scientific reports

Check for updates

OPEN

Channa argus optimizer for solving numerical optimization and engineering problems

Da Fang[1]✉, Jun Yan & Quan Zhou[2]

In this study, we introduce the Channa Argus Optimizer (CAO), a novel swarm-based meta-heuristic algorithm that draws inspiration from the distinctive hunting and escaping behavior observed in Channa Arguses in the natural world. The CAO algorithm mainly emulates the hunting and escaping behavior of Chinna Argus to realize a tradeoff between exploitation and exploration in the solution space and discourage premature convergence. The competitiveness and effectiveness of CAO are validated utilizing 29 typical CEC2017 and 10 CEC2020 unconstrained benchmarks and 5 real-world constrained optimization mechanical engineering issues. The CAO algorithm was tested on CEC2017 and CEC2020 functions and compared with 7 algorithms to evaluate performance. In addition, the CAO algorithm is tested on the CEC2017 benchmark functions with dimensions of 10-D, 30-D, 50-D, and 100-D. It is then compared and evaluated against other algorithms, using the Wilcoxon rank-sum test and Friedman mean rank. Finally, the CAO algorithm is utilized to tackle five intricate engineering problems to show its robustness. These results have demonstrated the effectiveness and potential of the CAO algorithm, yielding outstanding results and ranking first among other algorithms.

Keywords Channa Argus optimizer, Meta-heuristic global optimization, Engineering optimization, Friedman mean rank, Wilcoxon rank-sum test

As society and technology have advanced, scientists now face a vast array of challenges[1,2]. These issues get more complicated with time. Problems in optimization can be roughly divided into a number of categories, including single-objective or multi-objective, continuous or discrete and static or dynamic[3]. These categories make it feasible to create resolution techniques that are as effective as possible while taking into account the inherent characteristics of the situations. In order to identify the optimal solution, resolution techniques scan the search space or solution space. They frequently call for iterative procedures that enhance one or more solutions simultaneously. As a result, the search space is explored and gradually advances in the direction of the best answer.

Numerous techniques have been developed throughout time to address optimization issues, which may be generally divided into two families: meta-heuristics and heuristics[4–6]. When obtaining an accurate solution in a finite amount of time is not feasible, heuristics are employed as an approach.

Although it might not be the precise answer, using heuristics yields a good approximation of the issue in a reasonable amount of time[7,8]. When it comes to tackling situations that call for approximations, heuristics are helpful. Conversely, meta-heuristics are higher-level heuristics that are employed to address optimization issues, especially those using partial data, such as those in machine learning and artificial intelligence[9–11]. Meta-heuristics can be used in situations where the solution set is too big to test thoroughly. They may also be used in combinatorial optimization and stochastic optimization, where they look for a sizable, discrete set of workable solutions. Meta-heuristics are a generic technique used to create a broad range of optimization algorithms and need less computing power than conventional approaches[12]. Any problem can be solved using meta-heuristic algorithms, even if the result isn't always the best or most precise.

Real-world optimization problems frequently have uncertain search spaces and stochastic behavior. As a result, meta-heuristic algorithms without derivatives and without limiting assumptions have been developed. Because of their great adaptability, metaheuristic algorithms may be used for a wide range of optimization issues. In complex and dynamic situations, metaheuristic algorithms offer helpful answers to a wide range of optimization issues due to their great degree of flexibility. There are two types of mathematical optimization techniques: deterministic and stochastic[13]. Deterministic methods, including linear and non-linear programming, explore the issue space and identify a solution by using the gradient knowledge of the problem[14]. These methods work

[1]Wuhan Technical College of Communications, Wuhan 430065, China. [2]Hubei Communications Technical College, Wuhan 430079, China. ✉email: fangda@whtcc.edu.cn

well for linear search space problems, but when used for non-linear search space problems—like real-world non-convex problems—they are susceptible to local optima entrapment. To solve these problems, these algorithms must be modified or hybridized[15,16].

One well-known class of algorithms created especially to handle complex optimization problems is called a metaheuristic. They are founded on human-based (HU), physics-based (PH), swarm intelligence (SI), and evolutionary algorithms (EA)[17–19]. One of the primary causes of the SI's increased acceptance across all courses is the mathematical models' simplicity. In optimization as well as many other fields, metaheuristics have become increasingly prominent[20]. This category includes stochastic optimization methods, which are useful in many sectors and scientific fields. More and more new algorithms have emerged in recent years, such as Builder Optimization Algorithm(BOA)[21], Candle Flame Optimization(CFA)[22], Greylag Goose Optimization(GGO)[23], Makeup Artist Optimization Algorithm(MAO)[24], Tailor Optimization Algorithm (TOA)[25], Orangutan Optimization Algorithm(OOA)[26], Paper Publishing Based Optimization(PPBO)[27], Perfumer Optimization Algorithm(POA)[28], Revolution Optimization Algorithm(ROA)[29], Singer Optimization Algorithm(SOA)[30], Spider-Tailed Horned Viper Optimization(SHVO)[31]. However, the applications of these algorithms are not thoroughly discussed because the theoretical component of this study is its primary focus. Other pertinent sites are recommended for researchers who wish to investigate the real-world applications[32].

The primary objective of this research paper is to introduce a novel metaheuristic algorithm named Channa Argus Optimizer (CAO), which is specifically designed to address optimization problems characterized by extensive search spaces. It is based on animals' behavior and mimics Channa Argus' behavior. The proposed algorithm is evaluated against five popular and recent metaheuristic algorithms using CEC2017, CEC2020, and industrial engineering problems. It is crucial to remember, nonetheless, that the No Free Lunch theorem for optimization states that an algorithm's performance on one optimization issue does not ensure that it will succeed on another problem with distinct features. Therefore, in order to maximize the efficiency and usefulness of metaheuristic algorithms, it is essential to carefully examine and modify them to fit specific problem domains.

The main contributions of this paper are presented as follows.

1. An innovative optimization algorithm named Channa Argus Optimizer is introduced for global optimization industrial engineering problems.
2. The performance of the Channa Argus Optimizer is calculated utilizing three challenging problems: CEC2017, CEC2020, and industrial engineering problems.
3. The performance of the Channa Argus Optimizer is analyzed with the state-of-the-art swarm intelligence (SI) algorithms, physical inspiration algorithms, and biological inspiration-based algorithms.

The subsequent sections of the paper are structured as follows: Sect. "Literature review" presents an elaborate literature review, while Sect. "Channa argus optimizer(CAO)" provides a comprehensive explanation of the inspiration behind and the mathematical model of the newly proposed Channa Argus Optimizer. The outcomes derived from the experimentation are discussed in Sect. "Results on benchmark functions". The paper concludes by summarizing the findings and outlining potential future directions for research in Sect. "Engineering optimization test problems".

Literature review

Due to the increasing popularity of metaheuristic algorithms in solving different types of problems, various types of metaheuristic algorithms have been proposed. Each algorithm has its own characteristics and methods for solving optimization problems[33,34]. The inspiration for optimizing algorithms can include different types of natural phenomena, including animals and humans, physics, and evolution[35,36]. Various algorithms have been proposed based on the source of inspiration. In the context of mathematical modeling of natural behavior, these algorithms have led to the emergence of new methods and technologies in optimization methods[37]. Several optimization methods have been proposed in the literature, each with unique inspirations[38,39]. The inspiration for metaheuristic algorithms can be roughly divided into several natural sources. The general classification of meta-heuristic algorithms can be seen in Fig. 1.

The main inspiration for metaheuristic algorithms comes from animal life and behavior. These algorithms have been influenced by the collective behavior of social insects and animals and have also made significant contributions to human evolution[40].

The main inspiration for metaheuristic algorithms comes from animal life and behavior. These algorithms , influenced by the collective behavior of social insects and animals, have also made significant contributions to the development of metaheuristic algorithms in human evolution[41]. The Puma Optimizer (PO), a heuristic metaheuristic algorithm, is essentially such a method[42]. It is proposed as a new optimization algorithm inspired by the intelligence and life of Pumas. Whale Optimization Algorithm (WOA) is a new metaheuristic optimization algorithm[43–45]. Its foam search is a response to the social behavior of Humpback whales. Pied kingfisher optimizer, a heuristic metaheuristic algorithm, is inspired by the unique hunting behavior and symbiosis of kingfishers in nature. Mayfly Algorithm (MA) is inspired by the flight behavior and the mating process of mayflies. The suggested method combines the main benefits of evolutionary algorithms with swarm intelligence[46–48]. Nature has an impact on Ant Lion Optimization (ALO). It mimics the five essential hunting processes that ants and lions use when hunting[49–52]. To handle optimization issues with various structures, another approach called the Pathfinder approach (PFA) has been presented. It looks for the best food source or prey, is inspired by the collective evolution of animals, and establishes a hierarchical structure of group leadership.

The second source of inspiration is physical processes or mathematical models. For instance, the arithmetic optimization algorithm (AOA), a novel metaheuristic technique, was proposed by Abualigah et al[53–55]. This approach makes use of how popular mathematical operations—like multiply, divide, add, and subtract—behave

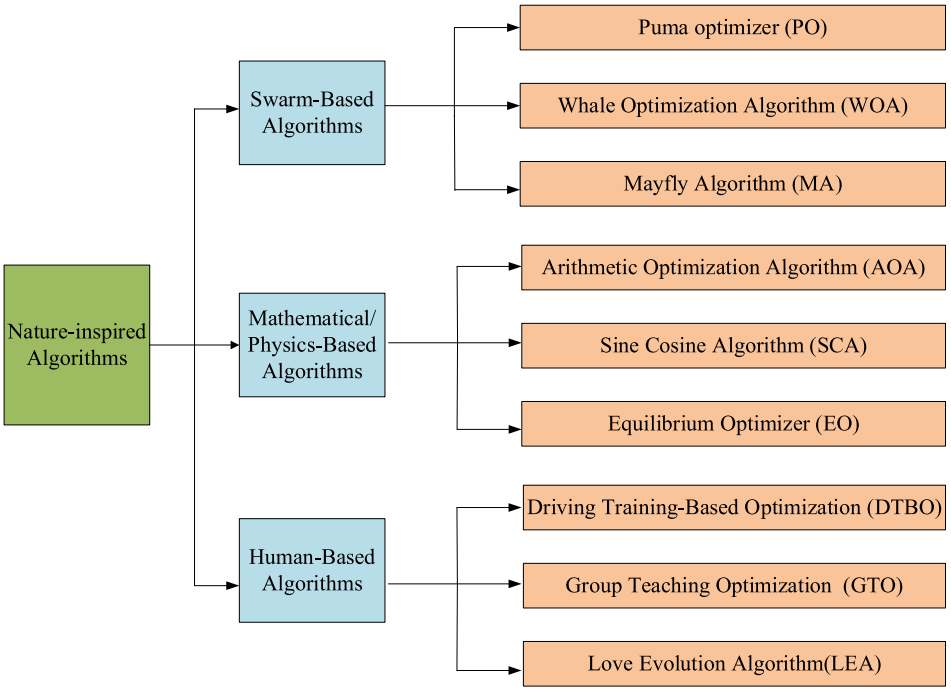

Fig. 1. Corresponding classification of meta-heuristic algorithms.

in different search spaces. Optimization algorithms are carried out using the mathematical description of AOA. Using mathematical models based on sine and cosine functions, Mirjalili created the[56–59] Technique (SCA), a population-based optimization method. In order to find and use the search space at different optimization milestones, SCA employs a number of random and adaptive factors and fluctuates either outward or towards the optimal solution. Furthermore, based on regulated volume and mass, Faramarzi et al. presented a novel equilibrium optimizer (EO) that uses each solution as a search agent whose location can reach equilibrium state[33].

