Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2024.054334

ARTICLE

An Improved Artificial Rabbits Optimization Algorithm with Chaotic Local Search and Opposition-Based Learning for Engineering Problems and Its Applications in Breast Cancer Problem

Feyza Altunbey Özbay[1], Erdal Özbay[2] and Farhad Soleimanian Gharehchopogh[3,*]

[1]Department of Software Engineering, Firat University, Elazig, 23119, Turkey

[2]Department of Computer Engineering, Firat University, Elazig, 23119, Turkey

[3]Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, 44867-57159, Iran

*Corresponding Author: Farhad Soleimanian Gharehchopogh. Email: bonab.farhad@gmail.com

Received: 25 May 2024 Accepted: 22 August 2024 Published: 27 September 2024

ABSTRACT

Artificial rabbits optimization (ARO) is a recently proposed biology-based optimization algorithm inspired by the detour foraging and random hiding behavior of rabbits in nature. However, for solving optimization problems, the ARO algorithm shows slow convergence speed and can fall into local minima. To overcome these drawbacks, this paper proposes chaotic opposition-based learning ARO (COARO), an improved version of the ARO algorithm that incorporates opposition-based learning (OBL) and chaotic local search (CLS) techniques. By adding OBL to ARO, the convergence speed of the algorithm increases and it explores the search space better. Chaotic maps in CLS provide rapid convergence by scanning the search space efficiently, since their ergodicity and non-repetitive properties. The proposed COARO algorithm has been tested using thirty-three distinct benchmark functions. The outcomes have been compared with the most recent optimization algorithms. Additionally, the COARO algorithm's problem-solving capabilities have been evaluated using six different engineering design problems and compared with various other algorithms. This study also introduces a binary variant of the continuous COARO algorithm, named BCOARO. The performance of BCOARO was evaluated on the breast cancer dataset. The effectiveness of BCOARO has been compared with different feature selection algorithms. The proposed BCOARO outperforms alternative algorithms, according to the findings obtained for real applications in terms of accuracy performance, and fitness value. Extensive experiments show that the COARO and BCOARO algorithms achieve promising results compared to other metaheuristic algorithms.

KEYWORDS

Artificial rabbit optimization; binary optimization; breast cancer; chaotic local search; engineering design problem; opposition-based learning

1 Introduction

In today's world, real-world optimization problems have become more complex. Their difficulty has increased due to developments in various application fields such as engineering, medicine, computer science, and manufacturing design. These problems include complexities such as multi-objective, non-linear, multi-dimensional, multi-disciplinary, and non-convex regions [1]. Practical and dependable optimization algorithms can be developed to solve these problems. Classical optimization methods based on gradient information, including calculating first and second derivatives, cannot provide satisfactory results reasonably for solving such issues [2]. These methods require a large number of complex mathematical calculations. In these methods, issues with convergence speed and getting stuck at the local optimum point may be encountered. Complex implementation, mathematical calculations, difficult convergence for discrete optimization problems, etc., are some disadvantages of classical optimization methods [3].

In recent years, researchers have taken great interest in metaheuristic approaches to eliminate the shortcomings of classical optimization methods and solve complex optimization problems with high efficiency and accuracy. Metaheuristic algorithms aim to find the optimal or approximate solution to complex optimization problems under limited conditions by using search methods inspired by different natural methods. These search strategies help find nearly optimal solutions by effectively searching the search space. Metaheuristic algorithms are not problem-specific and are flexible to solve various optimization problems [4]. They are stochastic algorithms that start the search process with random solutions. They have low computational complexity. Additionally, these algorithms can escape local optima due to randomness-based search strategies. In recent years, metaheuristic algorithms have been successfully used to solve various optimization problems in many research areas, such as image segmentation [5], robotic and path planning [6], medical application [7], sensor networks [8], water resources management [9], thin-walled structures [10], Internet of Things (IoT) [11], bioinformatics [12], and engineering problems [13]. Metaheuristic algorithms can be examined in different categories: physics, social, music, swarm, chemistry, biology, sports, mathematics, and hybrid-based.

Artificial Rabbits Optimization (ARO) is a new metaheuristic optimization method presented in 2022 [14]. ARO mimics rabbits' natural foraging and hiding behaviors. The problem of falling into local optima and premature convergence are the main drawbacks of ARO. These limitations restrict the exploration of the search space and prevent finding the global optimum. The random initialization of the rabbits, the initial population in ARO, and the imbalance between exploration and exploitation also affect the algorithm's performance. Chaotic opposition-based learning ARO (COARO), an improved version of ARO, has been proposed to overcome these limitations and improve the effectiveness of ARO. The COARO algorithm is proposed by combining the strengths of chaotic local search (CLS) and opposition-based learning (OBL) and incorporating ten different chaotic maps into the optimization process. CLS ensures that solution quality increases by directing the local search around the global best solution. Adding OBL to the algorithm improves the initial population and enhances the problem of getting stuck in local minima.

The effectiveness of the proposed algorithm was evaluated using different benchmarks. The COARO algorithm's performance is compared to that of ARO, Grey Wolf Optimization (GWO), Multi-Verse Optimizer (MVO), Particle Swarm Optimization (PSO), and Transient Search Optimization (TSO). Performance comparison of the COARO algorithm was supported by Wilcoxon's signed rank (WSR) and Friedman tests. Box plot comparison has been employed to check the proposed COARO algorithm's consistency. The COARO algorithm's capacity for problem-solving has been tested on six different engineering design problems, including pressure vessel, rolling element bearing design, cantilever beam, speed reducer, welded beam, and tension/compression spring. Simulation results indicate that the COARO algorithm outperforms ARO in most cases. It has also been observed that COARO achieves promising results compared to recent optimization algorithms. Furthermore, the paper proposes the binary version of the COARO algorithm (BCOARO) to solve the issue of feature selection in classification tasks. The V-shaped transfer function is integrated into the algorithm to convert the continuous COARO algorithm into a binary version. The BCOARO algorithm's performance is compared using different feature selection methods.

The significant contributions of the paper are:

- This paper introduces COARO, which is proposed to eliminate the shortcomings of ARO and improve its performance.
- COARO algorithm is proposed by adding CLS with OBL to the original ARO and incorporating ten different chaotic maps into the optimization process. CLS has improved the performance and convergence speed of the proposed algorithm by using chaotic maps. Additionally, with the use of OBL, population diversity has increased, and the problem of getting stuck in local minima has improved.
- The effectiveness of the proposed algorithm was evaluated using different benchmarks. The COARO algorithm's performance is compared to ARO, GWO, MVO, PSO, and TSO. Performance comparison of the COARO algorithm was supported by Wilcoxon's signed rank (WSR) and Friedman tests. Box plot comparison has been employed to check the proposed COARO algorithm's consistency.
- The COARO algorithm's capacity for problem-solving has been tested on six engineering design problems, including pressure vessel, rolling element bearing design, cantilever beam, speed reducer, welded beam, and tension/compression spring.
- This paper also proposes a binary COARO algorithm (BCOARO) version. It has been tested on the breast cancer dataset, and the results have been compared with different feature selection methods.

The structure of the paper is as follows: The ARO algorithm's mathematical model is described in Section 2. In Section 3, the proposed COARO algorithm is explained. CLS, OBL, and an overview of COARO and its complexity are detailed. Section 4 evaluates the proposed COARO algorithm using unimodal, multimodal, fixed-dimension multimodal, and CEC2019 functions. This section also compares the COARO algorithm's performance to various metaheuristic algorithms in the literature. In the same section, performance assessments of WSR and Friedman statistical tests are made. The results of COARO algorithms on six different engineering problems are examined in this section. Section 5 concludes with a discussion of future research and conclusions.

2 The Mathematical Model of ARO Algorithm

ARO algorithm was developed with a mathematical model inspired by the survival strategies of rabbits in nature [15]. Rabbits' survival strategies are based on exploration and exploitation. The first behavior, the exploration strategy, is described as detour foraging [16]. Rabbits aim to minimize the chance of being caught by digging many burrows to protect themselves from predators and mislead them. In this respect, the second behavior, the exploitation strategy, is described as random hiding. Depending on their energy state, rabbits must adaptively switch between circuitous foraging and random hiding strategies. In this respect, the third behavior, the switch from exploration to exploitation strategy, is described as energy shrink [17].

2.1 Detour Foraging (Exploration)

Rabbits prefer far places rather than near nests when searching for food. The location of each search individual tends to update towards the other search individual picked at random in the swarm.

This behavior is a clear indication of the detour foraging behavior of the ARO [18]. The following is the mathematical model of rabbits' detour foraging:

$$\overrightarrow{v}_i(t+1) = \overrightarrow{x}_j(t) + R \cdot \left(\overrightarrow{x}_i(t) - \overrightarrow{x}_j(t)\right) + round(0.5 \cdot (0.05 + r_1)) \cdot n_1 \tag{1}$$

$$R = L \cdot \mathrm{c} \tag{2}$$

$$L = \left(e - e^{\left(\frac{t-1}{T}\right)^2}\right) \cdot \sin(2\pi r_2) \tag{3}$$

$$c(k) = \begin{cases} 1 & if\ k == g(l) \\ 0 & else \end{cases} \tag{4}$$

$$g = randperm(d) \tag{5}$$

$$n_1 \sim N(0,1) \tag{6}$$

here, $i, j = 1, \ldots, n$ and $j \neq i$. $k = 1, \ldots, d$ and $l = 1, \ldots, \lceil r_3 \cdot d \rceil$. The ith rabbit's potential location at a time (t+1) is $\overrightarrow{v}_i(t+1)$. $\overrightarrow{x}_i(t)$ represents the ith rabbit's location at (t). A population of rabbits has a size of n. The problem's dimension is d. The max iteration number is T. $\lceil \cdot \rceil$ indicates the ceiling function. Rounding to the closest integer is indicated by a round. A random permutation of numbers from 1 to d is returned by randperm. r_1, r_2 and r_3 are defined as the random three numbers in (0, 1). L is the running distance that reflects the rate of movement when engaging in detour foraging. The conventional normal distribution applies to n_1. The L can provide a longer step during the initial iterations, according to Eq. (3). The mapping vector c is used to help the algorithm randomly select a set of search individual components to alter the foraging behavior. R defines the running operator [14].

2.2 Random Hiding (Exploitation)

Rabbits dig tunnels around their caves to protect themselves from predators. According to the ARO algorithm, a rabbit always digs d tunnels around it in each dimension of the search space, and it always picks one of the caves at random to hide in to lessen the likelihood of being attacked [18]. The following is defined of jth nest of the ith rabbit:

$$\overrightarrow{b}_{i,j}(t) = \overrightarrow{x}_i(t) + H \cdot g \cdot \overrightarrow{x}_i(t) \tag{7}$$

$$H = \frac{T - t + 1}{T} \cdot r_4 \tag{8}$$

$$n_2 \sim N(0,1) \tag{9}$$

$$g(k) = \begin{cases} 1 & if\ k == j \\ 0 & else \end{cases} \tag{10}$$

here, $i = 1, \ldots, n$, $j = 1, \ldots, d$, and $k = 1, \ldots, d$. According to Eq. (7), a rabbit creates burrows in its immediate neighborhood in all directions. H defines the hiding parameter and iteratively decreases linearly from 1 to $1/T$ with a random perturbation [19]. The random selection of one of the rabbit's burrows for shelter is rejected to avoid being caught by predators [14]. The following is described as the mathematical model for this random hiding strategy:

$$\overrightarrow{v}_i(t+1) = \overrightarrow{x}_i(t) + R \cdot \left(r_4 \cdot \overrightarrow{b}_{i,r}(t) - \overrightarrow{x}_i(t)\right) \tag{11}$$

$$g_r(k) = \begin{cases} 1 & \text{if } k == \lfloor r_5 \cdot d \rfloor \\ 0 & \text{else} \end{cases} \tag{12}$$

$$\overrightarrow{b}_{i,r}(t) = \overrightarrow{x}_i(t) + H \cdot g_r \cdot \overrightarrow{x}_i(t) \tag{13}$$

here, $i = 1, \ldots, n$ and $k = 1, \ldots, d$ defines a randomly chosen burrow to hide from its d burrows. r_4 and r_5 define two random numbers in (0, 1). Using Eq. (11), the ith search individual updates its location toward one of the caves from its d burrows that were randomly chosen. Following the successful completion of either detour foraging or random hiding, the ith rabbit's position is updated in Eq. (14):

$$parent\overrightarrow{x}_i(t+1) = \begin{cases} \overrightarrow{x}_i(t) & f\left(\overrightarrow{x}_i(t)\right) \le f\left(\overrightarrow{v}_i(t+1)\right) \\ \overrightarrow{v}_i(t+1) & f\left(\overrightarrow{x}_i(t)\right) > f\left(\overrightarrow{v}_i(t+1)\right) \end{cases} \tag{14}$$

When the position of ith rabbit is checked according to Eq. (14), when the candidate position is better than the current position, the rabbit will leave its current position and move to the candidate positions determined by Eqs. (1) and (11).

2.3 Energy Decline (from Exploration to Exploitation)

Modeling the transition from the exploration process associated with detour foraging to the exploitation stage associated with random hiding includes an energy component. The energy factor designed in ARO is given in Eq. (15).

$$A(t) = 4\ln\frac{1}{r}\left(1 - \frac{t}{T}\right) \tag{15}$$

here, r represents a random number in (0, 1). $A(t)$, the energy factor, tends to decrease towards zero as the number of iterations increases. The ARO pseudo-code is provided in Algorithm 1.

Algorithm 1: The pseudo-code of the ARO algorithm

Input: Maximum iteration number, population size
Output: X_{best} best solutions and F_{best} its fitness value
Initialize the first population of rabbits randomly (X_i)
Evaluate the fitness value of each rabbit (Fit_i)
X_{best} = the best solution found so far
  while (termination condition is not met)
    for each individual (X_i) **do**
    Calculate the energy factor A using Eq. (15)
      if A > 1
        Select a rabbit randomly from other individuals
        Calculate R using Eqs. (2)–(6)
        Perform detour foraging using Eq. (1)

(Continued)

Algorithm 1 (continued)

Calculate the fitness Fit_i
Update the position of the current individual using Eq. (14)
else
Generate d burrows and randomly pick one as hiding using Eq. (13)
Perform random hiding using Eq. (11)
Calculate the fitness Fit_i
Update the position of the individual using Eq. (14)
end if
Update the best solution found so far X_{best}
end for
end while
Return X_{best}

3 The Proposed COARO Algorithm

In this section, the proposed COARO algorithm is presented, which combines CLS and OBL techniques to improve the performance of the ARO algorithm and overcome its difficulties. The section includes general information about CLS and OBL techniques, the motivation for integrating them into ARO, and the general structure of the COARO algorithm.

3.1 Chaotic Local Search (CLS)

The term "chaos" describes the highly unexpected behavior of a complex system [20]. Mathematically, chaos is deterministic randomness found in a nonlinear, dynamic, and non-converging [21]. Due to this definition, it can be assumed that the source of randomness is chaotic systems. An optimization algorithm's search strategy is often implemented within the search space based on random values. Chaos maps use a function to relate or match the chaos behavior in the optimization method based on a parameter. This way, optimization algorithms based on chaotic maps can scan the search space more dynamically and generally. Chaotic maps, which are used as an alternative to random number generators in the search space, often obtain better results than them. Although chaos has an unexpected behavior structure, it also has harmony within. Local optima can be avoided when randomness is adjusted by utilizing chaotic maps in optimization algorithms. A chaotic local search-based search strategy is introduced to enhance ARO's ability to obtain optimal global solutions. In this study, the chaotic maps given in Table 1 are used.

