



Chaotic-Based Mountain Gazelle Optimizer for Solving Optimization Problems

Priteesha Sarangi¹ · Prabhujit Mohapatra¹

Received: 17 October 2023 / Accepted: 3 March 2024
© The Author(s) 2024

Abstract

The Mountain Gazelle Optimizer (MGO) algorithm has become one of the most prominent swarm-inspired meta-heuristic algorithms because of its outstanding rapid convergence and excellent accuracy. However, the MGO still faces premature convergence, making it challenging to leave the local optima if early-best solutions neglect the relevant search domain. Therefore, in this study, a newly developed Chaotic-based Mountain Gazelle Optimizer (CMGO) is proposed with numerous chaotic maps to overcome the above-mentioned flaws. Moreover, the ten distinct chaotic maps were simultaneously incorporated into MGO to determine the optimal values and enhance the exploitation of the most promising solutions. The performance of CMGO has been evaluated using CEC2005 and CEC2019 benchmark functions, along with four engineering problems. Statistical tests like the t-test and Wilcoxon rank-sum test provide further evidence that the proposed CMGO outperforms the existing eminent algorithms. Hence, the experimental outcomes demonstrate that the CMGO produces successful and auspicious results.

Keywords Meta-heuristics · Optimization · Chaotic maps · Engineering problems

1 Introduction

Recently, several challenging non-linear optimization problems have been addressed by employing non-conventional meta-heuristic algorithms inspired by natural phenomena as conventional algorithms often fail to yield desired results in certain situations [1]. Meta-heuristic Algorithms (MAs) use social behavior of animals or natural systems to find the optimal solution for specific problems by generating a random population through stochastic processes and iterations [2]. The majority of MAs may be categorized into two main groups: Firstly, those influenced by biological processes inherent in nature and secondly, those entirely dependent on natural phenomena. Several meta-heuristics that have attracted many scholars are cited over the past 20 years. Some of the MAs which are influenced by biological process are: Evolutionary Programming (EP) [3], Evolutionary

Strategy (ES) [4], Genetic Algorithm (GA) [5], Genetic Programming (GP) [6], Differential Evolution (DE) [7], Particle Swarm Optimization (PSO) [8], Ant Colony Optimization (ACO) [9], Cuckoo Search (CS) [10], Artificial Bee Colony (ABC) [11], Grey Wolf Optimizer (GWO) [12], Whale Optimization Algorithm (WOA) [13], Salp Swarm Algorithm (SSA) [14], Moth-Flame Optimization (MFO) [15], Ant Lion Optimizer (ALO) [16], Aquila Optimizer (AO) [17], White Shark Optimizer (WSO) [18], American zebra optimization algorithm (AZOA) [19], and Modified Whale Optimisation Algorithm (MWOA) [20]. Numerous MAs which are dependent on natural phenomenon are: Simulated Annealing (SA) [21], Big Bang-Big Crunch (BBBC) [22], Gravitational Search Algorithm (GSA) [23], Black Hole Algorithm (BHA) [24], Sine Cosine Algorithm (SCA) [25], Multi-verse Optimizer (MVO) [26], Equilibrium Optimizer (EO) [27], Tabu Search (TS) [28], Harmony Search (HS) [29], Teaching Learning-Based Optimization (TLBO) [30], Successful History-based Adaptive DE variants with linear population size reduction (LSHADE) [31], Covariance Matrix Adaptation of Evolution Strategy (CMAES) [32], Biogeography-Based Optimization (BBO) [33], Farmland Fertility Algorithm (FFA) [34], Cosine Swarm Algorithm

✉ Prabhujit Mohapatra
prabhujit.mohapatra@vit.ac.in

Priteesha Sarangi
priteesha.sarangi@vit.ac.in

¹ Department of Mathematics, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India

(CSA) [35], and Young double-slit experiment optimizer (YDSE) [36].

Despite the fact that most of the aforementioned MAs have been extensively deployed to address optimization problems, poor convergence or getting stuck in local optima are still prevalent problems. Thus, the growth of thought-provoking optimization problems has led to the expansion of new optimization algorithms that consistently generate better outcomes and amend existing techniques. The “No Free Lunch” (NFL) [37] theory is applicable in these circumstances. This theorem has demonstrated that no optimization technique can effectively address every optimization challenge. According to this NFL theorem, several scholars have projected many natural-inspired meta-heuristic algorithms. Therefore, in this study, the Mountain Gazelle Optimizer (MGO) [38], which was projected by Benyamin Abdollahzadeh et al. in 2022, was a thought-provoking MA that was an alternative for global optimization. MGO draws its inspiration from the hierarchy and social organization of wild mountain gazelles and is mathematically simulated based on this inspiration. The algorithm is modeled using the fundamental factors, including motherhood herds, bachelor male herds, territorial males, solitary, and foraging movements. The exploitation (intensification) and exploration (diversification) stages of the MGO are performed concurrently using four processes. Despite MGO’s superior performance, simulations show that early iterations may become stuck in less desirable search space domains when interacting with high-dimensional problems, indicating that solutions may not reach diversity, but interesting domains are discovered after a few cycles [39]. Examining the advantages and disadvantages of MGO algorithm and improving their effectiveness by implementing new or modified mechanisms have become fascinating research challenges [40, 41]. Chaos theory is one of the fascinating concepts that is widely used in non-linear dynamic systems due to its ergodicity, unpredictability, and regularity properties. It has been effectively applied in technical optimization procedures, as its initial population diversity helps avoid local optima and early convergence trappings, making it an effective tool in various applications [42]. They have not offered a strong theoretical foundation for improving the effectiveness of meta-heuristic algorithms with chaotic maps. Therefore, numerous chaos maps have been applied to enhance variants of meta-heuristic algorithms by analyzing the literature [43]. Inspired by the benefits of chaotic maps and the shortcomings of MGO, as explained above, this paper proposes a unique chaos-based algorithm termed the Chaotic Mountain Gazelle Optimizer (CMGO). The chaotic maps are incorporated into MGO to enhance its diversification and intensification while preventing premature convergence and local optimum traps. It is thus anticipated that a detailed experimental study will be carried out later in the article to demonstrate the robustness

of the newly designed CMGO approach. The main contribution of the paper is provided as below:

- In this work, the MGO algorithm is combined with ten distinct chaotic maps to introduce the CMGO technique for the first time.
- Chaos theory has enhanced the exploitation ability of the classical MGO and helped it avoid falling into local optimal solutions.
- The effectiveness of 23 standard benchmarks of CEC2005 and 10 complex functions of CEC2019 has been assessed through the CMGO approach.
- To validate that the suggested CMGO is statistically superior, the t-test and Wilcoxon statistical tests have been performed.
- For evaluating the proposed CMGO algorithm’s problem-solving capability, four real engineering design challenges are addressed.

The remaining section of the research is summarized as follows: Sect. 2 describes related works on the modified MAs and hybridization of chaotic maps. Section 3 yields details of the Mountain Gazelle Optimizer and chaotic maps. Section 4 embodies the description of the projected CMGO algorithm. The numerical experiments and result analysis are established in Sect. 5. Section 6 describes the implementation of the CMGO algorithm to solve practical engineering challenges, and Sect. 7 presents the final conclusion and suggests some few ideas for further research.

2 Related Work

An efficient meta-heuristic strikes a good balance between exploration and exploitation to preserve population diversity and increase the algorithm’s reliability and rate of convergence. Hence, several attempts have been tried in past to increase the efficiency of the existing meta-heuristics. The African Vultures Optimization Algorithm (AVOA) has been enhanced with the Quantum Rotation Gate mechanism by enhancing population diversity and optimizing local trap escapes [44]. The proposed ResNet50 Convolutional Neural Network (CNN) model utilizes a hybridization of Particle Swarm Optimization (PSO) to introduce different subtypes [45]. To overcome the shortcomings of WOA, an alternative based on multi-population evolution (MEWOA) has been proposed recently [46]. Similarly, the chaotic functions are mostly used to balance the exploration and exploitation phases in meta-heuristic optimization algorithms. It can increase convergence speed and diversity in meta-heuristic algorithms with chaotic maps. The chaos theory is one of the characteristics of nonlinear systems [47] which defines as the randomness generated by the systems. Many researchers

have added chaos theory to different meta-heuristic optimization algorithms to increase the algorithm's ability to obtain the optimum solution. In this section, some of the hybridization, modification and chaos-based algorithms are discussed. Gupta et al. [48] proposed novel algorithm OCS-GWO that enhances the performance of original GWO by introducing the opposition-based learning to approximate the closer search candidate solution to the global optima and chaotic local search for the exploitation of the search regions efficiently. In OCS-GWO, a chaotic local search is used for balancing the exploration and exploitation operators that are the underlying features of any stochastic search algorithm. Kumar et al. [49] proposed chaotic teaching learning-based algorithm premature convergence and lack of trade-off between local search and global search. Li et al. [50] suggested chaotic arithmetic optimization algorithm (AOA) to improve the exploration and exploitation capabilities of AOA. Firstly, ten chaotic maps are separately embedded into two parameter Arithmetic Optimization Accelerator (MOA) and Arithmetic Optimization Probability, then a combination test was carried out by embedding ten chaotic maps into MOA and MOP at the same time. Kaur et al. [51] introduces chaos theory into WOA to enhance the global convergence speed. The fast random opposition-based learning Golden Jackal Optimization algorithm (FROBL-GJO) was introduced by Mohapatra et al. in 2023 [52], a new technique that enhances the precision and convergence speed of the GJO algorithm. Botnet detection in the Internet of Things is addressed by an innovative binary multi-objective dynamic Harris Hawks Optimization (HHO) improved with mutation operator (MODHHO) proposed by Gharehchopogh et al. [53]. A multi-strategy improved Harris Hawks optimization method has been presented by Gharehchopogh for social network community discovery [54]. Chandran, Vanisree et al. introduces an Enhanced Opposition-Based Learning and is incorporated into Grey Wolf Optimizer to form EOBGWO, a novel technique, to boost the efficiency of the traditional GWO method [55]. An improved version of the Golden Jackal Optimization (GJO) algorithm, incorporates the opposition-based learning (OBL) approach with a probability rate, enabling the algorithm to escape from local optima [56]. The proposed chaotic IAS (Interactive Autodidactic School) model utilizes ten chaotic maps and the intra-cluster summation fitness function to enhance the results of the chaotic IAS algorithm and is introduced by Gharehchopogh et al. [57]. An improved chaotic PSO algorithm, based on adaptive inertia weight (AIWCPSO), enhances population diversity and particle periodicity by appropriately generating the initial population [58]. The Chaotic Marine Predators Algorithm (CMPA) is a novel metaheuristic introduced by Sumit et al. that optimizes engineering problems by combining the exploration capabilities of the Marine Predators Algorithm with the exploitation

capabilities of chaotic maps [59]. New 10 chaotic maps are incorporated in SHO to increase the performance of SHO by producing chaotic values instead of random ones, aiming to improve convergence speed and avoid local optimums [60].

3 Overview of MGO and Chaotic Maps

3.1 Mountain Gazelle Optimizer (MGO)

Mountain Gazelle Optimizer is a novel nature-inspired algorithm proposed in 2022 [38], which is inspired by social structure and hierarchy of wild mountain gazelles. A mathematical representation of the MGO algorithm has been developed using the fundamental ideas underlying the social and group behavior of mountain gazelles. The four primary aspects of the mountain gazelle's life—territorial solitary males, maternity herds, bachelor male herds, and migration in search of food—are employed by the MGO algorithm to execute optimization procedures. Figure 1 illustrates the herd of mountain gazelles, known for their continuous long-distance journey in search of food. Indigenous to the Arabian Peninsula and its surroundings, these gazelles boast a widespread distribution across the region, despite its low population density. Its habitat shares a close connection with the habitat of the Robinia tree species.

3.1.1 Territory Solitary Males

When mountain gazelle males are powerful enough and have reached adulthood, they establish solitary territories and are fiercely protective of themselves, with considerable distances separating the territories. Adult male gazelles engage in conflict over the ownership or territory of the female. The young males attempt to claim either the female or the territory, while mature males work to preserve their habitat. Below is an illustration of the territory of an adult male:

$$TSM = male_{gazelle} - |ri_1 * BH - ri_2 * X(t) * F| * Cof_r \quad (1)$$

$$BH = X_{ra} * \lfloor r_1 \rfloor + M_{pr} * \lfloor r_2 \rfloor, \quad ra = \left\{ \left\lfloor \frac{N}{3} \right\rfloor \dots N \right\} \quad (2)$$

$$F = N_1(D) * \exp\left(2 - it * \left(\frac{2}{Maxit}\right)\right) \quad (3)$$

$$Cof_i = \begin{cases} (a + 1) + r_3, \\ a * N_2(D), \\ r_4(D), \\ N_3(D) * N_4(D)^2 * \cos(2r_4N_3(D)) \end{cases} \quad (4)$$

Fig. 1 Mountain Gazelle herd



$$a = -1 + it * \frac{-1}{Maxit}. \tag{5}$$

Here, the best male adult’s position vector is denoted as $male_{gazelle}$. The coefficient vector of the young male herd is identified as BH . ri_1 and ri_2 are random numbers with values of 1 or 2. X_{ra} signifies as young male in the range of ra , while M_{pr} implies the average number of populations $\left[\frac{N}{3}\right]$ that were arbitrarily nominated. Cof_i symbolizes an arbitrarily generated coefficient vector that is updated after every iteration to increase the effectiveness of the search region. The total number of gazelles is N , while r_1, r_2, r_3 and r_4 designate as arbitrary values in between 0 to 1. The cosine function denoted as \cos , whereas, \exp embodies the exponential function. N_1 symbolizes the arbitrary number drawn from the standard distribution. N_2, N_3 and N_4 characterizes the arbitrary numbers in the normal range and the dimensions of the problem. Finally, $Maxit$ and it expresses the maximum iterations and the current iteration, respectively.

3.1.2 Maternity Herds

Mountain gazelles are a species that depend on maternity herds to produce strong male offspring, which is a crucial part of their life cycle. Male gazelles may additionally be associated with the birth of gazelles and the attempts of young males to seize females. This behavior is articulated as:

$$MH = (BH + Cof_{1,r}) + (ri_3 * male_{gazelle} - ri_4 * X_{rand}) * Cof_{1,r} \tag{6}$$

Here, the arbitrary integers 1 or 2 are represented by ri_3 and ri_4 . A gazelle is arbitrarily chosen from the total gazelles, and X_{rand} displays its vector position.

