Knowledge and Information Systems
https://doi.org/10.1007/s10115-025-02422-5

RESEARCH

A multi-strategy chimp optimization algorithm for solving global and constraint engineering problems

Ferzat Anka[1,2]

Received: 1 January 2024 / Revised: 19 December 2024 / Accepted: 22 March 2025

Abstract

The chimp optimization algorithm (ChOA) is a recently introduced metaheuristic algorithm inspired by nature. This algorithm identified four types of chimpanzee groups: attacker, barrier, chaser, and driver, and proposed a suitable mathematical model for them, based on the various intelligence and sexual motivations of chimpanzees. However, this algorithm is not more successful in the convergence rate and escaping of the local optimum trap in solving high-dimensional problems. Although it and some of its variants use some strategies to overcome these problems, it is observed that it is not sufficient. Therefore, in this study, a newly expanded variant is described. In the algorithm, called Ex-ChOA, hybrid models are proposed for position updates of search agents, and a dynamic switching mechanism is provided for transition phases. This flexible structure solves the slow convergence problem of ChOA and improves its accuracy in multi-dimensional problems. Therefore, it tries to achieve success in solving global, complex, and constrained problems. The performance of the proposed algorithm was analyzed on a total of 34 benchmark functions and a total of 17 real-world optimizations, including classical, constrained, and modern engineering problems. According to the results obtained, the proposed algorithm performs better or equivalent performance than the compared algorithms.

Keywords Optimization · Metaheuristic · Chimp optimization algorithm · Engineering constrained problems

1 Introduction

Many types of problems can be solved using traditional analytical methods. However, these methods take a long time and cause inefficient use of resources. In particular, different approaches may be required in solving complex and global engineering problems that we

✉ Ferzat Anka
fanka@fsm.edu.tr

1 Computer Engineering Dept., Faculty of Engineering, Fatih Sultan Mehmet Vakif University, Istanbul, Türkiye

2 Data Science Application and Research Center (VEBIM), Fatih Sultan Mehmet Vakif University, Istanbul, Türkiye

frequently encounter in real life. The bigger and more complex a problem, the harder it is to solve [1]. Such problems are called non-deterministic polynomial time (NP-hard) in the literature [2]. The approach that is at the forefront of the solution mechanisms and has become widespread in recent years is the metaheuristic approach [3]. These algorithms try to find the best solution (or near optimal) with the least cost. Due to the nature of these algorithms, they do not fall into the local optimal trap and have very simple and flexible structures. In addition, they are advantageous both in terms of time, resources, and the problem of finding technical solutions by eliminating the negative aspects of traditional or heuristic approaches. Therefore, they can play an important role in solving very complex and multi-dimensional problems. In short, algorithms based on this approach can find global solutions simply and quickly with low requirements because they can scan the search space in detail. Moreover, according to the no free lunch (NFL) [4] approach, not every algorithm may be good for every optimization problem. Therefore, it is important and natural to propose and develop new metaheuristic algorithms for various problems. The main reasons for recommending different metaheuristic algorithms for various problems are the use of simple concepts, the use of simple mathematical equations and structures, the use of non-derivative mechanisms, the avoidance of local optima, and their fast convergence [5]. They are also flexible as they can be applied to different problems without very specific modifications. Thanks to these features, it can be easily embedded even in many hardware devices. Accordingly, this approach can also be used in trend application areas such as IoT, big data, and parallel structures [6, 7]. In addition, metaheuristic methods consisting of exploration and exploitation phases are simple and easy to implement, regardless of the type and size of the problem, with their stochastic structure. The main purpose of the proposed algorithms is to avoid falling into local traps and to get closer to the target. In the exploration phase, while intelligent agents in the population try to find a resource, they try to use the other acting resource. In the discovery phase of the algorithm, in the first generation, intelligent agents in the population are used to generate new solutions. In subsequent generations, population members try to increase the accuracy of the solutions by taking advantage of the solutions obtained in the discovery phase. These algorithms attempt to strike a balance between exploitation and exploration, despite differences in inspiration and search mechanisms. Transitions between phases and management of this process are one of the success criteria of these algorithms. So, they can be good candidates for finding suitable solutions for all types of optimization problems, especially NP-hard problems [8, 9]. Therefore, as the complexity and size of the problems increase, there is a need to develop new algorithms or improve/modify existing algorithms [10].

Metaheuristic approaches are algorithms that return near-optimal results for solving large-scale optimization problems. For example, the artificial bee colony (ABC) [11], which examines the foraging behavior of bees in nature and is adapted to solve real-world problems with these movements, or the particle swarm optimization (PSO) [12], which optimizes the process by adapting the movements that birds follow as a flock to find food in nature to a real-world problem, are examples of developed algorithms. The generally accepted approach in literature is to examine metaheuristic algorithms in five main categories (Fig. 1). These are evolutionary-based (EA), physics-based (PA), human and social behavior-based (HS), swarm intelligence-based (SI), and hybrid algorithms (HA) [13]. The algorithms in each category are adapted as unimodal and multi-modal, single-objective and multi-objective, in accordance with the problem and application. On the other hand, these algorithms are discussed in different categories in another study [14]. The authors mentioned two classes of metaheuristic algorithms: single-solution-based and ensemble-based methods. Accordingly, metaheuristic methods are basically classified as nature-inspired and non-nature-inspired

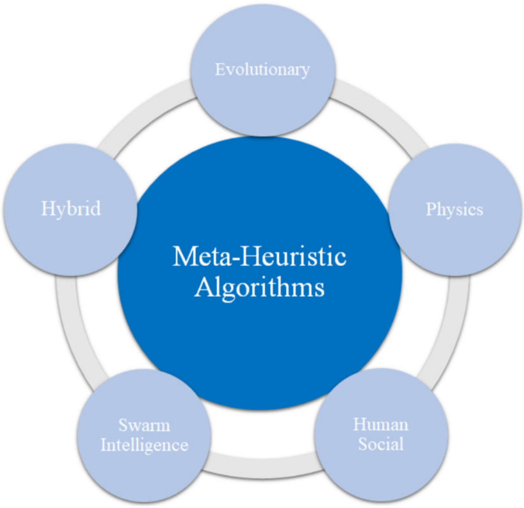

Fig. 1 A classification of MH algorithms

methods with dynamic and static objective functions. Furthermore, they are classified as neighborhood structure and variable neighborhood structure, memory-using or memory-free, single-solution or ensemble-based methods. In single-solution metaheuristics, optimization problems are solved by optimizing a single target. Single-solution-based metaheuristics focus on moving from a particular solution to another neighboring solution by walking through neighborhoods. In this study, we focus on 5 commonly used categories.

EAs are basically inspired by Darwin's theory. The most famous algorithms in this category and their variants are listed below. Genetic algorithm (GA) [15], differential evolution (DE) [16], biogeography-based optimizer (BBO) [17], wildebeests herd optimization (WHO) [18], and improved quantum-inspired cooperative coevolution algorithm (MSQCCEA) [19]. The WHO is inspired by the migrations of wildebeest in Africa. During migration, each antelope tries to choose its own path, taking into account the position and mobility of the mature one in the herd. In the black widow optimization (BWO) [20] algorithm, an approach close to genetic logic is used, inspired by the life of hummocks. Here, it tries to find the best solution to various problems through processes such as mutation. PAs are inspired by the laws of physics in nature and behave randomly according to assumed physical events. We should note that this random behavior helped bring them closer to the answer. In general, algorithms in other categories also have this feature. Famous methods in this category are henry gas solubility (HGS) [21], chaotic fractal walk trainer (CFWT) [22], adaptive best-mass gravitational search (ABGS) [23], atom search optimization (ASO) [24], arithmetic optimization algorithm (AOA) [25], special relativity search (SRS) [26], and Al-Biruni earth radius (BER) [27]. The ASO is inspired by fundamental molecular dynamics. This algorithm mimics the pattern of atomic motion in nature, where atoms interact through interaction forces due to the Lennard–Jones potential and constraint forces due to the bond length potential. The AOA This method attempts to find answers in various search spaces using popular arithmetic operators such as multiplication, addition, subtraction, and division. The SRS algorithm is based on the concept that time is relative and depends on the observer's frame of reference, inspired by Einstein's theory of special relativity. This algorithm consists of three main stages: evaluation, relativity update and selection. This proposed algorithm has been claimed to have good performance in multi-modal and high-dimensional problems.

The third category, which focuses on many characteristics of humans such as lifestyle and social behavior, is the human behavior algorithms. The soccer league competition (SLC) [28] algorithm, social mimic optimization (SMO) [29], poor and rich optimization (PRO) [30], Olympiad optimization algorithm [31], driving training-based optimization (DTBO) [32], and teaching–learning-based optimization (TLBO) [33] are well-known algorithms in this group. The DTBO is inspired by the way humans behave when learning to drive. For this, the algorithm was developed from three different perspectives. One of them is the practice model, the other is the pattern consisting of the course teacher and the last one is the teaching approach used by the driving instructor. TLBO is inspired by the teaching and learning process observed in the classroom environment and consists of two basic phases. One of these is the teaching phase, where the teacher with the best solution shares his knowledge with other students in the population. In the other phase, learning, students interact with each other, exchange information, and learn from better solutions in the population.

The fourth category is swarm intelligence (SI) algorithms, which have been the focus of researchers lately. These algorithms generally work in the swarm (herd) mechanism. A herd is used to mean a collection of scattered individuals interacting with each other. Individuals represent a herd that works together to perform purposeful behavior and achieve a goal. This easily observable 'collective intelligence' arises from frequently repeated behaviors among representatives. The search agents use simple individual rules to govern their activities and achieve herd goals through interaction with the rest of the group. A kind of self-organization arises from the sum of group activities. In SI-based algorithms, there is a swarm distributed architecture and a self-organizing structure. Members here can interact with each other and be successful in finding the target. This definition can be applied in many areas today and can be blended with other concepts such as machine learning, artificial intelligence, software, sensor networks and IoT-based smart systems [34–40]. Ant colony optimization (ACO) [41], PSO [12], grey wolf optimization (GWO) [42], ABC [11], sand cat swarm optimization (SCSO) [43], mountain gazelle optimizer (MGO) [44], puma optimizer algorithm (POA) [45], and binary artificial electric field algorithm (AEFA) [46] are few studies in this category. The SCSO algorithm is inspired by the lives of desert cats and their special hearing ability. This algorithm is used to solve many problems with its balanced and simple approach in the attack and search phases. The MGO algorithm mimics the hunting behavior of mountain gazelles, known for their ability to evade predators. In this algorithm, the optimization process involves simulating the movements of gazelles in the search space and consists of three main stages: exploration, homecoming, and herding. POA was designed inspired by puma hunting strategies and consists of four main schemes: search, movement, jumping and repositioning. Fast convergence and low probability of falling into local optima are the distinctive features of this algorithm.

Algorithms belonging to the last category are hybrid-based algorithms. Studies in this category have emerged to more efficiently overcome difficult and complex problems encountered in the real world. These algorithms can be designed in two different ways. 1) It is achieved by combining the strengths of existing algorithms. Combined algorithms do not have to be from the same category family. Additionally, they may consist of improved versions of existent algorithms. 2) It may propose through a combination of approaches that exist in nature but have not been studied in the literature. An example of an algorithm gaining popularity due to the identified need is the PSCSO algorithm [47], which is proposed as a combination of political optimizer (PO) [48] and SCSO [43] algorithms. As mentioned above, PO is an HS-based algorithm while SCSO is an SI-based algorithm. This proposed hybrid algorithm offers a more balanced and faster mechanism by using the advantages of both methods. The PSO-GA [49], GWO-WOA [50], AVO-HA [51], and RLWOA [52] methods can be given as examples of this category. In the PSO-GA algorithm proposed on the flexible flow layout planning problem, transport constraints are also included. In this algorithm, PSO is used to efficiently explore the solution space while GA is used to improve and refine the solutions. The AVO-HA hybrid algorithm, created by combining the African Vulture Optimization Algorithm and the Cohesion Search Algorithm, is designed to solve clustering problems. The paper conducts experiment on different datasets to evaluate the performance of this new algorithm. In RLWOA, hybrid algorithms created by combining two popular optimization approaches, reinforcement learning (RL) and whale optimization algorithm (WOA) [53], are designed to provide solutions to complex optimization problems. Q-learning method based on RL approach was used to make decisions efficiently, smoothly, and stable in the exploration and exploitation phases. In this hybrid structure, if the exploration phase is progressing well in MH, RL is encouraged to continue the same process; otherwise, the decision resulting from MH is changed and moves to the exploitation phase. A similar situation also applies to the opposite scenario. Each of these algorithms has its own characteristics. It is noteworthy that when the first versions of all these algorithms were presented and introduced, they were mainly applied to different benchmark functions (CEC groups) and/or known complex engineering fields. The characteristics of some of the algorithms considered in these categories are summarized in more detail in Table 1.

A recently published study, chimp optimization algorithm (ChOA) [55], is a nature-inspired, achievable algorithm that includes features in the fourth category. Since this algorithm has some general and special features, it has suitable convergence behavior in finding the optimum solution. This algorithm is inspired by the extraordinary group-hunting mechanism of chimpanzees. Also, this algorithm memorizes the search domain information throughout iteration and almost uses memory to keep the best solution they find. However, there is a possibility of falling to local optima in the exploration phase. Although this algorithm tries to solve this problem by the custom chaotic map, it may increase the transaction cost. Also, since it is semi-deterministic, it is observed that it moves away from probability and stochastic. In this paper, a new variant of ChOA is proposed with a new idea and perspective to solve complex and global benchmarking and engineering optimization problems in order to eliminate its problems and shortcomings, and its name is extended-ChOA (Ex-ChOA). We offer a more flexible and effective solution by using this mechanism as well as this map. In addition, the balance between the exploitation and exploration phases cannot be well established. The ChOA uses a special dynamic strategy to balance global and local searches. Although it is dynamic, presenting selection models may not be a good strategy for solving various problems. The Ex-ChOA offers a second solution model to make this algorithm more flexible. Indeed, the proposed algorithm aims to make phase transitions more balanced and stable with a new strategy. In addition, ChOA has a low speed in convergence and is not very accurate in multi-dimensional, complex, and constrained engineering problems. In this paper, new mathematical models are suggested by considering these problems. The Ex-ChOA will be able to explore the search space in various problems, especially complex and high-dimensional problems. In addition to these, the main contributions of this paper are briefly included as follows.

- It improves the accuracy of the ChOA.
- It solves the slow convergence problem of the ChOA.
- It proposes new hybrid movement strategy models for position updates of search agents.
- It provides a dynamic switching mechanism between phases.
- It provides success in solving global, complex, and constrained problems.

Table 1 A brief review of MH algorithms according to various categories

Category	Algorithm names	Year	Strategies	Basic focus	Advantages	Disadvantages
EA	GA [15]	1992	Crossover, mutation, elite selection	Global optimization	Powerful in exploration phase	Weak convergence
	DE [17]	1997	Crossover, mutation, selection	Global optimization	Various solutions	Poor intensification component
	BBO [16]	2008	Geographical distribution of biological organisms	Global optimization	Simplicity and efficiency	Sensitive to dimensionality
	WHO [18]	2019	Wildebeest herding behavior	Global optimization	Simplicity and efficiency	Demand for thermostat settings
	MSQCCEA [19]	2021	Random rotation direction, hamming adaptive rotation angle	Global optimization	Good in exploration	Slow convergence speed, Insufficient in exploration phase
PA	HGSO [21]	2019	The behavior of Henry's law	Global optimization	Various solutions	Slow convergence speed, easy to local optimal trap
	ASO [24]	2019	Molecular dynamics parameter	Global optimization	Simple and easy	Easily traps in the local optimum and premature convergence
	AOA [25]	2021	Multiplication, subtraction, and division	Global optimization	Simple and easy	Easily traps in the local optimum
	SRS [26]	2023	Interaction of particles	Global optimization	Good ability of component	Insufficient in exploration phase, easy to local optimal trap, low convergence accuracy
	BER [27]	2023	Earth radius computation	Global optimization	Simple and easy	Not efficient in some problems
HS	TLBO [33]	2011	Teaching and learning stages	Global optimization	Very strong convergence ability	Easy to fall into a local search

Table 1 (continued)

Category	Algorithm names	Year	Strategies	Basic focus	Advantages	Disadvantages
	PRO [30]	2019	Mutation, position update, new population	Global optimization	Balance in exploration and exploitation phases	low convergence and easily traps in the local optimum
	DTBO [32]	2022	Drive training process	Global optimization	Balance behavior in both phases	The necessity of tuning the hyper parameter; Insufficient in exploration phase
	PO [48]	2021	Election campaign, Ticket allocation, Inter-party election, parliamentary affairs	Global optimization	Strong in exploitation	Need sufficient time for exploration
	OOA [31]	2023	Learning procedure	Global optimization	Simple structure	traps in the local optimum
SI	PSO [12]	1995	Velocity and position updates	Global optimization	Good ability of the intensification component	Sensitivity to parameters
	ABC [11]	2007	Employed bee, scout bee, and onlooker bee	Constrained optimization	Good ability to solve unimodal problems	The difficult setting of parameters
	GWO [42]	2015	Encircling and Hunting	Global optimization	Balance behavior	Convergence accuracy and instability
	SCSO [43]	2022	Searching and attacking	Global optimization	Balance behavior	Slow convergence
	PO [46]	2023	Discovery, attacking and hyper heuristic	Global optimization	Good in exploitation	Slightly high process cost

Table 1 (continued)

Category	Algorithm names	Year	Strategies	Basic focus	Advantages	Disadvantages
HA	RLWOA [52]	2021	Q-table, surround and attack	Global optimization	A good decision mechanism in phase transitions	Slightly high process cost
	PSO-GA [49]	2022	GA for explore, PSO for exploit	Global optimization	Good in find solutions	Inefficient in some high-dimensional problems
	PSCSO [47]	2023	PO in exploitation phase	Global optimization	Good in exploitation	Slow convergence
	CSCSO [54]	2023	Chaotic maps in both phases	Global optimization	Good in exploration	Slightly high process cost

The rest of this paper is organized as follows. Section 2 is an overview of the standard chimp optimization algorithm. Section 3 presents an expanded variant of the ChOA algorithm based on an adaptive reform and hybrid mechanism. Sections 4 and 5 contain the results and discussions about various test functions and different types of engineering optimization problems. The conclusion and future works take place in the last section.

