\vec{x}) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leq 0 \tag{39}$$

$$g_5(\vec{x}) = \frac{\left[ \left( 745 \left( \frac{x_4}{x_2x_3} \right) \right)^2 + 16.9 \times 10^6 \right]^{1/2}}{110x_6^3} - 1 \leq 0 \tag{40}$$

$$g_6(\vec{x}) = \frac{\left[ \left( 745 \left( \frac{x_5}{x_2x_3} \right) \right)^2 + 157.5 \times 10^6 \right]^{1/2}}{85x_7^3} - 1 \leq 0 \tag{41}$$

$$g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \leq 0 \tag{42}$$

$$g_8(\vec{x}) = \frac{5x_2}{x_1} - 1 \leq 0 \tag{43}$$

$$g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \leq 0 \tag{44}$$

$$g_{10}(\vec{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0 \tag{45}$$

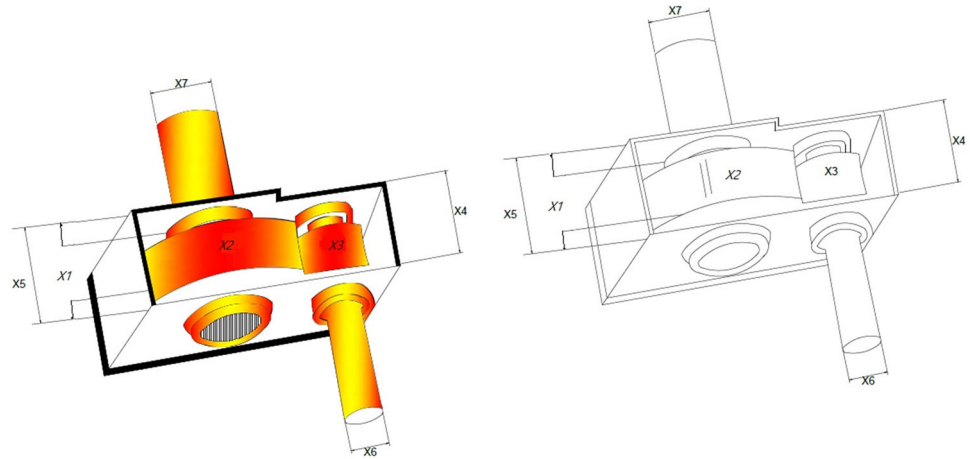
$$g_{11}(\vec{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0 \tag{46}$$

$$2.6 \leq x_1 \leq 3.6 \tag{47}$$

Table 12 Results of solving TCS design problem

Algorithm	$x_1$	$x_2$	$x_3$	$g_1$	$g_2$	$g_3$	$g_4$	$f_{min}$
CC (Arora 2004)	0.050000	0.315900	14.250000	- 1.267270E - 03	- 3.782126E - 03	- 3.9383018	- 0.7560667	0.0128334
GA (Coello Coello 2000)	0.051480	0.351661	11.632201	- 3.366613E - 03	- 1.097013E - 04	- 4.0263181	- 0.7312393	0.0127048
HS (Chakraborty, et al. 2009)	0.051154	0.349871	12.076432	- 5.222927E - 02	1.369146E - 02	- 3.8601180	- 0.7326500	0.0128872
CSCA (Huang et al. 2007)	0.051609	0.354714	11.410831	- 3.864163E - 05	- 1.828934E - 04	- 4.0486266	- 0.7291180	0.0126702
PSO (He and Wang 2007a)	0.051728	0.357644	11.244543	- 8.250947E - 04	- 2.527414E - 05	- 4.0513067	- 0.7270853	0.0126747
CPHO (He and Wang 2007a)	0.051728	0.357644	11.244543	- 8.250947E - 04	- 2.527414E - 05	- 4.0513067	- 0.7270853	0.0126747
ES (Mezura-Montes and Coello 2008)	0.051643	0.355360	11.397926	- 1.732813E - 03	- 5.667214E - 04	- 4.0393009	- 0.7286647	0.0126978
RO (Kaveh and Khayatizad 2012)	0.051370	0.349096	11.762790	- 1.089702E - 03	1.292176E - 05	- 4.0330497	- 0.7330227	0.0126786
WOA (Mirjalili and Andrew 2016)	0.051207	0.345215	12.004032	- 5.644802E - 04	- 3.699671E - 05	- 4.0274136	- 0.7357187	0.0126766
GSA (Rashedi et al. 2009)	0.050276	0.323680	13.525410	- 4.941516E - 05	3.773418E - 05	- 3.9831079	- 0.7506960	0.0127022
MVO (Mirjalili et al. 2016)	0.052510	0.376020	10.335130	- 6.808796E - 03	- 1.660788E - 03	- 4.0469143	- 0.7143133	0.0127891
OBSCA (Abd Elaziz et al. 2017)	0.052300	0.317280	12.548540	2.537560E-01	- 1.179261E - 01	- 4.8149347	- 0.7536133	0.0126260
AOA (Abualigah et al. 2021a)	0.050000	0.349809	11.863700	- 1.318774E - 01	8.033800E - 02	- 3.8373736	- 0.7334607	0.0121241
<b>GMO</b>	<b>0.051792</b>	<b>0.359198</b>	<b>11.145041</b>	<b>0</b>	<b>0</b>	<b>- 4.0586559</b>	<b>- 0.7260069</b>	<b>0.0126654</b>