3.2 Opposition-Based Learning (OBL)

OBL proposed by Tizoosh is used to improve the search capability of algorithms and increase the convergence speed [22]. Classical metaheuristic algorithms start the search process with a randomly generated population containing solutions to the problem to be tested. In this case, the algorithm's convergence rate may be reduced and the calculation time may increase. To overcome this problem, the OBL strategy considering inverse solutions is introduced. OBL calculates an inverse solution $\overline{X}_i$ for each solution X in the range lb (lower bound) and ub (upper bound) by Eq. (16).

$$\overline{X}_i = lb_i + ub_i - X_i \tag{16}$$

Table 1: Chaotic maps used in the COARO algorithm

Algorithm	Chaotic map	Mathematical expression	Range
COARO1	Chebyshev	$x_{n+1} = \cos\left(kcos^{-1}x_n\right)$	$(-1, 1)$
COARO2	Circle	$x_{n+1} = x_n + b - \left(\frac{a}{2\pi}\right) sin(2\pi x_n)mod(1)$; $a = 0.5$ and $b = 0.2$	$(0, 1)$
COARO3	Gauss/Mouse	$x_{n+1} = \left\{\begin{matrix} 0, & x_n = 0 \\ \frac{1}{x_n mod\,(1)}, & x_n \in (0,1) \end{matrix}\right\}, \quad \frac{1}{x_n mod\,(1)} = \frac{1}{x_n} - \left\lfloor \frac{1}{x_n} \right\rfloor.$	$(0, 1)$
COARO4	Iterative	$x_{n+1} = sin\left(\frac{a\pi}{x_n}\right)$ $a = 0.7$	$(-1, 1)$
COARO5	Logistic	$x_{n+1} = ax_n\,(1 - x_n)$ $a = 0.4$	$(0, 1)$
COARO6	Piecewise	$x_{n+1} = \left\{\begin{matrix} \frac{x_n}{P}, & 0 \le x_n < P \\ \frac{x_n - P}{0.5 - P}, & P \le x_n < 0.5 \\ \frac{1 - P - x_n}{0.5 - P}, & 0.5 \le x_n < 1 - P \\ \frac{1 - x_n}{P}, & 1 - P \le x_n < 1 \end{matrix}\right\}; P = 0.4$	$(0, 1)$
COARO7	Sine	$x_{n+1} = \frac{a}{4}\sin(\pi x_n), 0 < a \le 4$	$(0, 1)$
COARO8	Singer	$x_{n+1} = \mu\left(7.86x_n - 23.31x_n^2 + 28.75x_n^3 - 13.302875x_n^4\right), \mu = 1.07$	$(0, 1)$
COARO9	Sinusoidal	$x_{n+1} = ax_n^2 sin(\pi x_n)$, $a = 2.3$ and $x_0 = 0.7$	$(0, 1)$
COARO10	Tent	$x_{n+1} = \left\{\begin{matrix} \frac{x_n}{0.7}, & x_n < 0.7 \\ \frac{10}{3x_n(1 - x_n)}, & otherwise \end{matrix}\right\}$	$(0, 1)$

3.3 The Overview of Proposed COARO

As with all metaheuristic algorithms, ARO suffers from inefficient search, premature convergence, and local optima problems. To overcome these problems, the COARO algorithm was proposed by adding CLS and OBL techniques to the ARO algorithm. CLS is integrated into the ARO algorithm with the advantages of nonlinear dynamics and advanced exploration. In this way, ARO's performance and convergence speed has increased with the proposed COARO algorithm. In addition, COARO's convergence speed and performance has been increased by OBL's ability to increase diversity and bring it closer to the global optimal solution.

3.3.1 Initial Stage of COARO

The algorithm's performance is not stable since a randomly generated initial population is used in the ARO algorithm. The initial population can be generated using chaotic maps, considering its ergodicity and unpredictability characteristics. Therefore, in the COARO algorithm, the OBL and CLS strategies are combined to create a more reliable initial population when initializing the rabbit population. Eq. (17) refers to generating rabbit population X with chaotic maps.

$$x_i = lb_{ij} + ch_{ij} \times (ub_{ij} - lb_{ij}) \tag{17}$$

The ch_{ij} value is the chaotic map value calculated using the equations in Table 1. $X_i \in X$ ($i = 1, 2, \ldots N; j = 1, 2, \ldots D$), lb and ub represent rabbits X's lower and upper bounds, respectively. Then, opposition-based learning is incorporated into this process to increase the effect of chaos on the initial population.

3.3.2 Updating Stage of COARO

Local optimum traps, inefficient search, and early convergence are some of the issues that optimization algorithms may encounter. Chaotic maps are employed to increase the success of global optimal searching, accelerate the search, and avoid being mired in the local optimum. As mentioned in the ARO algorithm, predators frequently pursue and attack rabbits. As a result, for rabbits to survive, they need to locate a secure hiding area. The rabbits' random selection of burrow $\overrightarrow{b}_{i,r}$ from its d burrows is expressed by Eq. (18) in the COARO algorithm and is called the random hiding strategy. The random variable in this equation has been used to determine the rabbits' potential positions after selecting the burrow with a random hiding strategy.

$$\overrightarrow{v}_i(t+1) = \overrightarrow{x}_i(t) + R \cdot \left(\mathrm{CM} \cdot \overrightarrow{b}_{i,r}(t) - \overrightarrow{x}_i(t)\right) \tag{18}$$

In the Eq. (18), CM refers to the value produced by the chaotic maps in Table 1. Furthermore, CLS and OBL approaches are also used to optimize the positions of the rabbits in this stage of COARO. If the rabbit's position remains the same or changes slightly, the algorithm will fall into the local optimum. The CLS in Eq. (17) was used to eliminate this issue. As a result of the algorithm finding a better location, the chaotic local search is terminated and OBL is applied. The COARO pseudo-code is provided in Algorithm 2.

Algorithm 2: The pseudo-code of the COARO algorithm

Input: Maximum iteration number T and population size N
Output: X_{best} best solutions and F_{best} its fitness value
Initialize the first population of rabbits randomly (X_i)
Generate a new population of rabbits (X_C) using Eq. (17)
Compute the opposite point X_{OC} of X_C using Eq. (16)
Evaluate the fitness value (Fit_i), (Fit_{OC}) of X_i and X_{OC}
if (Fit_{OC}) $\leq$ (Fit_i)
$X_i = X_{OC}$
end if
Evaluate the fitness value of X_i
X_{best} = the best solution found so far
    while (termination condition is not met)

(Continued)

Algorithm 2 (continued)

for each individual (X_i) **do**
Calculate the energy factor A using Eq. (15)
if A > 1
Select a rabbit randomly from other individuals
Calculate R using Eqs. (2)–(6)
Perform detour foraging using Eq. (1)
else
Generate d burrows and randomly pick one as hiding using Eq. (13)
Perform random hiding using Eq. (18)
end if
end for
Calculate the fitness Fit_i
Update the position of the current individual using Eq. (14)
Update the best solution found so far X_{best}
Generating new solution X_{new} by using chaotic maps and opposition-based learning
Calculate the fitness value (Fit_{best}), (Fit_{new}) of X_{best} and X_{new}
if (Fit_{new}) ≤ (Fit_{best})
$X_{best} = X_{new}$
end if
end while
Return X_{best} and its fitness value (Fit_{best})

3.4 Computational Complexity of COARO

In this section, the complexity of the proposed COARO algorithm is examined. The complexity of the COARO adheres to initializing the search agents and evaluating the fitness function. The search agents are updated based on the value of the fitness function, chaotic local search, and opposition-based learning strategies. The initialization stage of the search agent has an O($n*d$) complexity, where n is the population number and d is the number of dimensions. The complexity cost for evaluating the fitness functions of all search agents is O($n*T$); here, T represents the max iteration numbers. The complexity cost of updating the search agents based on the fitness function value is O($n*T*d$). The complexity cost of CLS and OBL strategies is O($n*T$). Consequently, the overall complexity cost of COARO is calculated as O($n*T*d$).

4 Experimental Results

The experimental test results and evaluations that were utilized to gauge how well the suggested COARO algorithm performed are included in this section. Six different engineering design problems and 33 benchmark functions—including unimodal, multimodal, fixed-dimension multimodal, and CEC2019 functions—are used to assess it. The performance of the COARO algorithm is evaluated by comparing the results obtained with COARO algorithms for 23 classical and 10 CEC2019 benchmark functions with ARO and the well-known GWO, MVO, PSO, and TSO algorithms. Additionally, the evaluation of performance is supported by WSR and Friedman tests. The experimental results obtained in this paper were carried out in the MATLAB R2021b environment, and the test results were taken on a machine with Core i7 4.7 GHz CPU, 16 GB memory, and GeForce GTX4060 GPU.

4.1 The Performance Comparison for Classical and CEC2019 Benchmark Functions

This section uses 33 benchmark functions to assess the efficacy of the proposed COARO. F1–F7 are unimodal functions used to measure the local exploitation capacity of the algorithm. F8–F13 functions are used to evaluate the algorithm's exploration ability. F14–F23 functions assess the algorithm's ability to explore fixed-dimension optimization problems. Mathematical expressions for these functions are shown in Tables 2–4.

Table 2: Unimodal benchmark functions

(F_i) Function	Range	Dim	f_{min}
$F1 = \sum_{i=1}^{n} x_i^2$	[−100, 100]	30	0
$F2 = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[−10, 10]	30	0
$F3 = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	[−100, 100]	30	0
$F4 = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	[–100, 100]	30	0
$F5 = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	[−30, 30]	30	0
$F6 = \sum_{i=1}^{n} ([x_i + 0.5])^2$	[−100, 100]	30	0
$F7 = \sum_{i=1}^{n} ix^4 + random[0, 1]$	[−1.28, 1.28]	30	0

Table 3: Multimodal benchmark functions

(F_i) Function	Range	Dim	f_{min}
$F8 = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	[−500, 500]	30	428.989 × n
$F9 = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	[−5.12, 5.12]	30	0
$F10 = -20exp\left(-0.2\left(\frac{1}{n}\sum_{i=1}^{n} x_i^2\right)^{0.5}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$	[−32, 32]	30	0
$F11 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[−600, 600]	30	0

(Continued)

Table 3 (continued)

(F_i) Function	Range	Dim	f_{min}
$F12 = \frac{\pi}{n}\left\{10\sum_{i=1}^{n-1}\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2[1+10sin^2(\pi y_{i+1})] + (y_n-1)^2\right\} + \sum_{i=1}^{n}u(x_i,10,100,4), y_i = 1+\frac{x_i+1}{4}u(x_i,a,m) = \begin{cases} k(x_i-a)^m) & x_i > a \\ 0 & a < x_i < a \\ k(-x_i-a)^m & x_i < -a \end{cases}$	[−50, 50]	30	0
$F13 = 0.1\left\{sin^2(3\pi x_1) + \sum_{i-1}^{n}(x_i-1)^2\left[1+sin^2(3\pi x_i+1)\right] + (x_n-1)^2\left[1+sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n}u(x_i,5,100,4)$	[−50, 50]	30	0

Table 4: Fixed-dimension multimodal benchmark functions

(F_i) Function	Range	Dim	f_{min}
$F14 = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right)^{-1}$	[−65, 65]	2	1
$F15 = \sum_{i=1}^{11}\left[a_i - \frac{x_1\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right]^2$	$[-5, 5]^4$	2	1
$F16 = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5, 5]^2$	2	−1.0316
$F17 = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1-\frac{1}{8\pi}\right)\cos x_1 + 10$	$[-5, 5]^2$	2	0.398
$F18 = \left[1+(x_1+x_2+1)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]\left[30+(2x_1-3x_2)^2\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$	[−2, 2]	2	3

(Continued)

Table 4 (continued)

(F_i) Function	Range	Dim	f_{min}
$F19 = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	[1, 3]	3	−3.86
$F20 = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	[0, 1]	6	−3.32
$F21 = -\sum_{i=1}^{5}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	[0, 10]	4	−10.1532
$F22 = -\sum_{i=1}^{7}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	[0, 10]	4	−10.4028
$F23 = -\sum_{i=1}^{10}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	[0, 10]	4	−105.363

In this study, CEC2019 benchmark functions were also used to evaluate the performance of the COARO algorithm. These functions are minimization problems. cec01–cec03 are uncomplicated problems with different dimensions and ranges. cec04–cec10 are rotated and shifted and have the same dimension and range. Mathematical expressions, dimensions, ranges, and minimum values of function for CEC2019 functions are indicated in Table 5.

Table 5: CEC2019 benchmark functions

Function	Range	Dim	f_{min}
cec01- Store's Chebyshev Polynomial Fitting	[−8192, 8192]	9	1
cec02- Inverse Hilbert Matrix	[−16384, 16384]	16	1
cec03- Lennard Jones Min Energy Cluster	[−4, 4]	18	1
cec04- Rastrigin	[−100, 100]	10	1
cec05- Griewangk	[−100, 100]	10	1
cec06- Weierstrass	[−100, 100]	10	1
cec07- Modified Schwefels	[−100, 100]	10	1
cec08- Expanded Schaffer F6	[−100, 100]	10	1
cec09- Happy Cat	[−100, 100]	10	1
cec10- Ackley	[−100, 100]	10	1

Before performing the experimental tests, the running parameters of the metaheuristic algorithms were adjusted. For a fair evaluation, the max iteration number was chosen as 1000 and the number of population was 50. Metaheuristic algorithms were executed 30 times in all experiments. Specific parameter values of GWO, MVO, PSO, and TSO algorithms were derived based on parameter values widely utilized in the literature and these values are included in Table 6.

Table 6: Parameters settings of the metaheuristic algorithms

Algorithm	Parameter setting
ARO	–
GWO	$a = [2, 0]$
MVO	$WEP_{min} = 0.2$
	$WEP_{max} = 1$
PSO	$C_1 = 1.5$
	$C_2 = 2.0$
	Inertia weight = 1
TSO	k = 1

The best, mean, standard deviation (SD), and Friedman mean rank (MR) values obtained for the classical and CEC2019 benchmark functions with the COARO and ARO algorithms are given in Tables 7 and 8, respectively.

Table 7 analysis reveals that for unimodal functions, the proposed COARO method performs better than ARO. The best results have been achieved with COARO3 for F1–F4 and F7, and COARO4 for F5, according to MR values. It is observed that for the F6 function, the suggested COARO algorithm and ARO yield the same value. In multimodal benchmark functions, COARO3 for F8 and F12 functions and COARO10 for F13 function showed the best performance. The proposed COARO algorithm and ARO achieve similar performance for F9–F11. It is observed that for fixed-dimension multimodal benchmark functions, the proposed COARO algorithms with ten chaotic maps perform similarly to the ARO algorithm.

Table 8 shows that the proposed COARO algorithm and ARO achieve similar results for cec02 and cec03. Additionally, it is observed that COARO9 for cec01, COARO3 for cec04 and cec05, COARO5 for cec06, cec09 and cec10, COARO6 for cec07 and COARO10 for cec08 achieved the best results.

Table 7: Experimental results on classical benchmark functions

Benchmark	Criteria	Algorithms										
		ARO	COAROs									
			1	2	3	4	5	6	7	8	9	10
F1	**Best**	7.15E-143	4.90E-172	5.13E-131	2.34E-206	2.20E-153	4.99E-165	1.78E-143	6.78E-169	7.32E-150	1.26E-137	4.35E-142
	Mean	3.08E-125	2.37E-161	7.04E-124	1.64E-199	1.15E-145	2.53E-157	5.26E-135	6.33E-160	4.12E-147	8.52E-135	5.50E-133
	SD	7.80E-125	4.82E-161	1.44E-123	0	2.10E-145	5.94E-157	8.80E-135	9.77E-160	6.20E-147	1.20E-134	1.10E-132
	MR	9.05	2.1	11	1	5.7	4	7.05	2.9	5.3	8.85	9.05
F2	**Best**	1.87E-76	4.90E-84	1.26E-82	7.56E-110	4.98E-78	5.49E-77	2.61E-73	1.86E-88	5.18E-81	1.12E-85	3.62E-77
	Mean	2.51E-69	2.96E-83	6.62E-80	2.10E-106	4.00E-74	3.47E-74	1.54E-70	3.51E-85	1.16E-78	7.96E-82	1.38E-74
	SD	5.14E-69	2.43E-83	8.59E-80	3.50E-106	3.88E-74	3.81E-74	1.64E-70	5.58E-85	1.46E-78	1.02E-81	1.82E-74
	MR	10.5	3.05	5	1	9.05	8.5	9.85	2	6	3.95	7.1
F3	**Best**	1.42E-115	6.69E-149	8.31E-125	1.37E-173	1.04E-125	2.80E-121	7.04E-117	1.08E-145	3.80E-126	5.75E-117	7.41E-116
	Mean	2.03E-93	1.73E-137	2.52E-119	1.82E-162	8.57E-120	3.52E-109	2.34E-109	3.20E-135	2.28E-122	1.03E-111	2.02E-110
	SD	8.96E-93	4.30E-137	4.46E-119	4.45E-162	1.30E-119	5.90E-109	6.10E-109	5.20E-135	2.70E-122	1.60E-111	3.66E-110
	MR	10.35	2.1	5.6	1	5.4	9.45	9.45	2.9	4	7.8	7.95
F4	**Best**	2.36E-59	1.55E-74	1.67E-64	1.01E-86	8.85E-67	3.57E-74	1.28E-60	3.21E-73	1.08E-66	2.64E-60	1.56E-61
	Mean	8.02E-52	2.79E-71	3.73E-62	1.69E-83	4.39E-63	2.62E-69	9.28E-57	6.65E-70	1.08E-63	1.59E-58	2.33E-58
	SD	2.97E-51	4.37E-71	4.98E-62	1.51E-83	5.18E-63	3.23E-69	1.38E-56	9.38E-70	1.34E-63	1.83E-58	2.19E-58
	MR	10.85	2	7	1	5.95	3.75	10.1	3.25	5.05	8.2	8.85
F5	**Best**	2.16E-04	5.14E-05	8.00E-05	9.59E-05	1.58E-04	4.14E-05	1.80E-04	3.42E-05	1.47E-04	1.99E-04	1.09E-04
	Mean	4.98E-04	3.28E-04	5.41E-04	5.90E-04	3.18E-04	3.80E-04	6.28E-04	3.53E-04	4.57E-04	3.67E-04	6.06E-04
	SD	1.66E-04	1.48E-04	2.81E-04	2.91E-04	1.11E-04	1.71E-04	2.52E-04	1.94E-04	1.76E-04	1.23E-04	2.31E-04
	MR	7	2.5	7.95	8.9	2.33	4.3	10.15	3.3	6.55	4.28	8.75
F6	**Best**	0	0	0	0	0	0	0	0	0	0	0
	Mean	0	0	0	0	0	0	0	0	0	0	0
	SD	0	0	0	0	0	0	0	0	0	0	0
	MR	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
F7	**Best**	2.38E-05	4.33E-06	4.17E-06	2.93E-06	4.26E-06	9.90E-06	3.81E-06	8.80E-06	3.54E-06	5.53E-06	8.60E-06
	Mean	6.73E-05	4.91E-05	4.67E-05	4.33E-05	6.76E-05	5.63E-05	4.91E-05	5.88E-05	4.99E-05	5.44E-05	7.92E-05
	SD	3.28E-05	2.77E-05	3.70E-05	1.81E-05	2.40E-05	2.67E-05	2.00E-05	2.92E-05	3.09E-05	2.61E-05	2.77E-05
	MR	8.89	4.16	3.32	3.05	9.11	6.08	3.47	6.79	4.79	5.66	10.68
F8	**Best**	−11286.3	−12331.9	−12095.69	−12332.5	−12213.9	−12214	−12213.8	−12095.7	−12095.4	−11799.6	−12331.9
	Mean	−11144.1	−12039.11	−11687.01	−12053	−11792.6	−11754.9	−11669.2	−11802.2	−11580.4	−11182.7	−11788.6
	SD	99.9002	166.8398	203.2545	147.0704	183.4033	157.24	215.1382	133.438	194.5418	228.4033	166.8398
	MR	10.5	1.5	6.75	1.5	4.45	5.95	7.45	3.7	9	10.5	4.7
F9	**Best**	0	0	0	0	0	0	0	0	0	0	0
	Mean	0	0	0	0	0	0	0	0	0	0	0
	SD	0	0	0	0	0	0	0	0	0	0	0
	MR	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
F10	**Best**	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	Mean	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	SD	0	0	0	0	0	0	0	0	0	0	0
	MR	6	6	6	6	6	6	6	6	6	6	6