3.1.3 Bachelor Male Herd

Male gazelles have a tendency to generate territories and seize control of female gazelles as they mature. An abundance of violence may be involved when the young male gazelles engage in this conflict with the adult males for dominance of the female gazelles’ territory. The mathematical model of this behavior is expressed below:

$$BMH = (X(t) - D) + (ri_5 * male_{gazelle} - ri_6 * BH) * Cof_r \tag{7}$$

$$D = (|X(t)| + |male_{gazelle}|) * (2 * r_6 - 1). \tag{8}$$

Here, the random number between 1 and 2 is selected as ri_5 and ri_6 . The random number ranges from 0 to 1 is denoted as r_6 . The vector position of gazelle in the current generation is indicated by the symbol $X(t)$.

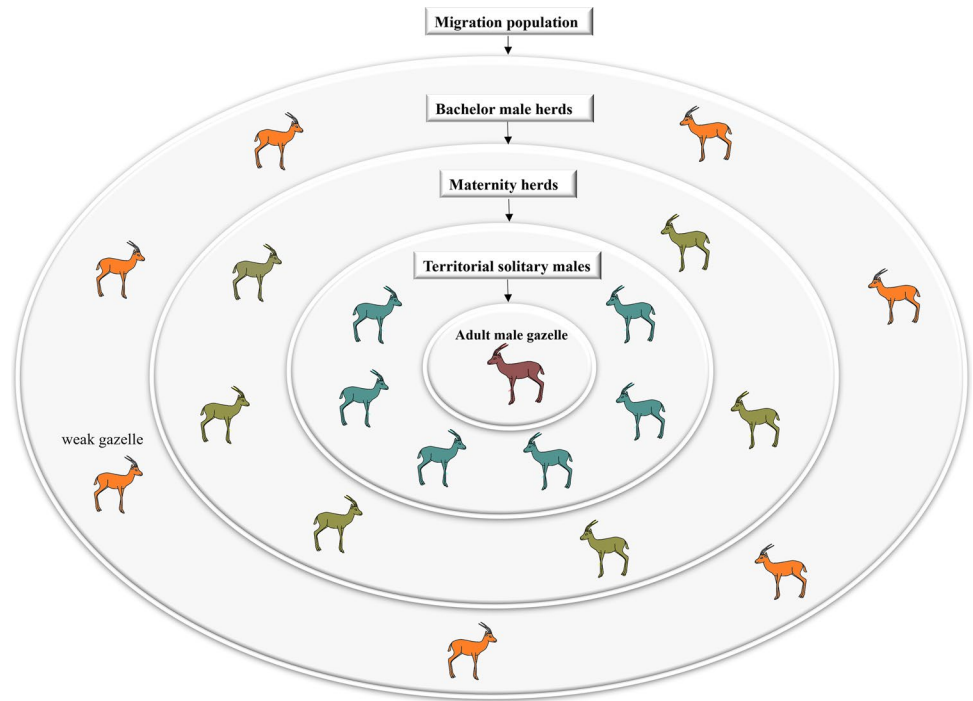
3.1.4 Migration in Search of Food

Mountain gazelles are perpetually on the lookout for sources of food and will travel large distances to seek food and migrate. Mountain gazelles, on the other hand, have a rapid running gait and strong leaping abilities. This behavior of gazelles has been numerically described using the below equation:

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

Here, r_7 indicates the arbitrary number between 0 to 1. lb and ub illustrate the lower and upper bounds, respectively. An overview of the optimization method based on the four MGO’s components is displayed in Fig. 2. The MGO

Fig. 2 An overview of the optimization procedure based on the four components of MGO



algorithm operates on exploiting and exploring phases simultaneously by employing four mechanisms in parallel, according to its inherent nature.

3.2 Chaotic Maps

Chaos is a phenomenon that can exhibit non-linear behavior changes in response to minor changes in its beginning phase. It may be defined mathematically as the arbitrary nature of a fundamental dynamic deterministic framework, and can be regarded as a cause of randomness. It possesses the characteristics of non-repetition and ergodicity; as a result, it will execute searching at a faster rate than random

searches, which rely largely on probability. Chaos exhibits regularity as it is produced by specified functions called the chaotic functions or chaotic maps. These chaotic maps are extensively utilized in numerical analytic techniques, image encryption, and cryptology, as well as modeling complex problems in the domains of medicine, ecology, economics, and engineering [61]. In this study, chaotic sets are produced using 10 discrete non-invertible one-dimensional mappings. Table 1 emphasizes the mathematical model of the well-known chaotic maps. Figure 3 illustrates how the deterministic chaotic map equations contain stochastic variations. According to observations, Cof_i is crucial for maintaining an appropriate balancing in MGO between the intensification

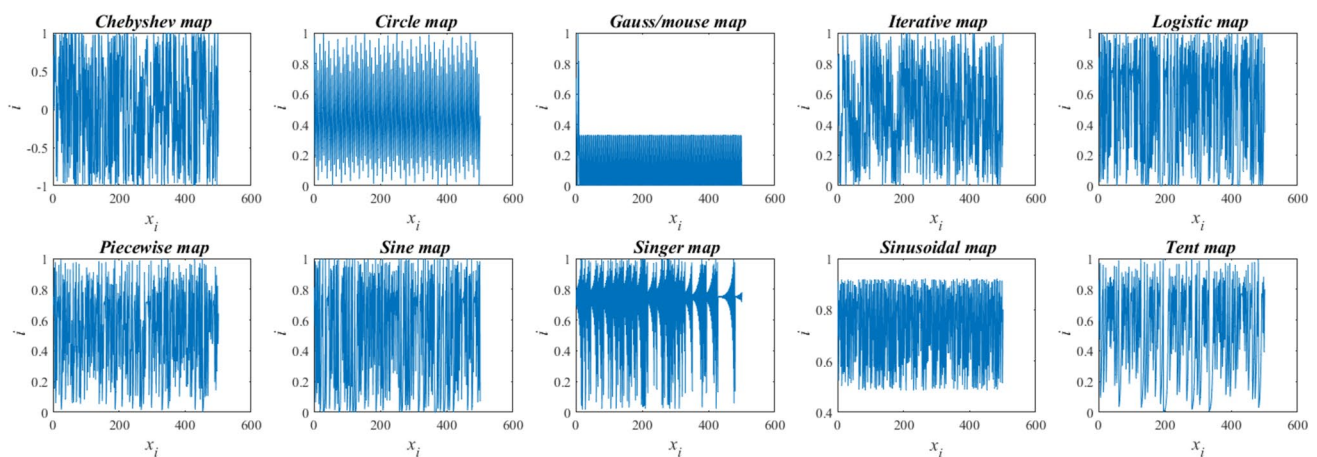


Fig. 3 Stochastic variations of ten different chaotic maps

Table 1 Chaotic maps and their mathematical expression

Sl.no	Name of the maps	Mathematical expression	Limit
1	Chebyshev map	$x_{k+1} = \cos(k\cos^{-1}(x_k))$	(-1, 1)
2	Circle map	$x_{k+1} = \text{mod}(x_k + b - \frac{a}{2\pi} \sin(2\pi x_k), 1), a = 0.5, b = 0.2$	(0, 1)
3	Gauss/mouse map	$x_{k+1} = \begin{cases} 1 & x_k = 0 \\ \frac{1}{\text{mod}(x_k, 1)} & \text{otherwise} \end{cases}$	(0, 1)
4	Iterative map	$x_{k+1} = \sin(\frac{a\pi}{x_k}), a = 0.7$	(-1, 1)
5	Logistic map	$x_{k+1} = ax_k(1 - x_k), a = 4$	(0, 1)
6	Piecewise map	$x_{k+1} = \begin{cases} \frac{x_k}{a} & 0 \leq x_k \leq a \\ \frac{x_k - a}{0.5 - a} & a \leq x_k \leq 0.5 \\ \frac{1 - a - x_k}{0.5 - a} & 0.5 \leq x_k \leq 1 - a \\ \frac{1 - x_k}{a} & 1 - a \leq x_k \leq 1, a = 0.4 \end{cases}$	(0, 1)
7	Sine map	$x_{k+1} = \frac{a}{4} \sin(\pi x_k), a = 4$	(0, 1)
8	Singer map	$x_{k+1} = \mu(7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.302875x_k^4), \mu = 1.07$	(0, 1)
9	Sinusoidal map	$x_{k+1} = ax_k^2 \sin(\pi x_k), a = 2.3$	(0, 1)
10	Tent map	$x_{k+1} = \begin{cases} \frac{x_k}{0.7} & x_k \leq 0.7 \\ \frac{10}{3}(1 - x_k) & x_k \geq 0.7 \end{cases}$	(0, 1)

and diversification phases. MGO’s exploitation and convergence speed will be significantly improved by incorporating chaotic maps.

4 Proposed Chaotic MGO (CMGO)

Chaotic maps have a significant impact on the quality of the solutions generated by optimization algorithms. Despite having a better convergence rate, MGO is still unable to outperform other methods in locating global optima that have an impact on the algorithm’s convergence rate. Therefore, the CMGO algorithm was established by incorporating chaos into the MGO algorithm to improve efficiency in searching for the global optimum by preventing getting entangled in local optima. In the MGO algorithm, four strategies are applied to simultaneously complete the exploration (diversification) and exploitation (intensification) stages. According to observations in MGO, the coefficient vector (Cof_i) is crucial for maintaining an appropriate balance in MGO between the intensification and diversification phases. MGO’s exploitation and convergence speeds will be significantly improved by incorporating chaotic maps. The mathematical expression of the three strategies is given by Eqs. (1), (6), and (7). To enhance the exploration and exploitation phases of the MGO algorithm, the proposed CMGO approach generates the values of these coefficient variables using 10 distinct chaotic maps which is displayed in Fig. 4. Chaos variables are generated through the



Fig. 4 Implementation of 10 different chaotic maps in MGO

utilization of chaotic maps instead of coefficient factors. The variables acquired from chaotic maps are generated in a certain order, while they are unordered with coefficient factor. This explains why chaotic maps ought to be incorporated into optimization techniques to improve their efficacy [49]. Mathematically, Eqs. (1), (6) and (7) in the MGO has been transformed to Eqs. (10), (11) and (12), resulting in the CMGO proposal.

$$TSM = male_{gazelle} - |ri_1 * BH - ri_2 * X(t) * F| * Chaos(t) \quad (10)$$

$$MH = (BH + Chaos(t)) + (ri_3 * male_{gazelle} - ri_4 * X_{rand}) * Chaos(t) \quad (11)$$

$$BMH = (X(t) - D) + (ri_5 * male_{gazelle} - ri_6 * BH) * Chaos(t) \quad (12)$$

Here, $Chas(t)$ is utilized instead of the coefficient vector as the chaotic variable. Ten different chaotic maps are simultaneously used in the formation of CMGO. Algorithm-1 demonstrates the CMGOs pseudo-code. Figure 5 displays the flowchart of the proposed CMGO algorithm.

Algorithm 1 The proposed CMGO algorithm

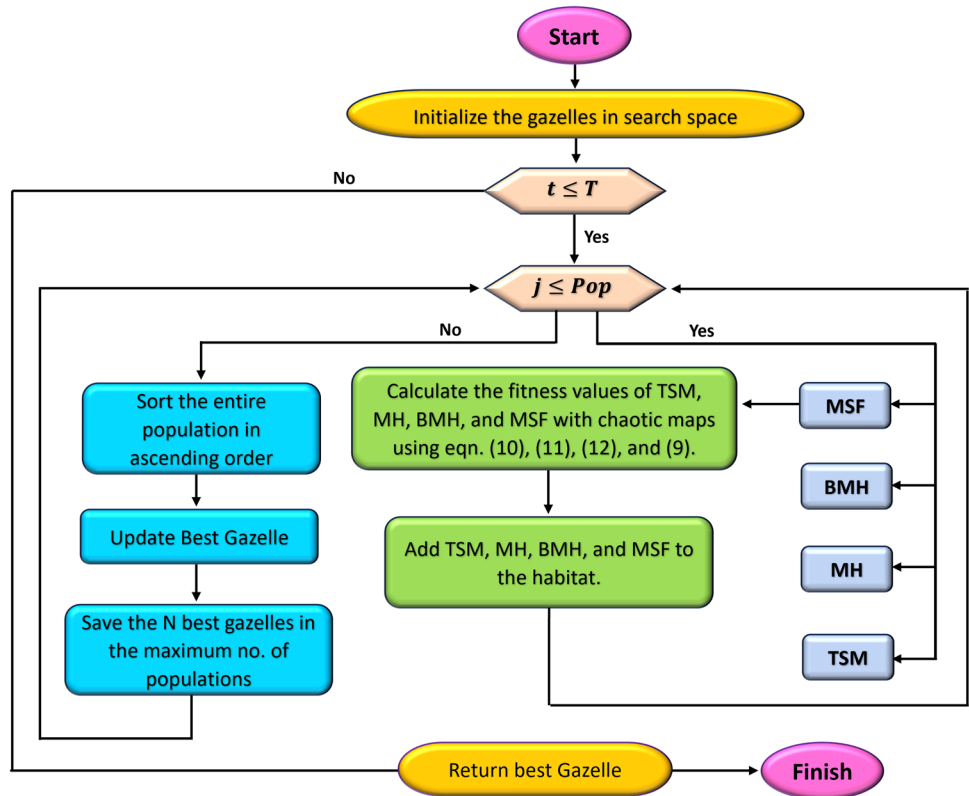
Inputs: The gazelle population N and maximum iterations $Maxit$
Output: Fitness value and positions of each gazelle
 % Initialization
 Initialize the random population using $X_i(i = 1, 2, \dots, N)$
 Evaluate Gazelle's fitness value
while (stopping criteria is not met) **do**
 for each gazelle X_i **do**
 % *Alone male realm*
 Evaluate TSM using eqn.(10)
 % *Mother and child herd*
 Evaluate MH eqn.(11)
 % *Young male herd*
 Evaluate BMH using eqn.(12)
 % *Migration in search of food*
 Evaluate MSF using eqn.(9)
 Evaluate the fitness values of TSM, MH, BMH, and MSF using chaotic maps, then add them to the habitat
 end for
 Sort the entire population in ascending order
 Update $Bestgazelle$
 Save the N best gazelles in the maximum number of population
end while
 Return $X_{Bestgazelle}$

5 Numerical Experiments and Result Analysis

This section stipulates the outcomes of simulations and a comprehensive analysis of the newly introduced CMGO algorithm to address global optimization problems with less time and greater accuracy. The proposed CMGO was assessed over 33 distinct benchmark functions (23 from CEC2005 [62] and 10 from CEC2019 [63]) and applied to four real-world engineering design issues to measure its performance. In this method, several chaotic maps were implemented in MGO individually to generate CMGO1 to CMGO10, which characterize Chebyshev maps, circle maps, Gauss/mouse maps, Iterative maps, logistic maps, piecewise maps, sine maps, singer maps, sinusoidal maps,

and tent maps, and was compared to MGO. Furthermore, the validation process of the CMGO algorithm involved conducting both t -test [64] and Wilcoxon-rank tests [65]. These statistical assessments were performed to validate the robustness and effectiveness of the proposed CMGO algorithm. The results obtained from CMGO across the 33 comparison functions were contrasted with those derived from the original MGO, alongside widely recognized meta-heuristic optimization algorithms like PSO [8], GWO [12], DE [7], LSHADE [31], CMAES [32], FFA [34], and WSO [18] found in the existing literature. For each function, 30 individual runs are conducted, employing 30 search agents and performing 500 iterations. The CMGO algorithm was rigorously tested and executed on an 11th Gen Intel Core CPU running Windows 11 with a

Fig. 5 The flowchart of the proposed CMGO algorithm



processing power of 2.42GHz and 16GB of RAM using MATLAB R2021b.