2 Standard chimp optimization algorithm

The chimp optimization algorithm (ChOA) is a new metaheuristic algorithm based on the recently proposed SI category [55]. A chimpanzee colony is a fission–fusion society, and tasks in the hierarchy may change over time [56, 57]. This shows the dynamism in group mechanisms. Based on this basic knowledge, the concept of independent groups is proposed in this algorithm. Although individuals have different abilities and intelligence, they are all responsible for fulfilling their duties in the hunting process. This feature not only enhances diversity but also adds dynamism, which can be advantageous in the prey capture process. In this algorithm, the members in each group try to explore the search space with their own strategy independently of other groups. These groups are divided into four types: attacker, barrier, chaser, and driver. In the exploration phase, the driver, barrier, and chaser groups are responsible, and in the exploitation phase, members of the attacker group. In short, the ChOA is an algorithm that works in a hierarchical structure with few parameters inspired by nature. Because chimpanzees are made up of various individuals in groups, various responsibilities are easily identified, and they can access their prey quickly and easily. For this reason, the ChOA algorithm can work well in the exploration and exploitation phases.

In the remainder of this section, the mathematical perspective and working model of the algorithm are explained. Chimpanzees flank the prey to attack, and accordingly, the mathematical model has been made in two interrelated aspects. The encircle of prey is given based on Eqs. 1, 2, and 3. Moreover, the position update of each agent is also calculated according to Eq. 4.

$$\begin{aligned} \vec{\mathrm{D}}_{\text{Attacker}} &= \left|\mathrm{C}_1 \cdot \vec{\mathrm{X}}_{\text{Attacker}} - \overrightarrow{\mathrm{m}_1} \cdot \vec{X}(t)\right| \\ \vec{\mathrm{D}}_{\text{Barrier}} &= \left|\mathrm{C}_2 \cdot \vec{\mathrm{X}}_{\text{Barrier}} - \overrightarrow{\mathrm{m}_2} \cdot \vec{X}(t)\right| \\ \vec{\mathrm{D}}_{\text{Chaser}} &= \left|\mathrm{C}_3 \cdot \vec{\mathrm{X}}_{\text{Chaser}} - \overrightarrow{\mathrm{m}_3} \cdot \vec{X}(t)\right| \\ \vec{\mathrm{D}}_{\text{Driver}} &= \left|\mathrm{C}_4 \cdot \vec{\mathrm{X}}_{\text{Driver}} - \overrightarrow{\mathrm{m}_4} \cdot \vec{X}(t)\right| \end{aligned} \tag{1}$$

$$C = 2 \cdot \mathrm{r}_1 \tag{2}$$

$$\begin{aligned} \overrightarrow{X_1}(t) &= \vec{X}_{\text{attacker}} - a_1 \cdot \vec{D}_{\text{Attacker}} \\ \overrightarrow{X_2}(t) &= \vec{X}_{\text{barrier}} - a_2 \cdot \vec{D}_{\text{Barrier}} \\ \overrightarrow{X_3}(t) &= \vec{X}_{\text{chaser}} - a_3 \cdot \vec{D}_{\text{Chaser}} \\ \overrightarrow{X_4}(t) &= \vec{X}_{\text{driver}} - a_4 \cdot \vec{D}_{\text{Driver}} \end{aligned} \tag{3}$$

$$a = 2f \cdot \mathrm{r}_2 - f \tag{4}$$

where $\overrightarrow{D_i}$ represents the distance between each chimp and prey and it is a vector that depends on the location of the target. $\overrightarrow{X}$ is the position vector of the prey and $\overrightarrow{X_i}$ is the position vector

Table 2 The dynamic coefficients of the f parameter

Groups	ChOA1	ChOA2
1 (Attacker)	$1.95 - 2t^{1/4}/T^{1/3}$	$2.5 - (2\log(t)/\log(T))$
2 (Barrier)	$1.95 - 2t^{1/3}/T^{1/4}$	$(-2t^3/T^3) + 2.5$
3 (Chaser)	$(-3t^3/T^3) + 1.5$	$0.5 + 2\exp[-(4t/T)^2]$
4 (Driver)	$(-2t^3/T^3) + 1.5$	$2.5 + 2(t/T)^2 - 2(2t/T)$

of the four groups of chimps ($l = 1, \ldots, 4$). Besides, 't' represents the current iteration and 'T' represents the maximum number of iterations. 'r_1' and 'r_2' are random vectors that are between range [0, 1]. The 'a', and 'C' coefficient vectors that lead to encircling the prey. The 'C' coefficient provides random weights to increase or decrease the emphasis ($C > 1$ or $C < 1$) in the effect of prey position in determining distance. In the ChOA algorithm, the leader member surrounds the prey (based on Eq. 1), then hunts and ultimately lunges it based on the 'a' value based on Eq. 4. If $|a| < 1$, the chimpanzee is obliged to attack its prey; otherwise, it tries to find another prey. In addition, 'f' is a control parameter that is nonlinearly decreased from 2.5 to 0 throughout the iterations and is achieved using Table 2. Since two different scenarios are assumed in this algorithm, the approach that seems appropriate and used for each is presented in this table. Here, separate 'f' values were used for each group. This parameter is obtained with a dynamic strategy and is used to balance actions in both phases and to solve complex optimization problems. Since these dynamic common coefficients are chosen with various curves and slopes, it is thought that they will be effective in improving the performance of the algorithm.

Also, $\overrightarrow{m}$ is a chaotic vector based on various used chaotic maps. This vector represents the influence of chimpanzees' diverse motivations in hunting. Six chaotic maps are used in the ChOA algorithm as shown in Table 3 [5, 55]. Thanks to the specific behavior of each of these maps, it can produce different results. Therefore, they may have positive contributions to the relevant algorithm.

We should note that the attacker is supposed as the best candidate solution and the other three groups tracks it. In this algorithm, two alternative solutions are proposed for position

Table 3 Chaotic maps used in ChOA [55]

No.	Name	Chaotic map	Range
1	Quadratic	$X_{i+1} = X_i^2 - c, \quad c = 1$	(0, 1)
2	Gauss/mouse	$X_{i+1} = \begin{cases} 1 & X_i = 0 \\ \frac{1}{\mathrm{mod}\,(X_i, 1)} & \text{otherwise} \end{cases}$	(0, 1)
3	Logistic	$X_{i+1} = aX_i(1 - X_i), \quad a = 4$	(0, 1)
4	Singer	$X_{i+1} = \mu(7.86X_i - 23.31X_i^2 + 28.75X_i^3 - 13.302875X_i^4), \quad \mu = 1.07$	(0, 1)
5	Bernoulli	$X_{i+1} = 2X_i(\mathrm{mod}\,1)$	(0, 1)
6	Tent	$X_{i+1} = \begin{cases} \frac{X_i}{0.7} & X_i < 0.7 \\ \frac{10}{3}(1 - X_i), & X_i \geq 0.7 \end{cases}$	(0, 1)

updating. This model works as presented in Eq. 5. The position update mechanism is an approach based on the normal model in one and the chaotic model in the other [5, 55, 58]. Each of them is given a 50% chance to have a balanced weight. The 'μ' is a random number in [0,1]. If the μ value is less than 0.5, the position is updated according to the average of four chimpanzees; otherwise, chaotic values are activated.

$$\overrightarrow{X}(t+1) = \begin{cases} \frac{\overrightarrow{X_1}(t)+\overrightarrow{X_2}(t)+\overrightarrow{X_3}(t)+\overrightarrow{X_4}(t)}{4} & if \quad \mu < 0.5 \\ Chaotic\,Value & if \quad \mu \geq 0.5 \end{cases} \tag{5}$$

The pseudocode of the standard ChOA, which was inspired by [55], is represented in Algorithm 1. You can find the detailed description of this pseudocode in the relevant reference. Also, the position update mechanism of the ChOA algorithm is presented in Fig. 2.

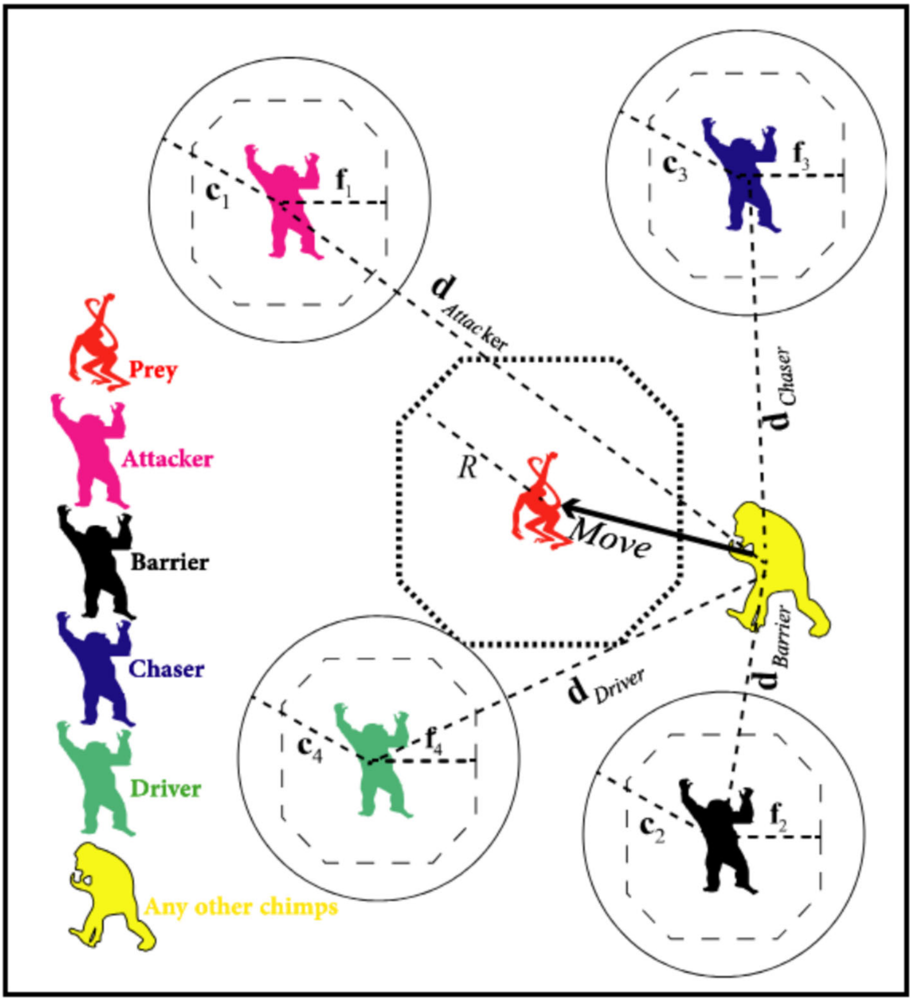

Fig. 2 Position updating in ChOA [55]

Algorithm 1 Pseudocode of ChOA

```
Initialize the population size: N and the maximum number of iterations: T
Initialize positions of chimps
Calculate f, m, a and C    //m based on Table 3; f and C based on Table 2 and Equation (2)
Calculate the fitness of each chimp
Select attacker, barrier, chaser, and driver
While (t<T)
  For each agent
    if (μ<0.5)
         Update the position of the current search agent by the Equation (5) based on position update mechanism
    else
         Update the position of the current search agent by the Equation (5) based on chaotic values
  end
  Update f, m, a and C    //m based on Table 3; f and C based on Table 2 and Equation (2)
  Update x_a, x_b, x_c, x_d
  t=t+1
end
return x_a
```

3 Expanded Chimp optimization algorithm

While the ChOA algorithm offers several advantages, it is prone to falling into local optima during the exploration phase. Additionally, it suffers from slow convergence and limited accuracy in high-dimensional problems. In spite of the fact that it tried to prevent this with dynamic coefficients and chaotic maps, it still suffers from these problems. Therefore, in this study, a multi-strategy-based algorithm is proposed to solve these problems and improve performance. In the proposed algorithm, each search agent (chimpanzee) expresses the values of the problem variables. It should be emphasized that in this study, explanations of all parameters used are presented in Appendix Table 22.

3.1 Population initialization

In this algorithm, each agent is allowed to work up to the dimensional (d) of the problem. The population is shown in Eq. 6, and the result of each member's action in all dimensions is represented by X's, as represented in Eq. 7. In these equations, the total number of agents is assumed to be 'n'.

$$\text{Chimp}_i = \{\text{Ch}_1, \text{Ch}_2, \ldots, \text{Ch}_n\};\ 1 < i \le n \tag{6}$$

$$X = \begin{bmatrix} X_{11} & \cdots & X_{1d} \\ \vdots & X_{ij} & \vdots \\ X_{n1} & \cdots & X_{nd} \end{bmatrix};\ 1 < i \le n,\ 1 \le j \le d \tag{7}$$

where, 'i' represents the index of each agent and 'j' denotes the number of dimensions. 'Xij' represents the action of the 'i^{th}' agent in the 'j^{th}' dimension. This is a two-dimensional matrix where the rows represent search agents, and the columns represent the dimensions of the problem or the number of uncertain parameters. Therefore, in the first step of the Ex-ChOA algorithm, a candidate matrix is created according to relevant matrix (Fig. 3). In this regard, a cost calculation is made for each agent as a result of its actions on all dimensions. This cost calculation is represented by a fitness function appropriate to the problem. The general definition of this function is presented in Eq. 8. In each iteration, the relevant function is evaluated for each member, and updates are made by getting the best solution as it moves

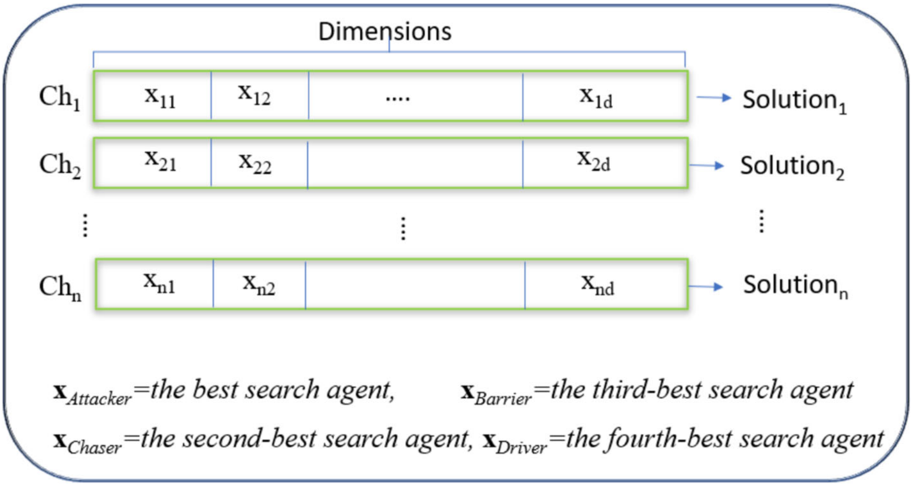

Fig. 3 Initial step of Ex-ChOA

on to the next iteration. In this study, the top four results are used for the next position updates. In each iteration, each agent finds a solution according to this fitness function. The four agents that find the best answer represent the groups defined in the algorithm. For the next iteration, while these four agents try to approach the prey, the other agents update their positions according to these four agents. Fitness calculation continues until the last iteration. At this stage, the best answer found is considered the most optimal solution that the algorithm can find for the relevant problem. The appropriate fitness function must be defined for each problem, and agents try to find answers accordingly. For example, they focus on finding the minimum or maximum.

$$\text{Fitness} = \begin{bmatrix} f(X_{11} \cdots X_{1d}) \\ \vdots \quad \ddots \quad \vdots \\ f(X_{n1} \cdots X_{nd}) \end{bmatrix} \tag{8}$$

3.2 Search and attack mechanisms

In the proposed algorithm, prey search and attack mechanisms are mathematically modeled below in different situations. These mechanisms are used in the exploration and exploitation phases of the algorithm. In these two phases, intelligent agents update their positions according to their fitness to find the best or a closed optimal solution. The position update mechanism occurs from two different perspectives, like standard ChOA. Therefore, the proposed algorithm is considered as Ex-ChOA1 and Ex-ChOA2. These are normal and chaos-based approaches. Equations 9 and 10 defined in normal position updating are used. While Eq. 9 is proposed for the exploration phase, Eq. 10 is introduced for the exploitation phase.

$$\overrightarrow{X_1}(\mathrm{t}) = \frac{\mathrm{C}_1}{2} \cdot \vec{X}_a - \mathrm{rand} \cdot \vec{X}(t); \quad \overrightarrow{X_2}(\mathrm{t}) = \frac{\mathrm{C}_2}{2} \cdot \vec{X}_\mathrm{b} - \mathrm{rand} \cdot \vec{X}(t)$$
$$\overrightarrow{X_3}(\mathrm{t}) = \frac{\mathrm{C}_3}{2} \cdot \vec{X}_\mathrm{c} - \mathrm{rand} \cdot \vec{X}(t); \quad \overrightarrow{X_4}(\mathrm{t}) = \frac{\mathrm{C}_4}{2} \cdot \vec{X}_\mathrm{d} - \mathrm{rand} \cdot \vec{X}(t) \tag{9}$$
$$\vec{\mathrm{X}}(\mathrm{t}+1) = \frac{4}{\frac{1}{\overrightarrow{X_1}(\mathrm{t})} + \frac{1}{\overrightarrow{X_2}(\mathrm{t})} + \frac{1}{\overrightarrow{X_3}(\mathrm{t})} + \frac{1}{\overrightarrow{X_4}(\mathrm{t})}}$$

The Ex-ChOA focused on harmonic mean instead of position update with arithmetic mean. A similar approach is also used in the exploitation phase. In short, in the normal position updating, the harmonic mean-based approach is recommended in this study. The '*a'* index was used for the Attacker chimpanzee, the *'b'* index for the Barriers chimpanzee, the *'c'* index for the Chaser chimpanzee, and the '*d'* index for the Driver chimpanzee. Based on this, 'X_a' shows the Attacker chimp's position and 'D_a' represents the Attacker chimp's distance relative to the prey. Similar considerations apply to chimpanzees in the other category. Equation 10 is proposed for the exploitation phase in normal position updating mechanism. The values of the '*C'* and '*f*' parameters are calculated according to Table 4. The value '*a'* is calculated according to Eq. 2.

$$\overrightarrow{D_a} = \left| C_1 \cdot \overrightarrow{X_a} - \vec{X}(t) \right|, \overrightarrow{X_1}(t) = \vec{X}_a - a_1 \cdot \overrightarrow{D_a};$$
$$\overrightarrow{D_b} = \left| C_2 \cdot \overrightarrow{X_b} - \vec{X}(t) \right|, \overrightarrow{X_2}(t) = \vec{X}_b - a_2 \cdot \overrightarrow{D_b}$$
$$\overrightarrow{D_c} = \left| C_3 \cdot \overrightarrow{X_c} - \vec{X}(t) \right|, \overrightarrow{X_3}(t) = \vec{X}_c - a_3 \cdot \overrightarrow{D_c};$$
$$\overrightarrow{D_d} = \left| C_4 \cdot -\vec{X}(t) \right|, \overrightarrow{X_4}(t)\vec{X}_d - a_4 \cdot \overrightarrow{D_d}$$
$$\vec{\mathrm{X}}(\mathrm{t}+1) = \frac{4}{\frac{1}{\overrightarrow{X_1}(\mathrm{t})} + \frac{1}{\overrightarrow{X_2}(\mathrm{t})} + \frac{1}{\overrightarrow{X_3}(\mathrm{t})} + \frac{1}{\overrightarrow{X_4}(\mathrm{t})}} \tag{10}$$