(Continued)

Table 7 (continued)

Benchmark	Criteria	Algorithms										
		ARO	COAROs									
			1	2	3	4	5	6	7	8	9	10
F11	**Best**	0	0	0	0	0	0	0	0	0	0	0
	Mean	0	0	0	0	0	0	0	0	0	0	0
	SD	0	0	0	0	0	0	0	0	0	0	0
	MR	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
F12	**Best**	5.44E-09	3.95E-09	3.19E-09	3.12E-09	4.79E-09	3.79E-09	3.62E-09	1.92E-09	3.20E-09	3.36E-09	3.95E-09
	Mean	1.70E-08	1.68E-08	1.84E-08	6.59E-09	1.05E-08	9.31E-09	1.19E-08	7.53E-09	1.22E-08	2.10E-08	1.68E-08
	SD	6.56E-09	5.03E-09	7.18E-09	2.16E-09	2.92E-09	2.65E-09	4.79E-09	2.51E-09	3.80E-09	6.36E-09	5.03E-09
	MR	9.05	6.35	9.2	1.2	4.4	2.95	5.55	1.85	6.2	10.65	8.6
F13	**Best**	5.08E-08	1.10E-08	2.08E-08	9.41E-09	2.99E-08	2.37E-08	4.94E-09	6.22E-09	1.52E-08	2.60E-08	1.10E-08
	Mean	1.48E-07	6.35E-08	1.38E-07	4.76E-08	8.62E-08	8.66E-08	4.70E-08	6.64E-08	4.34E-08	4.90E-08	4.22E-08
	SD	5.42E-08	1.48E-08	5.87E-08	1.83E-08	3.36E-08	3.54E-08	1.65E-08	3.17E-08	1.51E-08	1.15E-08	1.48E-08
	MR	10.7	5.85	10.05	3.95	8.45	8.55	3.8	6.2	2.2	4.6	1.65
F14	**Best**	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01
	Mean	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01
	SD	1.18E-05	5.90E-06	8.86E-06	7.68E-06	6.49E-06	1.00E-05	8.27E-06	7.09E-06	9.45E-06	1.06E-05	1.12E-05
	MR	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
F15	**Best**	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04
	Mean	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04	3.07E-04
	SD	3.64E-09	1.82E-09	2.74E-09	2.36E-09	2.00E-09	3.09E-09	2.55E-09	2.18E-09	2.91E-09	3.27E-09	3.45E-09
	MR	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
F16	**Best**	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00
	Mean	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00
	SD	1.22E-05	6.10E-06	9.15E-06	7.93E-06	6.71E-06	1.04E-05	8.54E-06	7.32E-06	9.77E-06	1.10E-05	1.16E-05
	MR	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
F17	**Best**	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01
	Mean	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01
	SD	4.71E-06	2.35E-06	3.53E-06	3.06E-06	2.59E-06	4.00E-06	3.30E-06	2.82E-06	3.77E-06	4.24E-06	4.47E-06
	MR	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
F18	**Best**	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00
	Mean	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00
	SD	3.55E-05	1.77E-05	2.66E-05	2.31E-05	1.95E-05	3.02E-05	2.48E-05	2.13E-05	2.48E-05	3.19E-05	3.37E-05
	MR	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
F19	**Best**	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00
	Mean	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00
	SD	4.57E-05	2.29E-05	3.43E-05	2.97E-05	2.51E-05	3.88E-05	3.20E-05	2.74E-05	3.66E-05	4.11E-05	4.34E-05
	MR	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
F20	**Best**	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00
	Mean	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00
	SD	3.93E-05	1.97E-05	2.95E-05	2.55E-05	2.16E-05	3.34E-05	2.75E-05	2.36E-05	3.14E-05	3.54E-05	3.73E-05
	MR	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00

(Continued)

Table 7 (continued)

Benchmark	Criteria	Algorithms										
		ARO	COAROs									
			1	2	3	4	5	6	7	8	9	10
F21	**Best**	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01
	Mean	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01
	SD	1.20E-04	6.01E-05	9.02E-05	7.81E-05	6.61E-05	1.02E-04	8.41E-05	7.21E-05	9.61E-05	1.08E-04	1.14E-04
	MR	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
F22	**Best**	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−-1.04E+01
	Mean	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01
	SD	1.23E-04	6.15E-05	9.23E-05	8.00E-05	6.77E-05	1.05E-04	8.62E-05	7.39E-05	9.85E-05	1.11E-04	1.17E-04
	MR	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
F23	**Best**	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01
	Mean	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01
	SD	1.25E-04	6.23E-05	9.35E-05	8.10E-05	6.86E-05	1.06E-04	8.73E-05	7.48E-05	9.97E-05	1.12E-04	1.18E-04
	MR	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00

Table 8: Experimental results on CEC2019 benchmark functions

Benchmark	Criteria	Algorithms										
		ARO	COAROs									
			1	2	3	4	5	6	7	8	9	10
cec01	**Best**	3.65E+04	3.58E+04	3.41E+04	3.53E+04	3.56E+04	3.50E+04	3.50E+04	3.46E+04	3.31E+04	3.41E+04	3.21E+04
	Mean	3.73E+04	3.77E+04	3.66E+04	3.66E+04	3.68E+04	3.66E+04	3.66E+04	3.63E+04	3.67E+04	3.61E+04	3.64E+04
	SD	4.44E+02	7.49E+02	9.14E+02	5.73E+02	4.63E+02	5.48E+02	4.84E+02	7.70E+02	1.03E+03	9.24E+02	1.17E+03
	MR	10.13	10.88	6.98	4.48	7.65	5.3	4.8	2.53	7.45	1.45	4.38
cec02	**Best**	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01
	Mean	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01
	SD	1.56E-03	1.58E-03	1.78E-03	1.45E-03	1.15E-03	1.73E-03	1.96E-03	1.09E-03	1.66E-03	1.68E-03	1.48E-03
	MR	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0
cec03	**Best**	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01
	Mean	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01
	SD	7.59E-05	7.51E-05	9.02E-05	3.98E-06	9.77E-05	1.28E-04	1.05E-04	1.20E-04	8.49E-05	8.56E-05	8.34E-05
	MR	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0
cec04	**Best**	7.96E+00	4.97E+00	3.98E+00	2.98E+00	4.97E+00	2.98E+00	4.97E+00	3.98E+00	1.99E+00	3.98E+00	3.98E+00
	Mean	1.23E+01	1.02E+01	9.31E+00	9.11E+00	1.09E+01	1.05E+01	1.08E+01	1.03E+01	9.55E+00	9.96E+00	1.08E+01
	SD	1.63E+00	2.82E+00	2.94E+00	2.20E+00	2.19E+00	2.79E+00	2.72E+00	3.10E+00	2.47E+00	3.43E+00	2.51E+00
	MR	10.5	5.83	3	2.65	7.63	6.45	7.58	6.1	3.53	5.3	7.45
cec05	**Best**	1.02E+00	1.00E+00	1.00E+00	1.00E+00	1.01E+00	1.00E+00	1.01E+00	1.00E+00	1.01E+00	1.01E+00	1.01E+00
	Mean	1.03E+00	1.02E+00	1.03E+00	1.02E+00	1.02E+00	1.03E+00	1.03E+00	1.02E+00	1.03E+00	1.03E+00	1.03E+00
	SD	7.13E-03	8.12E-03	1.16E-02	1.18E-02	8.38E-03	1.14E-02	9.60E-03	1.10E-02	9.05E-03	8.95E-03	8.41E-03
	MR	8.93	3.2	7.7	2.45	5.88	6.1	6.68	3.95	7.9	7.13	6.1
cec06	**Best**	1.55E+00	1.07E+00	1.03E+00	1.30E+00	1.20E+00	1.00E+00	1.46E+00	1.19E+00	1.14E+00	1.01E+00	1.67E+00
	Mean	2.51E+00	2.04E+00	2.59E+00	2.37E+00	2.31E+00	1.91E+00	2.38E+00	2.22E+00	2.40E+00	2.30E+00	2.57E+00
	SD	5.21E-01	4.01E-01	6.22E-01	6.27E-01	4.22E-01	5.77E-01	4.42E-01	3.88E-01	4.58E-01	6.61E-01	4.01E-01
	MR	8.28	2.18	9.78	6.18	5.63	1.55	6.43	4.15	6.98	5.25	9.63
cec07	**Best**	−2.91E+02	−3.14E+02	−3.13E+02	−3.66E+02	−3.53E+02	−3.54E+02	−4.17E+02	−2.79E+02	−3.54E+02	−3.16E+02	−3.38E+02
	Mean	−1.94E+02	−1.85E+02	−1.69E+02	−1.84E+02	−1.66E+02	−1.76E+02	−2.22E+02	−1.96E+02	−2.15E+02	−2.20E+02	−2.11E+02
	SD	4.71E+01	4.90E+01	5.50E+01	7.15E+01	5.80E+01	6.94E+01	7.83E+01	3.07E+01	5.19E+01	3.11E+01	4.21E+01
	MR	6.15	7.33	8.98	7.38	9.48	8.48	2.68	6.03	3.15	2.8	3.58
cec08	**Best**	1.71E+00	1.66E+00	1.60E+00	1.63E+00	1.49E+00	1.55E+00	1.83E+00	1.53E+00	1.65E+00	1.51E+00	1.50E+00
	Mean	2.46E+00	2.27E+00	2.33E+00	2.39E+00	2.23E+00	2.32E+00	2.41E+00	2.39E+00	2.35E+00	2.47E+00	2.19E+00
	SD	3.35E-01	3.22E-01	3.76E-01	3.30E-01	2.76E-01	3.10E-01	2.39E-01	3.40E-01	3.43E-01	3.28E-01	2.83E-01
	MR	9.08	3.78	5.7	7.75	2.93	5.35	6.45	7.4	6.33	9.58	1.68
cec09	**Best**	2.34E+00	2.35E+00	2.35E+00	2.35E+00	2.35E+00	2.35E+00	2.35E+00	2.34E+00	2.35E+00	2.35E+00	2.35E+00
	Mean	2.41E+00	2.41E+00	2.41E+00	2.41E+00	2.41E+00	2.41E+00	2.41E+00	2.41E+00	2.41E+00	2.41E+00	2.41E+00
	SD	4.88E-03	5.42E-03	5.75E-03	5.16E-03	5.88E-03	5.25E-03	5.41E-03	4.97E-03	3.93E-03	3.60E-03	5.25E-03
	MR	6.8	5.7	5.98	6.25	5.7	5.15	5.7	5.7	5.43	6.8	6.8
cec10	**Best**	1.16E+00	2.01E+00	1.35E+00	1.16E+00	1.16E+00	1.16E+00	1.16E+00	1.16E+00	1.99E+01	2.00E+01	1.16E+00
	Mean	1.85E+01	1.90E+01	1.71E+01	1.51E+01	1.50E+01	1.43E+01	1.61E+01	1.77E+01	2.00+01	1.80E+01	1.81E+01
	SD	4.67E+00	4.12E+00	6.80E+00	8.30E+00	8.28E+00	8.49E+00	7.54E+00	5.80E+00	1.64E-02	6.56E-03	5.80E+00
	MR	6.95	7	5.97	5.08	4.55	4.34	5.76	6.34	7.05	6.45	6.5

Based on the above analyses, combining CLS and OBL techniques and adding 10 chaotic maps to the optimization process, providing efficiency in determining global solutions, contributed to the COARO algorithm achieving promising results compared to the ARO algorithm in the classical and CEC2019 benchmark functions. Additionally, the results include validation of enhanced exploration advantage, increased diversity, and the ability to find better global resolution.

By integrating CLS into the COARO algorithm, dynamic behaviors are added to the movement rules, allowing rabbits to better explore the search space. With CLS, it is also possible to reduce the possibility of getting stuck in local optima and increase the convergence speed by maintaining a balance between exploration and exploitation. The addition of OBL to COARO provides the ability to explore opposing directions based on existing solutions. OBL encourages rabbits to explore opposite directions and discover potentially different but feasible solutions, rather than simply advancing towards optimal solutions. Exploring opposite directions has allowed COARO to discover solutions that ARO might have overlooked. Thus, the robustness of COARO is increased by providing alternative search areas. These gains are supported by the effective experimental results obtained in Tables 7 and 8.

The average Friedman MR values for the classic and CEC2019 benchmark functions are shown in Table 9. According to Table 9, the best performance for classical benchmark functions was achieved with the COARO3 algorithm. According to the same table, COARO1 achieved the second-best performance by following the COARO3 algorithm. The best performance for CEC2019 benchmark functions was achieved with the COARO3 and COARO7 algorithms. These algorithms are followed by the COARO5 algorithm with a close value.

Table 9: The values of average Friedman MR for classical and CEC2019 benchmark functions

	ARO	COAROs									
		1	2	3	4	5	6	7	8	9	10
Classical	7.73	4.94	6.51	**4.63**	6.03	5.98	6.56	5.08	5.78	6.45	6.58
CEC2019	7.88	5.79	6.61	**5.42**	6.14	5.47	5.80	**5.42**	5.98	5.67	5.81

Fast convergence of metaheuristic algorithms to the optimal solution is crucial to its effectiveness. Figs. 1 and 2 show the convergence graphs of the some classic and CEC2019 benchmark functions for COARO and ARO, respectively. Convergence curves show that the behavior of the proposed COARO algorithm and ARO varies throughout iterations for given functions.

