Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 192 (2022) 514–536
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original articles

Chaotic hunger games search optimization algorithm for global optimization and engineering problems

Funda Kutlu Onay, Salih Berkan Aydemir*

Computer Engineering Department, Amasya University, Amasya, Turkey

Received 5 July 2021; received in revised form 27 August 2021; accepted 16 September 2021
Available online 29 September 2021

Abstract

Chaotic maps have the characteristics of ergodicity and non-repeatability. Owing to these properties, they provide fast convergence by effectively scanning the search space in a metaheuristic optimization algorithm. The Hunger Games Search (HGS) is a metaheuristic algorithm modeled on the foraging and hunger instincts of animals. In this study, ten chaotic maps have been applied to the classical HGS method. The control of two random values in the HGS algorithm has been carried out with chaotic maps in three alternative scenarios. Accordingly, it has been observed that Scenario 2 exhibits a more stable and faster convergence than other scenarios. The performance of the proposed chaotic HGS has been evaluated on CEC2017 and 23 classical benchmark problems. The proposed algorithm has been applied to real engineering problems for cantilever beam design, tension/compression, and speed reducer, and the results have been compared with classical HGS and state-of-art algorithms in the literature. It can be seen that chaotic HGS yields promising results compared to other studies in the literature.

Keywords: Metaheuristic algorithms; Chaotic maps; Optimization; Engineering design problems; Hunger games search

1. Introduction

Metaheuristic optimization algorithms (MOAs) are frequently used optimization methods to solve real-world problems. The optimization process finds the optimal decision variables of a problem by minimizing or maximizing the objective function. They are also used in solving engineering problems due to their high efficiency and low computational complexity. Both mathematical and engineering problems can produce successful results with MOAs [59]. MOAs replace traditional optimization algorithms due to gradient-free mechanisms and high local optimal avoidance ability [8,86]. MOAs consist of two main phases as exploration and exploitation. The balance between these two phases is the cornerstone of an MOA. If exploration is effective in the MOA algorithm, random operators are generated to discover different search space regions. However, MOA may get stuck at the local optimum. On the other hand, if the exploitation phase is effective, the optimization method tries to determine the most suitable solution in the search field. It is still possible for the MOA to fall into the local minimum trap. So it means that, the balance between these two phases is very critical to be able to converge to the global optimum. In the last 10 years, many new MOA algorithms have been proposed to balance the exploitation and exploration phases.

* Corresponding author.
E-mail addresses: funda.kutlu@amasya.edu.tr (F. Kutlu Onay), salih.aydemir@amasya.edu.tr (S.B. Aydemir).

https://doi.org/10.1016/j.matcom.2021.09.014

MOAs can be discussed under four main headings: Physics–Mathematics based algorithms (PMA), Human-based algorithms, Evolutionary Algorithms (EA), and Swarm Intelligence (SI).

PMAs have been discovered by inspiring from physical phenomena and mathematical models. Optimization algorithms based on some mathematical models and physics laws are given as follows: Sine cosine algorithm (SCA) [53], Arithmetic optimization algorithm [1] (AOA), Simulated annealing (SA) [40], Multi-Verse Optimizer (MVO) [56], Gravitational search algorithm (GSA) [61], Charged System Search (CSS) [33], Blackhole (BH) algorithm [23], Supernova optimizer [29], Tabu Search (TS) [19]. Human-based optimization is recommended by inspiring from human behavior and cooperation. The algorithms of this method are given as follows: Imperialist Competitive Algorithm (ICA) [7], Teaching–Learning-Based Optimization Algorithm (TLBOA) [60], Political optimizer (PO) [5], and Socio Evolution Learning Optimization Algorithm (SELOA) [43]. PO and SELOA algorithms are also known as socio-inspired metaheuristics.

The evolutionary algorithm (EA) is inspired by biological evolution in nature such as reproduction, mutation, recombination, and selection. During evolution, individuals compete and may pair up to find the most suitable candidate solution among them. EA algorithms frequently used in the literature are Genetic algorithms (GA) [26], Evolution strategy (ES) [65], Novelty search for global optimization (NS) [18], Biogeography based optimizer (BBO), [72], Evolutionary Programming (EP) [42], Evolutionary algorithm based on decomposition [85], and Differential Evolution (DE) [73]. The (key) features of swarm-based algorithms are self-organization and division of labor. Self-organization is described as the ability to transform the components of a system into a suitable form without any outside assistance. Various movements of swarms have a significant role in SI algorithms. Some swarm-based optimization algorithms that produce effective results in the literature are as follows: Particle swarm optimization (PSO) [39], Crow search algorithm (CSA) [6], Dragonfly algorithm (DA) [52], Artificial bee colony (ABC) [33], Cuckoo Search (CS) [81], Moth flame optimization [51], Grey wolf optimizer (GWO) [57], Whale optimization algorithm (WOA) [55], Salp swarm algorithm (SSA) [54], and Grasshopper optimization algorithm (GOA) [66].

A new MOA can be obtained using three different methodologies. Firstly, a new MOA can be proposed, inspired by different areas, such as the aforementioned methods (PMA, EA, and SI). In this methodology, the proposed algorithm represents a real-world event. Second, the exploitation and/or exploration steps of different optimization algorithms can be hybridized. The search mechanisms of the two different methods and the escape strategies from the local optimum can be changed in terms of their superiority to each other. Some hybrid methods can be given as follows: SCA-GWO [22], SCA-ABC [21], SA-WOA [48], GWO-PSO [70], GWO-GOA [58], and WOA-DE [47]. Lastly, various learning mechanisms or methods that enable a broader scanning of the solution space can be integrated into the MOAs. For example, opposition-based learning, quantum-behaved approaches, approaches with chaotic maps, levy flight-based approaches, etc. Some optimization algorithms proposed to improve the exploitation or exploration phase are given as follows: Selective opposition based-GWO [14], opposition based WOA [2], Levy flight based WOA [45], Levy flight based ABC [71], quantum based-SCA [63], quantum based-PSO [25], and Networked evolutionary algorithm (Swarm-networked algorithm, NEA) [17]. Chaotic-based optimization algorithms will be discussed in detail in the literature review section.

In this study, two random variables in the exploitation and exploration phases of the HGS algorithm have been controlled with 10 different chaotic maps. The proposed chaotic HGS algorithm is tested on fundamental benchmark problems, CEC2017 and real-world engineering problems. The chaotic HGS algorithm gives better results than the classical HGS in most cases. It can also be seen that chaotic HGS has promising convergence capability when compared to other state of art optimization algorithms in the literature.

The rest of this study is given as follows: Section 2 discusses an in-depth literature review on hybrid chaotic algorithms. In Section 3, the hunger games algorithm is explained. In Section 4, chaotic maps based on chaos theory are defined. In Section 5, the proposed chaotic HGS algorithm is explained. In Section 6, the performance of the chaotic HGS is evaluated. Section 7 deals with the results of chaotic HGS in engineering problems. In Section 8, the statistical relationship between the proposed method and the existing methods in the literature is presented. Finally, Section 9 includes conclusions and future work.

2. Literature review for chaotic based optimizations

Comprehensive scanning of the solution space is performed by means of random numbers. If jumping to different points of the solution space is provided, convergence to the global optimum point can be achieved. Hence,

Table 1
Literature review for chaotic based optimizations.

Algorithm	Chaotic maps
Chaotic dragonfly [69]	Chebyshev, Circle, Gauss/mouse, Iterative, Logistic, Piecewise, Sine, Singer, Sinusoidal, Tent
Chaotic salp swarm [68]	Chebyshev, Circle, Gauss/mouse, Iterative, Logistic, Piecewise, Sine, Singer, Sinusoidal, Tent
Chaos-enhanced moth-flame [27]	Chebyshev, Circle, Gauss/mouse, Iterative, Logistic, Piecewise, Sine, Singer, Sinusoidal, Tent
Chaotic crow search [64]	Chebyshev, Circle, Gauss/mouse, Iterative, Logistic, Piecewise, Sine, Singer, Sinusoidal, Tent
Chaotic cuckoo search [77]	Chebyshev, Circle, Gauss/mouse, Intermittency, Iterative, Liebovitch, Logistic, Piecewise, Sine, Singer, Sinusoidal, Tent
Chaotic multi-verse optimization [67]	Chebyshev, Circle, Gauss/mouse, Iterative, Logistic, Piecewise, Sine, Singer, Sinusoidal, Tent
Chaotic Henry gas solubility [83]	Chebyshev, Circle, Gauss/mouse, Iterative, Logistic, Piecewise, Sine, Singer, Sinusoidal, Tent
Chaotic performance-based optimization [38]	Chebyshev
Chaotic Krill Herd algorithm [78]	Chebyshev, Circle, Gauss/mouse, Intermittency, Iterative, Liebovitch, Logistic, Piecewise, Sine, Singer, Sinusoidal, Tent
Chaotic genetic algorithm [84]	Logistic
Chaotic charged system search [75]	Logistic, Tent, Sinusoidal, Gauss/mouse, Circle, Sinus, Henon, Ikeda, Liebovitch, Zaslavskii
Chaotic bee colony [3]	Logistic, Circle, Gauss, Henon, Sinusoidal, Sinus, Tent
Chaotic Slime Mould Algorithms [15,31,87]	Chebyshev, Piecewice, Sinusoidal
Chaotic Harris Hawks [30]	Chebyshev, Circle, Gauss/mouse, Iterative, Logistic, Piecewise, Sine, Singer, Sinusoidal, Tent
Chaotic Imperialist competitive algorithm [74]	Logistic, Tent, iterative, Sinusoidal, Circle, Gauss, Sinus
Chaotic whale optimization [35]	Logistic, Cubic, Sine, Sinusoidal, Singer, Circle, Iterative Tent, Piecewise, Gauss/mouse
Chaotic grey wolf optimization [41]	Bernoulli, Logistic , Chebyshev, Circle, Cubic, iterative, Piecewise, Singer, Sinusoidal, Tent
Chaotic grey wolf optimization [41]	Bernoulli, Logistic , Chebyshev, Circle, Cubic, iterative, Piecewise, Singer, Sinusoidal, Tent

randomness is a substantial concept for an optimization algorithm. Chaotic maps have an active role in the control of randomness. In this section, the main MOA studies hybridized with chaotic mapping in the literature are mentioned.

Table 1 shows the metaheuristic algorithms in the literature in terms of chaotic maps. In the chaotic dragonfly algorithm, chaotic maps are used instead of the parameters (a,s,c,e,f,w) that provide the balance between the exploitation and exploration [69]. The authors tested their proposed method in feature selection problems. In the chaotic salp swarm algorithm, r2, the parameter which takes into account the update position of salp, is replaced with chaotic maps [68]. The authors tested the chaotic salp swarm in both basic benchmark functions and feature selection problems. In Chaos-enhanced moth-flame optimization, upper and lower bounds of moths were controlled by chaotic variables [27]. In the related study, basic benchmark problems and engineering problems are tested. Rızk et al. integrated chaotic maps into the crowd search algorithm [64]. In the chaotic crow search algorithm, four different scenarios are proposed. These scenarios are used with chaotic vectors based on random selections in the position update vector. In another study, chaotic vectors are used as a random parameter in Henry gas solubility optimization algorithm [83]. Yıldız et al. applied the chaotic Henry gas solubility optimization algorithm to real-world engineering problems. Kaur and Arora integrated chaos theory into the whale optimization process [35]. The authors have shown that the tent map produces effective results. Kohli and Arora proposed the Chaotic grey wolf optimization algorithm for constrained optimization problems [41]. Another study among the current chaotic studies is the slime mould algorithm. Dhawale et al. used a chaotic slime mould based on sinusoidal mapping for numerical and multidisciplinary design optimization problems [15].

Handling the randomness in the structure of an MOA leads to multiple scenarios. In addition, each of the chaotic maps with different structures exhibits different chaotic behavior. All the methods mentioned in Table 1 are either

dependent on a single scenario or the chaotic map type usage is low level. Considering the aforementioned chaotic-based hybrid methods, the types of chaotic maps and which part they are used play an important role in global convergence. In other words, the type of a chaotic map is as important as the phase in which the chaotic map is used. Due to the nature of chaotic maps, their non-repeatability and ergodic structures have enabled the development of optimization methods. In this study, first of all, all the random variables included in the HGS algorithm were discussed, and then interpretations were made on the variables in which chaotic maps were effective.

3. Hunger games search

Animals survive with their sensory knowledge within some calculation rules and in interaction with their environment. The survival, reproduction and food chances of animals are provided by their numerical logical rules [62]. Hunger is the most important factor in the life of animals, as it directly affects the homeostatic balance, behavior, decisions and actions. As such, animals perpetually seek food to maintain this balance and alternate between exploration, defense, and competitive activities as required, indicating the amount of fluid feeding strategies [10].

3.1. Mathematical model

In this section, the classical HGS algorithm is mentioned which is the first part of the proposed method. The mathematical model of this method, which is based on hunger, has been clarified in all details and application stages. The method consists of two main steps: Approach food and hunger role [80].

