



Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch



A modified seahorse optimization algorithm based on chaotic maps for solving global optimization and engineering problems

Feyza Altunbey Özbay¹

Firat University, Faculty of Engineering, Software Engineering, Elazig, Turkey



ARTICLE INFO

Article history:

Received 27 December 2022

Revised 13 March 2023

Accepted 28 March 2023

Available online 10 April 2023

Keywords:

Chaotic map

Global optimization

Metaheuristics

Nature-inspired optimization

Seahorse optimization

ABSTRACT

Metaheuristic optimization algorithms are global optimization approaches that manage the search process to efficiently explore search spaces associated with different optimization problems. Seahorse optimization (SHO) is a novel swarm-based metaheuristic optimization method inspired by certain behaviors of sea horses. The SHO algorithm mimics the movement, hunting, and breeding behavior of sea horses in nature. Chaotic maps are effectively used to improve the performance of metaheuristic algorithms by avoiding the local optimum and increasing the speed of convergence. In this study, 10 different chaotic maps have been employed for the first time to produce chaotic values rather than random values in SHO, increasing the performance of the method. The purpose of using chaotic maps that generate chaotic values to their random values in SHO is to increase the convergence speed of the original SHO algorithm and avoid the local optimum. 33 different benchmark functions, consisting of unimodal, multimodal, fixed-dimension multimodal, and CEC2019, have been utilized to assess the performance of Chaotic SHO (CSHO), which is first introduced in this study. In addition, the proposed CSHO has been compared with four metaheuristic algorithms in the literature, namely Sine Cosine Algorithm, Salp Swarm Algorithm, Whale Optimization Algorithm, and Particle Swarm Optimization. Statistical analyses of the obtained results have been also performed. The proposed CSHO is then implemented to 4 different real-world engineering design problems, including the welded beam, pressure vessel, tension/compression spring, and speed reducer. The results obtained with CSHO are compared with popular metaheuristic methods in the literature. Experimental results show that it gives successful and promising results compared to the original SHO algorithm.

© 2023 Karabuk University. Publishing services by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

In recent years, the complexity of real-world problems has been increasing with the development of society. Today, there are many optimization problems in different fields such as engineering, computer science, and medicine. This situation has led to the need to develop effective and reliable optimization methods [1]. The creation of a mathematical model is mostly difficult for complex systems. Even if the model is created, the high cost and long solution time prevent its usage [2]. Recently, methods inspired by nature for the creation of mathematical models of optimization problems are very popular. The limitations of classical optimization techniques in handling large-scale combinatorial and nonlinear problems have required alternative approaches. Due to the benefits listed below, metaheuristic optimization algorithms are now widely used because they are effective at tackling a wide range

of optimization issues: i) They are simple to implement and rely on straightforward notions, ii) They do not need gradient information, iii) They are capable of avoiding local optima., iv) They have the flexibility to be applied to different optimization problems without the need for special changes [3].

Metaheuristic approaches are widely used for solving many different optimization problems due to their high performance and low computational complexity. These algorithms are mostly stochastic and produce near-optimal solutions in a reasonable period for large-scale optimization problems with complex mathematical models [4]. In recent years, metaheuristic optimization algorithms have become very popular for solving optimization problems in different engineering applications. When the literature is reviewed, some of the applications are: Jiao et al. presented the application of these methods for the optimization of a photovoltaic system [5], Altunbey Özbay and Alatas adapted these algorithms to detect fake news on social media [6], Huang et al. have proposed a method using optimization algorithms to find a solution to the vehicle routing problem with a drone [7], Cai et al. com-

¹ ORCID ID: 0000-0003-0629-6888.

E-mail address: faltunbey@firat.edu.tr

bined different metaheuristic methods to predict soil liquefaction potential [8]. Shadkam presented these methods for the management of COVID-19 waste [9], Özbay proposed a new approach using these methods for the detection of diabetic retinopathy from fundus images [10], Fang et al. proposed a method combining machine learning and optimization algorithms for solving small chemical process processes [11]. Mohammadi and Reja Hejazi developed a new method for controlling the growth of cancer cells, combining two different optimization algorithms [12], Yildirim et al. have adapted different metaheuristic algorithms for feature selection in speech-recognition system design [13].

Metaheuristic algorithms consist of two main stages: exploration (diversification) and exploitation (intensification) [14]. The exploration process is the investigation of the search space as broadly as possible. Avoidance of local optima is associated with this stage. For this step to be successful, random operators are required for metaheuristic optimization algorithms. In this way, the metaheuristic explores the search space globally. In the exploitation phase, which takes place after the exploration phase, the entire search space is not examined, only the search space determined during the exploration phase is taken into account [15]. A suitable balance between these two stages is very crucial for the success of an optimization algorithm. All of the optimization algorithms based on swarm intelligence have been developed based on these features. However, each of them has adopted different operators and search mechanisms. In recent years, many metaheuristic optimization algorithms have been proposed to solve optimization problems with different constraints. While developing these algorithms, inspiration is taken from the elements, plants, physical phenomena, and the movement, hunting, and communication of animals in nature. Optimization algorithms modeled based on these behaviors are as follows: Particle Swarm Optimization (PSO) [16], Ant Colony Optimization (ACO) [17], Wasp Swarm Optimization (WSO) [18], Cat Swarm Optimization (CSA) [19], Artificial Bee Colony Algorithm (ABC) [20], Bacterial Swarming (BS) [21], Gravitational Search Algorithm (GSA) [22], Firefly Algorithm (FA) [23], Bacterial Foraging Algorithm (BFA) [24], Fruit Fly Optimization Algorithm (FOA) [25], Cuckoo Search Algorithm (CSA) [26], Grey Wolf Optimizer (GWO) [27], Moth Flame Optimization (MFO) [28], Ant Lion Optimizer (ALO) [29], Multi-Verse Optimizer (MVO) [30], Water Evaporation Optimization (WEO) [31], Electromagnetic Field Optimization (EFO) [32], Whale Optimization Algorithm (WOA) [33], Spotted Hyena Optimization Algorithm (SHOA) [34], Salp Swarm Algorithm (SSA) [35], Grasshopper Optimization Algorithm (GSA) [36], Coyote Optimization Algorithm (COA) [37], Seagull Optimization Algorithm (SOA) [38], Atom Search Optimization (ASO) [39], Chimp Optimization (CO) [40], Red Fox Optimization Algorithm (RFO) [41], Firebug Swarm Optimization (FSO) [42], Gazelle Optimization Algorithm (GOA) [43], Dandelion Optimizer (DO) [44], Artificial Rabbits Optimization (ARO) [45].

The emergence of difficult optimization problems has caused optimization algorithms to be proposed that consistently achieve better results and improve existing algorithms. This situation follows the no-free lunch (NFL) theory. This theorem has proven that there is no optimization method that best solves all optimization problems [46]. According to the NFL theorem, many researchers have proposed or developed many metaheuristic optimization algorithms inspired by nature: Enhanced moth swarm algorithm [47], Complex-valued encoding symbiotic organisms search algorithm [48], Teaching-learning-based pathfinder algorithm [49], Modified remora optimization algorithm [50], and so on.

The chaotic functions are mostly used to balance the exploration and exploitation phases in metaheuristic optimization algorithms. It can increase convergence speed and diversity in metaheuristic algorithms with chaotic maps. Thus, better results

are obtained from these algorithms [51]. The chaos theory is one of the characteristics of nonlinear systems [52]. In the context of mathematics, chaos is defined as the randomness generated by actual systems. Many researchers have added chaos theory to different metaheuristic optimization algorithms to increase the algorithm's ability to obtain the optimum solution. Some of the chaos-based algorithms in the literature: chaotic grey wolf optimizer [53], chaotic squirrel search algorithm [54], chaotic golden section search [55], chaotic teaching learning-based optimization [56], Chaotic arithmetic optimization [57], Chaotic slime mould optimization [58], chaotic artificial immune system optimization [59], chaotic artificial bee optimization [60], chaotic krill herd optimization [61], chaotic symbiotic organisms search algorithm [52], chaotic henry gas solubility optimization [62], Chaotic grasshopper optimization algorithm [63], chaotic water cycle optimization [64], Chaotic whale optimization [65], chaotic bird swarm optimization [66]. These approaches, developed by different researchers, are used to solve various optimization problems in different fields of engineering.

The Seahorse Optimization (SHO) algorithm is a new metaheuristic optimization algorithm proposed in 2022 [67]. SHO mimics the movement, hunting, and breeding behavior of seahorses in nature. In this study, an effective optimization method modified by integrating chaotic maps into the SHO algorithm is proposed. In the original SHO algorithm, random variables are used to model the hunting behavior of seahorses. In the proposed CSHO algorithm, 10 different chaotic maps are used instead of random variables to improve the exploration and exploitation stages of the SHO algorithm. The aim of this proposed new method is to increase the convergence rate and avoid local optimum points. The results show that CSHOs outperform the original SHO and its competitors in terms of convergence speed and solution quality.

The main contribution of this paper is highlighted below:

- SHO is a fairly new optimization algorithm. This study is the first study on SHO in the literature.
- In this study, for the first time, the CSHO algorithm is proposed by integrating 10 different chaotic maps into the SHO algorithm.
- The efficiency and performance of the proposed CSHO algorithm have been evaluated in unimodal, multimodal, fixed-dimension multimodal, and CEC2019 functions.
- To verify the performance of the proposed CSHO algorithm, the algorithm is compared with the original SHO, SSA, SCA, WOA, and PSO algorithms.
- Friedman and Wilcoxon statistical tests have been conducted to confirm the statistical superiority of the proposed CSHO.
- Boxplot analysis has been performed to show the consistency of the proposed CSHO algorithm.
- Four real-world engineering design problems (welded beam design problem, pressure vessel design problem, tension/compression spring design problem, and speed reducer design problem) are solved to evaluate the problem-solving ability of the proposed CSHO algorithm.

The remaining parts of the paper are organized as follows: In section 2, the mathematical model, working principle, and pseudo-code of the SHO algorithm, which is inspired by sea horses, are introduced. The chaotic maps and equations used in this study are examined in section 3. The new CSHO algorithm proposed for the first time in this study is explained in section 4. The proposed CSHO algorithm has been applied to 33 different benchmark functions and 4 different real-world engineering design problems in section 5. In addition, in this section, the results obtained from the CSHO algorithm are compared with the metaheuristic optimization algorithms in the literature, and Friedman and Wilcoxon

statistical analysis is applied to verify the performance of CSHO. The conclusions of this research and the work to come are outlined in section 6.

2. Seahorse optimization (SHO) algorithm

The SHO algorithm's details are provided in this section. The motivation for the proposed CSHO algorithm is initially presented in this work. This section examines the SHO algorithm's fundamental functioning. Then, mathematical models are developed.

2.1. Seahorses

The SHO algorithm's details are provided in this section. The motivation for the proposed CSHO algorithm is initially presented in this work. This section examines the SHO algorithm's fundamental functioning. Then, mathematical models are developed.

Seahorses have long been among the most well-known and popular animals in the world, but scientists are still learning new things about these critters, sometimes known as the hippocampus. These creatures have played important roles in fairy tales, superstitions, medicine, and economics for thousands of years. In recent years, just as the future of these wonderful animals is in jeopardy, some scientists have gained new information about this creature [68].

The seahorses, one of which is shown in Fig. 1a, remained on earth for 40 million years, although only a small number of them survived. Nowadays their future is in danger and many species have been listed as endangered animals by the World Wildlife Fund organization [69].

Environmental conditions are the leading cause of the extinction of seahorses. Experts state that seagrass meadows, as well as most of the seahorse's habitat, will disappear in the last 50 years due to the pollution of the seas. In addition, approximately 26 million seahorses have hunted annually and used to make souvenirs. It is also an important resource for the pharmaceutical industry. This animal, which is hunted in about 30 countries, is used in the treatment of diseases such as asthma, headache, and cough, but the most used place is drugs produced against sexual inadequacy [70].

These creatures are mostly found in coastal waters where currents are low; they appear to live in tropical and subtropical seas, the Mediterranean, and even the cold North Sea [71]. Exactly how many species this creature has, of which about 80 species have been discovered so far, is still unclear [72].

Seahorses swim upright, are relatives of sea needles, and range in length from 1.5 cm to 35 cm. Because of their body shape, seahorses are very clumsy swimmers and can easily die of exhaustion if they are caught in the current. They propel themselves forward by flapping the small fin on their back 35 times per second. They

also use even smaller fins on the back of their heads for orientation [73]. They usually have a tubular nose. Nose length restricts head movement and is assumed to be directly related to nutrition [74]. Thanks to the curling and grasping structure of their tails, they hold on to seagrass and corals and inhale plankton and small crustaceans drifting around them [75]. They are very voracious and can eat 3,000 or more brine shrimp a day by feeding constantly. It has been determined that a seahorse in the larval stage has a strong hunting ability. During feeding, the extended tube in the shape of a kiss is brought closer to the food and opens its mouth to swallow the food. The head of a seahorse, resembling a horse figure, often forms a crown on top. The tail of seahorses is quite atrophied in terms of propulsion. Seahorses use their dorsal fins almost entirely on the back and pectoral fins on either side of their heads to swim and orient themselves. The tail is mainly used to cling to algae or coral and has evolved accordingly. The biggest feature that distinguishes the seahorse tail from all other tails is that it is square (4-corners) [76]. A radial crest is typically present on the opercular, and the gill portion is high. There are no ventral or caudal fins, and there is only one dorsal fin for propulsion between the body and tail. Membranous bone plates fully encircle its entire body. To rest and flee their prey, seahorses adapt their color to their surroundings. The seahorse's tail is unlike any other fish's in terms of both anatomy and function [77].

As shown in Fig. 1b, seahorses are the only animals in which a male individual gives birth to offspring. During mating, the female enters the water-filled sac in the male's abdomen through the resulting oviduct. She then lays her eggs here to be fertilized with her partner's sperm. Delivery takes place at the end of 4–6 weeks. The birth process can be so troublesome that some fathers may become exhausted and die [78].

2.2. Inspiration

Movement, predation, and breeding behaviors, which are defined as the behaviors that complement the life characteristics of sea horses, have critical importance in the definition of the algorithm and are explained below.

- Seahorses curl their tail to algae or leaves in locomotion behavior. The seahorse performs a spiral movement at this time, as the branches show changes that spiral around the roots of the algae under the influence of the sea eddies. When seahorses hang upside-down from floating algae or other things and move randomly with the waves that are another movement instance called Brownian motion.
- In the movement, which is described as a predation behavior, seahorses catch their prey with a minimum of 90% success by



Fig. 1. a) A sample of seahorse b) a male seahorse giving birth.

taking advantage of the special shape of their heads and sneaking close to their prey.

- In breeding behavior, male and female seahorses mate randomly to give birth to new offspring, which then inherit some excellent traits from the parents to the offspring.

Generally, these three actions help sea horses better adapt to their surroundings and survive. The three behaviors listed above serve as the major inspiration for the SHO algorithm. These behaviors inspired us to create a mathematical model of the SHO optimizer.

2.3. Mathematical model of SHO

SHO basically consists of three important components: movement, predation, and breeding. Local and global search strategies developed to balance the discovery and use of SHO are designed for the social behaviors of movement and predation, respectively. If the behaviors of the first two components are completed, the reproductive behavior is performed. In this section, the mathematical models of SHO are explained in detail [67].

Initialization: As with most MOAs, SHO has also an initial population. The whole population of seahorses can be described as given in Eq. (1) assuming that each seahorse represents a potential solution in the search space of the problems [67].

$$\text{Seahorses} = \begin{bmatrix} x_1^1 & \dots & x_1^{\text{Dim}} \\ \vdots & \ddots & \vdots \\ x_{\text{pop}}^1 & \dots & x_{\text{pop}}^{\text{Dim}} \end{bmatrix} \quad (1)$$

where *pop* describes the size of the population and *Dim* is the variable's dimension. As indicated by *BU* and *BL*, each solution is produced at random between the problem's upper bound and lower bound. $[BU, BL]$ in the search space where the *i*th individual expression of X_i is represented is calculated as given in Eq. (2):

$$X_i = [x_i^1, \dots, x_i^{\text{Dim}}], x_i^j = \text{random} \times (BU^j - BL^j) + BL^j \quad (2)$$

where *random* represents a random value in the range $[0,1]$. x_i^j , *i*th individual *j*th represents a dimension. *i* is a positive integer with a value between 1 and *pop*. A positive integer in the range $[1, \text{Dim}]$ is used to express *j*. BU^j and BL^j indicate the optimized problem of variable *j*th. The individual with the minimum fitness, from which the minimum optimization problem is taken as an example, is considered the elite individual indicated by X_{elite} . X_{elite} is obtained as given in Eq. (3) [67]: where $f(\cdot)$ is the value of the problem's objective function.

$$X_{\text{elite}} = \text{argmin}(f(X_i)) \quad (3)$$

The seahorses' movement behavior: The various seahorse movement patterns for the initial behavior roughly match the *random* $(0, 1)$ normal distribution. Half of the $r_1 = 0$ cut-off is taken for local mining and the other half is taken for global search to compromise exploration and operational performance. Therefore, the movements are examined in two separate cases [67]:

Case 1 - Spiral: It has been defined as the spiral movement of the seahorse together with the vortex event in the sea. The random normal value r_1 effectively realizes the local application of SHO when it is situated to the right of the breakpoint. Seahorses follow the spiral movement and move towards the X_{elite} . To prevent the excessive exploitation of SHO locally, Lévy flight is employed to simulate the movement step size of seahorses, which are likely to travel to different positions in early iterations [79]. To broaden the range of already-existing regional solutions, the seahorse's spiral motion constantly modifies its rotational angle. In this case, the

creation of a seahorse's new position is expressed mathematically as given in Eq. (4):

$$X_{\text{new}}^1(t+1) = X_i(t) + Levy(\lambda)((X_{\text{elite}}(t) - X_i(t)) \times x \times y \times z + X_{\text{elite}}(t)) \quad (4)$$

where $x = \rho \times \cos(\theta)$, $y = \rho \times \sin(\theta)$, and $z = \rho \times \theta$ represent the 3D components of the x, y , and z coordinates with a spiral motion. It aids in updating the search agents' positions. The logarithmic spiral constants u and v specify the length of the stems, which are represented by $\rho = u \times e^{v\theta}$ (u and v are set to 0.05). θ is defined as a random value in the range $[0, 2\pi]$. $Lévy(z)$ is the Lévy flight distribution function and is calculated as given in Eq. (5) [79]:

$$Levy(z) = s \times \frac{w \times \sigma}{|k|^{\lambda}} \quad (5)$$

here λ is a random number defined in the interval $[0, 2]$ ($\lambda = 1.5$ in the study). s is a constant value defined as 0.01. k and w are random numbers in the range $[0, 1]$. σ is calculated as given in Eq. (6):

$$\sigma = \frac{\Gamma(1+\lambda) \times \sin(\frac{\pi\lambda}{2})}{\Gamma(\frac{1+\lambda}{2}) \times \lambda \times 2^{\frac{\lambda-1}{2}}} \quad (6)$$

Case 2 - Brownian: It is described as the seahorse's Brownian motion in conjunction with the waves. When r_1 is found to be to the left of the breakpoint, SHO is found under the influence of drift. The search procedure is crucial for SHO to prevent the local extremum in this situation. Brownian motion is used to mimic another moving length of the seahorse to improve search space exploration. Eq. (7) provides the mathematical expression for this circumstance:

$$X_{\text{new}}^1(t+1) = X_i(t) + \text{random} * l * \beta_t * (X_i(t) - \beta_t * X_{\text{elite}}) \quad (7)$$

here l is the constant coefficient ($l = 0.05$ in the study). Eq. (8) is used to calculate β_t , the random walk coefficient of Brownian motion that matches the standard normal distribution as a random value [80].

$$\beta_t = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \quad (8)$$

In general, in the sum of these two cases, the formula given in Eq. (9) is applied to obtain the new position of the seahorse in the t iteration:

$$\begin{cases} X_{\text{new}}^1(t+1) = \\ X_i(t) + Levy(\lambda)\left((X_{\text{elite}}(t) - X_i(t)) \times x \times y \times z + X_{\text{elite}}(t)\right) r_1 > 0 \\ X_i(t) + \text{random} * l * \beta_t * (X_i(t) - \beta_t * X_{\text{elite}}) r_1 \leq 0 \end{cases} \quad (9)$$

here, $r_1 = \text{random}()$ is used for the random normal number. Fig. 2 illustrates the position update diagram for two different kinds of seahorses, namely spiral or Brownian motion cases. The seahorse moves randomly because of the ambiguous conditions in the sea, as shown by both the Lévy and Brownian motion patterns.