Metaheuristic algorithms also draw inspiration from the motivations of populations and their behaviors. People's driving behavior while learning serves as the basis for the Driving Training Based Optimization (DTBO) algorithm, which is based on driving training[60]. Three phases make up the demonstration: practice, teacher-created modes, and instruction by the teacher driving instructor. To assess and test the DTBO algorithm, the author used 53 industry-standard features, including CEC 2017 functionality. This work proposes a new metaheuristic algorithm, the Group Teaching Optimization Algorithm (GTOA), to tackle a variety of optimization problems. It is a mechanism influenced by community teaching. Four fundamental guidelines for modifying group instruction, utilizing group technology, and implementing group teaching mode to facilitate workflow were described. Presented a novel human behavior-based optimization method for the election and leader selection process. On the basis of this, the algorithm guides search agents in two stages: exploration and exploitation. The author tested the algorithm using 33 objective functions of different dimensions and complexities. The results demonstrate how the algorithm effectively handles a range of optimization problems. To solve numerical and structural design optimization problems, Jahangiri et al. Presented an efficient and reliable metaheuristic method for a novel called 'Interactive Self-directed Teaching School'. International Accounting Standards are group based algorithms inspired by the experience of a self-study school where students can enhance their knowledge through self-study, collaborative discourse, feedback, and competition.

According to all the explanations in this section, each optimization algorithm has its own advantages and disadvantages. A significant drawback of optimizer algorithms is their poor performance in intensive and diverse components, lack of balance between exploration and development stages, staying at local optima, lack of adaptation to different mechanisms to solve discrete problems, and high execution time. The ability of optimizer algorithms can be improved by adapting powerful mechanisms that can increase the diversity of output solutions or quickly move towards the best solution to easily explore the entire optimization space. However, in order to minimize execution time to a reasonable level, these techniques should be as computationally simple as possible. In addition, intelligent mechanisms and programs can be used to balance the exploration and development stages, significantly improving the performance of algorithms[61]. Regarding the provided explanation, in these cases, it prompts us to provide a powerful algorithm, a powerful mechanism in the exploration and development phase with minimal execution time, and then introduce a new intelligent mechanism phase transition to achieve maximum performance from the proposed algorithm. Finally, due to its unique functionality, the proposed algorithm can be used to solve various optimization problems in different optimization spaces. We studied cases of global optimization problems and engineering technical problems.

Channa argus optimizer(CAO)

In this section, the main inspiration for the Channa Argus algorithm was reviewed, followed by a comprehensive description of the proposed algorithm and the establishment of a mathematical model.

Inspiration

Channa argus (see Fig. 2 which is Photographed by the corresponding author Da Fang), also known as black fish, money fish, mullet and snake head fish, belongs to the perciformes Ophiocephalus genus fish. The adult length is about 40 ~ 60 cm, and the maximum length can reach 1 m. The general weight is about 0.5 to 1 kg, and the maximum is 8 to 9 kg. The body is fat and elongated, cylindric at the front and flat at the rear. The head is large and long pointed, slightly flattened at the front, slightly raised at the back, and the top of the skull is covered with irregular scales. The snout is short and blunt, the mouth is large, the mouth cleft is slightly oblique, and the jaw is slightly prominent. The teeth in the mouth are clustered, the upper jaw has a fine tooth band, and the teeth on both sides of the lower jaw are sharp. The body color is gray-black, the back of the head and the back of the body are darker and darker, the abdomen is light white, there are about 11 irregular large black spots on the side of the body, and there is 1 small black spot along the middle line of the back. Channa argus is a large benthic freshwater fish. It is native to the river basins of the East Asian and Pacific river systems, and its worldwide distribution can extend from the Korean Peninsula, the Heilongjiang River basin and the Ussuri River basin on the border of China and Russia to the Xingkai Lake and the Yangtze River basin in the south. In China, it is mainly distributed in Hunan, Hubei, Anhui, Henan, Shandong, Hebei, Liaoning and other provinces. Later, it was widely introduced into Japan, Central Asian countries, and eastern North America. Channa argus is a ferocious carnivorous fish that feeds on other fish, frogs, crustaceans, and insects. It has a special structure of mouthparts adapted to its predation behavior and mainly adopts ambush mode of predation. Channa argus chooses different foods at different stages of growth.

It is a fierce carnivorous fish that feeds mainly on other fish, frogs, freshwater crustaceans, and aquatic insects. Channa argus preys by ambush. They usually hide near grass or other cover, and when they spot an approaching fish or shrimp, they rush to swallow their prey in one gulp. Their food intake is quite large, and their maximum stomach capacity can even reach 60% of their body weight. After laying their eggs, the Channa argus will lurk beneath or near the nest, guarding the eggs. This protective behavior continues until the fry hatch and are able to swim freely and feed independently, which usually takes about 4 weeks, by which time the fry have reached 2 cm, at which point the fry have the ability to live independently.

Inspired by the behaviors of Channa argus, we have developed a novel meta-heuristic algorithm named the Channa argus optimizer (CAO). In the subsequent subsection, we establish the mathematical model of CAO as follows.

Mathematical model

This section provides a detailed description of the mathematical model for CAO. We first present the mathematical expressions for the hunting and escaping strategies, followed by an analysis of the mathematical models for the hunting and escaping strategy.

Initialization

CAO starts the search process by creating a set of initial solutions at random from the search space as the first trial, much like many other population-based techniques. The initial population was created using the following equation:

$$\begin{array}{c} X_{i,j} = LB + (UB - LB) \times rand, \\ i = 1,2,\ldots,Nandj = 1,2,\ldots,Dim \end{array} \tag{1}$$

where UB and LB stand for the upper and lower boundaries of the search range, rand is a random value between 0 and 1, and $X_{i,j}$ is the location of the ith individual at the jth dimension.

A fitness function is used to assess each person's fitness value according to their capacity to solve the challenge after the first population has been created.

Fig. 2. Channa argus in nature(Photographed by the corresponding author Da Fang).

Hunting strategies (exploration phase)

CAO's exploratory phase was inspired by the predatory behavior of Channa Argus. The Channa Argus, known as the "tiger of the water", holds a top predatory position in the pond ecosystem with its powerful hunting ability and unique survival strategy. This carnivorous fish not only preys on various small fish and shrimp, but when it is large enough, it may even hunt frogs, young birds and small mammals. They prefer to inhabit still water environments with abundant aquatic plants and soft mud substrates, and their range of activities is relatively fixed. Channa Argus often lurk quietly in hidden spots at the bottom of the water, patiently waiting for fish, shrimp and other prey to pass by. Then they strike with lightning speed, catching the prey in one fell swoop without chasing. The Channa Argus will look around for a partner who has found prey, as shown in Fig. 3. The optimal solution, the suboptimal solution and the central position of the prey could all be the locations where Channa Argus ambushed, and the attack direction was in one direction between the optimal solution and the central position.

In CAO, the location of the search agent is determined by the location of the parent and the center, and the direction is determined by the optimal prey and Channa argus center. The center location of Channa arguses is updated according to the following equation:

$$Z_i(t+1) = S(t) + P(t) \otimes (r \times (G(t) - Z_i(t)) + (1-r) \times (\overline{Z}(t) - Z_i(t))) \tag{2}$$

Among them, *Zi*(*t*) denotes the *i* th individual during the *t* th iteration, P(*t*) indicates a vector including random numbers on the basis of Gaussian distribution denoting the Brownian motion, the sign $\otimes$ represents entry-wise multiplications, r indicates a number randomly chosen from [0,1]. Furthermore, *G*(*t*) refers to the current best solution, S(*t*) is a random individual selected from a set of three elites in the swarm, and $\overline{Z}(t)$ denotes the centroid position of the whole swarm. The corresponding mathematical expressions are presented in the following:

$$\overline{Z}(t) = \frac{1}{N}\sum_{i=1}^{N} Z_i(t) \tag{3}$$

$$S(t) \in [G(t), Z_{second}(t), Z_c(t)] \tag{4}$$

where *Zsecond* (*t*) represent the second-best individual in the current population, respectively. $Z_c(t)$ denotes the centroid position of individuals whose fitness values ranked in the top 50%. In this study, for simplicity, the individuals whose fitness values ranked in the top 50% are named leaders. Additionally, *Zc* (*t*) is calculated utilizing the mathematical expression in Eq. (5).

$$Z_c(t) = \frac{1}{N_1}\sum_{i=1}^{N_1} Z_i(t) \tag{5}$$

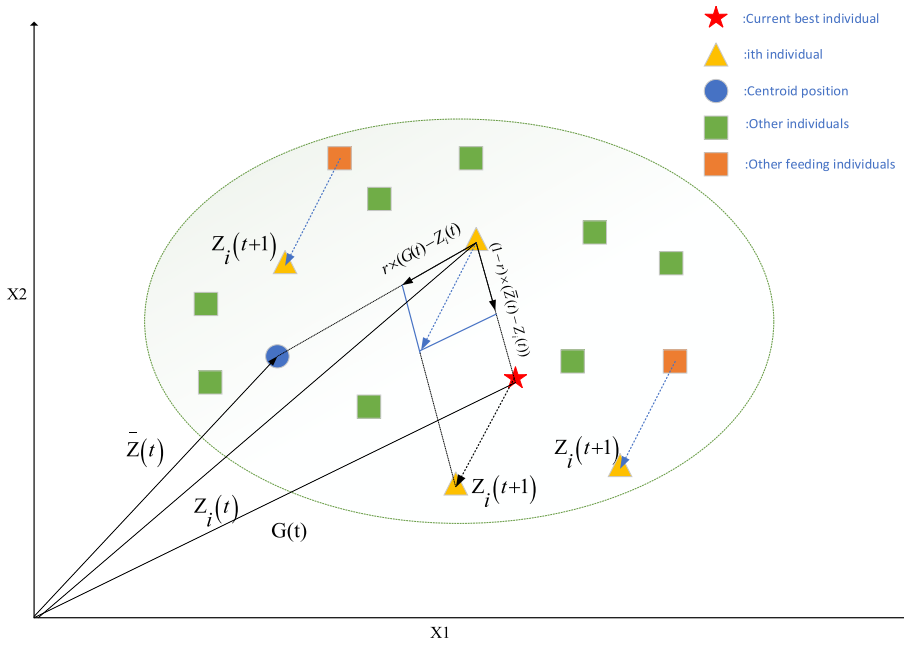

Fig. 3. The predatory behavior of the Channa argus.

where $N1$ indicates the number of leaders, that is, $N1$ is equal to half the size of the whole swarm, and $Zi(t)$ represents the ith best leader. Therefore, during each iteration, the $Elite(t)$ is randomly selected from a set that consists of the current best solution, second-best individual, and centroid position of leaders.