3.1.1. Approach food

Usually, social animals cooperate in search of food. However, individuals who will not participate in cooperation are also possible. In this case, the main equation of the classical HGS method represents both individual and collaborative foraging and is expressed as follows:

$$\overrightarrow{X}_{t+1} = \begin{cases} g_1 : \overrightarrow{X}_t.(1+rnd), r_1 > l \\ g_2 : \overrightarrow{WH_1} \cdot \overrightarrow{X}_{best} + RA \cdot \overrightarrow{WH_2} \cdot \left| \overrightarrow{X}_{best} - \overrightarrow{X}_t \right|, \\ r_1 > l, r_2 > E \\ g_3 : \overrightarrow{WH_1} \cdot \overrightarrow{X}_{best} - RA \cdot \overrightarrow{WH_2} \cdot \left| \overrightarrow{X}_{best} - \overrightarrow{X}_t \right|, \\ r_1 > l, r_2 < E \end{cases} \tag{1}$$

where RA is in the range of $[-a, a]$; r_1 and r_2 are two random numbers in the range [0,1]; rnd is a random number that provides a normal distribution; t is the current iteration index; $\overrightarrow{WH_1}$ and $\overrightarrow{WH_2}$ are weights of hunger; $\overrightarrow{X}_{best}$ is the best individual of current iteration; $\overrightarrow{X}_t$ is the location of each individual; l is an important parameter for improving the algorithm; $\overrightarrow{X}_t.(1+rnd)$ refers to the hunger and randomness of an agent in the current situation in foraging; $\left| \overrightarrow{X}_{best} - \overrightarrow{X}_t \right|$ is the range of activity of the current individual in the current time and if this is multiplied by $\overrightarrow{WH_2}$, it affects the influence of hunger on the range of activity. RA is a controller and is used to limit the range of activity. When an individual is not hungry it stops seeking, in which case RA is gradually reduced to 0. By adding or subtracting $\overrightarrow{WH_1} \cdot \overrightarrow{X}_{best}$ to the activity range, the individual is informed by his/her peers of reaching the food location and then it is simulated to search for food at the current location. $\overrightarrow{WH_1}$ is the error in grasping and the definition of E, which controls the variation for all positions, are given in the following equation:

$$E = sech(|AllFit_i - DestFit|), \mathrm{i} \in 1, 2, \ldots, n \tag{2}$$

where $AllFit_i$ shows the fitness of each individual in each iteration; $DestFit$ represents the destination fitness; $sech$ is a hyperbolic function and its equation is:

$$sech(x) = \frac{2}{e^x + e^{-x}} \tag{3}$$

The equation of RA is defined as follows:

$$RA = sh(2.rand - 1), sh = 2(1 - \frac{t}{MaxIt}) \tag{4}$$

where $rand$ is a random number in the interval of [0,1], and $MaxIt$ is the maximum number of iterations.

Here, the search directions take place according to two source points: $\overrightarrow{X}$ and $\overrightarrow{X}_{best}$. The search according to $\overrightarrow{X}$, the first game instruction, represents individuality. In other words, there is an individual who is not involved in cooperation and is only looking for food for itself. In the second game instruction, there is a search according to the best, $\overrightarrow{X}_{best}$. In this method of representing cooperation, there are three effective parameters: RA, $\overrightarrow{WH_1}$ and $\overrightarrow{WH_2}$. In the main equation, the rules allow individuals to discover places that are close to and far from the optimal solution, and this guarantees a certain search of the solution domain classes. It can also be applied to higher-dimensional search areas.

3.1.2. Hunger role

The hunger characteristics of the searchers are simulated at this stage. $\overrightarrow{WH_1}$ and $\overrightarrow{WH_2}$ are expressed as follows:

$$\overrightarrow{WH_1} = \begin{cases} Hun_i \cdot \frac{N}{sHun} \times r_4, r_3 < l \\ 1, r_3 > l \end{cases} \tag{5}$$

$$\overrightarrow{WH_2} = (1 - e^{(-|Hun_i - sHun|)}) \times r_5 \times 2 \tag{6}$$

where Hun_i is the hunger of each individual; N is the number of individuals; $sHun$ is equal to $sum(Hun)$ and means hunger feelings of all individuals; r_3, r_4 and r_5 are the random numbers as r_1 and r_2. Hun_i for each individual is calculated as follows:

$$Hun(i) = \begin{cases} 0, AllFit_i == DestFit \\ Hun_i + H, AllFit_i != DestFit \end{cases} \tag{7}$$

where $AllFit_i$ is as described in Eq. (2). At each iteration, the hunger of the best individual is set to 0 and an H value is added to the *original hunger* value for the others. H is called *hunger sensation*. From here, we can see that each individual's H value is different. This H value is obtained as follows:

$$TH = \frac{AllFit_i - DestFit}{WorstFit - DestFit} \times r_6 \times 2 \times (UpperB - LowerB) \tag{8}$$

$$H = \begin{cases} \text{LH} \times (1 + r), TH < LH \\ TH, TH \geq LH \end{cases} \tag{9}$$

where r_6 is the random number similar to other random parameters; $WorstFit(i)$ is the worst fitness so far in the current iteration; $UpperB$ and $LowerB$ correspond to upper and lower bounds of feature space, respectively. The H value has a lower bound of LH. To make the algorithm guarantee the best performance, the upper and lower limits must be controlled.

The pseudo code of the HGS algorithm is given in Algorithm 1:

4. Chaotic maps

The chaos theory, a sub-branch of mathematics, is a case of deterministic randomness based on nonlinear and complex dynamical systems [82]. In optimization algorithms, searches in the solution space are usually performed randomly. Thanks to chaotic mapping, an ergodic and chaos-free system is obtained by using chaotic values instead of this randomness [68]. Therefore, it allows more general searches than stochastic searches. The ergodic and non-ergodic system behaviors can be illustrated as shown in Fig. 1. Chaotic systems are self-contained, and local minimum and maximum problems can be avoided by chaotic mapping. The ten best known one-dimensional chaotic maps used in this study are explained in Table 2.

Algorithm 1 Pseudo-code of classical HGS

procedure CLASSICAL HGS
  Initialize the parameters:N, l, dim and $sHun$
  max. no of iter. $MaxIt$,
  pos. of solutions X_i, $i = 1, 2, 3, \ldots, N$.
  while $t <= MaxIt$ **do**
    Calculate the fitness of each individual
    Update the $BestFit$, $WorstFit$, and X_{best}.
    Calculate the Hun by Equation (7),
    Calculate WH_1 by Equation (5) (r_3 and r_4 are obtained by rand() function),
    Calculate WH_2 by Equation (6) (r_5 parameter is obtained by rand() function),
    for $eachindividual$ **do**
      Calculate E by Equation (2),
      Calculate RA by equation (4)(r_2 is obtained by rand() function),
      Update positions by Equation (1) (r_1 and r_2 are obtained by rand() function),
    t=t+1
return Best solutions: $BestFit$ and X_{best}

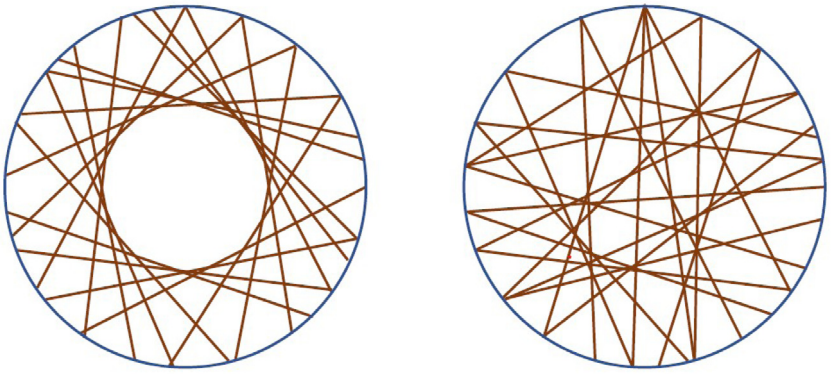

Fig. 1. Behaviors of ergodic and non-ergodic systems.

5. Chaotic hunger games search algorithm

By using chaotic maps instead of random values used in metaheuristic optimization algorithms, repetition and non-ergodic problems caused by randomness are controlled and this contributes to performance of algorithms. The HGS is an algorithm consisting of *approach food* and *hunger role* phases. The realization of these two phases is due to six different random values (named r1, r2, ...,r6).

In this study, ten different chaotic maps have been applied for six different random parameters of HGS. Here, r1 and r2, are expressed in Eq. (1) and affect the exploration phase. By controlling the chaotic maps of these values, it is aimed to provide a wide scanning of the solution space. These mapping processes have been tried separately for r1 and r2 and it has been observed that r2 exhibits a more stable behavior. For this reason, it was decided to control only the r2 parameter with chaotic maps in the exploration phase. In the exploitation phase, there are r3, r4, r5 and r6 parameters and these are used in Eqs. (5), (6) and (8). Similar to the exploration phase, chaotic mapping for all four parameters in this phase have been examined and it has been observed that r3 has a more stable structure than the others.

Table 2
Chaotic maps and their mathematical expression.

Name of the map	Mathematical expression
Chebyshev map	$x_{k+1} = cos(kcos^{-1}(x_k))$
Circle map	$x_{k+1} = mod(x_k + b - \frac{a}{2\pi} sin(2\pi x_k), 1)$
Gauss/mouse map	$x_{k+1} = \begin{cases} 1, & x_k = 0 \\ \frac{1}{mod(x_k, 1)}, & otherwise \end{cases}$
Iterative map	$x_{k+1} = sin(\frac{a\pi}{x_k})$
Logistic map	$x_{k+1} = ax_k(1 - x_k)$
Piecewise map	$x_{k+1} = \begin{cases} \frac{x_k}{P}, & 0 \le x_k < P \\ \frac{x_k - P}{0.5 - P}, & P \le x_k < 0.5 \\ \frac{1 - P - x_k}{0.5 - P}, & 0.5 \le x_k < 1 - P \\ \frac{1 - x_k}{P}, & 1 - P \le x_k < 1 \end{cases}$
Sine map	$x_{k+1} = \frac{a}{4} sin(\pi x_k)$
Singer map	$x_{k+1} = \mu(7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.302875x_k^4)$
Sinusoidal map	$x_{k+1} = ax_k^2 sin(\pi x_k)$
Tent map	$X_{k+1} = \begin{cases} \frac{x_k}{0.7}, & x_k < 0.7 \\ \frac{10}{3}(1 - x_k), & x_k \ge 0.7 \end{cases}$

Accordingly, in the next part of the proposed algorithm, the r2 and r3 random values in the classical HGS will be replaced by the values obtained from the chaotic maps. This process has been carried out under three different scenarios:

- **Scenario 1-R2**: The chaotic maps are applied only for r_2.
- **Scenario 2-R3**: The chaotic maps are applied only for r_3.
- **Scenario 3-R2_R3**: The chaotic maps are applied for r_2 and r_3.

The pseudo-code of the HGS algorithm with applied chaotic map is given in Algorithm 2 and r_2 and r_3, the values to which the chaotic map will be applied in the code are given. However, according to the scenarios mentioned above, r_2 and r_3 values can be obtained with rand() function or chaotic maps.

6. Experimental results

The proposed chaotic HGS has been applied to both benchmark functions and engineering problems. In this section, the performance and validity of the proposed method are discussed. The performances of all algorithms are obtained by calculating the average and standard deviation of fitness values after 30 individual runs and 500 epochs. All experiments have been performed on MATLAB R2020a computer with Intel (R) Core (TM) i7 CPU @ 2.40 GHz- 2.40 GHz, 8 GB RAM.

6.1. Application for benchmark problems

Benchmark problems are mathematical equations with a certain range and optimum values. The performance of the proposed chaotic HGS is evaluated for 23 well-known benchmark functions. These functions fall into two categories: Unimodal and multimodal benchmark functions. Unimodal benchmark functions have a single optima and they are the most suitable functions for comparing the exploitation phase of optimization algorithms.

Multimodal benchmark functions have multiple optima. Therefore, convergence of multimodal functions is more difficult than unimodal functions. Unimodal and multi-modal benchmark functions used in this study to measure

Table 3
Description of benchmark functions.

Unimodal Func.	Description	Range	Dim.	f_{min}
F1	$f(x) = \sum_{i=1}^{n} x_i^2$	[−100, 100]	30	0
F2	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \Pi_{i=1}^{n} \lvert x_i \rvert$	[−10, 10]	30	0
F3	$f(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	[−100, 100]	30	0
F4	$f(x) = \max_i (\lvert x_i \rvert, 1 \leq i \leq n)$	[−100, 100]	30	0
F5	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[−30, 30]	30	0
F6	$f(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	[−100, 100]	30	0
F7	$f(x) = \sum_{i=1}^{n} i x_i^4 + random[0, 1)$	[−1.28, 1.28]	30	0
F8	$f(x) = \sum_{i=1}^{n} -x_i sin(\sqrt{\lvert x_i \rvert})$	[−500, 500]	30	−418.982*dim
Multimodal Func.				
F9	$f(x) = \sum_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10]$	[−5.12, 5.12]	30	0
F10	$f(x) = -20exp(-0.2\sqrt{1/n \sum_{i=1}^{n} x_i^2})$ $-exp(1/n \sum_{i=1}^{n} cos(2\pi x_i)) + 20 + e$	[−32, 32]	30	0
F11	$f(x) = 1/4000 \sum_{i=1}^{n} x_i^2 - \Pi_{i=1}^{n} cos(x_i/\sqrt{i}) + 1$	[−600, 600]	30	0
F12	$f(x) = \pi/n\{10sin(\pi y_i) + \sum_{i=1}^{n-1} [(y_i - 1)^2]$ $[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	[−50, 50]	30	0
F13	$f(x) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2 [1 + sin^2(3\pi x_i + 1)]$ $+(x_n - 1)^2 [1 + sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	[−50, 50]	30	0
F14	$f(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{1 + \sum_{i=1}^{2}} (x_i - a_{ij}) \right)^{-1}$	[−65, 65]	2	1
F15	$f(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1 (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	[−5, 5]	4	0.0003
F16	$f(x) = 4x_1^2 - 2.1x_1^4 + (1/3)x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	[−5, 5]	2	−1.0316
F17	$f(x) = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) cos(x_1) + 10$	[−5, 5]	2	0.398
F18	$f(x) = \left[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)\right] \times$ $\left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_i + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)\right]$	[−2, 2]	2	3
F19	$f(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{3} a_{ij} (x_j - p_{ij})^2)$	[1, 3]	3	−3.86
F20	$f(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{6} a_{ij} (x_j - p_{ij})^2)$	[0, 1]	6	−3.32
F21	$f(x) = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	[0, 10]	4	−10.1532
F22	$f(x) = -\sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	[0, 10]	2	−10.4028
F23	$f(x) = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	[0, 10]	4	−10.5363

the performance of chaotic HGS are given in Table 3. To observe the effect of chaotic HGS, two different random vectors have been replaced with chaotic maps. The aim is to control the randomness in the exploitation and/or exploration phase with chaotic maps. The phase at which chaotic maps are more effective was determined as a result of benchmark problems. Chaotic HGS is compared with 12 different optimization algorithms for 3 different scenarios and 10 chaotic maps, including classical HGS. Firstly, we compared all maps under 3 Scenarios to compare chaotic hgs and classical HGS performance. Tables 10 and 11 show the case where vector r2 is controlled with chaotic maps. Considering Tables 10 and 11, the F1, F9, F11, F16, F17, and F19 functions converge to the same value as the classical HGS in all chaotic maps. On the other hand, chaotic HGS in all functions except F18 has better convergence ability than classical HGS in various chaotic maps. In terms of chaotic maps for Scenario 1, the Piecewise map has effective convergence behavior in 16 benchmark functions. Considering Scenario 2 (Tables 12 and 13), it can be seen that 3 different chaotic maps are effective on HGS. In Scenario 2, the Sine and Chebyshev map have effective convergence capability in 18 benchmark functions. Followed by sinusoidal with 16 benchmark functions. On the other hand, the Sinusoidal map produces effective results on 16 benchmark functions. Considering Scenario 3 (Tables 14 and 15), the Sine map has the ability to effectively converge for 18 benchmark functions. As a result of the comparison of the 3 Scenarios, although Scenario 2 and Scenario 3 show similar convergence