The seahorses' predation behavior: The effort of the seahorse to hunt zooplankton and small crustaceans results in two results, success, and failure. Given that the seahorse has a greater than 90% probability of succeeding in catching food, the random number r_2 of the SHO designed to distinguish these two outcomes is set to a critical value of 0.1. Hunting success underscores the SHO's exploitative ability, as the elite is assumed to indicate, to a certain extent, the approximate location of the prey. If $r_2 > 0.1$, it is accepted that the predatory movement of the seahorse is successful, that is, the seahorse sneaks up on its prey (elite) and catches it by moving faster than the prey. Conversely, when hunting is

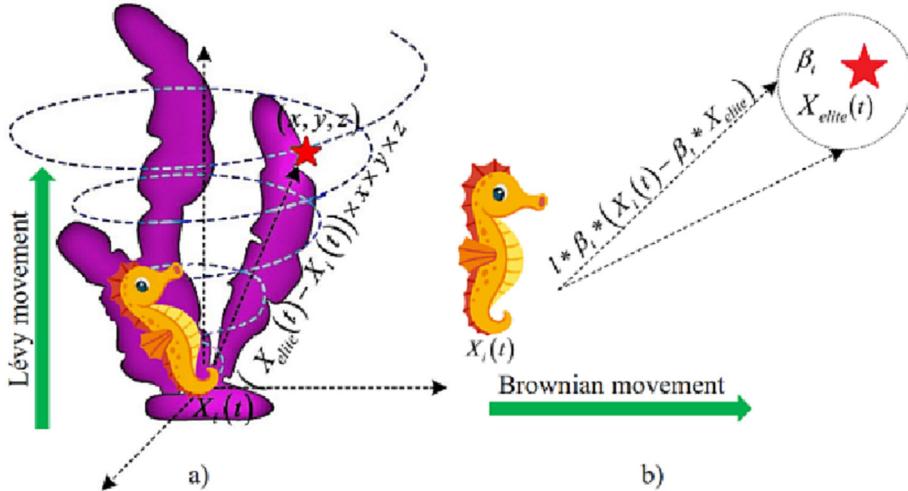


Fig. 2. a) Lévy and b) Brownian motion patterns of the seahorse in the sea.

unsuccessful, the response rate of both is considered inverse to the previous one, implying the seahorse's propensity to explore the foraging area. The mathematical expression of this predatory behavior is given in Eq. (10):

$$X_{new}^2(t+1) = \begin{cases} \alpha * (X_{elite} - random * X_{new}^1(t))r_2 > 0.1 \\ \quad + (1 - \alpha) * X_{elite} \\ (1 - \alpha) * (X_{new}^1(t) - random)r_2 \leq 0.1 \\ \quad * X_{elite} + \alpha * X_{new}^1(t) \end{cases} \quad (10)$$

here $X_{new}^1(t)$ is the seahorse's new location following its movement in iteration t , and r_2 is defined as a random number between $[0,1]$. α is determined as stated in Eq. (11) and is linearly lowered by iterations to adapt the seahorse's stride size towards the prey to be sought. Where T stands for the maximum allowed the number of iterations.

$$\alpha = \left(1 - \frac{t}{T}\right)^{\frac{2t}{T}} \quad (11)$$

Two possible outcomes of the seahorse's predatory behavior are illustrated in Fig. 3. Accordingly, the red star position shows the updated position of the seahorse, and the approximate position of the prey is marked with a blue dot. If the seahorse is successfully caught, it is seen in Fig. 3a that the seahorse becomes *elite*. In this case, under the control of the parameter α , it will gradually converge to the global optimal individual with increasing iterations. In Fig. 3b, a global search is performed because prey cannot be caught. The parameter $1 - \alpha$ is applied to the vector between the current individual and the elite, and α acts on the currently updated individual. This approach is designed to allow seahorses to search

globally in the first iterations and avoid overuse in subsequent iterations.

The seahorses' breeding behavior: Depending on their fitness value, the population is separated into male or female divisions. SHO, it should be noted that because male seahorses are the ones who reproduce, only half of those with the highest fitness ratings are considered to be fathers and only the other half to be mothers. This divide will make it easier for mothers and fathers to pass on positive features to the next generation, ensuring its continuity, and stopping the over-localization of new solutions. Eq. (12) gives the mathematical expression of the seahorse's assigned role in reproduction.

$$fathers = X_{sort}^2 \left(1 : \frac{pop}{2}\right), mothers = X_{sort}^2 \left(\frac{pop}{2} + 1 : pop\right) \quad (12)$$

here the X_{new}^2 's are listed in increasing order of fitness value under the symbol X_{sort}^2 . Fathers and mothers, correspondingly, represent the male and female populations. Males and females are randomly mated together to create new offspring. It was believed that each pair of seahorses gave birth to just one child to conveniently implement the SHO. The mathematical expression of the offspring is given in Eq. (13):

$$X_i^{offspring} = r_3 X_i^{father} + (1 - r_3) X_i^{mother} \quad (13)$$

here r_3 indicates the number in the $[0,1]$ range randomly. i is a positive integer with a range of $[1, pop/2]$. Male and female populations, respectively, are represented by X_i^{father} and X_i^{mother} which are selected from two random individuals.

Fig. 4 depicts the seahorse breeding process. Each individual is ranked in ascending order based on their fitness values, as shown in Fig. 4a. A newly generated offspring's approximate location is

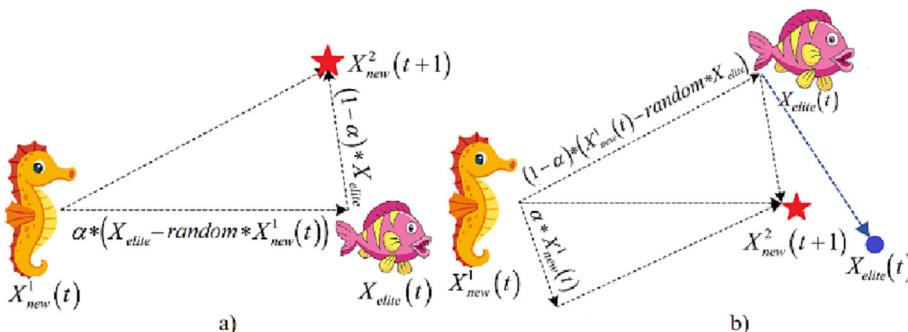


Fig. 3. Predation behaviors of the seahorse a) prey successfully b) prey failure.

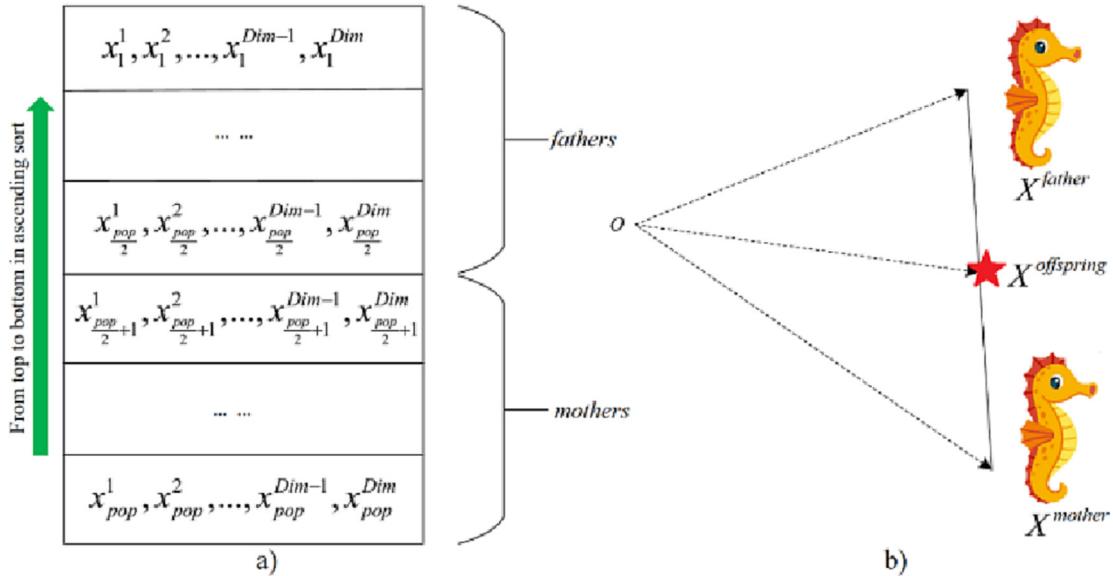


Fig. 4. Seahorse breeding process a) individual sorting b) offspring location.

shown in Fig. 4b. The genetic information is efficiently passed between the two subpopulations since it is randomly created along the line between parents.

Algorithm-1 displays the SHO's pseudo-code. The population is first initialized by producing a collection of random solutions as part of the SHO implementation procedure. Eqs. (9) and (10) are used to update the population of seahorses, while Eq. (13) is utilized to rear the offspring. Seahorses that have recently been upgraded and their pups make form a new population. The new population uses 1.5 pops. Each individual in the new population is calculated to prevent unrestricted population growth. From top to bottom, individuals are ranked according to their fitness, and the first pop seahorses are selected iteratively as the new population for the following evolutionary phase.

Algorithm-1

Input: pop -size of population, Dim -dimension of variable, T -max iteration
Output: X_{best} search agent (optimum) and f_{best} its fitness value

- 1: Initializing of seahorses $X_i (i = 1, \dots, N)$
- 2: Compute fitness values for each seahorse
- 3: Determination of best seahorse X_{elite}
- 4: **while** ($t < T$) **do**
- 5: **if** ($r_1 = random > 0$) **do** //Movement//
- 6: Setting $u = 0.05$ and $v = 0.05$ constant parameter values
- 7: Rotation angle θ , $Rand[0, 2\pi]$
- 8: Implement of Lévy coefficient using Eq. (5)
- 9: Updating of seahorse position using Eq. (4)
- 10: **else if** **do**
- 11: Setting constant parameter $l = 0.05$
- 12: Updating of seahorse position using Eq. (7)
- 13: **end if**
- 14: Updating of seahorse position using Eq. (10) //Predation//
- 15: Handling of out-of-bounds variables
- 16: Compute fitness values for each seahorse
- 17: Selection of $father$ and $mother$ using Eq. (12) //Breeding//
- 18: Breed offspring using Eq. (13)
- 19: Handling of out-of-bounds variables
- 20: Compute fitness values for each offspring

1 (continued)

Algorithm-1

- 21: The next population of iterations is selected from the offspring and the best parents are ranked by their fitness values on top pop
- 22: Updating of X_{elite} position
- 23: $t = t + 1$
- 24: **end while**

3. Chaotic maps

Chaotic maps are known as discrete-time dynamical systems. As in most of the metaheuristic optimization methods, long-period random number sequences have an important place in this algorithm. It is desirable that the generated numbers are not the same, have a spread spectrum, do not need much time while generating the numbers, and have low storage costs. Chaotic maps have successful applications in a variety of domains because of their semi-stochastic, initial conditional, and ergodicity properties. Chaotic maps are frequently used in modeling complex problems in the fields of medicine, engineering, ecology, and economics, numerical analysis methods, image encryption, and cryptology. It is possible to avoid local optimum points or reduce the probability of getting stuck in local optima by using chaotic maps. It is especially preferred for increasing the solution performance of various optimization methods. Various chaotic maps have produced effective results in improving the constrained stochastic nature of some optimization algorithms. In this study, 10 different chaotic maps, including Chebyshev, Circle, Gauss/mouse, Iterative, Logistic, Piecewise, Sine, Singer, Sinusoidal, and Tent have been used to reduce the probability of the original SHO getting stuck in the local optimum and to increase the global convergence speed. Fig. 5 demonstrates the graphs of the 10 chaotic maps in 100 iterations that have been employed. Parameters and equations related to the maps used are given in this section:

The Chebyshev map is given in Eq. (14):

$$x_{n+1} = \cos(k \cos^{-1} x_n) \quad (14)$$

The Circle map with its parameters is given in Eq. (15). $a = 0.5$ and $b = 0.2$ are taken.

$$x_{n+1} = x_n + b - \left(\frac{a}{2\pi}\right) \sin(2\pi x_n) \bmod(1) \quad (15)$$

The Gauss/mouse map with its parameters is given in Eq. (16):

$$x_{n+1} = \begin{cases} 0, & x_n = 0 \\ \frac{1}{x_n \bmod(1)}, & x_n \in (0, 1) \end{cases}, \quad \frac{1}{x_n \bmod(1)} = \frac{1}{x_n} - \left\lfloor \frac{1}{x_n} \right\rfloor \quad (16)$$

The Iterative map with its parameters is given in Eq. (17). $a = 0.7$ is taken.

$$x_{n+1} = \sin\left(\frac{a\pi}{x_n}\right) \quad (17)$$

The Logistic map with its parameters is given in Eq. (18). $a = 0.4$ is taken.

$$x_{n+1} = ax_n(1 - x_n) \quad (18)$$

The Piecewise map with its parameters is given in Eq. (19). $P = 0.4$ is taken.

$$x_{n+1} = \begin{cases} \frac{x_n}{P}, & 0 \leq x_n < P \\ \frac{x_n - P}{0.5 - P}, & P \leq x_n < 0.5 \\ \frac{1 - P - x_n}{0.5 - P}, & 0.5 \leq x_n < 1 - P \\ \frac{1 - x_n}{P}, & 1 - P \leq x_n < 1 \end{cases} \quad (19)$$

The Sine map with its parameters is given in Eq. (20). $a = 4$ is taken.

$$x_{n+1} = \frac{a}{4} \sin(\pi x_n), \text{ for } (0 < a \leq 4) \quad (20)$$

The Singer map with its parameters is given in Eq. (21). $\mu = 1.07$ is taken.

$$x_{n+1} = \mu(7.86x_n - 23.31x_n^2 + 28.75x_n^3 - 13.302875x_n^4) \quad (21)$$

The Sinusoidal map is given in Eq. (22).

$$x_{n+1} = ax_n^2 \sin(\pi x_n) \quad (22)$$

when $a = 2.3$ and $x_0 = 0.7$ are selected, Eq. (22) is simplified with Eq. (23).

$$x_{n+1} = \sin(\pi x_n) \quad (23)$$

The Tent map with its parameters is given in Eq. (24).

$$x_{n+1} = \begin{cases} \frac{x_n}{0.7}, & x_n < 0.7 \\ \frac{10}{3x_n(1-x_n)}, & \text{otherwise} \end{cases} \quad (24)$$

4. Proposed chaotic seahorse optimization (CSHO) algorithm

This section describes the proposed CSHO algorithm to improve the performance of the SHO algorithm. The quality of the solutions produced with optimization algorithms can be considerably improved by using chaotic maps. The chaos strategy is used to prevent becoming stuck in local optima and enhance the effectiveness of searching the global optimum. In the SHO algorithm, the hunting strategy of seahorses results in two forms, success, and failure. In SHO, the hunting success of prey emphasizes exploitation ability, while failure emphasizes the propensity to explore the search space. This hunting behavior is mathematically expressed by Eq. (10). In this equation, random variables have been used while determining the position of the sea horses after the movement. In the proposed CSHO algorithm, the values of these random variables are generated with 10 different chaotic maps to improve the exploration and exploitation stages of the SHO algorithm. Fig. 6 illustrates the flowchart of the CSHO algorithm.

Chaotic maps are employed in place of random number generators to produce random variables. The numbers produced by the chaotic maps are distinct from the numbers produced by the random function. While the numbers obtained with chaotic maps are produced in a certain order, they are unordered with the random function. This is another justification for using chaotic maps in the optimization methods to increase their effectiveness [56].

The random variable in Eq. (10) is calculated with the chaotic map determined during the execution of the algorithm. Mathematically, in order to propose the CSHO, Eq. (10) in the original SHO has been replaced by Eq. (25).

$$X_{new}^2(t+1) = \begin{cases} \alpha * (X_{elite} - \text{chaos}(t) * X_{new}^1(t))r_2 > 0.1 \\ \quad + (1 - \alpha) * X_{elite} \\ (1 - \alpha) * (X_{new}^1(t) - \text{chaos}(t) * X_{elite})r_2 \leq 0.1 \\ \quad + \alpha * X_{new}^1(t) \end{cases} \quad (25)$$

In Eq. (25), t is the current iteration of the algorithm. $\text{chaos}(t)$ is used instead of the randomly assigned random variable in Eq. (10). 10 different chaotic maps are used in the proposed CSHO: Chebyshev map-based SHO (CSHO1), circle map-based SHO (CSHO2), gauss/mouse map-based SHO (CSHO3), iterative map-based SHO (CSHO4), logistic map-based SHO (CSHO5), piecewise map-based SMA (CSHO6), sine map-based SHO (CSHO7), singer map-based SHO (CSHO8), sinusoidal map-based SHO (CSHO9), and tent map-based SHO (CSHO10). Algorithm-2 displays the CSHOs pseudo-code.