Escaping strategy (exploitation phase)
As shown earlier, Channa argus seedlings will swim into the mouth of large Channa argus if they are in danger while prowling. But it is not in order to save the mother, the mother Channa argus will not be blind because of production, the big Channa argus rise fierce, has a strong attack ability, but the small Channa argus is easy to become the prey of other fish, so when the fish found danger, it will suck the small Channa argus into the mouth. Wait for the danger to pass, and then spit out the baby Channa argus. The baby Channa argus will also react normally and will swim into the mouth of the adult Channa argus when danger comes, so that it can be safe. After becoming a mother, a Channa argus will exhibit a completely different instinct—an extreme protection of its children. Black fish protecting their young usually lasts for about a month. When they protect their young, they usually show these two behaviors: first, they will act like dogs protecting their food. As soon as a strange object approaches their cubs, they will be particularly fierce and launch an attack at any time. Second, when they sense danger approaching, they will put their cubs in their mouths. This is also a normal phenomenon of the law of survival. Figure 4 shows the escape of the Channa argus.

In this part, the exploitative characteristic of CAO is introduced. Instead of expanding with a high-decentralized feature in the solution space, search agents are encouraged to exploit high-quality solutions around the current best solution when Channa argus encounter danger. The escaping strategy can be mathematically modeled in Eq. 6 and K(t) parameter can be calculated as follows:

$$Z_i(t+1) = K(t) \times G(t) + P(t) \otimes (r \times (G(t) - Z_i(t)) + (1-r) \times (\overline{Z}(t) - Z_i(t))) \tag{6}$$

$$K(t) = 0.1 * (e^{\frac{t}{maxIter}} - 1) \tag{7}$$

where K(t) is the Original position inertia, r indicates the random number chosen from [0,1], $Zi(t)$ denotes the i th individual during the t th iteration, P(t) indicates a vector including random numbers on the basis of Gaussian distribution denoting the Brownian motion, the sign $\otimes$ represents entry-wise multiplications, t indicates current iteration value, maxIter indicates maximum iteration value. Furthermore, $G(t)$ refers to the current best solution, S(t) is a random individual selected from a set of three elites in the swarm, and $\overline{Z}(t)$ denotes the centroid position of the whole swarm.

Computational complexity analysis of CAO

This section explains the operational capabilities of the proposed CAO in terms of its time and space complexity.

Time complexity
To properly describe the computational complexity of an optimization algorithm, a function is typically utilized to relate the algorithm's running time to the size of the input problem. This function is commonly expressed using Big-O notation. The time complexity of the algorithm is influenced by various factors, including the population size (N), the dimensions of the problem (Dim), the number of iterations (T), and the cost of function evaluations (C).Thus, the time complexity of the CAO algorithm can be more precisely expressed as:

$$O(CAO)\& = O(\text{Initialization}) + O(\text{cost function}) + O(\text{Updating strategy}) \tag{8}$$

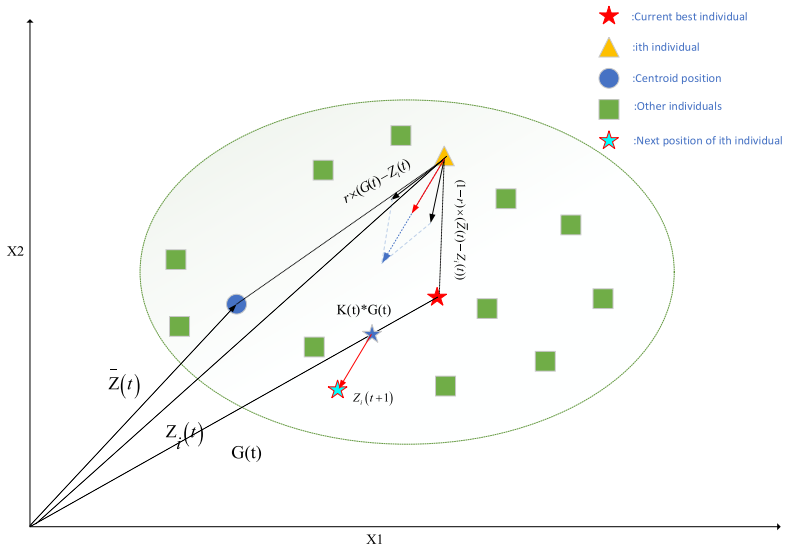

Fig. 4. The escaping behavior of the Channa argus.

where the components of Eq. (8) can be characterized by their time complexities as follows:

(1) The generation of the population initialization requires $O(N \times Dim)$.
(2) The evaluation of the cost function requires $O(T \times N)$ time.
(3) The position updating in exploration or exploitation phase requires $O(T \times N \times Dim)$ time.

Therefore, the overall time complexity of CAO can be formulated as follows:

$$O(PKO) = O(NDim + TN + TNDim) \tag{9}$$

Space complexity
The space complexity of CAO is determined by two parameters: the number of Channa arguses and the dimensions of the problem, and it affects the memory space.

In short, CAO's model of behavior involves constantly exploring and using the surface of the water to find potential prey, then moving toward the prey to capture the prey, while encountering predators that swim quickly like the motherfish in search of safety. Mathematical models have been proposed for CAO to demonstrate its ability to solve optimization problems. It is important to note that the framework of CAO for addressing optimization problems bears a resemblance to other meta-heuristic algorithms. The optimization process starts with a randomly generated population of potential solutions, which is subject to iterative improvements until a predefined termination condition is met (i.e., when $t < T + 1$). The specific operators used to update the population are what distinguish one algorithm from another.

Pseudo-code and flowchart of CAO

This section describes the pseudo-code and flowchart of CAO. In our study, the mechanism is devised to reflect this situation and maintain exploitation and exploration. As presented in Algorithm 1, in the early stages of the iteration, individuals from the entire population are randomly exploited and explored.

Input: Population size N, maximum number of iterations T and parameter settings
Output: Channa arguses's location and fitness potential
Initialize the population of Channa arguses
while t < T **do**
  Calculate the fitness of each solution to obtain the optimal individual
  for i=1:N **do**
    Calculate the best, second, average position of all individuals.
    if rand< 0.5+0.5*(t/T)^0.2 **then**
      Update the position according to equation (2)
    else
      Update the position according to equation (6)
    end if
    Check the boundary of the new position.
  end for
  t=t+1
end while

Algorithm 1.Main steps of CAO algorithm.

The above procedure is shown in Fig. 5.

Results on benchmark functions

Assessment of CAO on CEC-2017 benchmark test functions

Description of the benchmark test functions
We employ the well-known CEC-2017 test suite to assess the CAO algorithm's global exploration, local extremum avoidance, local mining, and other performance metrics. Due to space constraints, we concentrate on the Dim = 10, 30, 50, and 100 dimensions situation in this experiment. The specifics of the CEC-2017 test functions, which are separated into four groups as indicated, are given in Table 1.

The algorithms' capacity for exploration is examined using multimodal functions Fun4–Fun10, while their capacity for exploitation is assessed using unimodal functions Fun1 and Fun3. The more challenging test functions used to assess the algorithms' local extremum avoidance are the composition functions Fun21–Fun30 and hybrid functions Fun11–Fun20. Note that because of its unpredictable nature, we did not utilize the Fun2 test function from the CEC-2017 benchmark.

Experimental setup
We compared the proposed CAO algorithm with seven well-known algorithms, namely Mayfly Algorithm (MA)[46], Loin Swarm Optimization (LSO)[58], Grey Wolf Optimizer(GWO)[62,63], Harris Hawks Optimization (HHO)[64–66],Whale Optimization Algorithm (WOA)[43,44], Equilibrium Optimizer (EO)[33], and Marine Predators Algorithm (MPA)[50]. Table 2 displays the parameter settings for all algorithms. The parameters of CAO, in addition to the consistent parameters for all algorithms, r is a random number of 0 and 1, as well as original