**Fig. 9** Speed reducer scheme (Abualigah et al. 2021b)



**Table 13** Results of solving speed reducer design problem

Algorithm	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$f_{min}$
MFO (Mirjalili 2015)	3.497455	0.700000	17.000000	7.827750	7.712457	3.351787	5.286352	2998.940830
WSA (Baykasoğlu and Akpınar 2017)	3.500000	0.700000	17.000000	7.300000	7.800000	3.350215	5.286683	2996.348225
LGSI4 (Baykasoğlu and Akpınar 2017)	3.501000	0.700000	17.000000	7.300000	7.800000	3.350214	5.286683	2996.348205
AAO (Czerniak et al. 2017)	3.499000	0.699900	17.000000	7.300000	7.800000	3.350200	5.287200	2996.783000
PSO-DE (Liu et al. 2010)	3.500000	0.700000	17.000000	7.300000	7.800000	3.350210	5.286680	2996.348100
LGSI2 (Baykasoğlu and Akpınar 2017)	3.500000	0.700000	17.000000	7.300000	7.800000	3.350215	5.286683	2996.348166
GWO (Mirjalili et al. 2014)	3.501000	0.700000	17.000000	7.300000	7.811013	3.350704	5.287411	2997.819650
APSO (Ben Guedria 2016)	3.501313	0.700000	18.000000	8.127814	8.042121	3.352446	5.287076	3187.630486
CS (Gandomi et al. 2013)	3.501500	0.700000	17.000000	7.605000	7.818100	3.352000	5.287500	3000.981000
FA (Baykaso and Ozsoydan 2015)	3.507495	0.700100	17.000000	7.719674	8.080854	3.351512	5.287051	3010.137492
AOA (Abualigah et al. 2021a)	3.503840	0.700000	17.000000	7.300000	7.729330	3.356490	5.286700	2997.915700
<b>GMO</b>	<b>3.500000</b>	<b>0.700000</b>	<b>17.000000</b>	<b>7.300000</b>	<b>7.727800</b>	<b>3.350200</b>	<b>5.286700</b>	<b>2994.749500</b>

$$0.7 \leq x_2 \leq 0.8 \tag{48}$$

$$17 \leq x_3 \leq 28 \tag{49}$$

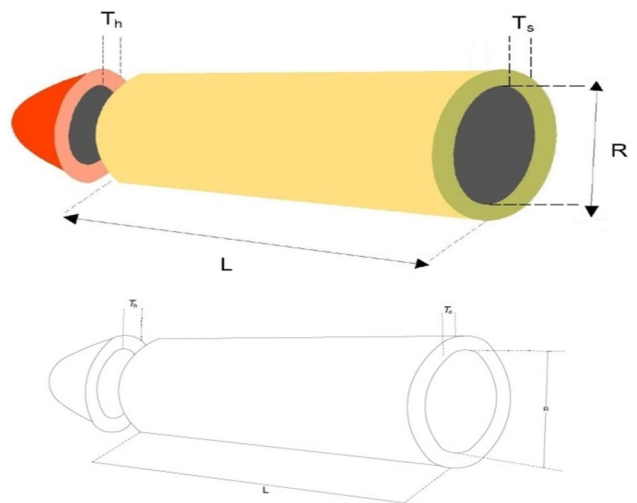
$$7.3 \leq x_4 \leq 8.3 \tag{50}$$

$$7.3 \leq x_5 \leq 8.3 \tag{51}$$

$$2.9 \leq x_6 \leq 3.9 \tag{52}$$

$$5.0 \leq x_7 \leq 5.6 \tag{53}$$

Table 13 shows the final results calculated by the algorithms upon solving this problem. As can be seen, the proposed GMO algorithm reaches the minimum weight of the speed reducer achieved by all the competitors. The success of the proposed algorithm in solving such complex problem confirms the strength of the GMO and the novel approach it has utilized to effectively explore and intensively exploit the search space when dealing with an optimization problem.



**Fig. 10** Pressure vessel scheme (Abualigah et al. 2021b)

### 4.4 Pressure vessel design problem

The goal of this problem is to reach the minimum fabrication cost of a pressure vessel depicted in Fig. 10. The design variables are  $T_s, T_h, R,$  and  $L,$  represented by  $x_1, x_2, x_3,$  and  $x_4,$  respectively, in the problem formulation which is presented as follows.

$$\text{Minimize } f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_2 + 19.84x_1^2x_3 \tag{54}$$

Subject to:

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0 \tag{55}$$

$$g_2(\vec{x}) = -x_2 + 0.00954x_3 \leq 0 \tag{56}$$

$$g_3(\vec{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0 \tag{57}$$

$$g_4(\vec{x}) = x_4 - 240 \leq 0 \tag{58}$$

$$0 \leq x_1 \leq 99 \tag{59}$$

$$0 \leq x_2 \leq 99 \tag{60}$$

$$10 \leq x_3 \leq 200 \tag{61}$$

$$10 \leq x_4 \leq 200 \tag{62}$$

The results of solving this problem by several examined algorithms are shown in Table 14. As can be seen, the GMO can find the lowest cost among the other competitive algorithms. As is clear in the problem formulation, this problem is a highly constrained one and the superiority of the proposed GMO to its rivals demonstrates its high competence to solve such problems.

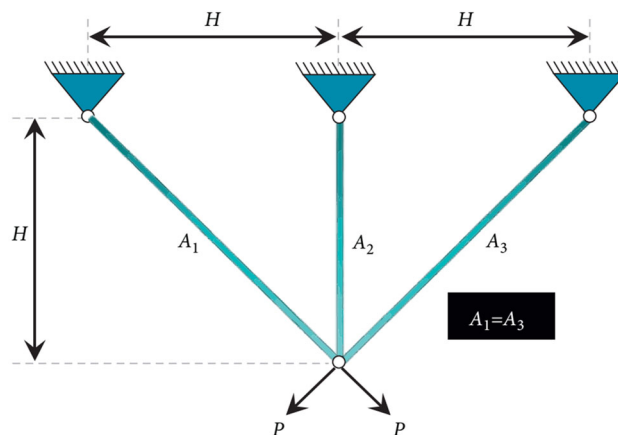


Fig. 11 Three-bar truss structure (Abd Elaziz et al. 2017)

### 4.5 Three-bar truss design problem

The goal of this problem is minimizing the weight of a three-bar truss subject to stress, deflection, and buckling constraints. This truss has three variable cross section areas marked by  $A_1, A_2,$  and  $A_3.$  As depicted in Fig. 11,  $A_1 = A_3.$  Thus, there are only two areas as the decision variables to be optimized which are denoted by  $x_1$  and  $x_2$  in the problem formulation presented below.