5.1 Test Functions and Parameter Settings

A test function is often described as an artificial problem that may be employed to estimate how well an algorithm works under various challenging conditions. For the purpose of assessing CMGO performance, a set of 23 benchmark functions from CEC2005 [62] are separated into three groups: The first (F1–F7) unimodal groups which have one global optimum are utilized to determine the exploitation (intensification) of the algorithm. The second (F8–F13) multi-modal groups that have multiple local extrema are employed to evaluate an algorithm’s capacity to prevent local sluggishness in sub-optimal regions. Finally, the third (F14–F23) fixed-dimension multi-modal groups have several optimal points and fewer local minima to assess the algorithm’s exploration (diversification) ability. The derivation of the benchmark functions and their global optimum values are presented in Tables 2, 3 and 4.

The complexity of 10 functions in CEC2019 [63] is significantly higher than standard functions. The algorithm evaluation method also emphasizes the need for generating accurate outcomes. The mean and standard deviation (std) measurements were frequently considered to be the

best options for these benchmark functions. The purpose of standard deviation analysis is to verify that the performance of the algorithms has remained constant across the thirty different runs. Setting algorithm parameters to their default values is a prudent and appropriate method, as illustrated in Table 5.

5.2 Performance of Different Chaotic Maps with MGO.

This subsection describes the approach of employing 10 different alternative chaotic maps separately to enhance and refine the performance of the MGO algorithm. The results of different chaotic maps and MGO algorithm obtained from

Table 2 The details of unimodal functions

Functions	Dim	Range	H_{min}
$H1(Z) = \sum_{i=1}^n Z_i^2$	30	[-100, 100]	0
$H2(Z) = \sum_{i=1}^n Z + \prod_i Z_i $	30	[-10, 10]	0
$H3(Z) = \sum_{i=1}^n (\sum_{j=1}^i Z_j)^2$	30	[-100, 100]	0
$H4(Z) = \max_i \{ Z_i , 1 \leq i \leq n\}$	30	[-100, 100]	0
$H5(Z) = \sum_{i=1}^{n-1} [100(Z_{i+1} - Z_i^2)^2 + (Z_i - 1)^2]$	30	[-30, 30]	0
$H6(Z) = \sum_{i=1}^n ((Z_i + 0.5)^2)$	30	[-100, 100]	0
$H7(Z) = \sum_{i=1}^n iZ_i^4 + random[0, 1)$	30	[-1.28, 1.28]	0

Table 3 The details of multi-modal functions

Functions	Dim	Range	H_{min}
$H8(z) = \sum_{i=1}^n -z_i \sin(\sqrt{ z_i })$	30	[-500, 500]	-418.982*5
$H9(Z) = \sum_{i=1}^n [Z_i^2 - 10 \cos(2\pi Z_i) + 10]$	30	[-5.12, 5.12]	0
$H10(Z) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n Z_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi Z_i)) + 20 + e$	30	[-32, 32]	0
$H11(Z) = \frac{1}{4000} \sum_{i=1}^n Z_i^2 - \prod_{i=1}^n \cos(\frac{z_i}{\sqrt{i}}) + 1$	30	[-600, 600]	0
$H12(Z) = \frac{\pi}{n} \left\{ 10 \sin \pi y_1 + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(z_i, 10, 100, 4)$ $y_i = 1 + \frac{Z_i + 1}{4}$ $u(Z_i, a, k, m) = \begin{cases} k(Z_i - a)^m & Z_i > a \\ 0 & -a < z_i < a \\ k(-Z_i - a)^m & Z_i < -a \end{cases}$	30	[-50, 50]	0
$H13(Z) = 0.1 \{ (3\pi Z_1) + \sum_{i=1}^n (Z_i - 1)^2 [1 + \sin^2(3\pi Z_i + 1)] + (Z_n - 1)^2 [1 + \sin^2(2\pi Z_n)] \} + \sum_{i=1}^n u(Z_i, 5, 100, 4)$	30	[-50, 50]	0

Table 4 The details of fixed-dimension multi-modal functions

Functions	Dim	Range	H_{min}
$H14(Z) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (Z_i - a_{ij})^6})^{-1}$	2	[-65, 65]	1
$H15(Z) = \sum_{i=1}^{11} [a_i - \frac{Z_i(b_i^2 + b_i Z_i)}{b_i^2 + b_i Z_i + z_i}]^2$	4	[- 5, 5]	0.0003
$H16(Z) = Z_1^2 - 2.1Z_1^4 + \frac{1}{3}Z_1^6 + Z_1Z_2 - 4Z_2^2 + 4Z_2^4$	2	[- 5, 5]	-1.0316
$H17(Z) = (Z_2 - \frac{5.1}{4\pi^2}Z_1^2 + \frac{5}{\pi}Z_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos Z_1 + 10$	2	[- 5, 5]	0.398
$H18(Z) = [1 + (Z_1 + Z_2 + 1)^2(19 - 14Z_1 + 3Z_1^2 - 14Z_2 + 6Z_1Z_2 + 3Z_2^2)] [30 + (2Z_1 - 3Z_2)^2(18 - 32Z_1 + 12Z_1^2 + 48Z_2 - 36Z_1Z_2 + 27Z_2^2)]$	2	[- 2, 2]	3
$H19(Z) = - \sum_{i=1}^4 c_i \exp(- \sum_{j=1}^3 a_{ij}(Z_j - p_{ij})^2)$	3	[1, 3]	-3.86
$H20(Z) = - \sum_{i=1}^4 c_i \exp(- \sum_{j=1}^6 a_{ij}(Z_j - p_{ij})^2)$	6	[0, 1]	-3.32
$H21(Z) = - \sum_{i=1}^5 [(Z - a_i)(Z - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.1532
$H22(Z) = - \sum_{i=1}^7 [(Z - a_i)(Z - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.4028
$H23(Z) = - \sum_{i=1}^{10} [(Z - a_i)(Z - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.5363

23 functions of CEC2005 and 10 functions of CEC2019 with averages (Mean) and standard deviations (Std) for each test function are separately presented in Tables 6, 7, respectively. The best mean values have been highlighted in boldfaced in Table 6, 7. As observed in Table 6, the optimization efficiency of the method CMGO for the unimodal functions F1–F7 is considerably increased, indicating that CMGO has excellent exploitation capability and can discover the best global solution (F1–F5 and F7). It has been observed that the best outcomes achieved with CMGO2–CMGO10 for F1, CMGO3–CMGO10 for F2, CMGO1 for F3, CMGO8–CMGO9 for F4, CMGO6 and CMGO9 for F5, and CMGO10 for F7. In the group of multi-modal functions, F8, F9 and F11 shows the similar performance for MGO and all other chaotic MGO. CMGO2–CMGO10 for F10 and CMGO6 for F13 outperforms better than MGO. This demonstrates that

the newly proposed CMGO has excellent exploration and flexibility when dealing with multimodal functions. In the group of fixed-dimension multi-modal functions, F14 and F16 achieve the same performance in MGO and all other chaotic MGOs. Except for F15 and F18, CMGO1 outperforms better results than MGO. The three enhancement techniques increase the algorithm’s optimization performance and convergence speediness. Moreover, the optimization accuracy of the CMGO method for the CEC2019 functions has been enhanced to different degrees when compared to the original MGO algorithm. On the basis of the results observed in Table 7, CMGO1 for cec02, cec03 and cec06 along with CMGO2 for cec10 achieved the best outcomes when compared to MGO. Therefore, from the aforementioned experiment research, the projected chaotic MGO (CMGO) outperform better in maximum functions and is

Table 5 Parameter settings of comparison algorithms

Sl.no	Algorithm	Parameter	Value
1	PSO	Inertia factor	0.3
		c_1	2
		c_2	2
2	GWO	a (converging constant)	[2 0]
		J_r (jumping probability)	0.1
3	DE	P_{cr} (crossover probability)	0.2
		β_{\min}	0.2
		β_{\max}	0.8
4	LSHADE	Memory size	5
		arc_{rate}	1.4
		$pbest_{rate}$	0.11
5	CMAES	Free parameter	0
		arc_{rate}	1.4
6	FFA	K value	2
		α	0.6
		β	0.4
		W	1
		Q	0.7
		f_{\min}	0.07
7	WSO	f_{\max}	0.75
		a_0	6.25
		a_1	100
		a_2	0.0005
		Free parameter	–
8	MGO	Free parameter	–
9	CMGO	Initial point	0.7

appropriate for evaluating exploration (diversification) and exploitation (intensification) capability.

5.3 Performance of CMGO with Other State-of-the-Art Algorithms

This study compares the performance and overall efficiency of the proposed CMGO technique against several top-performing algorithms, including PSO, GWO, DE, LSHADE, CMAES, FFA, WSO, and MGO. The outcomes presented in Table 8 demonstrate that the newly designed CMGO algorithm outperformed the comparative algorithms for the majority of the first group of unimodal functions (F1–F5 and F7). Therefore, the CMGO approach is best suited to assess its exploitation potential. The suggested CMGO technique may produce fascinating results on the second group of multi-modal functions for F9, F11, and F13, whereas WSO and MGO offer superior results for F8 and F12, respectively. It reveals that for addressing multimodal functions, the projected CMGO has outstanding exploration and adaptability. On the group of fixed-dimension multi-modal functions, the designed CMGO algorithm achieved superior results for F14, F21, F22, and F23, while achieving comparable

results for F16 and F17. The experimental investigation serves as an illustration of how the CMGO algorithm appropriately blends exploitation and exploration. However, Table 9 divulges that the proposed CMGO algorithm outperforms better in fewer functions like cec02, cec03, and cec10, whereas LSHADE performs better than the CMGO algorithm. In general, the suggested CMGO delivers complimentary benefits to improve the global search capacity, enabling it to identify an accurate solution.

5.4 Sensitivity Analysis

The sensitivity analysis of the CMGO algorithm is examined to assess the quality of the proposed CMGO. Here, the impact of varying the parameters of the CMGO on how it operates is investigated. Generally speaking, an initial point is the only major parameter in CMGO; hence, several scenarios are constructed based on the values of these factors. These factors are assessed at one of the following values: 0.1, 0.3, or 0.7. The test functions that have been selected for the experiment are F1, F4, and F7. The graphical and statistical results for each of the conditions are presented in Fig. 6 and Table 10, respectively. The outcomes demonstrate that the CMGO approach yields excellent results when the initial point (IP) is set at 0.7.

5.5 Statistical Analysis

The purpose of the statistical tests is to identify significant differences between the suggested CMGO approach and other well-established methods. In this work, the benchmark functions applied to all other algorithms are statistically analyzed employing the commonly-used t -test [64] and the non-parametric Wilcoxon rank-sum test [65]. The t -values for each function in the t -test may be determined by considering both approaches simultaneously. The statistical t -values are presented in Tables 8, 9. The last row in Tables 8, 9, gives the win, tie, and loss total for CMGO, which contributes in producing more efficient outcomes than other various meta-heuristics. Additionally, the outcomes of the two-tailed Wilcoxon rank-sum test are presented in Tables 11, 12. Wilcoxon rank-sum test is a non-parametric statistical test that analyses the importance of two sets of parameters depending on the differences between them. It is performed at a 5% significance level. Throughout the experiment, when p -value < 0.05 , W-value is determined as ‘+’, which demonstrates that there is a statistically significant difference between the two groups’ data, otherwise W-value is determined as ‘–’ which shows that there is no significant difference. NA stands for “not applicable” that symbolizes ‘=’ which demonstrate that the proposed CMGO algorithm have equivalent difference with other meta-heuristics. From the Table 11, it is

Table 6 Results on unimodal, multi-modal and fixed-dimension multi-modal functions with different chaotic maps