In this study, the characteristics of the concept of chaos are also used. Mathematically, chaos describes the randomness of a simple deterministic dynamical system, from which chaotic systems can be considered sources of randomness. In this regard, the chaotic maps are used to take advantage of the chaos feature in this algorithm [54]. In this way, problems such as local optimum trap, low search consistency, premature convergence, inefficient search, and low population diversity can be prevented or minimized. Therefore, in one of the strategies, chaotic maps with statistical and dynamic properties that have similar randomness properties are used in the proposed algorithm. As mentioned earlier, we assume that the probability of choosing between the normal or chaotic model in the position update is 50%. That is, with equal probability, either the natural hunting model is chosen, or the chaotic model is used to update the position. Equation 11 is suggested for the algorithm to work according to the chaotic map.

$$\overrightarrow{D_a} = \left| C_1 \cdot \overrightarrow{X_a} - \overrightarrow{m_1} \cdot \vec{X}(t) \right|, \overrightarrow{X_1}(t) = \vec{X}_a - a_1 \cdot \overrightarrow{D_a};$$
$$\overrightarrow{D_b} = \left| C_2 \cdot \overrightarrow{X_b} - \overrightarrow{m_2} \cdot \vec{X}(t) \right|, \overrightarrow{X_2}(t) = \vec{X}_b - a_2 \cdot \overrightarrow{D_b};$$
$$\overrightarrow{D_c} = \left| C_3 \cdot -\overrightarrow{m_3} \cdot \vec{X}(t) \right|, \overrightarrow{X_3}(t) = \vec{X}_c - a_3 \cdot \overrightarrow{D_c};$$
$$\overrightarrow{D_d} = \left| C_4 \cdot \overrightarrow{X_d} - \overrightarrow{m_4} \cdot \vec{X}(t) \right|, \overrightarrow{X_4}(t) = \vec{X}_d - a_4 \cdot \overrightarrow{D_d};$$

Table 4 Calculation of the coefficients of the f and C parameters

Algorithm	f	C
Ex-ChOA1	Equation 14	Equation 3
Ex-ChOA2	Table 2	Equation 13

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1}^2(t) + \overrightarrow{X_2}^2(t) + \overrightarrow{X_3}^2(t) + \overrightarrow{X_4}^2(t)}{4\left[\overrightarrow{X_1}(t) + \overrightarrow{X_2}(t) + \overrightarrow{X_3}(t) + \overrightarrow{X_4}(t)\right]} \tag{11}$$

As a result, the working model of the proposed algorithm is summarized in Eq. 12. Additionally, if the values of the 'f' parameter are distributed evenly, the value of the 'a' parameter will also be balanced. and therefore, the chances of operations between the two phases will be appropriate according to the problem. In other words, 'a' is a random value in the interval [-2f, 2f] where 'f' is reduced over iterations both linearly and nonlinearly in two different scenarios named Ex-ChOA1 and Ex-ChOA2. These two scenarios differ only in the calculation of the 'f' and 'C' parameters, and the main backbone operation of the algorithm is the same in both.

$$\overrightarrow{X}(t+1) = \begin{cases} Equation\ 9 & if \quad (\mu < 0.5 \text{ and } |a| \geq 1) \\ Equation\ 10 & if \quad (\mu < 0.5 \text{ and } |a| < 1) \\ Equation\ 11 & if \quad \mu \geq 0.5 \end{cases} \tag{12}$$

To evaluate the performance of Ex-ChOA with chaotic maps, 12 chaotic maps were used as shown in Table 5. In order to increase the evaluation accuracy and attention of the proposed algorithm, it will be possible to analyze the performances in different scenarios by doubling the number of maps. In general, the creation of various models makes the algorithm more flexible and can be used in various types of problems. The chaotic map can increase the cost of transition costs while avoiding falling into local optima. It also causes the algorithm to behave semi-deterministic [54].

The 'C' parameter supports exploration and avoidance of the local optimum, with a more random behavior throughout the optimization. Two different scenarios are assumed for our proposed algorithm here. In one of them (Ex-ChOA2), the 'C' value is based on a completely random model, while in the other (Ex-ChOA1), the effect of the f parameter in the calculation of this parameter is also reflected (Eq. 13). Therefore, in the Ex-ChOA1 scenario, besides the randomness, the parameter 'f' influences both 'a' and 'C'. For the calculation of 'f', two different methods are used for the two scenarios. Ex-ChOA1 uses a dynamic approach, which is presented in Table 2, and Ex-ChOA2 behaves based on Eq. 14. These two parameters ('C' and 'f') are summarized in Table 4.

$$C = 2f \bullet r_3 \tag{13}$$

$$f = s - s\left(\frac{\sqrt[T]{e^t} - 1}{e - 1}\right); s = \begin{cases} 2^k; k > 1 \\ \cos(e * \text{rand}) + 1 \end{cases}; \tag{14}$$

where 'r_3' is a random value in [0,1]. Besides, 's' parameter can be obtained in two different ways. This ensures that the useful one can be selected for different kinds of problems. In this study, a value of 2 is assumed for the 's' parameter in Ex-ChOA2, but different values or our other defined method can also be used. Based on this value, we are giving more than a 60% chance of exploration. On the other hand, if $''s = cos(e * rand) + 1''$ is assumed, random transitions between the two phases will be possible and there will be a mutant-like behavior. The point to be noted is that when s is kept equal with $''cos(e * rand) + 1''$, the first iteration will always be 2 and the last iteration will always be 0. Due to this behavior of the proposed algorithm, it has fast and accurate convergence.

The working mechanism of the Ex-ChOA algorithm in position update is shown schematically in Fig. 4. In this figure, X_P(t) represents the location of the prey (solution) at time "t". In fact, X_P is the vector of prey position. In addition to this, the flowchart and pseudocode of the Ex-ChOA are given in Fig. 5 and Algorithm 2, respectively.

Algorithm 2 Pseudocode of Ex-ChOA algorithm

```
Input: The population size N and total number of iterations T
Calculate the fitness function based on the objective function.
Initialize f, m, a and C      //m based on Table 5; f and C based on Table 4
Calculate the position of each chimp.
Divide chimps randomly into independent groups.
Select attacker, barrier, chaser, and driver.
  while (t < T)
    for each search agent
         if (μ<0.5)
              if (|a| >= 1)
                Update the position of the current search agent by the Equation (9)
             else
                Update the position of the current search agent by the Equation (10)
             end
        else
             Update the position of the current search by the Equation (11)
        end
    end
  Update f, m, a and C
  Update x_a, x_b, x_c, x_d
  t=t+1
 end
return x_a
```

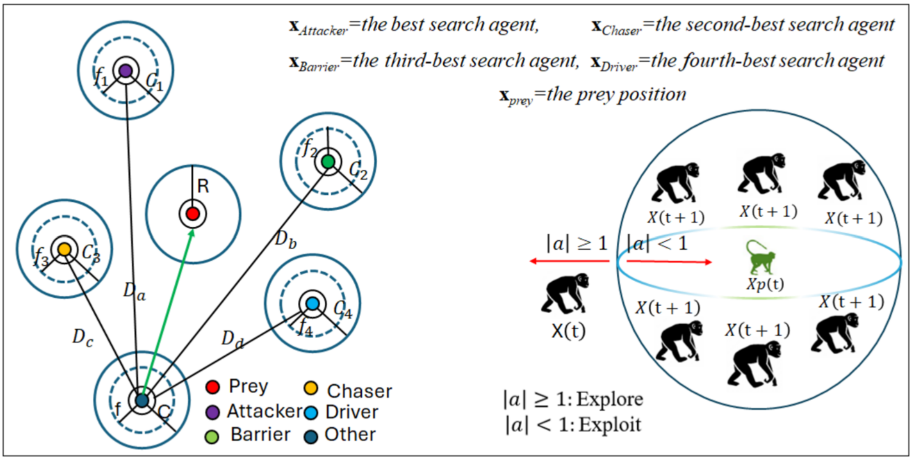

Fig. 4 Working mechanism of Ex-ChOA algorithm in position updating

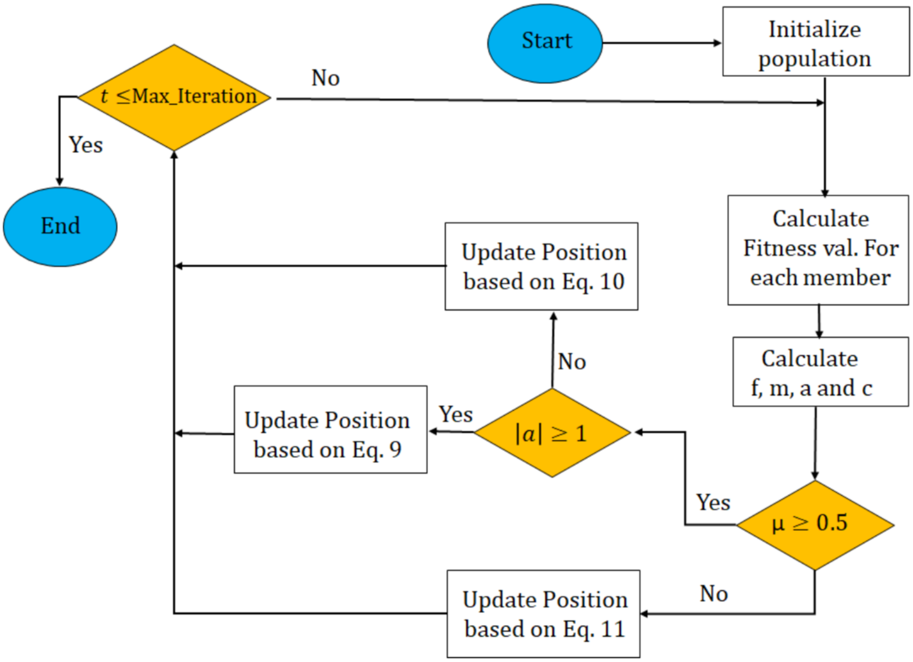

Fig. 5 The flowchart of the Ex-ChOA algorithm

3.3 Complexity analysis

Computational complexity qualitatively represents the running time of an algorithm. The parameters chimpanzee, size and total number of iterations and cost of the function are important in calculating the computational complexity of Ex-ChOA. These are represented by 'n', 'd', 'T', and 'F', respectively. While the initial complexity is O (N) in the worst case, it is equal to O (T × N × F) in the functional evaluation. This is also evident from the pseudocode presented in Algorithm 2. Therefore, while the proposed Ex-ChOA algorithm improves the performance of the standard ChOA algorithm, it is observed to be equal to the original ChOA in terms of complexity analysis. This means that the working mechanism of the Ex-ChOA algorithm is scientific.

4 Results and discussion

4.1 Simulation parameters and setup phase

In this section, the performance of the Ex-ChOA algorithm is analyzed over 34 benchmark functions and complex problems from CEC2015, and 2019 [59, 60]. The 24 functions of CECE 2015 are unimodal, multi-modal, and fixed-dimension multi-modal. Moreover, in another subsection, 10 CEC-C06 complex functions are selected for evaluation. These are competition functions that have been selected from IEEE Congress of Evolutionary Computation Benchmark Test Functions [60]. Both categories of selected benchmark functions are well-known functions utilized by many researchers [61–63]. The detailed information on each test function is represented in Appendixes Tables 23 and 24. These functions are used to examine the effectiveness of the Ex-ChOA algorithm. In unimodal benchmark functions, there is only a general optimum without any local optimum. Multi-modal functions, like unimodal, have only one global optimum but can have more than one local optima. Such benchmarking functions generally play an important role in the performance measurement of metaheuristic algorithms in the exploration and exploitation phases and are widely used in the literature. In the fixed-size multi-mode functions, the size of each comparison function is static and cannot be set to a size number, whereas, in the multi-mode function, it can be set.

The obtained results are compared with the related state-of-the-art studies in recent years (ChOA1 [55], ChOA2 [55], W-ChOA [57], E-ChOA [62], and I-GWO [63]). The standard and several novel variants of ChOA are discussed in the comparisons. In addition to these, a good-performance GWO-based algorithm was chosen. This is because the position update mechanism is like the chimp algorithm. The proposed algorithm is evaluated in two different scenarios (Ex-ChOA1, Ex-ChOA2) and tried to reveal its performance in different situations. All algorithms in this study were simulated using MATLAB, on the same PC with a Core i7 processor and 16GB of RAM. In addition, Table 6 presents the specific simulation parameters of each algorithm. Specific and general simulation parameters are defined for all algorithms used in the study. In this study, the number of populations, maximum iteration and total number of runs are assumed to be 50, 500 and 10, respectively, and these parameters are common for all used algorithms.

4.2 CEC2015 benchmark functions

In this section, the proposed algorithm is analyzed independently on several famous CEC2015 functions, and their results are compared with other metaheuristic algorithms. Tables 7, 8, and 9 represent the performance of all algorithms over 23 functions of CEC2015. Detailed information on these functions is presented in Appendix Table 23. As mentioned before, these functions consist of unimodal, multi-modal, and fixed-dimension multi-modal test functions. The results are evaluated based on three metrics (*Best*, *Mean*, and *Std*).

According to the results, the Ex-ChOA1 algorithm found the best solution in all functions except for F_5 F_6, F_8, F_{15}, and F_{21}. Also, Ex-ChOA2 algorithm outperforms than other in F_6, F_9, F_{10}, F_{11}, and from F_{14} to F_{19}. When we compare the performance of the algorithms, we mainly pay attention to the mean and standard deviation values. If the performance of the proposed algorithm in two scenarios is examined, it is determined that it cannot find the best answer only in F_5, F_8 and F_{21}, and this demonstrates the success of the general working model of the Ex-ChOA algorithm. Ex-ChOA1 performed best from F_1 to F_4. In the F_1 evaluation, Ex-ChOA1 ranked first, while W-ChOA, Ex-ChOA2, E-ChOA, I-GWO, ChOA2, and ChOA1 algorithms take place in the other ranks, respectively. In the F_2, F_3, and F_4 evaluation, those who follow the Ex-ChOA1 algorithm, which is by far the first, are the same as in F_1. When the F_5 results are analyzed, it is understood that the situation is different from the previous functions. I-GWO algorithm is in the first place in F_5. While the E-ChOA algorithm is in second place, Ex-ChOA1, Ex-ChOA2 algorithms are in the third and fourth places. In this function, the ChOA2 and W-ChOA algorithms are in 5th place together. The worst ranked algorithm is ChOA1. In the evaluation of F_6, Ex-ChOA2 and Ex-ChOA1 reserved the first two rows. In other ranks were the I-GWO, ChOA1, W-ChOA, E-ChOA, and ChOA2 algorithms. In the evaluation of F_7, Ex-ChOA1 and Ex-ChOA2 took the first two places, and I-GWO, W-ChOA, E-ChOA, ChOA1, and ChOA2 algorithms were in the other

Table 5 Chaotic maps used in Ex-ChOA

No	Name	Chaotic map	Range
1	Chebyshev	$X_{i+1} = \cos\left(icos^{-1}x_i\right)$	(−1, 1)
2	Circle	$X_{i+1} = \mathrm{mod}\left(x_i + \mathrm{b} - \left(\frac{a}{2\pi}\right)\sin(2\pi x_i), 1\right),\ \mathrm{a} = 0.5,\ \mathrm{b} = 0.2$	(0, 1)
3	Iterative	$X_{i+1} = \sin\left(\frac{a\pi}{x_i}\right), a = 0.7$	(−1, 1)
4	Piecewise	$X_{i+1} = \begin{cases} \frac{x_i}{P} & 0 \le x_i < P \\ \frac{x_i - P}{0.5 - P} & P \le x_i < 0.5 \\ \frac{1 - P - x_i}{0.5 - P} & 0.5 \le x_i < 1 - P \\ \frac{1 - x_i}{P} & 1 - P \le x_i < 1 \end{cases} \quad P = 0.4$	(0, 1)
5	Sinusoidal	$X_{i+1} = \mathrm{a}{X_i}^2\sin(\pi x_i), a = 2.3$	(0, 1)
6	Sine	$X_{i+1} = \frac{\mathrm{a}}{4}\sin(\pi x_i), a = 4$	(0, 1)
7	Quadratic	$X_{i+1} = X_i^2 - c, c = 1$	(0, 1)
8	Gauss/mouse	$X_{i+1} = \begin{cases} 1 & X_i = 0 \\ \frac{1}{mod(X_i, 1)}, & otherwise \end{cases}$	(0, 1)
9	Logistic	$X_{i+1} = \mathrm{a}X_i(1 - X_i), a = 4$	(0, 1)
10	Singer	$X_{i+1} = \mu\left(7.86X_i - 23.31{X_i}^2 + 28.75{X_i}^3 - 13.302875{X_i}^4\right), \mu = 1.07$	(0, 1)
11	Bernoulli	$X_{i+1} = 2X_i(mod1)$	(0, 1)
12	Tent	$X_{i+1} = \begin{cases} \frac{X_i}{0.7} & X_i < 0.7 \\ \frac{10}{3}(1 - X_i), & X_i \ge 0.7 \end{cases}$	(0, 1)

Table 6 Experimental parameters

Algorithm	Parameter	Value	Algorithm	Parameter	Value
ChOA/ W-ChOA/ E-ChOA	f	Table 2	Ex-ChOA	f	Table 4
	m	Table 3		m	Table 5
	a	[−2, 2]		C	Table 4
				a	[−2, 2]
	C	2.rand (0,1)	I-GWO	C	2.rand (0,1)
				a	[−2, 2]

ranks, respectively. It is understood from the results that the I-GWO algorithm performed successfully in F_8. Ex-ChOA2 took the second place, while Ex-ChOA1, ChOA2, E-ChOA, W-ChOA, and ChOA1 algorithms took place in the other ranks.