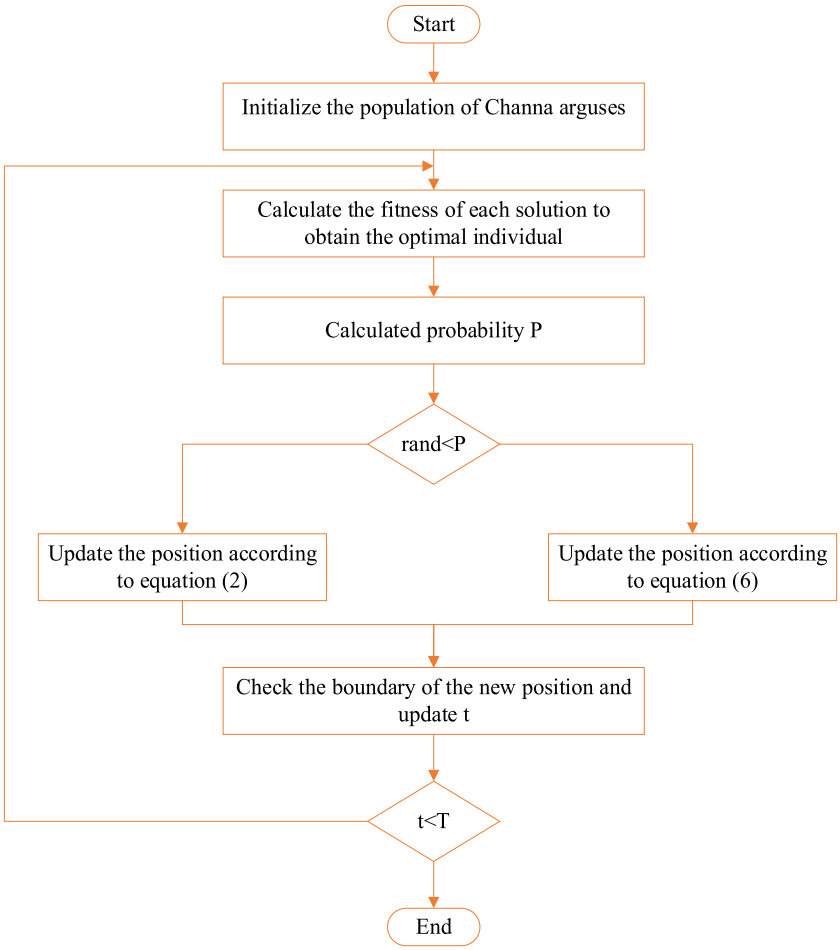

Fig. 5. The flow chart of Channa arguses optimizer.

position inertia, which is set to 0.2 here. The comparison algorithm parameters were set based on their respective literature. To ensure a fair comparison, each function was executed separately for 20 trials. The statistical analysis was performed using the mean fitness value (Mean) and the standard deviation (Std) of the 20 trials. The experiments were implemented on MATLAB 2018b on a Windows 11 Operating System, utilizing a core R7 CPU and 32 GB RAM.

Qualitative analysis

To better understand how the CAO algorithm improves with each iteration, this section of the paper provides a qualitative assessment of the CAO's performance. Specifically, in this experiment, the convergence behavior of CAO is reflected by the search history, convergence graph, history of average fitness, and diagram of trajectory in the first dimension. As depicted in Fig. 6, the first column is the description of the parametric space, and it reveals the smooth structure of unimodal problems such as $C2017_1$ and $C2017_3$. Meanwhile, a substantial number of local optima exist in simple multimodal problems and complicated hybrid problems well emulate the real solution space. The convergence graph in the second column is the most broadly utilized metric to validate the performance of metaheuristic techniques. As described in Fig. 6, the convergence graphs attained by CAO suggest that the algorithm has a rapid convergence rate on all benchmarks. For unimodal problems, due to the interaction and learning between individuals, CAO presents a good exploitative characteristic to approach the global optimum. When handling simple multimodal problems and hybrid problems, CAO sometimes falls temporarily into local optima, but the algorithm achieves a high precision under the guidance of the elites in the swarm. Meanwhile, in the last steps of iteration, the dynamic step length generated by Brownian motion can discourage premature convergence effectively. Also, in the third column, the descending behavior can be observed in the diagrams of average fitness history. Then in the fifth column, the search history diagram visually presents all individuals' position history during the iteration procedure. Note that individuals tend to discover potential and promising areas in early iterations and finally cluster around the global optimal solution, which indicates CAO realizes a great tradeoff between exploration and exploitation. Especially for hybrid problem $C2017_{24}$, the CAO has focused on exploiting the left region in the solution space for a long time, whereas the best outcomes are attained in the right space. Actually, this shows the excellent exploration capability possessed by the develop technique, which enables the swarm's diversity to be preserved and facilitates local optima avoidance.

	No	Function	Range	Fun_{Best}
Unimodal functions	$C2017_1$	Shifted & Rotated Bent Cigar Function	[-100 100]	100
	$C2017_3$	Shifted & Rotated Zakharov Function	[-100 100]	300
Simple multimodal functions	$C2017_4$	Shifted & Rotated Rosenbrock's Function	[-100 100]	400
	$C2017_5$	Shifted & Rotated Rastrigin's Function	[-100 100]	500
	$C2017_6$	Shifted & Rotated Expanded Scaffer's Function	[-100 100]	600
	$C2017_7$	Shifted & Rotated Lunacek Bi_Rastrigin Function	[-100 100]	700
	$C2017_8$	Shifted & Rotated Non-Continuous Rastrigin's Function	[-100 100]	800
	$C2017_9$	Shifted & Rotated Levy Function	[-100 100]	900
	$C2017_{10}$	Shifted & Rotated Schwefel's Function	[-100 100]	1000
Hybrid functions	$C2017_{11}$	Hybrid-Function-1(N = 3)	[-100 100]	1100
	$C2017_{12}$	Hybrid-Function-2(N = 3)	[-100 100]	1200
	$C2017_{13}$	Hybrid-Function-3(N = 3)	[-100 100]	1300
	$C2017_{14}$	Hybrid-Function-4(N = 4)	[-100 100]	1400
	$C2017_{15}$	Hybrid-Function-5(N = 4)	[-100 100]	1500
	$C2017_{16}$	Hybrid-Function-6(N = 4)	[-100 100]	1600
	$C2017_{17}$	Hybrid-Function-6(N = 5)	[-100 100]	1700
	$C2017_{18}$	Hybrid-Function-6(N = 5)	[-100 100]	1800
	$C2017_{19}$	Hybrid-Function-6(N = 5)	[-100 100]	1900
Composition functions	$C2017_{20}$	Hybrid-Function-6(N = 6)	[-100 100]	2000
	$C2017_{21}$	Composition-Function-1(N = 3)	[-100 100]	2100
	$C2017_{22}$	Composition-Function-2(N = 3)	[-100 100]	2200
	$C2017_{23}$	Composition-Function-3(N = 4)	[-100 100]	2300
	$C2017_{24}$	Composition-Function-4(N = 4)	[-100 100]	2400
	$C2017_{25}$	Composition-Function-5(N = 5)	[-100 100]	2500
	$C2017_{26}$	Composition-Function-6(N = 5)	[-100 100]	2600
	$C2017_{27}$	Composition-Function-7(N = 6)	[-100 100]	2700
	$C2017_{28}$	Composition-Function-8(N = 6)	[-100 100]	2800
	$C2017_{29}$	Composition-Function-9(N = 3)	[-100 100]	2900
	$C2017_{30}$	Composition-Function-10(N = 3)	[-100 100]	3000

Table 1. Summary of the CEC2017 benchmark functions.

Method	Parameter	Value
MA	Attraction coefficient of male mayfly	$a_1 = 1$
	Attraction coefficient of female mayfly	$a_2 = 1$
	Visibility coefficient of male mayfly	$beta = 2$
	Dance coefficient	$dance = 5$
LSO	Probability factor of the lion cub behavior	$q \in [01]$
	Lioness movement range perturbation factor	$a_f = a_1\,(\overline{x}_{max} - \overline{x}_{min})\exp(-30t/T)^{10}$
	Lion cub movement range perturbation factor	$a_c = a_2\,(\overline{x}_{max} - \overline{x}_{min})\,(\mathrm{T} - \mathrm{t})/\mathrm{T}$
GWO	Convergence parameter	Linear reduction from 2 to 0
HHO	Beta	1.5
WOA	Convergence parameter	Linear reduction from 2 to 0
	Random vector	$r \in [01]$
	Random number	$l \in [-11]$
EO	exploration quantity factor	$a_1 = 2$
	exploitation quantity factor	$a_2 = 1$
MPA	Fish gathering device	$FADs = 0.2$

Table 2. Parameter setting of each algorithm.

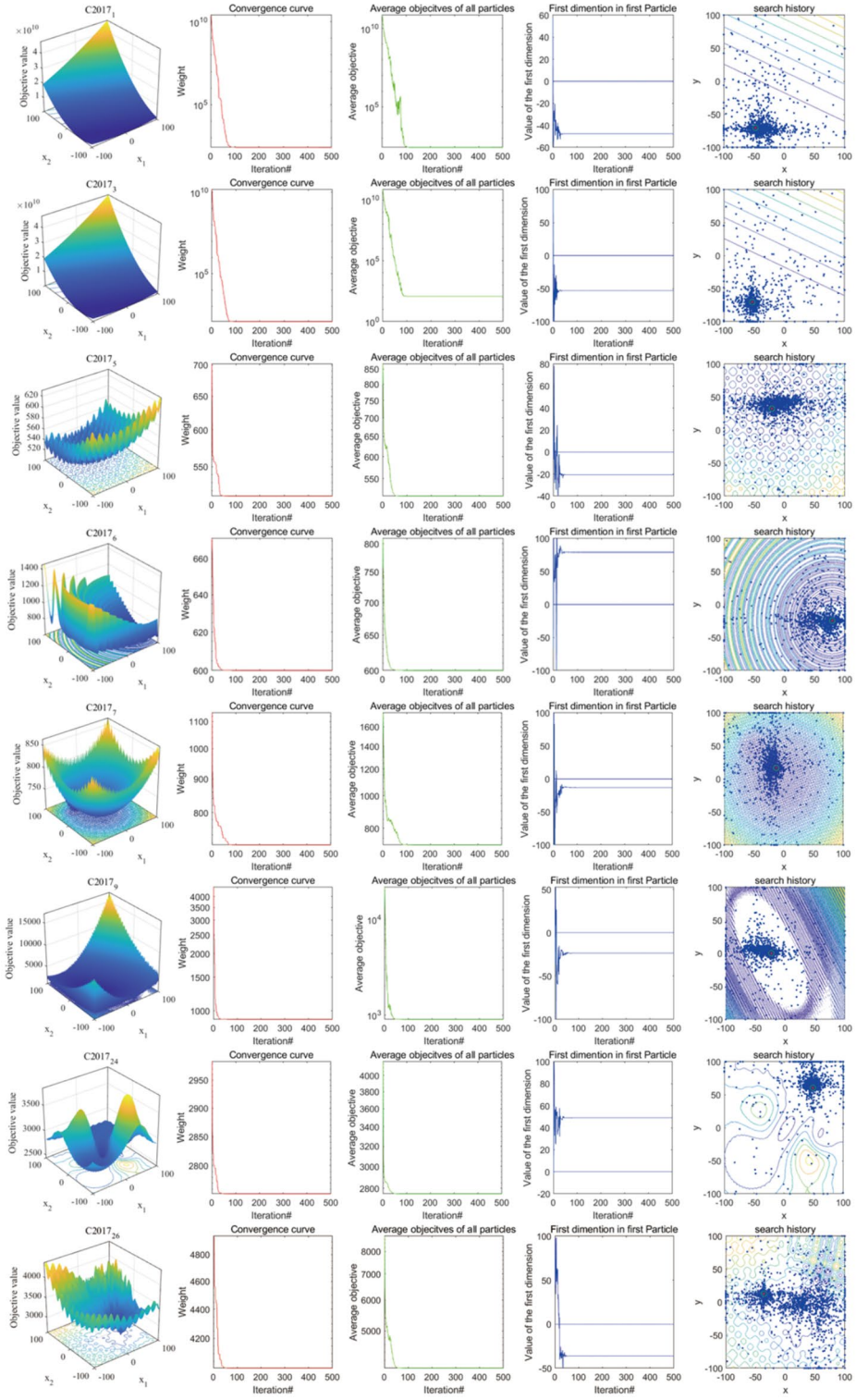

Fig. 6. Trajectory in two dimensions, convergence curve , average fitness, and search history.