Functions	MGO	CMGO1	CMGO2	CMGO3	CMGO4	CMGO5	CMGO6	CMGO7	CMGO8	CMGO9	CMGO10
<i>F1</i>	Mean 4.478E-70	1.966E-150	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Std 2.449E-69	9.181E-150	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
<i>F2</i>	Mean 1.060E-41	4.408E-81	8.425E-230	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Std 4.697E-41	2.128E-80	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
<i>F3</i>	Mean 1.226E-07	1.670E-64	2.623E-38	1.516E-04	1.734E-34	1.341E-24	3.106E-23	9.568E-21	3.639E-24	2.566E-18	7.512E-13
	Std 5.140E-07	9.147E-64	1.436E-37	6.661E-04	9.497E-34	7.222E-24	1.694E-22	5.241E-20	1.881E-23	1.397E-17	4.097E-12
<i>F4</i>	Mean 4.215E-24	1.363E-69	4.533E-177	6.633E-52	9.270E-248	4.638E-270	2.231E-279	4.308E-252	0.000E+00	0.000E+00	5.939E-299
	Std 2.053E-23	4.584E-69	0.000E+00	2.582E-51	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
<i>F5</i>	Mean 2.549E-20	2.888E-31	1.153E-29	6.232E-29	1.277E-30	3.337E-30	0.000E+00	1.688E-30	2.060E-30	0.000E+00	1.360E-30
	Std 1.371E-19	1.201E-30	1.670E-29	8.232E-29	5.072E-30	9.851E-30	0.000E+00	7.272E-30	9.232E-30	0.000E+00	5.376E-30
<i>F6</i>	Mean 2.713E-08	4.547E-05	4.125E-05	1.702E-05	4.307E-05	4.941E-05	2.066E-05	6.356E-08	1.537E-07	1.273E-05	1.226E-06
	Std 9.843E-08	1.167E-04	1.341E-04	5.086E-05	1.340E-04	2.535E-04	8.335E-05	2.146E-07	5.027E-07	6.794E-05	6.332E-06
<i>F7</i>	Mean 4.005E-04	1.865E-04	5.384E-05	4.415E-05	4.881E-05	5.291E-05	5.042E-05	4.547E-05	4.859E-05	4.595E-05	4.119E-05
	Std 3.084E-04	1.478E-04	5.327E-05	6.009E-05	4.302E-05	5.746E-05	4.859E-05	3.286E-05	7.194E-05	5.041E-05	3.665E-05
<i>F8</i>	Mean -1.257E+04	-1.257E+04	-1.257E+04	-1.257E+04	-1.257E+04	-1.257E+04	-1.257E+04	-1.257E+04	-1.257E+04	-1.257E+04	-1.257E+04
	Std 2.473E-08	3.338E-04	6.120E-03	9.655E-04	8.534E-04	1.105E-03	2.408E-04	6.009E-03	2.011E-03	2.918E-03	8.220E-04
<i>F9</i>	Mean 0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Std 1.000E-16	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
<i>F10</i>	Mean 1.243E-15	1.125E-15	8.882E-16	8.882E-16	8.882E-16	8.882E-16	8.882E-16	8.882E-16	8.882E-16	8.882E-16	8.882E-16
	Std 1.084E-15	9.014E-16	1.003E-31	1.003E-31	1.003E-31	1.003E-31	1.003E-31	1.003E-31	1.003E-31	1.003E-31	1.003E-31
<i>F11</i>	Mean 0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Std 1.000E-16	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
<i>F12</i>	Mean 2.561E-26	3.166E-06	1.458E-07	1.092E-07	9.432E-08	2.351E-08	2.535E-07	4.647E-07	2.044E-08	1.596E-08	6.518E-08
	Std 7.754E-26	8.383E-06	4.998E-07	3.039E-07	2.676E-07	658E-08	1.357E-06	2.507E-06	7.828E-08	4.906E-08	1.865E-07
<i>F13</i>	Mean 1.600E-32	1.407E-32	1.568E-32	9.295E-32	1.518E-32	1.522E-32	1.350E-32	1.605E-32	1.444E-32	1.411E-32	1.383E-32
	Std 4.113E-33	1.794E-33	5.135E-33	1.707E-31	3.877E-33	4.253E-33	5.567E-48	6.641E-33	3.390E-33	3.376E-33	1.371E-33
<i>F14</i>	Mean 9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01
	Std 3.388E-16	1.414E-11	1.699E-11	8.998E-11	4.835E-11	3.209E-11	3.900E-11	1.814E-11	8.221E-11	2.249E-11	3.168E-11
<i>F15</i>	Mean 3.103E-04	3.513E-04	3.929E-04	3.580E-04	3.549E-04	3.339E-04	3.846E-04	3.541E-04	3.775E-04	5.574E-04	4.229E-04
	Std 1.094E-05	4.859E-05	1.318E-04	7.900E-05	8.266E-05	3.278E-05	9.388E-05	1.002E-04	1.597E-04	1.668E-04	1.028E-04
<i>F16</i>	Mean -1.032E+00	- 1.032E+00	- 1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.031E+00	-1.024E+00	-1.031E+00
	Std 1.000E-16	6.604E-08	4.136E-05	7.194E-05	8.996E-06	2.081E-05	2.120E-04	9.664E-06	1.679E-04	1.909E-02	8.232E-04
<i>F17</i>	Mean 3.979E-01	3.979E-01	4.024E-01	4.228E-01	4.292E-01	4.125E-01	4.392E-01	4.152E-01	4.660E-01	6.783E-01	5.078E-01
	Std 1.129E-16	4.613E-05	9.388E-03	4.588E-02	6.108E-02	3.604E-02	7.153E-02	4.874E-02	1.332E-01	5.054E-01	1.917E-01
<i>F18</i>	Mean 3.000E+00	6.601E+00	9.510E+00	1.135E+01	1.155E+01	1.439E+01	8.732E+00	1.263E+01	1.156E+01	5.357E+00	1.061E+01
	Std 4.723E-15	9.338E+00	1.201E+01	1.296E+01	1.330E+01	1.419E+01	1.166E+01	1.386E+01	1.330E+01	5.787E+00	1.269E+01

Table 6 (continued)

Functions	MGO	CMGO1	CMGO2	CMGO3	CMGO4	CMGO5	CMGO6	CMGO7	CMGO8	CMGO9	CMGO10
F19	Mean	-3.863E+00	-3.863E+00	-3.854E+00	-3.799E+00	-3.833E+00	-3.802E+00	-3.840E+00	-3.809E+00	-3.673E+00	-3.786E+00
	Std	2.710E-15	1.914E-04	5.632E-03	6.297E-03	1.957E-01	1.412E-01	2.648E-02	1.434E-01	2.416E-01	1.437E-01
F20	Mean	-3.251E+00	-3.253E+00	-3.146E+00	-3.102E+00	-3.105E+00	-3.074E+00	-3.154E+00	-2.980E+00	-2.404E+00	-2.636E+00
	Std	5.924E-02	9.476E-02	1.385E-01	1.402E-01	1.292E-01	1.363E-01	9.607E-02	1.519E-01	5.116E-01	2.840E-01
F21	Mean	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01
	Std	1.807E-15	6.608E-06	1.538E-05	5.333E-05	9.294E-06	3.592E-05	2.614E-05	3.567E-05	8.566E-06	2.576E-05
F22	Mean	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01
	Std	1.000E-16	8.351E-06	5.771E-05	8.089E-05	4.930E-05	4.756E-05	3.712E-05	2.901E-05	4.610E-05	2.500E-05
F23	Mean	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01
	Std	9.034E-15	4.410E-06	7.923E-05	2.416E-04	5.716E-05	3.061E-05	1.765E-05	2.462E-05	5.391E-05	5.922E-05

observed that the proposed CMGO algorithm has significant difference for the functions F1–F8, F17, F19 and F22 compared with all other algorithms. However, compared with MGO, CMGO has no significant results for the functions F10 and F20. Compared with LSHADE, FFA, WSO and MGO, CMGO illustrates equivalent outcomes for the functions F14 and F9. Therefore, when compared to prominent algorithms, the suggested CMGO demonstrates outstanding results for the majority of the functions. As observed in Table 12, it is perceived that the proposed CMGO has significant difference for the functions cec01, cec05 and cec06 when compared with other meta-heuristics. Compared with GWO, CMAES, and FFA, the CMGO has significant results in the maximum number of functions except for cec07 and cec08. In general, CMGO proved to be the best optimizer based on the aforementioned statistical analysis, outperforming the other recent and top-performing algorithms on the majority of the problems with the environment.

5.6 Convergence Analysis

This subsection illustrates the graphical investigation for a more comprehensive evaluation of the performance of all the algorithms. Line graphs exhibiting the convergence of several benchmark functions using the CMGO method and other algorithms have been displayed in Figs. 7, 8, which make it simple to analyze the rate of convergence of each algorithm over the course of numerous iterations. These figures exhibit the convergence curves of the CMGO for several unimodal, multi-modal, and fixed-dimensional multi-modal functions, as well as CEC2019 benchmark functions. As can be observed from the convergence curves in Fig. 7, over the period of each iteration, CMGO tends to accelerate its convergence. In other words, searching individuals can identify desirable areas in the early iterations and accelerate convergence. Functions F1–F5 and F7 each exhibit this behavior. Consequently, while solving the multi-modal functions, CMGO might reveal an outstanding convergence rate. Based on these curves, CMGO sustains a consistent rate of convergence in the final stage of the iterations. This is because the explorers are covering all of the search space, and functions F9, F11 and F13 exhibit this apparent behavior. However, CMGO performs equivalent results than the compared algorithms in context of convergence rate in most of the functions like F17, F21–F23. On the basis of convergence curve in Fig. 8, it is obvious that the CMGO algorithm has better convergence accuracy than competing algorithms at certain functions, such as cec02, cec03, cec06, and cec10. These evaluations demonstrate that the search region is initially explored by the CMGO gazelles to evaluate multiple potential destinations. Then, the search area is exploited by the CMGO gazelles to seek out the global optimal solution while compelling them to change locally rather than

Table 7 Results on CEC2019 benchmark functions with different chaotic maps

Functions	MGO	CMGO1	CMGO2	CMGO3	CMGO4	CMGO5	CMGO6	CMGO7	CMGO8	CMGO9	CMGO10
cec01	Mean	5.022E+04	1.599E+05	8.754E+04	1.958E+05	1.280E+05	1.807E+05	1.240E+05	3.231E+05	4.309E+05	4.183E+05
	Std	8.165E+03	1.877E+05	1.082E+05	2.821E+05	7.028E+04	1.635E+05	2.112E+05	1.456E+05	1.281E+05	1.864E+05
cec02	Mean	1.734E+01	1.734E+01	1.765E+01	1.750E+01	1.779E+01	1.783E+01	1.786E+01	1.789E+01	1.804E+01	1.792E+01
	Std	1.000E-15	4.301E-13	3.139E-01	1.127E-01	3.336E-01	2.711E-01	2.568E-01	2.592E-01	3.304E-01	3.893E-01
cec03	Mean	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01
	Std	2.541E-09	9.816E-13	2.282E-05	3.232E-05	5.098E-05	1.166E-04	7.121E-05	1.557E-04	6.295E-05	1.053E-04
cec04	Mean	3.403E+01	1.013E+02	3.371E+03	5.695E+03	6.307E+03	6.352E+03	6.061E+03	6.900E+03	7.188E+03	6.860E+03
	Std	1.318E+01	5.313E+01	1.904E+03	1.988E+03	2.617E+03	1.647E+03	2.037E+03	2.430E+03	2.587E+03	2.633E+03
cec05	Mean	1.135E+00	1.411E+00	3.419E+00	3.442E+00	3.559E+00	3.324E+00	3.608E+00	3.804E+00	3.651E+00	3.946E+00
	Std	9.599E-02	3.668E-01	6.545E-01	7.357E-01	7.178E-01	9.073E-01	7.892E-01	8.155E-01	7.715E-01	6.086E-01
cec06	Mean	1.023E+01	8.896E+00	1.078E+01	1.094E+01	1.107E+01	1.099E+01	1.101E+01	1.098E+01	1.107E+01	1.109E+01
	Std	6.304E-01	1.230E+00	6.235E-01	6.891E-01	4.014E-01	7.606E-01	3.900E-01	5.137E-01	7.282E-01	4.116E-01
cec07	Mean	2.246E+02	5.067E+02	8.570E+02	8.542E+02	9.358E+02	1.115E+03	8.992E+02	1.045E+03	1.147E+03	1.056E+03
	Std	1.420E+02	3.146E+02	2.108E+02	2.964E+02	2.178E+02	2.903E+02	2.114E+02	3.162E+02	2.585E+02	2.490E+02
cec08	Mean	4.840E+00	5.811E+00	6.191E+00	6.336E+00	6.541E+00	6.446E+00	6.576E+00	6.743E+00	6.723E+00	6.765E+00
	Std	9.987E-01	9.812E-01	5.232E-01	6.058E-01	4.143E-01	4.275E-01	5.969E-01	4.825E-01	5.234E-01	4.686E-01
cec09	Mean	2.425E+00	3.564E+00	2.281E+02	4.171E+02	6.640E+02	9.272E+02	8.927E+02	1.277E+03	1.261E+03	1.115E+03
	Std	4.617E-02	3.765E-01	1.883E+02	3.687E+02	5.781E+02	5.726E+02	5.537E+02	5.815E+02	7.091E+02	7.154E+02
cec10	Mean	2.033E+01	2.011E+01	1.873E+01	2.050E+01	2.047E+01	2.053E+01	2.044E+01	2.041E+01	2.042E+01	1.980E+01
	Std	1.329E-01	1.295E-01	4.180E+00	1.114E-01	1.100E-01	9.463E-02	1.261E-01	6.797E-02	6.656E-02	9.264E-02

Table 8 Results on unimodal, multi-modal and fixed-dimension multi-modal functions with other state-of-art algorithms