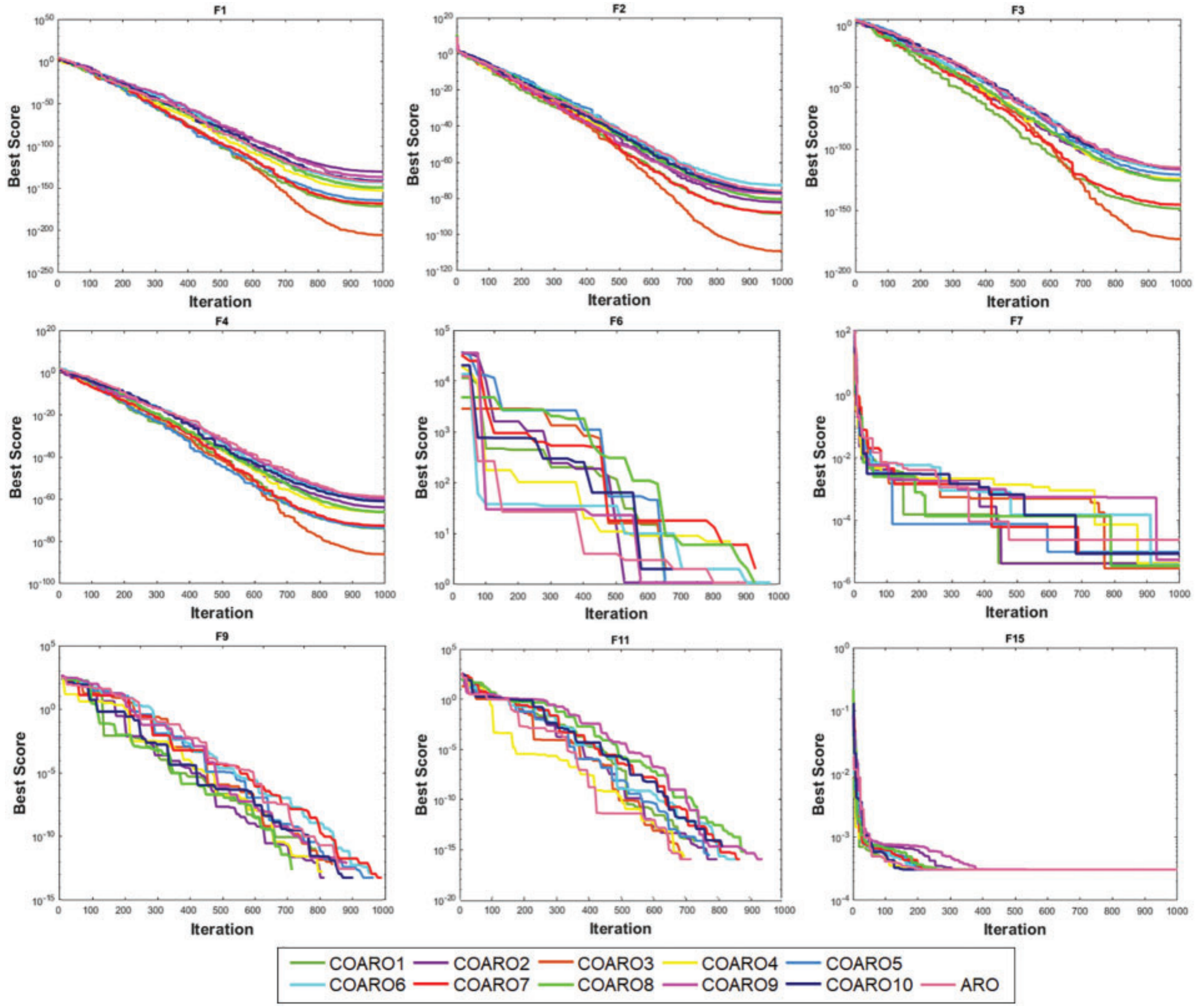

Figure 1: Convergence curve of some classical benchmark functions for COAROs and ARO

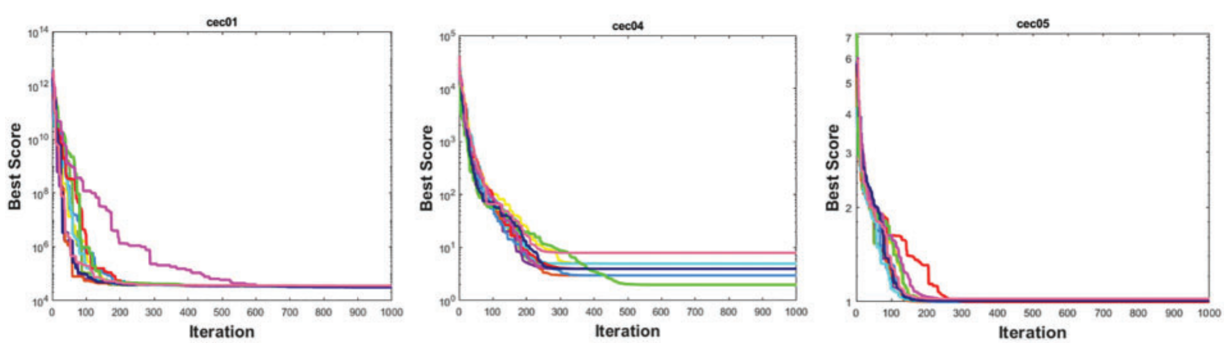

Figure 2: (Continued)

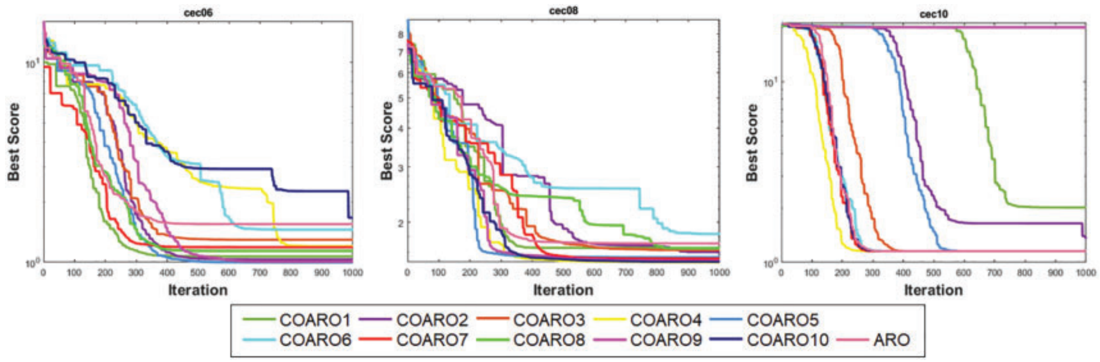

Figure 2: Convergence curve of some CEC2019 benchmark functions for COAROs and ARO

In addition to ARO, four cutting-edge metaheuristic algorithms: GWO, MVO, PSO, and TSO are employed to assess the performance of the proposed COARO algorithm. Statistical results such as best, mean and SD values obtained for the classic and CEC2019 benchmark functions are given in Tables 10 and 11. Examining the mean values in Table 10, it can be said that in 22 of the 23 classical benchmark functions except for F8, the COARO performs better than the other metaheuristic algorithms. Likewise, according to Table 11, the proposed COARO algorithm achieved superior performances than other algorithms for CEC2019 benchmark functions, except for the cec09 function. Figs. 3 and 4 show the convergence graphs of the some classical and CEC2019 benchmark functions for COARO and competitive algorithms. When Figs. 3 and 4 are examined, it has been noticed that the COARO algorithm's convergence speed is faster than the ARO, GWO, MVO, PSO, and TSO algorithms.

Table 10: The classical benchmark functions result of COARO and other algorithms

Benchmark	Criteria	Algorithms					
		ARO	COARO	GWO	MVO	PSO	TSO
F1	**Best**	7.15E-143	2.34E-206	6.55E-92	5.49E-04	1.60E-22	6.01E-17
	Mean	3.08E-125	1.64E-199	1.83E-91	1.01E-03	2.82E-19	1.23E-12
	SD	7.8E-125	0	1.14E-91	2.70E-04	5.35E-19	1.63E-12
F2	**Best**	1.87E-76	7.56E-110	1.01E-55	5.36E-03	1.52E-14	8.12E-15
	Mean	2.51E-69	2.10E-106	5.06E-53	9.68E-03	1.80E-09	1.89E-12
	SD	5.14E-69	3.5E-106	9.06E-53	1.44E-03	1.80E-09	1.97E-12
F3	**Best**	1.42E-115	1.37E-173	2.79E-33	1.34E-03	3.03E-02	5.28E-06
	Mean	2.03E-93	1.82E-162	1.83E-24	3.10E-03	8.91E-02	4.37E-02
	SD	8.96E-93	4.45E-162	3.21E-24	1.33E-03	3.20E-02	4.50E-02
F4	**Best**	2.36E-59	1.01E-86	4.10E-26	1.80E-02	1.81E+00	1.41E-02
	Mean	8.02E-52	1.69E-83	3.34E-13	2.09E-02	1.81E+00	3.73E-02
	SD	2.97E-51	1.51E-83	5.95E-13	1.83E-03	0	1.72E-02
F5	**Best**	2.16E-04	1.58E-04	2.51E+01	2.15E+00	5.19E-04	3.68E-03
	Mean	4.98E-04	3.18E-04	2.66E+01	5.34E+00	8.69E-04	4.11E-02
	SD	1.66E-04	1.11E-04	1.44E+00	1.72E+00	1.46E-04	2.26E-02
F6	**Best**	0	0	2.65E-05	5.91E-04	3.49E-24	9.38E-05
	Mean	0	0	4.01E-01	9.07E-04	3.90E-20	1.26E-02
	SD	0	0	4.29E-01	1.83E-04	8.23E-20	6.65E-03
F7	**Best**	2.38E-05	2.93E-06	5.91E-05	1.89E-04	1.81E-03	4.21E-05
	Mean	6.73E-05	4.33E-05	5.17E-04	4.46E-04	2.22E-02	4.23E-03

(Continued)

Table 10 (continued)

Benchmark	Criteria	Algorithms					
		ARO	COARO	GWO	MVO	PSO	TSO
	SD	3.19E-05	1.81E-05	5.65E-04	1.51E-04	1.84E-02	4.44E-03
F8	**Best**	−11286.3	−12332.5	−12566.3	−3735.8	−12569.5	−12493.8
	Mean	−11144.1	−12053	−12566.2	−3531.8	−12535.9	−12481.0
	SD	99.9002	147.0704	0.1572	133.4417	74.4729	9.5867
F9	**Best**	0	0	1.56E+01	2.99E+00	5.68E-14	5.68E-14
	Mean	0	0	1.56E+01	4.78E+00	7.39E-14	5.51E-13
	SD	0	0	1.58E-08	1.13E+00	2.75E-14	7.69E-13
F10	**Best**	8.88E-16	8.88E-16	7.99E-15	1.25E-02	3.29E-14	5.76E-10
	Mean	8.88E-16	8.88E-16	7.17E-01	1.33E-02	1.01E-12	5.21E-08
	SD	0	0	1.51E+00	5.21E-04	4.88E-13	5.19E-08
F11	**Best**	0	0	1.05E-02	8.36E-02	3.43E-02	1.11E-16
	Mean	0	0	2.78E-02	1.34E-01	4.44E-02	3.41E-13
	SD	0	0	1.40E-02	2.74E-02	1.35E-02	4.52E-13
F12	**Best**	5.44E-09	3.12E-09	2.27E-06	8.14E-06	5.20E-12	3.12E-06
	Mean	1.70E-08	6.59E-09	1.18E-03	3.71E-05	1.87E-01	1.36E-03
	SD	6.56E-09	2.16E-09	2.45E-03	9.97E-06	1.53E-01	2.40E-03
F13	**Best**	5.08E-08	1.10E-08	3.83E-05	4.39E-05	1.10E-02	1.15E-05
	Mean	1.48E-07	4.22E-08	2.37E-01	9.70E-05	2.20E-02	3.23E-03
	SD	5.42E-08	1.48E-08	3.12E-01	2.92E-05	1.79E-02	5.09E-03
F14	**Best**	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01
	Mean	9.98E-01	9.98E-01	1.99E+00	9.98E-01	9.98E-01	9.98E-01
	SD	1.18E-05	8.86E-06	1.05E+00	2.00E-13	3.15E-10	4.80E-13
F15	**Best**	3.07E-04	3.07E-04	1.22E-03	3.08E-04	3.07E-04	3.07E-04
	Mean	3.07E-04	3.07E-04	7.04E-03	3.39E-04	3.07E-04	3.08E-04
	SD	3.64E-09	1.82E-09	9.20E-03	3.08E-05	1.36E-19	3.71E-08
F16	**Best**	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00
	Mean	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00	−1.04E+00
	SD	1.22E-05	9.15E-06	6.97E-10	1.90E-09	0	6.19E-10
F17	**Best**	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01
	Mean	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01	4.00E-01
	SD	4.71E-06	2.35E-06	1.88E-08	5.22E-09	3.56E-06	7.67E-09
F18	**Best**	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00
	Mean	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00	2.99E+00
	SD	3.55E-05	1.77E-05	3.41E-06	1.65E-08	1.60E-15	8.57E-08
F19	**Best**	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00
	Mean	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00	−3.90E+00
	SD	4.57E-05	3.43E-05	4.45E-07	3.79E-09	4.15E-03	4.21E-06
F20	**Best**	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00
	Mean	−3.29E+00	−3.29E+00	−3.29E+00	−3.29E+00	−3.16E+00	−3.29E+00
	SD	3.93E-05	2.95E-05	6.31E-07	3.79E-09	1.60E-01	4.55E-05
F21	**Best**	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01
	Mean	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−5.61E+00	−1.02E+01
	SD	1.20E-04	9.02E-05	1.08E-04	6.68E-06	1.60E+00	2.68E-03
F22	**Best**	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01
	Mean	−1.04E+01	−1.04E+01	−6.68E+00	−7.24E+00	−5.66E+00	−1.04E+01
	SD	1.23E-04	8.00E-05	2.57E+00	2.72E+00	1.66E+00	5.74E-04
F23	**Best**	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01
	Mean	−1.04E+01	−1.04E+01	−1.04E+01	−6.25E+00	−1.04E+01	−1.04E+01
	SD	1.25E-04	8.10E-05	2.76E-04	2.26E+00	1.87E-15	7.21E-04

Table 11: CEC2019 benchmark functions results of COARO and other algorithms

Benchmark	Criteria	Algorithms					
		ARO	COARO	GWO	MVO	PSO	TSO
cec01	**Best**	3.65E+04	3.41E+04	3.63E+06	2.14E+07	1.17E+05	7.39E+04
	Mean	3.73E+04	3.61E+04	1.02E+07	2.46E+08	3.87E+06	5.25E+05
	SD	4.44E+02	9.24E+02	5.98E+06	1.20E+08	3.73E+06	3.01E+05
cec02	**Best**	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01
	Mean	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01	1.73E+01
	SD	1.56E-03	1.15E-03	1.44E-04	4.93E-05	3.16E-05	1.72E-04
cec03	**Best**	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01
	Mean	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01	1.30E+01
	SD	7.59E-05	1.28E-04	7.54E-07	4.10E-11	4.57E-07	5.16E-10
cec04	**Best**	7.96E+00	2.98E+00	9.08E+00	1.07E+01	9.95E+00	6.67E+01
	Mean	1.23E+01	9.11E+00	1.20E+03	1.24E+01	1.48E+01	9.33E+01
	SD	1.63E+00	2.20E+00	1.20E+03	8.92E-01	3.46E+00	1.23E+01
cec05	**Best**	1.02E+00	1.00E+00	1.56E+00	1.06E+00	1.03E+00	1.28E+00
	Mean	1.03E+00	1.02E+00	1.59E+00	1.09E+00	1.04E+00	1.35E+00
	SD	7.13E-03	1.18E-02	2.16E-02	1.43E-02	5.24E-03	3.28E-02
cec06	**Best**	1.55E+00	1.00E+00	4.74E+00	4.33E+00	3.60E+00	6.79E+00
	Mean	2.51E+00	1.91E+00	9.42E+00	4.90E+00	8.28E+00	7.15E+00
	SD	5.21E-01	5.77E-01	1.91E+00	3.35E-01	3.21E+00	2.97E-01
cec07	**Best**	−2.91E+02	−4.17E+02	−1.36E+02	−1.86E+02	−1.30E+02	1.99E+01
	Mean	−1.94E+02	−2.22E+02	−2.80E+01	−7.07E+01	−6.35E+01	1.22E+02
	SD	4.71E+01	7.83E+01	5.19E+01	5.76E+01	3.66E+01	5.05E+01
cec08	**Best**	1.71E+00	1.50E+00	2.71E+00	2.85E+00	2.37E+00	4.26E+00
	Mean	2.46E+00	2.19E+00	5.09E+00	3.44E+00	4.32E+00	5.93E+00
	SD	3.35E-01	2.83E-01	1.46E+00	3.11E-01	1.96E+00	1.08E+00
cec09	**Best**	2.34E+00	2.35E+00	2.55E+00	2.34E+00	2.35E+00	2.86E+00
	Mean	2.41E+00	2.41E+00	4.31E+00	2.35E+00	2.37E+00	5.96E+00
	SD	4.88E-03	5.25E-03	1.55E+00	1.97E-03	1.37E-02	3.14E+00
cec10	**Best**	1.16E+00	1.16E+00	3.20E+00	2.00E+01	2.05E+01	2.01E+01
	Mean	1.86E+01	1.43E+01	1.63E+01	2.00E+01	2.05E+01	2.01E+01
	SD	4.67E+00	8.49E+00	7.04E+00	1.72E-03	1.92E-02	1.28E-02

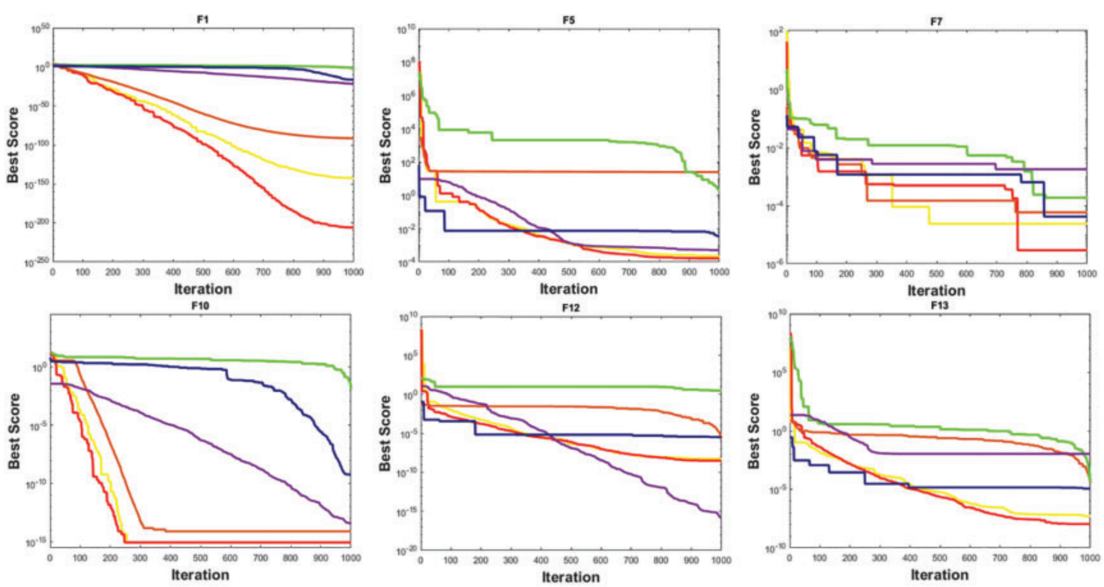

Figure 3: (Continued)