More investigation into the exploration and exploitation balance of the CAO algorithm is required to better understand why particular metaheuristics perform better for global optimization issues. Therefore, an empirical analysis was carried out to identify the factors influencing the CAO's capacity for exploration and exploitation, and the findings were presented. Figures 7 and 8 display the CAO's diversity and balance analyses, respectively. As observed in Fig. 7, CAO showed high exploration and low exploitation at the start of the iterations. It became increasingly exploitative as the iterations went on. As can be seen in Fig. 8, every graph has a declining trend, signifying a shift from early exploration to late exploitation.

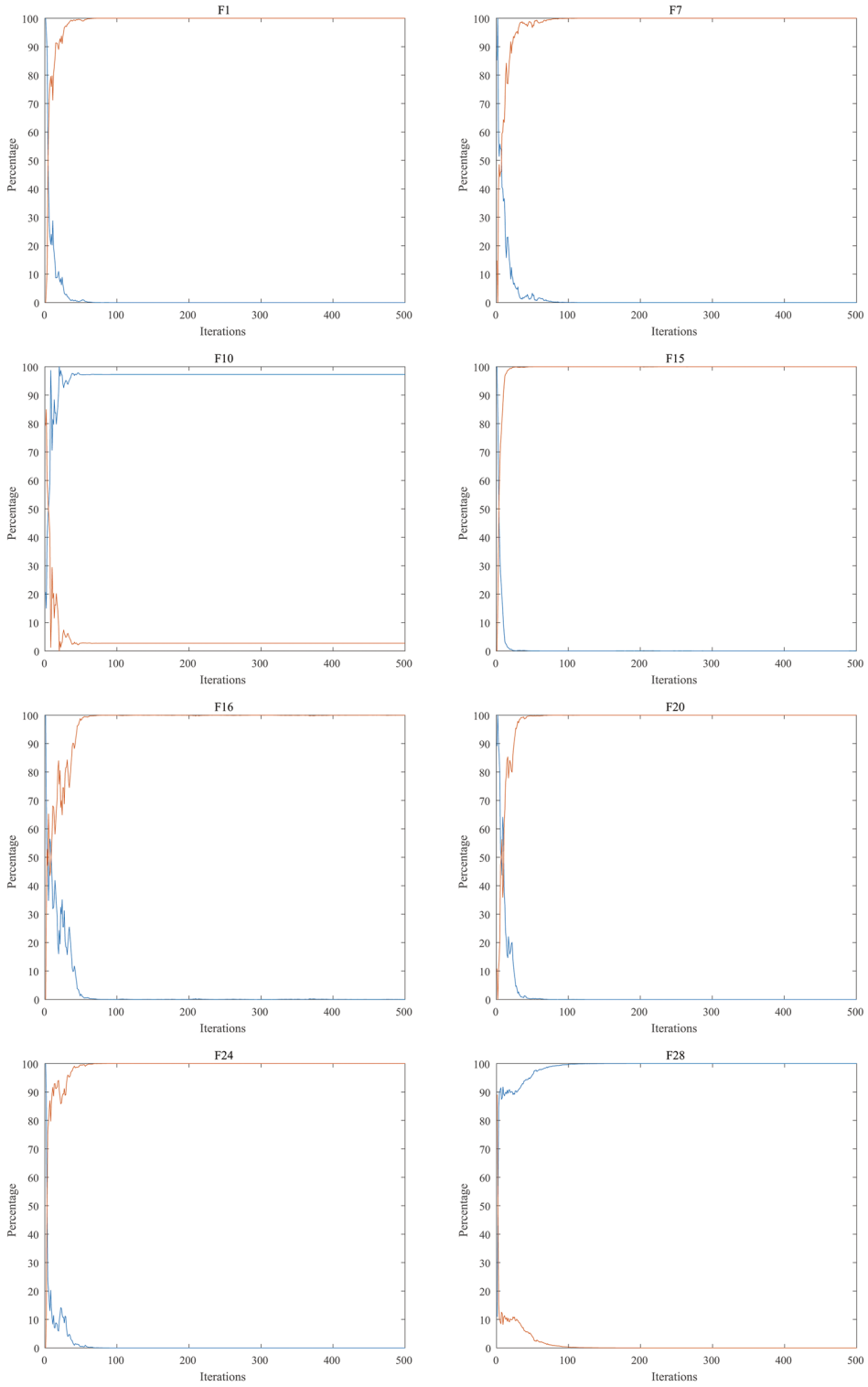

Fig. 7. Exploration and exploitation phases of CAO under CEC-2017 test suite.

In conclusion, Fig. 6's convergence and average fitness graphs show that CAO can accomplish consistent optimization for issues with varying degrees of complexity. The diversity, trajectory, and search history data show that CAO can appropriately strike a balance between exploitation and exploration. The capacity of CAO to identify the global optimum solution is due to this characteristic.

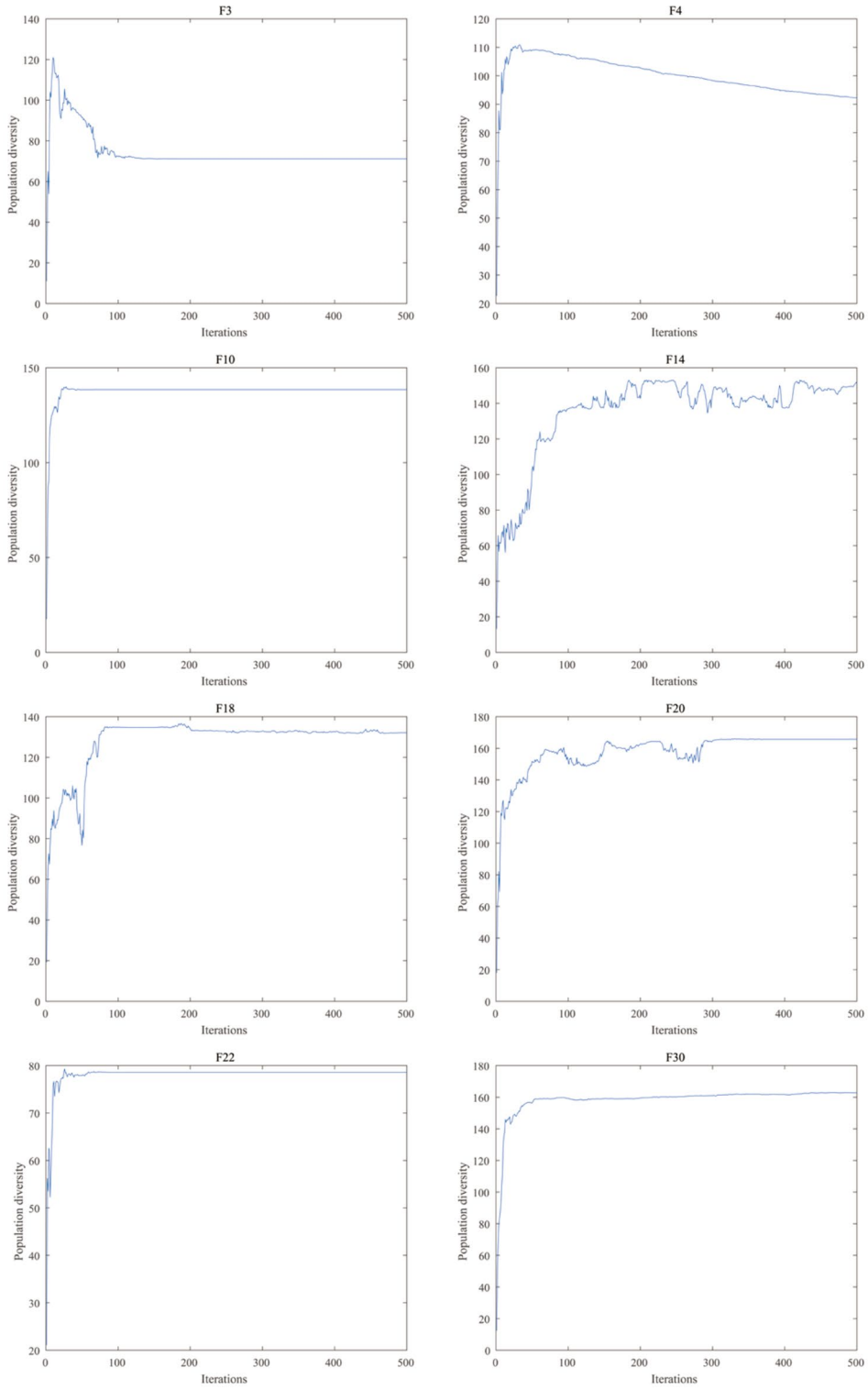

Fig. 8. Diversity of CAO under CEC-2017 test suite.