$$\text{Minimize } f(\vec{x}) = (2\sqrt{2}x_1 + x_2) \times l \tag{63}$$

Subject to:

$$g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0 \tag{64}$$

$$g_2(\vec{x}) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0 \tag{65}$$

Table 14 Results of solving pressure vessel design problem

Algorithm	$x_1$	$x_2$	$x_3$	$x_4$	$f_{min}$
Branch-bound (Sandgren 1990)	1.125000	0.625000	48.970000	106.720000	7982.500000
GA (Coello Coello 2000)	0.812500	0.437500	42.097398	176.654050	6059.946340
HS (Mahdavi et al. 2007)	1.125000	0.625000	58.290150	43.692680	7197.730000
CSCA (Huang et al. 2007)	0.812500	0.437500	42.098411	176.637690	6059.734000
PSO-SCA (Liu et al. 2010)	0.812500	0.437500	42.098446	176.636600	6059.714330
CPSO (He and Wang 2007a)	0.812500	0.437500	42.098400	176.636600	6059.714300
HPSO (He and Wang 2007b)	0.812500	0.437500	42.098087	176.640518	6059.745600
ES (Mezura-Montes and Coello 2008)	0.812500	0.437500	42.098353	176.637751	6059.725800
ACO (Kaveh and Talatahari 2010)	0.812500	0.437500	42.098353	176.637751	6059.725800
WOA (Mirjalili and Andrew 2016)	0.812500	0.437500	42.098270	176.638998	6059.741000
GSA (Rashedi et al. 2009)	1.125000	0.625000	55.988660	84.454203	8538.835900
MVO (Mirjalili et al. 2016)	0.812500	0.437500	42.090738	176.738690	6060.806600
AOA (Abualigah et al. 2021a)	0.830374	0.416206	42.751270	169.345400	6048.784400
<b>GMO</b>	<b>0.8</b>	<b>0.4</b>	<b>41.441252</b>	<b>184.954065</b>	<b>5938.866432</b>

**Table 15** Results of solving three-bar truss design problem

Algorithm	$x_1$	$x_2$	$g_1$	$g_2$	$g_3$	$f_{min}$
DEDS (Zhang et al. 2008)	0.788675	0.408248	<u>1.777971E - 08</u>	- 1.464102	- 0.535898	263.895841
SSA (Mirjalili et al. 2017)	0.788665	0.408276	<u>1.247906E - 08</u>	- 1.464070	- 0.535930	263.895842
MBA (Sadollah et al. 2013)	0.788565	0.408560	<u>1.418869E - 07</u>	- 1.463748	- 0.536252	263.895834
PSO-DE (Liu et al. 2010)	0.788675	0.408248	<u>1.427175E - 07</u>	- 1.464102	- 0.535898	263.895825
Tsa (Tsai 2005)	0.788000	0.408000	<u>1.636731E - 03</u>	- 1.463566	- 0.534798	263.680057
Rai and Saini (Ray and Saini 2001)	0.795000	0.395000	-3.375515E-03	- 1.480901	- 0.522474	264.359956
CS (Gandomi et al. 2013)	0.788670	0.409020	- 5.733901E - 04	- 1.463512	- 0.537062	263.971562
MFO (Mirjalili 2015)	0.788245	0.409467	- 1.190244E - 09	- 1.462717	- 0.537283	263.895980
AOA (Abualigah et al. 2021a)	0.793690	0.394260	- 1.103007E - 05	- 1.480113	- 0.519898	263.915432
<b>GMO</b>	<b>0.788671</b>	<b>0.408261</b>	<b>- 4.678179E - 09</b>	<b>- 1.464088</b>	<b>- 0.535912</b>	<b>263.895844</b>

$$g_3(\vec{x}) = \frac{1}{\sqrt{2x_2 + x_1}} P - \sigma \leq 0 \tag{66}$$

$$0 \leq x_1, x_2 \leq 1 \tag{67}$$

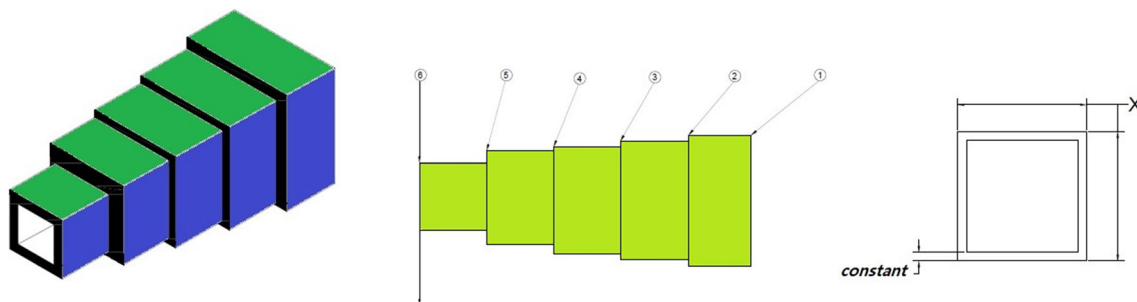
where

$$l = 100\text{cm}, P = 2 \frac{KN}{\text{cm}^2}, \sigma = 2 \frac{KN}{\text{cm}^2} \tag{68}$$

Table 15 shows the results obtained by numerous algorithms employed to solve this problem. As this table suggest, the results found by the first five algorithms in the table, violate the constraint  $g_1$ . As a result, their optimal objective function values are not validated. The constraints violated are underlined in the table. Among the remaining four algorithms; the MFO finds the best objective value, however, it is larger than the value obtained by the proposed GMO. Accordingly, the GMO is superior to all its competitors on such a problem with a difficult constrained search space.

### 4.6 Cantilever beam design problem

The objective of this problem is to minimize the weight of a cantilever beam consisting of five hollow blocks with variable heights and constant thicknesses, depicted in



**Fig. 12** Cantilever beam problem (Mahdavi et al. 2007)

Fig. 12. Thus, there are five decision variables needed to be optimized denoted by  $x_1$  to  $x_5$  as the blocks' heights. This problem is formulated as follows.

$$\text{Minimize } f(\vec{x}) = 0.06224(x_1 + x_2 + x_3 + x_4 + x_5) \tag{69}$$

Subject to:

$$g(\vec{x}) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \tag{70}$$

$$0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100 \tag{71}$$

As the results obtained by several algorithms including the GMO suggest in Table 16, the proposed GMO reaches the minimum weight of the cantilever beam among those obtained by all the other competitors.