In the analysis of F_9, it is understood that the best performance belongs to Ex-ChOA1 and Ex-ChOA2, while the worst performance belongs to the ChOA1 algorithm. In a similar analysis, the best result for F_{10} is obtained by Ex-ChOA1 and Ex-ChOA2 algorithms, while the weakest performance belongs to Ex-ChOA2 algorithm. Except for ChOA1, ChOA2 in

Table 7 The simulation results on the test benchmark functions (CEC2015)

Algorithm		F1	F2	F3	F4	F5	F6	F7	F8
Ex-ChOA1	Best	**1.31E−280**	**6.45E−143**	**1.09E−273**	**7.46E−138**	2.81E + 01	9.95E−01	**9.19E−06**	−6.38E + 03
	Mean	**2.46E−279**	**5.08E−140**	**1.34E−272**	**1.26E−137**	2.83E + 01	1.70E + 00	**8.19E−05**	−5.92E + 03
	Std	**0.00E + 00**	**7.18E−140**	**0.00E + 00**	**7.21E−138**	1.96E + 01	4.59E−01	**7.96E−05**	6.77E + 01
Ex-ChOA2	Best	4.61E−108	2.51E−57	8.78E−101	5.71E−52	2.81E + 01	**3.83E−01**	2.21E−05	−7.51E + 03
	Mean	1.14E−103	1.05E−55	7.15E−95	1.68E−49	2.85E + 01	**6.62E−01**	8.98E−05	−6.16E + 03
	Std	3.33E−103	1.56E−55	2.08E−94	3.45E−49	3.11E−01	**9.90E−01**	4.48E−05	6.73E + 02
ChOA1	Best	2.42E−10	2.10E−06	6.63E−01	3.45E−02	2.87E + 01	3.03E + 00	2.33E−04	−3.83E + 03
	Mean	6.84E−06	2.59E−05	3.75E + 01	4.01E−01	2.89E + 01	3.75E + 00	1,86E−03	−2.67E + 03
	Std	1.25E−05	2.95E−05	2.09E + 01	3.96E−01	1.00E−01	4.62E−01	2,07E−03	6.21E + 02
ChOA2	Best	7.69E−13	3.42E−09	5.97E−02	1.52E−04	2.87E + 01	6.34E + 00	7.94E−05	−5.77E + 03
	Mean	2.56E−10	5.74E−08	5.94E + 00	1.11E−01	2.88E + 01	6.59E + 00	2,42E−03	−5.67E + 03
	Std	4.81E−10	7.95E−08	1.26E + 01	1.83E−02	1.05E−01	3.52E−01	2,32E−03	5.05E + 01
W-ChOA	Best	1.94E−150	6.30E−79	5.78E−135	2.03E−66	2.87E + 01	2.84E + 00	4.26E−05	−3.74E + 03
	Mean	8.96E−146	1.79E−69	2.99E−125	5.20E + 01	2.88E + 01	4.07E + 00	2.78E−04	−3.37E + 03
	Std	1.62E−145	5.65E−69	8.49E−125	4.49E + 01	1.77E−01	4.58E−01	2.48E−04	3.02E + 02
E-ChOA	Best	9.25E−36	9.87E−22	4.20E−09	1.77E−11	2.62E + 01	4.34E + 00	2.25E−04	−6.18E + 03
	Mean	1.65E−33	9.55E−21	1.73E−07	5.42E−10	2.74E + 01	5.08E + 00	9.65E−04	−4.36E + 03
	Std	2.13E−33	5.80E−21	2.24E−07	9.63E−10	7.46E−01	4.85E−01	7.46E−04	1.17E + 03
I-GWO	Best	8.39E−28	4.21E−17	4.64E−07	1.76E−07	**2.61E + 01**	1.81E + 00	2.76E−04	**−8.49E + 03**
	Mean	8.39E−28	4.21E−17	4.64E−07	7.95E−07	**2.70E + 01**	2.18E + 00	1,45E−03	**−6.43E + 03**
	Std	0.00E + 00	0.00E + 00	0.00E + 00	8.07E−07	**6.62E−01**	2.00E−01	7.32E−04	**3.86E + 01**

The optimum results obtained from the algorithms are bold and highlighted

Table 8 The simulation results on the test benchmark functions (CEC2015)-Cont

Algorithm		F9	F10	F11	F12	F13	F14	F15	F16
Ex-ChOA1	Best	**0.00E + 00**	**4.44E−16**	**0.00E + 00**	**1,19E−02**	**3.05E−01**	**9.98E−01**	7.75E−04	**−1.03E + 00**
	Mean	**0.00E + 00**	**4.44E−16**	**0.00E + 00**	**3,60E−02**	**5.40E−01**	**9.98E−01**	8,51E−03	**−1.03E + 00**
	Std	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**1,41E−02**	**2.39E−01**	**9.85E−06**	1,73E−02	**4.38E−07**
Ex-ChOA2	Best	**0.00E + 00**	**4.44E−16**	**0.00E + 00**	3.82E−01	2.46E + 00	**9.98E−01**	**3.08E−04**	**−1.03E + 00**
	Mean	**0.00E + 00**	**4.44E−16**	**0.00E + 00**	6.65E−01	2.75E + 00	**9.98E−01**	**4,40E−04**	**−1.03E + 00**
	Std	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	2.05E−01	7.52E−02	**9.16E−05**	**6,30E−03**	**3.90E−07**
ChOA1	Best	3.88E−05	1.95E + 00	2.28E−07	3.41E−01	2.21E + 00	9.98E−01	1,25E−03	**1.03E + 00**
	Mean	6.99E + 00	1.96E + 00	2,15E−03	5.13E−01	2.65E + 00	1.14E + 00	1,29E−03	**1,03E + 00**
	Std	9.06E + 00	1,46E−05	2,86E−03	1.83E−01	2.00E−01	4.77E−01	3.21E−05	**1.96E−04**
ChOA2	Best	2.55E−12	1.85E + 00	7.77E−12	1.93E−01	2.67E + 00	2.98E + 00	1,25E−03	**1.03E + 00**
	Mean	7.69E−04	1.85E + 00	1,57E−03	4.69E−01	2.78E + 00	2.98E + 00	8,78E−03	**1,03E + 00**
	Std	1,99E−03	1,01E−06	2,76E−03	2.79E−01	1.45E−01	7.89E−03	1,78E−02	**1,00E−02**
W-ChOA	Best	0.00E + 00	4.00E−15	0.00E + 00	1,17E−01	2.52E + 00	9,98E−01	6.05E−04	**−1.03E + 00**
	Mean	1.14E + 00	4.00E−15	1,90E−02	1,79E−01	2.64E + 00	1.12E + 00	4,41E−03	**−1.03E + 00**
	Std	3.14E + 00	0.00E + 00	3,21E−02	3,24E−02	9.07E−02	3,11E−01	9,60E−03	**9,86E−03**
E-ChOA	Best	0.00E + 00	1.47E−14	**0.00E + 00**	1.05E + 00	3.98E−01	9.98E−01	3.09E−04	**−1.03E + 00**
	Mean	1.14E−14	2.25E−14	**0.00E + 00**	1.09E + 00	1.11E + 00	3.55E + 00	5.90E−04	**−1.03E + 00**
	Std	3.59E−14	5.50E−15	**0.00E + 00**	1.86E−02	3.31E−01	3.91E + 00	1.19E−04	**6.14E−04**
I-GWO	Best	5.68E−14	8.57E−14	**0.00E + 00**	8.98E−01	1.76E + 00	9.98E−01	5.16E−04	**−1.03E + 00**
	Mean	5.58E + 00	1.18E−13	**0.00E + 00**	1.02E + 00	1.88E + 00	2.07E + 00	6.98E−04	**−1.03E + 00**
	Std	5.12E + 00	2.33E−14	**0.00E + 00**	4.95E + 00	5.10E−02	3.07E + 00	1.33E−04	**2.59E−06**

The optimum results obtained from the algorithms are bold and highlighted

Table 9 The simulation results on the test benchmark functions (CEC2015)-Cont

Algorithm		F17	F18	F19	F20	F21	F22	F23	F24
Ex-ChOA1	Best	**3.98E−01**	**3.00E + 00**	**−3.86E + 00**	**−3.32E + 00**	−10.15E + 00	**−10.40E + 00**	**−10.54E + 00**	**4.01E−272**
	Mean	**3.98E−01**	**3.00E + 00**	**−3.86E + 00**	**−3.24E + 00**	−6.65E + 00	**−10.40E + 00**	**−10.50E + 00**	**1.01E−259**
	Std	**1.49E−04**	**2.03E−06**	**7.17E−07**	**9.12E−02**	4.21E + 00	**1,42E−03**	**1,34E−02**	**0.00E + 00**
Ex-ChOA2	Best	**3.98E−01**	**3.00E + 00**	**−3.86E + 00**	−3.02E + 00	−10.15E + 00	−10.39E + 00	−10.54E + 00	2.05E−203
	Mean	**3.98E−01**	**3.00E + 00**	**−3.86E + 00**	−2.90E + 00	−7.17E + 00	−10.35E + 00	−9.96E + 00	2.90E−193
	Std	**2.02E−04**	**1.21E−07**	**5.52E−06**	2.53E−01	3,35E + 00	2,32E−02	1,71E + 00	0.00E + 00
ChOA1	Best	**3.98E−01**	**3.00E + 00**	**−3.86E + 00**	−3.05E + 00	−10.15E + 00	−10.23E + 00	−10.14E + 00	4.22E−09
	Mean	**3.98E−01**	**3.00E + 00**	**−3.86E + 00**	−2.55E + 00	−6.60E + 00	−8.38E + 00	−9.24E + 00	1.07E−04
	Std	**7.62E−04**	**2.50E−04**	**3.42E−03**	4.56E−01	3.17E + 00	3.46E + 00	2.58E + 00	2.67E−04
ChOA2	Best	3.98E−01	**3.00E + 00**	−3.86E + 00	−3.13E + 00	−10.15E + 00	−10.14E + 00	−10.03E + 00	1.69E−17
	Mean	8.64E−01	**3.00E + 00**	−3.85E + 00	−2.52E + 00	−5.70E + 00	−7,40E + 00	−9,90E + 00	1.74E−05
	Std	1.47E + 00	**2.10E−04**	3.41E−03	5.34E−01	4.39E + 00	3,85E + 00	1,19E−01	4.02E−05
W-ChOA	Best	3.98E−01	**3.00E + 00**	−3.85E + 00	−3.04E + 00	−10.15E + 00	−10.38E + 00	−10.52E + 00	8.59E−82
	Mean	4.14E−01	**3.00E + 00**	−3.79E + 00	−2.16E + 00	−6.57E + 00	−10.35E + 00	−8.53E + 00	1.01E−27
	Std	1.81E−02	**1.61E−04**	1.02E−01	5.71E−01	2.44E + 00	2,45E−02	3.22E + 00	3.20E−07
E-ChOA	Best	**3.98E−01**	**3.00E + 00**	**−3.86E + 00**	−3.22E + 00	−10.15E + 00	−10.39E + 00	−10.47E + 00	7.23E−40
	Mean	**3.98E−01**	**3.00E + 00**	**−3.86E + 00**	−3.09E + 00	−6.61E + 00	−8.68E + 00	−9.82E + 00	1.49E−26
	Std	**4.94E−04**	**2.71E−05**	**3,18E−03**	6.69E−01	3.17E + 00	2.77E + 00	1.69E−01	4.72E−26
I-GWO	Best	**3.98E−01**	**3.00E + 00**	**−3.86E + 00**	−3.32E + 00	**−10.15E + 00**	−10.38E + 00	−10.54E + 00	1.52E−28
	Mean	**3.98E−01**	**3.00E + 00**	**−3.86E + 00**	−3.16E + 00	**−9.65E + 00**	−9.31E + 00	−9.99E + 00	6.73E−20
	Std	**6.67E−03**	**4.55E−05**	**1,30E−03**	1.50E−01	**1.59E + 00**	2.22E + 00	1.71E + 00	1.97E−19

The optimum results obtained from the algorithms are bold and highlighted

F_{11}, all the other algorithms found the optimum result. As it turns out, some functions seem to have more than one good answer algorithm (for example, F_{11}). In this study, we counted all of them as full points, but for detailed reporting, the best answer in which iteration they found is presented below. Ex-ChOA1 found the best response in the F_{11} function at the 17th iteration. For example, in the best case, Ex-ChOA2 found the best answer in the 39th iteration, the I-GWO algorithm in the 201st iteration, the E-ChOA algorithm in the 245th iteration, and the W-ChOA algorithm in the 432nd iteration. The best result in F_{12} belonged to Ex-ChOA1 algorithm again, while W-ChOA, ChOA2, ChOA1, Ex-ChOA2, I-GWO, and E-ChOA algorithms took place in the other ranks. While Ex-ChOA1 algorithm is in the first place in F_{13}, the worst result was obtained from ChOA2 algorithm. In the F_{14} function, the best results are obtained by the proposed algorithm, while the weakest performance belongs to the E-ChOA algorithm. In F_{15}, Ex-ChOA2 was the best and ChOA2 algorithm was the worst. Finally, the F_{16} and F_{18} results were examined. According to the results, all algorithms were successful in finding the optimum value.

In F_{17} and F_{19}, all algorithms found the best results, except the ChOA2 and W-ChOA algorithms. In F_{20}, the Ex-ChOA1 algorithm found the best result, while the weakest performance belonged to the W-ChOA algorithm. In F_{21}, the best performance belongs to I-GWO. The proposed algorithm is ranked second in this function. It is understood from the analysis of the results that the Ex-ChOA1 algorithm found the best result in F_{22}, F_{23} and F_{24}. As a result, the analysis results for all 24 functions were obtained as follows. The Ex-ChOA1 took first place, followed by Ex-ChOA2. The W-ChOA and I-GWO algorithms took third place jointly. However, we should not ignore that I-GWO has managed to achieve the best results in three functions alone. The E-ChOA algorithm took fifth place, followed by ChoA2 and ChoA1, respectively. In these 24 test functions, in total, Ex-ChOA1 achieved the best result with 55.5%, and Ex-ChOA2 achieved this success with 18.46%, taking the second place. Following these two is the I-GWO algorithm, which ranked third with a rate of 16.42%. The E-ChOA algorithm ranked fourth with a success rate of 3.92%, ChOA1 was in fifth place with a success rate of 2.87%, and W-ChOA and ChOA2 came in next with a success rate of 1.19% each.

Another analysis considered in this study is convergence behavior. In this context, the convergence with which the agents of each algorithm, especially the proposed algorithm, reaches the best answer they find is examined. For this analysis, each algorithm is applied 500 iterations on all functions, and the movements and behaviors of each algorithm in two phases are examined. It is also compared with other algorithms to get a clear understanding of its performance. In general, search agents in optimization algorithms can act in unexpected ways during their first steps, and this is considered a perfectly normal phenomenon in the literature [64, 65]. This change in movement causes the search space to be explored more broadly and, therefore, to benefit from it. However, these changes are normally reduced in final iterations for good use. Therefore, this analysis is considered an important tool used to show how well each algorithm performs. In other words, this analysis is an examination of how the algorithm progresses toward an optimal solution over time. It involves evaluating how the algorithm behaves as iterations progress and aims to understand how quickly and effectively the algorithm approaches a solution that meets a certain criterion or achieves a desired solution quality. The convergence behaviors of the algorithms are examined in Fig. 6, and according to the analysis, it is seen that Ex-ChOA1 outperforms the others. Regarding the results according to the working mechanism of this algorithm, search agents explore the area in early iterations and try to use them after certain iterations. At the same time, the chances of finding other possible solutions increase as the transitions between phases are faster.

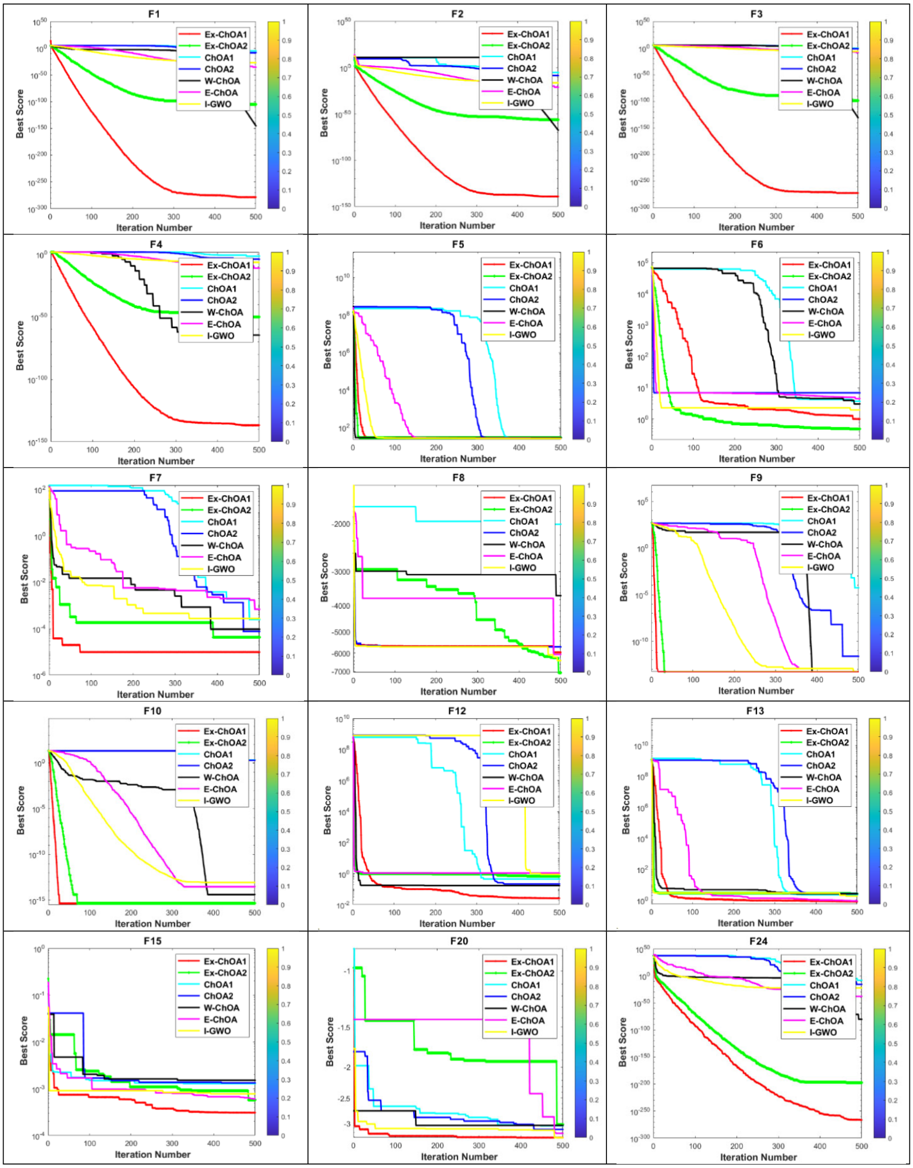

Fig. 6 The convergence curve analysis of each algorithm in some CEC2015 test functions

In addition to the analysis, statistical tests were carried out for the proposed algorithm. Tables 10 and 11 present the p-values calculated by the nonparametric Wilcoxon rank-sum tests for the pair-wise comparison over two independent samples. One of the tables represents Ex-ChOA1 and the other represents Ex-ChOA2. p-values are produced by the Wilcoxon test with a significance level of 0.05 and over 10 independent studies.