Statistical results

In this subsection, CAO and other compared algorithms have been evaluated. A total of 29 standard functions have been used for evaluation, each of which has been discussed in its respective sections. Firstly, the scalability analyses have been done using functions F1 to F13, which are scalable functions, and their dimensions can be changed. For this evaluation, functions were tested with dimensions of 10, 30, 100, 500, and 1000, respectively, in four separate implementations and the corresponding results are depicted in four Tables 3, 4, 5, 6. This

Fun	Indice	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_1	Mean	**3.343E+03**	1.182E+07	3.362E+08	6.378E+07	1.861E+10	1.016E+07	8.780E+09	3.073E+05
	Std	**3.389E+03**	4.167E+06	2.318E+08	1.466E+08	3.841E+10	8.475E+06	1.700E+09	1.235E+06
Fun_3	Mean	**3.000E+02**	1.460E+04	5.177E+03	1.330E+03	2.054E+04	2.582E+03	1.322E+04	7.108E+03
	Std	**0**	4.870E+03	1.978E+03	1.422E+03	1.609E+03	1.456E+03	1.919E+03	3.294E+03
Fun_4	Mean	**4.029E+02**	4.113E+02	4.371E+02	4.152E+02	2.327E+03	4.517E+02	9.498E+02	4.454E+02
	Std	**7.318E−01**	1.404E+01	2.588E+01	1.326E+01	1.109E+03	4.414E+01	1.183E+02	4.338E+01
Fun_5	Mean	**5.138E+02**	5.325E+02	5.336E+02	5.195E+02	6.516E+02	5.539E+02	6.066E+02	5.411E+02
	Std	**6.678E+00**	5.929E+00	8.586E+00	8.477E+00	2.283E+01	1.842E+01	1.293E+01	1.166E+01
Fun_6	Mean	**6.000E+02**	6.019E+02	6.194E+02	6.013E+02	6.802E+02	6.368E+02	6.531E+02	6.187E+02
	Std	**3.641E−03**	5.361E−01	3.433E+00	1.632E+00	1.148E+01	1.149E+01	6.684E+00	1.116E+01
Fun_7	Mean	**7.212E+02**	7.463E+02	7.699E+02	7.304E+02	8.689E+02	7.929E+02	8.219E+02	7.510E+02
	Std	**4.361E+00**	5.342E+00	1.620E+01	1.066E+01	1.123E+01	2.198E+01	1.093E+01	2.012E+01
Fun_8	Mean	**8.138E+02**	8.310E+02	8.328E+02	8.154E+02	8.872E+02	8.446E+02	8.569E+02	8.487E+02
	Std	6.451E+00	5.877E+00	7.905E+00	9.487E+00	6.666E+00	1.174E+01	**5.092E+00**	1.434E+01
Fun_9	Mean	**9.008E+02**	9.034E+02	1.054E+03	9.120E+02	2.420E+03	1.438E+03	1.650E+03	1.386E+03
	Std	3.170E+00	**1.274E+00**	9.794E+01	1.870E+01	1.347E+02	3.421E+02	1.378E+02	2.512E+02
Fun_{10}	Mean	1.689E+03	2.269E+03	2.309E+03	**1.676E+03**	2.938E+03	2.221E+03	2.438E+03	1.819E+03
	Std	2.679E+02	1.900E+02	1.416E+02	3.655E+02	**9.835E+01**	2.804E+02	1.351E+02	2.403E+02
Fun_{11}	Mean	**1.128E+03**	8.276E+03	2.099E+03	8.041E+05	4.336E+04	1.489E+03	2.244E+03	6.686E+03
	Std	**8.567E+01**	1.399E+04	1.153E+03	7.489E+05	1.854E+04	6.316E+02	6.513E+02	5.436E+03
Fun_{12}	Mean	**1.335E+04**	3.250E+06	6.213E+06	8.923E+05	1.426E+09	2.980E+06	1.122E+08	4.383E+06
	Std	**1.006E+04**	2.530E+06	6.859E+06	1.283E+06	6.669E+08	3.235E+06	4.805E+07	4.112E+06
Fun_{13}	Mean	1.174E+04	1.040E+04	1.697E+04	1.959E+04	1.204E+08	1.600E+04	1.789E+06	**1.039E+04**
	Std	**8.674E+03**	5.943E+03	2.235E+04	2.076E+04	1.288E+08	1.057E+04	1.672E+06	1.067E+04
Fun_{14}	Mean	8.953E+03	1.823E+03	1.955E+03	3.041E+03	1.772E+03	2.284E+03	**1.598E+03**	2.327E+03
	Std	9.106E+03	4.572E+02	5.646E+02	1.855E+03	**2.593E−01**	1.128E+03	4.631E+01	1.141E+03
Fun_{15}	Mean	8.573E+03	**3.724E+03**	7.660E+03	4.279E+03	1.221E+04	1.001E+04	8.128E+03	5.930E+03
	Std	9.212E+03	2.232E+03	5.213E+03	1.880E+03	**2.613E+01**	6.924E+03	1.710E+03	7.112E+03
Fun_{16}	Mean	1.723E+03	**1.657E+03**	1.793E+03	1.684E+03	2.498E+03	1.939E+03	2.121E+03	1.763E+03
	Std	1.106E+02	**3.679E+01**	6.895E+01	6.904E+02	2.530E+02	1.476E+02	6.658E+01	1.360E+02
Fun_{17}	Mean	1.769E+03	1.764E+03	1.775E+03	**1.759E+03**	2.108E+03	1.805E+03	1.844E+03	1.797E+03
	Std	4.819E+01	8.414E+00	**1.451E+01**	2.402E+01	8.168E+01	5.612E+01	2.749E+01	4.736E+01
Fun_{18}	Mean	**1.500E+04**	3.015E+04	9.704E+04	3.388E+04	4.194E+08	1.587E+04	4.557E+07	3.551E+04
	Std	**1.043E+04**	1.471E+04	1.864E+05	2.642E+04	4.165E+08	1.315E+04	4.447E+07	1.400E+04
Fun_{19}	Mean	1.022E+04	**4.062E+03**	1.481E+04	7.418E+03	2.727E+07	4.752E+04	4.045E+05	1.357E+04
	Std	1.047E+04	**1.971E+03**	5.506E+03	6.107E+03	6.126E+07	9.819E+04	4.055E+05	1.340E+04
Fun_{20}	Mean	2.082E+03	2.317E+03	2.167E+03	**2.059E+03**	2.424E+03	2.175E+03	2.248E+03	2.124E+03
	Std	6.553E+01	3.083E+01	4.001E+01	**2.962E+01**	9.617E+01	8.005E+01	3.723E+01	5.807E+01
Fun_{21}	Mean	2.292E+03	2.317E+03	2.274E+03	2.315E+03	2.429E+03	2.350E+03	2.298E+03	**2.207E+03**
	Std	4.728E+01	3.083E+01	4.974E+01	2.725E+01	2.471E+01	3.690E+01	3.560E+01	**1.165E+00**
Fun_{22}	Mean	2.301E+03	2.308E+03	2.357E+03	**2.300E+03**	3.756E+03	2.364E+03	2.870E+03	2.305E+03
	Std	**1.881E+00**	1.281E+01	2.835E+01	2.749E+01	5.696E+02	2.200E+02	1.452E+02	1.062E+01
Fun_{23}	Mean	**2.617E+03**	2.622E+03	2.650E+03	2.650E+03	2.790E+03	2.655E+03	2.723E+03	2.635E+03
	Std	**6.868E+00**	2.397E+01	8.830E+00	7.458E+00	1.352E+01	2.069E+01	1.132E+01	1.396E+01
Fun_{24}	Mean	2.732E+03	2.763E+03	2.772E+03	2.748E+03	2.947E+03	2.777E+03	2.913E+03	**2.731E+03**
	Std	5.507E+01	**5.499E+00**	1.863E+01	1.162E+01	9.912E+01	4.674E+01	8.106E+01	8.262E+01
Fun_{25}	Mean	**2.937E+03**	2.943E+03	2.973E+03	2.938E+03	4.066E+03	2.957E+03	3.400E+03	2.979E+03
	Std	3.052E+01	1.510E+01	**1.176E+01**	1.293E+01	4.631E+02	3.442E+02	2.022E+02	3.240E+01
Fun_{26}	Mean	3.065E+03	**2.960E+03**	3.154E+03	2.999E+03	4.593E+03	3.351E+03	3.962E+03	3.005E+03
	Std	3.207E+02	**3.647E+01**	5.267E+01	1.082E+02	4.178E+02	4.215E+02	1.689E+02	6.733E+01
Fun_{27}	Mean	3.104E+03	3.095E+03	3.104E+03	**3.079E+03**	3.354E+03	3.144E+03	3.199E+03	3.096E+03
	Std	2.853E+01	**2.265E+00**	3.124E+00	1.562E+01	8.754E+01	4.448E+01	2.921E+01	4.239E+00
Fun_{28}	Mean	3.380E+03	3.363E+03	3.377E+03	**3.281E+03**	3.861E+03	3.422E+03	3.770E+03	3.339E+03
	Std	9.584E+01	7.216E+01	6.115E+01	**1.096E+01**	9.255E+01	1.470E+02	4.815E+01	8.124E+01
Fun_{29}	Mean	3.217E+03	3.320E+03	3.248E+03	**3.186E+03**	3.825E+03	3.316E+03	3.458E+03	3.242E+03
	Std	5.361E+01	3.746E+01	**2.892E+01**	4.630E+01	1.313E+02	9.634E+01	4.545E+01	6.524E+01

Continued

Fun	Indice	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_{30}	Mean	4.128E+05	8.158E+05	9.715E+05	**4.932E+04**	8.282E+08	8.090E+05	5.892E+06	6.353E+05
	Std	5.887E+05	5.610E+05	4.488E+05	**3.632E+03**	4.281E+07	8.409E+05	2.267E+06	7.545E+04

Table 3. The Statistical results of the different algorithms on the CEC-2017 test suite in dimensions of 10. The bold refers to the best results.

experiment shows whether the proposed algorithm can maintain its search capacities if facing problems with large dimensions.

The statistics from this analysis can be seen in Table 3. CAO proved to be superior to the other algorithms on CEC-2017 benchmark functions for 25 out of 58 indicators and for 13 out of 29 functions. Through the observation of experimental results, we find that CAO algorithm is better than other algorithms in solving unimodal functions. For multimodal functions, CAO was the most effective for two out of seven functions in terms of mean indicators. For functions, Fun6 and Fun9, CAO can reach the global optimal value, whereas the other algorithms such as MA, LSO, GWO, HHO, WOA, EO, and MPA can only go as far as a lower precision. For Fun10 function, GWO ranked first according to the mean metric, and CAO ranked second. For standard deviation, HHO has the minimum value, but its mean is the maximum among the 8 algorithms. For Fun5, Fun6, and Fun7 functions, the mean metric and standard deviation results of CAO were better than those of other algorithms. Regarding hybrid functions, CAO demonstrated outstanding exploration capabilities. Compared to the other algorithms, CAO achieved superior results for the Fun11, Fun12, and Fun18 functions. MA obtains the optimal value in functions 15,16 and 19, GWO obtains the optimal value in functions 17,20,22,28,29 and 30, EO obtains the optimal value in functions 14,and EO obtains the optimal value in functions 13,21 and 24. As can be seen from the Table 3, CAO is second only to GWO in the complex hybrid functions, and the performance of EO is comparable to that of other functions. By expanding the spatial dimension to 30, 50 and 100 dimensions, we can get a conclusion similar to that of 10 dimensions from Table 4,5 and 6. CAO is far better than other algorithms in unimodal and multimodal functions, while only GWO is better than the other 6 algorithms except EO in the complex hybrid functions.