Functions	PSO	GWO	DE	LSHADE	CMAES	FFA	WSO	MGO	CMGO
<i>F1</i>	Mean	1.430E-27	3.685E-04	1.082E-01	1.133E-05	1.687E-05	2.301E+02	4.478E-70	1.172E-130
	Std	1.788E-27	1.212E-04	1.498E-01	4.601E-06	1.897E-05	8.317E+01	2.449E-69	6.418E-130
	<i>t-values</i>	- 2.649E+00	- 1.665E+01	- 3.956E+00	- 1.349E+01	- 4.871E+00	- 1.516E+01	- 1.002E+00	
<i>F2</i>	Mean	5.044E-02	8.810E-17	2.148E-03	5.179E-03	1.327E-04	4.218E+00	1.060E-41	1.478E-71
	Std	6.232E-02	7.020E-17	4.199E-04	1.602E-01	1.249E-04	1.388E+00	4.697E-41	8.081E-71
	<i>t-values</i>	- 4.433E+00	- 6.874E+00	- 2.802E+01	- 4.999E+00	- 5.819E+00	- 1.664E+01	- 1.236E+00	
<i>F3</i>	Mean	8.171E+01	8.330E-06	3.183E+04	4.366E-01	1.428E+04	1.346E+03	1.226E-07	1.643E-61
	Std	5.865E+01	1.730E-05	5.334E+03	2.200E-01	3.896E+03	4.787E+02	5.140E-07	6.944E-61
	<i>t-values</i>	- 7.631E+00	- 2.637E+00	- 3.269E+01	- 8.727E+00	- 2.008E+01	- 1.540E+01	- 1.307E+00	
<i>F4</i>	Mean	2.150E+00	7.620E-07	1.356E+01	2.332E-02	2.966E+01	1.346E+03	4.215E-24	3.992E-64
	Std	8.272E-01	5.860E-07	1.812E+00	3.042E+00	4.660E+00	4.787E+02	2.053E-23	1.406E-63
	<i>t-values</i>	- 1.424E+01	- 7.122E+00	- 4.099E+01	- 2.608E+01	- 3.486E+01	- 1.540E+01	- 1.124E+00	
<i>F5</i>	Mean	4.966E+01	2.724E+01	1.587E+02	4.365E+01	8.242E+01	2.088E+04	2.549E-20	0.000E+00
	Std	3.762E+01	8.837E-01	4.690E+01	1.156E+02	4.349E+01	2.591E+04	1.371E-19	0.000E+00
	<i>t-values</i>	- 7.231E+00	- 1.688E+02	- 1.853E+01	- 7.408E+00	- 5.497E+00	- 4.413E+00	- 1.019E+00	
<i>F6</i>	Mean	2.458E-05	7.877E-01	4.416E-04	5.139E-02	1.083E-05	2.403E+02	2.713E-08	3.597E-05
	Std	1.191E-04	3.531E-01	2.055E-04	4.509E-02	4.006E-06	1.646E+02	9.843E-08	4.886E-05
	<i>t-values</i>	4.847E-01	- 1.222E+01	- 1.051E+01	- 6.238E+00	2.809E+00	- 7.999E+00	4.030E+00	
<i>F7</i>	Mean	3.008E-02	2.353E-03	6.087E-02	8.420E-02	4.597E-02	1.486E-01	4.005E-04	1.626E-04
	Std	1.849E-02	1.298E-03	1.400E-02	3.955E-02	6.499E-03	6.374E-02	3.084E-04	1.051E-04
	<i>t-values</i>	- 8.862E+00	- 9.217E+00	- 2.375E+01	- 1.164E+01	- 2.029E+01	- 1.275E+01	- 4.000E+00	
<i>F8</i>	Mean	- 6.530E+03	- 6.076E+03	- 9.936E+03	- 1.212E+04	- 1.177E+05	- 5.033E+03	- 1.257E+04	- 1.257E+04
	Std	7.432E+02	6.583E+02	5.444E+02	1.964E+02	6.558E+04	1.488E+03	2.473E-08	1.144E-03
	<i>t-values</i>	- 4.451E+01	- 5.403E+01	- 2.650E+01	- 1.262E+01	8.781E+00	- 3.869E+01	- 2.774E+01	
<i>F9</i>	Mean	5.018E+01	4.852E+00	8.652E+01	5.468E+00	1.311E+02	4.637E+01	0.000E+00	0.000E+00
	Std	1.316E+01	4.724E+00	9.219E+00	2.570E+00	7.085E+01	1.858E+01	1.000E-16	0.000E+00
	<i>t-values</i>	- 2.088E+01	- 5.625E+00	- 5.141E+01	- 1.166E+01	- 1.013E+01	- 2.113E+01	0.000E+00	
<i>F10</i>	Mean	1.654E+00	1.079E-13	5.907E-03	2.917E+00	1.036E-03	6.474E+00	1.243E-15	1.362E-15
	Std	7.009E-01	1.902E-14	1.458E-03	8.370E-01	2.464E-04	7.691E-01	1.084E-15	1.228E-15
	<i>t-values</i>	- 1.292E+01	- 3.062E+01	- 2.219E+01	- 1.909E+01	- 2.304E+01	- 4.610E+01	3.959E-01	
<i>F11</i>	Mean	3.717E-02	6.564E-03	6.502E-03	1.518E-01	3.219E-02	3.434E+00	0.000E+00	0.000E+00
	Std	3.609E-02	1.178E-02	8.967E-03	1.243E-01	4.556E-05	1.409E+00	1.000E-16	0.000E+00
	<i>t-values</i>	- 5.641E+00	- 3.052E+00	- 3.972E+00	- 6.692E+00	- 1.636E+01	- 4.167E+00	0.000E+00	
<i>F12</i>	Mean	7.604E-02	6.961E-02	5.823E-05	1.622E+00	6.892E-01	3.952E+00	2.561E-26	3.001E-06
	Std	1.387E-01	1.040E-01	4.100E-05	1.478E+00	6.060E-07	1.966E+00	7.754E-26	7.479E-06
	<i>t-values</i>	- 3.002E+00	- 3.667E+00	- 7.259E+00	- 6.010E+00	1.205E+00	- 8.089E+00	- 1.101E+01	2.197E+00

Table 8 (continued)

Functions	PSO	GWO	DE	LSHADE	CMAES	FFA	WSO	MGO	CMGO
F13	Mean	6.157E-01	2.691E-04	3.016E+00	1.507E-05	2.744E-01	2.520E+02	1.600E-32	1.350E-32
	Std	7.185E-01	2.876E-01	1.339E-04	3.733E+00	1.956E-01	5.626E+02	4.113E-33	5.567E-48
F14	<i>t-values</i>	- 1.609E+00	- 1.173E+01	- 1.100E+01	- 4.424E+00	- 7.685E+00	- 2.453E+00	- 3.337E+00	
	Mean	3.692E+00	4.459E+00	1.360E+00	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01
F15	Std	2.623E+00	4.109E+00	1.255E+00	3.388E-16	3.388E-16	5.152E-12	3.388E-16	6.152E-11
	<i>t-values</i>	- 5.625E+00	- 4.614E+00	- 1.579E+00	1.662E+00	1.662E+00	1.573E+00	1.662E+00	
F16	Mean	1.099E-03	8.491E-03	7.344E-04	4.419E-04	5.524E-04	3.380E-04	3.103E-04	3.548E-04
	Std	3.651E-03	1.125E-02	1.113E-04	3.537E-04	1.340E-03	1.672E-04	1.094E-05	5.827E-05
F17	<i>t-values</i>	- 1.117E+00	- 3.963E+00	- 1.655E+01	- 1.332E+00	- 8.155E+00	5.182E-01	4.103E+00	
	Mean	- 1.032E+00	- 1.032E+00	- 1.032E+00	- 1.032E+00	- 1.032E+00	- 1.032E+00	- 1.032E+00	- 1.032E+00
F18	Std	1.000E-11	1.835E-08	1.000E-13	1.000E-14	1.000E-14	4.498E-06	1.000E-16	1.026E-06
	<i>t-values</i>	3.418E+00	3.321E+00	3.418E+00	3.418E+00	3.418E+00	- 1.393E+00	3.418E+00	
F19	Mean	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.981E-01
	Std	1.129E-16	1.221E-04	1.129E-16	1.129E-16	1.129E-16	1.129E-16	1.129E-16	4.237E-04
F20	<i>t-values</i>	3.023E+00	2.535E+00	3.023E+00	3.023E+00	3.023E+00	3.023E+00	3.023E+00	
	Mean	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00	4.800E+00
F21	Std	3.030E-15	5.235E-05	4.517E-16	2.566E-15	5.719E-15	4.025E-15	4.723E-15	6.851E+00
	<i>t-values</i>	1.439E+00	1.439E+00	1.439E+00	1.439E+00	1.439E+00	1.439E+00	1.439E+00	
F22	Mean	- 3.005E-01	- 3.862E+00	- 3.005E-01	- 3.005E-01	- 3.005E-01	- 3.005E-01	- 3.005E-01	- 3.835E+00
	Std	1.000E-11	1.997E-03	1.000E-13	1.000E-14	2.710E-15	1.000E-13	1.000E-14	1.408E-01
F23	<i>t-values</i>	- 1.375E+02	1.020E+00	- 1.375E+02	- 1.375E+02	- 1.375E+02	- 1.375E+02	1.066E+00	
	Mean	- 3.286E+00	- 3.243E+00	- 3.322E+00	- 3.302E+00	- 3.270E+00	- 3.322E+00	- 3.294E+00	- 3.264E+00
F24	Std	5.542E-02	8.016E-02	1.706E-03	4.507E-02	7.462E-05	5.115E-02	5.924E-02	8.950E-02
	<i>t-values</i>	1.186E+00	- 9.217E-01	3.557E+00	2.112E+00	3.525E-01	1.632E+00	- 6.575E-01	
F25	Mean	- 5.486E+00	- 9.147E+00	- 9.508E+00	- 8.652E+00	- 9.989E+00	- 9.902E+00	- 1.015E+01	- 1.015E+01
	Std	3.450E+00	2.336E+00	1.676E+00	2.830E+00	3.217E+00	1.373E+00	1.807E-15	1.052E-05
F26	<i>t-values</i>	- 7.411E+00	- 2.360E+00	- 2.107E+00	- 2.906E+00	- 1.036E+00	- 1.000E+00	3.369E+00	
	Mean	- 7.839E+00	- 1.005E+01	- 1.036E+01	- 9.620E+00	- 1.040E+01	- 1.040E+01	- 1.040E+01	- 1.040E+01
F27	Std	3.455E+00	1.348E+00	1.788E-01	2.065E+00	1.284E-03	1.000E-16	1.000E-16	2.778E-05
	<i>t-values</i>	- 4.064E+00	- 1.447E+00	- 1.274E+00	- 2.076E+00	7.815E+00	- 8.347E-01	7.815E+00	
F28	Mean	- 8.136E+00	- 1.009E+01	- 1.054E+01	- 1.013E+01	- 1.054E+01	- 1.040E+01	- 1.054E+01	- 1.054E+01
	Std	3.525E+00	1.747E+00	3.978E-03	1.540E+00	1.482E+00	2.409E-03	0.000E+00	1.503E-05
w/ll	<i>t-values</i>	- 3.730E+00	- 1.413E+00	- 9.928E-01	- 1.430E+00	- 9.999E-01	- 4.864E+04	9.615E+00	
		16/5/2	16/5/2	16/4/3	13/5/5	13/7/3	15/6/2	2/12/9	

Table 9 Results on CEC2019 benchmark functions with recent and topmost algorithms

Functions		PSO	GWO	DE	LSHADE	CMAES	FFA	WSO	MGO	CMGO
cec01	Mean	7.020E+08	2.630E+08	1.676E+10	2.088E+07	3.866E+08	5.637E+09	7.069E+14	5.022E+04	1.023E+05
	Std	6.960E+08	3.140E+08	1.026E+10	2.691E+07	2.975E+08	3.924E+09	0.000E+00	8.165E+03	9.954E+04
	<i>t-values</i>	-5.524E+00	-4.586E+00	-8.947E+00	-4.230E+00	-7.115E+00	-7.868E+00	-3.890E+10	-3.890E+10	2.854E+00
cec02	Mean	1.734E+01	1.734E+01	1.734E+01	1.734E+01	1.065E+02	1.881E+01	1.734E+01	1.734E+01	1.734E+01
	Std	1.000E-27	5.110E-04	1.000E-27	3.441E-14	4.606E+01	7.260E+00	5.417E-08	1.000E-15	4.764E-13
	<i>t-values</i>	-3.285E+07	-1.375E+01	2.410E+00	2.241E+00	-1.060E+01	-1.104E+00	-7.071E+00	2.410E+00	
cec03	Mean	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01
	Std	9.030E-15	2.213E-06	4.449E-06	9.034E-15	1.333E-04	2.884E-06	9.034E-15	2.541E-09	5.141E-10
	<i>t-values</i>	-6.160E+04	5.491E+00	-6.986E+00	1.984E+00	-2.060E+00	-1.945E+00	1.984E+00	-2.542E+00	
cec04	Mean	2.099E+01	5.596E+01	2.219E+01	9.863E+00	2.257E+01	4.978E+01	1.075E+02	3.403E+01	1.038E+02
	Std	8.920E+00	1.544E+01	3.781E+00	4.792E+00	5.667E+00	2.629E+01	1.032E+02	1.318E+01	6.093E+01
	<i>t-values</i>	7.369E+00	4.172E+00	7.326E+00	8.422E+00	7.274E+00	4.462E+00	-1.681E-01	6.134E+00	
cec05	Mean	1.157E+00	1.422E+00	1.161E+00	1.036E+00	1.003E+00	1.165E+00	1.230E+00	1.135E+00	1.278E+00
	Std	1.210E-01	2.192E-01	5.682E-02	2.250E-02	1.391E-02	1.083E-01	2.503E-01	9.599E-02	1.854E-01
	<i>t-values</i>	3.013E+00	-2.732E+00	3.326E+00	7.122E+00	8.117E+00	2.884E+00	8.576E-01	3.764E+00	
cec06	Mean	7.851E+00	1.115E+01	8.173E+00	7.591E+00	1.093E+01	1.063E+01	1.134E+01	1.023E+01	9.554E+00
	Std	1.730E+00	4.796E-01	5.569E-01	1.037E+00	8.544E-01	6.759E-01	7.089E-01	6.304E-01	1.368E+00
	<i>t-values</i>	4.229E+00	-6.049E+00	5.119E+00	6.262E+00	-4.679E+00	-3.850E+00	-6.357E+00	-2.473E+00	
cec07	Mean	2.059E+02	3.749E+02	2.417E+02	9.054E+01	2.815E+02	6.661E+02	8.030E+02	2.246E+02	4.431E+02
	Std	1.220E+02	2.354E+02	1.000E+02	7.245E+01	2.411E+02	1.440E+02	1.264E+02	1.420E+02	2.681E+02
	<i>t-values</i>	4.409E+00	1.046E+00	3.855E+00	6.952E+00	2.454E+00	-4.013E+00	-6.650E+00	3.945E+00	
cec08	Mean	5.208E+00	5.198E+00	5.572E+00	4.573E+00	6.616E+00	5.790E+00	4.989E+00	4.840E+00	5.626E+00
	Std	7.110E-01	9.140E-01	4.076E-01	5.373E-01	6.861E-01	4.509E-01	8.977E-01	9.987E-01	8.120E-01
	<i>t-values</i>	2.122E+00	1.920E+00	3.289E-01	5.925E+00	-5.098E+00	-9.659E-01	2.882E+00	3.344E+00	
cec09	Mean	2.391E+00	4.489E+00	2.480E+00	2.468E+00	2.393E+00	2.780E+00	4.026E+00	2.425E+00	3.703E+00
	Std	3.750E-02	9.466E-01	5.026E-02	1.110E-01	1.626E-02	3.780E-01	1.062E+00	4.617E-02	8.035E-01
	<i>t-values</i>	8.934E+00	-3.465E+00	8.321E+00	8.338E+00	8.931E+00	5.693E+00	-1.328E+00	8.698E+00	
cec10	Mean	2.014E+01	2.046E+01	2.016E+01	2.010E+01	2.043E+01	2.044E+01	2.053E+01	2.033E+01	2.012E+01
	Std	1.440E-01	1.111E-01	1.419E-01	3.976E-02	8.390E-02	1.607E-01	9.328E-02	1.329E-01	1.444E-01
	<i>t-values</i>	-5.918E-01	-1.028E+01	-1.074E+00	9.195E-01	-1.024E+01	-8.031E+00	-1.294E+01	-5.858E+00	
w/ll	3/1/6	6/1/3	2/2/6	1/1/8	6/0/4	5/2/3	5/3/2	3/0/7		