Table 10 Pair-wise comparison with Ex-ChOA1 and other algorithms

Functions	Ex-ChOA1 vs. Ex-ChOA2	Ex-ChOA1 vs. ChOA1	Ex-ChOA1 vs. ChOA2	Ex-ChOA1 vs. W-ChOA	Ex-ChOA1 vs. E-ChOA	Ex-ChOA1 vs. I-GWO
	h	*h*	*h*	*h*	*h*	*h*
1	+	+	+	+	+	+
2	+	+	+	+	+	+
3	+	+	+	+	+	+
4	+	+	+	+	+	+
5	+	+	+	+	−	−
6	−	+	+	+	+	+
7	+	+	+	+	+	+
8	−	+	+	+	+	−
9	~	+	+	+	+	+
10	~	+	+	+	+	+
11	~	+	+	+	~	~
12	+	+	+	+	+	+
13	+	+	+	+	+	+
14	~	+	+	+	+	+
15	−	−	+	−	−	−
16	~	~	~	~	~	~
17	~	~	+	+	~	~
18	~	~	~	~	~	~
19	~	~	+	+	~	~
20	+	+	+	+	+	+
21	−	+	+	+	+	+
22	+	+	+	+	+	+
23	+	+	+	+	+	+
24	+	+	+	+	+	+
Total + sign	**12**	**19**	**22**	**21**	**17**	**16**

In this table, the plus notation (+) indicates that the proposed algorithm is superior to the others. Also, the minus notation (-) means that the results of the proposed algorithm are worse than other comparison algorithms. Finally, the (~) notation represents that the pairwise comparison algorithm has equal rank. The sum (+) sign indicates how many times the proposed algorithm is better than the other in two different scenarios in pairwise comparisons over 24 functions

4.3 CEC-2019-CE06 competition benchmark functions

In this study, the performance of the proposed algorithm is also analyzed on the modern 10 benchmark functions of CEC-2019. They are the 100-digit challenge also known as "*CEC-C06 benchmark test functions*" [60]. Detailed information on these functions is presented in Appendix Table 24. These functions are utilized to evaluate the Ex-ChOA1 and Ex-ChOA2. All these functions are scalable, and their global optimum converges to 1. In terms of the general characteristics of these functions, the worst results are also presented for a better

Table 11 Pair-wise comparison with Ex-ChOA2 and other algorithms

Functions	Ex-ChOA2 vs. Ex-ChOA1	Ex-ChOA2 vs. ChOA1	Ex-ChOA2 vs. ChOA2	Ex-ChOA2 vs. W-ChOA	Ex-ChOA2 vs. E-ChOA	Ex-ChOA2 vs. I-GWO
	h	*h*	*h*	*h*	*h*	*h*
1	−	+	+	+	+	+
2	−	+	+	−	+	+
3	−	+	+	−	+	+
4	−	+	+	−	+	+
5	−	+	+	+	−	−
6	+	+	+	+	+	+
7	−	+	+	+	+	+
8	+	+	+	+	+	−
9	~	+	+	+	+	+
10	~	+	+	+	+	+
11	~	+	+	+	~	~
12	−	−	−	−	+	+
13	−	−	+	−	−	−
14	~	+	+	+	+	+
15	+	+	+	+	+	+
16	~	~	~	~	~	~
17	~	~	+	+	~	~
18	~	~	~	~	~	~
19	~	~	+	+	~	~
20	−	+	+	+	−	−
21	+	+	+	+	+	−
22	−	+	+	+	+	+
23	−	+	+	+	+	−
24	−	+	+	+	+	+
Total + sign	**4**	**18**	**21**	**17**	**16**	**13**

In this table, the plus notation (+) indicates that the proposed algorithm is superior to the others. Also, the minus notation (-) means that the results of the proposed algorithm are worse than other comparison algorithms. Finally, the (~) notation represents that the pairwise comparison algorithm has equal rank. The sum (+) sign indicates how many times the proposed algorithm is better than the other in two different scenarios in pairwise comparisons over 24 functions

understanding of the performance of each algorithm. The CEC01-CEC03 functions are non-shiftable, while CEC04-CEC10 functions can be shifted and rotated. The performance of each algorithm over these functions is presented in Table 12. According to the results obtained, it is understood that the proposed algorithm performs better than the others. Our algorithm found the best results in CEC2, 3, 4, 5, 7, and 8 in the Ex-ChOA1 scenario, while it ranked first in CEC1, 6, and 10 in Ex-ChOA2. In CEC01, Ex-ChOA1 is in second place, while W-ChOA, E-ChOA, I-GWO, ChOA2, and ChOA1 algorithms take place in the other ranks. In the CEC02 competitive function, E-ChOA and I-GWO algorithms ranked second and

Table 12 The simulation results on the competitive functions

Functions		Ex-ChOA1	Ex-ChOA2	ChOA1	ChOA2	W-ChOA	E-ChOA	I-GWO
CEC01	Best	5.23E + 04	**4.36E + 04**	4.91E + 05	5.84E + 04	5.39E + 04	6.95E + 05	5.41E + 06
	Worst	7.79E + 04	**9.33E + 04**	3.99E + 10	7.48E + 09	1.63E + 05	2.23E + 09	1.81E + 09
	Std	8.32E + 03	**1.64E + 04**	1.21E + 10	1.76E + 09	3.22E + 04	6.95E + 08	6.14E + 08
	Mean	6.49E + 04	**5.86E + 04**	6.47E + 09	1.18E + 09	9.77E + 04	4.46E + 08	4.96E + 08
CEC02	Best	**1.69E + 01**	1.76E + 01	1.74E + 01	1.74E + 01	1.75E + 01	1.73E + 01	1.73E + 01
	Worst	**1.79E + 01**	1.88E + 01	1.74E + 01	1.74E + 01	1.76E + 01	1.73E + 01	1.76E + 01
	Std	**2.95E−01**	3.91E−01	1.55E−03	1.50E−03	3.63E−02	2.36E−03	9.85E−02
	Mean	**1.72E + 01**	1.82E + 01	1.74E + 01	1.74E + 01	1.75E + 01	1.73E + 01	1.73E + 01
CEC03	Best	**1.27E + 01**	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01
	Worst	**1.27E + 01**	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01
	Std	**4.66E−07**	4.86E−07	1.23E−06	1.81E−06	9.38E−06	1.05E−06	1.12E−06
	Mean	**1.27E + 01**	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01
CEC04	Best	**3.57E + 01**	5.73E + 01	7.26E + 02	1.53E + 03	4.68E + 03	1.31E + 04	5.49E + 02
	Worst	**8.54E + 01**	1.36E + 02	8.99E + 03	1.14E + 04	6.64E + 03	3.05E + 04	1.31E + 03
	Std	**1.79E + 01**	2.41E + 01	2.95E + 03	3.66E + 03	1.33E + 03	7.81E + 03	3.08E + 02
	Mean	**5.42E + 01**	8.82E + 01	5.18E + 03	6.81E + 03	4.68E + 03	1.79E + 04	1.03E + 03
CEC05	Best	**1.06E + 00**	1.85E + 00	2.60E + 00	1.91E + 00	2.47E + 00	4.07E + 00	1.16E + 00
	Worst	**1.77E + 00**	4.07E + 00	3.83E + 00	4.69E + 00	3.52E + 00	6.21E + 00	1.81E + 00
	Std	**2.91E−01**	9.99E−01	3.58E−01	8.15E−01	3.51E−01	7.06E−01	2.11E−01
	Mean	**1.42E + 00**	2.50E + 00	3.00E + 00	2.92E + 00	2.79E + 00	5.66E + 00	1.50E + 00
CEC06	Best	8.99E + 00	**6.68E + 00**	1.03E + 01	1.02E + 01	1.06E + 01	1.11E + 01	1.02E + 01
	Worst	1.20E + 01	**1.15E + 01**	1.26E + 01	1.26E + 01	1.21E + 01	1.34E + 01	1.18E + 01

Table 12 (continued)

Functions		Ex-ChOA1	Ex-ChOA2	ChOA1	ChOA2	W-ChOA	E-ChOA	I-GWO
	Std	9.79E−01	**1.82E + 00**	7.09E−01	8.36E−01	4.12E−01	6.79E−01	5.22E−01
	Mean	1.09E + 01	**9.81E + 00**	1.17E + 01	1.14E + 01	1.14E + 01	1.21E + 01	1.11E + 01
CEC07	Best	**1.49E + 02**	3.00E + 02	7.64E + 02	6.65E + 02	7.25E + 02	6.48E + 02	5.61E + 02
	Worst	**9.98E + 02**	1.25E + 03	1.25E + 03	1.87E + 03	1.04E + 03	1.31E + 03	1.04E + 03
	Std	**2.48E + 02**	2.51E + 02	1.49E + 02	3.41E + 02	1.06E + 02	2.26E + 02	9.21E + 02
	Mean	**7.18E + 02**	8.10E + 02	1.11E + 03	9.78E + 02	8.62E + 02	9.39E + 02	6.71E + 02
CEC08	Best	**3.63E + 00**	3.79E + 00	6.67E + 00	6.35E + 00	5.84E + 00	5.49E + 00	5.49E + 00
	Worst	**5.33E + 00**	6.94E + 00	7.09E + 00	7.09E + 00	6.77E + 00	7.10E + 00	6.24E + 00
	Std	**4.92E−01**	1.08E + 00	1.32E−01	2.53E−01	3.12E−01	7.18E−01	2.89E−01
	Mean	**4.56E + 00**	5.48E + 00	6.86E + 00	6.76E + 00	6.13E + 00	6.53E + 00	5.90E + 00
CEC09	Best	2.77E + 01	1.18E + 02	4.68E + 01	1.55E + 02	5.46E + 01	4.13E + 00	**2.82E + 00**
	Worst	6.83E + 02	2.25E + 03	6.76E + 02	7.90E + 02	4.49E + 02	2.72E + 02	**5.59E + 00**
	Std	2.57E + 02	7.57E + 02	2.06E + 02	1.92E + 02	1.37E + 02	8.46E + 01	**1.03E + 00**
	Mean	4.80E + 02	1.36E + 03	3.93E + 02	4.62E + 02	2.89E + 02	3.16E + 01	**4.34E + 00**
CEC10	Best	2.02E + 01	**2.02E + 01**	2.02E + 01	2.03E + 01	2.04E + 01	2.03E + 01	2.05E + 01
	Worst	2.06E + 01	**2.05E + 01**	2.06E + 01	2.06E + 01	2.06E + 01	2.07E + 01	2.06E + 01
	Std	1.05E−01	**9.07E−02**	1.05E−01	7.63E−01	6.01E−02	9.48E−02	4.67E−02
	Mean	2.05E + 01	**2.03E + 01**	2.05E + 01	2.05E + 01	2.05E + 01	2.05E + 01	2.05E + 01

The optimum results obtained from the algorithms are bold and highlighted

third. Ex-ChOA2 took the last place. In CEC03, the results are very close to each other and with a slight difference, Ex-ChOA1 is in the first place and W-ChOA algorithm is in the last place. While Ex-ChOA1 was the first in CEC04 and CEC05, the E-ChOA algorithm took the last place. Again, poor performance in CEC06 belongs to E-ChOA algorithm and best result belongs to Ex-ChOA2. In CEC07 and CEC08, Ex-ChOA1 algorithm showed the best performance, while ChOA1 found the weakest result. In CEC08, the best result belongs to the I-GWO algorithm, and the worst belongs to the Ex-ChOA1 algorithm. In CEC10, the best performing algorithm was Ex-ChOA1, while the weakest was ChOA2 algorithm. According to these results, the proposed algorithm was successful in finding the best result in other functions except CEC09. As a result, Ex-ChOA1 ranked first in competitive functions with a 60% success rate. Ex-ChOA2 ranked second with 30% and the I-GWO algorithm ranked third with 10%. The remaining algorithms did not find the best results for any of the functions in this category, but W-ChOA was ranked fourth. Then, E-ChOA, ChOA2 and ChOA1 took place, respectively.

As shown in Fig. 7, the convergence behavior of the proposed algorithm and other methods is examined. As the iterations progress, the sensitivity ranges are reduced for attempting of search agents to use and find the global optimum. In these curves, sudden changes of the proposed algorithm are observed. These cause agents to discover a possible solution and take advantage of it.

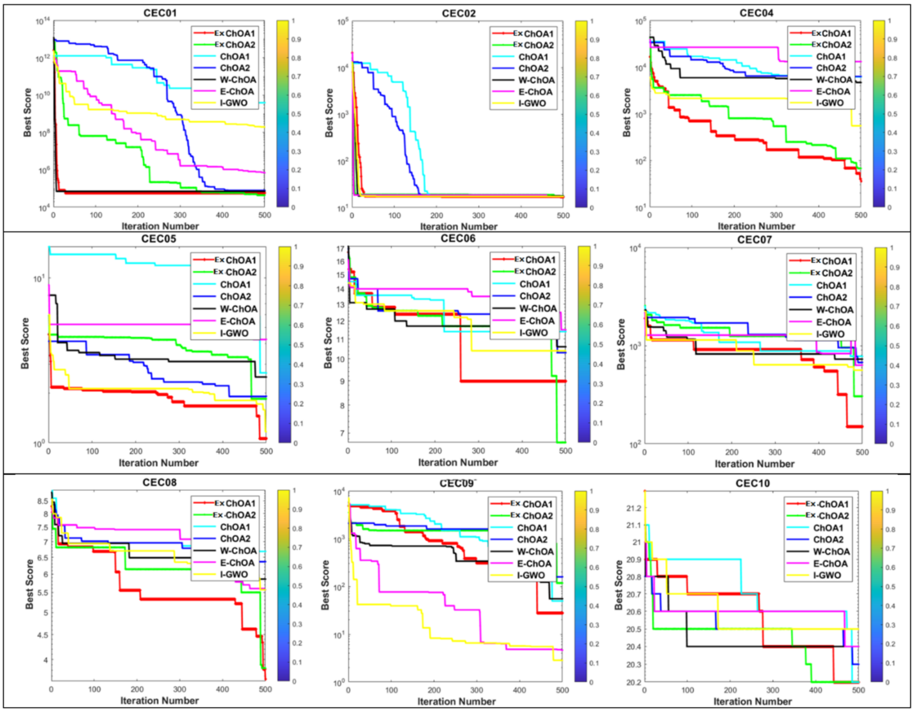

Fig. 7 The convergence curve analysis of each algorithm in CEC-2019 functions

4.4 Exploration–exploitation and diversity analysis

In the exploration phase, it evaluates how effective the algorithm is at searching for new and diverse solutions in the solution space. It usually aims to find potentially better solutions using random search or various search strategies. In this study, the probability increases even more due to the use of chaotic maps in combination with dynamic strategies. In the exploitation phase, it evaluates how effectively the algorithm moves toward the best available solution. It generally aims to refine the solution by concentrating on the region closest to the best-known solution. In the proposed algorithm, it is aimed to increase this sensitivity with the newly defined '*C*' and '*f*' parameters. Additionally, a position update strategy has been defined with two different approaches. The analysis in this section is used to measure the algorithm's ability to find solutions in a large search space, as well as the algorithm's ability to focus on and improve the best solution found. In Fig. 8, the behavioral analysis of the proposed algorithm in the exploration and exploitation phases is shown representatively on different functions. A sample is taken from each benchmark function, and its behavior is analyzed. As can be seen, in unimodal and multi-modal functions, the exploration and exploitation rates of the algorithm are close to each other, while the rates vary in the other two types of functions. In the analysis, it was observed that while other algorithms focused on exploitation after a while, the proposed algorithm maintained a certain and stable trend. As a matter of fact, while in some scenarios the emphasis is on the discovery phase, in some cases it is possible to follow the opposite results.

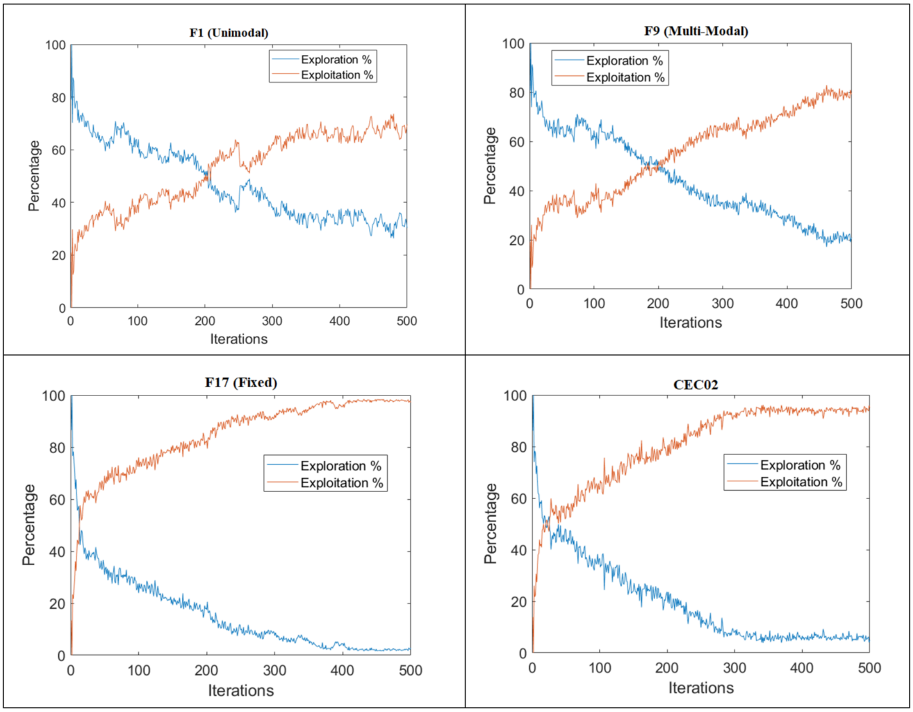

Fig. 8 Exploration and exploitation of proposed algorithm on some types of benchmark functions

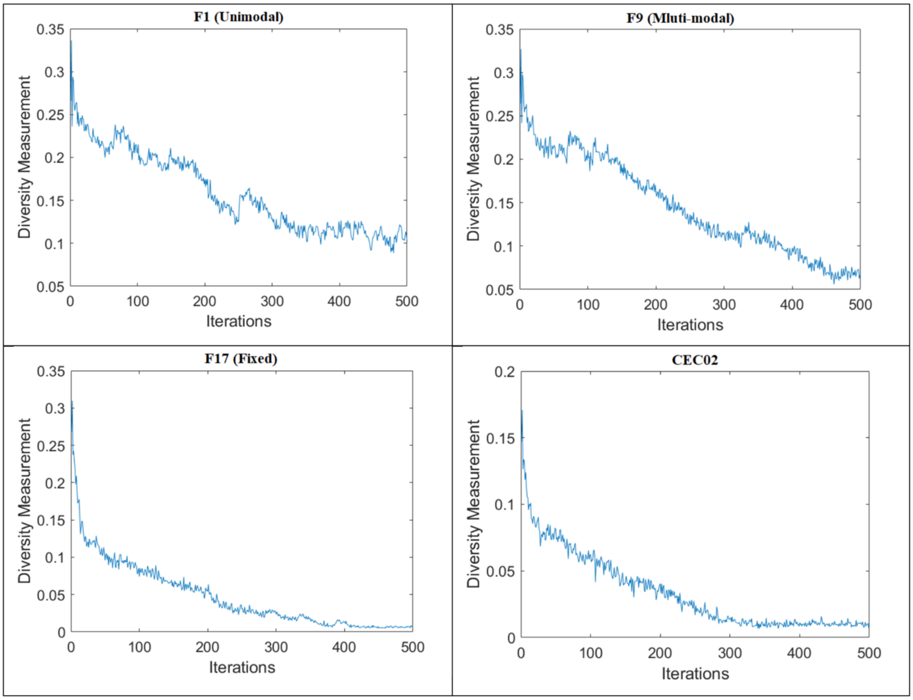

Fig. 9 Diversity analysis of the proposed algorithm on some types of benchmark functions

The other analysis performed is diversity analysis. In general, algorithms that emphasize exploitation phase will have a good convergence rate. At the same time, the risk of falling into an optimum trap is reduced. Additionally, when the exploration phase is effectively concentrated, the chance of thoroughly scanning the search space and therefore finding the global solution increases. Moreover, diversity is important for the robustness and effectiveness of metaheuristic algorithms. However, the probability of the convergence speed decreasing increases. While diversity prevents premature convergence by encouraging exploration of the search, too much diversity can prevent convergence toward the optimal solution. Therefore, in a balanced and reasonable manner, algorithms should give importance to both phases. This analysis was inspired by some studies in literature [66]. Since the proposed algorithm makes use of chaos theory, the diversity level of the population can be maintained. Additionally, due to the use of various strategies, the proposed algorithm reduces the possibility of falling into the local optimum trap by focusing on the diversity level of the population. The analysis results of the proposed approach are presented in Fig. 9, covering the whole population according to the characteristics of each sampled function.