Algorithm-2

Input: pop-size of population, Dim-dimension of variable, T-max iteration

Output: X_{best} search agent (optimum) and f_{best} its fitness value

- 1: Initializing of seahorses $X_i (i = 1, \dots, N)$
 - 2: Compute fitness values for each seahorse
 - 3: Determination of best seahorse X_{elite}
 - 4: **while** ($t < T$) **do**
 - 5: **if** ($r_1 = \text{random} > 0$) **do** //Movement//
 - 6: Setting $u = 0.05$ and $v = 0.05$ constant parameter values
 - 7: Rotation angle θ , $\text{Rand}[0, 2\pi]$
 - 8: Implement of Lévy coefficient using Eq. (5)
 - 9: Updating of seahorse position using Eq. (4)
 - 10: **else if** **do**
 - 11: Setting constant parameter $l = 0.05$
 - 12: Updating of seahorse position using Eq. (7)
 - 13: **end if**
 - 14: Updating of seahorse position using Eq. (10) using with chaotic maps //Predation//
 - 15: Handling of out-of-bounds variables
 - 16: Compute fitness values for each seahorse
 - 17: Selection of *father* and *mother* using Eq. (12) //Breeding//
 - 18: Breed offspring using Eq. (13)
 - 19: Handling of out-of-bounds variables
 - 20: Compute fitness values for each offspring
 - 21: The next population of iterations is selected from the offspring and the best parents are ranked by their fitness values on top pop
 - 22: Updating of X_{elite} position
 - 23: $t = t + 1$
 - 24: **end while**
-

In this section, the computational complexity of CSHO is analyzed. The computational complexity of CSHO depends on the population

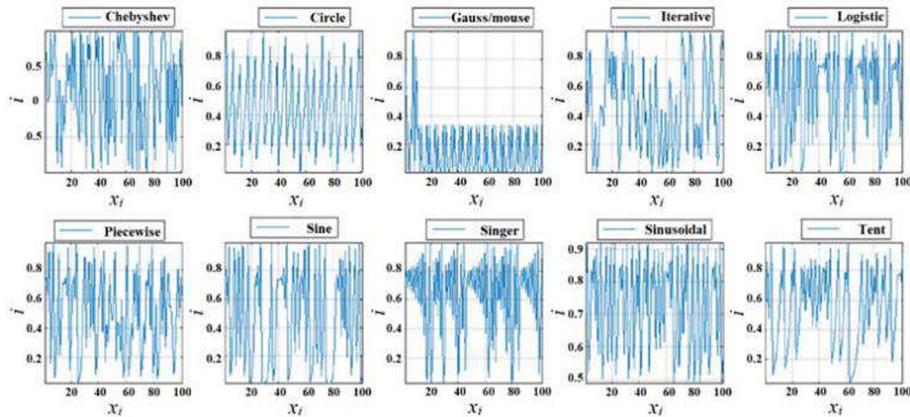


Fig. 5. Illustration of 10 different chaotic maps.

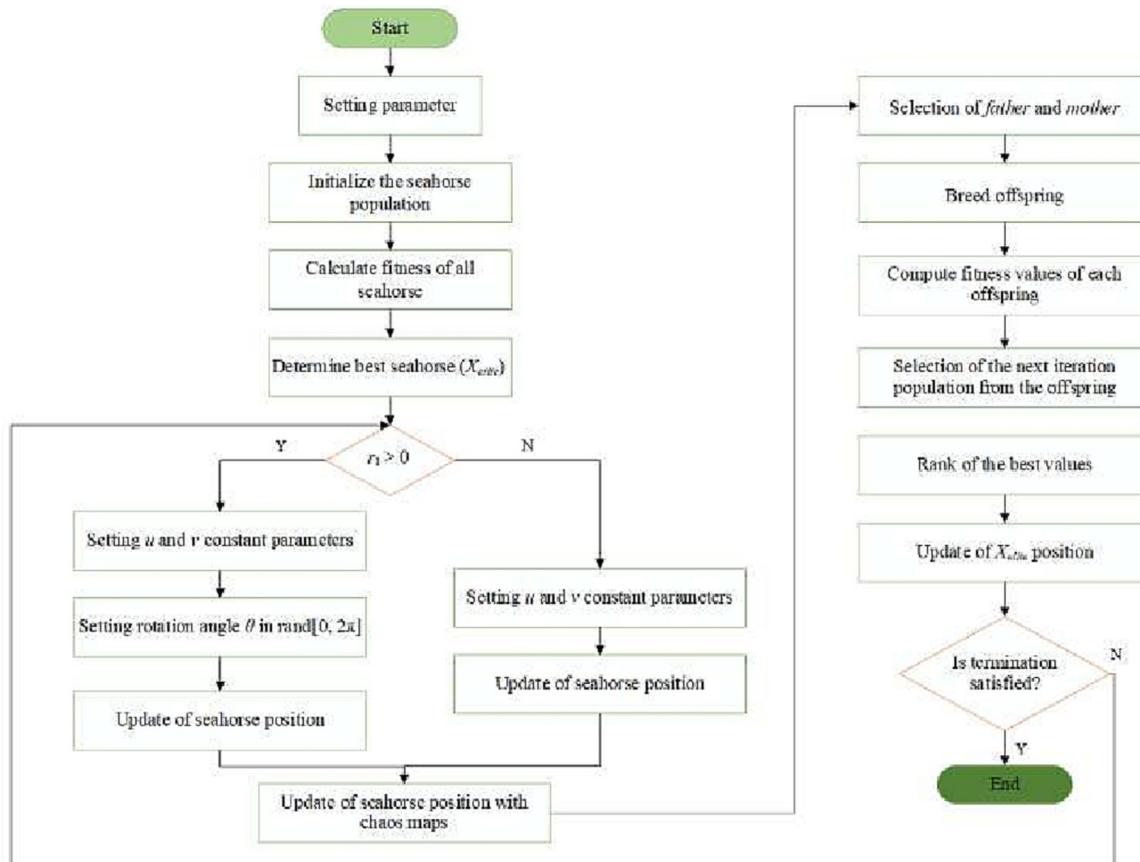


Fig. 6. The flowchart of the CSHO algorithm.

size N , the dimension of problem D , and the maximum number of iterations T_{max} . The computational complexity of the proposed CSHO algorithm using 10 different chaotic maps is $O(T_{max} \times N \times D)$. Thus, the computational complexity of CSHO can be simplified to $O(N \times D)$.

5. Experimental result and analysis

This section contains simulation results and an analysis of the proposed CSHO algorithm. The performance of the proposed CSHO algorithm is tested on 33 different benchmark functions and 4 different real-world engineering design problems. The obtained results from CSHO for 33 comparison functions are compared with

the original SHO and the popular metaheuristic optimization algorithms SCA, SSA, WOA, and PSO in the literature. Moreover, Friedman and Wilcoxon signed-rank tests are done to validate the performance of the proposed CSHO algorithm.

All tests are performed on a machine with Windows 11 operating system, 32 GB RAM, and Intel(R) core i7-7700 processor (4.00 GHz). MATLAB 2021b is used to code the proposed CSHO method and other metaheuristic algorithms.

5.1. Standard benchmark functions and CEC2019

In this study, 33 different benchmark functions were used to examine the performance of the proposed CSHO algorithm. These

functions are unimodal, multimodal, fixed-dimension multimodal functions, and CEC2019 functions [81]. The function equation, range, dimension, and optimum values of these functions used for experimental evaluation are listed in Tables 1, 2, 3, and 4. In these tables, the range represents the range values of the function, dim depicts the size of the problem, and f_{min} stands for the optimal value of the function.

All metaheuristic algorithms were run 20 times, the population size is set to 30, and the number of maximum iterations is 500 for fairness of experiments. The results obtained from the algorithms are presented as a mean solution (Avg) and standard deviation (Std). The settings of the specific parameters of the original SHO, SCA, SSA, WOA, and PSO algorithms used to compare the performance of CSHOs are given in Table 5. The parameter values of the SCA, SSA, WOA, and PSO algorithms were determined based on the parameter selections commonly used in the literature.

This study aims to enhance the performance of the original SHO with the proposed CSHOs utilizing 10 different chaotic maps. The results obtained with 33 different benchmark functions from the proposed CSHO in this study for the first time were compared with the original SHO.

The performance of the proposed CSHOs was also validated using the Friedman statistical test. The values of Avg, Std, and Mean Rank (MR) obtained from the CSHOs and the original SHO algorithms for the F₁-F₃₃ functions are given in Table 6. The performances of the algorithms are evaluated according to average values in this table. Table 6 shows that the proposed CSHOs outperform the original SHO for all unimodal functions. It is seen that the best values have been obtained with Sinusoidal map-based SHO for F₁, and F₂, Singer map-based SHO for F₃ and F₄, Sinusoidal and Tent map-based SHO for F₅, Tent map-based SHO for F₆, and Logistic map-based SHO for F₇. The best results are displayed in bold in all tables.

In multimodal benchmark functions, all CSHOs performed considerably better than the original SHO for F₈, F₁₂, and F₁₃ functions. It has been seen that the CSHOs and SHO algorithms show the same performance for the F₉-F₁₁ functions.

Examining the fixed-dimension multimodal benchmark functions, it is seen that all CSHOs achieve the same performance as the original SHO in F₁₄, F₁₆-F₂₀. For the remaining functions, the best values have been obtained with Sinusoidal map-based SHO for F₁₅, Chebyshev map-based and Sine map-based SHO for F₂₁, Piecewise map-based SHO for F₂₂, and Tent map-based SHO for F₂₃.

In the CEC2019 functions, it is observed that the CSHOs give the same values for F₂₅, F₂₆, and F₃₂. At the same time, the best values have been obtained with Sine map-based SHO for F₂₄ and F₂₇, Gauss/mouse map-based SHO for F₂₈ and F₃₀, Logistic map-based SHO for F₂₉, Circle map-based SHO for F₃₁ and Chebyshev map-based SHO for F₃₃.

When Table 6 is examined, it is seen that the results obtained with the proposed CSHOs using 10 different chaotic maps have better results than the original SHO for 33 benchmark functions of different types.

The convergence curves of the benchmark function are shown in Fig. 7. It has been observed that the convergence speed of the proposed CSHOs algorithms using 10 different chaotic maps according to the convergence curves is better than the original SHO algorithm.

The Friedman test has been applied to further analyze the performance of the CSHOs and SHO. The performance results obtained have been analyzed with the Friedman test to determine whether there is a significant difference between the CSHOs and the original SHO [82]. The results of this test are represented in Table 7. For the Friedman test, $X_{table}^2 = 18.307$ at a confidence level of 0.05 in the chi-square table. If, $X_R^2 > X_{table}^2$ the H_0 hypothesis is rejected. As seen in

Table 1

Description of unimodal benchmark functions.

(F _i) Function	Range	Dim	f_{min}
$F_1(X) = \sum_{i=1}^n x_i^2$	[-100,100]	30	0
$F_2(X) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	[-10,10]	30	0
$F_3(X) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	[-100,100]	30	0
$F_4(X) = \max_i \{x_i, 1 \leq i \leq n\}$	[-100,100]	30	0
$F_5(X) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	[-30,30]	30	0
$F_6(X) = \sum_{i=1}^n (x_i + 0.5)^2$	[-100,100]	30	0
$F_7(X) = \sum_{i=1}^n ix^4 + \text{random}[0,1]$	[-1.28,1.28]	30	0

Table 7, the H_0 hypothesis is rejected since $X_R^2 > X_{table}^2$ for the F₁-F₈, F₁₂, F₁₃, F₁₅, F₂₁-F₂₄, and F₂₇-F₃₃ functions. Considering these results, it can be interpreted that there is a statistical difference between the results obtained from the proposed CSHOs and the original SHO. Since the results obtained with the CSHOs and the original SHO are the same for the remaining functions, it can be said that there is no statistical difference between them.

The performance of the proposed CSHO has been compared with SHO as well as SCA, SSA, WOA, and PSO methods in the literature. The values of Avg and Std obtained from these optimization methods for the F₁-F₃₃ functions are given in Table 8. According to the average values in Table 8, the proposed CSHO outperformed other optimization methods in 26 of 33 functions. If this is stated as a percentage, CSHO gave better results than SSA, SCA, WOA, and PSO methods in 78.78% of 33 different benchmark functions. The convergence curves are depicted in Fig. 8. Fast convergence to the optimal solution is important for optimization methods. In the optimization process of 33 comparison functions, it is observed that the convergence speed and accuracy of the CSHO algorithm are better than the original SHO, SCA, SSA, WOA, and PSO methods when the number of iterations is the same in the early iteration. Therefore, the proposed CSHO algorithm is promising for solving optimization problems.

The distributional properties of the data can be evaluated using boxplot analysis. Fig. 9 shows the boxplots for the proposed CSHO, original SHO, SCA, SSA, WOA, and PSO algorithms on benchmark functions. To evaluate the data distribution of the obtained results in detail, it is shown with boxplots. When the boxplots in Fig. 9 are examined, it is seen that the boxplots of results obtained from the proposed CSHO method are narrow compared to the other methods. The narrow box plots show that there is a high agreement between the results obtained with CSHO and that the CSHO algorithm is reliable.

Additionally, the CSHO algorithm's boxplot values are lower than those of other algorithms. Wilcoxon signed-rank test is applied at a 95% confidence interval to statistically confirm the effectiveness of the proposed CSHO over competitive other optimization methods. The proposed CSHO algorithm is determined as the control algorithm. It has been pairwise compared with the original SHO, SCA, SSA, WOA, and PSO algorithms. The p-values calculated for the Wilcoxon signed-rank test are given in Table 9. In this table, '↑' means the proposed CSHO is significantly better than the competitive optimization algorithm, while '↓' means the competitive optimization algorithm is significantly better than CSHO. '≈' indicates that the performance between the proposed CSHO and the competitive algorithm is insignificant.

According to Table 9, the proposed CSHO produced significantly better solutions than the original SHO in other functions except for F₉-F₁₁, and F₂₆. According to these 4 functions, the difference between the proposed CSHO and the original SHO is insignificant. Also, CSHO results are significantly better than SCA for all benchmark functions. In the case of SSA, the proposed CSHO achieved

Table 2

Description of multimodal benchmark functions.

(F _i) Function	Range	Dim	f _{min}
F ₈ (X) = $\sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	[-500,500]	30	428,989 × n
F ₉ (X) = $\sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5,12,5,12]	30	0
F ₁₀ (X) = $-20 \exp(-0.2(\frac{1}{n} \sum_{i=1}^n x_i^2)^{0.5}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	[-32,32]	30	0
F ₁₁ (X) = $\frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,600]	30	0
F ₁₂ (X) = $\frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 \left[1 + 10 \sin^2(\pi y_{i+1}) \right] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	[-50,50]	30	0
$y_i = 1 + \frac{x_i+1}{4} u(x_i, a, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
F ₁₃ (X) = $0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 \left[1 + \sin^2(3\pi x_i + 1) \right] + (x_n - 1)^2 \left[1 + \sin^2(2\pi x_n) \right] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	[-50,50]	30	0

Table 3

Description of fixed-dimension multimodal benchmark functions.

(F _i) Function	Range	Dim	f _{min}
F ₁₄ (X) = $\left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	[-65,65]	2	1
F ₁₅ (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	2	1
F ₁₆ (X) = $4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_4^2$	[-5,5] ²	2	-1.0316
F ₁₇ (X) = $(x_2 - \frac{5}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$	[-5,5] ²	2	0.398
F ₁₈ (X) = $[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	[-2,2]	2	3
F ₁₉ (X) = $-\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2\right)$	[1,3]	3	-3.86
F ₂₀ (X) = $-\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2\right)$	[0,1]	6	-3.32
F ₂₁ (X) = $-\sum_{i=1}^5 \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	[0,10]	4	-10.1532
F ₂₂ (X) = $-\sum_{i=1}^7 \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	[0,10]	4	-10.4028
F ₂₃ (X) = $-\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	[0,10]	4	-105.363

Table 4

CEC2019 benchmark functions.

(F _i) Function	Range	Dim	f _{min}
F ₂₄ - Problem of Storn's Chebyshev Polynomial Fitting	[-8192,8192]	9	1
F ₂₅ - Problem of Inverse Hilbert Matrix	[-16384, 16384]	16	1
F ₂₆ - Lennard-Jones Minimum Energy Cluster	[-4,4]	18	1
F ₂₇ - Funciton of Rastrigin	[-100,100]	10	1
F ₂₈ - Funciton of Griewangk	[-100,100]	10	1
F ₂₉ - Function of Weierstrass	[-100,100]	10	1
F ₃₀ - Function of Modified Schwefels	[-100,100]	10	1
F ₃₁ - Function of Expanded Schaffer F6	[-100,100]	10	1
F ₃₂ - Function of Happy Cat	[-100,100]	10	1
F ₃₃ - Function of Ackley	[-100,100]	10	1

Table 5

Parameters settings of the SHO and selected algorithms.

Algorithm	Parameters	Value
SHO	r ₁	0
	Probability of success r ₁	0.1
SCA	Number of elite	2
SSA	Balance parameter (c ₁)	2e ^{-(4l/L)²}
WOA	a	decreases linearly from 2 to 0
	b	1
PSO	C ₁	1.5
	C ₂	2.0
	Inertia weight	1

significantly better results except for the 8 functions (F₁₂, F₁₄, F₁₈, F₁₉, F₂₂, F₂₃, F₂₅, and F₂₆). Likewise, in the case of WOA and PSO, CSHO performed better except for 5 (F₉, F₁₁, F₁₄, F₁₈, and F₂₆) and 6 (F₁₃, F₁₉, F₂₃, F₂₅, and F₂₆) functions, respectively.