Algorithm 2 Pseudo-code of chaotic HGS algorithm

procedure CHAOTIC HGS
  Initialize the parameters:N, l, dim and $sHun$
  max. no of iter. $MaxIt$,
  pos. of solutions X_i, $i = 1, 2, 3, \ldots, N$.
  while $t <= MaxIt$ **do**
    Calculate the fitness of each individual
    Update the $BestFit$, $WorstFit$, and X_{best}.
    Calculate the Hun by Equation (7),
    Calculate WH_1 by Equation (5) (r_3 is related with chaotic maps and r_4 is obtained by rand() function),
    Calculate WH_2 by Equation (6) (r_5 parameter is obtained by rand() function),
    for $eachindividual$ **do**
      Calculate E by Equation (2),
      Calculate RA by Equation (4)(r_2 is related with chaotic maps),
      Update positions by equation (1) (r_2 is related with chaotic maps and r_1 is obtained by rand() function),
    t=t+1
return Best solutions: $BestFit$ and X_{best}

Table 4
CEC2017 benchmark function problems.

Type	No	Description	F_{min}
Unimodal Functions	1	Shifted and Rotated Bent Cigar Function	100
	3	Shifted and Rotated Zakharov Function	300
Simple Multimodal Functions	4	Shifted and Rotated Rosenbrock's Function	400
	5	Shifted and Rotated Rastrigin's Function	500
	6	Shifted and Rotated Expanded Scaffer's F6 Function	600
	7	Shifted and Rotated Lunacek Bi-Rastrigin Function	700
	8	Shifted and Rotated Non-Continuous Rastrigin's Function	800
	9	Shifted and Rotated Levy Function	900
	10	Shifted and Rotated Schwefel's Function	1000
Hybrid Functions	11	Hybrid Function 1 (N = 3)	1100
	12	Hybrid Function 2 (N = 3)	1200
	13	Hybrid Function 3 (N = 3)	1300
	14	Hybrid Function 4 (N = 4)	1400
	15	Hybrid Function 5 (N = 4)	1500
	16	Hybrid Function 6 (N = 4)	1600
	17	Hybrid Function 6 (N = 5)	1700
	18	Hybrid Function 6 (N = 5)	1800
	19	Hybrid Function 6 (N = 5)	1900
	20	Hybrid Function 6 (N = 6)	2000
Composition Functions	21	Composition Function 1 (N = 3)	2100
	22	Composition Function 2 (N = 3)	2200
	23	Composition Function 3 (N = 4)	2300
	24	Composition Function 4 (N = 4)	2400
	25	Composition Function 5 (N = 5)	2500
	26	Composition Function 6 (N = 5)	2600
	27	Composition Function 7 (N = 6)	2700
	28	Composition Function 8 (N = 6)	2800
	29	Composition Function 9 (N = 3)	2900
	30	Composition Function 10 (N = 3)	3000

Table 5
Parameter settings of encountered algorithms.

Algorithm	Parameter setting
ABC	$a = 1$
ALO	$k = 500$
BAT	$alpha = 0.5,\ gamma = 0.5,\ ro = 0.001$
DE	Scaling factor $= 0.5$; crossover probability $= 0.5$
GWO	$a = [2, 0]$
MFO	$b = 1,\ t = [-1, 1],\ a \in [-1, 2]$
MVO	Existence probability $\in [0.21]$; traveling distance rate $\in [0.61]$
PSO	$c_1 = 1.5; c_2 = 2; v_{Max} = 6$
SCA	$A = 2$
SSA	$c_1 \in [0, 1], c_2 \in [0, 1]$
WOA	$a_1 = [2, 0]; a_2 = [-2, -1]; b = 1$

performance, if all chaotic maps are taken into account, it can be seen that Scenario 2 shows the best convergence performance. Therefore, only Scenario 2 has been applied to the other experiments (CEC2017 and Engineering applications).

6.2. Comparison with state of art algorithms in the literature

The proposed chaotic HGS has been compared with the classical HGS, DE, ABC, MVO, WOA, ALO, GWO, MFO, BAT, SCA, SSA, and PSO methods in the literature. Table 5 shows the parameter settings of the compared algorithms. Fig. 4 shows comparison results of Chaotic HGS with other optimization algorithms for some benchmark functions, while Fig. 5 shows the 2D representation of benchmark functions. It can be seen that the chaotic HGS converges in early iterations in the F1, F2, F3 and F4 functions. In F5, F7, F9, F10, F11, F14 and F15 functions, chaotic HGS has more efficient convergence capability than other algorithms. In comparative studies, the Sine map was used in the chaotic HGS algorithm. Fig. 4 is discussed in more detail below.

The proposed chaotic HGS tends to converge to the global minimum in both unimodal and multimodal benchmark functions. When the unimodal benchmark functions are considered, the chaotic HGS for the F1 function converges to the global minimum (0) value in the early iteration. It has been observed that F2 converges to the global minimum value earlier than the F1 function in the first 100 iterations. For F3 and F4 functions, it has reached the global minimum value earlier than F1 and F2 in the first 10 iterations. For the F5 function, it can be said that Chaotic HGS is the algorithm with the best convergence among the compared methods. It has also reached the global minimum value earlier than other methods. For the F7 function, converges earlier than other optimization methods during the maximum number of iterations and gives the closest result to the global minimum value. Multimodal functions are known to be more challenging benchmark equations than unimodal functions. In our study, it is clear that the proposed chaotic HGS also has an effective convergence ability in multimodal functions. Considering Fig. 4, the F9 function converges to the global minimum value in the first 50 iterations. Similarly, in F11, both the Chaotic HGS and the original HGS algorithms converges in the early iteration. However, the Chaotic HGS has caught the global minimum in the first few iterations. In F10, a significant convergence was observed in Chaotic HGS compared to other methods. Although all algorithms exhibited close characters in F14 and F15, the proposed method converged to the global minimum value in the fastest way. Considering all the aforementioned comments , it has been proven that the proposed method gives favorable results in benchmark problems. Thus, it is considered to apply both engineering problems and challenging CEC benchmark problems. Besides, the proposed chaotic HGS converged to the global minimum without getting caught in the local minima trap.

6.3. Application for CEC2017 problems

To evaluate the performance of optimization algorithms, some challenging mathematical models are required. In IEEE Congress on Evolutionary Computation (CEC), such difficult problems produced by the scholars are available [79]. CEC2017 has 30 problems with varying difficulty and consists of 30 test functions. Table 4 shows the CEC 2017 benchmark problems. F1 and F3 are unimodal, F4–F10 are multimodal, F11–F20 are hybrid, and

F21–F30 are composition functions. F2 was omitted because it exhibits unstable behavior. The CEC2017 functions test the capabilities of the proposed optimization method in the exploitation and exploration phases. In CEC2017 problems, it can be seen that the Gauss/Mouse map converges better than the classical HGS in F1, F3, F4, F13–F16, F18 and F21–F28 functions. For F23 and F27, the chaotic HGS converges closer to the optimum result. For F23, HGS: 2624.94, Chaotic HGS: 2623.03 For F27, HGS: 3102.60, Chaotic HGS: 3100.09.(See Tables 16 and 17.)

7. Application for engineering problems

In this section, the proposed chaotic HGS is applied to cantilever beam design, tension/compression and speed reducer constrained engineering problems. These problems are evaluated under certain constraints. Other hand, there are individual limit values for each variable that needs to be optimized. It is aimed to minimize the cost functions by considering the constraints of the problems.

7.1. Cantilever beam design problem

This problem aims to minimize the total weight of the cantilever beam by optimizing the hollow square section parameters. The structure of the cantilever beam design problem is given in Fig. 2 with 3D, side and rear views. Its mathematical form can be described by Eq. (10):

$$\begin{aligned} &\underset{f(x)}{\text{minimize}} \quad f(x) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5) \\ &\text{subject to:} \quad g(x) = \frac{60}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \end{aligned} \tag{10}$$

where the variable range is given as: $0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100$

In the cantilever beam design problem, chaotic HGS has been compared with well-known optimization methods in the literature such as GCA_I, GCA_II, MMA, CS, SMA, MFO, SOS, AO, GOA, PSO, WOA, ECBO, and BOA. Table 6 shows the performance of chaotic HGS. It can be seen that the iterative chaotic map converges better with the 1.3012293 value than the other maps. Although the Iterative map provides the best convergence, in fact, other chaotic maps also showed effective convergence comparing to the existing studies in the literature.

7.2. Tension/compression spring design problem

The purpose of the tension/compression spring design problem is to minimize the total weight of the tension/compression spring by optimizing wire diameter (d), mean coil diameter (D), and the number of active coils (P) variables. The structure of the tension/compression spring design problem is given in Fig. 3. Its mathematical form can be described by Eq. (11).

$$\begin{aligned} &\underset{f(x)}{\text{minimize}} \quad f(x) = (x_3 + 2)x_2x_1^2 \\ &\text{subject to:} \\ &g_1(x) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \leq 0 \\ &g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_1^3 - x_1^4)} - \frac{1}{5108x_1^2} \leq 0 \\ &g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \\ &g_4(x) = 1 - \frac{x_1 + x_2}{1.5} \leq 0 \end{aligned} \tag{11}$$

where $x = [x_1, x_2, x_3] = [d, D, P]$ and $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$, $2 \leq x_3 \leq 15$

Table 7 shows the comparison of chaotic HGS with other optimization methods for the tension/compression spring design problem. In the Tension/compression spring design problem, we have compared chaotic HGS with

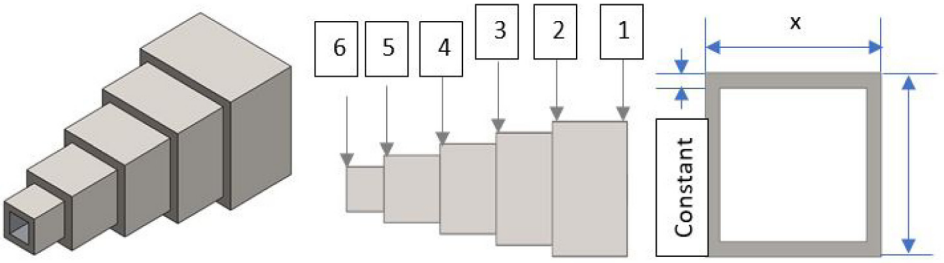

Fig. 2. The structure of cantilever beam design problem.

Table 6
Comparison of chaotic mapped HGS results with different algorithms for cantilever beam design problem.

Algorithm	Optimum variables					Optimum cost
	x1	x2	x3	x4	x5	
GCA_I [12]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA_II [12]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
MMA [12]	6.01s00	5.3000	4.4900	3.4900	2.1500	1.3400
CS [20]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
SMA [44]	6.0177	5.3108	4.4937	3.5011	2.1501	1.33996
MFO [51]	5.9830	5.3167	4.4973	3.5136	2.1616	1.33998
SOS [11]	6.0188	5.30344	4.49587	3.49896	2.1556	1.33996
AO [1]	5.8881	5.5451	4.3798	3.5973	2.1026	1.3390
GOA [36]	6.010043	4.795002	4.455917	3.529235	2.153681	1.306898
PSO [36]	6.033067	4.819485	4.524332	3.436107	2.131126	1.306913
WOA [36]	5.973752	4.862526	4.486167	3.487862	2.129218	1.306626
ECBO [36]	5.915208	4.908425	4.489401	3.478983	2.149223	1.306733
BOA [36]	5.969612	4.887111	4.461483	3.475294	2.145763	1.306610
Results for HGS with chaotic map functions						
Chebyshev Map	6.000212	4.860749	4.449184	3.464028	2.133206	1.301275264
Circle Map	5.945873	4.922445	4.471206	3.439338	2.129327	1.301325714
Gauss/Mouse Map	5.930081	4.844899	4.467362	3.512254	2.153045	1.301291608
Iterative Map	5.924605	4.883631	4.465333	3.489397	2.143675	1.301229362
Logistic Map	5.968981	4.893048	4.433932	3.487926	2.123315	1.301264201
Piecewise Map	5.942648	4.855852	4.513225	3.459382	2.136581	1.301294498
Sine Map	5.982856	4.877608	4.449687	3.480827	2.116192	1.301262218
Singer Map	5.962057	4.857028	4.433336	3.497702	2.157291	1.301277456
Sinusoidal Map	5.952092	4.845658	4.50668	3.488539	2.115007	1.301312341
Tent Map	5.931899	4.898078	4.496866	3.440374	2.140712	1.301309556
Classical HGS [80]	5.961096	4.863168	4.428171	3.514335	2.14092	1.301294685

well-known optimization methods in the literature such as CC, GA, HS, CSCA, PSO, CPSO, ES, RO, MVO, WOA, GSA and MVO. It can be seen that the Piecewise map gives the best results with 0.0126656. It is also clear that Logistic, Sine, and Chebyshev maps give better results than the classical HGS.