5 Real-world engineering optimization problems

In this section, the performance of the proposed algorithm is analyzed in solving optimization problems in various engineering fields. In this regard, 10 limited modern problems are taken from CEC2020 [67], the other two are classical engineering problems, and the last problem is a modern and complex type of a multi-objective-based problem related to microwave design.

5.1 Modern constrained engineering optimization problems

In this study, real-world engineering problems are used in the performance analysis of the proposed algorithm, as well as a wide variety of global test functions in the previous section. In this regard, several of the complex real-world problems in various fields of engineering are addressed at CEC 2020. These problems are categorized as follows. (1) Industrial Chemical Processes, 2() Process Synthesis and Design Problems, (3) Mechanical Engineering Problems, (4) Power System Problems, and (5) Power Electronic Problems groups. In this study, we selected two problems from each group, as presented in Table 13. In this table, the '*D*' represents the number of decision variables of the problem. The '*g*' is the number of inequality constraints, and '*h*' is the number of equation constraints. The complex optimization problems arising from real-life applications are generally referred to as real-world problems.

Solving these optimization problems is considered relatively difficult due to the complex nature of the functions involved with many parameters. Of these, the RC01 problem is known as industrial chemical processes. It is a case of heat exchanger network design. The RC11 is a two-reactor problem recognized as an industrial cooling system with 14 design variables and 15 inequalities design constraints. The names of other problems and their properties are presented in Table 13. RC01, the optimal shape of the heat exchanger structure, is considered in this problem. In three hot streams, a cold stream is heated to reduce the extensive area of the heat exchange structure. RC06 involves the separation of a three-component feed mix into two multi-component outputs using separators and splitting/mixing/combining. The

Table 13 Real-world engineering constrained problems

Problem	Category	Name	D	g	h	Optimum target
RC01	Industrial chemical process	Heat exchange network design (case 1)	9	0	8	1.8931162966E+02
RC06		Blending-pooling-separation problem	38	0	32	1.8638304088E+00
RC11	Process synthesis and design problems	Two-reactor problem	7	4	4	9.9238463653E+01
RC14		Multi-product batch plant	10	10	0	5.3638942722E+04
RC16	Mechanical engineering problem	Optimal design of industrial refrigeration system	14	15	0	3.2213000814E−02
RC23		Step-cone pulley problem	5	8	3	1.6069868725E+01
RC35	Power system problem	Optimal sizing of distributed generation for active power loss minimization	153	0	148	7.9963854000E−02
RC40		Microgrid power flow (Islanded case)	76	0	76	0.0000000000E+00
RC45	Power electronic problem	SOPWM for 3-level inverters	25	24	1	3.0739360000E−02
RC50		SOPWM for 13-level inverters	30	29	1	1.5051470000E−02

operating cost of each separator may depend on some separators and mixers in a linear or constraints fashion. Details of other problems with their mathematical models are given in [67].

According to obtained results, as shown in Table 14, the Ex-ChOA1 results take the lead in RC01, RC06, RC23, RC35, RC40, RC45, and RC50, whereas Ex-ChOA1 is better than others in RC11, RC16, RC40, and RC45. When we evaluate each function specifically, it is understood that the Ex-ChOA1 algorithm finds the best result in the RC01 function. The proposed algorithm found results with approximately 2 differences from the optimum target. In this function, the other case of the proposed algorithm took the second place. The I-GWO algorithm took the last place. In the RC06 evaluation, which is another industrial chemical-based problem, the proposed method is best in finding the optimum solution. The last rank belongs to standard ChOA algorithm, version 1. The RC11 and RC14 problems from the Process Synthesis and Design Problems family are taken as examples from CEC2020. According to the results, Ex-ChOA1 and Ex-ChOA were in first place of RC11 and I-GWO was in the last place. The optimum target value of this function is 9.9238463653E + 01. In the analysis of RC14, the I-GWO algorithm took the first place. Since Ex-ChOA2 found the same value in the "Mean" parameter, the partner took first place. The last place belongs to the E-ChOA algorithm. Besides, two famous problems from Mechanical Engineering published at CEC2020 were addressed. RC16 is one of them. In the result analysis of this function, Ex-ChOA1 and I-GWO algorithms were ranked first and last, respectively. In RC23 evaluation, Ex-ChOA1 algorithm showed the best performance and took the first place. In this function, the W-ChOA algorithm found the weakest result. In the analysis of the problem named RC35 in the field of Power Systems, Ex-ChOA1 and I-GWO were ranked first and last, respectively. In RC40, the best results were found jointly with Ex-ChOA1, Ex-ChOA2, and W-ChOA algorithms. The last row belongs to the ChOA2 algorithm by a small margin. RC45 and RC50 problems are discussed in the field of Power Electronics. Ex-ChOA1 and Ex-ChOA2 algorithms performed best in the RC45 problem. The ChOA1 algorithm took the last place. In the analysis of the RC50 problem, Ex-ChOA1 single-handedly allocated the first place to itself, and the last-ranked ChOA1 algorithm found the weakest result. As a result, Ex-ChOA1 ranked first in constrained modern and current real-world optimization problems with 68.3% success rate. Ex-ChOA2 secured the second position with 13.3%, followed by the W-ChOA algorithm in third place with 8.3%, and I-GWO in fourth place with 5%. The algorithms that follow these are E-ChOA, ChOA2 and ChOA1 algorithms which are ranked 5th, 6th and 7th, respectively.

5.2 Classic engineering problems

5.2.1 Welded beam design problem

One of the famous classical engineering optimization problems is welded beam design problem [68]. This problem is also defined in the real-world problems category with the code RC19. This shows that although the problem is not new, it is still a complex problem that has not been solved. The primary objective of this problem is designing a welded beam with the least production cost within certain constraints and variables. The restrictions are as follows. The shear stress (τ) is the end deflection of the beam (δ), the bending stress in the beam (Θ), and the buckling load (Pc) on the road. The variables of this problem are the length of the part attached to the rod (l), the thickness of the weld (h), the thickness of the rod (b), and the height of the rod (t). This problem is detailed in Fig. 10, and its mathematical models are

Table 14 Simulation result for each algorithm for real-world constrained problems

Functions		Ex-ChOA1	Ex-ChOA2	ChOA1	ChOA2	W-ChOA	E-ChOA	I-GWO
RC01	Mean	**1.91E + 02**	1.92E + 02	1.94E + 2	1.95E + 02	1.91E + 02	1.94E + 02	2.00E + 02
	Std	**1.01E−03**	1.03E−03	1.47E−03	1.46E−03	1.04E−03	1.96E−02	1.25E−03
RC06	Mean	**1.97E + 00**	1.99E + 00	2.07E + 00	2.04E + 00	2.00E + 00	2.03E + 00	2.01E + 00
	Std	**1.62E−02**	1.79E−02	2.26E−01	3.06E−01	1.90E−02	2.54E−02	1.96E−02
RC11	Mean	**9.89E + 01**	9.90E + 01	9.99E + 01	9.99E + 01	9.90E + 01	**9.89E + 01**	1.03E + 02
	Std	**7.23E + 00**	8.90E + 00	1.36E + 01	1.37E + 01	1.00E + 01	**8.96E + 00**	2.39E + 01
RC14	Mean	5.84E + 04	**5.83E + 04**	5.95E + 04	5.94E + 04	5.94E + 04	6.04E + 04	**5.83E + 04**
	Std	1.37E + 00	**1.00E + 00**	1.07E + 00	1.04E + 00	1.14E + 00	5.58E + 00	**7.88E + 00**
RC16	Mean	**3.19E−02**	3.20E−02	3.99E−02	3.89E−02	**3.19E−02**	3.49E−02	4.09E−02
	Std	**1.08E−02**	1.17E−02	1.53E−02	2.07E−02	**1.13E−02**	1.74E−02	3.03E−01
RC23	Mean	**2.00E + 01**	2.04E + 01	2.08E + 01	2.12E + 01	2.19E + 01	2.04E + 01	2.01E + 01
	Std	**4.44E−02**	5.83E−01	1.07E + 00	1.03E + 00	1.23E + 00	8.99E−01	9.99E−01
RC35	Mean	**8.57E−02**	8.59E−02	8.76E−02	8.82E−02	8.79E−02	8.69E−02	8.89E−02
	Std	**1.11E−01**	1.08E−01	6.03E−01	6.11E−01	6.13E−01	4.88E−01	9.63E−01
RC40	Mean	**0.00E + 00**	**0.00E + 00**	3.88E−15	4.95E−17	**0.00E + 00**	4.95E−11	0.00E + 00
	Std	**0.00E + 00**	**0.00E + 00**	6.48E−20	1.08E−22	**0.00E + 00**	1.48E−10	1.34E−27
RC45	Mean	**3.09E−01**	**3.09E−01**	4.07E−01	3.97E−01	3.10E−01	3.15E−01	3.80E−01
	Std	**1.00E−03**	**1.00E−03**	2.22E−03	1.53E−03	1.00E−03	1.09E−03	1.04E−03
RC50	Mean	**1.82E−02**	1.85E−02	2.67E−02	2.51E−02	2.37E−02	2.39E−02	2.36E−02
	Std	**1.05E−02**	4.20E−02	1.04E−01	8.99E−02	6.23E−02	6.23E−02	1.03E−02

The optimum results obtained from the algorithms are bold and highlighted

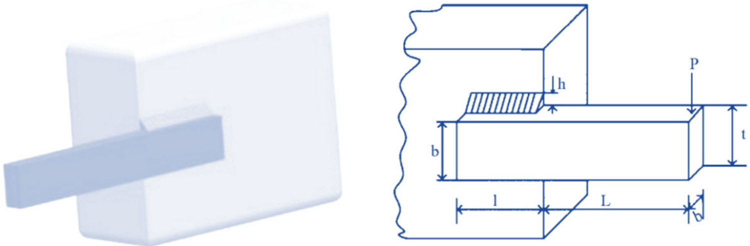

Fig. 10 The welded beam design problem

Table 15 Experimental results of the welded beam design problem

Algorithms	Optimum variables				Optimum cost
	h	l	t	b	
Ex-ChOA1	**0.2007**	**3.3476**	**9.0415**	**0.2058**	**1.7020**
Ex-ChOA2	0.2057	3.4705	9.0366	0.2057	1.7249
ChOA1	0.1656	4.0829	10.000	0.2046	1.9036
ChOA2	0.2214	3.5358	8.9115	0.2127	1.7737
W-ChOA	0.2021	3.4707	9.0366	0.2057	1.7249
E-ChOA	0.1680	3.8762	9.7041	0.2026	1.7453
I-GWO	0.1249	6.2775	9.0366	0.2057	1.7393

The best algorithm has been bold and highlighted

presented in Eq. 15.

$$\text{Consider}: \overrightarrow{x} = [x_1 x_2 x_3 x_4] = [hltb], \tag{15}$$

$$\text{Minimize}: f(\overrightarrow{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2),$$

$$\text{Subject to}: g_1(\overrightarrow{x}) = \tau(\overrightarrow{x}) - \tau_{\max} \le 0,\ g_2(\overrightarrow{x}) = \sigma(\overrightarrow{x}) - \sigma_{\max} \le 0,$$

$$g_3(\overrightarrow{x}) = \delta(\overrightarrow{x}) - \delta_{\max} \le 0,$$

$$g_4(\overrightarrow{x}) = x_1 - x_4 \le 0,\ g_5(\overrightarrow{x}) = p - p_c(\overrightarrow{x}) \le 0,\ g_6(\overrightarrow{x}) = 0.125 - x_1 \le 0,$$

$$g_7(\overrightarrow{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \le 0$$

Variables Range : $0.1 \le x_1 \le 2.00; 0.1 \le x_2 \le 10.0; 0.1 \le x_3 \le 10.0; 0.1 \le x_4 \le 2.00$

The obtained results in Table 15 demonstrate that Ex-ChOA1 is better than all other comparative algorithms. According to the results, Ex-ChOA2 and W-ChOA shared the second place jointly. In other ranks, I-GWO, E-ChOA, ChOA2, and ChOA2 algorithms took place.

5.2.2 Cantilever beam design problem

As the second classical engineering problem, the cantilever beam design problem is addressed. It is considered as a structural engineering design example regarding weight optimization of a square-section cantilever beam [69]. The beam is supported rigidly at the 1st node and faces a certain vertical force effect at the 6th node (Fig. 11). This beam consists

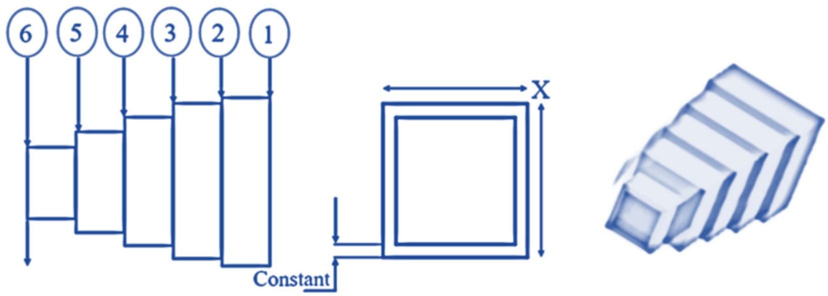

Fig. 11 The structure of cantilever beam

Table 16 Experimental results of the cantilever beam design problem

Algorithms	Optimum variables					Optimum cost
	x1	x2	x3	x4	x5	
Ex-ChOA1	**6.0164**	**5.3060**	**4.4935**	**3.5059**	**2.1516**	**1.339**
Ex-ChOA2	6.0143	5.3049	4.4987	3.5039	2.1517	1.340
ChOA1	6.2094	6.2094	6.2094	6.2094	6.2094	1.937
ChOA2	5.6052	4.9553	5.6619	3.1959	3.2026	1.411
W-ChOA	5.1261	5.6188	5.0952	3.9329	2.3219	1.379
E-ChOA	6.7628	5.1583	5.6537	2.9279	1.8854	1.397
I-GWO	6.0040	5.2950	4.4915	3.5125	2.1710	1.401

The best algorithm has been bold and highlighted

of five variables: the heights (or widths) of the different beam elements and the thickness is kept constant (here t = 2/3). The equation of this problem is presented in Eq. 16.

$$\text{Consider}: \overrightarrow{x} = [x_1 x_2 x_3 x_4 x_5] = [x_1 x_2 x_3 x_4 x_5], \tag{16}$$

$$\text{Minimize}: f(X) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5)$$
$$\text{Subject to}: G(X) = \frac{61}{x_1^3} + \frac{37}{x_1^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0$$

$\text{Variables Range}: 0.01 \le x_i \le 100; i \in 1, \ldots, 5$

Based on the results (are presented in Table 16), the ranking of the 7 algorithms is as follows. Ex-ChOA1 ranked first. The others were Ex-ChOA2, W-ChOA, E-ChOA, I-GWO, ChOA2, and ChOA1, respectively. Therefore, the performance of the proposed algorithm proves to be better than other algorithms used.