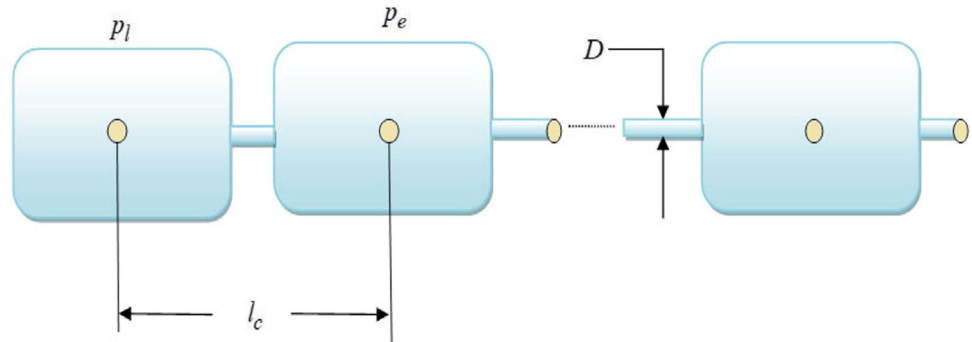
### 4.7 Gas transmission compressor design problem

The goal of this problem is to minimize the cost of delivering 100 million cft gas per a day. The decision variables  $x_1, x_2,$  and  $x_3$  are the gas transmission parameters expressed as the distance between the two compressors, the compression ratio value, and value of the inside diameter of the

**Table 16** Results of solving cantilever beam design problem

Algorithm	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$f_{min}$
MFO (Mirjalili 2015)	5.98487	5.31673	4.49733	3.51362	2.16162	1.33999
SOS (Cheng and Prayogo 2014)	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
CS (Gandomi et al. 2013)	6.00890	5.30490	4.50230	3.50770	2.15040	1.33999
MMA (Chickermane and GEA 1996)	6.01000	5.30000	4.49000	3.49000	2.15000	1.34000
GCA1 (Chickermane and GEA 1996)	6.01000	5.30000	4.49000	3.49000	2.15000	1.34000
GCA2 (Chickermane and GEA 1996)	6.01000	5.30000	4.49000	3.49000	2.15000	1.34000
<b>GMO</b>	<b>6.01517</b>	<b>5.30565</b>	<b>4.50197</b>	<b>3.50181</b>	<b>2.14910</b>	<b>1.33652</b>

**Fig. 13** Gas transmission compressor design problem (Kumar et al. 2021)



**Table 17** Results of solving gas transmission compressor design problem

Algorithm	$x_1$	$x_2$	$x_3$	$f_{min}$
ECS-AGQPSO (Kumar et al. 2021)	53.446716	1.190101	24.718579	2,964,375.495330
RCSOMGA (Duary et al. 2020)	53.446827	1.190101	24.718580	2,964,375.495330
SOMA (Duary et al. 2020)	53.347298	1.190142	24.737115	2,964,378.729000
RCGA (Duary et al. 2020)	53.520217	1.190361	24.723656	2,964,375.725000
PSO (PANT et al. 2011)	55.000000	1.195410	24.774900	2,964,460.000000
DE (PANT et al. 2011)	51.985700	1.183350	24.719500	2,964,480.000000
DE-PSO (PANT et al. 2011)	53.447400	1.190100	24.718600	2,964,375.503101
GP (Beightler and Phillips 1976)	52.600000	1.187000	24.800000	2,964,419.625000
<b>GMO</b>	<b>53.446787</b>	<b>1.190101</b>	<b>24.718577</b>	<b>2,964,375.495329</b>

gas pipe, respectively. A schematic image of this problem is shown in Fig. 13. This problem is formulated as follows.

$$\begin{aligned}
 \text{Minimize } f(\vec{x}) = & 3.69 \times 10^4 \times x_3 + 7.72 \times 10^8 x_1^{-1} x_2^{0.219} \\
 & - 765.43 \times 10^6 \times x_1^{-1} + 8.61 \times 10^5 \\
 & \times x_1^{\frac{1}{2}} x_2 (x_2^2 - 1)^{-\frac{1}{2}} x_3^{-\frac{2}{3}}
 \end{aligned} \tag{72}$$

Subject to:

$$10 \leq x_1 \leq 55 \tag{73}$$

$$1.1 \leq x_2 \leq 2 \tag{74}$$

$$10 \leq x_3 \leq 40 \tag{75}$$

As can be seen in Table 17, the proposal can calculate the best objective function value, contributing the GMO to be superior to all of its rivals taking part in this very close competition.

#### 4.8 Himmelblau’s nonlinear constrained optimization problems (version-I)

This problem is a highly constrained engineering problem, widely used in the literature to examine the eligibility of the optimization algorithms. The problem has two versions. The first version of this problem is formulated as follows.

$$\begin{aligned}
 \text{Minimize } f(\vec{x}) = & 0.8356891x_1x_2 + 37.293239x_1 \\
 & - 40792.141 + 5.3578547x_3^2
 \end{aligned} \tag{76}$$

Subject to:

$$0 \leq g_1(\vec{x}) \leq 92 \tag{77}$$

$$90 \leq g_2(\vec{x}) \leq 110 \tag{78}$$

$$20 \leq g_3(\vec{x}) \leq 25 \tag{79}$$

where

$$g_1(\vec{x}) = 85.334407 + 0.0006262x_1x_4 - 0.0022053x_3x_5 + 0.0056858x_2x_5 \tag{80}$$

$$g_2(\vec{x}) = 80.51249 + 0.0029955x_1x_2 - 0.0021813x_3^2 + 0.0071317x_2x_5 \tag{81}$$

$$g_3(\vec{x}) = 9.300961 + 0.0012547x_1x_3 + 0.0019085x_3x_4 + 0.0047026x_3x_5 \tag{82}$$

$$78 \leq x_1 \leq 102 \tag{83}$$

$$33 \leq x_2 \leq 45 \tag{84}$$

$$27 \leq x_3 \leq 45 \tag{85}$$

$$27 \leq x_4 \leq 45 \tag{86}$$

$$27 \leq x_5 \leq 45 \tag{87}$$

Table 18 shows the optimal results each of the examined algorithms can find when solving this problem. While the optimal objective values calculated by all the comparative algorithms are so close to each other, the proposed GMO algorithm can achieve significantly lowest objective value compared to the other competitors, demonstrating high merits of the proposal to solve such problem.