7.3. Speed reducer problem

The speed reducer problem is one of the benchmark problems in structural optimization. The design of the speed reducer is a more challenging benchmark, because it involves seven design variables. The design of the speed reducer is considered with the face width (x_1), the module of the teeth (x_2), the number of teeth on pinion (x_3), the length of the first shaft between bearings (x_4), the length of the second shaft between bearings (x_5), the diameter of the first shaft (x_6), and the diameter of the second shaft (x_7). The mathematical form of the speed reducer is given

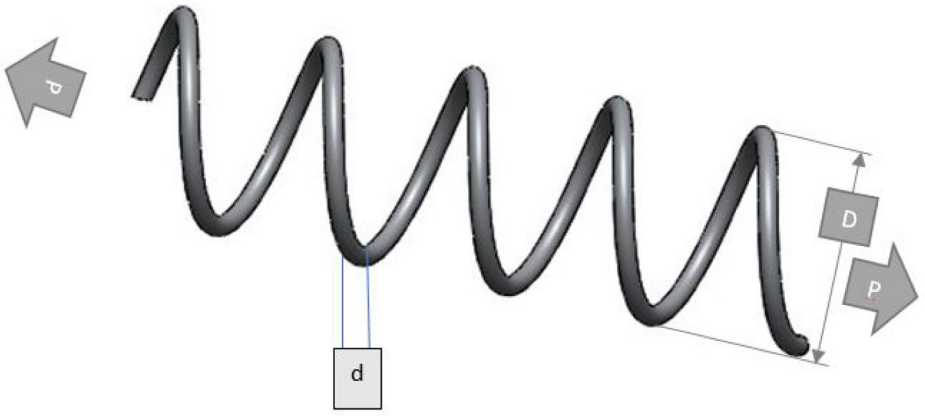

Fig. 3. The structure of tension/compression spring design problem.

Table 7
Comparison of chaotic mapped HGS results with different algorithms for tension/compression spring design problem.

Algorithm	Optimum variables			Optimum cost
	d	D	N	
CC [4]	0.05000	0.315900	14.250000	0.0128334
GA [13]	0.05148	0.351661	11.632201	0.01270478
HS [49]	0.05115	0.349871	12.076432	0.0126706
CSCA [28]	0.05160	0.354714	11.410831	0.0126702
PSO [24]	0.05172	0.357644	11.244543	0.0126747
CPSO [24]	0.05172	0.357644	11.244543	0.0126747
ES [50]	0.05164	0.355360	14.250000	0.012698
RO [37]	0.05137	0.349096	14.250000	0.0126788
WOA [55]	0.05120	0.345215	14.250000	0.0126763
GSA [56]	0.05027	0.323680	14.250000	0.0127022
MVO [56]	0.05251	0.376020	14.250000	0.01279
Results for HGS with chaotic map functions				
Chebyshev Map	0.052872665	0.385830368	9.775697135	0.012667298
Circle Map	0.050332894	0.324958185	13.42643709	0.01271927
Gauss/Mouse Map	0.05	0.315735166	14.26115576	0.012835547
Iterative Map	0.05	0.31740722	14.03018257	0.012720239
Logistic Map	0.05148265	0.351771769	11.58505509	0.012666137
Piecewise Map	0.051831888	0.360163516	11.08977362	0.012665604
Sine Map	0.051820211	0.359879738	11.10620335	0.012665801
Singer Map	0.05	0.317425029	14.02782106	0.012719079
Sinusoidal Map	0.052703537	0.381371116	10.00324399	0.012715281
Tent Map	0.053335765	0.397641618	9.239157438	0.012713427
Classical HGS	0.05	0.317425383	14.02777412	0.012719056

in Eq. (12).

$$\underset{f(x)}{\text{minimize}} \quad f(x) = \frac{27}{x_1 x_2^2 x_3} - 1 - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^2 + x_7^2)$$

subject to:

$$g_1(x) = \frac{27}{x_1 x_2^2 x_3} - 1 \leq 0$$

Table 8
Comparison of chaotic mapped HGS results with different algorithms for speed reducer problem.

Algorithm	Optimum variables							Optimum cost
	x1	x2	x3	x4	x5	x6	x7	
PSO [34]	3.5638	0.7010	17.0972	7.5893	7.9881	3.3998	5.3871	3127.7366
GA [34]	3.5658	0.7029	17.0905	7.5890	7.9429	3.4402	5.3142	3098.7993
CS [34]	3.5000	0.7000	17.0000	7.3009	7.7162	3.3503	5.2867	2994.5432
GSA [34]	3.5342	0.7009	17.8168	7.3978	8.2455	3.4921	5.4292	3301.5843
ABC [34]	3.5042	0.7000	17.0000	7.3557	8.2310	3.3526	5.2883	3009.6425
SHO [16]	3.50159	0.7000	17.0000	7.3000	7.8000	3.3513	5.2887	2998.5507
GWO [16]	3.506690	0.7000	17.0000	7.3809	7.8157	3.3578	5.2868	3001.2880
MFO [16]	3.5075	0.7000	17.0000	7.3024	7.8024	3.3235	5.2875	3009.5710
MVO [16]	3.5088	0.7000	17.0000	7.3928	7.8160	3.3581	5.2868	3002.9280
SCA [16]	3.5088	0.7000	17.0000	7.3000	7.8000	3.4610	5.2892	3030.5630
HS [16]	3.5201	0.7000	17.0000	8.3700	7.8000	3.3670	5.2887	3029.0020
Results for HGS with chaotic map functions								
Chebyshev Map	3.5	0.7	17	7.3	7.8	3.343372427	5.285352186	2993.761765
Circle Map	3.5	0.7	17	7.3	7.8	3.343372427	5.285352186	2993.761765
Gauss/Mouse Map	3.500021336	0.7	17	7.3	7.8	3.343373353	5.285365226	2993.778671
Iterative Map	3.500006905	0.7	17	7.3	7.800001334	3.343372508	5.28535224	2993.764562
Logistic Map	3.5	0.7	17	7.3	7.8	3.343372427	5.285352186	2993.761765
Piecewise Map	3.5	0.7	17	7.3	7.8	3.343372427	5.285352186	2993.761765
Sine Map	3.5	0.7	17	7.3	7.8	3.343372427	5.285352186	2993.761765
Singer Map	3.5	0.7	17	7.3	7.8	3.343372427	5.285352186	2993.761765
Sinusoidal Map	3.5	0.7	17	7.3	7.8	3.343372427	5.285352186	2993.761765
Tent Map	3.5	0.7	17	7.3	7.8	3.343372427	5.285352186	2993.761765
HGS	3.5	0.7	17	7.3	7.8	3.343372427	5.285352186	2993.761765

$$
\begin{aligned}
g_2(x) &= \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0 \\
g_3(x) &= \frac{1.93 x_4^3}{x_2 x_3 x_6^4} - 1 \le 0 \\
g_4(x) &= \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \le 0 \\
g_5(x) &= \frac{\sqrt{(\frac{745 x_4}{x_2 x_3})^2 + 16.9 \times 10^6}}{110 x_6^3} - 1 \le 0 \\
g_6(x) &= \frac{\sqrt{(\frac{745 x_4}{x_2 x_3})^2 + 157.5 \times 10^6}}{85 x_6^3} - 1 \le 0 \\
g_7(x) &= \frac{x_2 x_3}{40} - 1 \le 0 \\
g_8(x) &= \frac{5 x_2}{x_1} - 1 \le 0 \\
g_9(x) &= \frac{x_1}{12 x_2} - 1 \le 0 \\
g_{10}(x) &= \frac{1.5 x_6 + 1.9}{x_4} - 1 \le 0 \\
g_{11}(x) &= \frac{1.1 x_7 + 1.9}{x_5} - 1 \le 0
\end{aligned}
\tag{12}
$$

where $2.6 \le x_1 \le 3.6$, $0.7 \le x_2 \le 0.8$, $17 \le x_3 \le 28$, $7.3 \le x_4 \le 8.3$, $7.8 \le x_5 \le 8.3$, $2.9 \le x_6 \le 3.9$, $5 \le x_7 \le 5.5$

Table 9
Wilcoxon sign rank test results between classical HGS and chaotic HGS based on Scenario 2 for benchmark problems.

	Chebyshev		Circle		Gauss/Mouse		Iterative		Logistic	
F	h	p	h	p	h	p	h	p	h	p
F1	0	9.59E−01	0	3.09E−01	1	6.98E−06	0	4.53E−01	1	2.60E−06
F2	0	9.26E−01	1	1.48E−02	1	3.52E−06	0	2.37E−01	1	1.73E−06
F3	1	3.72E−05	1	6.64E−04	0	1.53E−01	1	4.07E−05	0	4.05E−01
F4	0	3.39E−01	0	1.92E−01	1	7.69E−06	0	3.93E−01	1	1.73E−06
F5	1	1.73E−06	1	4.29E−06	1	1.60E−04	1	1.92E−06	0	7.86E−02
F6	1	9.84E−03	0	9.78E−02	1	1.49E−05	1	4.20E−04	1	2.37E−05
F7	0	4.91E−01	0	6.27E−02	1	5.22E−06	0	5.30E−01	1	1.92E−06
F8	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06
F9	0	6.87E−02	0	6.29E−01	1	4.73E−06	0	2.62E−01	1	5.79E−05
F10	0	3.39E−01	0	2.37E−01	1	5.22E−06	0	5.86E−01	1	1.73E−06
F11	0	6.58E−01	0	1.65E−01	1	5.22E−06	0	7.34E−01	1	1.73E−06
F12	0	8.29E−01	1	4.28E−02	1	4.29E−06	0	7.52E−02	1	1.73E−06
F13	0	5.71E−02	0	8.13E−01	1	6.34E−06	1	4.95E−02	1	1.73E−06
F14	1	1.64E−05	1	2.05E−04	1	1.36E−05	1	1.73E−06	1	2.13E−06
F15	0	2.37E−01	1	6.32E−05	1	3.18E−06	1	1.89E−04	1	1.73E−06
F16	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06
F17	1	9.63E−04	1	4.11E−03	1	7.69E−06	1	3.11E−05	1	1.92E−06
F18	1	1.73E−06	1	2.22E−04	1	1.36E−05	1	1.73E−06	1	1.06E−04
F19	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06
F20	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06
F21	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06
F22	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06
F23	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06
(+ |=| −)	(13–0–10)		(15–0–8)		(22–0–1)		(15–0–8)		(21–0–2)	
	Piecewise		Sine		Singer		Sinusoidal		Tent	
F	h	p	h	p	h	p	h	p	h	p
F1	1	1.11E−02	0	9.10E−01	1	1.73E−06	1	1.85E−02	0	6.14E−01
F2	1	3.38E−03	0	5.58E−01	1	1.73E−06	0	6.27E−02	0	8.45E−01
F3	1	1.85E−02	1	8.47E−06	1	1.73E−06	1	6.42E−03	1	9.32E−06
F4	1	4.28E−02	0	6.44E−01	1	1.73E−06	0	1.31E−01	0	2.45E−01
F5	1	8.31E−04	1	1.92E−06	1	1.73E−06	1	1.40E−02	1	1.92E−06
F6	0	7.04E−01	1	2.41E−04	1	1.73E−06	0	7.50E−01	1	6.64E−04
F7	1	7.73E−03	0	5.58E−01	1	1.73E−06	0	9.37E−02	1	9.84E−03
F8	1	1.73E−06	1	1.73E−06	0	2.71E−01	1	1.73E−06	1	1.73E−06
F9	0	5.04E−01	0	1.41E−01	1	1.73E−06	0	4.28E−01	1	3.00E−02
F10	1	1.40E−02	0	8.77E−01	1	1.73E−06	0	6.56E−02	0	1.71E−01
F11	1	1.40E−02	0	7.81E−01	1	1.73E−06	0	8.59E−02	0	1.25E−01
F12	1	1.66E−02	0	5.98E−02	1	1.73E−06	0	1.06E−01	0	9.43E−01
F13	0	4.91E−01	1	2.96E−03	1	1.73E−06	0	1.71E−01	1	2.58E−03
F14	0	4.53E−01	1	1.73E−06	1	1.73E−06	0	8.61E−01	1	1.36E−04
F15	1	4.45E−05	1	2.05E−04	1	1.73E−06	1	1.48E−02	0	3.49E−01
F16	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06
F17	0	7.81E−01	1	1.73E−06	1	1.73E−06	0	8.77E−01	1	3.59E−04
F18	0	1.78E−01	1	1.73E−06	1	1.73E−06	0	5.04E−01	1	2.35E−06
F19	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06
F20	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06
F21	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06
F22	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06
F23	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06	1	1.73E−06
(+ |=| −)	(17–0–6)		(15–0–8)		(22–0–1)		(11–0–12)		(16–0–7)	

In the speed reducer problem, we compared chaotic HGS with well-known optimization methods in the literature such as PSO, GA, CS, GSA, ABC, SHO, GWO, MFO, MVO, SCA and HS. Table 8 shows the results for the

Table 10
Results for Benchmark functions based on r2 parameter determined by chaotic map functions 1–5.