5.2.3 Microwave design optimization problem

Another engineering optimization problem discussed in this section is the transistor low-noise amplifier (LNA) design problem from the field of electronics and transistors [70]. This problem is an important type of problem in the field of microwave design. In this design, which is a multi-dimensional and multi-objective problem, is tried to reach the required objectives based on a nonlinear viable design target space. These can be achieved by optimal selection

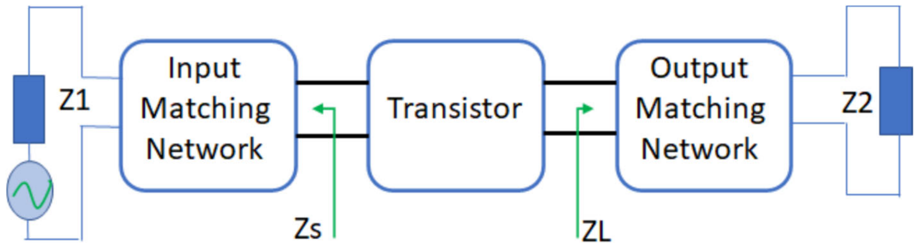

Fig. 12 The low-noise amplifier design problem

of source (Z_S) and Load (Z_L) terminations [71]. The main objective in designing a high performance low-noise amplifier is to achieve a design with maximum gain and minimum noise. In other words, the main purpose of LNAs that need very weak input signals is to keep the noise as low as possible and to strengthen the weak signals. The performance of LNAs is most often measured by their gain to noise ratio. Apart from these, there are also criteria such as dynamic range, loss of rotation, and stability. LNAs are one of the most important circuit components found in radio and other signal receivers. Low noise amplifiers are an essential part of a receiver circuit where the received signal is processed and converted into information. LNAs are designed to be close to the receiving device with minimal loss due to interference. As the name suggests, it adds a minimal amount of noise (useless data) to the received signal because it greatly distorts the now weak signals. An LNA is used when the signal-to-noise ratio (SNR) is high and needs to be reduced by about 50 percent and increased power. The LNA is the first component of the receiver to capture a signal, which is a vital part in the communication process. The two-dimensional representation of the problems is given in Fig. 12. Its mathematical equations are given in Eq. 17.

$$f(R_s, X_s, R_L, X_L) = a\left|F - F_{req}\right| + e^{\frac{G_T}{b}} + c\left|V_{out} - V_{outreq}\right| + d|V_{in}| \tag{17}$$

where (R_S, X_S) is associated with 'F' and (G_T, V_{in}, V_{out}) is associated with (R_S, X_S, R_L, X_L). In this problem, the aim is to find the optimal values for these four parameters. In this way, it will be possible to find the most suitable values for the optimal source (Z_S) and load (Z_L) terminations with the determined cost function as presented in Eq. 18.

$$\begin{aligned} F &\cong F(Z_S) = F_{\min} + \frac{R_n\left|Z_S - Z_{\text{opt}}\right|^2}{R_S\left|Z_{\text{opt}}\right|^2}; \\ G_T(Z_S, Z_L) &= \frac{4R_S R_L |Z_{21}|^2}{|(Z_{11} + Z_S)(Z_{22} + Z_L) - Z_{12}Z_{21}|^2}; \\ V_{\text{in}}(Z_S, Z_L) &= \frac{1 + |\rho_{\text{in}}|}{1 - |\rho_{\text{in}}|}; \\ V_{\text{out}}(Z_S, Z_L) &= \frac{1 + |\rho_{\text{out}}|}{1 - |\rho_{\text{out}}|} \end{aligned} \tag{18}$$

where 'F' depends on the signal and noise power, 'G_T' represents the relationship between the power delivered to the load and the available source power. The relationship between the reflected power and the input power at the input port is expressed by 'ρ_{in}', and the relationship between the reflected power at the load and the load power is expressed by 'ρ_{out}'.

According to the results presented in Table 17, it is understood that the proposed algorithm

Table 17 Experimental results of the low-noise amplifier design problem

Algorithms	Optimum variables				Optimum cost
	V_{in}	V_{out}	F	G_T	
Ex-ChOA1	**1.000**	**1.201**	**1.297**	**12.993**	**7.587**
Ex-ChOA2	1.021	1.208	1.280	12.979	7.589
ChOA1	1.166	1.200	1.248	12.884	10.553
ChOA2	1.053	1.230	1.284	12.945	8.484
W-ChOA	1.011	1.808	1.236	12.431	8.042
E-ChOA	1.017	1.201	1.311	12.986	8.189
I-GWO	1.000	1.202	1.300	12.994	8.174

The best algorithm has been bold and highlighted

Table 18 Experimental results of the pressure vessel design problem

Algorithms	Optimum variables				Optimum cost
	T_s	T_h	R	L	
Ex-ChOA1	**1.0000**	**1.0000**	**40.3196**	**200.0000**	**4348.8630**
Ex-ChOA2	1.0000	1.0000	40.3206	200.0000	4349.0192
ChOA1	1.0000	1.0000	40.3265	200.0000	4349.9979
ChOA2	1.0000	1.0000	40.3678	199.8468	4356.2796
W-ChOA	1.0000	1.0000	40.3219	200.0000	4349.2291
E-ChOA	1.0000	1.0000	40.3228	200.0000	4349.3874
I-GWO	1.0000	1.0000	42.0288	177.5018	4562.8664

The best algorithm has been bold and highlighted

outperforms other metaheuristic algorithms. The Ex-ChOA1 found the best solution than others. Ex-ChOA2 took second place. W-ChOA, I-GWO, E-ChOA, ChOA2 and ChOA1 algorithms are in the other rows, respectively (Table 18).

5.2.4 Pressure vessel design problem

Another classical engineering optimization problem addressed is the design of a pressure vessel. The goal is to minimize the overall cost associated with the cylindrical pressure vessel, accounting for material, welding, and forming expenses [65, 72]. The vessel features a hemispherical head and capped ends. The design variables include the head thickness (T_h), the length of the cylindrical body excluding the head (L), the shell thickness (T_s), and the internal radius (R), as illustrated in Fig. 13. Additionally, the problem includes four constraints expressed through mathematical equations (Eq. 19). The simulation results demonstrate that the proposed algorithm achieved superior performance compared to other algorithms in optimizing the total cost.

$$\text{Consider} : \overrightarrow{x} = [x_1 x_2 x_3 x_4] = [T_s T_h R L], \tag{19}$$

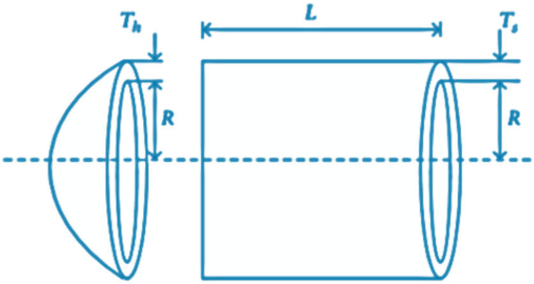

Fig. 13 Key design variables for the pressure vessel optimization problem

$$
\begin{aligned}
&\text{Minimize}: f(\overrightarrow{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3,\\
&\text{Subject to}: g_1(\overrightarrow{x}) = -x_1 + 0.0193x_3 \le 0,\\
&g_2(\overrightarrow{x}) = -x_2 + 0.00954x_3 \le 0,\\
&g_3(\overrightarrow{x}) = -\pi x_3^2x_4 - \tfrac{4}{3}\pi x_3^3 + 1296000 \le 0,\\
&g_4(\overrightarrow{x}) = x_4 - 240 \le 0,\\
&\text{Variables Range}: 0 \le x_1 \le 99,\\
&0 \le x_2 \le 99,\\
&10 \le x_3 \le 200,\\
&10 \le x_4 \le 200
\end{aligned}
$$

5.2.5 Tension spring design problem

Another classical engineering optimization problem addressed in this study is the design of a tension/compression spring. As illustrated in Fig. 14, the primary objective of this problem is to minimize the spring's weight [43, 65]. The design variables include the wire diameter (d), the mean coil diameter (D), and the number of active coils (N). The problem is subject to three constraints: surge frequency, minimum deflection, and shear stress. Also, mathematical formulations related to these constraints are provided in Eq. 20. Additionally, Table 19 summarizes the results obtained using each algorithm. The findings indicate that the SCSO algorithm outperforms the other metaheuristic methods in identifying the optimal cost.

$$\text{Consider}: \overrightarrow{x} = [x_1x_2x_3] = [dDN], \tag{20}$$

$$\text{Minimize}: f(\overrightarrow{x}) = (x_3 + 2)x_2x_1^2,$$

Fig. 14 Key variables for the tension spring optimization problem

Table 19 Experimental results of the tension spring design problem

Algorithms	Optimum variables			Optimum cost
	d	D	N	
Ex-ChOA1	**0.0500**	**0.2500**	**2.0000**	**0.0247**
Ex-ChOA2	**0.0500**	**0.2500**	**2.0000**	**0.0247**
ChOA1	0.0500	0.4998	15.0000	0.2096
ChOA2	0.0515	0.9926	8.6235	0.2761
W-ChOA	0.0500	0.3223	5.7424	0.0616
E-ChOA	0.0500	1.3000	2.4070	0.1414
I-GWO	0.0500	0.3174	14.0373	0.1272

The best algorithm has been bold and highlighted

Subject to : $g_1(\overrightarrow{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0,$

$g_2(\overrightarrow{x}) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0,$

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

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

Variables Range : $0.05 \leq x_1 \leq 2.00,$

$0.25 \leq x_2 \leq 1.30,$

$2.00 \leq x_3 \leq 15.0$

5.2.6 Rocket propulsion design problem

The rocket propulsion design problem involves the engineering design process aimed at optimizing the performance, efficiency, and reliability of a rocket engine [73]. This problem considers the physical and thermodynamic parameters of propulsion systems, thrust generation, fuel consumption, and environmental impacts. The primary goal is to develop a propulsion system that maximizes performance while minimizing costs, weight, and environmental impact. This involves achieving optimal thrust-to-weight ratios, efficient fuel utilization, and ensuring the system's reliability and safety under various operational conditions. The mathematical representation of this problem is shown in Eq. 21. Here, the constraints are the thrust (N), chamber pressure (M), and fuel flow rate (F). The unit of calculation for fuel flow rate is kilogram per second. The representative image describing the problem is presented in Fig. 15, and the results of the algorithms are presented in Table 20.

$$\text{Consider : } \overrightarrow{x} = [x_1 x_2 x_3] = [NFM], \tag{21}$$

Maximize : $F = \alpha_1 T + \alpha_2 Isp - \beta_1 m - \beta_2 C$

Subject to: $I_{sp} = \frac{T}{\dot{m} g_0},$

$$T = \dot{m} v_e + (p_e - p_a) A_e,$$

$$F_{\text{safety}} = \frac{\sigma_{\text{material}}}{\sigma_{\text{operating}}} \geq 1.5$$

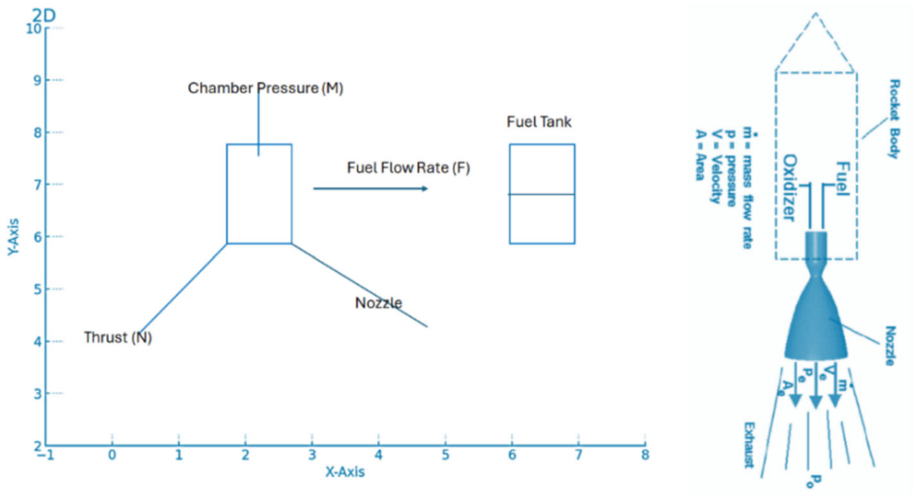

Fig. 15 Key variables for the rocket propulsion design problem

Table 20 Experimental results of the rocket propulsion design problem

Algorithms	Optimum variables			Optimum cost
	N	F	M	
Ex-ChOA1	**500.0000**	**50.0000**	**0.5000**	**10.0000**
Ex-ChOA2	494.8894	51.1035	0.5000	9.8978
ChOA1	404.5716	50.0000	0.5000	8.0914
ChOA2	439.4695	50.0000	0.5000	8.7894
W-ChOA	475.3224	50.0000	0.5000	9.5064
E-ChOA	500.0000	54.2327	0.5000	9.2195
I-GWO	448.8718	50.0000	0.9582	8.9774

The best algorithm has been bold and highlighted

$$m_{\text{propellant}} + m_{\text{structure}} \leq m_{\text{total}}, T_{\text{chamber}} \leq T_{\text{max,material}}, E_{\text{emissions}} \leq E_{\text{threshold}}$$

$$\begin{aligned} &\text{Variables Range}: 100 \leq x_1 \leq 500,\\ &50 \leq x_2 \leq 200,\\ &0.5 \leq x_3 \leq 5 \end{aligned}$$

5.2.7 Water distribution network optimization

This classical optimization problem aims to minimize costs, energy consumption or losses by optimizing the design parameters of the water distribution network [74, 75]. This problem usually involves optimizing pump sizes, pipe diameters and flow rates. In this problem, there are three defined pipe diameters, and the aim is to find the optimum pipe diameter value by finding optimum values for them. The mathematical definition of the problem is presented in Eq. 22.

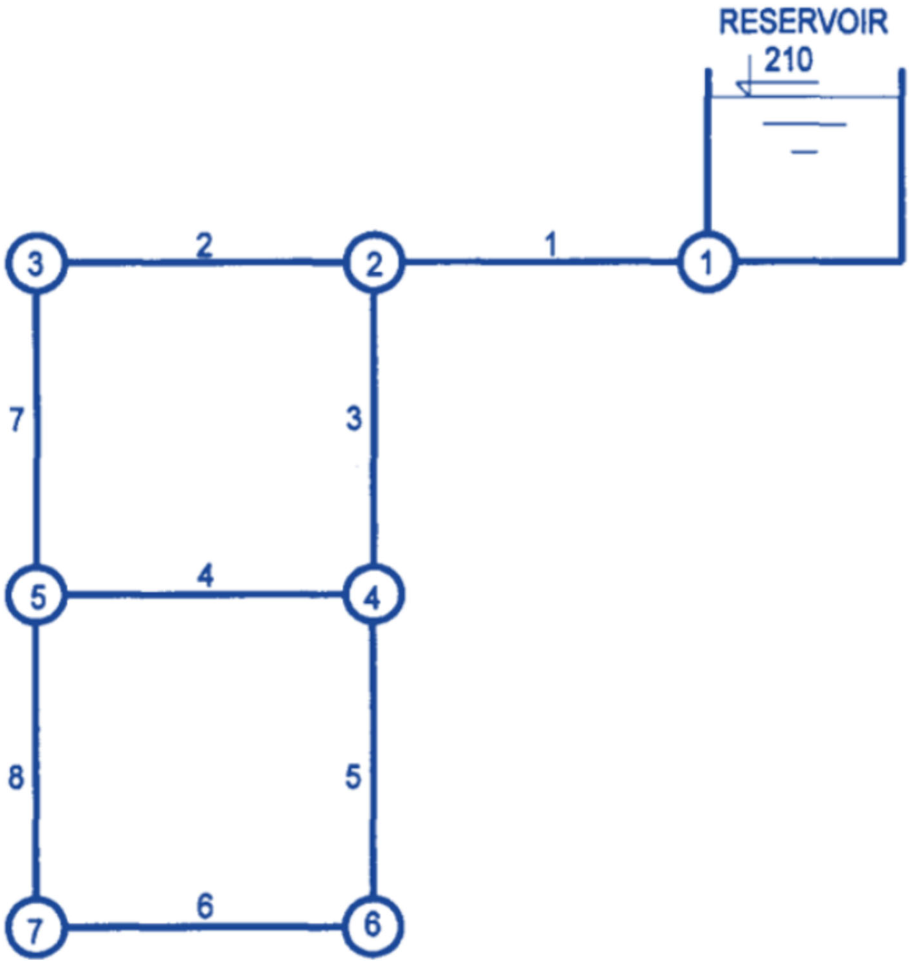

Fig. 16 A sample case about water distribution network problem

Consider

$$\overrightarrow{x} = [x_1 x_2 x_3] = [x_1 x_2 x_3], \tag{22}$$

$$\text{Minimize}: \ f(x) = \sum_{i=1}^{n} L_i.C.x_i$$

$$\text{Subjectto}: \sum \text{inflows} = \sum \text{outflows}$$

$$P(x) = \sum_{i=1}^{n} \frac{Q_i}{\pi.{x_i}^2} \leq P_{\max}$$

$$v_{\min} \leq \frac{4Q_i}{\pi.{x_i}^2} \leq v_{\max}$$

Variables Range : $0.1 \leq x_1 \leq 2$

Table 21 Experimental results of the water distribution network optimization problem

Algorithms	Optimum variables			Optimum cost
	X_1	X_2	X_3	
Ex-ChOA1	**0.5031**	**0.5031**	**0.5031**	**159,789.3647**
Ex-ChOA2	0.5030	0.5031	0.5031	159,789.3666
ChOA1	0.5026	0.5030	0.5035	159,789.4401
ChOA2	0.5026	0.5038	0.5023	159,789.6845
W-ChOA	0.5028	0.5032	0.5033	159,789.3929
E-ChOA	0.5029	0.5035	0.5031	159,789.4138
I-GWO	0.5021	0.5037	0.5014	159,789.4301

The best algorithm has been bold and highlighted

$0.1 \leq x_2 \leq 2$
$0.1 \leq x_3 \leq 2$
where 'L_i' is length of ith pipe (m), 'C' represents unit cost per meter per diameter (\$/m^2). 'x_i' symbolizes diameter of the ith pipe (m) and 'n' means number of pipes. The problem can be made more complex as the number of X's increases. Figure 16 presents a representative example of this problem. The performances of the algorithms are presented in Table 21.

6 Conclusion and future works

This study proposes an expanded variant, inspired by the chimpanzee's awesome hierarchy and hunting model, and discards some of the shortcomings of the recently proposed population-based metaheuristic algorithm (ChOA). The proposed algorithm, with its dynamic and flexible structure, tried to minimize problems such as slow convergence in the exploitation and exploration phases. At the same time, it tried to improve accuracy in high-dimensional problems. In this regard, it proposed hybrid models for position updates of search agents by providing a dynamic switching mechanism between phases. In this direction, some new mathematical models have been proposed. For this, two alternative solutions were suggested for position updating. One of them is the normal position update mechanism, and the other is based on the chaotic model. For this, two alternative solutions were suggested for position updating. One of them is the normal position update mechanism and the other is based on the chaotic model. Each of these approaches was given a half chance based on a normal distribution. In the solution mechanism based on the chaotic model, 12 different chaotic maps presented were used. On the other hand, exploration and exploitation phases were also taken into account in the updating according to the normal mechanism. Accordingly, harmonic mean and weight-coefficient averaging mathematical models proposed. With the effect of the control parameter presented in the algorithm, it will be possible to make balanced and fast movements, especially during the operation phase. In addition, the proposed algorithm was defined in two different scenarios (Ex-ChOA1 and Ex-ChOA2) and its performance was examined. The appropriate one can be selected and applied according to the needs of different problems, with this flexible approach. The performance of Ex-ChOA was analyzed on a total of 51 benchmarking functions and various engineering problems, and the results obtained accordingly were compared with some new well-known methods in the literature (ChOA1, ChOA2, W-ChOA, E- ChOA, I-GWO), and according to the analyses, the proposed algorithm achieved an acceptable level of success. In this regard, 34 benchmark functions, and 17 classical, constrained, and modern multi-objective engineering problems are used for performance analysis. Experiments were also conducted on a wide variety of well-known test functions to increase the reliability of the results as well as real-world problems. These functions were collected from CEC2015, 2019, and 2020 having different dimensions. According to the results analyzed in various ranges, it was observed that the proposed algorithm provides the best performance in the Ex-ChOA1 case scenario. It was noteworthy that Ex-ChoA2 took second place. This demonstrates that the developed method achieves strong overall performance.