### 4.9 Himmelblau’s nonlinear constrained optimization problems (version-II)

A group of researchers used 0.00026 instead of 0.0006262 as the coefficient of the second term in the first constraint, resulting in the second version of the Himmelblau’s problem. Table 19 shows the optimal results that several algorithms can find for this problem. The results of three algorithms violate the constraint  $g_1$  and are invalid. The constraints violated are underlined in the table. Among all examined algorithms, the proposed GMO can reach the far lowest objective value and thus significantly outperforms all its competitors.

As can be seen, the GMO is superior to the numerous optimizers when solving all of the engineering problems. These problems are highly constrained and these constraints strongly limit the search space. Since the optimal solution of these problems must be located in the feasible region whose boundaries are delineated by the numerous constraints embedded in the problems’ formulations, superiority of the proposed GMO in solving these problems indicates the great and promising performance of this algorithm. This means that the GMO not only can desirably solve a variety of the standard benchmark test problems,

**Table 18** Results of solving Himmelblau’s problem (version-I)

Algorithm	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$g_1$	$g_2$	$g_3$	$f_{min}$
ECS-AGQPSO (Kumar et al. 2021)	78.000000	33.000000	29.995256	45.000000	36.775813	92.000000	94.915402	20.000000	− 30,665.538672
T1-AGQPSO (Kumar et al. 2020)	78.000000	33.000000	29.995250	45.000000	36.775813	−	−	−	− 30,665.538700
GSA-GA (Garg 2019)	77.960999	32.991971	29.992126	45.000000	36.794135	91.999701	94.912286	20.000000	− 30,668.004000
GA-ACO-PSO (Tam et al. 2019)	78.000000	33.000000	29.995300	45.000000	36.775800	92.000000	98.840500	20.000000	− 30,665.538700
CS (Gandomi et al. 2013)	78.000000	33.000000	29.996160	45.000000	36.776050	91.999960	98.840670	20.000360	− 30,665.233000
Nelder-Mead (Mehta and Dasgupta 2012)	78.000000	33.000000	29.995300	45.000000	36.775800	−	−	−	− 30,665.538700
PSOA/PSOStr (Dimopoulos 2007)	78.000000	33.000000	29.995256	45.000000	36.775813	92.000000	98.840500	20.000000	− 30,665.540000
HM (Lee and Geem 2005)	78.000000	33.000000	29.995000	45.000000	36.776000	92.000000	98.840500	<u>19.999900</u>	− 30,665.500000
GA (He and Wang 2007a)	−	−	−	−	−	−	−	−	− 30,665.500000
Himmelblau (Sharma 2006)	−	−	−	−	−	−	−	−	− 30,373.949000
<b>GMO</b>	<b>78.000001</b>	<b>33.000000</b>	<b>29.997840</b>	<b>42.803207</b>	<b>37.660825</b>	<b>92.000000</b>	<b>95.123348</b>	<b>20.000000</b>	<b>− 30,910.829843</b>

**Table 19** Results of solving Himmelblau's problem (version-II)

Algorithm	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$g_1$	$g_2$	$g_3$	$f_{min}$
ECS-AGQPSO (Kumar et al. 2021)	78.000000	33.000000	27.071000	45.000000	44.969200	92.000000	97.207700	20.000000	- 31,025.560242
T1-AGQPSO (Kumar et al. 2020)	78.000000	33.000000	27.071000	45.000000	44.969200	-	-	-	- 31,025.560200
GSA-GA (Garg 2019)	77.961000	32.999500	27.072800	45.000000	44.973900	91.999700	97.204181	20.000000	- 31,027.640800
GA-ACO-PSO (Tam et al. 2019)	78.000000	33.000000	27.071000	45.000000	44.969200	92.000000	100.404800	20.000000	- 31,025.560200
CODEQ (Omran and Salman 2009)	78.000000	33.000000	27.071000	45.000000	44.969200	<u>93.285400</u>	100.404800	20.000000	- 31,025.556300
HHSA (Fesanghary et al. 2008)	78.000000	33.000000	27.085100	45.000000	44.925300	<u>93.278300</u>	100.396100	20.000000	- 31,024.316600
Coello (COELLO 2000)	78.595800	33.010000	27.646000	45.000000	45.000000	91.956400	100.545100	20.251900	- 30,810.359000
Coello (Coello Coello 2000)	78.049500	33.010000	27.646000	45.000000	44.940000	<u>93.283800</u>	100.407900	20.001900	- 31,020.859000
<b>GMO</b>	<b>78.000001</b>	<b>33.000000</b>	<b>27.071109</b>	<b>44.998242</b>	<b>44.969608</b>	<b>92.000000</b>	<b>97.207774</b>	<b>20.000000</b>	<b>- 31,805.727802</b>

but can be noticed as a favorite algorithm to solve engineering problems as the real-world cases in the optimization field.

The key factor contributing to such performance of the GMO may be hidden in the new approach this algorithm takes when designating the guide search agents. In GMO, the abilities to measure the fitness and diversity as two key factors to be focused in selecting or computing the guide solutions are integrated. This integration is carried out using the geometric mean operator in mathematics whose unique properties to compute the mean of a number of variables are found highly effective in designating a weight for each solution in the search space to determine its overall eligibility to guide the other solutions. Presenting such comprehensive weights can highly assure the algorithm that the most competent search agents are computed as the guides. These guides are consistently calculated so throughout the optimization process, whether the GMO is in the exploration or exploitation phase. Since there is no boundary between these two major phases of the optimization process in the GMO, there is no chance for the possible disruptions and interruptions occurring when needing the algorithm to show more exploration or exploitation activities in some stages of the optimization, contributing all the movements of the search agents to be made balanced and reliable to help the proposal accomplish the optimization process in the best way. Imposing

mutation on the guides to mitigate their intrinsic uncertainty, embedding an elitism mechanism in the algorithm, gradually propelling the search agents toward their guides, and having the efficient computations for this algorithm are all among the other facilities the proposed algorithm benefits from. Hence, the GMO can be strongly recommended as one of the best algorithms in the field of meta-heuristic optimization.