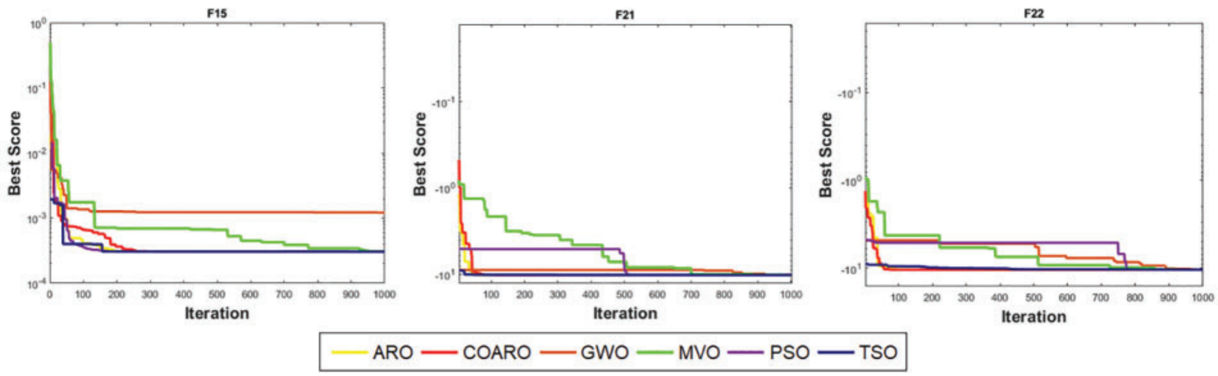

Figure 3: Convergence curve of some classical benchmark functions for all algorithms

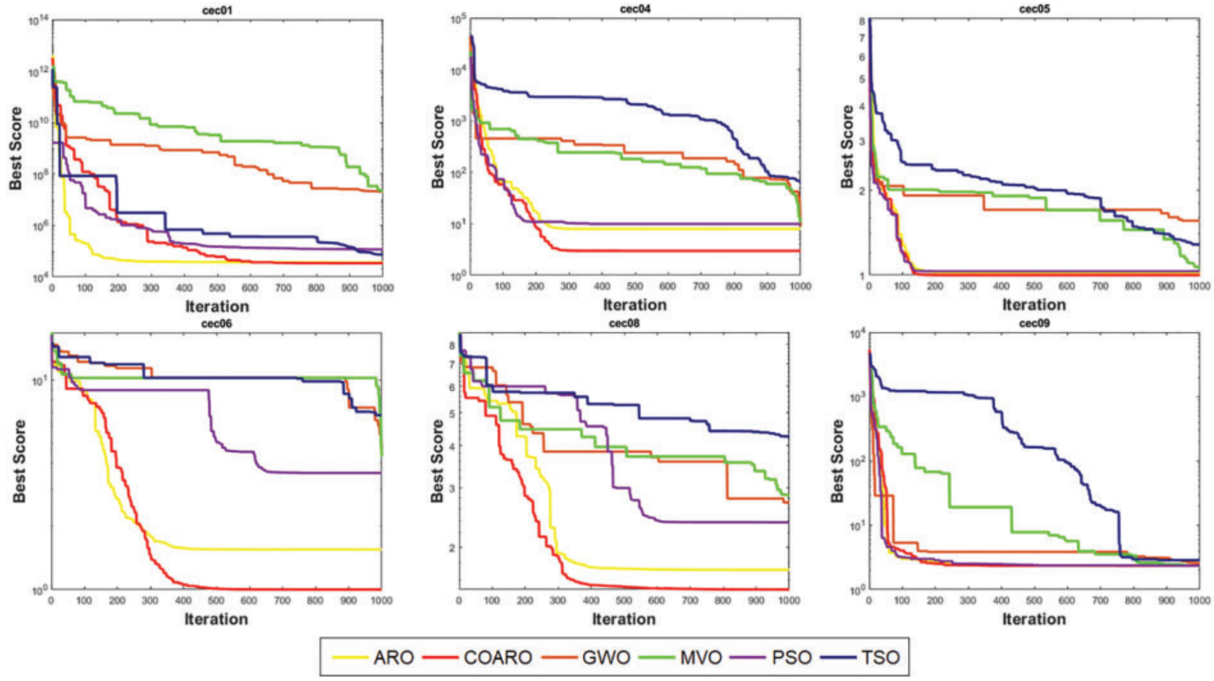

Figure 4: Convergence curve of some CEC2019 benchmark functions for all algorithms

The proposed COARO algorithm has achieved promising results for the classical and CEC2019 benchmark functions. However, whether there is a notable distinction between the COARO algorithm and ARO, GWO, MVO, PSO, and TSO should be tested. Therefore, the WSR statistical test has been used to differentiate between COARO and competitive algorithms. To verify the effectiveness of the COARO algorithm, the WSR test was applied after 30 runs at a 95% confidence level. In the evaluations, the maximum number of iterations was chosen as 1000. There is a discernible difference between the compared algorithms if the p-value determined by the comparisons is less than 0.05. Otherwise, there aren't any notable distinctions between the two metaheuristic algorithms. Tables 12 and 13 provide the outcomes of the WSR test for the standard and CEC2019 benchmark functions comparing COARO and competitor algorithms. In these tables, "+" indicates the superiority of the proposed COARO algorithm, "–" demonstrates the proposed COARO algorithm is worse than the competitive metaheuristic algorithm, and "=" means that COARO and the compared algorithm obtained the same values.

Table 12: WSR test results between COARO and competitive algorithms for classical benchmark functions

Benchmark	COARO *vs.* ARO		COARO *vs.* GWO		COARO *vs.* MVO		COARO *vs.* PSO		COARO *vs.* TSO	
	p-value	win	*p*-value	win	*p*-value	win	*p*-value	win	*p*-value	win
F1	2.00E-03	+	2.00E-03	+	4.88E-03	+	2.00E-03	+	4.88E-03	+
F2	2.00E-03	+	2.00E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
F3	2.00E-03	+	2.00E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
F4	2.00E-03	+	4.00E-03	+	4.88E-03	+	2.00E-03	+	4.88E-03	+
F5	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
F6	1.00	=	4.88E-03	+	4.88E-03	+	2.00E-03	+	4.88E-03	+
F7	7.00E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
F8	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
F9	1.00	=	2.00E-03	+	4.88E-03	+	4.00E-03	+	4.88E-03	+
F10	1.00	=	4.00E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
F11	1.00	=	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.00E-03	+
F12	4.88E-03	+	4.88E-03	+	4.88E-03	+	7.00E-03	+	4.88E-03	+
F13	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
F14	7.88E-03	+	1.39E-01	=	4.88E-03	+	7.88E-03	+	7.88E-03	+
F15	1.10E-02	+	4.88E-03	+	4.88E-03	+	1.00E-02	+	4.88E-03	+
F16	7.88E-03	+	7.88E-03	+	7.88E-03	+	7.88E-03	+	7.88E-03	+
F17	7.88E-03	+	7.00E-03	+	7.00E-03	+	7.88E-03	+	7.88E-03	+
F18	7.88E-03	+	7.00E-03	+	7.00E-03	+	7.88E-03	+	7.00E-03	+
F19	7.88E-03	+	4.88E-03	+	4.88E-03	+	7.88E-03	+	4.88E-03	+
F20	7.88E-03	+	4.88E-03	+	4.88E-03	+	7.88E-03	+	4.88E-03	+
F21	7.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
F22	7.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
F23	7.88E-03	+	4.88E-03	+	4.88E-03	+	7.88E-03	+	4.88E-03	+
(+, − , =)	(19, 0, 4)		(22, 0, 1)		(23, 0, 0)		(23, 0, 0)		(23, 0, 0)	

Table 13: WSR test results between COARO and competitive algorithms for CEC2019 benchmark functions

cec_i	COARO *vs.* ARO		COARO *vs.* GWO		COARO *vs.* MVO		COARO *vs.* PSO		COARO *vs.* TSO	
	p-value	win	*p*-value	win	*p*-value	win	*p*-value	win	*p*-value	win
cec01	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
cec02	4.88E-03	+	4.88E-03	+	4.88E-03	+	3.00E-03	+	4.88E-03	+

(Continued)

Table 13 (continued)

cec_i	COARO *vs.* ARO		COARO *vs.* GWO		COARO *vs.* MVO		COARO *vs.* PSO		COARO *vs.* TSO	
	p-value	win	*p*-value	win	*p*-value	win	*p*-value	win	*p*-value	win
cec03	7.88E-03	+	7.00E-03	+	7.88E-03	+	7.88E-03	+	7.88E-03	+
cec04	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
cec05	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
cec06	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
cec07	7.40E-02	=	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
cec08	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
cec09	1.30E-02	+	4.88E-03	+	4.88E-03	+	7.00E-03	+	4.88E-03	+
cec10	7.00E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+	4.88E-03	+
(+, −, =)	(9, 0, 1)		(10, 0, 0)		(10, 0, 0)		(10, 0, 0)		(10, 0, 0)	

Except for the F9–F11 benchmark functions, the proposed COARO algorithm beat the ARO algorithm, according to the WSR statistical test results in Table 12. The *p*-values obtained for the F9-F11 indicate that the difference between COARO and ARO is insignificant. Moreover, the results of the COARO algorithm are significantly better than the GWO algorithm, except for the F14 function. The results of the COARO algorithm are better than the remaining three metaheuristic algorithms (MVO, PSO, and TSO) for all classical benchmark functions.

Except for the cec07 function, Table 13 shows that the proposed COARO algorithm performs better than the ARO algorithm. The *p*-values obtained for COARO and the other four competitive algorithms show that COARO performs better than the different algorithms.

Figs. 5 and 6 show the boxplots of the proposed COARO, ARO, GWO, MVO, PSO, and TSO algorithms for the classical and CEC2019 functions, respectively. The x-axis represents the metaheuristic algorithms being compared. Boxplots have been used to evaluate the distribution of results from the COARO, ARO, GWO, MVO, PSO, and TSO algorithms.

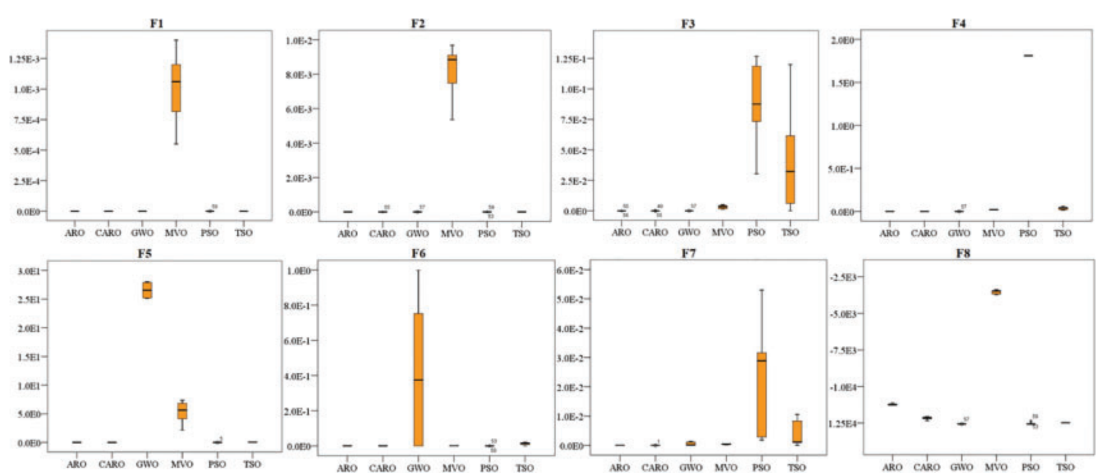

Figure 5: (Continued)

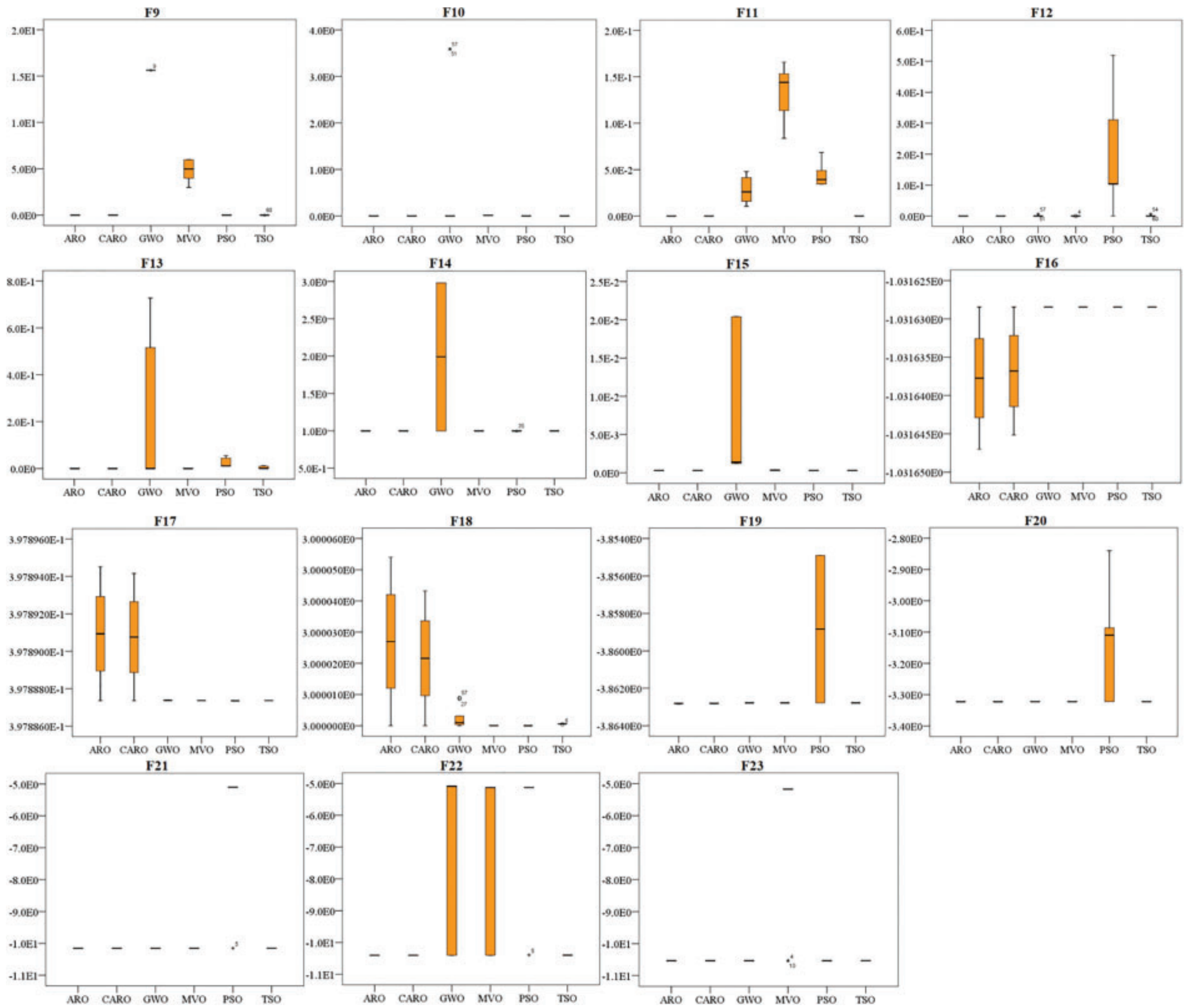

Figure 5: Boxplot of classical benchmark function (y-axis indicates values of fitness function)

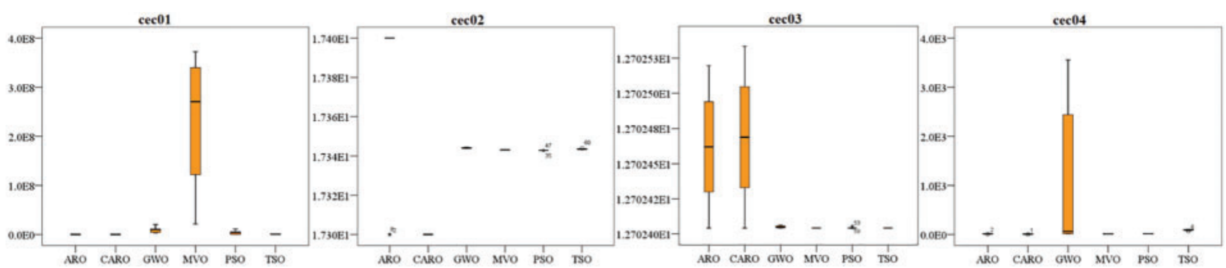

Figure 6: (Continued)

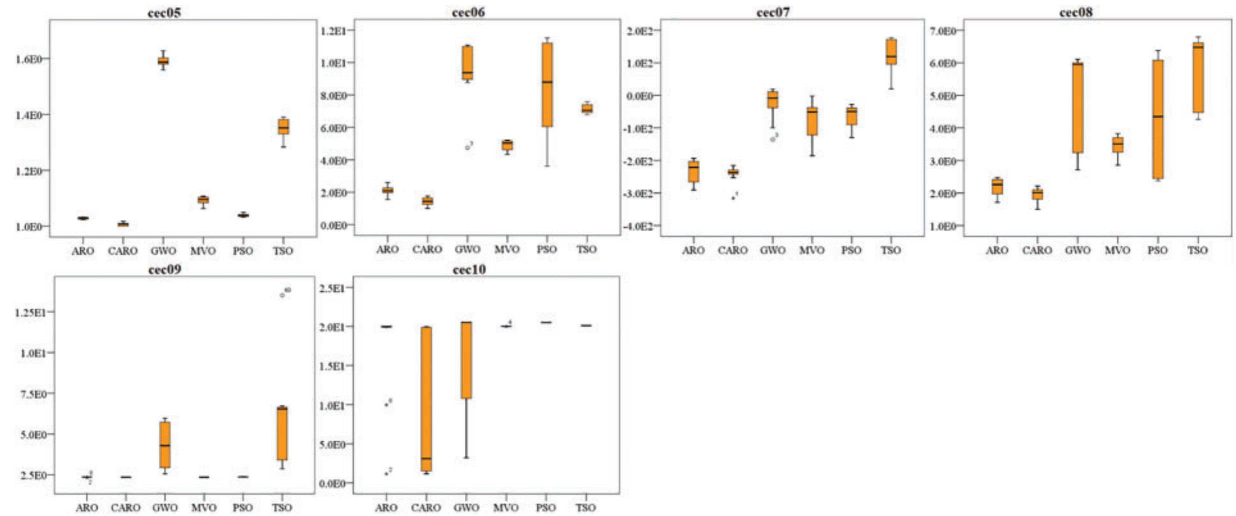

Figure 6: Boxplot of CEC2019 benchmark function (y-axis indicates values of fitness function)

OBL and CLS are strategies used to enhance the COARO algorithm. Considering the results of COARO in the Classic and CEC2019 benchmark functions, it can be concluded that the proposed COARO algorithm provides effectiveness and efficiency in determining global solutions by combining CLS and OBL techniques and adding ten chaotic maps to the optimization process. These improvements are due to CLS's advantage of nonlinear dynamics and advanced exploration and OBL's ability to increase diversity and bring closer to the global solution.