As future studies, the suggested method can be used in a wide range of applications such as path planning, complex network analysis in bioinformatics, feature selection in various image processing problems, and optimization of hyperparameters in deep learning-based models. Although the use of various maps in the multi-strategies based ChOA algorithm is its strength, detecting the uncertain hyperparameters of each map is a research topic that will be focused on in another study. By tuning these parameters properly and optimized, the distribution of agents in the search space will become more efficient and therefore, the chance of finding the global solution will be possible in earlier iterations. Additionally, if different constraints are identified, they can be overcome with hybrid approaches such as reinforcement learning, other MH algorithms, or alternative strategies such as levy flight and Brownian.

Appendixes

See Tables 22, 23, 24.

Table 22 Explanations of used parameters in this study

Parameters	Explanation	Parameters	Explanation
$\overrightarrow{D}_a/\overrightarrow{D}_{Attacker}$	Distance between attacker chimp and prey	t	Current iteration
$\overrightarrow{D}_b/\overrightarrow{D}_{Barrier}$	Distance between barrier chimp and prey	T	Maximum number of iterations
$\overrightarrow{D}_c/\overrightarrow{D}_{Chaser}$	Distance between chaser chimp and prey	μ	Random number in [0,1]
$\overrightarrow{D}_d/\overrightarrow{D}_{Driver}$	Distance between driver chimp and prey	n	Total of agent numbers
$\overrightarrow{X}_a/\overrightarrow{X}_{Attacker}$	Attacker chimp's position	d	Total of dimensions
$\overrightarrow{X}_b/\overrightarrow{X}_{Barrier}$	Barrier chimp's position	r_1, r_2, r_3	Random number in [0,1]
$\overrightarrow{X}_c/\overrightarrow{X}_{Chaser}$	Chaser chimp's position	k	A natural number k = 1,…,m
$\overrightarrow{X}_d/\overrightarrow{X}_{Driver}$	Driver chimp's position	s	Parameter used in calculating f
$\overrightarrow{Xp}(t)$	Position vector of the prey	F	Cost of the function
f	Control parameter decreased from 2.5 to 0	m	Chaotic value
a	Coefficient value between $-2f$ and $2f$	C	Coefficient value between 0 and 2

Table 23 Benchmark test functions from CEC2015

Function details	Dim	Range	f_{min}	Type
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100,100]	0	Unimodal
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10,10]	0	Unimodal
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{i=j-1}^{i} x_j\right)^2$	30	[−100,100]	0	Unimodal
$f_4(x) = max_i\{\lvert x_i, 1 \le i \le n\}$	30	[−100,100]	0	Unimodal
$f_5(x) = \sum_{i=1}^{n-1} [100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2]$	30	[−30,30]	0	Unimodal
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[−100,100]	0	Unimodal
$f_7(x) = \sum_{i=1}^{n} ix_i^4 + random[0,1)$	30	[−1.28,1.28]	0	Unimodal
$f_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	[−500,500]	−418.9829*5	Multi-modal
$f_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10cos(2\pi x_i) + 10\right]$	30	[−5.12, 5.12]	0	Multi-modal
$f_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$	30	[−32,32]	0	Multi-modal
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{X_i}{\sqrt{i}}\right) + 1$	30	[−600,600]	0	Multi-modal
$f_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2\left[1 + 10sin^2(\pi y_{1+1})\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10,100,4)$	30	[−50,50]	0	Multi-modal
$y_i = 1 + \frac{x_i+1}{4}; u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < 1 \\ k(-x_i - a)^m & x_i < -a \end{cases}$				
$f_{13}(x) = 0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2\left[1 + sin^2(3\pi x_i + 1)\right] + (x_n - 1)^2\left[1 + sin^2(2\pi x_{ni})\right]\right\} + \sum_{i=1}^{n} u(x_i, 5,100,4)$	30	[−50,50]	0	Multi-modal
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}\right)^{-1}$	2	[−65,65]	1	Fixed dimension

Table 23 (continued)

Function details	Dim	Range	f_{min}	Type
$f_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$	4	[−5,5]	0.00030	Fixed dimension
$f_{16}(x) = 4x_1^2 + 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5,5]	−1.0316	Fixed dimension
$f_{17}(x) = \left(x^2 + \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{4\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cos x_1 + 10$	2	[−5,5]	0.398	Fixed dimension
$f_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2 \times \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right] \times \left[30 + (2x_1 - 3x_2)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$	2	[−2,2]	3	Fixed dimension
$f_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	3	[1, 3]	−3.86	Fixed dimension
$f_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	6	[0, 1]	−3.32	Fixed dimension
$f_{21}(x) = -\sum_{i=1}^{5}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	[0, 10]	− 10.1532	Fixed dimension
$f_{22}(x) = -\sum_{i=1}^{7}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	[0, 10]	− 10.4028	Fixed dimension
$f_{23}(x) = -\sum_{i=1}^{10}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	[0, 10]	− 10.5363	Fixed dimension
$f_{24}(x) = (\sum_{i=1}^{n}\lvert(x_i - m)\rvert)\exp(-\sum_{i=1}^{n}\sin((x_i - m)^2))$	50	[−10, 10]	0	Multi-modal

Table 24 Modern benchmark competitive test functions from CEC-2019 (CEC-C06)

Function	Benchmark function	Dim	Range	$f_{\min}$
CEC01	Stron's Chebyshev polynomial fitting problem	9	[−8192, 8192]	1
CEC02	Inverse Hilbert matrix problem	16	[−16384, 16384]	1
CEC03	Lennard–Jones minimum energy cluster	8	[−4, 4]	1
CEC04	Rastrigin's function	10	[−100, 100]	1
CEC05	Griewank's function	10	[−100, 100]	1
CEC06	Weierstrass function	10	[−100, 100]	1
CEC07	Modified Schwefel's function	10	[−100, 100]	1
CEC08	Expanded Schaffer's F6 function	10	[−100, 100]	1
CEC09	Happy cat function	10	[−100, 100]	1
CEC10	Ackley function	10	[−100, 100]	1

Author's contribution All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by FA, who is the sole author of the study. FA contributed to the methodology, validation, project administration, conceptualization, writing—review and editing, software, and formal analysis.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). Not applicable.

Data availability statement (DAS) The data that support the findings of this study are available on request from the corresponding author.

Declarations

Conflict of interest The authors declare no competing interests.

Ethics approval Not applicable.

Consent to publish All the authors agreed to publish the manuscript.

References

1. Talbi EG (2009) Metaheuristics: from design to implementation. John Wiley & Sons 74:5–39
2. Kiani F, Seyyedabbasi A, Aliyev R et al (2021) Adapted-RRT: novel hybrid method to solve three-dimensional path planning problem using sampling and metaheuristic-based algorithms. Neural Comput Appl 33:15569–15599
3. Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A (2019) A survey on new generation metaheuristic algorithms. Comput Ind Eng 137:106040

4. Wolpert DH, Macready WG et al (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
5. Saremi S, Mirjalili S, Lewis A (2014) Biogeography-based optimization with chaos. Neural Comput Appl 25(5):1077–1097
6. Lin Sh, Liu A, Wang A, Kong X (2023) An intelligence-based hybrid PSO-SA for mobile robot path planning in warehouse. J Comput Sci 67:1–11
7. Kiani F, Seyyedabbasi A, Nematzadeh S (2021) Improving the performance of hierarchical wireless sensor networks using the metaheuristic algorithms: efficient cluster head selection. Sens Rev 41:368–381
8. Kalinin KP, Berloff NG (2022) Computational complexity continuum within Ising formulation of NP problems. Nat Commun Phys 5(20):1–10
9. Salgotra R, Singh S, Singh U, Mirjalili S, Gandomi A (2023) Marine predator inspired naked mole-rat algorithm for global optimization. Expert Syst Appl 212:1–23
10. Anka F, Agaoglu N, Nematzadeh S, Torkamanian-afshar M, Gharehchopogh FS (2024) Advances in artificial rabbits optimization: a comprehensive review. Arch Comput Methods Eng, pp 1–36
11. Karaboga D, Basturk B (2007) Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. International fuzzy systems association world congress. Springer, Berlin, Heidelberg, pp 789–798
12. Kennedy J, Eberhart R (1995). Particle swarm optimization. IEEE, in Proceedings of ICNN'95-international conference on neural networks, 4, 1942–1948
13. Trojovský P, Dehghani M (2023) A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior. Sci Rep 13(1):8775
14. Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput Surv 35(3):268–308
15. Holland JH (1992) Genetic algorithms. Sci Am 267(1):66–73
16. Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
17. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713
18. Amali D, Dinakaran M (2019) Wildebeest herd optimization: a new global optimization algorithm inspired by wildebeest herding behaviour. J Intell Fuzzy Syst 37(6):8063–8076
19. Cai X, Zhao H, Shang S, Zhou Y, Deng W, Chen H, Deng W (2021) An improved quantum-inspired cooperative co-evolution algorithm with muli-strategy and its application. Expert Syst Appl 171:114629
20. Hayyolalam V, Kazem AAP (2020) Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems. Eng Appl Artif Intell 87:103249
21. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: a novel physics-based algorithm. Futur Gener Comput Syst 101:646–667
22. Khishe M, Mosavi MR, Moridi A (2018) Chaotic fractal walk trainer for sonar data set classification using multi-layer perceptron neural network and its hardware implementation. Appl Acoust 137:121–139
23. Mosavi MR, Khishe M et al (2019) Training multi-layer perceptron utilizing adaptive best-mass gravitational search algorithm to classify sonar dataset. Arch Acoust 44(1):137–151
24. Zhao W, Wang L, Zhang Z (2019) Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. Knowl Based Syst 163:283–304
25. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376. Article 113609
26. Goodarzimehr V, Shojaee S, Hamzehei-Javaran S, Talatahari S (2022) Special relativity search: a novel metaheuristic method based on special relativity physics. Knowl Based Syst 257:109484
27. El-kenawy EM, Abdelhamid A, Ibrahim A, Mirjalili S et al (2023) Al-biruni earth radius (ber) metaheuristic search optimization algorithm. Comput Syst Sci Eng 45(2):1917–1934
28. Moosavian N, Roodsari BK (2014) Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks. Swarm Evol Comput 17:14–24
29. Balochian S, Baloochian H (2019) Social mimic optimization algorithm and engineering applications. Expert Syst Appl 134:178–191
30. Moosavi SHS, Bardsiri VK (2019) Poor and rich optimization algorithm: a new human-based and multi populations algorithm. Eng Appl Artif Intell 86:165–181
31. Arasteh B, Sadegi R, Arasteh K, Gunes P, Kiani F, Torkamanian-Afshar M (2023) A bioinspired discrete heuristic algorithm to generate the effective structural model of a program source code. J King Saud Univ Comput Inf Sci 35(8):101655
32. Dehghani M, Trojovská E, Trojovský P (2022) A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process. Sci Rep 12(1):9924
33. Rao R, Savsani V, Vakharia D (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315

34. Arasteh B, Aghaei B, Farzad B, Arasteh K, Kiani F, Torkamanian-Afshar M (2024) Detecting SQL injection attacks by binary gray wolf optimizer and machine learning algorithms. Neural Comput Appl 36(12):6771–6792
35. Torkamanian-Afshar M, Lanjanian H et al (2020) RPINBASE: an online toolbox to extract features for predicting RNA-protein interactions. Genomics 112(3):2623–2632
36. Seyyedabbasi A, Dogan G, Kiani F (2020) HEEL: A new clustering method to improve wireless sensor network lifetime. IET Wireless Sens Syst 10(3):130–136
37. Kiani F, Nematzadehmiandoab S, Seyyedabbasi A (2019) Designing a dynamic protocol for real-time industrial internet of things-based applications by efficient management of system resources. Adv Mech Eng 11:1–20
38. Nematzadeh S, Torkamanian-Afshar M et al. (2022) Maximizing coverage and maintaining connectivity in WSN and decentralized IoT: an efficient metaheuristic-based method for environment-aware node deployment. Neural Comput Appl, pp 1–28
39. Kiani F (2016) A novel channel allocation method for time synchronization in wireless sensor networks. Int J Numer Model Electron Networks Devices Fields 29(5):805–816
40. Arasteh B, Gharehchopogh FS et al (2023) A novel metaheuristic based method for software mutation test using the discretized and modified forrest optimization algorithm. J Electron Test 39(3):347–370
41. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39
42. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
43. Seyyedabbasi A, Kiani F (2023) Sand Cat swarm optimization: a nature-inspired algorithm to solve global optimization problems. Engineering with Computers 39:2627–3265
44. Abdollahzadeh B, Gharehchopogh FS, Khodadadi N, Mirjalili S (2022) Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Adv Eng Softw 174:103282
45. Abdollahzadeh B, Khodadadi N, Barshandeh S, Mirjalili S (2024) Puma optimizer (PO): a novel metaheuristic optimization algorithm and its application in machine learning. Cluster Comput, pp 1–49
46. Chauhan D, Yadav A (2023) Binary artificial electric field algorithm. Evol Intel 16(4):1155–1183
47. Kiani F, Anka FA, Erenel F (2023) PSCSO: Enhanced sand cat swarm optimization inspired by the political system to solve complex problems. Adv Eng Softw 178:103423
48. Askari Q, Younas I, Saeed M (2020) Political optimizer: a novel socio-inspired meta-heuristic for global optimization. Knowl Based Syst 195:105709
49. Teimoori A, Mahdavi I et al (2022) A parallel hybrid PSO-GA algorithm for the flexible flow-shop scheduling with transportation. Comput Ind Eng 173:1–16
50. Mohammed H, Rashid T (2020) A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design. Neural Comput & Applic 32:14701–14718
51. Gharehchopogh FS, Abdollahzadeh B, Khodadadi N, Mirjalili S (2022) A hybrid African vulture optimization algorithm and harmony search: Algorithm and application in clustering. Advances in swarm intelligence: Variations and adaptations for optimization problems. Springer International Publishing, Cham, pp 241–254
52. Seyyedabbasi A, Aliyev R, Kiani F, Gulle MU, Basyildiz H, Shah MA (2021) Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems. Knowl-Based Syst 223:107044
53. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
54. Kiani F, Nematzadeh S, Anka FA, Findikli MA (2023) Chaotic sand cat swarm optimization. Mathematics 11(10):2340
55. Khishe M, Mosavi MR (2020) Chimp optimization algorithm. Expert Syst Appl 149:113338. https://doi.org/10.1016/j.eswa.2020.113338
56. Couzin ID, Laidre ME (2009) Fission-fusion populations. Curr Biol 19(15):633–635
57. Khishe M, Nezhadshahbodaghi M, Mosavi MR, Martín D (2021) A weighted chimp optimization algorithm. IEEE Access 9:158508–158539
58. Gharehchopogh FS, Maleki I, Dizaji ZA (2021) Chaotic vortex search algorithm: metaheuristic algorithm for feature selection. Evol Intell, pp 1–32
59. . Awad MZA, Reynolds RG (2015) A differential evolution algorithm with success-based parameter adaptation for CEC2015 learning-based optimization. In: 2015 IEEE congress on evolutionary computation (CEC), Sendai, Japan, pp 1098–1105. https://doi.org/10.1109/CEC.2015.7257012
60. Brest J, Maučec MS, Bošković B (2019) The 100-digit challenge: algorithm jDE100. In: 2019 IEEE congress on evolutionary computation (CEC), Wellington, New Zealand, pp 19–26. https://doi.org/10.1109/CEC.2019.8789904
61. Jamil M, Yang X (2013) A literature survey of benchmark functions for global optimization problems. Int J Math Modell Numer Optim 4(2):1–47

62. Kaur M, Kaur R, Singh N (2022) Enhanced chimp optimization algorithm for high level synthesis of digital filters. Sci Rep 12(1):21389
63. Seyyedabbasi A, Kiani F (2021) I-GWO and Ex-GWO: improved algorithms of the Grey Wolf Optimizer to solve global optimization problems. Eng Comput 37:509–532
64. Van den Bergh F, Engelbrecht AP (2006) A study of particle swarm optimization particle trajectories. Inf Sci 176(8):937–971
65. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41(2):113–127
66. Chauhan D, Yadav A (2023) Optimizing the parameters of hybrid active power filters through a comprehensive and dynamic multi-swarm gravitational search algorithm. Eng Appl Artif Intell 123:106469
67. Kumar A, Wu G, Ali MZ, Mallipeddi R, Suganthan PN, Das S (2020) A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. Swarm Evol Comput 56:100693. https://doi.org/10.1016/j.swevo.2020.100693
68. Üstüner B, Doğan E (2024) Structure optimization with metaheuristic algorithms and analysis by finite element method. KSCE J Civ Eng 28(1):328–341
69. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191
70. Gunes F, Uluslu A, Mahouti P (2020) Pareto optimal characterization of a microwave transistor. IEEE Access 8:47900–47913
71. Kiani F, Seyyedabbasi A, Mahouti P (2021) Optimal characterization of a microwave transistor using grey wolf algorithms. Analog Integr Circ Sig Process 109:599–609
72. Chattopadhyay S (2004) Pressure vessels: design and practice (1st ed) CRC Press. https://doi.org/10.1201/9780203492468
73. Seymour DC, Martin MA, Nguyen HH, Greene WD (2005) Transient mathematical modeling for liquid rocket engine systems: methods, capabilities, and experience. In: Fifth international symposium on liquid space propulsion
74. Khalifeh S, Akbarifard S, Khalifeh V, Zallaghi E (2020) Optimization of water distribution of network systems using the Harris Hawks optimization algorithm (Case study: Homashahr city). MethodsX 7:100948
75. Kim DH, Jang DW (2024) Optimization of water distribution network demand patterns using real-coded genetic algorithms. Water 16(20):2971

Ferzat Anka received his undergraduate and graduate degrees in computer software engineering in 2006 and 2009, respectively. Later on, he received a Ph.D. degree in computer engineering at Dokuz Eylul University, in 2014. He has domestic and international academic and sectoral experiences, and he has founding and leadership experience in research laboratories such as WSN, IoT and ML. He also served as department chair in the computer and software engineering departments. His research interests include wireless sensor networks, smart cities, IoT, bioinformatics, reinforcement learning, distributed systems and optimization algorithms. Dr.Anka has undertaken various roles in national and international projects and has publications in many indexed journals and three books. He is currently working as an associate professor at Fatih Sultan Mehmet Vakif University's computer engineering department.