Functions	Chebyshev Map		Circle Map		Gauss/Mouse Map		Iterative Map		Logistic Map		Classical HGS	
	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	5.03E-213	0.00E+00	1.88E-238	0.00E+00	0.00E+00	0.00E+00	8.48E-224	0.00E+00	5.45E-208	0.00E+00	1.18E-221	0.00E+00
F3	4.71E-268	0.00E+00	2.43E-308	0.00E+00	1.78E-250	0.00E+00	1.09E-263	0.00E+00	2.07E-279	0.00E+00	8.82E-249	0.00E+00
F4	1.43E-194	0.00E+00	1.59E-226	0.00E+00	4.43E-147	2.43E-146	5.03E-212	0.00E+00	1.49E-192	0.00E+00	2.31E-199	0.00E+00
F5	9.70E+00	1.13E+01	1.66E+01	1.10E+01	8.12E+00	1.09E+01	1.91E+01	8.71E+00	9.58E+00	1.11E+01	1.31E+01	1.16E+01
F6	1.42E-07	2.80E-07	2.47E-07	4.46E-07	9.73E-06	2.09E-05	1.19E-07	1.47E-07	1.79E-07	6.39E-07	6.53E-07	1.82E-06
F7	2.15E-04	2.84E-04	1.56E-04	2.20E-04	3.39E-04	5.12E-04	2.61E-04	3.27E-04	2.88E-04	5.96E-04	1.82E-04	2.17E-04
F8	-1.26E+04	5.90E-02	-1.26E+04	1.95E-01	-1.26E+04	3.02E-02	-1.26E+04	9.66E-02	-1.26E+04	2.86E-02	-1.26E+04	2.57E-02
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	1.21E-09	1.51E-09	1.19E-10	8.14E-11	7.03E-07	1.15E-06	6.27E-10	1.07E-09	6.73E-10	5.24E-10	5.15E-10	5.08E-10
F13	1.43E-08	1.36E-08	1.91E-09	1.53E-09	6.16E-06	1.91E-05	1.56E-02	8.53E-02	1.30E-08	1.33E-08	8.45E-09	7.86E-09
F14	2.95E+00	3.97E+00	2.17E+00	3.56E+00	1.32E+00	1.78E+00	5.02E+00	5.41E+00	1.65E+00	2.48E+00	1.97E+00	2.98E+00
F15	5.61E-04	3.20E-04	5.08E-04	2.08E-04	8.90E-04	3.41E-04	4.14E-04	1.75E-04	3.74E-04	1.29E-04	6.52E-04	3.00E-04
F16	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16	-1.03E+00	1.07E-05	-1.03E+00	6.71E-16	-1.03E+00	6.71E-16	-1.03E+00	6.78E-16
F17	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	2.90E-05	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00
F18	1.74E+01	1.37E+01	1.02E+01	1.21E+01	9.30E+00	1.16E+01	1.47E+01	1.36E+01	7.50E+00	1.02E+01	3.00E+00	2.96E-15
F19	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15	-3.86E+00	5.85E-05	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15
F20	-3.29E+00	5.83E-02	-3.24E+00	7.91E-02	-3.26E+00	7.65E-02	-3.27E+00	6.60E-02	-3.27E+00	6.72E-02	-3.25E+00	7.52E-02
F21	-1.02E+01	6.45E-15	-1.02E+01	6.39E-15	-9.98E+00	9.31E-01	-1.02E+01	6.02E-15	-1.02E+01	6.33E-15	-1.02E+01	6.68E-15
F22	-1.01E+01	1.73E+00	-1.02E+01	9.70E-01	-1.04E+01	1.38E-04	-9.45E+00	2.90E+00	-1.04E+01	8.73E-16	-1.04E+01	8.73E-16
F23	-1.05E+01	1.78E-15	-1.05E+01	1.87E-15	-1.05E+01	1.25E-04	-1.05E+01	2.82E-15	-1.05E+01	1.36E-15	-1.05E+01	1.48E-15

Table 11
Results for Benchmark functions based on r2 parameter determined by chaotic map functions 6–10.

Functions	Piecewise Map		Sine Map		Singer Map		Sinusoidal Map		Tent Map		Classical HGS	
	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	2.92E-245	0.00E+00	5.64E-217	0.00E+00	3.85E-205	0.00E+00	4.72E-229	0.00E+00	3.33E-233	0.00E+00	1.18E-221	0.00E+00
F3	2.80E-295	0.00E+00	3.35E-265	0.00E+00	4.18E-283	0.00E+00	0.00E+00	0.00E+00	2.18E-302	0.00E+00	8.82E-249	0.00E+00
F4	1.14E-213	0.00E+00	2.95E-194	0.00E+00	1.78E-206	0.00E+00	8.48E-217	0.00E+00	7.30E-211	0.00E+00	2.31E-199	0.00E+00
F5	1.06E+01	1.16E+01	9.64E+00	1.12E+01	1.26E+01	1.20E+01	1.82E+01	1.41E+01	1.15E+01	1.17E+01	1.31E+01	1.16E+01
F6	2.18E-07	5.46E-07	1.89E-07	3.91E-07	2.15E-07	3.99E-07	3.33E-07	5.03E-07	2.78E-07	9.14E-07	6.53E-07	1.82E-06
F7	1.61E-04	2.24E-04	2.62E-04	3.16E-04	1.25E-04	1.59E-04	2.61E-04	4.41E-04	2.87E-04	5.68E-04	1.82E-04	2.17E-04
F8	-1.25E+04	2.83E+02	-1.26E+04	2.54E-02	-1.26E+04	1.01E-01	-1.25E+04	4.78E+02	-1.26E+04	4.74E-02	-1.26E+04	2.57E-02
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	1.01E-15	6.49E-16	8.88E-16	0.00E+00	8.88E-16	0.00E+00
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	2.04E-10	1.65E-10	8.42E-10	1.32E-09	1.84E-09	2.19E-09	7.80E-09	1.01E-08	3.08E-10	3.29E-10	5.15E-10	5.08E-10
F13	3.06E-09	3.11E-09	1.13E-08	1.29E-08	4.08E-08	5.04E-08	1.72E-07	2.09E-07	3.33E-03	1.83E-02	8.45E-09	7.86E-09
F14	3.28E+00	4.20E+00	1.97E+00	2.98E+00	4.76E+00	5.43E+00	6.71E+00	5.83E+00	2.04E+00	3.19E+00	1.97E+00	2.98E+00
F15	4.20E-04	1.62E-04	4.95E-04	2.45E-04	5.32E-04	3.75E-04	4.30E-04	2.52E-04	4.24E-04	2.06E-04	6.52E-04	3.00E-04
F16	-1.03E+00	6.71E-16	-1.03E+00	6.65E-16	-1.03E+00	6.78E-16	-1.03E+00	5.83E-16	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16
F17	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00
F18	1.11E+01	1.26E+01	1.29E+01	1.32E+01	1.47E+01	1.36E+01	1.65E+01	1.37E+01	1.92E+01	1.35E+01	3.00E+00	2.96E-15
F19	-3.86E+00	2.68E-15	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15	-3.86E+00	2.50E-15	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15
F20	-3.27E+00	8.62E-02	-3.25E+00	8.76E-02	-3.26E+00	7.03E-02	-3.26E+00	8.09E-02	-3.26E+00	8.07E-02	-3.25E+00	7.52E-02
F21	-1.02E+01	6.35E-15	-9.98E+00	9.31E-01	-9.37E+00	2.49E+00	-1.02E+01	5.39E-15	-1.02E+01	6.28E-15	-1.02E+01	6.68E-15
F22	-1.04E+01	1.19E-15	-9.91E+00	1.96E+00	-1.04E+01	9.33E-16	-1.04E+01	3.03E-10	-1.04E+01	9.33E-16	-1.04E+01	8.73E-16
F23	-1.02E+01	1.75E+00	-1.05E+01	1.28E-15	-1.02E+01	1.75E+00	-1.05E+01	6.41E-12	-1.04E+01	9.87E-01	-1.05E+01	1.48E-15

speed reducer problem. Except for the Gauss/Mouse and Iterative maps, all maps converge to similar results with 2993.761765. It has been observed that the Iterative and Gauss/Mouse maps reach lower costs compared to the studies in the literature and converged to the 2993.764562 and 2993.778671 values, respectively. Ultimately, HGS methods converge better than other methods in the literature.

8. Wilcoxon signed rank-test

The chaotic HGS has yielded promising results on many benchmark and engineering problems. However, it should be tested whether there is a significant difference among the proposed method and other methods in the literature. In this study, the difference between chaotic HGS and classical HGS has been tested. We have performed the Wilcoxon signed-rank test which is used in many studies. If the p-value generated by the comparison is less than 0.05, then the algorithm in the comparison has a clear significance. Otherwise, there is no significant difference between the two methods. The (+ , = , -) signs in the test results indicate that the proposed algorithm is significantly better, worse, and similar, respectively. In Table 9, Wilcoxon sign rank test results between classical HGS and chaotic HGS are presented. Statistically, the Sinusoidal map produces the closest results to classical HGS. For the Singer and Gauss/Mouse maps, chaotic HGS and classical HGS differ by 0.05 significance level.

Table 12
Results for Benchmark functions based on r3 parameter determined by chaotic map functions 1–5.

Functions	Chebyshev Map		Circle Map		Gauss/Mouse Map		Iterative Map		Logistic Map		Classical HGS	
	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	0.00E+00	0.00E+00	1.46E-245	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.18E-221	0.00E+00
F3	0.00E+00	0.00E+00	3.38E-163	2.22E-162	0.00E+00	0.00E+00	2.00E-323	0.00E+00	0.00E+00	0.00E+00	8.82E-249	0.00E+00
F4	5.45E-235	0.00E+00	2.23E-200	0.00E+00	0.00E+00	0.00E+00	7.42E-220	0.00E+00	2.65E-228	0.00E+00	2.31E-199	0.00E+00
F5	9.27E+00	1.15E+01	1.56E+01	1.12E+01	1.80E+01	1.11E+01	1.69E+01	1.04E+01	1.40E+01	1.16E+01	1.31E+01	1.16E+01
F6	3.47E-11	6.70E-11	1.31E-16	1.52E-16	4.04E-06	4.93E-06	2.21E-13	2.32E-13	1.36E-13	2.01E-13	6.53E-07	1.82E-06
F7	9.53E-05	9.86E-05	2.18E-04	2.70E-04	5.11E-05	4.16E-05	9.32E-05	1.07E-04	1.32E-04	2.55E-04	1.82E-04	2.17E-04
F8	-1.26E+04	1.23E-05	-1.26E+04	2.06E-09	-1.26E+04	1.64E-01	-1.26E+04	1.55E-07	-1.26E+04	8.77E-09	-1.26E+04	2.57E-02
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	9.64E-13	8.59E-13	2.83E-17	7.37E-17	1.05E-07	1.18E-07	1.67E-14	1.37E-14	1.21E-14	1.49E-14	5.15E-10	5.08E-10
F13	1.44E-11	1.67E-11	1.49E-16	1.78E-16	1.34E-06	1.48E-06	3.38E-13	3.59E-13	8.72E-14	7.84E-14	8.45E-09	7.86E-09
F14	1.32E+00	1.78E+00	1.65E+00	2.48E+00	1.32E+00	1.78E+00	1.32E+00	1.78E+00	1.97E+00	2.98E+00	1.97E+00	2.98E+00
F15	3.52E-04	1.79E-04	4.85E-04	2.35E-04	4.34E-04	3.16E-04	4.82E-04	3.24E-04	4.41E-04	2.81E-04	6.52E-04	3.00E-04
F16	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16	-1.03E+00	6.48E-10	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16
F17	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	1.94E-10	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00
F18	3.00E+00	4.31E-15	3.00E+00	1.79E-15	3.00E+00	1.32E-08	3.00E+00	2.01E-15	3.00E+00	3.19E-15	3.00E+00	2.96E-15
F19	-3.86E+00	2.67E-15	-3.86E+00	2.71E-15	-3.86E+00	6.37E-07	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15
F20	-3.26E+00	6.38E-02	-3.28E+00	6.44E-02	-3.26E+00	7.07E-02	-3.27E+00	6.30E-02	-3.27E+00	6.60E-02	-3.25E+00	7.52E-02
F21	-9.98E+00	9.31E-01	-1.02E+01	6.86E-15	-1.02E+01	2.53E-05	-1.02E+01	6.16E-13	-1.02E+01	6.74E-15	-1.02E+01	6.68E-15
F22	-1.04E+01	7.38E-16	-1.04E+01	1.09E-15	-1.04E+01	9.63E-05	-1.04E+01	1.19E-15	-1.04E+01	8.08E-16	-1.04E+01	8.73E-16
F23	-1.05E+01	1.28E-15	-1.04E+01	9.87E-01	-1.05E+01	8.13E-05	-1.05E+01	1.32E-15	-1.05E+01	2.21E-15	-1.05E+01	1.48E-15

Table 13
Results for Benchmark functions based on r3 parameter determined by chaotic map functions 1–5.

Functions	Piecewise Map		Sine Map		Singer Map		Sinusoidal Map		Tent Map		Classical HGS	
	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	5.82E-211	0.00E+00	0.00E+00	0.00E+00	2.36E-229	0.00E+00	1.02E-190	0.00E+00	1.36E-228	0.00E+00	1.18E-221	0.00E+00
F3	5.74E-151	3.05E-150	0.00E+00	0.00E+00	1.66E-217	0.00E+00	7.39E-118	4.00E-117	9.43E-162	5.14E-161	8.82E-249	0.00E+00
F4	3.55E-200	0.00E+00	8.11E-244	0.00E+00	6.22E-194	0.00E+00	1.48E-182	0.00E+00	1.16E-193	0.00E+00	2.31E-199	0.00E+00
F5	1.25E+01	1.19E+01	1.17E+01	1.19E+01	9.41E+00	1.17E+01	1.33E+01	1.18E+01	1.26E+01	1.20E+01	1.31E+01	1.16E+01
F6	1.13E-16	2.69E-16	2.40E-13	3.70E-13	1.27E-16	2.42E-16	1.25E-18	4.93E-18	5.70E-18	8.48E-18	6.53E-07	1.82E-06
F7	2.18E-04	3.16E-04	1.03E-04	1.29E-04	1.96E-04	2.71E-04	2.67E-04	3.09E-04	2.20E-04	2.80E-04	1.82E-04	2.17E-04
F8	-1.26E+04	4.54E-06	-1.26E+04	5.27E-08	-1.26E+04	4.19E-05	-1.26E+04	4.84E-07	-1.26E+04	1.65E-09	-1.26E+04	2.57E-02
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	1.13E-17	1.39E-17	1.39E-14	2.19E-14	1.47E-17	2.42E-17	3.08E-19	1.03E-18	1.82E-18	3.04E-18	5.15E-10	5.08E-10
F13	6.40E-17	8.72E-17	3.25E-03	1.78E-02	3.39E-16	6.36E-16	2.06E-18	4.51E-18	2.11E-17	5.47E-17	8.45E-09	7.86E-09
F14	9.98E-01	1.75E-16	9.98E-01	9.22E-17	1.32E+00	1.78E+00	9.98E-01	1.17E-16	9.98E-01	1.24E-16	1.97E+00	2.98E+00
F15	4.86E-04	2.95E-04	4.43E-04	3.18E-04	4.71E-04	3.33E-04	6.19E-04	2.72E-04	4.76E-04	2.69E-04	6.52E-04	3.00E-04
F16	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16	-1.03E+00	6.71E-16	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16
F17	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00
F18	3.00E+00	2.75E-15	3.00E+00	2.85E-15	3.00E+00	2.66E-15	3.00E+00	2.36E-15	3.00E+00	2.44E-15	3.00E+00	2.96E-15
F19	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15
F20	-3.26E+00	6.72E-02	-3.28E+00	6.78E-02	-3.26E+00	7.32E-02	-3.28E+00	5.70E-02	-3.27E+00	6.73E-02	-3.25E+00	7.52E-02
F21	-9.81E+00	1.29E+00	-1.02E+01	6.36E-15	-1.02E+01	7.01E-15	-1.02E+01	6.90E-15	-1.02E+01	6.96E-15	-1.02E+01	6.68E-15
F22	-1.02E+01	9.70E-01	-1.04E+01	8.73E-16	-1.04E+01	1.14E-15	-1.04E+01	1.04E-15	-1.04E+01	5.56E-10	-1.04E+01	8.73E-16
F23	-1.05E+01	1.19E-15	-1.05E+01	5.99E-14	-1.05E+01	2.53E-15	-1.05E+01	1.44E-15	-1.04E+01	9.87E-01	-1.05E+01	1.48E-15

Table 14
Results for Benchmark functions based on r2 and r3 parameters determined by chaotic map functions 1–5.