## 5 The pros and cons of the proposed GMO

As the results of solving global optimization problems (classical benchmark functions and the CEC2017 test suite) as well as the constrained engineering design problems clearly indicated, the high capabilities of the proposed GMO algorithm to both explore and exploit the search space of the optimization problems with a variety of search domains, dimensionality, and difficulties are demonstrated. Furthermore, the GMO is capable of handling complicated engineering design problems with a large number of constraints, stressing its power to effectively explore the search space and detect the real position of the optimal solution wherever it is placed in the domain. Among the advantages of the proposed GMO, its ability to simultaneously evaluate the fitness and diversity by means of the dual-fitness index, computational efficiency, having no parameter to



tune, and designating multiple and distinct guides for every solution as a way to help the algorithm avoid premature convergence, are the most eminent and effective ones. However, there are some minor weaknesses that can be handled such as lack of a strategy to set the accurate value of decrease in the steps of the guides when mutated, lack of a self-learning mechanism to accurately determine the rate of decay in the inertia weight multiplied by the velocity term, and occasional occurrence of singularity when calculating the guide positions that makes the algorithm to add  $\varepsilon$  to the denominator of the guides when handling the unimodal or simple problems and not to add  $\varepsilon$  when dealing with the real-world and more complicated problems.

## 6 Conclusion

This paper presented a novel meta-heuristic optimization algorithm, called geometric mean optimizer (GMO), inspired by the unique properties of the geometric mean operator in mathematics. The geometric mean operator can simultaneously represent the average and similarity of several variables of the same scale. If these variables are assumed as the normalized objective function values of a number of the opposites of a focused search agent in the search space, the geometric mean of these variables can simultaneously represent the fitness and diversity of the focused agent. The higher this geometric mean, the higher the similarity between the opposite agents, and thus, the higher the density of the area these agents are distributed in. Simultaneously, the higher the geometric mean of the opposite agents of an agent, the higher the average objective values of these agents. In this way, the focused agent may be more diversified and has the lower objective/higher fitness value at the same time, in a minimization process. This geometric mean operator, which is turned into a root-free form and named the pseudo-geometric mean operator in the present work, can be employed as a weight assigned to each personal best-so-far agent found by the GMO algorithm at each of the iterations. Then, a weighted averaging process is conducted over the opposite agents of a focused agent and the resulting average agent is adopted as the unique and only guide of the focused search agent. In this algorithm, the guides are all imposed a Gaussian mutation operator. This act can help further strengthen the exploration ability of the proposed GMO.

The strengths of the proposed GMO algorithm to handle the optimization problems can be summarized in five main points: (1) integrating the abilities to evaluate fitness and diversity of the search agents to yield them a weight to better diversify the search space while facilitating access to the high-fitness areas; (2) mitigating the uncertainty of the guide agents via mutating them to impede the GMO to get

trapped in prematurely identified local optima while reducing the possibility of drift occurrence during the search agents' movements toward their guides; (3) incorporating an elitism mechanism into the GMO to further enhance the exploitation capability and expedite its convergence to the optimum; (4) considering the velocity for each search agent to keep pushing the agents to continue their exploration in the search space; and (5) having no parameter to tune.

The proposed GMO was then validated via comparison with two sets of classical, newly proposed and popular meta-heuristics, with reference to some performance criteria. The results obtained in solving 52 standard benchmark test problems and nine engineering problems demonstrated the superiority of the proposal to all its competitors.

As future work, it is suggested to apply GMO to solve some large-scale and broader engineering problems to further investigate the strengths and weaknesses of this algorithm in finding the optimum of such problems to remove the shortcomings the proposal may suffer from.

**Author's Contributions** The authors contributed to each part of this paper equally. The authors read and approved the final manuscript.

**Funding** There is no fund for this paper.

**Data availability** Data will be made available on reasonable request.

## Declarations

**Conflict of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- Abd Elaziz M, Oliva D, Xiong S (2017) An improved opposition-based sine cosine algorithm for global optimization. *Expert Syst Appl* 90:484–500. <https://doi.org/10.1016/j.eswa.2017.07.043>
- Abualigah L, Diabat A, Mirjalili S et al (2021a) The arithmetic optimization algorithm. *Comput Methods Appl Mech Eng* 376:113609. <https://doi.org/10.1016/j.cma.2020.113609>
- Abualigah L, Elaziz MA, Sumari P et al (2021b) Reptile Search Algorithm (RSA): a nature-inspired meta-heuristic optimizer. *Expert Syst Appl* 191:116158. <https://doi.org/10.1016/j.eswa.2021.116158>
- Abualigah L, Yousri D, Abd Elaziz M et al (2021c) Aquila optimizer: a novel meta-heuristic optimization algorithm. *Comput Ind Eng* 157:107250. <https://doi.org/10.1016/j.cie.2021.107250>
- Ahmadianfar I, Bozorg-Haddad O, Chu X (2020) Gradient-based optimizer: a new metaheuristic optimization algorithm. *Inf Sci (ny)* 540:131–159. <https://doi.org/10.1016/j.ins.2020.06.037>
- Arora J (2004) *Introduction to optimum design*. Elsevier, Netherlands
- Baykaso A, Ozsoydan FB (2015) Adaptive firefly algorithm with chaos for mechanical design optimization problems. *Appl Soft Comput* 36:152–154