COARO has achieved promising results on the classic and CEC2019 benchmark functions. The proposed algorithm may exhibit different performances under different conditions. For example, in noisy environments, noise on the objective function may lead to incorrect fitness values, affecting the quality of the solutions obtained by the COARO algorithm. It can make it difficult for the algorithm to distinguish between real improvements and fluctuations caused by randomness due to noise. This can result in convergence to seemingly good but suboptimal solutions with random fluctuations. As a different example, in Dynamic Optimization Problems, the algorithm must be constantly adapted to meet new conditions since the optimization criteria, constraints, and variables change over time. Additionally, the algorithm needs to explore and exploit the changing environment efficiently.

4.2 Application of the COARO for Engineering Design Problems

This section uses six well-known engineering design problems to assess the COARO algorithm's problem-solving abilities. Detailed information and mathematical formulations on engineering design problems can be found in [23]. Each engineering issue's population and maximum iteration numbers are 30 and 500, respectively. For every engineering problem, all optimization algorithms have been performed 30 times. The results achieved by COARO for each engineering problem are compared with different metaheuristic algorithms.

4.2.1 Welded Beam

Reducing the cost of creating the welded beam is the aim of this minimization problem. The welded beam problem has four variables: weld thickness (h), attached part to bar length (l), bar height (t), and bar thickness (b).

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

Table 14 compares the ARO algorithm's outcomes and the suggested COARO algorithms applied to the welded beam.

Table 14: Comparison of the best optimum solution for welded beam

Algorithms	Optimum variables				Optimum cost
	x_1	x_2	x_3	x_4	
ARO	0.20573	3.470493	9.036624	0.20573	1.7248526
COARO1	0.20573	3.470489	9.036625	0.20573	1.7248524
COARO2	0.20573	3.470489	9.036624	0.20573	1.7248524
COARO3	0.20573	3.47049	9.036624	0.20573	1.7248525
COARO4	0.20573	3.47049	9.036624	0.20573	1.7248524
COARO5	0.20573	3.470491	9.036624	0.20573	1.7248525
COARO6	0.20573	3.470491	9.036624	0.20573	1.7248525
COARO7	0.20573	3.470491	9.036624	0.20573	1.7248525
COARO8	0.20573	3.47049	9.036624	0.20573	1.7248524
COARO9	0.20573	3.470489	9.036624	0.20573	**1.7248523**
COARO10	0.20573	3.470489	9.036623	0.20573	1.7248524

As seen from Table 14, the COARO9 algorithm achieved better performance than others for the welded beam with the decision variable values (0.20573, 3.470489, 9.036624, 0.20573) and the corresponding objective function value equal to 1.7248523. The statistical outcomes of the proposed COARO algorithms and ARO algorithm are given in Table 15.

Table 15: Statistical outcomes of algorithms for welded beam

Algorithms	Best	Worst	Mean	STD
ARO	1.7248526	2.2386529	1.7562230	0.1159667
COARO1	1.7248524	1.8255664	1.7307169	0.0224871
COARO2	1.7248524	1.7403046	1.7262629	0.0040104
COARO3	1.7248525	1.7983002	1.7313609	0.0178092
COARO4	1.7248524	1.7283227	**1.7254714**	0.0011201
COARO5	1.7248525	1.8488614	1.7375643	0.0346834
COARO6	1.7248525	1.7323581	1.7261905	0.0025579
COARO7	1.7248525	1.7342267	1.7257824	0.0022121
COARO8	1.7248524	1.7307182	1.7254864	0.0015111
COARO9	**1.7248523**	1.7342529	1.7257222	0.0023609
COARO10	1.7248524	1.7369788	1.7258341	0.0031111

The different criteria outcomes of the used algorithms are displayed in Table 15. Upon examining Table 15, it is evident that the COARO9 algorithm outperforms the other algorithms about the best value, whereas the COARO4 algorithm outperforms the others regarding the mean value. Fig. 7a shows the convergence graphs of the ARO algorithm on the welded beam and proposed COARO algorithms.

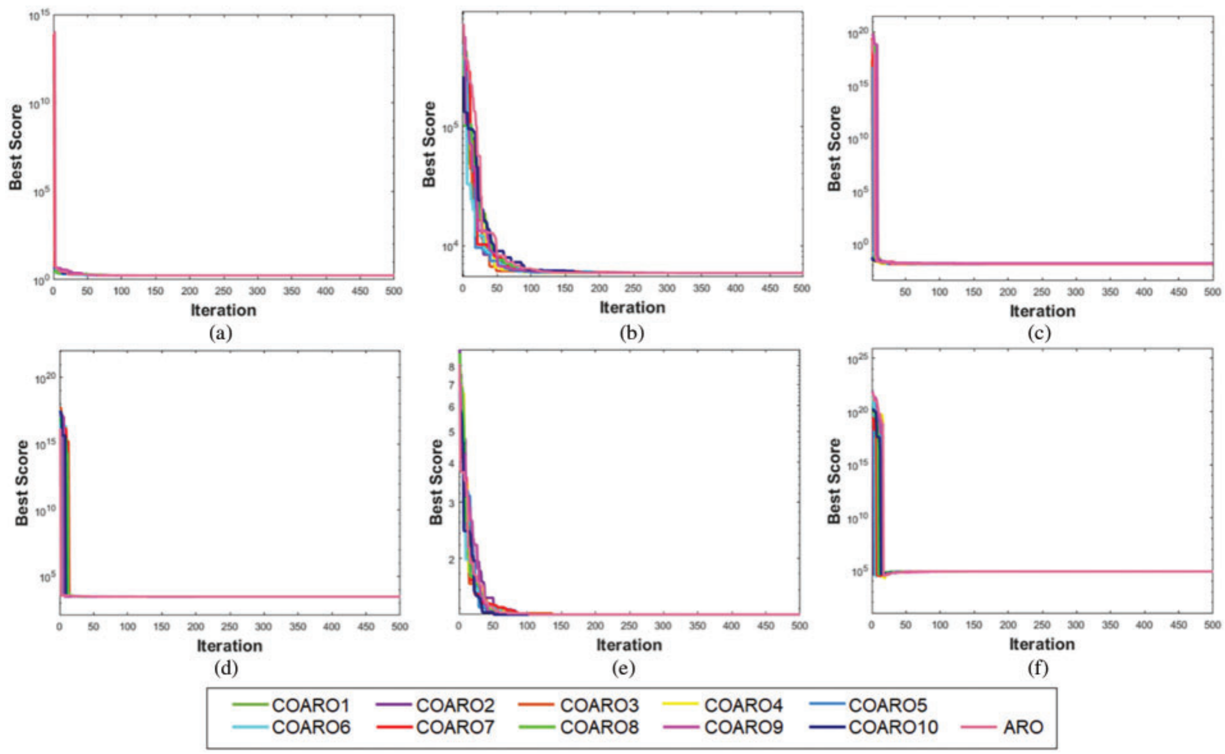

Figure 7: Convergence graphs of engineering design problems (a) Welded beam (b) Pressure vessel (c) Tension/Compression spring (d) Speed reducer (e) Cantilever beam (f) Rolling element bearing

4.2.2 Pressure Vessel

This minimization problem aims to reduce the total cost of a cylindrical vessel, including the cost of materials, welding, and forming. The problem has four variables: head thickness (T_h), length of the cylindrical vessel without head (L), shell thickness (T_s), and internal radius (R).

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

Table 16 compares the pressure vessel results using the ARO and proposed COARO algorithms.

As seen from Table 16, the COARO8 algorithm achieved better performance than others for the pressure vessel with the decision variable values (0.778177, 0.384653, 40.32003, 199.9943) and the corresponding objective function value equal to 5885.346358. The statistical outcomes of the proposed COARO algorithms and ARO algorithm are given in Table 17.

Table 17 displays the different criteria outcomes of the used algorithms. Upon closer inspection, the COARO8 algorithm outperforms the others in terms of best and mean values. Fig. 7b shows the convergence graphs of the ARO algorithm on the pressure vessel and the proposed COARO algorithm.

Table 16: Comparison of the best optimum solution for pressure vessel

Algorithms	Optimum variables				Optimum cost
	x_1	x_2	x_3	x_4	
ARO	0.778195	0.384694	40.3207	199.9867	5885.548110
COARO1	0.778225	0.384687	40.32253	199.96	5885.470900
COARO2	0.778188	0.384662	40.32051	199.9878	5885.393193
COARO3	0.778186	0.384691	40.3205	199.9909	5885.527732
COARO4	0.778201	0.384671	40.32127	199.9787	5885.439338
COARO5	0.778226	0.384682	40.32255	199.9596	5885.456881
COARO6	0.778233	0.384681	40.32273	199.9572	5885.485775
COARO7	0.778218	0.384673	40.32214	199.9679	5885.484743
COARO8	0.778177	0.384653	40.32003	199.9943	**5885.346358**
COARO9	0.778249	0.384687	40.32346	199.9469	5885.516485
COARO10	0.778234	0.384686	40.32293	199.9562	5885.521180

Table 17: Statistical outcomes of algorithms for pressure vessel

Algorithms	Best	Worst	Mean	STD
ARO	5885.54811	6457.424141	6200.877104	108.9598287
COARO1	5885.4709	5933.895679	5907.888281	22.6023744
COARO2	5885.393194	5927.311074	5903.619229	18.33119678
COARO3	5885.527732	5906.954286	5895.45643	9.869452674
COARO4	5885.439339	5907.11665	5894.914793	9.489448966
COARO5	5885.456882	5908.000954	5895.707967	10.22962712
COARO6	5885.485775	5949.080407	5914.555475	29.30144986
COARO7	5885.484744	5907.833449	5895.513748	9.97289147
COARO8	**5885.346358**	5890.513737	**5888.032798**	2.236531811
COARO9	5885.516486	5905.219431	5894.772226	9.243446177
COARO10	5885.52118	5973.951426	5921.299059	40.3021418

4.2.3 Tension/Compression Spring

This minimization problem aims to reduce the weight of this engineering problem. Three variables are involved in this problem: the number of active coils (P), wire diameter (d), and mean coil diameter (D).

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

Table 18 compares the outcomes produced on the tension/compression spring using the ARO and proposed COARO algorithms.

Table 18: Comparison of the best optimum solution for the tension/compression spring

Algorithms	Optimum variables			Optimum cost
	x_1	x_2	x_3	
ARO	0.0519	0.3617	11.3611	0.01266602
COARO1	0.0519	0.3617	11.4450	0.01266602
COARO2	0.0519	0.3617	11.1170	0.01266602
COARO3	0.0519	0.3617	11.1065	0.01266602
COARO4	0.0519	0.3617	11.0620	0.01266602
COARO5	0.0519	0.3617	10.7157	0.01266602
COARO6	0.0519	0.3617	11.3386	0.01266602
COARO7	0.0519	0.3617	10.6296	0.01266602
COARO8	0.0519	0.3617	10.6856	0.01266602
COARO9	0.0519	0.3617	11.1136	0.01266602
COARO10	0.0519	0.3617	10.7946	0.01266602

Table 18 shows the best ARO results, and the proposed COARO algorithms are relatively similar. Table 19 gives the statistical outcomes of the proposed COARO algorithms and the ARO algorithm.

Table 19: Statistical outcomes of algorithms for tension/compression spring

Algorithms	Best	Worst	Mean	STD
ARO	0.01266602	0.01272711	0.01268241	2.20E-05
COARO1	0.01266602	0.01268043	0.01266951	4.17E-06
COARO2	0.01266602	0.01268268	0.01266913	4.19E-06
COARO3	0.01266602	0.01267328	**0.01266783**	2.24E-06
COARO4	0.01266602	0.01272711	0.01269347	2.95E-05
COARO5	0.01266602	0.01418224	0.01277350	3.34E-04
COARO6	0.01266602	0.01532719	0.01302410	6.50E-04
COARO7	0.01266602	0.01355402	0.01276457	2.05E-04
COARO8	0.01266602	0.01318114	0.01271530	1.13E-04
COARO9	0.01266602	0.01270265	0.01267477	1.02E-05
COARO10	0.01266602	0.01272711	0.01269242	2.86E-05

Table 19 displays the different criteria outcomes of the employed algorithms. According to Table 19, the COARO3 algorithm is more successful regarding mean value. The convergence graphs of the proposed COARO algorithms and the ARO algorithm on the tension/compression spring are illustrated in Fig. 7c.

4.2.4 Speed Reducer

The main goal of the speed reducer is to reduce its weight. The problem has seven variables, such as the face width (x_1), the tooth's module (x_2), the number of teeth on the pinion (x_3), the diameter of the first shaft (x_6), the diameter of the second shaft (x_7), the length of the first shaft between bearings (x_4), and the length of the second shaft between bearings (x_5).

Table 20 compares the outcomes produced by the ARO algorithm and the proposed COARO algorithms on the speed reducer.

As seen from Table 20, the COARO3 algorithm achieved better performance than others for the speed reducer with the decision variable values (3.5, 0.7, 17, 7.3, 7.71532, 3.350215, 5.286654) and the corresponding objective function value equal to 2994.471066. The statistical outcomes of the proposed COARO algorithms and ARO algorithm are given in Table 21.

Table 20: Comparison of the best optimum solution for speed reducer

Algorithms	Optimum variables							Optimum cost
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
ARO	3.5	0.7	17	7.300001	7.715326	3.350215	5.286654	2994.471637
COARO1	3.5	0.7	17	7.300005	7.71532	3.350215	5.286655	2994.471535
COARO2	3.5	0.7	17	7.300008	7.715324	3.350215	5.286654	2994.471602
COARO3	3.5	0.7	17	7.3	7.71532	3.350215	5.286654	**2994.471066**
COARO4	3.5	0.7	17	7.300009	7.715321	3.350215	5.286655	2994.471325
COARO5	3.5	0.7	17	7.300002	7.715324	3.350215	5.286655	2994.471361
COARO6	3.5	0.7	17	7.3	7.71532	3.350215	5.286655	2994.471474
COARO7	3.5	0.7	17	7.300005	7.715322	3.350215	5.286655	2994.471581
COARO8	3.5	0.7	17	7.300001	7.715324	3.350215	5.286655	2994.471490
COARO9	3.5	0.7	17	7.300002	7.715322	3.350215	5.286654	2994.471593
COARO10	3.5	0.7	17	7.300003	7.71532	3.350215	5.286655	2994.471456

Table 21 displays the different criteria outcomes of the employed algorithms. The COARO3 algorithm beats the others in terms of greatest value, while the COARO5 algorithm beats the others in terms of mean value. The proposed COARO algorithm and the ARO algorithm's convergence curves on the speed reducer are displayed in Fig. 7d.

4.2.5 Cantilever Beam

This problem uses a hollow square portion to decrease the cantilever beam's total weight. This problem has five decision variables: x_1, x_2, x_3, x_4, and x_5.

Table 22 compares the outcomes produced on the cantilever beam using the ARO and proposed COARO algorithms.

As seen from Table 22, the COARO8 algorithm achieved better performance than others for the cantilever beam with the decision variable values (6.01421, 5.3116, 4.4953, 3.5037, 2.1488) and the corresponding objective function value equal to 1.339957. The statistical outcomes of the proposed COARO algorithms and ARO algorithm are given in Table 23.