Functions	Chebyshev Map		Circle Map		Gauss/Mouse Map		Iterative Map		Logistic Map		Classical HGS	
	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	0.00E+00	0.00E+00	1.46E-245	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.18E-221	0.00E+00
F3	0.00E+00	0.00E+00	3.38E-163	2.22E-162	0.00E+00	0.00E+00	2.00E-323	0.00E+00	0.00E+00	0.00E+00	8.82E-249	0.00E+00
F4	5.45E-235	0.00E+00	2.23E-200	0.00E+00	0.00E+00	0.00E+00	7.42E-220	0.00E+00	2.65E-228	0.00E+00	2.31E-199	0.00E+00
F5	9.27E+00	1.15E+01	1.56E+01	1.12E+01	1.80E+01	1.11E+01	1.69E+01	1.04E+01	1.40E+01	1.16E+01	1.31E+01	1.16E+01
F6	3.47E-11	6.70E-11	1.31E-16	1.52E-16	4.04E-06	4.93E-06	2.21E-13	2.32E-13	1.36E-13	2.01E-13	6.53E-07	1.82E-06
F7	9.53E-05	9.86E-05	2.18E-04	2.70E-04	5.11E-05	4.16E-05	9.32E-05	1.07E-04	1.32E-04	2.55E-04	1.82E-04	2.17E-04
F8	-1.26E+04	1.23E-05	-1.26E+04	2.06E-09	-1.26E+04	1.64E-01	-1.26E+04	1.55E-07	-1.26E+04	8.77E-09	-1.26E+04	2.57E-02
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	9.64E-13	8.59E-13	2.83E-17	7.37E-17	1.05E-07	1.18E-07	1.67E-14	1.37E-14	1.21E-14	1.49E-14	5.15E-10	5.08E-10
F13	1.44E-11	1.67E-11	1.49E-16	1.78E-16	1.34E-06	1.48E-06	3.38E-13	3.59E-13	8.72E-14	7.84E-14	8.45E-09	7.86E-09
F14	1.32E+00	1.78E+00	1.65E+00	2.48E+00	1.32E+00	1.78E+00	1.32E+00	1.78E+00	1.97E+00	2.98E+00	1.97E+00	2.98E+00
F15	3.52E-04	1.79E-04	4.85E-04	2.35E-04	4.34E-04	3.16E-04	4.82E-04	3.24E-04	4.41E-04	2.81E-04	6.52E-04	3.00E-04
F16	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16	-1.03E+00	6.48E-10	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16
F17	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	1.94E-10	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00
F18	3.00E+00	4.31E-15	3.00E+00	1.79E-15	3.00E+00	1.32E-08	3.00E+00	2.01E-15	3.00E+00	3.19E-15	3.00E+00	2.96E-15
F19	-3.86E+00	2.67E-15	-3.86E+00	2.71E-15	-3.86E+00	6.37E-07	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15
F20	-3.26E+00	6.38E-02	-3.28E+00	6.44E-02	-3.26E+00	7.07E-02	-3.27E+00	6.30E-02	-3.27E+00	6.60E-02	-3.25E+00	7.52E-02
F21	-9.98E+00	9.31E-01	-1.02E+01	6.86E-15	-1.02E+01	2.53E-05	-1.02E+01	6.16E-13	-1.02E+01	6.74E-15	-1.02E+01	6.68E-15
F22	-1.04E+01	7.38E-16	-1.04E+01	1.09E-15	-1.04E+01	9.63E-05	-1.04E+01	1.19E-15	-1.04E+01	8.08E-16	-1.04E+01	8.73E-16
F23	-1.05E+01	1.28E-15	-1.04E+01	9.87E-01	-1.05E+01	8.13E-05	-1.05E+01	1.32E-15	-1.05E+01	2.21E-15	-1.05E+01	1.48E-15

Table 15
Results for Benchmark functions based on r2 and r3 parameters determined by chaotic map functions 6–10.

Functions	Piecewise Map		Sine Map		Singer Map		Sinusoidal Map		Tent Map		Classical HGS	
	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	5.82E-211	0.00E+00	0.00E+00	0.00E+00	2.36E-229	0.00E+00	1.02E-190	0.00E+00	1.36E-228	0.00E+00	1.18E-221	0.00E+00
F3	5.74E-151	3.05E-150	0.00E+00	0.00E+00	1.66E-217	0.00E+00	7.39E-118	4.00E-117	9.43E-162	5.14E-161	8.82E-249	0.00E+00
F4	3.55E-200	0.00E+00	8.11E-244	0.00E+00	6.22E-194	0.00E+00	1.48E-182	0.00E+00	1.16E-193	0.00E+00	2.31E-199	0.00E+00
F5	1.25E+01	1.19E+01	1.17E+01	1.19E+01	9.41E+00	1.17E+01	1.33E+01	1.18E+01	1.26E+01	1.20E+01	1.31E+01	1.16E+01
F6	1.13E-16	2.69E-16	2.40E-13	3.70E-13	1.27E-16	2.42E-16	1.25E-18	4.93E-18	5.70E-18	8.48E-18	6.53E-07	1.82E-06
F7	2.18E-04	3.16E-04	1.03E-04	1.29E-04	1.96E-04	2.71E-04	2.67E-04	3.09E-04	2.20E-04	2.80E-04	1.82E-04	2.17E-04
F8	-1.26E+04	4.54E-06	-1.26E+04	5.27E-08	-1.26E+04	4.19E-05	-1.26E+04	4.84E-07	-1.26E+04	1.65E-09	-1.26E+04	2.57E-02
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	1.13E-17	1.39E-17	1.39E-14	2.19E-14	1.47E-17	2.42E-17	3.08E-19	1.03E-18	1.82E-18	3.04E-18	5.15E-10	5.08E-10
F13	6.40E-17	8.72E-17	3.25E-03	1.78E-02	3.39E-16	6.36E-16	2.06E-18	4.51E-18	2.11E-17	5.47E-17	8.45E-09	7.86E-09
F14	9.98E-01	1.75E-16	9.98E-01	9.22E-17	1.32E+00	1.78E+00	9.98E-01	1.17E-16	9.98E-01	1.24E-16	1.97E+00	2.98E+00
F15	4.86E-04	2.95E-04	4.43E-04	3.18E-04	4.71E-04	3.33E-04	6.19E-04	2.72E-04	4.76E-04	2.69E-04	6.52E-04	3.00E-04
F16	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16	-1.03E+00	6.71E-16	-1.03E+00	6.78E-16	-1.03E+00	6.78E-16
F17	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00
F18	3.00E+00	2.75E-15	3.00E+00	2.85E-15	3.00E+00	2.66E-15	3.00E+00	2.36E-15	3.00E+00	2.44E-15	3.00E+00	2.96E-15
F19	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15	-3.86E+00	2.71E-15
F20	-3.26E+00	6.72E-02	-3.28E+00	6.78E-02	-3.26E+00	7.32E-02	-3.28E+00	5.70E-02	-3.27E+00	6.73E-02	-3.25E+00	7.52E-02
F21	-9.81E+00	1.29E+00	-1.02E+01	6.36E-15	-1.02E+01	7.01E-15	-1.02E+01	6.90E-15	-1.02E+01	6.96E-15	-1.02E+01	6.68E-15
F22	-1.02E+01	9.70E-01	-1.04E+01	8.73E-16	-1.04E+01	1.14E-15	-1.04E+01	1.04E-15	-1.04E+01	5.56E-10	-1.04E+01	8.73E-16
F23	-1.05E+01	1.19E-15	-1.05E+01	5.99E-14	-1.05E+01	2.53E-15	-1.05E+01	1.44E-15	-1.04E+01	9.87E-01	-1.05E+01	1.48E-15

Table 16
Results for CEC2017 functions based on r3 parameter determined by chaotic map functions 1–5.

Functions	Chebyshev Map		Circle Map		Gauss/Mouse Map		Iterative Map		Logistic Map		Classical HGS	
	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD
F1	6.04E+03	4.23E+03	7.62E+03	4.46E+03	1.80E+05	3.04E+05	6.95E+03	4.97E+03	5.97E+03	4.49E+03	6.65E+03	4.46E+03
F3	3.01E+02	6.98E+00	3.00E+02	4.47E-04	3.55E+02	1.50E+02	3.00E+02	1.34E-06	3.01E+02	3.05E+00	3.00E+02	2.10E-05
F4	4.07E+02	1.23E+01	4.14E+02	2.34E+01	4.17E+02	2.61E+01	4.13E+02	2.32E+01	4.08E+02	1.36E+01	4.09E+02	2.01E+01
F5	5.29E+02	1.04E+01	5.30E+02	1.11E+01	5.23E+02	8.24E+00	5.29E+02	1.07E+01	5.29E+02	9.17E+00	5.20E+02	8.74E+00
F6	6.03E+02	3.04E+00	6.04E+02	3.57E+00	6.03E+02	2.63E+00	6.02E+02	3.02E+00	6.03E+02	3.86E+00	6.01E+02	9.06E-01
F7	7.33E+02	1.16E+01	7.39E+02	1.31E+01	7.43E+02	1.45E+01	7.36E+02	1.06E+01	7.40E+02	1.08E+01	7.30E+02	7.05E+00
F8	8.27E+02	8.98E+00	8.27E+02	9.20E+00	8.22E+02	7.79E+00	8.22E+02	8.89E+00	8.25E+02	8.13E+00	8.20E+02	6.63E+00
F9	9.27E+02	1.04E+02	9.19E+02	2.87E+01	9.14E+02	3.17E+01	9.14E+02	5.04E+01	9.15E+02	4.84E+01	9.01E+02	6.78E+00
F10	1.68E+03	2.70E+02	1.78E+03	3.03E+02	1.65E+03	2.42E+02	1.69E+03	2.76E+02	1.65E+03	2.39E+02	1.59E+03	2.40E+02
F11	1.14E+03	5.49E+01	1.15E+03	5.71E+01	1.14E+03	2.37E+01	1.13E+03	4.04E+01	1.14E+03	3.63E+01	1.13E+03	1.91E+01
F12	1.80E+05	5.01E+05	9.47E+04	2.74E+05	1.04E+06	2.14E+06	1.37E+05	3.69E+05	4.78E+04	1.69E+05	2.22E+04	1.87E+04
F13	1.17E+04	1.22E+04	1.37E+04	1.13E+04	1.32E+04	1.16E+04	1.07E+04	1.06E+04	1.29E+04	1.09E+04	1.04E+04	1.08E+04
F14	1.92E+03	9.88E+02	2.52E+03	2.14E+03	1.87E+03	1.06E+03	2.13E+03	1.49E+03	2.55E+03	1.58E+03	1.90E+03	7.26E+02
F15	2.74E+03	1.16E+03	3.86E+03	1.89E+03	1.94E+03	6.65E+02	2.59E+03	1.25E+03	2.93E+03	1.46E+03	3.98E+03	2.38E+03
F16	1.72E+03	1.08E+02	1.75E+03	1.17E+02	1.72E+03	1.07E+02	1.73E+03	1.02E+02	1.73E+03	1.18E+02	1.75E+03	1.23E+02
F17	1.77E+03	4.56E+01	1.78E+03	4.87E+01	1.77E+03	4.88E+01	1.77E+03	4.74E+01	1.77E+03	5.50E+01	1.75E+03	4.92E+01
F18	2.30E+04	1.34E+04	2.24E+04	1.57E+04	2.18E+04	1.57E+04	2.72E+04	1.38E+04	2.40E+04	1.37E+04	2.21E+04	1.29E+04
F19	1.22E+04	1.08E+04	9.49E+03	9.39E+03	1.25E+04	1.03E+04	1.14E+04	1.20E+04	1.50E+04	1.24E+04	8.76E+03	7.23E+03
F20	2.04E+03	2.25E+01	2.04E+03	3.09E+01	2.04E+03	2.86E+01	2.04E+03	1.45E+01	2.05E+03	3.97E+01	2.02E+03	2.31E+01
F21	2.32E+03	4.08E+01	2.32E+03	5.24E+01	2.29E+03	5.66E+01	2.30E+03	5.69E+01	2.31E+03	5.58E+01	2.32E+03	4.13E+01
F22	2.31E+03	7.46E+01	2.35E+03	2.09E+02	2.30E+03	1.13E+01	2.32E+03	1.27E+02	2.30E+03	9.66E+00	2.36E+03	2.28E+02
F23	2.63E+03	7.29E+00	2.63E+03	7.76E+00	2.62E+03	8.45E+00	2.63E+03	9.99E+00	2.63E+03	1.08E+01	2.62E+03	7.25E+00
F24	2.76E+03	5.05E+01	2.76E+03	5.10E+01	2.74E+03	6.63E+01	2.75E+03	6.90E+01	2.74E+03	8.21E+01	2.77E+03	1.30E+01
F25	2.94E+03	3.58E+01	2.94E+03	2.28E+01	2.92E+03	2.39E+01	2.94E+03	2.98E+01	2.94E+03	2.96E+01	2.93E+03	2.44E+01
F26	3.09E+03	2.51E+02	3.09E+03	3.70E+02	3.05E+03	2.82E+02	3.17E+03	3.77E+02	3.06E+03	2.43E+02	3.17E+03	4.06E+02
F27	3.10E+03	1.59E+01	3.10E+03	2.09E+01	3.10E+03	1.25E+01	3.10E+03	1.68E+01	3.10E+03	1.86E+01	3.10E+03	2.44E+01
F28	3.28E+03	1.22E+02	3.31E+03	1.17E+02	3.29E+03	1.18E+02	3.30E+03	1.93E+02	3.29E+03	1.17E+02	3.31E+03	1.45E+02
F29	3.22E+03	4.96E+01	3.24E+03	6.41E+01	3.21E+03	5.93E+01	3.25E+03	6.28E+01	3.24E+03	7.20E+01	3.21E+03	4.27E+01
F30	2.25E+05	3.86E+05	4.06E+05	4.75E+05	2.21E+05	3.65E+05	3.01E+05	4.18E+05	2.77E+05	3.76E+05	1.26E+05	2.82E+05

9. Conclusion and future works

Real-world problems can be represented by nonlinear and complex systems. These systems can be expressed more clearly with chaotic maps. Therefore, chaotic maps gain better convergence ability when they are used with metaheuristic algorithms. In metaheuristic algorithms, the balance between the exploitation and exploration phase plays an important role in the effective convergence of the algorithm. In this study, randomness in both exploitation and exploration phases is controlled with chaotic maps. There are two different random variables (r_2, r_3) that significantly affect these two phases, and their control with chaotic maps has been examined with three different scenarios.