- Baykasoğlu A, Akpınar Ş (2017) Weighted Superposition Attraction (WSA): a swarm intelligence algorithm for optimization problems – Part 1: Unconstrained optimization. *Appl Soft Comput* 56:520–540. <https://doi.org/10.1016/j.asoc.2015.10.036>
- Bightler CS, Phillips DT (1976) *Applied geometric programming*. John Wiley & Sons, UK
- Ben Guedria N (2016) Improved accelerated PSO algorithm for mechanical engineering optimization problems. *Appl Soft Comput* 40:455–467. <https://doi.org/10.1016/j.asoc.2015.10.048>
- Bui DT, Moayedi H, Kalantar B et al (2019) Harris Hawks optimization: a novel swarm intelligence technique for spatial assessment of landslide susceptibility. *Sensors* 19:3590. <https://doi.org/10.3390/s19163590>
- Chakraborty P, Roy GG, Das S et al (2009) An improved harmony search algorithm with differential mutation operator. *Fundam Informaticae* 95:401–426. <https://doi.org/10.3233/FI-2009-157>
- Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput Struct* 139:98–112. <https://doi.org/10.1016/j.compstruc.2014.03.007>
- Chickermane H, Gea HC (1996) Structural optimization using a new local approximation method. *Int J Numer Methods Eng* 39:829–846. [https://doi.org/10.1002/\(SICI\)1097-0207\(19960315\)39:5%3c829::AID-NME884%3e3.0.CO;2-U](https://doi.org/10.1002/(SICI)1097-0207(19960315)39:5%3c829::AID-NME884%3e3.0.CO;2-U)
- Coello CAC (2000) Treating constraints as objectives for single-objective evolutionary optimization. *Eng Optim* 32:275–308. <https://doi.org/10.1080/03052150008941301>
- Coello Coello CA (2000) Use of a self-adaptive penalty approach for engineering optimization problems. *Comput Ind* 41:113–127. [https://doi.org/10.1016/S0166-3615\(99\)00046-9](https://doi.org/10.1016/S0166-3615(99)00046-9)
- Czerniak JM, Zarzycki H, Ewald D (2017) AAO as a new strategy in modeling and simulation of constructional problems optimization. *Simul Model Pract Theory* 76:22–33. <https://doi.org/10.1016/j.simpat.2017.04.001>
- Deb K (1991) Optimal design of a welded beam via genetic algorithms. *AIAA J* 29:2013–2015. <https://doi.org/10.2514/3.10834>
- Dimopoulos GG (2007) Mixed-variable engineering optimization based on evolutionary and social metaphors. *Comput Methods Appl Mech Eng* 196:803–817. <https://doi.org/10.1016/j.cma.2006.06.010>
- Duary A, Rahman MS, Shaikh AA et al (2020) A new hybrid algorithm to solve bound-constrained nonlinear optimization problems. *Neural Comput Appl* 32:12427–12452. <https://doi.org/10.1007/s00521-019-04696-7>
- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: *Micro Machine and Human Science, 1995. MHS'95.*, In: *Proceedings of the sixth international symposium on*. pp 39–43
- El AMA, Ewees AA, Hassanien AE (2017) Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation. *Expert Syst Appl* 83:242–256. <https://doi.org/10.1016/j.eswa.2017.04.023>
- Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S (2020) Equilibrium optimizer: a novel optimization algorithm. *Knowledge-Based Syst* 191:105190–105454. <https://doi.org/10.1016/j.knsys.2019.105190>
- Fesanghary M, Mahdavi M, Minary-Jolandan M, Alizadeh Y (2008) Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems. *Comput Methods Appl Mech Eng* 197:3080–3091. <https://doi.org/10.1016/j.cma.2008.02.006>
- Gandomi AH (2014) Interior search algorithm (ISA): a novel approach for global optimization. *ISA Trans* 53:1168–1183. <https://doi.org/10.1016/j.isatra.2014.03.018>
- Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng Comput* 29:17–35. <https://doi.org/10.1007/s00366-011-0241-y>
- Garg H (2019) A hybrid GSA-GA algorithm for constrained optimization problems. *Inf Sci* 478:499–523. <https://doi.org/10.1016/j.ins.2018.11.041>
- Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. *Mach Learn* 3:95–99. <https://doi.org/10.1023/A:1022602019183>
- Hatamlou A (2013) Black hole: a new heuristic optimization approach for data clustering. *Inf Sci* 222:175–184. <https://doi.org/10.1016/j.ins.2012.08.023>
- He Q, Wang L (2007a) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng Appl Artif Intell* 20:89–99. <https://doi.org/10.1016/j.engappai.2006.03.003>
- He Q, Wang L (2007b) A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Appl Math Comput* 186:1407–1422. <https://doi.org/10.1016/j.amc.2006.07.134>
- He S, Wu QH, Saunders JR (2009) Group search optimizer: an optimization algorithm inspired by animal searching behavior. *IEEE Trans Evol Comput* 13:973–990. <https://doi.org/10.1109/TEVC.2009.2011992>
- Heidari AA, Mirjalili S, Faris H et al (2019) Harris hawks optimization: algorithm and applications. *Futur Gener Comput Syst* 97:849–872. <https://doi.org/10.1016/j.future.2019.02.028>
- Hosseini HS (2011) Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. *Int J Comput Sci Eng* 6:132. <https://doi.org/10.1504/IJCSE.2011.041221>
- Huang FZ, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. *Appl Math Comput* 186:340–356. <https://doi.org/10.1016/j.amc.2006.07.105>
- Karami H, Anaraki MV, Farzin S, Mirjalili S (2021) Flow direction algorithm (FDA): a novel optimization approach for solving optimization problems. *Comput Ind Eng* 156:107224–115115. <https://doi.org/10.1016/j.cie.2021.107224>
- Kaveh A, Khayatazad M (2012) A new meta-heuristic method: ray optimization. *Comput Struct* 112–113:283–294. <https://doi.org/10.1016/j.compstruc.2012.09.003>
- Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: a novel meta-heuristic method. *Comput Struct* 139:18–27. <https://doi.org/10.1016/j.compstruc.2014.04.005>
- Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. *Eng Comput* 27:155–182. <https://doi.org/10.1108/02644401011008577>
- Kumar N, Mahato SK, Bhunia AK (2020) A new QPSO based hybrid algorithm for constrained optimization problems via tournament process. *Soft Comput* 24:11365–11379. <https://doi.org/10.1007/s00500-019-04601-3>
- Kumar N, Mahato SK, Bhunia AK (2021) Design of an efficient hybridized CS-PSO algorithm and its applications for solving constrained and bound constrained structural engineering design problems. *Results Control Optim* 5:100064. <https://doi.org/10.1016/j.rico.2021.100064>
- Kuo RJ, Zulueta FE (2015) The gradient evolution algorithm: a new metaheuristic. *Inf Sci* 316:246–265. <https://doi.org/10.1016/j.ins.2015.04.031>
- Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Comput Methods Appl Mech Eng* 194:3902–3933.