Table 21: Statistical outcomes of algorithms for speed reducer

Algorithms	Best	Worst	Mean	STD
ARO	2994.471637	3001.328499	2996.136259	1.973584069
COARO1	2994.471535	2996.872811	2994.622798	0.530218617
COARO2	2994.471602	3010.193327	2996.095656	3.766608929
COARO3	**2994.471066**	2995.512821	2994.524031	0.232769897
COARO4	2994.471325	2995.470134	2994.608258	0.262437814
COARO5	2994.471361	2994.627357	**2994.518515**	0.054887027
COARO6	2994.471474	2996.567455	2994.680377	0.469851793
COARO7	2994.471581	2995.352142	2994.550356	0.197369751
COARO8	2994.471490	2999.413216	2994.947779	1.252926709
COARO9	2994.471593	2996.014319	2994.616967	0.35696662
COARO10	2994.471456	2995.996723	2994.747002	0.4681583

Table 22: Comparison of the best optimum solution for the cantilever beam

Algorithms	Optimum variables					Optimum cost
	x_1	x_2	x_3	x_4	x_5	
ARO	6.0135	5.3185	4.4892	3.4977	2.1547	1.339960
COARO1	6.0175	5.3082	4.4978	3.5030	2.1472	1.339960
COARO2	6.0118	5.3069	4.4967	3.5088	2.1494	1.339959
COARO3	6.0134	5.3069	4.4929	3.5066	2.1538	1.339959
COARO4	6.0164	5.3079	4.4975	3.5049	2.1469	1.339959
COARO5	6.0118	5.3144	4.4895	3.5042	2.1538	1.339958
COARO6	6.0238	5.3117	4.4931	3.4958	2.1493	1.339961
COARO7	6.0208	5.3084	4.4879	3.5055	2.1511	1.339958
COARO8	6.01421	5.3116	4.4953	3.5037	2.1488	**1.339957**
COARO9	6.0169	5.3026	4.4962	3.5069	2.1512	1.339958
COARO10	6.0204	5.3144	4.4932	3.4961	2.1497	1.339959

Table 23 displays the different criteria outcomes of the used algorithms. Examining Table 23 reveals that the COARO8 algorithm outperforms the others regarding the best value, while the COARO1 method outperforms the others regarding the mean value. Fig. 7e shows the convergence graphs of the ARO algorithm on the cantilever beam and the proposed COARO algorithm.

4.2.6 Rolling Element Bearing

The rolling element bearing problem aims to maximize the load capacity by considering ten design variables and nine constraints. Table 24 compares the rolling element-bearing results using the ARO and the proposed COARO algorithms.

Table 23: Statistical outcomes of algorithms for the cantilever beam

Algorithms	Best	Worst	Mean	STD
ARO	1.339960	1.340886	1.340277	0.000338334
COARO1	1.339960	1.340963	**1.340225**	0.000299814
COARO2	1.339959	1.342045	1.340448	0.000557866
COARO3	1.339959	1.340667	1.340252	0.000214615
COARO4	1.339959	1.341281	1.340414	0.000393155
COARO5	1.339958	1.340685	1.340247	0.000219441
COARO6	1.339961	1.341184	1.340453	0.000383546
COARO7	1.339958	1.341434	1.340274	0.000314256
COARO8	**1.339957**	1.341051	1.340291	0.000314593
COARO9	1.339958	1.341336	1.340316	0.000341019
COARO10	1.339959	1.341356	1.340329	0.000319468

Table 24: Comparison of the best optimum solution for rolling element bearing

Algorithms	Optimum variables										Optimum cost
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	
ARO	125.719	21.42558	11.40363	0.515	0.515115	0.46908	0.6433	0.300001	0.069386	0.70508	85549.165638
COARO1	125.719	21.42558	10.92733	0.515	0.515014	0.489367	0.681997	0.300001	0.04572	0.611225	85549.194537
COARO2	125.719	21.42559	11.24849	0.515	0.515133	0.472623	0.655669	0.3	0.025772	0.690266	85549.183172
COARO3	125.7191	21.42559	10.98129	0.515	0.515092	0.47815	0.654484	0.3	0.058815	0.640921	85549.177948
COARO4	125.719	21.42558	10.66787	0.515	0.515008	0.455802	0.670355	0.3	0.062129	0.664941	85549.178314
COARO5	125.719	21.42559	10.59076	0.515	0.515147	0.45075	0.692508	0.3	0.070533	0.692081	85549.190693
COARO6	125.7191	21.42559	11.42134	0.515	0.515021	0.436863	0.652134	0.3	0.042243	0.699072	**85549.216495**
COARO7	125.719	21.42559	11.24861	0.515	0.515051	0.427389	0.640772	0.3	0.064626	0.627065	85549.188577
COARO8	125.719	21.42558	11.23125	0.515	0.515026	0.482662	0.63974	0.3	0.080328	0.710014	85549.172942
COARO9	125.719	21.42558	10.58058	0.515	0.515056	0.426017	0.666648	0.300001	0.028812	0.682922	85549.166227
COARO10	125.719	21.42558	11.30117	0.515	0.515004	0.446034	0.651252	0.300001	0.085675	0.704554	85549.164527

As seen from Table 24, the COARO6 algorithm achieved better performance than other algorithms for the rolling element bearing with the decision variable values (125.7191, 21.42559, 11.42134, 0.515, 0.515021, 0.436863, 0.652134, 0.3, 0.042243, 0.699072) and the corresponding objective function value equal to 85549.216495. The statistical outcomes of the proposed COARO algorithms and ARO algorithm are given in Table 25.

The different criteria outcomes of the employed algorithms are displayed in Table 25. Table 25 clearly shows that the COARO3 algorithm performs better than the others in terms of mean value, while the COARO6 algorithm performs better than the others in terms of best value. Fig. 7f shows the convergence graphs of the ARO algorithm on the rolling element bearing and the proposed COARO algorithms.

Table 25: Statistical outcomes of algorithms for rolling element-bearing

Algorithms	Best	Worst	Mean	STD
ARO	85549.165638	77416.35105	83008.19849	3740.059887
COARO1	85549.19454	77070.27012	81460.99637	4194.729804
COARO2	85549.18317	77056.97501	82089.23191	4051.852567
COARO3	85549.17795	77179.73319	**83187.4337**	3531.931723
COARO4	85549.17831	77022.02841	82970.94662	3782.111316
COARO5	85549.19069	77331.14936	81475.93122	4159.762648
COARO6	**85549.2165**	77247.46207	82513.00518	3868.355991
COARO7	85549.18858	76909.03228	82264.89189	4127.295742
COARO8	85549.17294	77431.22899	82699.92899	3956.314121
COARO9	85549.16623	77439.87626	81499.18125	4155.019965
COARO10	85549.16453	77367.68304	82657.8327	3944.527031

4.3 Analysis of COARO for Engineering Design Problems with Other Metaheuristic Algorithms

Many metaheuristic algorithms have found the best solution for the welded beam. The best cost and relevant decision variables obtained from the proposed COARO algorithm and other compared metaheuristic algorithms for the welded beam are given in Table 26.

Table 26: Best solutions obtained for welded beam

Algorithms	Optimum variables				Optimum cost
	x_1	x_2	x_3	x_4	
HGSO [24]	0.2054	3.4476	9.0269	0.2060	1.7260
GWO [25]	0.2057	3.4754	9.0369	0.2062	1.726995
DE [26]	0.2065	3.6359	10	0.2032	1.836250
AOA [27]	0.2015	3.4707	10	0.2025	1.7837
GSA [28]	0.1471	5.4907	10	0.2177	2.172858
BWOA [29]	0.1552	5.6462	8.9621	0.2114	1.9406383
SHO [30]	0.2056	3.4748	9.0358	0.2058	1.725661
MVO [31]	0.2056	3.4721	9.0409	0.2057	1.725472
RSA [32]	0.1964	3.5366	9.9520	0.2182	1.9831072
HHO [33]	0.2040	3.5311	9.0275	0.2061	1.7319905
ARO	0.2057	3.4705	9.0366	0.2057	1.7248526
COARO9	0.2057	3.4705	9.0366	0.2057	**1.7248523**

Upon examining Table 26, it can be noted that the COARO9 method performs better than both ARO and the competitive optimization techniques. The literature has employed a variety of metaheuristic algorithms to determine the best solution for the pressure vessel. ARO and the proposed COARO algorithm were compared with ten different algorithms in the literature. Table 27 presents the optimal cost and pertinent choice variables derived from the proposed COARO algorithm and other comparative metaheuristic algorithms for the pressure vessel. The potential applications of the COARO algorithm in various fields, such as image processing, wireless sensor networks, and decision support systems, are inspiring.

Table 27: Best solutions obtained for pressure vessel

Algorithms	Optimum variables				Optimum cost
	x_1	x_2	x_3	x_4	
SHO [34]	0.7782	0.3847	40.3223	199.9623	5885.4926
SMA [35]	0.7931	0.3932	40.6711	196.2178	5994.1857
MSCA [36]	0.7806	0.3918	40.4191	198.9641	5917.5098
ABC [37]	0.8125	0.4375	42.0984	176.6366	6059.7143
EO [38]	0.8125	0.4375	42.0984	176.6366	6059.7143
MBA [39]	0.7802	0.3856	40.4292	198.4964	5889.3216
ALO [40]	0.7816	0.3863	40.4986	197.5273	5891.3929
SSA [41]	0.8176	0.4179	41.7494	183.5727	6137.3725
CS [42]	0.8125	0.4375	42.0984	176.6366	6059.7143
GSA [28]	1.0858	0.9496	49.3452	169.4874	11550.2976
ARO	0.7782	0.38467	40.3207	199.9867	5885.5481
COARO8	0.778177	0.384653	40.32003	199.9943	**5885.346358**

Upon examining Table 27, it's clear that the COARO8 method competes favorably with ARO and other competitive optimization techniques. The literature has employed various metaheuristic methods to find the best tension/compression spring solution. Nine different approaches from the literature were compared with ARO and the proposed COARO algorithm. Table 28 presents the optimal cost and relevant choice factors for the tension/compression spring derived from the proposed COARO algorithm and other comparative metaheuristic algorithms.

Table 28: Best solutions obtained for tension/compression spring

Algorithms	Optimum variables			Optimum cost
	x_1	x_2	x_3	
TLBO [43]	0.0515	0.3523	11.5559	0.012671
PSO [44]	0.0517	0.3576	11.2445	0.0126747
WOA [19]	0.0512	0.3452	12.0040	0.0126763
DE [45]	0.0516	0.3547	11.4108	0.0126702
GOA [46]	0.0526	0.3853	9.5862	0.012493
BOA [47]	0.0511	0.3415	12.3269	0.012789
HS [48]	0.0511	0.3499	12.0764	0.012671
ABC [49]	0.0527	0.3819	9.9519	0.01268595

(Continued)

Table 28 (continued)

Algorithms	Optimum variables			Optimum cost
	x_1	x_2	x_3	
GA [50]	0.0515	0.3517	11.6322	0.0127048
ARO	0.0519	0.3617	11.3611	**0.01266602**
COARO3	0.0519	0.3617	11.1065	**0.01266602**

Upon examining Table 28, it can be noted that the algorithms for COARO3 and ARO performed better than those for competitive optimization. Numerous metaheuristic algorithms have been used in the literature to find the optimal solution for the speed reducer. Nine different approaches from the literature were compared with ARO and the proposed COARO algorithm. The best cost and relevant decision variables obtained from the proposed COARO algorithm and other compared metaheuristic algorithms for the speed reducer are given in Table 29.

Table 29: Best solutions obtained for speed reducer

Algorithms	Optimum variables							Optimum cost
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
CS [42]	3.5015	0.7	17	7.6050	7.8181	3.3520	5.2875	3000.9810
MVO [31]	3.5088	0.7	17	7.3928	7.8160	3.3581	5.2868	3002.9280
GSA [28]	3.6000	0.7	17	8.3	7.8	3.3697	5.2892	3051.120
SSA [51]	3.5103	0.7	17	8.35	7.8	3.3622	5.2877	3067.561
GWO [25]	3.5005	0.7	17.003	7.3062	7.8781	3.3508	5.2873	2999.3605
WSA [52]	3.500	0.7	17	7.3	7.8	3.3502	5.2867	2996.348225
AO [53]	3.5021	0.7	17	7.3099	7.7476	3.3641	5.2994	3007.7328
CSA [54]	3.48	0.7	17	8.26	7.95	3.34	5.28	2997.70741
WOA [19]	3.48	0.7	17	7.30	7.80	3.34	5.28	2997.98729
SCA [55]	3.5087	0.7	17	7.3	7.8	3.4610	5.2892	3030.563
ARO	3.500001	0.7	17	7.300001	7.715326	3.350215	5.286654	2994.471637
COARO3	3.5	0.7	17	7.3	7.71532	3.350215	5.286654	**2994.471066**

Upon closer inspection of Table 30, it's evident that the COARO3 algorithms outperform ARO and other competitive optimization algorithms. The literature has employed various metaheuristic methods to find the best cantilever beam solution. ARO and the proposed COARO algorithm were compared with four different algorithms in the literature. Table 30 gives the best cost and relevant decision variables obtained from the proposed COARO algorithm and other compared metaheuristic algorithms for the cantilever beam.

Upon closer inspection of Table 30, it can be shown that the COARO8 algorithms perform better than both ARO and the competitive optimization algorithms. The literature has employed various metaheuristic methods to determine the rolling element bearing the ideal solution. ARO and the proposed COARO algorithm were compared with five different algorithms in the literature. The best cost and relevant decision variables obtained from the proposed COARO algorithm and other compared metaheuristic algorithms for the rolling element bearing are given in Table 31.

Table 30: Best solutions obtained for cantilever beam

Algorithms	Optimum variables					Optimum cost
	x_1	x_2	x_3	x_4	x_5	
CSO [54]	6.7628	5.1583	5.6537	2.9279	1.8854	1.397024
WOA [19]	5.1261	5.6188	5.0952	3.9329	2.3219	1.378732
MVO [31]	6.0239	5.3060	4.4950	3.49602	2.1527	1.3399595
GSA [28]	5.6052	4.9553	5.6619	3.1959	3.2026	1.41
ARO	6.0135	5.3185	4.4892	3.4977	2.1547	1.339960
COARO8	6.01421	5.3116	4.4953	3.5037	2.1488	**1.339957**

Table 31: Best solutions obtained for rolling element bearing

Algorithms	Optimum Variables										Optimum cost
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	
MBA [39]	125.7153	21.4233	11.0000	0.515	0.5150	0.48881	0.627829	0.300149	0.097305	0.646095	85535.6911
TLBO [43]	125.7191	21.42559	11	0.515	0.515	0.424266	0.633948	0.3	0.068858	0.799498	81859.74
SCA [55]	125	21.14834	10.92928	0.515	0.515	0.5	0.7	0.3	0.02778	0.62912	83431.117
HPO [23]	125.000	21.8750	10.7770	0.515	0.515	0.40	0.7	0.3	0.0290	0.6	83918.4925
HHO [33]	125.000	21.0000	11.09207	0.515	0.515	0.40	0.6	0.3	0.050474	0.6	83011.88329
ARO	125.719	21.42558	11.40363	0.515	0.515115	0.46908	0.6433	0.300001	0.069386	0.70508	85549.165638
COARO6	125.7191	21.42559	11.42134	0.515	0.515021	0.436863	0.652134	0.3	0.042243	0.699072	**85549.216495**

Upon closer inspection of Table 31, it can be shown that the COARO6 algorithms perform better than both ARO and the competitive optimization algorithms. In addition to all the evaluations in this section, the proposed COARO algorithm performs better for six engineering design problems than other metaheuristic algorithms.

4.4 Real Application

Performance of BCOARO for Feature Selection in Breast Cancer Problem. The BCOARO algorithm shows promising results in feature selection for the breast cancer dataset, instilling hope for its potential in real-world applications. This section uses the breast cancer dataset to analyze the performance of binary COARO algorithms. This analysis's main objective is to assess the proposed continuous COARO algorithm's (BCOARO) binary version's applicability in the real world and its efficiency in feature selection for the breast cancer dataset. It is also to compare its performance with different algorithms on this dataset. The breast cancer dataset used to consider the BCOARO algorithm performance contains a total of 569 data, 212 of which are malignant and 357 of which are benign [56]. It consists of 31 features in Table 32 in the breast cancer dataset.

Table 32: The attributes of the breast cancer dataset

1	Diagnosis	12	Radius_se	23	Texture_worst
2	Radius_mean	13	texture_se	24	perimeter_worst
3	texture_mean	14	perimeter_seperimeter_se	25	area_worst
4	perimeter_mean	15	area_se	26	smoothness_worst
5	area_mean	16	smoothness_se	27	compactness_worst
6	smoothness_mean	17	compactness_se	28	concavity_worst
7	compactness_mean	18	concavity_se	29	concave points_worst
8	concavity_mean	19	concave points_se	30	symmetry_worst
9	concave points_mean	20	symmetry_se	31	fractal_dimension_worst
10	symmetry_mean	21	fractal_dimension_se		
11	fractal_dimension_mean	22	radius_worst		

The results obtained for the breast cancer dataset with the BCOARO algorithm are compared with those of the ARO, WOA, BGWO, BASO, HGSO, and BHHO algorithms. For a fair evaluation, the maximum iteration number was set to 100, and the number of populations was 30 for all algorithms. Table 33 displays the accuracy and fitness values acquired using these algorithms.