In future studies, the proposed chaotic HGS method can be used in the nonlinear coefficient inverse problem for partial differential equations [9,32,46]. In particular, the constraints can be handled with differential equations and effective solutions can be produced with the relevant fitness function [76]. The proposed chaotic HGS algorithm can be applied to various engineering problems such as 3-bar truss design problem, Pressure vessel engineering problem,

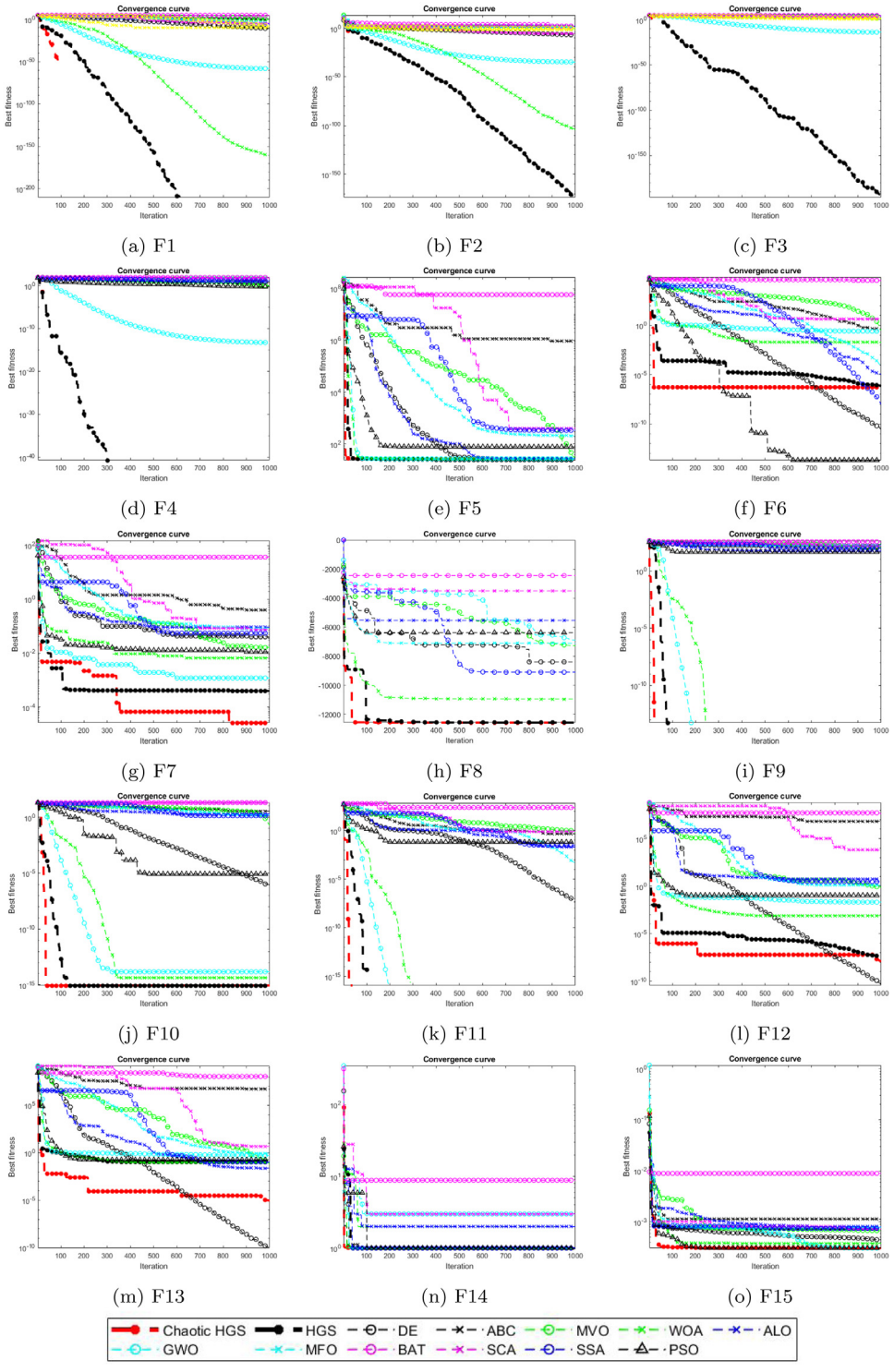

Fig. 4. Comparison with other state of the art optimization algorithms on benchmark functions.

Welded beam design, Gear train design and can be compared with other studies [15]. Also, different chaotic maps such as Henon, Ikeda, Liebovitch and Zaslavskii can be applied on HGS [75].

Table 17
Results for CEC2017 functions based on r3 parameter determined by chaotic map functions 6–10.

Functions	Piecewise Map		Sine Map		Singer Map		Sinusoidal Map		Tent Map		Classical HGS	
	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD	Avg	SD
F1	7.51E+03	3.97E+03	7.77E+03	4.34E+03	6.55E+03	4.19E+03	7.02E+03	4.41E+03	7.25E+03	4.63E+03	6.65E+03	4.46E+03
F3	3.00E+02	1.17E-04	3.00E+02	6.90E-05	3.00E+02	8.63E-08	3.00E+02	4.87E-04	3.00E+02	3.37E-04	3.00E+02	2.10E-05
F4	4.11E+02	2.32E+01	4.10E+02	1.97E+01	4.08E+02	1.50E+01	4.19E+02	2.88E+01	4.16E+02	2.81E+01	4.09E+02	2.01E+01
F5	5.31E+02	1.18E+01	5.28E+02	1.13E+01	5.30E+02	1.10E+01	5.32E+02	1.23E+01	5.34E+02	8.94E+00	5.20E+02	8.74E+00
F6	6.04E+02	4.19E+00	6.02E+02	2.63E+00	6.02E+02	3.01E+00	6.07E+02	6.53E+00	6.05E+02	4.81E+00	6.01E+02	9.06E-01
F7	7.40E+02	1.21E+01	7.36E+02	1.19E+01	7.35E+02	9.02E+00	7.38E+02	9.05E+00	7.37E+02	1.28E+01	7.30E+02	7.05E+00
F8	8.28E+02	1.03E+01	8.26E+02	9.76E+00	8.26E+02	8.98E+00	8.30E+02	1.26E+01	8.29E+02	1.12E+01	8.20E+02	6.63E+00
F9	9.42E+02	1.05E+02	9.06E+02	1.92E+01	9.06E+02	1.88E+01	9.95E+02	2.89E+02	9.46E+02	1.13E+02	9.01E+02	6.78E+00
F10	1.72E+03	2.33E+02	1.69E+03	2.73E+02	1.78E+03	2.48E+02	1.79E+03	3.19E+02	1.78E+03	2.90E+02	1.59E+03	2.40E+02
F11	1.16E+03	8.41E+01	1.17E+03	7.42E+01	1.15E+03	7.29E+01	1.15E+03	6.19E+01	1.15E+03	7.38E+01	1.13E+03	1.91E+01
F12	5.01E+04	8.94E+04	1.62E+05	7.62E+05	9.15E+04	2.76E+05	2.96E+05	1.08E+06	5.66E+05	1.35E+06	2.22E+04	1.87E+04
F13	1.09E+04	1.27E+04	1.23E+04	1.17E+04	1.03E+04	1.06E+04	1.01E+04	9.53E+03	1.17E+04	1.11E+04	1.04E+04	1.08E+04
F14	2.45E+03	2.13E+03	2.50E+03	2.97E+03	2.73E+03	2.69E+03	3.34E+03	3.22E+03	3.57E+03	4.54E+03	1.90E+03	7.26E+02
F15	3.37E+03	1.80E+03	2.65E+03	1.21E+03	4.67E+03	2.53E+03	4.75E+03	3.19E+03	3.34E+03	1.78E+03	3.98E+03	2.38E+03
F16	1.75E+03	1.37E+02	1.74E+03	1.19E+02	1.72E+03	1.10E+02	1.84E+03	1.88E+02	1.73E+03	1.39E+02	1.75E+03	1.23E+02
F17	1.78E+03	5.03E+01	1.77E+03	4.31E+01	1.77E+03	4.29E+01	1.80E+03	5.08E+01	1.78E+03	5.04E+01	1.75E+03	4.92E+01
F18	2.49E+04	1.67E+04	2.38E+04	1.45E+04	2.03E+04	1.43E+04	2.51E+04	1.77E+04	2.73E+04	1.88E+04	2.21E+04	1.29E+04
F19	1.09E+04	1.08E+04	1.30E+04	1.20E+04	1.17E+04	1.20E+04	1.15E+04	1.06E+04	1.16E+04	1.07E+04	8.76E+03	7.23E+03
F20	2.06E+03	4.42E+01	2.04E+03	2.49E+01	2.06E+03	4.90E+01	2.05E+03	2.64E+01	2.08E+03	5.20E+01	2.02E+03	2.31E+01
F21	2.31E+03	5.74E+01	2.30E+03	6.31E+01	2.32E+03	4.35E+01	2.33E+03	3.78E+01	2.31E+03	5.06E+01	2.32E+03	4.13E+01
F22	2.30E+03	1.55E+01	2.30E+03	1.45E+01	2.31E+03	4.77E+00	2.31E+03	4.94E+01	2.31E+03	1.23E+01	2.36E+03	2.28E+02
F23	2.63E+03	1.08E+01	2.63E+03	1.08E+01	2.63E+03	9.55E+00	2.63E+03	1.44E+01	2.63E+03	9.85E+00	2.62E+03	7.25E+00
F24	2.77E+03	1.44E+01	2.74E+03	7.32E+01	2.77E+03	1.17E+01	2.74E+03	9.20E+01	2.74E+03	8.27E+01	2.77E+03	1.30E+01
F25	2.94E+03	2.73E+01	2.94E+03	3.11E+01	2.95E+03	3.95E+01	2.93E+03	6.90E+01	2.94E+03	2.36E+01	2.93E+03	2.44E+01
F26	3.13E+03	3.38E+02	3.22E+03	4.65E+02	3.17E+03	4.25E+02	3.17E+03	3.19E+02	3.11E+03	3.04E+02	3.17E+03	4.06E+02
F27	3.10E+03	2.22E+01	3.10E+03	1.51E+01	3.10E+03	1.66E+01	3.10E+03	1.73E+01	3.11E+03	2.89E+01	3.10E+03	2.44E+01
F28	3.32E+03	1.37E+02	3.31E+03	1.18E+02	3.35E+03	1.33E+02	3.34E+03	1.56E+02	3.32E+03	1.44E+02	3.31E+03	1.45E+02
F29	3.25E+03	7.35E+01	3.22E+03	6.99E+01	3.24E+03	7.16E+01	3.25E+03	7.04E+01	3.26E+03	8.24E+01	3.21E+03	4.27E+01
F30	2.89E+05	4.14E+05	2.56E+05	4.05E+05	3.87E+05	4.96E+05	3.56E+05	3.92E+05	3.60E+05	4.35E+05	1.26E+05	2.82E+05

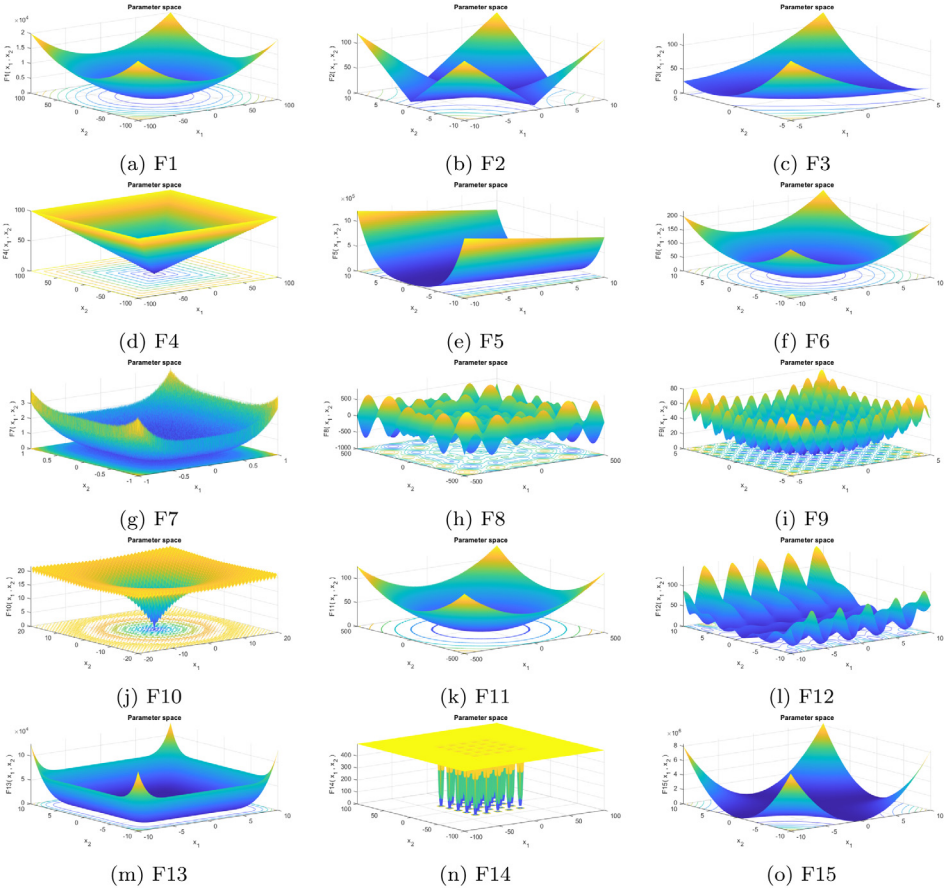

Fig. 5. Convergence behaviors and 3D representations of test functions.