5.2. Sensitivity analysis

Typically, an algorithm's parameter sensitivity analysis utilized full factorial and fractional factorial design methodologies. The high computational costs of these methodologies represent a significant challenge for this situation. In addition, metaheuristic algorithms are stochastic. Therefore, these algorithms produce different solutions each time they are run. Given the full factorial design that analysis all parameter combinations for all test functions, it's essentially impossible to run them multiple times during the initial study. As a result, sensitivity analysis has been performed to assess the impact of value l of CSHO, the number of iterations, and the number of search agents on the algorithm. From among the CSHOs, the Sine map-based SHO (CSHO7) was selected for the sensitivity analysis. This is because CSHO7 has achieved more success in Benchmark functions than other CSHO.

Firstly, the effect of the l value CSHO's specific parameter on the performance of the algorithm was analyzed. The value of l was taken as 0.03 and 0.05 in the analysis. The effect of different values of l on the performance of the CSHO is given in Table 10. According to Table 10, it is seen that CSMA achieves better results except for 8 functions when the value of l is 0.03.

Then, the effect of the change in the number of iterations on the performance of the algorithm was evaluated. In the analysis, the

Table 6

Comparison results of CSHOs and SHO on Benchmark functions.

Alg.	Avg	Std	MR	Avg	Std	MR
F₁						
CSHO1	7.40E-167	0	4.30	1.14E-91	1.12E-91	4.10
CSHO2	6.96E-157	0	9.00	4.22E-82	3.40E-82	9.00
CSHO3	3.17E-144	2.73E-144	10.00	6.20E-81	4.64E-81	10.00
CSHO4	6.78E-167	0	5.20	1.51E-89	9.45E-90	6.20
CSHO5	2.09E-170	0	3.00	1.36E-95	1.83E-95	3.00
CSHO6	5.67E-161	0	8.00	9.36E-88	8.8E-88	8.00
CSHO7	1.47E-166	0	5.90	2.74E-90	3.01E-90	4.90
CSHO8	1.87E-177	0	1.90	1.20E-100	1.5E-100	2.00
CSHO9	1.55E-177	0	1.10	4.35E-102	5.7E-102	1.00
CSHO10	4.91E-165	0	6.60	1.40E-88	1.44E-88	6.80
SHO	5.17E-142	4.96E-142	11.00	1.24E-77	1.44E-77	11.00
F₂						
CSHO1	1.41E-131	4.45E-131	3.60	2.17E-74	2.76E-74	3.60
CSHO2	2.51E-122	2.90E-122	9.00	3.84E-65	3.05E-65	9.00
CSHO3	6.97E-106	5.30E-106	10.00	4.16E-59	3.03E-59	10.00
CSHO4	1.12E-131	2.20E-131	5.00	3.45E-70	2.92E-70	5.70
CSHO5	1.89E-137	3.20E-137	2.40	5.63E-75	1.02E-74	3.40
CSHO6	8.59E-131	6.00E-131	6.10	1.77E-69	1.89E-69	7.00
CSHO7	4.35E-128	1.3E-127	7.00	2.19E-69	2.42E-69	7.20
CSHO8	6.26E-143	1.9E-142	1.10	1.86E-78	2.62E-78	1.40
CSHO9	8.50E-135	1.5E-134	3.00	1.07E-76	1.62E-76	1.60
CSHO10	1.50E-127	2.6E-127	7.80	3.49E-70	3.29E-70	6.10
SHO	1.19E-97	1.47E-97	11.00	1.03E-56	1.59E-56	11.00
F₃						
CSHO1	2.72E + 01	5.24E-02	4.05	2.26E + 00	4.15E-01	4.40
CSHO2	2.74E + 01	3.37E-01	5.85	2.23E + 00	4.19E-01	4.70
CSHO3	2.72E + 01	3.82E-02	5.75	2.37E + 00	7.90E-02	7.20
CSHO4	2.73E + 01	3.06E-01	5.95	2.07E + 00	3.06E-01	3.30
CSHO5	2.73E + 01	2.05E-01	5.60	2.47E + 00	2.80E-01	7.50
CSHO6	2.74E + 01	3.42E-01	6.75	2.11E + 00	4.33E-01	4.30
CSHO7	2.73E + 01	2.44E-01	5.85	2.17E + 00	2.36E-01	9.95
CSHO8	2.73E + 01	1.30E-01	5.80	2.54E + 00	2.63E-01	8.35
CSHO9	2.72E + 01	4.85E-02	5.50	2.44E + 00	6.70E-02	8.10
CSHO10	2.72E + 01	8.70E-02	4.00	2.06E + 00	2.37E-01	3.20
SHO	2.83E + 01	3.78E-01	10.90	3.49E + 00	1.89E-01	11.00
F₅						
CSHO1	5.55E-06	2.84E-06	3.40	2.26E + 00	4.15E-01	4.40
CSHO2	6.34E-06	2.24E-06	3.70	-8.96E + 03	2.28E + 02	2.60
CSHO3	4.19E-05	2.7E-05	8.00	-8.68E + 03	4.24E + 02	3.65
CSHO4	5.82E-06	1.73E-06	3.60	-6.62E + 03	4.44E + 02	9.05
CSHO5	5.34E-06	2.14E-06	3.30	-8.63E + 03	2.15E + 02	4.30
CSHO6	6.26E-06	2.8E-06	3.70	-8.93E + 03	2.46E + 02	2.60
CSHO7	6.03E-05	3.04E-05	8.80	-9.00E + 03	4.38E + 02	2.60
CSHO8	5.97E-05	3.08E-05	8.70	-8.01E + 03	3.16E + 02	7.25
CSHO9	5.15E-05	3.38E-05	8.30	-6.27E + 03	4.93E + 02	9.90
CSHO10	5.67E-06	2.86E-06	3.50	-7.78E + 03	6.44E + 02	7.30
SHO	2.09E-04	8.97E-05	11.00	-5.70E + 03	1.48E + 02	11.00
F₇						
CSHO1	5.55E-06	2.84E-06	3.40	F₈		
CSHO2	6.34E-06	2.24E-06	3.70	-8.96E + 03	2.28E + 02	2.60
CSHO3	4.19E-05	2.7E-05	8.00	-8.68E + 03	4.24E + 02	3.65
CSHO4	5.82E-06	1.73E-06	3.60	-6.62E + 03	4.44E + 02	9.05
CSHO5	5.34E-06	2.14E-06	3.30	-8.93E + 03	2.15E + 02	4.30
CSHO6	6.26E-06	2.8E-06	3.70	-8.34E + 03	2.27E + 02	5.75
CSHO7	6.03E-05	3.04E-05	8.80	-9.00E + 03	4.38E + 02	2.60
CSHO8	5.97E-05	3.08E-05	8.70	-8.01E + 03	3.16E + 02	7.25
CSHO9	5.15E-05	3.38E-05	8.30	-6.27E + 03	4.93E + 02	9.90
CSHO10	5.67E-06	2.86E-06	3.50	-7.78E + 03	6.44E + 02	7.30
SHO	2.09E-04	8.97E-05	11.00	-5.70E + 03	1.48E + 02	11.00
F₉						
CSHO1	0	0	6.00	F₁₀		
CSHO2	0	0	6.00	8.88E-16	0	6.00
CSHO3	0	0	6.00	8.88E-16	0	6.00
CSHO4	0	0	6.00	8.88E-16	0	6.00
CSHO5	0	0	6.00	8.88E-16	0	6.00
CSHO6	0	0	6.00	8.88E-16	0	6.00
CSHO7	0	0	6.00	8.88E-16	0	6.00
CSHO8	0	0	6.00	8.88E-16	0	6.00
CSHO9	0	0	6.00	8.88E-16	0	6.00
CSHO10	0	0	6.00	8.88E-16	0	6.00
SHO	0	0	6.00	8.88E-16	0	6.00
F₁₁						
CSHO1	0	0	6.00	F₁₂		
CSHO2	0	0	6.00	7.46E-02	1.87E-02	4.50
CSHO3	0	0	6.00	8.19E-02	9.61E-03	5.60
CSHO4	0	0	6.00	1.43E-01	4.29E-02	8.90
CSHO5	0	0	6.00	7.96E-02	1.52E-02	4.00
CSHO6	0	0	6.00	6.20E-02	1.83E-02	5.30
CSHO7	0	0	6.00	6.02E-02	2.05E-02	3.50
CSHO8	0	0	6.00	7.95E-02	1.39E-02	5.80
CSHO9	0	0	6.00	1.42E-01	1.85E-02	9.50
CSHO10	0	0	6.00	7.83E-02	2.05E-02	4.90
SHO	0	0	6.00	2.45E-01	6.44E-02	10.80

(continued on next page)

Table 6 (continued)

Alg.	Avg	Std	MR	Avg	Std	MR
	F₁₃			F₁₄		
CSHO1	1.28E + 00	1.66E-01	5.20	9.98E-01	3.02E-06	6.00
CSHO2	1.23E + 00	1.56E-01	4.65	9.98E-01	4.53E-06	6.00
CSHO3	1.86E + 00	8.40E-02	9.30	9.98E-01	3.93E-06	6.00
CSHO4	1.14E + 00	1.23E-01	3.65	9.98E-01	3.32E-06	6.00
CSHO5	1.54E + 00	2.14E-01	7.30	9.98E-01	5.14E-06	6.00
CSHO6	1.25E + 00	2.02E-01	5.25	9.98E-01	4.23E-06	6.00
CSHO7	7.73E-01	1.40E-01	1.15	9.98E-01	3.63E-06	6.00
CSHO8	1.27E + 00	2.05E-01	5.35	9.98E-01	4.83E-06	6.00
CSHO9	1.89E + 00	1.23E-01	9.70	9.98E-01	5.44E-06	6.00
CSHO10	1.14E + 00	1.72E-01	3.45	9.98E-01	5.74E-06	6.00
SHO	2.33E + 00	2.00E-01	11.00	9.98E-01	6.04E-06	6.00
	F₁₅			F₁₆		
CSHO1	3.08E-04	1.03E-07	3.85	-1.03E + 00	3.12E-06	6.00
CSHO2	3.08E-04	2.16E-07	4.25	-1.03E + 00	4.37E-06	6.00
CSHO3	3.09E-04	9.76E-01	8.30	-1.03E + 00	4.06E-06	6.00
CSHO4	3.08E-04	8.11E-07	5.45	-1.03E + 00	5.93E-06	6.00
CSHO5	3.08E-04	4.22E-07	5.05	-1.03E + 00	3.75E-06	6.00
CSHO6	3.08E-04	4.08E-07	4.60	-1.03E + 00	5.62E-06	6.00
CSHO7	3.08E-04	3.03E-07	4.65	-1.03E + 00	5.31E-06	6.00
CSHO8	3.08E-04	4.98E-07	5.35	-1.03E + 00	5.00E-06	6.00
CSHO9	3.11E-04	2.62E-06	8.85	-1.03E + 00	4.69E-06	6.00
CSHO10	3.08E-04	7.36E-07	4.75	-1.03E + 00	6.25E-06	6.00
SHO	4.39E-04	4.04E-05	10.90	-1.03E + 00	3.44E-06	6.00
	F₁₇			F₁₈		
CSHO1	3.98E-01	1.20E-06	6.00	3.00E + 00	9.08E-06	6.00
CSHO2	3.98E-01	7.23E-06	6.00	3.00E + 00	4.54E-05	6.00
CSHO3	3.98E-01	1.08E-05	6.00	3.00E + 00	1.82E-05	6.00
CSHO4	3.98E-01	2.41E-06	6.00	3.00E + 00	1.82E-05	6.00
CSHO5	3.98E-01	4.82E-06	6.00	3.00E + 00	7.27E-05	6.00
CSHO6	3.98E-01	9.64E-06	6.00	3.00E + 00	5.45E-05	6.00
CSHO7	3.98E-01	8.43E-06	6.00	3.00E + 00	6.36E-05	6.00
CSHO8	3.98E-01	6.02E-06	6.00	3.00E + 00	3.63E-05	6.00
CSHO9	3.98E-01	5.42E-06	6.00	3.00E + 00	5.00E-05	6.00
CSHO10	3.98E-01	3.61E-06	6.00	3.00E + 00	1.36E-05	6.00
SHO	3.98E-01	4.58E-06	6.00	3.00E + 00	6.81E-05	6.00
	F₁₉			F₂₀		
CSHO1	-3.86E + 00	9.43E-05	6.00	-3.32E + 00	3.33E-04	6.00
CSHO2	-3.86E + 00	3.84E-04	6.00	-3.32E + 00	8.46E-04	6.00
CSHO3	-3.86E + 00	2.44E + 00	6.00	-3.32E + 00	9.74E-04	6.00
CSHO4	-3.86E + 00	1.48E-04	6.00	-3.32E + 00	4.74E-04	6.00
CSHO5	-3.86E + 00	1.48E-04	6.00	-3.32E + 00	1.03E-04	6.00
CSHO6	-3.86E + 00	6.02E-04	6.00	-3.32E + 00	7.28E-04	6.00
CSHO7	-3.86E + 00	6.75E-05	6.00	-3.32E + 00	7.81E-04	6.00
CSHO8	-3.86E + 00	6.87E-04	6.00	-3.32E + 00	5.29E-03	6.00
CSHO9	-3.86E + 00	8.72E-04	6.00	-3.32E + 00	1.42E-02	6.00
CSHO10	-3.86E + 00	1.43E-04	6.00	-3.32E + 00	1.14E-03	6.00
SHO	-3.86E + 00	9.88E-04	6.00	-3.32E + 00	1.77E-03	6.00
	F₂₁			F₂₂		
CSHO1	-1.01E + 01	9.19E-02	3.10	-1.03E + 01	7.69E-02	4.20
CSHO2	-9.91E + 00	2.27E-01	4.80	-1.02E + 01	1.44E-01	5.70
CSHO3	-9.80E + 00	2.39E-01	6.50	-1.02E + 01	1.93E-01	6.20
CSHO4	-9.83E + 00	2.90E-01	6.20	-1.02E + 01	2.28E-01	6.30
CSHO5	-8.05E + 00	6.37E + 00	9.30	-1.03E + 01	6.24E-02	2.60
CSHO6	-9.92E + 00	1.67E-01	3.80	-1.02E + 01	1.70E-01	6.20
CSHO7	-1.01E + 01	5.12E-02	3.10	-1.04E + 01	5.05E-02	2.50
CSHO8	-9.81E + 00	3.05E-01	6.20	-1.01E + 01	2.67E-01	7.20
CSHO9	-9.13E + 00	5.14E-01	6.90	-9.39E + 00	4.23E-01	10.00
CSHO10	-9.88E + 00	3.36E-01	5.20	-1.02E + 01	1.87E-01	4.90
SHO	-5.05E + 00	5.91E-03	10.90	-6.48E + 00	2.28E + 00	10.20
	F₂₃			F₂₄		
CSHO1	-1.04E + 01	1.71E-01	3.20	4.19E + 04	5.25E + 02	6.40
CSHO2	-1.03E + 01	1.08E-01	5.60	4.16E + 04	4.68E + 02	4.70
CSHO3	-1.02E + 01	5.08E-01	6.85	4.16E + 04	5.84E + 02	4.40
CSHO4	-1.03E + 01	2.83E-01	5.80	4.18E + 04	7.84E + 02	5.70
CSHO5	-1.04E + 01	2.38E-01	4.05	4.16E + 04	8.64E + 02	5.30
CSHO6	-1.02E + 01	3.75E-01	7.05	4.18E + 04	7.50E + 02	5.90
CSHO7	-1.05E + 01	6.96E-02	3.00	4.11E + 04	1.20E + 03	4.10
CSHO8	-1.01E + 01	2.91E-01	7.45	4.16E + 04	4.39E + 02	4.30
CSHO9	-1.02E + 01	1.77E-01	6.40	4.19E + 04	3.53E + 02	5.60
CSHO10	-1.02E + 01	1.76E-01	6.10	4.26E + 04	4.49E + 02	8.60
SHO	-5.99E + 00	2.22E + 00	10.50	4.87E + 04	2.13E + 03	11.00

Table 6 (continued)