Table 33: Comparison of BCOARO with other algorithms for breast cancer dataset

Algorithms	Accuracy (%)	Fitness value
BCOARO1	98.40	0.01170
BCOARO2	96.46	0.03539
BCOARO3	98.23	0.01769
BCOARO4	95.58	0.04425
BCOARO5	97.35	0.02655
BCOARO6	97.35	0.02655
BCOARO7	99.38	0.00231
BCOARO8	98.23	0.01769
BCOARO9	98.23	0.01769
BCOARO10	99.12	0.00885
BARO	96.46	0.03539
WOA	97.06	0.02941
BGWO	97.35	0.02655
BASO	98.23	0.01986
HGSO	98.23	0.01770
BHHO	97.35	0.02655

A graphical representation of accuracy values is presented in Fig. 8. According to Fig. 8, BCOARO7 achieved the highest accuracy value, BCOARO10 had the second highest accuracy value, and BCOARO4 showed the lowest. According to Table 33, the BCOARO7 algorithm has a superior result to other algorithms in terms of 99.38% accuracy. Likewise, BCOARO7, which uses a sine chaotic map, is more successful than other compared algorithms with a fitness value of 0.00231. Since the sine map introduces a specific form of nonlinearity, unlike other chaotic maps, it exhibits different behavioral patterns and obtains different statistical results. Therefore, the 99.38% accuracy and 0.00231 fitness value obtained from the experimental results showed that BCOARO7 demonstrated an outlier and practical value and that its performance was replicability. The second-best algorithm for feature selection in the breast cancer dataset is the BCOARO10 algorithm, with an accuracy value of 99.12%. The fitness function value of this algorithm is 0.00885.

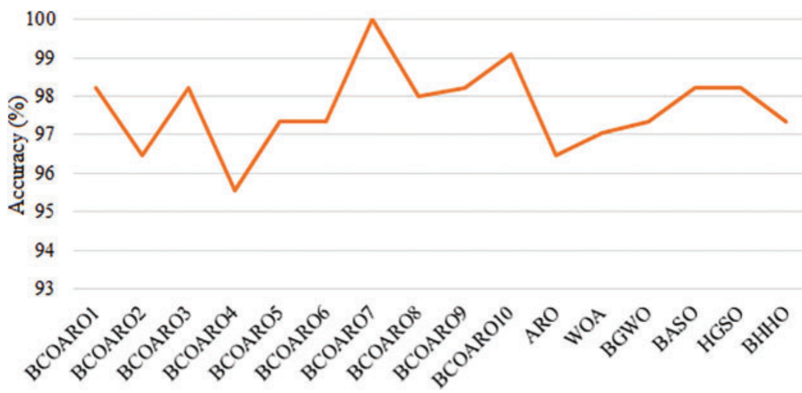

Figure 8: Comparison of accuracy values for breast cancer dataset

5 Conclusion and Future Works

The detour foraging and random hiding habits of rabbits in the wild inspired the biology-based metaheuristic algorithm known as the ARO algorithm. This paper proposed an enhanced COARO by integrating the concept of CLS and OBL and incorporating ten different chaotic maps into the optimization process of ARO. CLS ensures that solution quality increases by directing the local search around the global best solution. Adding OBL to the algorithm improves the initial population and enhances the problem of getting stuck in local minima. The concept of chaos is a very effective technique to overcome the issues of optimization algorithms, such as local optimum traps, early convergence, and inefficient search. The findings showed that the COARO algorithm outperformed ARO in reaching the optimal solution for 33 benchmark functions, including unimodal, multimodal, fixed-dimension multimodal, and CEC2019 functions. Furthermore, the literature compares the performance of the COARO algorithm with four popular metaheuristic algorithms: GWO, MVO, PSO, and TSO. As a result of the comparison, the COARO algorithm achieved better results than competitive algorithms. We evaluated the proposed COARO algorithm's performance on six engineering design problems. It is superior to other algorithms for engineering design problems. Additionally, a binary version of the COARO algorithm was developed, and its performance was evaluated as a feature selection method for an actual application. As a result of the analysis performed on the breast cancer dataset, it was observed that the results obtained with the BCOARO algorithm were promising.

Although it has the advantage of efficiently managing different types of applications, the proposed COARO algorithm has limitations. First of all, the first limitation that all metaheuristic methods have is that, according to the No-Free-Lunch theorem, there is the possibility of developing newer algorithms that achieve more effective performance than COARO. Another limitation is that although the COARO algorithm achieved successful performance for the optimization problems examined in this study, it does not guarantee successful performance for different optimization problems. This means that the COARO algorithm may still need changes and improvements. The proposed COARO algorithm can be applied to future research fields, such as image processing, wireless sensor networks, decision support systems, the Internet of Things, and logistics. Further study may focus on creating parallel or multi-objective versions of the proposed algorithm to address various optimization issues.

Acknowledgement: The authors thank the reviewers for their valuable comments, which significantly improved this paper.

Funding Statement: This article was funded by Firat University Scientific Research Projects Management Unit for the scientific research project of Feyza Altunbey Özbay, numbered MF.23.49.

Author Contributions: The authors confirm their contribution to the paper as follows: Methodology: Feyza Altunbey Özbay, Erdal Özbay and Farhad Soleimanian Gharehchopogh; Software: Feyza Altunbey Özbay, Erdal Özbay and Farhad Soleimanian Gharehchopogh; Validation: Feyza Altunbey Özbay; Formal analysis and investigation: Feyza Altunbey Özbay, Erdal Özbay and Farhad Soleimanian Gharehchopogh; Writing—original draft preparation: Feyza Altunbey Özbay, Erdal Özbay and Farhad Soleimanian Gharehchopogh; Writing—review and editing: Farhad Soleimanian Gharehchopogh; Visualization: Feyza Altunbey Özbay, Erdal Özbay and Farhad Soleimanian Gharehchopogh; Supervision: Feyza Altunbey Özbay, Erdal Özbay and Farhad Soleimanian Gharehchopogh. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: We use the available breast cancer dataset for evaluation: https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset/ (accessed on 24 May 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that there are no conflicts of interest to report regarding the present study.

References

1. Gharehchopogh FS, Gholizadeh H. A comprehensive survey: Whale Optimization Algorithm and its applications. Swarm Evol Comput. 2019;48(8):1–24. doi:10.1016/j.swevo.2019.03.004.
2. Özbay FA. A modified seahorse optimization algorithm based on chaotic maps for solving global optimization and engineering problems. Eng Sci Technol, Int J. 2023;41(1):101408. doi:10.1016/j.jestch.2023.101408.
3. Gharehchopogh FS, Shayanfar H, Gholizadeh H. A comprehensive survey on symbiotic organisms search algorithms. Artif Intell Rev. 2020;53(3):2265–312. doi:10.1007/s10462-019-09733-4.
4. Kiani F, Anka FA, Erenel F. PSCSO: enhanced sand cat swarm optimization inspired by the political system to solve complex problems. Adv Eng Softw. 2023;178(5):103423. doi:10.1016/j.advengsoft.2023.103423.
5. Shajin FH, Aruna Devi B, Prakash NB, Sreekanth GR, Rajesh P. Sailfish optimizer with Levy flight, chaotic and opposition-based multi-level thresholding for medical image segmentation. Soft Comput. 2023;27(17):12457–82. doi:10.1007/s00500-023-07891-w.
6. Kiani F, Seyyedabbasi A, Aliyev R, Shah MA, Gulle MU. 3D path planning method for multi-UAVs inspired by grey wolf algorithms. J Int Technol. 2021;22(4):743–55. doi:10.53106/160792642021072204003.

7. Özbay E. An active deep learning method for diabetic retinopathy detection in segmented fundus images using artificial bee colony algorithm. Artif Intell Rev. 2023;56(4):3291–318. doi:10.1007/s10462-022-10231-3.
8. Seyyedabbasi A, Dogan G, Kiani F. HEEL: a new clustering method to improve wireless sensor network lifetime. IET Wirel Sens Syst. 2020;10(3):130–6. doi:10.1049/iet-wss.2019.0153.
9. Bhavya R, Elango L. Ant-inspired metaheuristic algorithms for combinatorial optimization problems in water resources management. Water. 2023;15(9):1712. doi:10.3390/w15091712.
10. Xu S, Li W, Li L, Li T, Ma C. Crashworthiness design and multi-objective optimization for bio-inspired hierarchical thin-walled structures. Comput Model Eng Sci. 2022;131(2):929–47. doi:10.32604/cmes.2022.018964.
11. Nematzadeh S, Torkamanian-Afshar M, Seyyedabbasi A, Kiani F. Maximizing coverage and maintaining connectivity in WSN and decentralized IoT: an efficient metaheuristic-based method for environment-aware node deployment. Neural Comput Appl. 2023;35(1):611–41. doi:10.1007/s00521-022-07786-1.
12. Gonzalez-Sanchez B, Vega-Rodríguez MA, Santander-Jiménez S. A multi-objective butterfly optimization algorithm for protein-encoding. Appl Soft Comput. 2023;139(1):110269. doi:10.1016/j.asoc.2023.110269.
13. Trojovský P, Dehghani M. Migration algorithm: a new human-based metaheuristic approach for solving optimization problems. Comput Model Eng Sci. 2023;137(2):1695–730. doi:10.32604/cmes.2023.028314.
14. Wang L, Cao Q, Zhang Z, Mirjalili S, Zhao W. Artificial rabbits optimization: a new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell. 2022;114(4):105082. doi:10.1016/j.engappai.2022.105082.
15. Riad AJ, Hasanien HM, Turky RA, Yakout AH. Identifying the PEM fuel cell parameters using an artificial rabbit optimization algorithm. Sustainability. 2023;15(5):4625. doi:10.3390/su15054625.
16. Siddiqui SA, Gerini F, Ikram A, Saeed F, Feng X, Chen Y. Rabbit meat—production, consumption and consumers' attitudes and behavior. Sustainability. 2008;15(3):2008. doi:10.3390/su15032008.
17. Dobos P, Kulik LN, Pongrácz P. The amicable rabbit-interactions between pet rabbits and their caregivers based on a questionnaire survey. Appl Anim Behav Sci. 2023;260:105869. doi:10.1016/j.applanim.2023.105869.
18. Elshahed M, Tolba MA, El-Rifaie AM, Ginidi A, Shaheen A, Mohamed SA. An artificial rabbits' optimization to allocate PVSTATCOM for ancillary service provision in distribution systems. Mathematics. 2023;11(2):339. doi:10.3390/math11020339.
19. Seyedali M, Andrew L. The whale optimization algorithm. Adv in Eng Softw. 2016;95(12):51–67. doi:10.1016/j.advengsoft.2016.01.008.
20. Kiani F, Nematzadeh S, Anka FA, Findikli MA. Chaotic sand cat swarm optimization. Mathematics. 2023;11(10):2340. doi:10.3390/math11102340.
21. dos Santos Coelho L, Mariani VC. Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. Expert Syst Appl. 2008;34(3):1905–13. doi:10.1016/j.eswa.2007.02.002.
22. Tizhoosh HR. Opposition-based learning: a new scheme for machine intelligence. In: International Conference on Computational Intelligence for Modeling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, IEEE, (CIMCA-IAWTIC'06); 2005; Vienna, Austria. p. 695–701.
23. Naruei I, Keynia F, Sabbagh Molahosseini A. Hunter-prey optimization: algorithm and applications. Soft Comput. 2022;26(3):1279–314. doi:10.1007/s00500-021-06401-0.
24. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S. Henry gas solubility optimization: a novel physics-based algorithm. Future Gener Comput Syst. 2019;101(4):646–67. doi:10.1016/j.future.2019.07.015.
25. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw. 2014;69:46–61. doi:10.1016/j.advengsoft.2013.12.007.

26. Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim. 1997;11(4):341–59. doi:10.1023/A:1008202821328.
27. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH. The arithmetic optimization algorithm. Comput Methods Appl Mech Eng. 2021;376(2):113609. doi:10.1016/j.cma.2020.113609.
28. Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci. 2009;179(13):2232–48. doi:10.1016/j.ins.2009.03.004.
29. Peña-Delgado AF, Peraza-Vázquez H, Almazán-Covarrubias JH, Torres Cruz N, García-Vite PM, Morales-Cepeda AB, et al. A novel bio-inspired algorithm applied to selective harmonic elimination in a three-phase eleven-level inverter. Math Probl Eng. 2020;2020:1–10. doi:10.1155/2020/8856040.
30. Dhiman G, Kumar V. Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Adv Eng Softw. 2017;114(10):48–70. doi:10.1016/j.advengsoft.2017.05.014.
31. Mirjalili S, Mirjalili SM, Hatamlou A. Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl. 2016;27(2):495–513. doi:10.1007/s00521-015-1870-7.
32. Abualigah L, Abd Elaziz M, Sumari P, Geem ZW, Gandomi AH. Reptile Search Algorithm (RSA): a nature-inspired meta-heuristic optimizer. Expert Syst Appl. 2022;191(11):116158. doi:10.1016/j.eswa.2021.116158.
33. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: algorithm and applications. Future Gener Comput Syst. 2019;97:849–72. doi:10.1016/j.future.2019.02.028.
34. Zhao S, Zhang T, Ma S, Wang M. Sea-horse optimizer: a novel nature-inspired meta-heuristic for global optimization problems. Appl Intell. 2023;53(10):11833–60. doi:10.1007/s10489-022-03994-3.
35. Li S, Chen H, Wang M, Heidari AA, Mirjalili S. Slime mould algorithm: a new method for stochastic optimization. Future Gener Comput Syst. 2020;111(Supplement C):300–23. doi:10.1016/j.future.2020.03.055.
36. Shang C, Zhou T-T, Liu S. Optimization of complex engineering problems using modified sine cosine algorithm. Sci Rep. 2022;12(1):20528. doi:10.1038/s41598-022-24840-z.
37. Karaboga D, Basturk B. On the performance of artificial bee colony (ABC) algorithm. Appl Soft Comput. 2008;8(1):687–97. doi:10.1016/j.asoc.2007.05.007.
38. Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S. Equilibrium optimizer: a novel optimization algorithm. Knowl Based Syst. 2020;191:105190. doi:10.1016/j.knosys.2019.105190.
39. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M. Mine blast algorithm: a new population-based algorithm for solving constrained engineering optimization problems. Appl Soft Comput. 2013;13(5): 2592–612. doi:10.1016/j.asoc.2012.11.026.
40. Trivedi IN, Bhoye M, Parmar SA, Jangir P, Jangir N, Kumar A. Ant-Lion optimization algorithm for solving Real Challenging Constrained Engineering optimization problems. In: International Conference on Computing, Communication & Automation, 2016; Greater Noida, India, IEEE. doi:10.13140/RG.2.1.2929.2409.
41. Mokeddem D. A new improved salp swarm algorithm using logarithmic spiral mechanism enhanced with chaos for global optimization. Evol Intell. 2022;15(3):1745–75. doi:10.1007/s12065-021-00587-w.
42. Gandomi AH, Yang X-S, Alavi AH. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng with Comput. 2013;29(1):17–35. doi:10.1007/s00366-011-0241-y.
43. Yu K, Wang X, Wang Z. An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems. J Intell Manuf. 2016;27(4):831–43. doi:10.1007/s10845-014-0918-3.
44. Fakhouri HN, Hudaib A, Sleit A. Hybrid particle swarm optimization with sine cosine algorithm and nelder-mead simplex for solving engineering design problems. Arab J Sci Eng. 2020;45(4):3091–109. doi:10.1007/s13369-019-04285-9.
45. Li L-J, Huang Z, Liu F, Wu Q. A heuristic particle swarm optimizer for optimization of pin connected structures. Comput Struct. 2007;85(7–8):340–9. doi:10.1016/j.compstruc.2006.11.020.

46. Saremi S, Mirjalili S, Lewis A. Grasshopper optimisation algorithm: theory and application. Adv Eng Softw. 2017;105:30–47. doi:10.1016/j.advengsoft.2017.01.004.
47. Arora S, Singh S. Butterfly optimization algorithm: a novel approach for global optimization. Soft Comput. 2019;23(3):715–34. doi:10.1007/s00500-018-3102-4.
48. Lee KS, Geem ZW. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng. 2005;194(36–38):3902–33. doi:10.1016/j.cma.2004.09.007.
49. Akay B, Karaboga D. Artificial bee colony algorithm for large-scale problems and engineering design optimization. J Intell Manuf. 2012;23(4):1001–14. doi:10.1007/s10845-010-0393-4.
50. Coello CAC. Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind. 2000;41(2):113–27. doi:10.1016/S0166-3615(99)00046-9.
51. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM. Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw. 2017;114:163–91. doi:10.1016/j.advengsoft.2017.07.002.
52. Baykasoğlu A, Akpinar Ş. Weighted Superposition Attraction (WSA): a swarm intelligence algorithm for optimization problems–Part 2: constrained optimization. Appl Soft Comput. 2015;37(2):396–415. doi:10.1016/j.asoc.2015.08.052.
53. Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-Qaness MA, Gandomi AH. Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput Ind Eng. 2021;157(11):107250. doi:10.1016/j.cie.2021.107250.
54. Chu S-C, Tsai P-W, and Pan J-S, Eds. Cat swarm optimization. In: PRICAI 2006: Trends in Artificial Intelligence: 9th Pacific Rim International Conference on Artificial Intelligence, 2006 Aug 7–11; Guilin, China: Springer.
55. Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems. Knowl Based Syst. 2016;96(63):120–33. doi:10.1016/j.knosys.2015.12.022.
56. Yasser MH. Breast cancer dataset 2021. Available from: https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset/. [Accessed 2024].