- <https://doi.org/10.1016/j.cma.2004.09.007>
- Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Appl Soft Comput J* 10:629–640. <https://doi.org/10.1016/j.asoc.2009.08.031>
- Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. *Appl Math Comput* 188:1567–1579. <https://doi.org/10.1016/j.amc.2006.11.033>
- Mehta VK, Dasgupta B (2012) A constrained optimization algorithm based on the simplex search method. *Eng Optim* 44:537–550. <https://doi.org/10.1080/0305215X.2011.598520>
- Mezura-Montes E, Coello CAC (2005) Useful infeasible solutions in engineering optimization with evolutionary algorithms. In: Gelbukh A, de Albornoz Á, Terashima-Marín H (eds) *MICAI 2005: Advances in Artificial Intelligence*. Springer Berlin Heidelberg, Berlin, pp 652–662. [https://doi.org/10.1007/11579427\\_66](https://doi.org/10.1007/11579427_66)
- Mezura-Montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *Int J Gen Syst* 37:443–473. <https://doi.org/10.1080/03081070701303470>
- Minh H-L, Sang-To T, Abdel Wahab M, Cuong-Le T (2022) A new metaheuristic optimization based on K-means clustering algorithm and its application to structural damage identification. *Knowledge-Based Syst* 251:109189. <https://doi.org/10.1016/j.knsys.2022.109189>
- Minh H-L, Sang-To T, Theraulaz G et al (2023) Termite life cycle optimizer. *Expert Syst Appl* 213:119211. <https://doi.org/10.1016/j.eswa.2022.119211>
- Mirjalili S (2015) Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Syst* 89:228–249. <https://doi.org/10.1016/j.knsys.2015.07.006>
- Mirjalili S, Andrew L (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
- Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Mirjalili S, Mirjalili SM, Lewis A (2014) Grey Wolf Optimizer. *Adv Eng Softw* 69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-Verse Optimizer: a nature-inspired algorithm for global optimization. *Neural Comput Appl* 27:495–513. <https://doi.org/10.1007/s00521-015-1870-7>
- Mirjalili S, Gandomi AH, Zahra S, Saremi S (2017) Salp swarm algorithm : a bio-inspired optimizer for engineering design problems. *Adv Eng Softw*. <https://doi.org/10.1016/j.advengsoft.2017.07.002>
- Omran MGH, Salman A (2009) Constrained optimization using CODEQ. *Chaos, Solitons Fractals* 42:662–668. <https://doi.org/10.1016/j.chaos.2009.01.039>
- Pant M, Thangaraj R, Abraham A (2011) DE-PSO: a new hybrid meta-heuristic for solving global optimization problems. *New Math Nat Comput* 07:363–381. <https://doi.org/10.1142/S1793005711001986>
- Paper OS (2015) The ant lion optimization algorithm for flexible process. *J Product Eng* 18:3–6
- Ragsdell KM, Phillips DT (1976) Optimal design of a class of welded structures using geometric programming. *J Eng Ind* 98:1021–1025
- Rao RV, Savsani VJ, Vakharia D (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput Des* 43:303–315
- Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. *Inf Sci* 179:2232–2248. <https://doi.org/10.1016/j.ins.2009.03.004>
- Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng Optim* 33:735–748. <https://doi.org/10.1080/03052150108940941>
- Rezaei F, Safavi HR, Mirchi A, Madani K (2017) f-MOPSO: An alternative multi-objective PSO algorithm for conjunctive water use management. *J Hydro-Environ Res* 14:1–18. <https://doi.org/10.1016/j.jher.2016.05.007>
- 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 J* 13:2592–2612. <https://doi.org/10.1016/j.asoc.2012.11.026>
- Salih SQ, Alsewari AA (2020) A new algorithm for normal and large-scale optimization problems: nomadic People Optimizer. *Neural Comput Appl* 32:10359–10386. <https://doi.org/10.1007/s00521-019-04575-1>
- Sandgren E (1990) Nonlinear integer and discrete programming in mechanical design optimization. *J Mech Des* 112:223–229. <https://doi.org/10.1115/1.2912596>
- Shami TM, Grace D, Burr A, Mitchell PD (2022) Single candidate optimizer: a novel optimization algorithm. *Evol Intell*. <https://doi.org/10.1007/s12065-022-00762-7>
- Sharma S (2006) *Applied nonlinear programming*. New Age International, India
- Simon D, Member S (2008) Biogeography-based optimization. *IEEE Trans Evolut Comput* 12:702–713
- Storn R, Price K (1997) Differential evolution : a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11:341–359
- Tam JH, Ong ZC, Ismail Z et al (2019) A new hybrid GA–ACO–PSO algorithm for solving various engineering design problems. *Int J Comput Math* 96:883–919. <https://doi.org/10.1080/00207160.2018.1463438>
- Tsai J-F (2005) Global optimization of nonlinear fractional programming problems in engineering design. *Eng Optim* 37:399–409. <https://doi.org/10.1080/03052150500066737>
- Wang H, Wu Z, Liu Y, et al (2009) Space transformation search. In: *Proceedings of the first ACM/SIGEVO summit on genetic and evolutionary computation*. ACM, New York, NY, USA, pp 537–544
- Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1:67–82. <https://doi.org/10.1109/4235.585893>
- Wu G, Rammohan M, Ponnuthurai NS (2017) Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. National University of Defense Technology, Changsha
- Yang X-S (2010) A new metaheuristic Bat-inspired Algorithm. In: González JR, Pelta DA, Cruz C, Terrazas G, Krasnogor N (eds) *nature inspired cooperative strategies for optimization (NICSO 2010)*. Springer Berlin Heidelberg, Heidelberg, pp 65–74. [https://doi.org/10.1007/978-3-642-12538-6\\_6](https://doi.org/10.1007/978-3-642-12538-6_6)
- Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. *IEEE Trans Evolut Comput* 3:82–102
- Zadeh LA (1965) Fuzzy sets. *Inf Control* 8:338–353. [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
- Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. *Inf Sci* 178:3043–3074. <https://doi.org/10.1016/j.ins.2008.02.014>

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the

author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.