Alg.	Avg	Std	MR	Avg	Std	MR
F₂₅						
CSHO1	1.73E + 01	8.86E-04	6.00	1.27E + 01	3.85E-05	6.00
CSHO2	1.73E + 01	1.87E-03	6.00	1.27E + 01	4.07E-05	6.00
CSHO3	1.73E + 01	3.75E-05	6.00	1.27E + 01	5.77E-05	6.00
CSHO4	1.73E + 01	1.23E-03	6.00	1.27E + 01	4.23E-05	6.00
CSHO5	1.73E + 01	1.75E-03	6.00	1.27E + 01	4.62E-05	6.00
CSHO6	1.73E + 01	1.98E-03	6.00	1.27E + 01	4.94E-05	6.00
CSHO7	1.73E + 01	1.11E-03	6.00	1.27E + 01	5.38E-05	6.00
CSHO8	1.73E + 01	1.68E-03	6.00	1.27E + 01	5.77E-05	6.00
CSHO9	1.73E + 01	1.70E-03	6.00	1.27E + 01	6.15E-05	6.00
CSHO10	1.73E + 01	1.50E-03	6.00	1.27E + 01	6.54E-05	6.00
SHO	1.73E + 01	1.67E-03	6.00	1.27E + 01	6.92E-05	6.00
F₂₇						
CSHO1	3.42E + 01	4.64E + 00	2.67	1.18E + 00	2.41E-02	2.20
CSHO2	4.50E + 01	5.24E + 00	5.44	1.34E + 00	1.58E-02	9.80
CSHO3	3.88E + 01	7.43E + 00	3.56	1.17E + 00	7.10E-02	1.80
CSHO4	5.59E + 01	5.66E + 00	7.56	1.25E + 00	2.58E-02	5.00
CSHO5	3.70E + 01	4.24E + 00	2.89	1.23E + 00	4.20E-02	3.90
CSHO6	5.34E + 01	7.72E + 00	7.44	1.28E + 00	4.18E-02	6.70
CSHO7	3.34E + 01	5.06E + 00	1.78	1.23E + 00	3.21E-02	3.90
CSHO8	5.51E + 01	4.45E + 00	7.56	1.28E + 00	3.89E-02	6.90
CSHO9	9.03E + 01	5.78E + 00	10.00	1.28E + 00	3.79E-02	6.70
CSHO10	4.89E + 01	5.59E + 00	6.11	1.31E + 00	3.63E-02	8.10
SHO	1.24E + 02	1.81E + 01	11.00	1.69E + 00	3.06E-02	11.00
F₂₉						
CSHO1	5.14E + 00	6.06E-01	6.80	5.08E + 00	2.68E-01	8.10
CSHO2	5.26E + 00	4.88E-01	8.25	4.83E + 00	5.27E-01	6.60
CSHO3	4.70E + 00	3.75E-01	3.70	3.82E + 00	5.29E-01	1.80
CSHO4	5.04E + 00	4.20E-01	6.50	4.64E + 00	2.23E-01	4.50
CSHO5	4.14E + 00	8.91E-01	3.20	4.66E + 00	6.27E-01	6.30
CSHO6	5.04E + 00	3.73E-01	6.60	4.90E + 00	2.88E-01	7.00
CSHO7	4.72E + 00	2.44E-01	5.00	4.61E + 00	6.14E-01	5.10
CSHO8	4.83E + 00	2.87E-01	5.60	4.83E + 00	2.85E-01	5.90
CSHO9	4.93E + 00	2.40E-01	5.65	4.73E + 00	3.96E-01	5.20
CSHO10	4.70E + 00	7.88E-01	3.90	4.51E + 00	5.07E-01	4.50
SHO	7.91E + 00	1.30E + 00	10.80	7.55E + 00	1.15E + 00	11.00
F₃₁						
CSHO1	4.00E + 00	2.89E-01	6.60	2.01E + 01	3.67E-02	6.00
CSHO2	3.79E + 00	2.38E-01	3.70	2.00E + 01	1.28E-03	2.00
CSHO3	3.81E + 00	2.98E-01	3.90	2.00E + 01	3.57E-03	1.80
CSHO4	3.85E + 00	3.50E-01	4.40	2.01E + 01	5.35E-02	8.75
CSHO5	4.03E + 00	1.13E-01	7.00	2.00E + 01	2.78E-02	5.00
CSHO6	3.91E + 00	3.26E-01	6.20	2.01E + 01	3.81E-02	6.90
CSHO7	3.87E + 00	2.61E-01	5.60	2.01E + 01	5.58E-02	6.00
CSHO8	3.95E + 00	2.55E-01	5.90	2.01E + 01	5.70E-02	6.70
CSHO9	4.00E + 00	2.78E-01	7.00	2.01E + 01	3.84E-02	7.75
CSHO10	3.86E + 00	1.45E-01	5.50	2.01E + 01	4.96E-02	5.70
SHO	5.06E + 00	6.17E-01	10.20	2.01E + 01	6.49E-02	9.40
F₃₃						
CSHO1	4.91E + 00	1.98E + 00	1.50			
CSHO2	9.29E + 00	3.30E + 00	3.40			
CSHO3	5.97E + 00	3.48E-01	2.20			
CSHO4	1.85E + 01	4.77E + 00	7.20			
CSHO5	2.00E + 01	9.82E-02	7.40			
CSHO6	1.55E + 01	5.82E + 00	6.00			
CSHO7	1.83E + 01	5.23E + 00	6.90			
CSHO8	1.81E + 01	4.12E + 00	6.40			
CSHO9	1.83E + 01	2.87E + 00	6.70			
CSHO10	1.99E + 01	3.53E-01	7.30			
SHO	2.02E + 01	8.14E-02	11.00			

number of iterations was determined as 100, 500, and 1000. Based on these values, the obtained results from the CSHO are given in **Table 11**. When **Table 11** is examined, it is observed that the performance obtained from CSHO increases except for 7 functions as the number of iterations increases.

The impact of increasing search agents on the CSHO's performance is given in **Table 12**. 10, 30, and 100 search agents have been selected as number the of agents. It can be seen that, except for 4 functions, the performance gained through CSHO improves as the population grows examination of **Table 12**.

5.3. Applications of CSHO on engineering design problems

This section evaluates the efficiency of the proposed CSHO in solving four real engineering design problems, namely welded beam design, pressure vessel design, tension/compression spring design, and speed reducer design problems. The results obtained by CSHO are also compared with some metaheuristic algorithms in the literature.

Welded Beam Design Problem: In this problem, the objective function is to produce a welded beam with minimum cost accord-

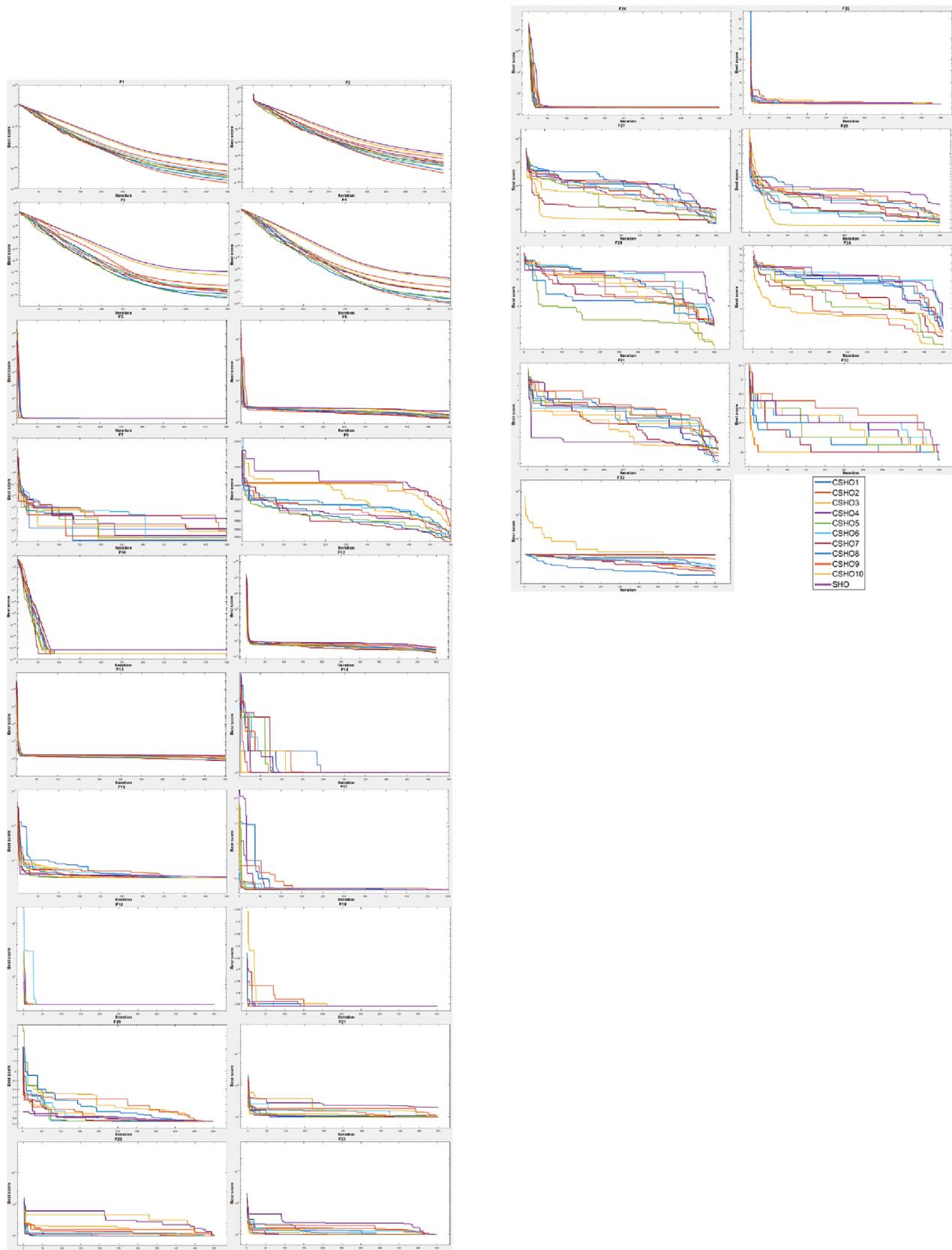


Fig. 7. Convergence curves of CSHOs and original SHO.

Table 7

The results of the Friedman test.

F_i	F_1	F_2	F_3	F_4	F_5	F_6	F_7
18.307 H_0	95.036 Reject	99.545 Reject	97.255 Reject	95.982 Reject	33.693 Reject	59.339 Reject	78.236 Reject
18.307 H_0	87.279 Reject	F_8 -	F_9 -	F_{10} -	F_{11} -	F_{12} 59.945 Reject	F_{13} 82.171 Reject
18.307 H_0	68.617 Reject	F_{15} -	F_{16} -	F_{17} -	F_{18} -	F_{19} -	F_{20} -
18.307 H_0	57.818 Reject	F_{22} 41.138 Reject	F_{23} 40.527 Reject	F_{24} -	F_{25} -	F_{26} -	F_{27} 84.964 Reject
18.307 H_0	43.815 Reject	F_{29} 49.509 Reject	F_{30} 29.745 Reject	F_{31} -	F_{32} 52.995 Reject	F_{33} 66.364 Reject	F_{21} 54.345 Reject

ing to certain constraints. The cost function of the problem and the seven constraint functions associated with the cost function are given in Eq. (26) and Eq. (27), respectively. The welded beam problem shown in Fig. 10 has four decision variables. From these parameters, h represents the weld thickness, l represents the length of the welded bar, t represents the height of the bar, and b represents the thickness of the bar.

$$x = [x_1 \ x_2 \ x_3 \ x_4] = [h \ l \ t \ b],$$

$$\text{Minimize } f(x) = 1.10471x_1^4x_2 + 0.04811x_3x_4(14 + x_2) \quad (26)$$

$$\left. \begin{array}{l} g_1(x) = \tau(x) + \tau_{\max} \leq 0 \\ g_2(x) = \sigma(x) + \sigma_{\max} \leq 0 \\ g_3(x) = \delta(x) + \delta_{\max} \leq 0 \\ g_4(x) = x_1 - x_4 \leq 0 \\ g_5(x) = P - P_c(x) \leq 0 \\ g_6(x) = 0.125 - x_1 \leq 0, \\ g_7(x) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) \\ \quad - 5 \leq 0, \\ 0.1 \leq x_1, x_4 \leq 2.0, \quad 0.1 \leq x_2, x_3 \leq 10.0 \end{array} \right\} \quad (27)$$

where;

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

$$\begin{aligned} M &= P\left(L + \frac{x_2}{2}\right), \quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \quad J \\ &= 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^3}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \end{aligned}$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2}, \quad \delta(x) = \frac{4PL^3}{Ex_4x_3^2}, \quad P_c(x) = \frac{4.013E\sqrt{\frac{x_2^2x_4}{36}}}{L^2} \left\{1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right\}$$

where; $P = 6000lb$, $L = 14in$, $E = 30 \times 10^6psi$, $G = 12 \times 10^6psi$, $\tau_{\max} = 13,600psi$, $\sigma(x) = 30,000psi$, $\delta(x) = 0.25in$

In the welded beam design problem, SHO and CSHOs were tested with a population number of 30 and a maximum number of iterations of 500. In addition, the algorithms have been executed 30 times. In the literature, many different metaheuristic optimization algorithms have solved this problem to find the best result. CSHOs have been compared with some of these methods in the literature. These methods are AOA, GEO, SCA, GSA, MVO, SHOA, EO, TEO, SSA, and WSA. The comparison of the optimum cost values obtained from these algorithms is listed in Table 13. According to

Table 13, it is seen that the minimum cost value for the welded beam design problem is obtained with Sinusoidal map-based SHO when compared to other metaheuristic algorithms. In addition, performance comparisons of the proposed CSHOs, the original SHO, and other metaheuristic algorithms in the literature for this problem are graphically presented in Fig. 11.

Pressure Vessel Design Problem: The purpose of the pressure vessel design problem shown in Fig. 12 is to minimize the production cost by taking into account the synthesis, welding, and material cost constraints. This problem has four decision variables: T_s -thickness, T_h -head thickness, R -inner radius, and L -head independent length of the cylinder. The cost function of the pressure vessel design problem and the four specific constraint functions are given in Eqs. (28) and (29), respectively.

$$x = [x_1 \ x_2 \ x_3 \ x_4] = [T_s \ T_h \ R \ L]$$

$$\begin{aligned} \text{Minimize } f(x) &= 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 \\ &\quad + 19.84x_1^2x_3 \end{aligned} \quad (28)$$

$$\left. \begin{array}{l} g_1(x) = -x_1 + 0.0193x_3 \leq 0 \\ g_2(x) = -x_2 + 0.00954x_3 \leq 0 \\ g_3(x) = -\pi x_3^2x_4 - (4/3)\pi x_3^3 + 1296000 \leq 0 \\ g_4(x) = x_4 - 240 \leq 0 \\ 0.0625 \leq x_1, x_2 \leq 99 \times 0.0625, \quad 10 \leq x_3, x_4 \leq 200 \end{array} \right\} \quad (29)$$

In the pressure vessel design problem, SHO and CSHOs were tested with a population number of 30 and a maximum number of iterations of 500. In addition, the algorithms have been executed 30 times. In the literature, many different metaheuristic optimization algorithms have solved this problem to find the best result. CSHOs have been compared with some of these methods in the literature. These methods are WWO, ALO, RFO, SOA, HHO, WEO, GWO, MBA, SMA, and MRFO. The comparison of the optimum cost values obtained from these algorithms is listed in Table 14. According to Table 14, it is seen that the minimum cost value for the pressure vessel design problem is obtained with Singer map-based SHO when compared to other metaheuristic algorithms. In addition, performance comparisons of the proposed CSHOs, the original SHO, and other metaheuristic algorithms in the literature for this problem are graphically presented in Fig. 13.

Tension/Compression Spring Design Problem: The main objective of the spring design problem is to minimize the total weight of the tension/ compression spring. As shown in Fig. 14, this problem consists of three design parameters. From these parameters, d represents the diameter of the wire, D represents the diameter of the mean coil, and N represents the number of active coils. The objec-

Table 8

Result comparison of CSHO with other metaheuristic algorithms.

Alg.	Avg	Std	Avg	Std	Avg	Std
	F₁		F₂		F₃	
CSHO	1.55E-177	0.00E + 00	4.35E-102	5.7E-102	6.26E-143	1.9E-142
SHO	5.17E-142	4.96E-142	1.24E-77	1.44E-77	1.19E-97	1.47E-97
SCA	2.15E-12	3.99E-12	4.71E-10	5.23E-10	5.60E-04	8.87E-04
SSA	1.71E-07	2.19E-07	1.05E-05	4.98E-06	3.59E-07	9.74E-07
WOA	2.09E-76	3.46E-76	1.69E-53	2.25E-53	4.56E + 04	1.31E + 04
PSO	7.12E-08	1.56E-07	3.40E-03	4.50E-03	5.31E + 01	8.83E + 01
	F₄		F₅		F₆	
CSHO	1.86E-78	2.62E-78	2.72E + 01	8.70E-02	2.06E + 00	2.37E-01
SHO	1.03E-56	1.59E-56	2.83E + 01	3.78E-01	3.49E + 00	1.89E-01
SCA	5.63E-04	5.35E-04	7.41E + 00	4.06E-01	3.47E-01	1.37E-01
SSA	1.78E-05	3.15E-06	1.44E + 02	9.90E + 01	7.26E-10	2.47E-10
WOA	7.04E + 01	1.96E + 01	2.79E + 01	1.93E-01	3.76E-01	3.40E-01
PSO	1.20E + 00	1.76E + 00	1.59E + 01	1.29E + 01	4.21E-09	5.38E-09
	F₇		F₈		F₉	
CSHO	5.34E-06	2.14E-06	-9.00E + 03	4.38E + 02	0	0
SHO	2.09E-04	8.97E-05	-5.70E + 03	1.48E + 02	0	0
SCA	5.12E-04	3.00E-04	-2.14E + 03	1.51E + 02	2.38E-12	4.88E-12
SSA	1.73E-02	3.62E-03	-2.86E + 03	2.96E + 02	1.92E + 01	8.60E + 00
WOA	3.63E-03	2.42E-03	-1.21E + 04	4.36E + 02	0	0
PSO	3.23E-02	2.10E-02	-1.26E + 04	0	2.30E + 01	1.09E + 01
	F₁₀		F₁₁		F₁₂	
CSHO	8.88E-16	0	0	0	6.02E-02	2.05E-02
SHO	8.88E-16	0	0	0	2.45E-01	6.44E-02
SCA	1.24E-07	2.73E-07	5.37E-11	8.15E-11	1.09E-01	0.022914
SSA	6.04E-01	8.82E-01	2.06E-01	9.63E-02	1.13E + 00	1.21E + 00
WOA	4.44E-15	0	0	0	1.91E-02	6.85E-03
PSO	8.90E-01	1.44E + 00	2.71E-03	8.57E-03	1.30E-08	3.73E-08
	F₁₃		F₁₄		F₁₅	
CSHO	7.73E-01	1.40E-01	9.98E-01	3.02E-06	3.08E-04	1.03E-07
SHO	2.33E + 00	2.00E-01	9.98E-01	6.04E-06	4.39E-04	4.04E-05
SCA	3.16E-01	7.81E-02	1.40E + 00	8.36E-01	9.47E-04	2.61E-04
SSA	4.39E-03	5.67E-03	9.98E-01	3.16E-05	9.41E-04	2.53E-04
WOA	4.66E-01	2.05E-01	3.93E + 00	4.72E + 00	1.18E-03	7.81E-04
PSO	7.73E-01	1.44E + 00	9.98E-01	0	4.64E-04	1.06E-04
	F₁₆		F₁₇		F₁₈	
CSHO	-1.03E + 00	3.12E-06	3.98E-01	1.20E-06	3.00E + 00	9.08E-06
SHO	-1.03E + 00	3.44E-06	3.98E-01	4.58E-06	3.00E + 00	6.81E-05
SCA	-1.03E + 00	6.99E-05	4.02E-01	3.12E-03	3.00E + 00	1.71E-04
SSA	-1.03E + 00	2.34E-16	3.98E-01	5.85E-17	3.00E + 00	3.16E-04
WOA	-1.03E + 00	6.32E-06	3.98E-01	3.16E-06	3.00E + 00	3.16E-05
PSO	-1.03E + 00	2.34E-16	3.98E-01	0	3.00E + 00	0
	F₁₉		F₂₀		F₂₁	
CSHO	-3.86E + 00	6.75E-05	-3.32E + 00	1.03E-04	-1.01E + 01	5.12E-02
SHO	-3.86E + 00	9.88E-04	-3.32E + 00	1.77E-03	-5.05E + 00	5.91E-03
SCA	-3.86E + 00	7.46E-04	-3.04E + 00	4.42E-02	-4.18E + 00	1.12E + 00
SSA	-3.86E + 00	3.16E-12	-3.23E + 00	6.86E-02	-5.39E + 00	3.43E + 00
WOA	-3.86E + 00	1.27E-03	-3.32E + 00	4.09E-03	-7.02E + 00	2.54E + 00
PSO	-3.86E + 00	0	-3.23E + 00	5.01E-02	-6.10E + 00	2.14E + 00
	F₂₂		F₂₃		F₂₄	
CSHO	-1.04E + 01	5.05E-02	-1.05E + 01	6.96E-02	4.11E + 04	1.20E + 03
SHO	-6.48E + 00	2.28E + 00	-5.99E + 00	2.22E + 00	4.87E + 04	2.13E + 03
SCA	-4.32E + 00	8.43E-01	-4.65E + 00	5.09E-01	4.57E + 09	5.35E + 09
SSA	-7.34E + 00	3.95E + 00	-6.61E + 00	3.50E + 00	1.22E + 10	9.82E + 09
WOA	-6.59E + 00	2.64E + 00	-5.80E + 00	2.63E + 00	2.47E + 10	2.90E + 10
PSO	-6.69E + 00	2.55E + 00	-6.63E + 00	3.49E + 00	5.20E + 08	5.19E + 08
	F₂₅		F₂₆		F₂₇	
CSHO	1.73E + 01	3.75E-05	1.27E + 01	3.85E-05	3.34E + 01	5.06E + 00
SHO	1.73E + 01	1.67E-03	1.27E + 01	6.92E-05	1.24E + 02	1.81E + 01
SCA	1.75E + 01	1.63E-02	1.27E + 01	9.19E-05	7.60E + 02	1.34E + 02
SSA	1.73E + 01	3.74E-15	1.27E + 01	1.87E-15	5.39E + 01	1.88E + 01
WOA	1.73E + 01	2.76E-03	1.27E + 01	1.49E-05	3.40E + 02	1.06E + 02
PSO	1.73E + 01	3.74E-15	1.27E + 01	1.87E-15	2.18E + 01	1.05E + 01