Declaration of competing 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.

Acknowledgments

The authors are extremely thankful to the editor and the reviewers for their valuable suggestions and comments, which helped improving the manuscript.

References

[1] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Engrg. 376 (2021) 113609.
[2] H.S. Alamri, Y.A. Alsariera, K.Z. Zamli, Opposition-based whale optimization algorithm, Adv. Sci. Lett. 24 (10) (2018) 7461–7464.
[3] B. Alatas, Chaotic bee colony algorithms for global numerical optimization, Expert Syst. Appl. 37 (8) (2010) 5682–5687.
[4] J.S. Arora, Introduction to Optimum Design, Elsevier, 2004.
[5] Q. Askari, I. Younas, M. Saeed, Political optimizer: A novel socio-inspired meta-heuristic for global optimization, Knowl.-Based Syst. 195 (2020) 105709.
[6] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm, Comput. Struct. 169 (2016) 1–12.
[7] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, Ieee, 2007, pp. 4661–4667.
[8] S. Barshandeh, M. Haghzadeh, A new hybrid chaotic atom search optimization based on tree-seed algorithm and levy flight for solving optimization problems, Eng. Comput. (2020) 1–44.
[9] L. Beilina, M.V. Klibanov, A globally convergent numerical method for a coefficient inverse problem, SIAM J. Sci. Comput. 31 (1) (2008) 478–509.
[10] C.J. Burnett, C. Li, E. Webber, E. Tsaousidou, S.Y. Xue, J.C. Brüning, M.J. Krashes, Hunger-driven motivational state competition, Neuron 92 (1) (2016) 187–201.
[11] M.Y. Cheng, D. Prayogo, Symbiotic organisms search: A new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112.
[12] H. Chickermane, H.C. Gea, Structural optimization using a new local approximation method, Internat. J. Numer. Methods Engrg. 39 (5) (1996) 829–846.
[13] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2) (2000) 113–127.
[14] S. Dhargupta, M. Ghosh, S. Mirjalili, R. Sarkar, Selective opposition based grey wolf optimization, Expert Syst. Appl. 151 (2020) 113389.
[15] D. Dhawale, V.K. Kamboj, P. Anand, An effective solution to numerical and multi-disciplinary design optimization problems using chaotic slime mold algorithm, Eng. Comput. (2021) 1–39.
[16] G. Dhiman, V. Kumar, Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications, Adv. Eng. Softw. 114 (2017) 48–70.
[17] W. Du, M. Zhang, W. Ying, M. Perc, K. Tang, X. Cao, D. Wu, The networked evolutionary algorithm: A network science perspective, Appl. Math. Comput. 338 (2018) 33–43.
[18] I. Fister, A. Iglesias, A. Galvez, J. Del Ser, E. Osaba, I. Fister Jr, M. Perc, M. Slavinec, Novelty search for global optimization, Appl. Math. Comput. 347 (2019) 865–881.
[19] L.J. Fogel, A.J. Owens, M.J. Walsh, Artificial intelligence through simulated evolution, 1966.
[20] A.H. Gandomi, X.S. Yang, A.H. Alavi, Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (1) (2013) 17–35.
[21] S. Gupta, K. Deep, Hybrid sine cosine artificial bee colony algorithm for global optimization and image segmentation, Neural Comput. Appl. 32 (13) (2020) 9521–9543.
[22] S. Gupta, K. Deep, H. Moayedi, L.K. Foong, A. Assad, Sine cosine grey wolf optimizer to solve engineering design problems, Eng. Comput. (2020) 1–27.
[23] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, Inform. Sci. 222 (2013) 175–184.
[24] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (1) (2007) 89–99.
[25] S.L. Ho, S. Yang, G. Ni, J. Huang, A quantum-based particle swarm optimization algorithm applied to inverse problems, IEEE Trans. Magn. 49 (5) (2013) 2069–2072.
[26] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73.
[27] L. Hongwei, L. Jianyong, C. Liang, B. Jingbo, S. Yangyang, L. Kai, Chaos-enhanced moth-flame optimization algorithm for global optimization, J. Syst. Eng. Electron. 30 (6) (2019) 1144–1159.
[28] F.z. Huang, L. Wang, Q. He, An effective co-evolutionary differential evolution for constrained optimization, Appl. Math. Comput. 186 (1) (2007) 340–356.
[29] A.A. Hudaib, H.N. Fakhouri, Supernova optimizer: A novel natural inspired meta-heuristic, Mod. Appl. Sci. 12 (1) (2018) 32–50.

[30] A. Ibrahim, H.A. Ali, M.M. Eid, E.-S.M. El-kenawy, Chaotic harris hawks optimization for unconstrained function optimization, in: 2020 16th International Computer Engineering Conference, ICENCO, IEEE, 2020, pp. 153–158.
[31] Z. Juan, G. Zheng Ming, B.L. Jia, The improved slime mould algorithm with piecewice map, in: 2020 International Symposium on Computer Engineering and Intelligent Communications, ISCEIC, IEEE, 2020, pp. 25–29.
[32] B. Kaltenbacher, W. Rundell, The inverse problem of reconstructing reaction–diffusion systems, Inverse Problems 36 (6) (2020) 065011.
[33] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471.
[34] H. Karami, M.V. Anaraki, S. Farzin, S. Mirjalili, Flow direction algorithm (FDA): A novel optimization approach for solving optimization problems, Comput. Ind. Eng. 156 (2021) 107224.
[35] G. Kaur, S. Arora, Chaotic whale optimization algorithm, J. Comput. Des. Eng. 5 (3) (2018) 275–284.
[36] A. Kaveh, M. Khanzadi, M.R. Moghaddam, Billiards-inspired optimization algorithm; A new meta-heuristic method, in: Structures, 27, Elsevier, 2020, pp. 1722–1739.
[37] A. Kaveh, M. Khayatazad, A new meta-heuristic method: ray optimization, Comput. Struct. 112 (2012) 283–294.
[38] A. Kaveh, R.M. Moghanni, S. Javadi, Chaotic optimization algorithm for performance-based optimization design of composite moment frames, Eng. Comput. (2021) 1–13.
[39] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.
[40] S. Kirkpatrick, Optimization by simulated annealing: Quantitative studies, J. Stat. Phys. 34 (5) (1984) 975–986.
[41] M. Kohli, S. Arora, Chaotic grey wolf optimization algorithm for constrained optimization problems, J. Comput. Des. Eng. 5 (4) (2018) 458–472.
[42] J.R. Koza, J.R. Koza, Genetic Programming: On the Programming of Computers By Means of Natural Selection, Vol. 1, MIT Press, 1992.
[43] M. Kumar, A.J. Kulkarni, S.C. Satapathy, Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology, Future Gener. Comput. Syst. 81 (2018) 252–272.
[44] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, Future Gener. Comput. Syst. 111 (2020) 300–323.
[45] Y. Ling, Y. Zhou, Q. Luo, Lévy flight trajectory-based whale optimization algorithm for global optimization, IEEE Access 5 (2017) 6168–6186.
[46] D. Lukyanenko, A. Borzunov, M.A. Shishlenin, Solving coefficient inverse problems for nonlinear singularly perturbed equations of the reaction-diffusion-advection type with data on the position of a reaction front, Commun. Nonlinear Sci. Numer. Simul. 99 (2021) 105824.
[47] J. Luo, B. Shi, A hybrid whale optimization algorithm based on modified differential evolution for global optimization problems, Appl. Intell. 49 (5) (2019) 1982–2000.
[48] M.M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, Neurocomputing 260 (2017) 302–312.
[49] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Appl. Math. Comput. 188 (2) (2007) 1567–1579.
[50] E. Mezura-Montes, C.A.C. Coello, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems, Int. J. Gen. Syst. 37 (4) (2008) 443–473.
[51] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.
[52] S. Mirjalili, Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (4) (2016) 1053–1073.
[53] S. Mirjalili, SCA: A sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.
[54] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.
[55] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[56] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: A nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2) (2016) 495–513.
[57] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[58] H. Moayedi, H. Nguyen, L.K. Foong, Nonlinear evolutionary swarm intelligence of grasshopper optimization algorithm and gray wolf optimization for weight adjustment of neural network, Eng. Comput. (2019) 1–11.
[59] S.S. Rao, Engineering Optimization: Theory and Practice, John Wiley & Sons, 2019.
[60] R.V. Rao, V.J. Savsani, D. Vakharia, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (3) (2011) 303–315.
[61] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: A gravitational search algorithm, Inform. Sci. 179 (13) (2009) 2232–2248.
[62] L.A. Real, Animal choice behavior and the evolution of cognitive architecture, Science 253 (5023) (1991) 980–986.
[63] R.M. Rizk-Allah, A quantum-based sine cosine algorithm for solving general systems of nonlinear equations, Artif. Intell. Rev. (2021) 1–52.
[64] R.M. Rizk-Allah, A.E. Hassanien, S. Bhattacharyya, Chaotic crow search algorithm for fractional optimization problems, Appl. Soft Comput. 71 (2018) 1161–1175.
[65] G. Rudolph, Evolution strategies, Evol. Comput. 1 (2000) 81–88.
[66] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, Adv. Eng. Softw. 105 (2017) 30–47.

[67] G.I. Sayed, A. Darwish, A.E. Hassanien, A new chaotic multi-verse optimization algorithm for solving engineering optimization problems, J. Exp. Theor. Artif. Intell. 30 (2) (2018) 293–317.
[68] G.I. Sayed, G. Khoriba, M.H. Haggag, A novel chaotic salp swarm algorithm for global optimization and feature selection, Appl. Intell. 48 (10) (2018) 3462–3481.
[69] G.I. Sayed, A. Tharwat, A.E. Hassanien, Chaotic dragonfly algorithm: An improved metaheuristic algorithm for feature selection, Appl. Intell. 49 (1) (2019) 188–205.
[70] M.A. Shaheen, H.M. Hasanien, A. Alkuhayli, A novel hybrid GWO-PSO optimization technique for optimal reactive power dispatch problem solution, Ain Shams Eng. J. 12 (1) (2021) 621–630.
[71] H. Sharma, J.C. Bansal, K. Arya, X.-S. Yang, Lévy flight artificial bee colony algorithm, Internat. J. Systems Sci. 47 (11) (2016) 2652–2670.
[72] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713.
[73] R. Storn, K. Price, Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[74] S. Talatahari, B.F. Azar, R. Sheikholeslami, A. Gandomi, Imperialist competitive algorithm combined with chaos for global optimization, Commun. Nonlinear Sci. Numer. Simul. 17 (3) (2012) 1312–1319.
[75] S. Talatahari, A. Kaveh, R. Sheikholeslami, An efficient charged system search using chaos, Iran Univ. Sci. Technol. 1 (2) (2011) 305–325.
[76] T. van Leeuwen, F.J. Herrmann, A penalty method for PDE-constrained optimization in inverse problems, Inverse Problems 32 (1) (2015) 015007.
[77] G.G. Wang, S. Deb, A.H. Gandomi, Z. Zhang, A.H. Alavi, Chaotic cuckoo search, Soft Comput. 20 (9) (2016) 3349–3362.
[78] G.G. Wang, L. Guo, A.H. Gandomi, G.S. Hao, H. Wang, Chaotic krill herd algorithm, Inform. Sci. 274 (2014) 17–34.
[79] G. Wu, R. Mallipeddi, P. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization, Tech. Rep. (2016).
[80] Y. Yang, H. Chen, A.A. Heidari, A.H. Gandomi, Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts, Expert Syst. Appl. 177 (2021) 114864.
[81] X.S. Yang, S. Deb, Cuckoo search via Lévy flights, in: 2009 World Congress on Nature & Biologically Inspired Computing, NaBIC, IEEE, 2009, pp. 210–214.
[82] G.P. Yang, S.Y. Liu, J.K. Zhang, Q.X. Feng, Control and synchronization of chaotic systems by an improved biogeography-based optimization algorithm, Appl. Intell. 39 (1) (2013) 132–143.
[83] B.S. Yıldız, N. Pholdee, N. Panagant, S. Bureerat, A.R. Yildiz, S.M. Sait, A novel chaotic henry gas solubility optimization algorithm for solving real-world engineering problems, Eng. Comput. (2021) 1–13.
[84] X. Yuan, Y. Yuan, Y. Zhang, A hybrid chaotic genetic algorithm for short-term hydro system scheduling, Math. Comput. Simulation 59 (4) (2002) 319–327.
[85] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[86] J. Zhang, M. Xiao, L. Gao, Q. Pan, Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems, Appl. Math. Model. 63 (2018) 464–490.
[87] J. Zhao, Z.M. Gao, The chaotic slime mould algorithm with Chebyshev map, in: J. Phys.: Conf. Ser., 1631, (1) IOP Publishing, 2020, 012071.