Boxplot analysis

The distribution of the thirty findings is depicted in Fig. 9 using boxplots of the various approaches applied to the CEC-2017 benchmark functions. Instances when the method was run 20 times are indicated by the (+) symbols outside the boxplots' edges. It indicates that the algorithm successfully searched the search space when the (+) signs are outside the lower border. The CAO algorithm's interim results outperform those of the eight competing algorithms, as illustrated in Fig. 9. However, especially for the whole set of functions, there is little variation between the top and lower bounds of CAO. There are fewer (+) signs in CAO, and there is no discernible difference between the upper and lower bounds. This suggests that even for difficult situations, CAO continues to produce satisfactory optimization outcomes. Additionally, the boxplot confirms the previously indicated analysis by reaffirming the consistency and robustness of CAO.

Convergence analysis

We performed a convergence analysis in order to assess the exploration and exploitation of the CAO and seven other comparison algorithms. The convergence curves of these eight algorithms on the CEC-2017 benchmark functions at 10 dimensions are shown in Fig. 10 and include unimodal (Fun1;F3), multimodal (Fun4; F5; F6), hybrid (Fun10; F16; F17), and composition functions (Fun23), which represent the differences in the iterative optimization of the algorithms on different functions. Other optimizers tend to approach a point of stagnation as the number of iterations grows, as seen in Fig. 10. However, even after other optimizers have reached the point of convergence, CAO continues to look for newer optimum values. This suggests that CAO can assist the channa argus in jumping out of the local optimum value and is more reliable in terms of exploration abilities when compared to the competing algorithms.

The reason is that CAO avoids using existing optimal values from the beginning, diligently searching for new spaces, discovering a new potential optimal value, evaluating it against the original optimal value, and then greedily retaining the one with the best optimization result. The convergence curves of different algorithms exhibit different potentials in global optimization, especially in the case of multimodal functions. Most notably, this method skips local optima at the beginning and then converges precisely to the global optimum. On the other hand, other algorithms require longer time to reach the point. For mixed and composite functions, CAO not only converges quickly in the early stages, but also becomes proficient in re exploring in the later stages.

Non-parametric statistical analysis

To determine if there was a statistically significant difference in accuracy between the optimization techniques used in the experiment, multiple statistical analyses were performed in this section, including the Wilcoxon Signed-Rank (WSR) test and Friedman's test.

To obtain an accurate evaluation of CAO, we ran the WSR on this experiment. The WSR assesses whether there is a significant statistical difference between the results of the proposed approach and other methods[67,68]. The WSR was conducted with a 95% level of confidence. The test results of the WSR with 29 functions and 20 runs are presented in Table 7. In these tables, a (-) denotes that CAO exhibits a 95% significance level in comparison to the other optimizers, whereas a (+) indicates the opposite. An (=) signifies that there is no

Fun	Indice	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_1	Mean	**5.669E+03**	1.845E+09	1.467E+10	3.736E+09	7.686E+10	2.154E+09	5.223E+10	8.624E+09
	Std	**5.731E+03**	4.048E+09	2.648E+09	2.363E+09	3.471E+09	9.008E+08	5.333E+09	4.913E+09
Fun_3	Mean	**6.810E+04**	1.768E+05	7.092E+04	5.971E+04	9.401E+04	2.546E+05	1.322E+04	7.108E+03
	Std	**3.771E+04**	2.830E+04	8.427E+03	1.130E+04	1.248E+02	8.335E+04	8.667E+04	3.294E+03
Fun_4	Mean	**4.647E+02**	4.113E+02	1.713E+03	6.752E+02	2.327E+03	4.517E+02	1.305E+04	9.537E+02
	Std	**7.612E+01**	1.404E+01	3.946E+02	1.728E+02	2.194E+04	9.992E+02	9.629E+02	3.715E+02
Fun_5	Mean	**5.682E+02**	7.260E+02	8.006E+02	6.132E+02	9.994E+02	8.002E+02	9.385E+02	7.793E+02
	Std	2.329E+01	**1.334E+01**	2.274E+01	2.934E+01	9.996E+00	4.021E+01	1.438E+01	4.668E+01
Fun_6	Mean	**6.008E+02**	6.176E+02	6.640E+02	6.133E+02	7.079E+02	6.771E+02	6.993E+02	6.553E+02
	Std	**1.103E+00**	2.538E+00	6.474E+00	5.301E+00	2.166E+00	1.010E+01	3.083E+00	8.793E+00
Fun_7	Mean	**7.954E+02**	1.052E+03	1.194E+03	8.841E+02	1.556E+03	1.264E+03	1.456E+03	1.582E+03
	Std	**1.359E+01**	2.539E+01	5.308E+01	5.377E+01	2.091E+01	1.036E+02	2.698E+01	1.533E+02
Fun_8	Mean	**8.604E+02**	1.037E+03	1.060E+03	9.059E+02	1.222E+03	1.049E+03	1.151E+03	1.060E+03
	Std	2.010E+01	1.821E+01	2.613E+01	3.321E+01	1.207E+01	5.232E+01	**1.332E+01**	4.182E+01
Fun_9	Mean	**1.089E+03**	2.282E+03	7.258E+03	2.536E+03	1.555E+04	9.873E+03	1.169E+04	8.263E+03
	Std	**1.571E+02**	4.571E+02	1.166E+03	1.184E+03	1.147E+03	3.224E+03	7.991E+02	1.212E+03
Fun_{10}	Mean	**4.303E+03**	8.690E+03	8.884E+03	5.551E+03	1.306E+04	7.028E+03	9.080E+03	5.591E+03
	Std	4.865E+02	3.112E+02	4.138E+02	1.377E+03	**3.279E+02**	8.414E+02	2.959E+02	7.012E+02
Fun_{11}	Mean	**2.781E+03**	9.752E+05	9.006E+03	6.053E+05	1.601E+04	2.396E+05	7.722E+03	4.398E+05
	Std	1.409E+03	2.793E+05	3.303E+03	4.490E+05	**2.822E+02**	2.299E+05	1.128E+03	1.821E+05
Fun_{12}	Mean	**4.127E+05**	9.833E+07	1.587E+09	1.016E+08	2.201E+10	3.268E+08	1.112E+10	1.448E+08
	Std	**4.983E+05**	4.473E+07	5.178E+08	1.035E+08	4.043E+09	2.639E+08	2.323E+09	1.267E+08
Fun_{13}	Mean	1.609E+04	1.660E+07	2.789E+08	1.473E+06	2.021E+10	4.456E+06	5.843E+09	**1.561E+07**
	Std	**1.286E+04**	9.887E+06	1.317E+08	2.123E+06	6.423E+09	5.379E+06	1.404E+09	3.491E+07
Fun_{14}	Mean	**8.953E+03**	4.661E+05	9.754E+05	4.538E+05	4.118E+08	1.753E+06	3.947E+06	6.032E+05
	Std	**9.106E+03**	5.337E+05	4.942E+05	6.372E+05	2.884E+07	2.482E+06	1.845E+06	6.241E+05
Fun_{15}	Mean	**7.911E+03**	3.662E+06	1.123E+07	8.260E+05	3.370E+09	1.389E+06	6.241E+08	8.134E+04
	Std	**6.828E+03**	3.790E+06	1.189E+07	1.657E+06	9.830E+08	1.639E+06	2.304E+08	2.974E+04
Fun_{16}	Mean	2.515E+03	**3.290E+03**	3.698E+03	2.802E+03	8.174E+03	4.118E+03	5.982E+03	3.124E+03
	Std	6.617E+02	**3.140E+02**	3.053E+02	4.655E+02	1.611E+03	4.898E+02	4.680E+02	3.677E+02
Fun_{17}	Mean	2.061E+03	2.494E+03	2.570E+03	2.094E+03	2.164E+04	2.683E+03	4.270E+03	2.616E+03
	Std	2.260E+02	1.718E+02	**2.223E+02**	1.502E+02	2.667E+03	3.725E+02	4.878E+02	2.932E+02
Fun_{18}	Mean	**5.195E+05**	7.604E+06	4.721E+06	2.944E+06	4.799E+08	1.012E+07	4.242E+07	3.212E+06
	Std	**3.181E+05**	4.671E+06	4.425E+06	6.571E+04	4.436E+08	1.098E+07	2.421E+07	4.004E+06
Fun_{19}	Mean	**8.143E+03**	9.386E+06	1.392E+07	7.310E+05	3.105E+09	1.688E+07	4.140E+08	9.339E+05
	Std	**5.391E+03**	8.308E+06	1.623E+07	8.109E+05	3.961E+08	1.220E+07	1.246E+08	7.916E+04
Fun_{20}	Mean	2.453E+03	2.768E+03	2.596E+03	**2.449E+03**	3.584E+03	2.876E+03	3.067E+03	2.826E+03
	Std	3.061E+02	1.588E+02	1.048E+02	**1.362E+02**	1.164E+03	2.239E+02	1.113E+02	1.858E+02
Fun_{21}	Mean	**2.357E+03**	2.528E+03	2.557E+03	2.420E+03	2.891E+03	2.616E+03	2.745E+03	2.552E+03
	Std	1.356E+01	**1.207E+01**	2.524E+01	3.911E+01	5.835E+01	5.551E+01	3.314E+01	3.122E+01
Fun_{22}	Mean	**4.937E+03**	2.754E+03	7.531E+03	6.951E+03	1.141E+04	7.402E+03	9.453E+03	6.439E+03
	Std	**1.709E+03**	1.281E+01	1.223E+03	2.445E+03	3.932E+02	2.324E+03	4.088E+02	1.087E+03
Fun_{23}	Mean	**2.716E+03**	2.873E+03	3.001E+03	2.793E+03	3.819E+03	3.124E+03	3.603E+03	2.933E+03
	Std	2.240E+01	**1.427E+01**	3.385E+01	4.730E+01	1.701E+02	1.168E+02	1.062E+02	5.534E+01
Fun_{24}	Mean	**2.897E+03**	3.038E+03	3.184E+03	3.004E+03	4.218E+03	3.244E+03	3.753E+03	3.069E+03
	Std	3.669E+01	**1.942E+01**	5.177E+01	7.437E+01	2.160E+02	6.537E+01	1.292E+02	4.071E+02
Fun_{25}	Mean	**2.888E+03**	3.028E+03	3.381E+03	3.008E+03	6.663E+03	3.137E+03	4.666E+03	3.603E+03
	Std	1.739E+00	5.055E+01	**7.598E+01**	5.719E+01	3.465E+02	6.207E+01	1.455E+02	3.294E+02
Fun_{26}	Mean	**4.480E+03**	5.987E+03	7.413E+03	4.744E+03	1.439E+04	8.461E+03	1.132E+04	6.984E+03
	Std	3.303E+02	**1.217E+02**	3.952E+02	3.776E+02	7.070E+01	7.483E+02	3.564E+02	7.408E+02
Fun_{27}	Mean	3.236E+03	3.247E+03	3.521E+03	**3.200E+03**	5.450E+03	3.472E+03	4.343E+03	3.323E+03
	Std	1.820E+01	1.061E+01	6.532E+01	**2.420E−04**	4.218E+02	1.937E+02	2.267E+02	5.550E+01
Fun_{28}	Mean	3.223E+03	3.526E+03	4.709E+03	**3.315E+03**	9.271E+03	3.604E+03	7.280E+03	5.827E+03
	Std	1.638E+01	8.359E+01	2.838E+02	**4.762E+01**	1.723E+02	1.052E+02	4.422E+02	1.292E+01
Fun_{29}	Mean	3.838E+03	4.457E+03	5.496E+03	**3.707E+03**	3.615E+04	5.392E+03	7.121E+03	4.576E+03
	Std	1.773E+02	2.080E+02	**3.835E+02**	2.333E+02	4.233E+04	5.192E+02	5.351E+02	4.142E+02