Table 8 (continued)

Alg.	Avg	Std	Avg	Std	Avg	Std
	F₂₈		F₂₉		F₃₀	
CSHO	1.17E + 00	7.10E-02	4.14E + 00	8.91E-01	3.82E + 00	5.29E-01
SHO	1.69E + 00	3.06E-02	7.91E + 00	1.30E + 00	7.55E + 00	1.15E + 00
SCA	2.26E + 00	7.31E-02	1.04E + 01	4.90E-01	7.63E + 02	1.05E + 02
SSA	1.29E + 00	1.30E-01	6.53E + 00	1.29E + 00	3.47E + 02	2.39E + 02
WOA	2.18E + 00	3.95E-01	8.97E + 00	9.32E-01	6.89E + 02	3.33E + 02
PSO	1.18E + 00	7.00E-02	1.04E + 01	9.01E-01	2.07E + 02	1.64E + 02
	F₃₁		F₃₂		F₃₃	
CSHO	3.79E + 00	2.38E-01	2.00E + 01	3.57E-03	4.91E + 00	1.98E + 00
SHO	5.06E + 00	6.17E-01	2.01E + 01	6.49E-02	2.02E + 01	8.14E-02
SCA	6.05E + 00	2.60E-01	6.10E + 01	1.43E + 01	2.05E + 01	4.50E-02
SSA	5.42E + 00	3.75E-01	2.73E + 00	2.48E-01	2.00E + 01	5.40E-02
WOA	6.08E + 00	7.21E-01	4.80E + 00	6.63E-01	2.02E + 01	1.01E-01
PSO	5.37E + 00	8.08E-01	2.38E + 00	3.02E-02	2.03E + 01	1.33E-01

tive function of the pressure vessel problem and its constraint functions are given in Eqs. (30) and (31), respectively.

$$x = [x_1 \ x_2 \ x_3] = [d, D, N]$$

$$\text{Minimize}_f(x) = (x_3 + 2)x_2x_1^2 \quad (30)$$

$$\left. \begin{array}{l} g_1(x) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \leq 0 \\ g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0 \\ g_3(x) = 1 - \frac{140.45x_3}{x_2^2 x_3} \leq 0 \\ g_4(x) = \frac{x_1 + x_2}{1.5} \leq 0 \\ 0.05 \leq x_1 \leq 2, 0.05 \leq x_1 \leq 2, 2 \leq x_3 \leq 15 \end{array} \right\} \quad (31)$$

In the tension/compression spring design problem, SHO and CSHOs were tested with a population number of 30 and a maximum number of iterations of 500. In addition, the algorithms have been executed 30 times. In the literature, many different metaheuristic optimization algorithms have solved this problem to find the best result. CSHOs have been compared with some of these methods in the literature. These methods are EBO, WOA, GSA, RO, MFO, SCA, MVO, GA3, SSA, and PSO. The comparison of the optimum cost values obtained from these algorithms is listed in Table 15. According to Table 15, the minimum values obtained with all proposed CSHOs are close to each other. According to these values, CSHOs achieved the minimum value compared to other metaheuristic algorithms. In addition, performance comparisons of the proposed CSHOs, the original SHO, and other metaheuristic algorithms in the literature for this problem are graphically presented in Fig. 15.

Speed Reducer Design Problem: The aim of the speed reducer design problem, schematically illustrated in Fig. 16, is to obtain a speed reducer with minimum weight. The design of the speed reducer problem is a more difficult problem as it has seven design variables. These variables are width (x_1), teeth module (x_2), the number of pinion teeth (x_3), the length of shaft 1 between bearings (x_4), the length of shaft 2 between bearings (x_5), the diameter of shaft 1 (x_6), and the diameter of shaft 2 (x_7). The objective function of the speed reducer design problem and its constraint functions are given in Eqs. (32) and (33), respectively.

$$\begin{aligned} \text{Minimize}_f(x) &= 0.785x_1x_2^2(3.333x_3^2 + 14.9334x_3 - 43.0934) \\ &\quad - 1.508x_1(x_6^2 + x_7^2) + 7.4777x_1(x_6^3 + x_7^3) \\ &\quad + 1.508x_1(x_4x_6^2 + x_5x_7^2) \end{aligned} \quad (32)$$

$$\left. \begin{array}{l} g_1(x) = \frac{27}{x_1 x_2^2 x_3} - 1 \\ g_2(x) = \frac{397.5}{x_1 x_2 x_3^2} - 1 \leq 0 \\ g_3(x) = \frac{1.93x_3^3}{x_1 x_3 x_6^4} - 1 \leq 0 \\ g_4(x) = \frac{1.93x_5^5}{x_1 x_3 x_7^6} - 1 \leq 0 \\ g_5(x) = \frac{1}{110x_6^3} \sqrt{\left(\frac{745x_4}{x_2 x_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0 \\ g_6(x) = \frac{1}{85x_7^3} \sqrt{\left(\frac{745x_5}{x_2 x_3}\right)^2 + 157.5 \times 10^6} - 1 \leq 0 \\ g_7(x) = \frac{x_2 x_3}{40} - 1 \leq 0 \\ g_8(x) = \frac{5x_2}{x_1} - 1 \leq 0 \\ g_9(x) = \frac{x_1}{12x_2} - 1 \leq 0 \\ g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0 \\ g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0 \\ 2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, \\ 7.3 \leq x_4 \leq 8.37, 8 \leq x_5 \leq 8.3, \\ 2.9 \leq x_6 \leq 3.9, 5.0 \leq x_7 \leq 5.5 \end{array} \right\} \quad (33)$$

In the speed reducer design problem, SHO and CSHOs were tested with a population number of 30 and a maximum number of iterations of 500. In addition, the algorithms have been executed 30 times. In the literature, many different metaheuristic optimization algorithms have solved this problem to find the best result. CSHOs have been compared with some of these methods in the literature. These methods are ABC, GSA, SHOA, GWO, ZOA, EO, MVO, AOA, SCA, and GA. The comparison of the optimum cost values obtained from these algorithms is listed in Table 16. According to Table 16, it is seen that the minimum cost value for the speed reducer design problem is obtained with Gauss/mouse map-based SHO when compared to other metaheuristic algorithms. In addition, performance comparisons of the proposed CSHOs, the original SHO, and other metaheuristic algorithms in the literature for this problem are graphically presented in Fig. 17.

As a result, better performance was obtained with the proposed CSHOs by using 10 different chaotic maps instead of using random variables produced by random number generators compared to the original SHO algorithm. The results obtained for 33 different benchmark functions and 4 different real-world engineering problems clearly show that the use of chaotic maps improves the performance of SHO. Taken together, the results of this study show that the CSHO algorithm can be considered a reliable alternative to existing optimization algorithms.



Fig. 8. Convergence curves of CSHOs and other metaheuristic algorithms.

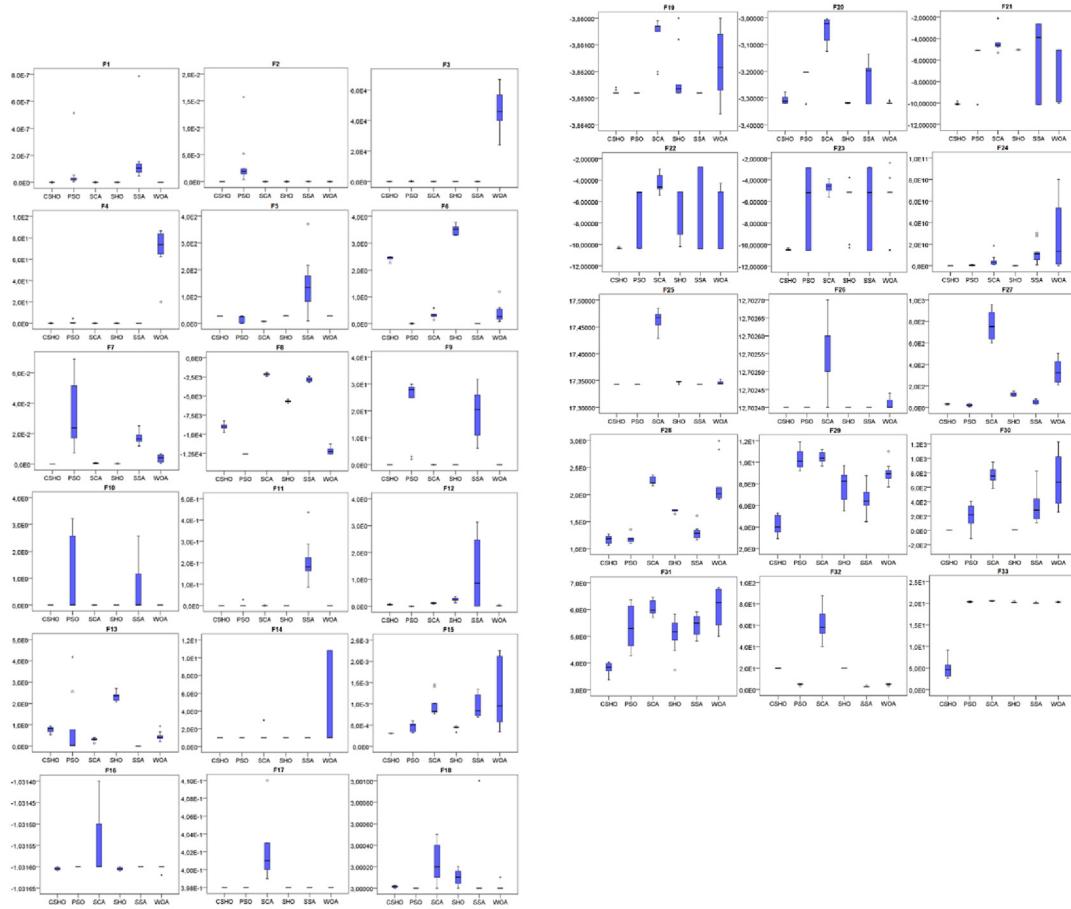


Fig. 9. Boxplots of the proposed CSHO and competitive algorithms on benchmark functions.

Table 9

Wilcoxon signed-rank test results of proposed CSHO and other metaheuristic algorithms on benchmark functions.

F_i	SHO p val.	SCA p val.	SSA p val.	WOA p val.	PSO p val.
F₁	2.00E-03	5.00E-03	5.00E-03	2.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
F₂	2.00E-03	5.00E-03	5.00E-03	2.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
F₃	2.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
F₄	2.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
F₅	5.00E-03	5.00E-03	7.00E-03	5.00E-03	7.00E-03
win	↑	↑	↑	↑	↑
F₆	5.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
F₇	5.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
F₈	5.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
F₉	1.0	5.00E-03	5.00E-03	1.0	5.00E-03
win	≈	↑	↑	≈	↑
F₁₀	1.0	5.00E-03	5.00E-03	2.00E-03	5.00E-03
win	≈	↑	↑	↑	↑
F₁₁	1.0	5.00E-03	5.00E-03	1.0	5.00E-03
win	≈	↑	↑	≈	↑
F₁₂	5.00E-03	9.00E-03	7.40E-02	5.00E-03	5.00E-03
win	↑	↑	≈	↑	↑
F₁₃	5.00E-03	5.00E-03	5.00E-03	1.40E-02	3.86E-01
win	↑	↑	↑	↑	≈
F₁₄	8.00E-03	1.70E-02	1.10E-01	8.59E-01	8.00E-03
win	↑	↑	≈	≈	↑
F₁₅	5.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03

(continued on next page)

Table 9 (continued)

F_i	SHO p val.	SCA p val.	SSA p val.	WOA p val.	PSO p val.
win	↑	↑	↑	↑	↑
F₁₆	8.00E-03	5.00E-03	8.00E-03	5.00E-03	8.00E-03
win	↑	↑	↑	↑	↑
F₁₇	8.00E-03	5.00E-03	8.00E-03	8.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
F₁₈	8.00E-03	7.00E-03	1.10E-01	1.10E-01	8.00E-03
win	↑	↑	≈	≈	↑
F₁₉	2.70E-02	5.00E-03	1.80E-01	4.70E-02	1.80E-02
win	↑	↑	≈	↑	≈
F₂₀	4.10E-02	5.00E-03	2.80E-02	4.70E-02	1.30E-02
win	↑	↑	↑	↑	↑
F₂₁	5.00E-03	5.00E-03	2.80E-02	5.00E-03	1.30E-02
win	↑	↑	↑	↑	↑
F₂₂	5.00E-03	5.00E-03	5.08E-01	2.80E-02	2.80E-02
win	↑	↑	≈	↑	↑
F₂₃	5.00E-03	5.00E-03	7.40E-02	9.00E-03	7.40E-02
win	↑	↑	≈	↑	≈
F₂₄	5.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
F₂₅	4.00E-03	5.00E-03	1.0	5.00E-03	1.0
win	↑	↑	≈	↑	≈
F₂₆	1.0	1.00E-02	1.0	6.80E-02	1.0
win	≈	↑	≈	≈	≈
F₂₇	5.00E-03	5.00E-03	2.20E-02	5.00E-03	2.80E-02
win	↑	↑	↑	↑	↑
F₂₈	5.00E-03	5.00E-03	3.70E-02	5.00E-03	8.38E-01
win	↑	↑	↑	↑	≈
F₂₉	5.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
F₃₀	5.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
F₃₁	7.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
F₃₂	6.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
F₃₃	5.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03
win	↑	↑	↑	↑	↑
(↑≈↓) (29/4/0)		(33/0/0)	(25/8/0)	(28/5/0)	(27/6/0)

Table 10Analysis results for different values of l parameter.

l	F₁	F₂	F₃	F₄	F₅	F₆
0.03	3,05E-165	1,05E-90	4,92E-130	3,06E-71	2,69E + 01	1,36E + 00
0.05	1,47E-166	2,74E-90	4,35E-128	2,19E-69	2,73E + 01	2,17E + 00
	F₇	F₈	F₉	F₁₀	F₁₁	F₁₂
0.03	3,82E-06	-8,87E + 03	0	8,88E-16	0	3,76E-02
0.05	6,03E-05	-9,00E + 03	0	8,88E-16	0	6,02E-02
	F₁₃	F₁₄	F₁₅	F₁₆	F₁₇	F₁₈
0.03	8,80E-01	9,98E-01	3,08E-04	-1,03E + 00	3,98E-01	3,00E + 00
0.05	7,73E-01	9,98E-01	3,08E-04	-1,03E + 00	3,98E-01	3,00E + 00
	F₁₉	F₂₀	F₂₁	F₂₂	F₂₃	F₂₄
0.03	-3,86E + 00	-3,32E + 00	-1,01E + 01	-1,03E + 01	-1,04E + 01	4,17E + 04
0.05	-3,86E + 00	-3,32E + 00	-1,01E + 01	-1,04E + 01	-1,05E + 01	4,11E + 04
	F₂₅	F₂₆	F₂₇	F₂₈	F₂₉	F₃₀
0.03	1,73E + 01	1,27E + 01	5,19E + 01	1,22E + 00	4,32E + 00	7,35E + 00
0.05	1,73E + 01	1,27E + 01	3,34E + 01	1,23E + 00	4,72E + 00	4,61E + 00
	F₃₁	F₃₂	F₃₃			
0.03	3,67E + 00	3,21E + 00	6,98E + 00			
0.05	3,87E + 00	2,01E + 01	1,83E + 01			

Table 11

Sensitivity analysis results for different numbers of iteration.