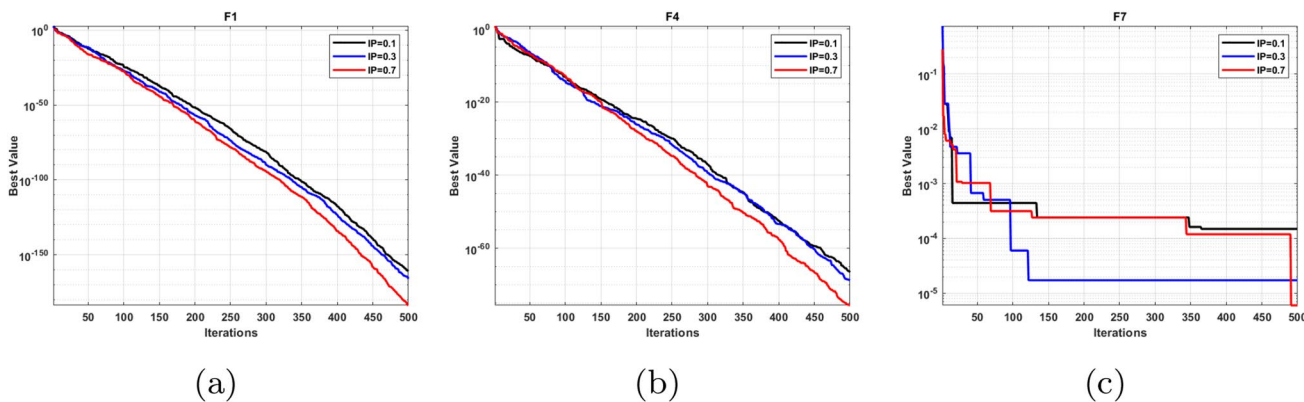


Fig. 6 Sensitivity analysis of CMGO algorithm

Table 10 Sensitivity analysis of CMGO with different values of parameter

Sl.no	Parameter	F1	F4	F7
1	IP = 0.1	1.5274E-161	3.0887E-67	1.4904E-04
2	IP = 0.3	1.1533E-166	1.8310E-69	1.7238E-05
3	IP = 0.7	2.6519E-184	2.7911E-76	5.9982E-06

globally. As a result, it is shown that the enhancements recommended in this study result in a better balance between the exploration and exploitation abilities of the classical MGO algorithm. Because of these enhancements, the convergence and search rate of CMGO outperform the other compared algorithms.

6 CMGO for Real-Life Engineering Problems

Engineering design is the technique of addressing the demands necessary to assemble a product. An intricate objective function and an extensive range of decision factors, including weight, strength, and wear, constitute this process of making decisions. Real design challenges might have a substantial number of design variables and a complex, nonlinear impact on the objective function that has to be optimized. Therefore, design considerations for seven traditional engineering design challenges have been addressed in this section to overcome the projected CMGO algorithm, and its outcomes are evaluated against other competing algorithms.

6.1 Tension/Compression Design Problem

The primary objective of this engineering design problem [66] is to reduce the weight of the spring through the combination of three decision variables: wire diameter (d), mean coil diameter (D), and number of active coils (N) as displayed in

Fig. 9. The problem may be described numerically as follows: Suppose $\vec{y} = [y_1, y_2, y_3] = [N, d, D]$

Minimize $f_1(\vec{y}) = (y_3 + 2)y_2y_1^2$ Subject to

$$c_1(\vec{y}) = 1 - \frac{y_2^3y_3}{71785y_1^4} \leq 0 \tag{13}$$

$$c_2(\vec{y}) = \frac{4y_2^2 - y_1y_2}{12566(y_2y_1^3 - y_1^4)} + \frac{1}{5108y_1^2} \leq 0 \tag{14}$$

$$c_3(\vec{y}) = 1 - \frac{140.45y_1}{y_2^2y_3} \leq 0 \tag{15}$$

$$c_4(\vec{y}) = \frac{y_1 + y_2}{1.5} - 1 \leq 0 \tag{16}$$

$$\begin{aligned} 0.05 &\leq y_1 \leq 2.00 \\ 0.25 &\leq y_2 \leq 1.30 \\ 2.00 &\leq y_3 \leq 15.0 \end{aligned} \tag{17}$$

The results of the analysis are summarized in Table 13, demonstrating that the CMGO can address this problem effectively by providing a better design than the enumerated algorithms such as PSO, GWO, WOA, MFO, ABC, and MGO.

6.2 Gear Train Design Problem

An excellent illustration of a mixed challenge is the design of a gear train, which requires determining a number of design elements that are continuous, integer variables, and discrete [30]. The challenge is basically that follows: Given a set input drive and a number of fixed output drive spindles,

Table 11 Statistical results obtained for the Wilcoxon rank-sum test on CEC2005 test functions

Functions	PSO vs CMGO		GWO vs CMGO		DE vs CMGO		LSHADE vs CMGO		CMAES vs CMGO		FFA vs CMGO		WSO vs CMGO		MGO vs CMGO	
	p-value	W	p-value	W	p-value	W	p-value	W	p-value	W	p-value	W	p-value	W	p-value	W
F1	1.779E-06	+	1.772E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+
F2	1.779E-06	+	1.778E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+
F3	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+
F4	1.779E-06	+	1.778E-06	+	1.779E-06	+	1.779E-06	+	1.778E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+
F5	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+
F6	4.622E-04	+	1.779E-06	+	1.778E-06	+	1.778E-06	+	1.621E-03	+	3.737E-01	+	1.779E-06	+	2.791E-05	+
F7	1.778E-06	+	1.778E-06	+	1.778E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+
F8	1.779E-06	+	1.778E-06	+	1.778E-06	+	1.778E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	3.383E-04	+
F9	1.778E-06	+	1.745E-06	+	1.778E-06	+	1.778E-06	+	1.778E-06	+	1.779E-06	+	1.779E-06	+	8.469E-06	+
F10	1.776E-06	+	1.723E-06	+	1.779E-06	+	1.779E-06	+	1.778E-06	+	1.779E-06	+	1.779E-06	+	NA	=
F11	1.777E-06	+	3.906E-03	+	1.778E-06	+	1.779E-06	+	1.778E-06	+	1.779E-06	+	1.779E-06	+	8.125E-01	-
F12	1.075E-01	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	4.314E-01	+	1.779E-06	+	1.779E-06	+	NA	=
F13	1.740E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.778E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+
F14	2.491E-05	+	4.607E-05	+	1.017E-03	+	NA	+	1.779E-06	+	NA	+	NA	+	9.766E-04	+
F15	9.303E-01	+	4.969E-05	+	1.779E-06	+	1.501E-02	+	1.779E-06	+	4.393E-06	+	3.183E-05	+	4.549E-05	+
F16	2.632E-06	+	6.491E-06	+	2.632E-06	+	2.632E-06	+	2.632E-06	+	2.632E-06	+	3.279E-01	-	2.632E-06	+
F17	1.779E-06	+	2.456E-03	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+
F18	1.779E-06	+	5.255E-02	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+
F19	1.779E-06	+	2.598E-02	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+	1.779E-06	+
F20	5.511E-02	-	6.325E-01	-	7.878E-06	+	1.388E-04	+	4.811E-01	-	1.779E-06	+	8.862E-03	+	6.770E-01	-
F21	2.666E-04	+	1.779E-06	+	2.645E-01	+	3.737E-01	+	3.737E-01	-	7.852E-01	+	3.182E-05	-	1.778E-06	+
F22	3.848E-01	+	1.779E-06	+	9.467E-01	+	1.779E-06	+	1.779E-06	+	3.183E-05	+	1.779E-06	+	1.779E-06	+
F23	6.472E-01	+	1.779E-06	+	3.660E-04	-	1.779E-06	+	3.183E-05	+	3.660E-04	-	1.779E-06	+	1.779E-06	+

Table 12 Statistical results obtained for the Wilcoxon rank-sum test on CEC2019 test functions

Functions	PSO vs CMGO		GWO vs CMGO		DE vs CMGO		LSHADE vs CMGO		CMAES vs CMGO		FFA vs CMGO		WSO vs CMGO		MGO vs CMGO		
	p-value	W	p-value	W	p-value	W	p-value	W	p-value	W	p-value	W	p-value	W	p-value	W	
cec01	1.778E-06	+	1.778E-06	+	1.779E-06	+	1.779E-06	+	1.778E-06	+	1.778E-06	+	1.778E-06	+	1.778E-06	+	9.411E-03
cec02	NA	=	1.777E-06	+	NA	=	NA	=	1.779E-06	+	1.779E-06	+	1.732E-06	+	NA	=	NA
cec03	NA	=	1.306E-04	+	1.779E-06	+	NA	=	1.779E-06	+	6.104E-05	+	NA	=	1.000E+00	+	1.000E+00
cec04	1.779E-06	+	6.274E-04	+	1.971E-06	+	1.779E-06	+	1.971E-06	+	3.126E-04	+	5.204E-01	-	3.604E-06	+	3.604E-06
cec05	5.405E-03	+	1.060E-02	+	3.007E-03	+	1.971E-06	+	1.971E-06	+	2.096E-02	+	1.604E-01	+	1.621E-03	+	1.621E-03
cec06	5.816E-04	+	1.268E-05	+	2.270E-04	+	7.043E-05	+	4.549E-05	+	1.219E-03	+	2.181E-06	+	1.339E-02	+	1.339E-02
cec07	1.134E-03	+	2.825E-01	+	5.389E-04	-	5.346E-06	+	3.201E-02	-	1.865E-03	+	2.909E-05	+	1.865E-03	+	1.865E-03
cec08	3.040E-02	+	7.272E-02	-	6.922E-01	-	5.920E-05	+	3.183E-05	+	5.204E-01	-	2.335E-02	+	1.407E-03	+	1.407E-03
cec09	1.779E-06	+	1.219E-03	+	1.779E-06	+	1.971E-06	-	1.779E-06	+	4.163E-05	+	2.077E-01	-	1.971E-06	+	1.971E-06
cec10	5.752E-01	-	5.346E-06	+	5.511E-02	-	9.959E-01	+	1.779E-06	+	9.540E-06	+	1.779E-06	+	1.388E-04	+	1.388E-04

how can the output drive spindles be driven by the input retaining the least amount of connecting gear in the train. The purpose of this challenge is to substantially reduce the cost of the gear ratio in the aforementioned mechanical engineering problem, depicted in Fig. 10. Four integer variables, $T_A, T_B, T_C,$ and $T_D,$ which represent the number of teeth on four different gearwheels, are contained in this problem. Below is an investigation of the challenge’s formulation.

Consider

$$\vec{y} = [y_1 \ y_2 \ y_3 \ y_4] = [T_A \ T_B \ T_C \ T_D]$$

Minimize

$$f(\vec{y}) = \left(\frac{1}{6.931} - \frac{y_3 y_2}{y_1 y_4} \right)^2$$

Variable range $12 \leq y_1, y_2, y_3, y_4 \leq 60$

In Table 14, the performance of CMGO is contrasted with that of the PSO, GSA, GWO, MVO, WOA, and MGO. These results illustrate how CMGO outperforms competing algorithms and generates outcomes comparable to MGO. These results show that CMGO is adept at addressing various problems.

6.3 Speed Reducer Design Problem

The objective of the speed reducer design challenge, which is schematically portrayed in Fig. 11, is to produce a speed reduction with the least amount of weight [67]. Since there are seven design variables in this problem, it is more challenging. The following factors constitute this category: width (x_1), teeth module (x_2), number of pinning teeth (x_3), length of shaft 1 between bearings (x_4), length of shaft 2 between bearings (x_5), diameter of shaft 1 (x_6), and diameter of shaft 2 (x_7). For this design problem, a mathematical equation is outlined as follows.

Consider

$$\vec{y} = [y_1 \ y_2 \ y_3 \ y_4 \ y_5 \ y_6 \ y_7] = [b \ m \ p \ l_1 \ l_2 \ D_1 \ d_1]$$

Minimize

$$f(\vec{y}) = 0.7854y_1 y_2^2 (3.3333y_3^2 + 14.9334y_3 - 43.0934) - 1.508y_1 (y_6^2 + y_7^2) + 7.4777(y_6^2 + y_7^2) + 0.7854(y_4 y_6^2 + y_5 y_7^2)$$