Iteration	F₁	F₂	F₃	F₄	F₅
100	3.95E-162	7.28E-84	3.29E-125	1.71E-70	2.75E + 01
500	1.47E-166	2.74E-90	4.35E-128	2.19E-69	2.73E + 01
1000	0	1.35E-175	3.18E-267	2.87E-139	2.72E + 01
	F₆	F₇	F₈	F₉	F₁₀
100	2.53E + 00	6.76E-05	-6.96E + 03	0	8.88E-16
500	2.17E + 00	6.03E-05	-9.00E + 03	0	8.88E-16
1000	1.55E + 00	1.08E-05	-9.42E + 03	0	8.88E-16
	F₁₁	F₁₂	F₁₃	F₁₄	F₁₅
100	0	2.10E-01	2.20E + 00	9.98E-01	4.33E-04
500	0	6.02E-02	7.73E-01	9.98E-01	3.08E-04
1000	0	3.35E-02	1.71E + 00	9.98E-01	3.41E-04
	F₁₆	F₁₇	F₁₈	F₁₉	F₂₀
100	-1.03E + 00	3.98E-01	3.00E + 00	-3.86E + 00	-3.32E + 00
500	-1.03E + 00	3.98E-01	3.00E + 00	-3.86E + 00	-3.32E + 00
1000	-1.03E + 00	3.98E-01	3.00E + 00	-3.86E + 00	-3.32E + 00
	F₂₁	F₂₂	F₂₃	F₂₄	F₂₅
100	-5.05E + 00	-5.08E + 00	-5.12E + 00	4.83E + 04	1.73E + 01
500	-1.01E + 01	-1.04E + 01	-1.05E + 01	4.11E + 04	1.73E + 01
1000	-7.05E + 00	-8.27E + 00	-9.36E + 00	4.39E + 04	1.73E + 01
	F₂₆	F₂₇	F₂₈	F₂₉	F₃₀
100	1.27E + 01	1.18E + 02	4.03E + 02	7.07E + 00	2.33E + 02
500	1.27E + 01	3.34E + 01	1.23E + 00	4.72E + 00	4.61E + 00
1000	1.27E + 01	3.58E + 01	1.24E + 00	4.81E + 00	-1.83E + 01
	F₃₁	F₃₂	F₃₃		
100	5.16E + 00	5.00E + 00	2.01E + 01		
500	3.87E + 00	2.01E + 01	1.83E + 01		
1000	3.87E + 00	3.04E + 00	1.60E + 01		

Table 12

Sensitivity analysis results for different search agent values.

Population	F₁	F₂	F₃	F₄	F₅
10	9.40E-155	2.52E-82	1.40E-121	1.22E-66	2.77E + 01
30	1.47E-166	2.74E-90	4.35E-128	2.19E-69	2.73E + 01
100	2.83E-178	7.86E-92	2.57E-133	4.82E-71	2.70E + 01
	F₆	F₇	F₈	F₉	F₁₀
10	3.02E + 00	3.18E-04	-6.41E + 03	0	8.88E-16
30	2.17E + 00	6.03E-05	-9.00E + 03	0	8.88E-16
100	1.27E + 00	5.03E-06	-9.90E + 03	0	8.88E-16
	F₁₁	F₁₂	F₁₃	F₁₄	F₁₅
10	0	3.33E-01	2.34E + 00	6.08E + 00	4.95E-04
30	0	6.02E-02	7.73E-01	9.98E-01	3.08E-04
100	0	4.15E-02	1.36E + 00	2.19E + 00	3.08E-04
	F₁₆	F₁₇	F₁₈	F₁₉	F₂₀
10	-1.03E + 00	3.98E-01	3.00E + 00	-3.86E + 00	-3.21E + 00
30	-1.03E + 00	3.98E-01	3.00E + 00	-3.86E + 00	-3.32E + 00
100	-1.03E + 00	3.98E-01	3.00E + 00	-3.86E + 00	-3.32E + 00
	F₂₁	F₂₂	F₂₃	F₂₄	F₂₅
10	-5.04E + 00	-5.06E + 00	-5.11E + 00	5.06E + 04	1.73E + 01
30	-1.01E + 01	-1.04E + 01	-1.05E + 01	4.11E + 04	1.73E + 01
100	-5.05E + 00	-5.09E + 00	-5.13E + 00	4.24E + 04	1.73E + 01
	F₂₆	F₂₇	F₂₈	F₂₉	F₃₀
10	1.27E + 01	1.08E + 03	3.79E + 03	8.85E + 00	3.56E + 02
30	1.27E + 01	3.34E + 01	1.23E + 00	4.72E + 00	4.61E + 00
100	1.27E + 01	2.93E + 01	1.37E + 00	4.46E + 00	1.09E + 02
	F₃₁	F₃₂	F₃₃		
10	5.60E + 00	5.50E + 00	2.01E + 01		
30	3.87E + 00	2.01E + 01	1.83E + 01		
100	3.85E + 00	3.28E + 00	2.00E + 01		

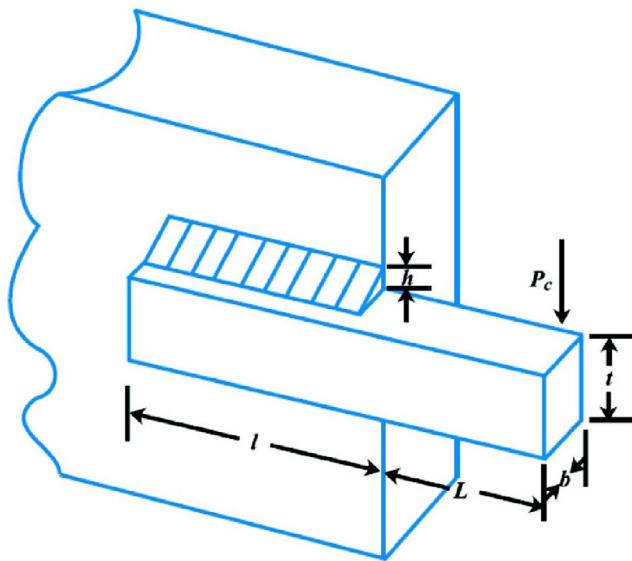


Fig. 10. Welded beam design problem.

The main limitation of all optimization algorithms is that there is no suitable optimization algorithm to solve all optimization problems. The proposed CSHO algorithm also has this limitation. This shows that the CSHO algorithm may still need improvements, modifications, and adjustments to solve different optimization problems. For example; CSHO can be extended for multiple purposes to solve combinatorial optimization problems. In addition, the CSHO algorithm can be adapted to binary space to solve feature selection and dimension reduction problems.

6. Conclusions

In this paper, CSHO based on 10 different chaotic maps was developed to improve the balancing ability of SHO between exploration and exploitation stages, to gain faster convergence, and to avoid local optimum. The proposed chaotic map-based SHO has

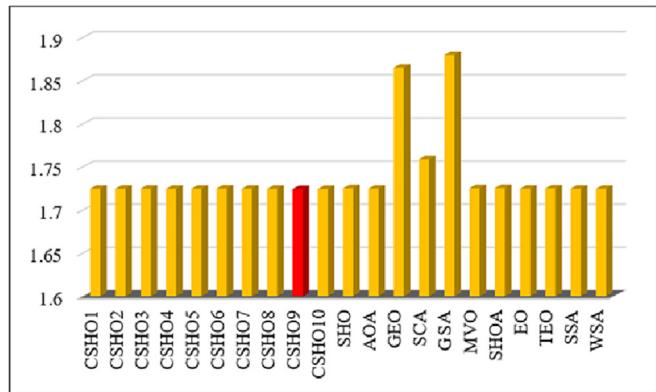


Fig. 11. Comparison of graphical results for welded beam design problem.

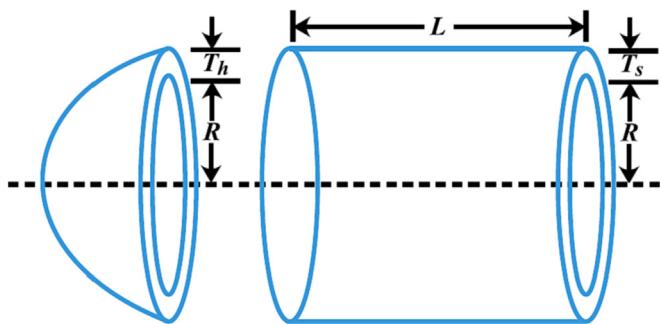


Fig. 12. Welded beam design problem.

been tested in 33 different benchmark functions, consisting of unimodal, multimodal, fixed dimension multimodal, and CEC2019. Obtained results have been analyzed statistically. It is seen that the proposed CSHO algorithm gives better results than the original SHO algorithm. The performance of CSHO is compared with different metaheuristic optimization algorithms (SSA, SCA, WOA, and PSO) in the literature. Compared to other metaheuristic optimiza-

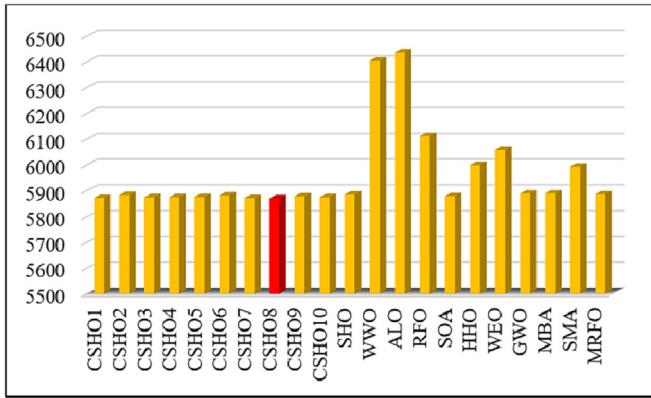
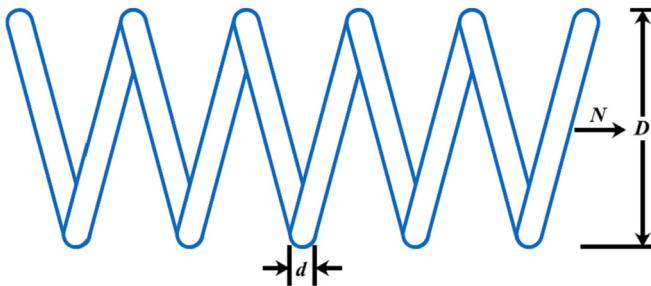
Table 13
Comparison results for welded beam design problem.

Algorithms	Optimum variables				Optimum cost
	<i>h</i>	<i>l</i>	<i>t</i>	<i>b</i>	
CSHO1	0.2056	3.4693	9.0354	0.2058	1.7248
CSHO2	0.2056	3.4709	9.0352	0.2058	1.7249
CSHO3	0.2057	3.4694	9.0350	0.2058	1.7249
CSHO4	0.2056	3.4703	9.0345	0.2058	1.7247
CSHO5	0.2057	3.4705	9.0335	0.2058	1.7248
CSHO6	0.2056	3.4692	9.0368	0.2058	1.7250
CSHO7	0.2057	3.4699	9.0346	0.2058	1.7249
CSHO8	0.2056	3.4697	9.0348	0.2058	1.7247
CSHO9	0.2056	3.4711	9.0364	0.2057	1.7244
CSHO10	0.2057	3.4698	9.0368	0.2057	1.7245
SHO	0.2058	3.4695	9.0327	0.2059	1.7254
AOA [83]	0.2057	3.4705	9.0366	0.2057	1.7249
GEO [84]	0.2444	3.0630	8.2914	0.2444	1.8653
SCA [85]	0.2047	3.5363	9.0042	0.2100	1.7591
GSA [36]	0.1821	3.8569	10.000	0.2037	1.8799
MVO [30]	0.2056	3.4721	9.0409	0.2057	1.7254
SHOA [34]	0.2055	3.4748	9.0357	0.2058	1.7256
EO [86]	0.2057	3.4705	9.0366	0.2057	1.7249
TEO [87]	0.2057	3.4723	9.0351	0.2058	1.7252
SSA [88]	0.2057	3.4714	9.0366	0.2057	1.7249
WSA [89]	0.2057	3.4705	9.0366	0.2057	1.7248

Table 14

Comparison results for pressure vessel design problem.

Algorithms	Optimum variables				Optimum cost
	x_1	x_2	x_3	x_4	
CSHO1	0.7745	0.3819	40.3347	199.7893	5872.5938
CSHO2	0.7796	0.3835	40.5791	196.4192	5883.3414
CSHO3	0.7772	0.3845	40.4588	198.0714	5874.8123
CSHO4	0.7776	0.3844	40.4736	197.8676	5875.3430
CSHO5	0.7777	0.3853	40.4498	198.1950	5875.7266
CSHO6	0.7813	0.3865	40.6807	195.0328	5882.3899
CSHO7	0.7743	0.3824	40.3352	199.7822	5871.7470
CSHO8	0.7749	0.3833	40.3346	199.7910	5870.6526
CSHO9	0.7798	0.3849	40.5447	196.8902	5878.7973
CSHO10	0.7777	0.3846	40.4600	198.0550	5875.0418
SHO	0.7782	0.3847	40.3223	199.9623	5885.4926
WWO [90]	0.8462	0.4601	42.3261	176.6321	6405.0180
ALO [29]	0.8514	0.4565	42.3652	176.6102	6436.9571
RFO [41]	0.8142	0.4452	42.2023	176.6214	6113.3195
SOA [38]	0.7781	0.3833	40.3151	200.0000	5879.5241
HHO [91]	0.8176	0.4073	42.0917	176.7196	6000.4626
WEO [31]	0.8125	0.4375	42.0985	176.6366	6059.7100
GWO [27]	0.7790	0.3846	40.3278	199.6503	5889.3679
MBA [92]	0.7802	0.3856	40.4292	198.4964	5889.3216
SMA [93]	0.7931	0.3932	40.6711	196.2178	5994.1857
MRFO [94]	0.7778	0.3849	40.3446	199.6515	5886.2000

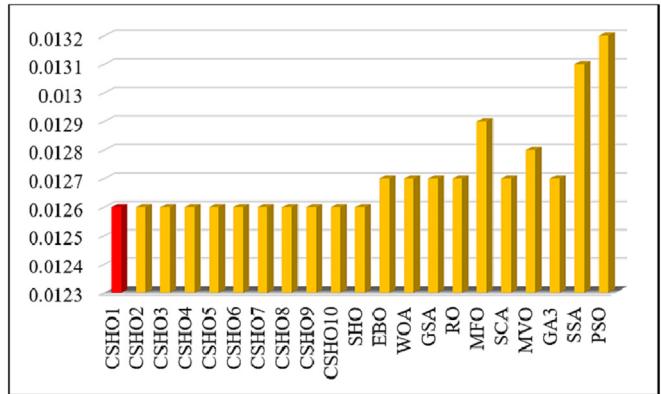
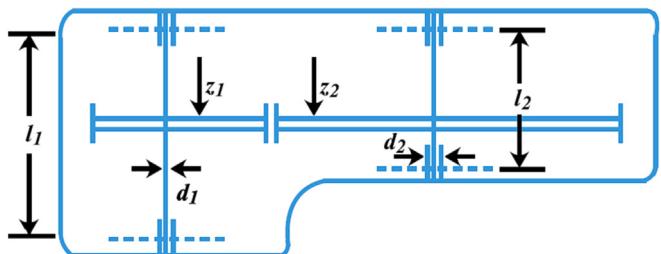
**Fig. 13.** Comparison of graphical results for pressure vessel design problem.**Fig. 14.** Tension/compression spring design problem.

tion algorithms, it is seen that the CSHO algorithm achieves better results than competitive algorithms. Additionally, CSHOs have been performed on four different real-world engineering problems: welded beam, pressure vessel, tension/compression spring, and speed reducer design problem. As a result of comparing the results obtained from CSHOs for these problems with different methods in the literature, it has been seen that CSHO outperforms competitive algorithms in solving engineering problems.

Table 15

Comparison results for tension/compression spring design problem.

Algorithms	Optimum variables			Optimum cost
	d	D	N	
CSHO1	0.0513	0.3486	11.7787	0.0126
CSHO2	0.0514	0.3499	11.6982	0.0126
CSHO3	0.0520	0.3645	10.8447	0.0126
CSHO4	0.0513	0.3470	11.8815	0.0126
CSHO5	0.0524	0.3752	10.2827	0.0126
CSHO6	0.0516	0.3550	11.3914	0.0126
CSHO7	0.0516	0.3548	11.4024	0.0126
CSHO8	0.0517	0.3576	11.2341	0.0126
CSHO9	0.0513	0.3475	11.8461	0.0126
CSHO10	0.0519	0.3637	10.8861	0.0126
SHO	0.0519	0.3629	10.9358	0.0126
EBO [95]	0.0533	0.3976	9.2519	0.0127
WOA [33]	0.0512	0.3452	12.0040	0.0127
GSA [36]	0.0512	0.3236	14.2500	0.0127
RO [96]	0.0513	0.3491	14.2500	0.0127
MFO [28]	0.0522	0.3695	10.7815	0.0129
SCA [85]	0.0507	0.3347	12.7227	0.0127
MVO [30]	0.0500	0.3159	14.2262	0.0128
GA3 [97]	0.0514	0.3516	11.6322	0.0127
SSA [35]	0.0501	0.3101	14.0000	0.0131
PSO [16]	0.0500	0.3104	15.0000	0.0132

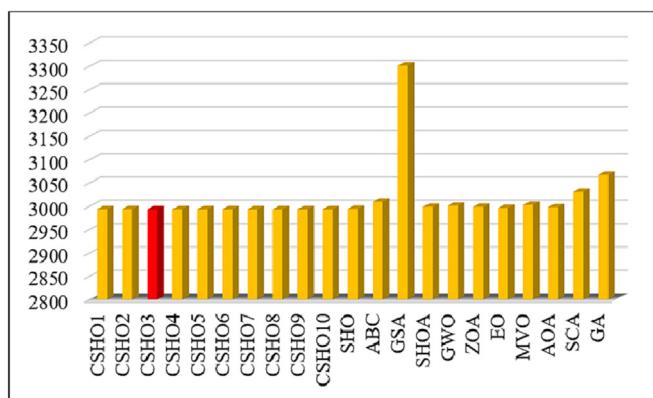
**Fig. 15.** Comparison of graphical results for tension/compression spring design problem.**Fig. 16.** Speed reducer design problem.

When evaluated in general, it is seen that the CSHO algorithm proposed in this study increased the convergence speed and solution quality compared to the SHO algorithm. Therefore, the application areas of the proposed CSHO algorithm can be increased. The CSHO algorithm can be applied for data mining and image processing, and can also be used to solve design problems in different engineering disciplines. CSHO can be adapted to optimize the neural network parameters so that the training performance of the

Table 16

Comparison results for speed reducer design problem.