Subject to

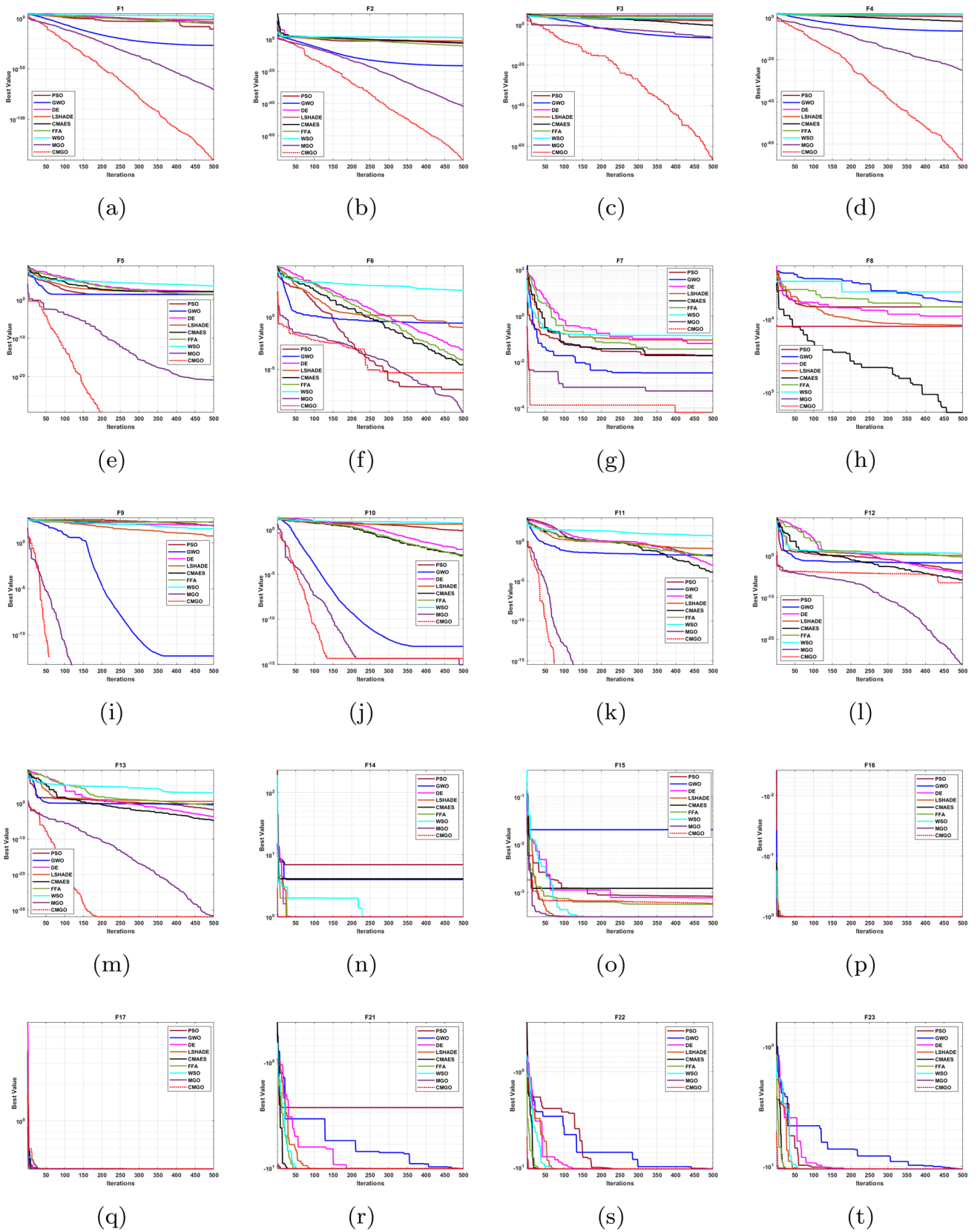


Fig. 7 Convergence curve of CEC2005 benchmark functions

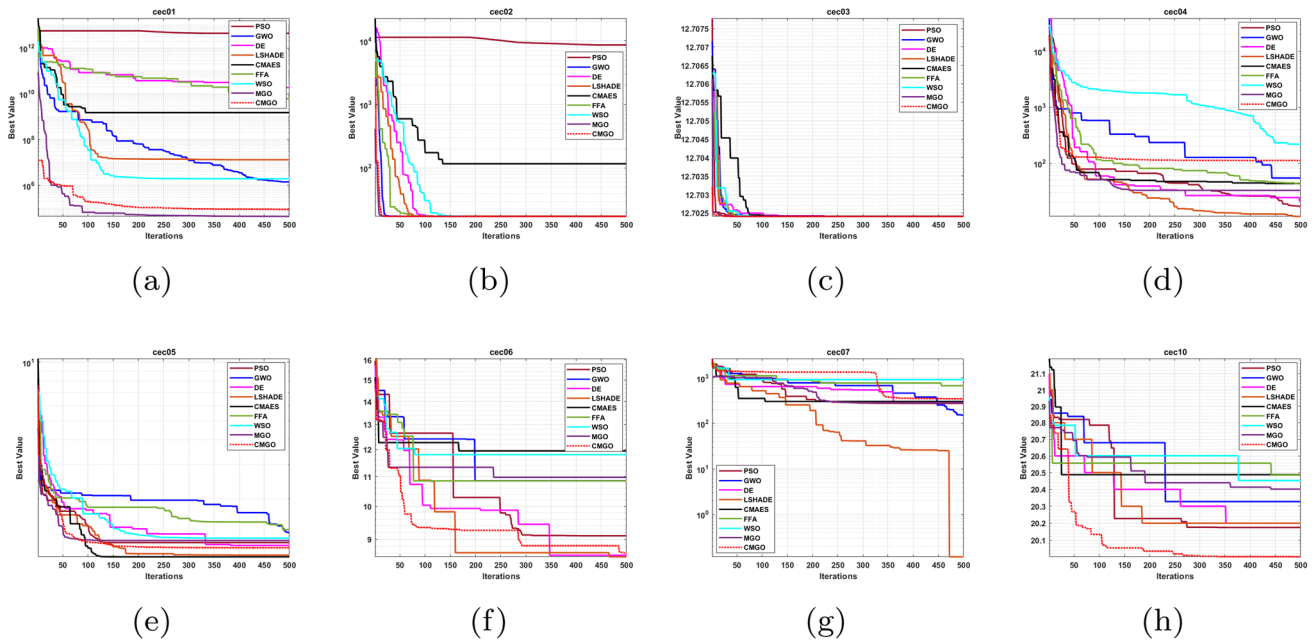


Fig. 8 Convergence curve of CEC2019 benchmark functions

Fig. 9 The schematic design of Tension/compression spring problem

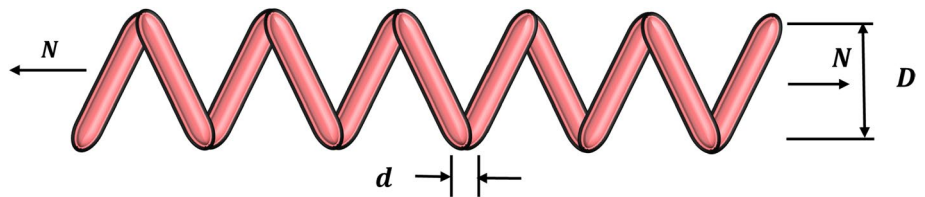


Table 13 The results obtained for solving tension/compression design problem

Algorithms	Optimal values of variable			Optimum weight
	d	D	N	
PSO	5.170E-02	3.580E-01	1.120E+01	1.270E-02
GWO	5.170E-02	3.570E-01	1.130E+01	1.270E-02
WOA	6.260E-02	6.804E-01	3.500E+00	1.466E-02
MFO	5.200E-02	3.642E-01	1.087E+01	1.271E-02
ABC	5.174E-02	3.581E-01	1.120E+01	1.276E-02
MGO	5.000E-02	3.174E-01	1.403E+01	1.272E-02
CMGO	5.058E-02	3.306E-01	1.300E+01	1.269E-02

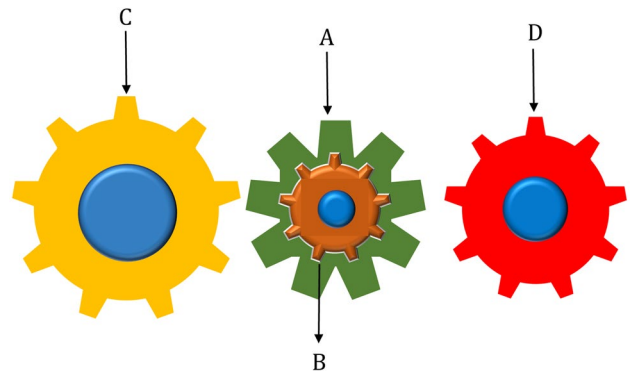


Fig. 10 The schematic design of Gear train design problem

Table 14 The results obtained in solving Gear train design problem

Algorithms	Optimal values of variable				Optimum Cost
	T_A	T_B	T_C	T_D	
PSO	5.200E+01	1.500E+01	2.900E+01	5.800E+01	2.358E-09
GWO	6.000E+01	2.416E+01	1.200E+01	3.349E+01	3.182E-12
SCA	4.395E+01	1.938E+01	1.479E+01	4.520E+01	1.537E-10
MFO	4.300E+01	1.900E+01	1.600E+01	4.900E+01	2.701E-12
ABC	4.900E+01	1.600E+01	1.900E+01	4.300E+01	2.701E-12
MGO	5.312E+01	1.304E+01	2.430E+01	4.133E+01	9.566E-14
CMGO	3.183E+01	1.242E+01	1.759E+01	4.755E+01	0.000E+00

$$\begin{aligned}
 c_1(\vec{y}) &= \frac{27}{y_1 y_2^2 y_3} - 1 \leq 0 \\
 c_2(\vec{y}) &= \frac{397.5}{y_1 y_2^2 y_3^2} - 1 \leq 0 \\
 c_3(\vec{y}) &= \frac{1.93 y_4^3}{y_2 y_6^4 y_3} - 1 \leq 0 \\
 c_4(\vec{y}) &= \frac{1.93 y_5^3}{y_2 y_7^4 y_3} - 1 \leq 0 \\
 c_5(\vec{y}) &= \frac{\sqrt{\left(\frac{745 y_4}{y_2 y_3}\right)^2 + 16.9 \times 10^6}}{110 y_6^3} - 1 \leq 0 \\
 c_6(\vec{y}) &= \frac{\sqrt{\left(\frac{745 y_5}{y_2 y_3}\right)^2 + 157.5 \times 10^6}}{85 y_7^3} - 1 \leq 0 \\
 c_7(\vec{y}) &= \frac{y_2 y_3}{40} - 1 \leq 0 \\
 c_8(\vec{y}) &= \frac{5 y_2}{y_1} - 1 \leq 0 \\
 c_9(\vec{y}) &= \frac{y_1}{12 y_2} - 1 \leq 0 \\
 c_{10}(\vec{y}) &= \frac{1.5 y_6 + 1.9}{y_4} - 1 \leq 0 \\
 c_{11}(\vec{y}) &= \frac{1.1 y_7 + 1.9}{y_5} - 1 \leq 0
 \end{aligned}
 \tag{17}$$

Where the range of the design variables b, m, p, l_1, l_2, D_1 , and d_1 were given as

$$\begin{aligned}
 2.6 \leq y_1 \leq 3.6 \\
 0.7 \leq y_2 \leq 0.8 \\
 y_3 \in \{17, 18, 19, \dots, 28\} \\
 7.3 \leq y_4 \\
 y_5 \leq 8.3 \\
 2.9 \leq y_6 \leq 3.9 \\
 5 \leq y_7 \leq 5.5
 \end{aligned}
 \tag{18}$$

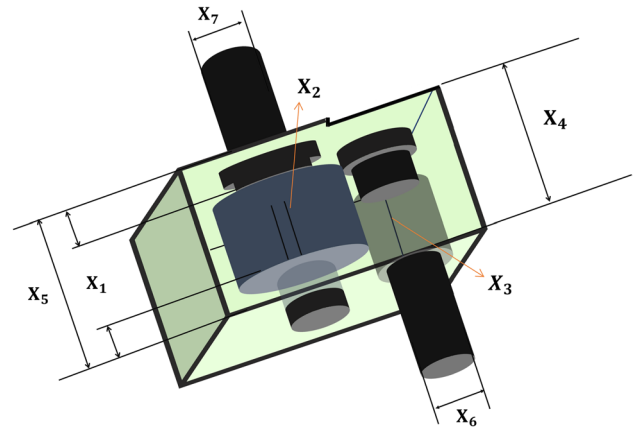


Fig. 11 The schematic design of Speed reducer design problem

Table 15 compares the best solutions obtained through CMGO with other optimizers like PSO, GWO, WOA, MVO, SCA, and MGO. The statistical results summarized in Table 15 divulge that CMGO generated the best optimum solutions among the competing algorithms.

6.4 Three-Bar Truss Design Problem

Three-bar truss design is a standard optimization problem in civil engineering [68]. Its primary objective is to establish the best values for two parameters (A_1 and A_2 , albeit a third value, $A_3 = A_1$, is also generated), which might help decrease weight when designing a truss similar to the one in Fig. 12 by determining the best values for those two parameters. The following equation quantitatively expresses the truss design problem:

Consider

Table 15 The results obtained in solving speed reducer design problem

Algorithms	Optimal values of variable							Optimal Cost
	<i>b</i>	<i>m</i>	<i>p</i>	<i>l</i> ₁	<i>l</i> ₂	<i>D</i> ₁	<i>d</i> ₁	
PSO	3.500E+00	7.000E-01	1.700E+01	8.300E+00	7.800E+00	3.352E+00	5.287E+00	3.006E+03
GWO	3.505E+00	7.000E-01	1.700E+01	7.463E+00	8.072E+00	3.402E+00	5.291E+00	3.022E+03
WOA	3.547E+00	7.068E-01	2.769E+01	8.130E+00	8.073E+00	3.347E+00	5.304E+00	5.508E+03
MVO	3.511E+00	7.000E-01	1.700E+01	7.525E+00	8.184E+00	3.474E+00	5.287E+00	3.044E+03
SCA	3.600E+00	7.000E-01	1.700E+01	7.300E+00	8.300E+00	3.470E+00	5.389E+00	3.145E+03
MGO	3.500E+00	7.000E-01	1.700E+01	7.300E+00	7.715E+00	3.350E+00	5.287E+00	2.994E+03
CMGO	3.500E+00	7.000E-01	1.700E+01	7.300E+00	7.716E+00	3.350E+00	5.287E+00	2.994E+03

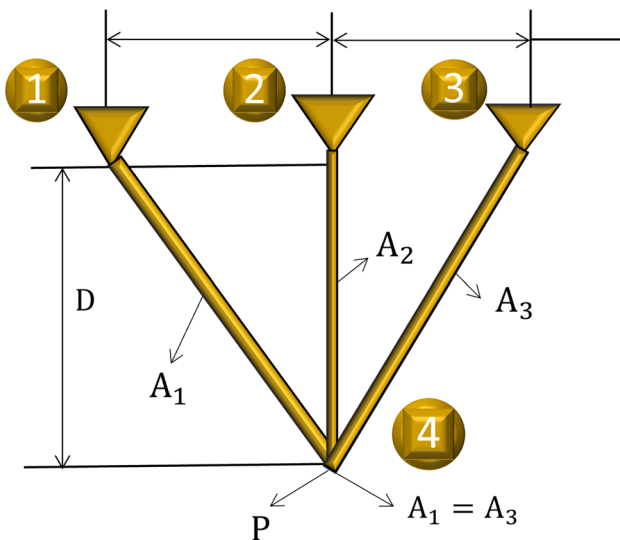


Fig. 12 The schematic design of Three-bar truss design problem

$$\vec{y} = [y_1 \ y_2] = [A_1 \ A_2]$$

Minimize

$$f(\vec{y}) = (2\sqrt{2y_1} + y_2) * l$$

Subject to

$$\begin{aligned} c_1(\vec{y}) &= \frac{\sqrt{2y_1} + y_2}{\sqrt{2y_1^2 + 2y_1y_2}} P - \sigma \leq 0, \\ c_2(\vec{y}) &= \frac{y_2}{\sqrt{2y_1^2 + 2y_1y_2}} P - \sigma \leq 0, \\ c_3(\vec{y}) &= \frac{1}{\sqrt{2y_2 + y_1}} P - \sigma \leq 0, \end{aligned} \tag{19}$$

Variable range $0 \leq y_1, y_2 \leq 1$

Here, $l = 100 \text{ cm}$, $\sigma = 2 \text{ kN/cm}^2$, $P = 2 \text{ kN/cm}^2$

The outcomes of the projected CMGO algorithm are juxtaposed with those of other optimizers like PSO, GSA,

Table 16 The results obtained in solving the three-bar truss problem

Algorithms	<i>y</i> ₁	<i>y</i> ₂	Optimal cost
PSO	7.878E-01	4.107E-01	2.639E+02
GWO	7.878E-01	4.110E-01	2.639E+02
WOA	7.623E-01	4.884E-01	2.645E+02
SCA	7.730E-01	4.555E-01	2.642E+02
SSA	8.722E-01	2.256E-01	2.693E+02
MGO	7.930E-01	3.963E-01	2.639E+02
CMGO	7.911E-01	4.014E-01	2.639E+02

GWO, MVO, WOA, and MGO to assess how effectively it performs in solving this problem. The approximate optimal cost achieved through the proposed approach and the added algorithms are summarized in Table 16. As this table demonstrates, the outcomes of the CMGO are relatively comparable to all the other algorithms. The suggested algorithm’s optimal cost is utilized as an indication of the outstanding performance of the CMGO in addressing this challenge.