Alg.	Optimum variables							Optimum cost
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
CSHO1	3.497	0.70	17.00	7.30	7.711	3.349	5.285	2993.78
CSHO2	3.498	0.70	17.00	7.30	7.716	3.350	5.284	2993.99
CSHO3	3.497	0.70	17.00	7.30	7.713	3.350	5.285	2993.63
CSHO4	3.497	0.70	17.00	7.30	7.710	3.349	5.285	2993.80
CSHO5	3.499	0.70	17.00	7.30	7.712	3.350	5.285	2993.68
CSHO6	3.497	0.70	17.00	7.30	7.711	3.350	5.285	2993.66
CSHO7	3.498	0.70	17.00	7.30	7.711	3.349	5.285	2993.70
CSHO8	3.498	0.70	17.00	7.30	7.710	3.350	5.284	2993.64
CSHO9	3.496	0.70	17.00	7.30	7.720	3.350	5.285	2993.70
CSHO10	3.497	0.70	17.00	7.30	7.712	3.350	5.285	2993.68
SHO	3.500	0.70	17.00	7.30	7.716	3.350	5.286	2994.50
ABC [98]	3.504	0.70	17.00	7.35	8.231	3.353	5.288	3009.64
GSA [36]	3.534	0.70	17.81	7.39	8.245	3.492	5.429	3301.58
SHOA [34]	3.050	0.70	17.00	7.30	7.800	3.351	5.288	2998.55
GWO [27]	3.506	0.70	17.00	7.38	7.815	3.357	5.286	3001.28
ZOA [99]	3.501	0.70	17.00	7.34	7.801	3.351	5.288	2998.51
EO [86]	3.500	0.70	17.00	7.30	7.800	3.350	5.286	2996.35
MVO [30]	3.508	0.70	17.00	7.39	7.816	3.358	5.286	3002.92
AOA [83]	3.503	0.70	17.00	7.30	7.729	3.356	5.286	2997.91
SCA [85]	3.508	0.70	17.00	7.30	7.800	3.461	5.289	3030.56
GA [100]	3.510	0.70	17.00	8.35	7.800	3.362	5.287	3067.56

**Fig. 17.** Comparison of graphical results for speed reducer design problem.

network can be increased. Finally, parallel, multi-objective, distributed, and binary versions of the proposed algorithm for solving complex optimization problems can be developed to contribute to the literature.

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.

References

- L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Eng.* 376 (2021) 113609.
- F. Altunbey Özbay, B. Alatas, Review of social-based artificial intelligence optimization algorithms for social network analysis, *Int. J. Pure Appl. Sci.* 1 (1) (2015) 33–52.
- F. Altunbey Özbay, E. Özbay, Performance analysis of seagull optimization algorithm for constrained engineering design problems, *J. Eng. Sci. Adiyaman Univ.* 15 (2021) (2021) 469–485.
- V. Kumar, J.K. Chhabra, D. Kumar, Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems, *J. Comput. Sci.* 5 (2) (2014) 144–155.
- S. Jiao, G. Chong, C. Huang, H. Hu, M. Wang, A.A. Heidari, X. Zhao, Orthogonally adapted Harris hawks optimization for parameter estimation of photovoltaic models, *Energy* 203 (2020) 117804.
- F.A. Özbay, B. Alatas, Adaptive salp swarm optimization algorithms with inertia weights for novel fake news detection model in online social media, *Multimed. Tools Appl.* 80 (26) (2021) 34333–34357.
- S.H. Huang, Y.H. Huang, C.A. Blazquez, C.Y. Chen, Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm, *Adv. Eng. Inf.* 51 (2022) 101536.
- M. Cai, O. Hocine, A.S. Mohammed, X. Chen, M.N. Amar, M. Hasanipanah, Integrating the LSSVM and RBFNN models with three optimization algorithms to predict the soil liquefaction potential, *Eng. Comput.* 38 (4) (2022) 3611–3623.
- E. Shadkam, Cuckoo optimization algorithm in reverse logistics: a network design for COVID-19 waste management, *Waste Manag. Res.* 40 (4) (2022) 458–469.
- E. Özbay, An active deep learning method for diabetic retinopathy detection in segmented fundus images using artificial bee colony algorithm, *Artif. Intell. Rev.* (2022).
- H. Fang, J. Zhou, Z. Wang, Z. Qiu, Y. Sun, Y. Lin, M. Pan, Hybrid method integrating machine learning and particle swarm optimization for smart chemical process operations, *Front. Chem. Sci. Eng.* 16 (2) (2022) 274–287.
- S. Mohammadi, S.R. Hejazi, Using particle swarm optimization and genetic algorithms for optimal control of non-linear fractional-order chaotic system of cancer cells, *Math. Comput. Simul.* (2022).
- S. Yildirim, Y. Kaya, F. Kiliç, A modified feature selection method based on metaheuristic algorithms for speech emotion recognition, *Appl. Acoust.* 173 (2021) 107721.
- L. Abualigah, A. Diabat, A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications, *Neural Comput. Appl.* 32 (19) (2020) 15533–15556.
- M. Khishe, M.R. Mosavi, Chimp optimization algorithm, *Expert Syst. Appl.* 149 (2020) 113338.
- J. Kennedy, R. Eberhart, (1995, November). Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks (Vol. 4, pp. 1942–1948). IEEE.
- M. Dorigo, G. Di Caro, (1999, July). Ant colony optimization: a new meta-heuristic, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406) (Vol. 2, pp. 1470–1477). IEEE.
- P. Pinto, T.A. Runkler, J.M. Sousa, Wasp swarm optimization of logistic systems, in: *Adaptive and Natural Computing Algorithms*, Springer, Vienna, 2005, pp. 264–267.
- S. C. Chu, P. W. Tsai, J. S. Pan, (2006, August). Cat swarm optimization, in: *Pacific Rim International Conference on Artificial Intelligence* (pp. 854–858). Springer, Berlin, Heidelberg.
- D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *J. Glob. Optim.* 39 (3) (2007) 459–471.
- Y. Chu, H. Mi, H. Liao, Z. Ji, Q.H. Wu, (2008, June). A fast bacterial swarming algorithm for high-dimensional function optimization, in: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)* (pp. 3135–3140). IEEE.
- E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, *GSA: a gravitational search algorithm*, *Inf. Sci.* 179 (13) (2009) 2232–2248.

- [23] X. S. Yang, 2010. Firefly algorithm, stochastic test functions and design optimisation. arXiv preprint arXiv:1003.1409.
- [24] K.M. Passino, Bacterial foraging optimization, in: Innovations and Developments of Swarm Intelligence Applications, IGI Global, 2012, pp. 219–234.
- [25] W.T. Pan, A new fruit fly optimization algorithm: taking the financial distress model as an example, *Knowl.-Based Syst.* 26 (2012) 69–74.
- [26] 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.
- [27] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61.
- [28] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowl.-Based Syst.* 89 (2015) 228–249.
- [29] S. Mirjalili, The ant lion optimizer, *Adv. Eng. Softw.* 83 (2015) 80–98.
- [30] 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.
- [31] A. Kaveh, T. Bakhshpoori, Water evaporation optimization: a novel physically inspired optimization algorithm, *Comput. Struct.* 167 (2016) 69–85.
- [32] H. Abedinpourshotorban, S.M. Shamsuddin, Z. Beheshti, D.N. Jawawi, Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm, *Swarm Evol. Comput.* 26 (2016) 8–22.
- [33] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.* 95 (2016) 51–67.
- [34] G. Dhiman, V. Kumar, Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications, *Adv. Eng. Softw.* 114 (2017) 48–70.
- [35] 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.
- [36] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, *Adv. Eng. Softw.* 105 (2017) 30–47.
- [37] J. Pierzzza, L.D.S. Coelho, Coyote optimization algorithm: a new metaheuristic for global optimization problems, in: 2018 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2018, pp. 1–8.
- [38] G. Dhiman, V. Kumar, Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems, *Knowl.-Based Syst.* 165 (2019) 169–196.
- [39] W. Zhao, L. Wang, Z. Zhang, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, *Knowl.-Based Syst.* 163 (2019) 283–304.
- [40] M. Khishe, M.R. Mosavi, Chimp optimization algorithm, *Expert Syst. Appl.* 149 (2020) 113338.
- [41] D. Polap, M. Woźniak, Red fox optimization algorithm, *Expert Syst. Appl.* 166 (2021) 114107.
- [42] M.M. Noel, V. Muthiah-Nakarajan, G.B. Amali, A.S. Trivedi, A new biologically inspired global optimization algorithm based on firebug reproductive swarming behaviour, *Expert Syst. Appl.* 183 (2021) 115408.
- [43] J.O. Agushaka, A.E. Ezugwu, L. Abualigah, Gazelle Optimization Algorithm: A novel nature-inspired metaheuristic optimizer, *Neural Comput. Appl.* (2022) 1–33.
- [44] S. Zhao, T. Zhang, S. Ma, M. Chen, Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications, *Eng. Appl. Artif. Intel.* 114 (2022) 105075.
- [45] L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, W. Zhao, Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems, *Eng. Appl. Artif. Intel.* 114 (2022) 105082.
- [46] P. Wang, Y. Zhou, Q. Luo, C. Han, Y. Niu, M. Lei, Complex-valued encoding metaheuristic optimization algorithm: A comprehensive survey, *Neurocomputing* 407 (2020) 313–342.
- [47] Q. Luo, X. Yang, Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, *Math. Comput. Simul.* 159 (2019) 57–92.
- [48] F. Miao, Y. Zhou, Q. Luo, Complex-valued encoding symbiotic organisms search algorithm for global optimization, *Knowl. Inf. Syst.* 58 (2019) 209–248.
- [49] C. Tang, Y. Zhou, Z. Tang, Q. Luo, Teaching-learning-based pathfinder algorithm for function and engineering optimization problems, *Appl. Intell.* 51 (2021) 5040–5066.
- [50] C. Wen, H. Jia, D. Wu, H. Rao, S. Li, Q. Liu, L. Abualigah, Modified remora optimization algorithm with multistrategies for global optimization problem, *Mathematics* 10 (19) (2022) 3604.
- [51] S. Talatahari, B.F. Azar, R. Sheikholeslami, A.H. Gandomi, Imperialist competitive algorithm combined with chaos for global optimization, *Commun. Nonlinear Sci. Numer. Simul.* 17 (3) (2012) 1312–1319.
- [52] S. Saha, V. Mukherjee, A novel chaos-integrated symbiotic organisms search algorithm for global optimization, *Soft. Comput.* 22 (11) (2018) 3797–3816.
- [53] S. Gupta, K. Deep, An opposition-based chaotic grey wolf optimizer for global optimisation tasks, *J. Exp. Theor. Artif. Intell.* 31 (5) (2019) 751–779.
- [54] M.S. Sanaj, P.J. Prathap, Nature inspired chaotic squirrel search algorithm (CSSA) for multi objective task scheduling in an IAAS cloud computing atmosphere, *Eng. Sci. Technol.* 23 (4) (2020) 891–902.
- [55] J.A. Koupaei, S.M.M. Hosseini, F.M. Ghaini, A new optimization algorithm based on chaotic maps and golden section search method, *Eng. Appl. Artif. Intel.* 50 (2016) 201–214.
- [56] Y. Kumar, P.K. Singh, A chaotic teaching learning based optimization algorithm for clustering problems, *Appl. Intell.* 49 (3) (2019) 1036–1062.
- [57] X.D. Li, J.S. Wang, W.K. Hao, M. Zhang, M. Wang, Chaotic arithmetic optimization algorithm, *Appl. Intell.* (2022) 1–40.
- [58] O. Altay, Chaotic slime mould optimization algorithm for global optimization, *Artif. Intell. Rev.* 55 (5) (2022) 3979–4040.
- [59] A. Rezaee Jordehi, A chaotic artificial immune system optimisation algorithm for solving global continuous optimisation problems, *Neural Comput. Appl.* 26 (4) (2015) 827–833.
- [60] B. Wu, S.H. Fan, Improved artificial bee colony algorithm with chaos, in: International Workshop on Computer Science for Environmental Engineering and EcoInformatics, Springer, Berlin, Heidelberg, 2011, pp. 51–56.
- [61] A. Mukherjee, V. Mukherjee, Solution of optimal reactive power dispatch by chaotic krill herd algorithm, *IET Gener. Transm. Distrib.* 9 (15) (2015) 2351–2362.
- [62] B.S. Yıldız, N. Pholdee, N. Panagant, S. Bureerat, A.R. Yıldız, S.M. Sait, A novel chaotic Henry gas solubility optimization algorithm for solving real-world engineering problems, *Eng. Comput.* 38 (2) (2022) 871–883.
- [63] S. Arora, P. Anand, Chaotic grasshopper optimization algorithm for global optimization, *Neural Comput. Appl.* 31 (8) (2019) 4385–4405.
- [64] A.A. Heidari, R. Ali Abbaspour, A. Rezaee Jordehi, An efficient chaotic water cycle algorithm for optimization tasks, *Neural Comput. Appl.* 28 (1) (2017) 57–85.
- [65] G. Kaur, S. Arora, Chaotic whale optimization algorithm, *J. Comput. Des. Eng.* 5 (3) (2018) 275–284.
- [66] E. Varol Altay, B. Alatas, Bird swarm algorithms with chaotic mapping, *Artif. Intell. Rev.* 53 (2) (2020) 1373–1414.
- [67] S. Zhao, T. Zhang, S. Ma, M. Wang, Sea-horse optimizer: A novel nature-inspired meta-heuristic for global optimization problems, *Appl. Intell.* (2022) 1–28.
- [68] H. Zhang, Y. Liu, G. Qin, Q. Lin, Identification of neurohypophysial hormones and the role of VT in the parturition of pregnant seahorses (*Hippocampus erectus*), *Front. Endocrinol.* 13 (2022) 923234.
- [69] C. Pierri, T. Lazic, M. Gristina, G. Corriero, M. Sinopoli, Large-scale distribution of the European Seahorses (*Hippocampus Rafinesque, 1810*): A systematic review, *Biology* 11 (2) (2022) 325.
- [70] J. Alfaro-Shigueto, E. Alfaro-Cordova, J.C. Mangel, Review of threats to the Pacific seahorse *Hippocampus ingens* (Girard 1858) in Peru, *J. Fish Biol.* (2022).
- [71] G. Del Vecchio, C.E. Galindo-Sánchez, M.A. Tripp-Valdez, E.A. López-Landaery, C. Rosas, M. Mascaró, Transcriptomic response in thermally challenged seahorses *Hippocampus erectus*: The effect of magnitude and rate of temperature change, *Compar. Biochem. Physiol. Part B: Biochem. Mol. Biol.* (2022) 110771.
- [72] L. He, X. Long, J. Qi, Z. Wang, Z. Huang, S. Wu, L. Zheng, Genome and gene evolution of seahorse species revealed by the chromosome-level genome of *Hippocampus abdominalis*, *Mol. Ecol. Resour.* 22 (4) (2022) 1465–1477.
- [73] R. F. Schneider, J. M. Woltering, D. Adriaens, O. Roth, 2022. A comparative analysis of the ontogeny of syngnathids (pipefishes & seahorses) reveals how heterochrony contributed to their diversification. *bioRxiv*.
- [74] G. Roos, S. Van Wassenbergh, A. Herrel, D. Adriaens, P. Aerts, Snout allometry in seahorses: insights on optimisation of pivot feeding performance during ontogeny, *J. Exp. Biol.* 213 (13) (2010) 2184–2193.
- [75] S.D. Job, H.H. Do, J.J. Meeuwig, H.J. Hall, Culturing the oceanic seahorse, *Hippocampus kuda*, *Aquaculture* 214 (1–4) (2002) 333–341.
- [76] M.M. Porter, D. Adriaens, R.L. Hatton, M.A. Meyers, J. McKittrick, Why the seahorse tail is square, *Science* 349 (6243) (2015) aaa6683.
- [77] D. Harasti, W. Gladstone, Does underwater flash photography affect the behaviour, movement and site persistence of seahorses?, *J. Fish Biol.* 83 (5) (2013) 1344–1353.
- [78] K.N. Stöltzing, A.B. Wilson, Male pregnancy in seahorses and pipefish: beyond the mammalian model, *Bioessays* 29 (9) (2007) 884–896.
- [79] R.N. Mantegna, Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes, *Phys. Rev. E* 49 (5) (1994) 4677.
- [80] A. Einstein, Investigations on the Theory of the Brownian Movement, Courier Corporation, 1956.
- [81] J.M. Abdullah, T. Ahmed, Fitness dependent optimizer: inspired by the bee swarming reproductive process, *IEEE Access* 7 (2019) 43473–43486.
- [82] W.W. Daniel, Friedman two-way analysis of variance by ranks, *Appl. Nonparametr. Stat.* (1990) 262–274.
- [83] F.A. Hashim, K. Hussain, E.H. Houssein, M.S. Mabrouk, W. Al-Atabay, Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems, *Appl. Intell.* 51 (3) (2021) 1531–1551.
- [84] A. Mohammadi-Balani, M.D. Nayeri, A. Azar, M. Taghizadeh-Yazdi, Golden eagle optimizer: A nature-inspired metaheuristic algorithm, *Comput. Ind. Eng.* 152 (2021) 107050.
- [85] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, *Knowl.-Based Syst.* 96 (2016) 120–133.
- [86] A. Faramarzi, M. Heidarnejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: A novel optimization algorithm, *Knowl.-Based Syst.* 191 (2020) 105190.
- [87] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: thermal exchange optimization, *Adv. Eng. Softw.* 110 (2017) 69–84.
- [88] H. Bayzidi, S. Talatahari, M. Saraei, C.P. Lamarche, 2021, Social network search for solving engineering optimization problems, *Comput. Intell. Neurosci..*

- [89] A. Kaveh, A. D. Eslamlou, 2020. Water strider algorithm: A new metaheuristic and applications. *Structures*, 25, 520–541.
- [90] Y.J. Zheng, Water wave optimization: a new nature-inspired metaheuristic, *Comput. Oper. Res.* 55 (2015) 1–11.
- [91] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Futur. Gener. Comput. Syst.* 97 (2019) 849–872.
- [92] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, *Appl. Soft Comput.* 13 (5) (2013) 2592–2612.
- [93] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Futur. Gener. Comput. Syst.* 111 (2020) 300–323.
- [94] W. Zhao, Z. Zhang, L. Wang, Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications, *Eng. Appl. Artif. Intel.* 87 (2020) 103300.
- [95] Y.J. Zheng, H.F. Ling, J.Y. Xue, Ecogeography-based optimization: enhancing biogeography-based optimization with ecogeographic barriers and differentiations, *Comput. Oper. Res.* 50 (2014) 115–127.
- [96] A. Kaveh, M. Khayatazad, A new meta-heuristic method: ray optimization, *Comput. Struct.* 112 (2012) 283–294.
- [97] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, *Comput. Ind.* 41 (2) (2000) 113–127.
- [98] D. Karaboga, B. Basturk, 2007, June. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, in: International fuzzy systems association world congress (pp. 789–798). Springer, Berlin, Heidelberg.
- [99] E. Trojovská, M. Dehghani, P. Trojovský, Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm, *IEEE Access* 10 (2022) 49445–49473.
- [100] S. Mirjalili, Genetic algorithm, in: *Evolutionary Algorithms and Neural Networks*, Springer, Cham, 2019, pp. 43–55.