7 Conclusions and Future Works

This study attempts to improve the recently proposed swarm-based algorithm called Mountain Gazelle Optimizer (MGO) by incorporating 10 distinct chaotic maps to form a newly developed algorithm, namely, Chaotic Mountain Gazelle Optimizer (CMGO), to accelerate convergence, prevent local optimums, and enhance MGO’s capacity to balance between exploration and exploitation stages. At first, 10 different chaotic maps were implemented individually and

compared to different chaotic maps along with MGO. Afterwards, ten chaotic maps were incorporated into MGO simultaneously and compared to top-performing algorithms. The outcomes of the investigation indicate that for the majority of the tested functions of CMGO, the accuracy of the generated solutions and the convergence rate were superior to those of other optimization techniques for the majority of the functions. Engineering challenge evaluations reveal that the recently established chaotic-based CMGO outperforms its competitors. Additionally, the statistical results of the Wilcoxon sign-rank test showed that the CMGO has a p-value of less than 0.05 for the CEC2005 and CEC2019 benchmark functions, indicating its statistically significant superiority over competitor algorithms. Regardless of its outstanding performance, the main limitation of the proposed CMGO method is that it has insignificant results when solving complex multi-modal functions and CEC2019 functions compared to other peers, which could affect the computational time. When optimizing problems with varying degrees of uncertainty, there is a deficiency of diverse search strategies during the exploration and exploitation stages. In future work, the CMGO method may be utilized for image processing, data mining, and design problems across a broad range of engineering disciplines. Furthermore, the CMGO can be modified to optimize the network's parameters to improve the network's performance.

Acknowledgements The authors would like to thank VIT University for supporting this research work.

Author Contributions Priteesha Sarangi: Conceptualization, Methodology, writing original draft. Prabhujit Mohapatra: Conceptualization, Methodology, Supervision, writing-review and editing.

Funding No funding was received for conducting this study.

Data availability All data generated or analyzed during this study are included in this article.

Declarations

Conflict of interest The authors declare no Conflict of interest relating to this work.

Informed consent Informed consent was obtained from all individual participants included in the study.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright

holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Gendreau, M., Potvin, J.-Y., et al.: Handbook of Metaheuristics, vol. 2. Springer (2010)
- Yang, X.-S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press (2010)
- Cao, Y., Wu, Q.: Evolutionary programming, In: Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97), IEEE, pp. 443–446 (1997)
- Rechenberg, I.: Evolutionsstrategien, In: Simulationsmethoden in der Medizin und Biologie: Workshop, Hannover, 29. Sept.–1. Okt. 1977, Springer, pp. 83–114 (1978)
- Holland, J.H.: Genetic algorithms. *Sci. Am.* **267**(1), 66–73 (1992)
- Koza, J.R.: Genetic programming as a means for programming computers by natural selection. *Stat. Comput.* **4**, 87–112 (1994)
- Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**(4), 341 (1997)
- Kennedy, J., Eberhart, R.: Particle swarm optimization, In: Proceedings of ICNN'95—international conference on neural networks, Vol. 4, IEEE, pp. 1942–1948 (1995)
- Dorigo, M., Di Caro, G.: Ant colony optimization: a new metaheuristic, In: Proceedings of the 1999 congress on evolutionary computation—CEC99 (Cat. No. 99TH8406), Vol. 2, IEEE, pp. 1470–1477 (1999)
- Yang, X.-S., Deb, S.: Cuckoo search via Lévy flights, In: 2009 World congress on nature & biologically inspired computing (NaBIC), Ieee, pp. 210–214 (2009)
- Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *J. Glob. Optim.* **39**, 459–471 (2007)
- Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014)
- Mirjalili, S., Lewis, A.: The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016)
- Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **114**, 163–191 (2017)
- Mirjalili, S.: Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **89**, 228–249 (2015)
- Mirjalili, S.: The ant lion optimizer. *Adv. Eng. Softw.* **83**, 80–98 (2015)
- Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A.A., Al-Qaness, M.A., Gandomi, A.H.: Aquila optimizer: a novel metaheuristic optimization algorithm. *Comput. Ind. Eng.* **157**, 107250 (2021)
- Braik, M., Hammouri, A., Atwan, J., Al-Betar, M.A., Awadallah, M.A.: White shark optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowl.-Based Syst.* **243**, 108457 (2022)
- Mohapatra, S., Mohapatra, P.: American zebra optimization algorithm for global optimization problems. *Sci. Rep.* **13**(1), 5211 (2023)
- Gopi, S., Mohapatra, P.: A modified whale optimisation algorithm to solve global optimisation problems, In: Proceedings of 7th International Conference on Harmony Search, Soft Computing and Applications: ICHSA 2022, Springer, pp. 465–477 (2022)
- Kirkpatrick, S., Gelatt, C.D., Jr., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983)

22. Erol, O.K., Eksin, I.: A new optimization method: big bang-big crunch. *Adv. Eng. Softw.* **37**(2), 106–111 (2006)
23. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: Gsa: a gravitational search algorithm. *Inf. Sci.* **179**(13), 2232–2248 (2009)
24. Hatamlou, A.: Black hole: a new heuristic optimization approach for data clustering. *Inf. Sci.* **222**, 175–184 (2013)
25. Mirjalili, S.: Sca: a sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **96**, 120–133 (2016)
26. Mirjalili, S., Mirjalili, S.M., Hatamlou, A.: Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **27**, 495–513 (2016)
27. Faramarzi, A., Heidarinejad, M., Stephens, B., Mirjalili, S.: Equilibrium optimizer: a novel optimization algorithm. *Knowl.-Based Syst.* **191**, 105190 (2020)
28. Glover, F.: Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* **13**(5), 533–549 (1986)
29. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. *SIMULATION* **76**(2), 60–68 (2001)
30. Rao, R.V., Savsani, V.J., Vakharia, D.: Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput. Aided Des.* **43**(3), 303–315 (2011)
31. Tanabe, R., Fukunaga, A.S.: Improving the search performance of shade using linear population size reduction, In: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, pp. 1658–1665 (2014)
32. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evol. Comput.* **11**(1), 1–18 (2003)
33. Simon, D.: Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **12**(6), 702–713 (2008)
34. Shayanfar, H., Gharehchopogh, F.S.: Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems. *Appl. Soft Comput.* **71**, 728–746 (2018)
35. Sarangi, P., Mohapatra, P.: A novel cosine swarm algorithm for solving optimization problems, In: Proceedings of 7th International Conference on Harmony Search, Soft Computing and Applications: ICHSA 2022, Springer, pp. 427–434 (2022)
36. Abdel-Basset, M., El-Shahat, D., Jameel, M., Abouhawwash, M.: Young's double-slit experiment optimizer: a novel metaheuristic optimization algorithm for global and constraint optimization problems. *Comput. Methods Appl. Mech. Eng.* **403**, 115652 (2023)
37. Wolpert, D.H., McReady, W.: No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**, 67–82 (1997). (**IEEE Press, NY, USA** **10**, 4235–585893)
38. Abdollahzadeh, B., Gharehchopogh, F.S., Khodadadi, N., Mirjalili, S.: Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. *Adv. Eng. Softw.* **174**, 103282 (2022)
39. Chandrasekaran, K., Thaveedhu, A.S.R., Manoharan, P., Periyasamy, V.: Optimal estimation of parameters of the three-diode commercial solar photovoltaic model using an improved Berndt-hall-hausman method hybridized with an augmented mountain gazelle optimizer. *Environ. Sci. Pollut. Res.* **30**(20), 57683–57706 (2023)
40. Gharehchopogh, F.S., Ucan, A., Ibrikci, T., Arasteh, B., Isik, G.: Slime Mould algorithm: a comprehensive survey of its variants and applications. *Arch. Comput. Methods Eng.* **30**(4), 2683–2723 (2023)
41. Piri, J., Mohapatra, P., Acharya, B., Gharehchopogh, F.S., Geroiannis, V.C., Kanavos, A., Manika, S.: Feature selection using artificial gorilla troop optimization for biomedical data: a case analysis with covid-19 data. *Mathematics* **10**(15), 2742 (2022)
42. Pecora, L.M., Carroll, T.L.: Synchronization in chaotic systems. *Phys. Rev. Lett.* **64**(8), 821 (1990)
43. Zhang, Y.-T., Zhou, W., Yi, J.: A novel adaptive chaotic bacterial foraging optimization algorithm, In: 2016 International Conference on Computational Modeling, Simulation and Applied Mathematics (2016)
44. Gharehchopogh, F.S., Ibrikci, T.: An improved African vultures optimization algorithm using different fitness functions for multi-level thresholding image segmentation. *Multimed. Tools Appl.* **83**(6), 16929–16975 (2023)
45. Özbay, E., Özbay, F.A., Gharehchopogh, F.S.: Peripheral blood smear images classification for acute lymphoblastic leukemia diagnosis with an improved convolutional neural network. *J. Bionic Eng.* (2023). <https://doi.org/10.1007/s42235-023-00441-y>
46. Shen, Y., Zhang, C., Gharehchopogh, F.S., Mirjalili, S.: An improved whale optimization algorithm based on multi-population evolution for global optimization and engineering design problems. *Expert Syst. Appl.* **215**, 119269 (2023)
47. Saha, S., Mukherjee, V.: A novel chaos-integrated symbiotic organisms search algorithm for global optimization. *Soft. Comput.* **22**, 3797–3816 (2018)
48. Gupta, S., Deep, K.: An opposition-based chaotic grey wolf optimizer for global optimisation tasks. *J. Exp. Theoret. Artif. Intell.* **31**(5), 751–779 (2019)
49. Kumar, Y., Singh, P.K.: A chaotic teaching learning based optimization algorithm for clustering problems. *Appl. Intell.* **49**(3), 1036–1062 (2019)
50. Li, X.-D., Wang, J.-S., Hao, W.-K., Zhang, M., Wang, M.: Chaotic arithmetic optimization algorithm. *Appl. Intell.* **52**(14), 16718–16757 (2022)
51. Kaur, G., Arora, S.: Chaotic whale optimization algorithm. *J. Comput. Des. Eng.* **5**(3), 275–284 (2018)
52. Mohapatra, S., Mohapatra, P.: Fast random opposition-based learning golden jackal optimization algorithm. *Knowl.-Based Syst.* **275**, 110679 (2023)
53. Gharehchopogh, F.S., Abdollahzadeh, B., Barshandeh, S., Arasteh, B.: A multi-objective mutation-based dynamic harris hawks optimization for botnet detection in iot. *Internet of Things* **24**, 100952 (2023)
54. Gharehchopogh, F.S.: An improved Harris hawks optimization algorithm with multi-strategy for community detection in social network. *J. Bionic Eng.* **20**(3), 1175–1197 (2023)
55. Chandran, V., Mohapatra, P.: Enhanced opposition-based grey wolf optimizer for global optimization and engineering design problems. *Alex. Eng. J.* **76**, 429–467 (2023)
56. Mohapatra, S., Mohapatra, P.: An improved golden jackal optimization algorithm using opposition-based learning for global optimization and engineering problems. *Int. J. Comput. Intell. Syst.* **16**(1), 147 (2023)
57. Gharehchopogh, F.S., Khargoush, A.A.: A chaotic-based interactive autodidactic school algorithm for data clustering problems and its application on covid-19 disease detection. *Symmetry* **15**(4), 894 (2023)
58. Li, J.-W., Cheng, Y.-M., Chen, K.-Z.: Chaotic particle swarm optimization algorithm based on adaptive inertia weight, In: The 26th Chinese Control and Decision Conference (2014 CCDC), IEEE, pp. 1310–1315 (2014)
59. Kumar, S., Yildiz, B.S., Mehta, P., Panagant, N., Sait, S.M., Mirjalili, S., Yildiz, A.R.: Chaotic marine predators algorithm for global optimization of real-world engineering problems. *Knowl.-Based Syst.* **261**, 110192 (2023)
60. Özbay, F.A.: A modified seahorse optimization algorithm based on chaotic maps for solving global optimization and engineering problems. *Eng. Sci. Technol. Int. J.* **41**, 101408 (2023)

61. dos Santos Coelho, L., Mariani, V.C.: Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. *Expert Syst. Appl.* **34**(3), 1905–1913 (2008)
62. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.-P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *KanGAL Report 2005005*(2005), 2005 (2005)
63. Liang, J.-J., Qu, B., Gong, D., Yue, C.: Problem definitions and evaluation criteria for the cec 2019 special session on multimodal multiobjective optimization, Computational Intelligence Laboratory, Zhengzhou University (2019)
64. Mohapatra, P., Das, K.N., Roy, S.: A modified competitive swarm optimizer for large scale optimization problems. *Appl. Soft Comput.* **59**, 340–362 (2017)
65. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Inf. Sci.* **180**(10), 2044–2064 (2010)
66. Arora, J.: *Introduction to Optimum Design*. Elsevier (2004)
67. Mezura-Montes, E., Coello, C.A.C.: Useful infeasible solutions in engineering optimization with evolutionary algorithms, In: *MICAI 2005: Advances in Artificial Intelligence: 4th Mexican International Conference on Artificial Intelligence*, Monterrey, Mexico, November 14–18, 2005. *Proceedings 4*, Springer, pp. 652–662 (2005)
68. Ray, T., Liew, K.-M.: Society and civilization: an optimization algorithm based on the simulation of social behavior. *IEEE Trans. Evol. Comput.* **7**(4), 386–396 (2003)

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