Continued

Fun	Indice	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_{30}	Mean	1.086E+05	3.960E+06	1.311E+08	**3.199E+06**	3.354E+09	5.758E+07	1.533E+09	6.303E+06
	Std	3.693E+03	5.610E+05	7.696E+07	**9.050E+06**	1.403E+09	4.650E+07	4.834E+08	8.074E+06

Table 4. The Statistical results of the different algorithms on the CEC-2017 test suite in dimensions of 30. The bold refers to the best results.

notable difference between CAO and the other optimizers. Table 7 provides the statistical results of the WSR in comparison to other algorithms when CAO is employed. Based on the data in Table 7, for the majority of functions, CAO demonstrates a clear superiority over all other methods.

To evaluate the effectiveness of our proposed method in comparison to its competitor algorithms, we employed the Friedman test in our study. This non-parametric test can detect significant differences in the performance of multiple optimizers, and we computed the ranks for a total of 29 functions across all techniques[69,70]. The radar chart in Fig. 11 presents the ranks of all the compared methods for each function, while Table 6 displays the average ranks. According to Table 8 and Fig. 13, CAO has the lowest average rank, indicating that it ranks first among all the algorithms, thus proving the efficacy of our method in quickly finding the global optimum for numerous problems.

Scalability analysis

Multiple choice variables are present in many real-world optimization issues, which can make them computationally difficult to solve. Scalability analysis can be used to determine the algorithm's constraints for large-scale problems and to comprehend how the number of variables (dimension) affects an optimization algorithm's performance[69,70]. The performance of the CAO algorithm is assessed in this section by means of a scalability test on a number of functions with varying dimensions, including the composition functions (Fun21, Fun23, Fun24,Fun27), the unimodal function (Fun1), the multimodal functions (Fun5, Fun6, Fun8, Fun9), and the hybrid functions (Fun16, Fun17, Fun20). The test uses the CEC-2017 test suite, which has dimensions of 10, 30, 50, and 100. Each algorithm can have up to 50,000 function evaluations, and 30 independent runs are carried out. The average fitness values of the various algorithms for dimensions 10, 30, 50, and 100 are displayed in Fig. 12.

With the exception of the Fun1 function, the CAO algorithm produced the best search results for the most of the functions that were studied. This experiment shows how the CAO algorithm can obtain higher convergence precision while managing the complexity of optimization brought on by an increase in the function's dimension.

Computational time analysis

The computational time of the CAO algorithm is examined in this subsection in relation to other optimizers on the CEC-2017 benchmark functions. Table 9 presents the results of 20 executions of each optimization strategy for each assigned function.

From the table, it can be seen that the computation time of CAO is slightly longer than some other algorithms because it requires more computing power to execute its operators. Nevertheless, CAO still outperforms MA, EO, MPA, and HHO in a relatively short period of time. Although it may take more time to complete, CAO has many advantages over other optimizers.

Sensitivity analysis

In optimization, the values given to an algorithm's parameters can have a significant impact on how effective the method is. It can be difficult to determine the ideal parameter settings, and it takes a lot of trial and error. Because parameter tuning entails running the algorithm several times with various parameter choices, it can be a time-consuming and computationally costly procedure. Additionally, various situations may require separate tuning runs because the ideal parameter settings can change based on the particular problem being solved. Sensitivity analysis is used to determine which parameters have the most effects on the algorithm's performance and to assess how changes to the algorithm's parameters affect the caliber of the solutions produced. In this study, we conducted a parameter sensitivity analysis to evaluate the effects of the original position inertia(OPI) on the performance of the algorithm. For four functions from the CEC-2017 benchmark test in 10 dim, we examined the effects of different control parameters and the values that are associated to them. We conducted 20 separate runs of the processes, each including up to 500 function evaluations. By changing each parameter's value separately while leaving the others constant, we were able to evaluate its impact. We display the sensitivity analysis's findings in terms of standard deviation (STD) and mean fitness (Mean). We ran the CAO algorithm for different values of OPI {0.1, 0.15, 0.2, 0.25, 0.3}. Table 10 show the results of the overall fitness values obtained by the algorithm for different OPI values. We observed that the performance of the algorithm is superior when OPI is set to 0.2. However, depending on the specific problem, experts may choose to adopt a different value for OPI.

Assessment of CAO on CEC-2020 benchmark test functions

To evaluate the performance of the CAO optimizer more precisely, we employed the CEC-2020 benchmark problems, which are widely used and complex. These test functions consist of composite, hybrid, multimodal, and rotated and shifted unimodal functions. The specific information of the CEC-2020 test suite is presented in Table 11, and Fig. 13 illustrates a two-dimensional format of some test functions. We analyzed the proposed CAO algorithm by comparing it with other well-known optimizers on benchmark functions. All tests were conducted

Fun	Indice	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_1	Mean	**1.275E+04**	1.831E+10	5.186E+10	1.303E+10	1.314E+11	1.134E+10	1.067E+11	5.967E+10
	Std	**1.845E+04**	3.118E+09	7.098E+09	4.751E+09	1.967E+09	2.461E+09	3.780E+09	1.740E+10
Fun_3	Mean	2.432E+05	3.665E+05	1.676E+05	2.014E+05	2.821E+05	3.110E+05	**2.023E+05**	3.532E+05
	Std	6.225E+04	3.895E+04	2.110E+04	5.403E+04	**9.628E+02**	1.127E+05	1.389E+04	5.737E+04
Fun_4	Mean	**5.572E+02**	2.025E+03	8.711E+03	1.929E+03	5.404E+04	3.071E+03	3.606E+04	7.046E+03
	Std	**4.204E+01**	4.212E+02	2.318E+02	8.062E+02	1.859E+03	8.824E+02	2.216E+03	5.993E+03
Fun_5	Mean	**6.294E+02**	1.001E+03	1.007E+03	7.779E+02	1.296E+03	1.071E+03	1.199E+03	1.056E+03
	Std	3.241E+01	2.819E+01	3.814E+01	5.213E+01	**1.098E+01**	8.043E+01	2.115E+01	7.009E+01
Fun_6	Mean	**6.034E+02**	6.361E+02	6.849E+02	6.293E+02	7.060E+02	6.917E+02	7.003E+02	6.682E+02
	Std	1.633E+00	3.702E+00	6.340E+00	7.250E+00	**1.090E+00**	8.738E+00	3.560E+00	8.387E+00
Fun_7	Mean	**9.223E+02**	1.716E+03	1.727E+03	1.120E+03	2.159E+03	1.787E+03	2.084E+03	2.744E+03
	Std	5.787E+01	1.032E+02	5.675E+01	8.066E+01	1.519E+01	9.421E+01	**1.466E+01**	2.493E+02
Fun_8	Mean	**9.545E+02**	1.300E+03	1.392E+03	1.077E+03	1.581E+03	1.360E+03	1.501E+03	1.362E+03
	Std	4.116E+01	3.026E+01	3.015E+01	7.546E+01	1.572E+01	5.738E+01	**1.368E+01**	7.731E+01
Fun_9	Mean	**1.733E+03**	1.011E+04	2.865E+04	1.073E+04	5.093E+04	3.618E+04	3.932E+04	2.154E+04
	Std	**6.843E+02**	2.524E+03	2.844E+03	4.688E+03	1.828E+03	1.025E+04	1.690E+03	4.410E+03
Fun_{10}	Mean	**6.995E+03**	1.531E+04	1.526E+04	9.883E+03	1.694E+04	1.265E+04	1.546E+04	9.573E+03
	Std	1.054E+03	5.285E+02	6.067E+02	3.098E+03	4.244E+02	1.212E+03	**3.625E+02**	8.860E+02
Fun_{11}	Mean	7.214E+04	3.953E+06	2.290E+04	8.173E+06	3.666E+04	**8.486E+03**	2.329E+05	1.421E+06
	Std	6.734E+04	1.169E+06	3.543E+03	7.857E+05	5.844E+03	3.530E+03	**2.100E+03**	6.177E+05
Fun_{12}	Mean	**3.577E+06**	1.612E+09	1.001E+10	3.337E+09	1.210E+11	2.096E+09	8.243E+10	1.878E+09
	Std	**2.618E+06**	4.872E+08	2.754E+09	2.453E+09	1.659E+10	7.891E+08	6.186E+09	9.872E+08
Fun_{13}	Mean	**5.502E+03**	1.664E+08	2.067E+09	1.716E+08	6.858E+10	2.263E+08	4.675E+10	1.739E+08
	Std	**4.829E+03**	6.310E+06	9.234+08	1.235E+08	1.835E+10	1.666E+08	8.361E+09	2.177E+08
Fun_{14}	Mean	**2.288E+05**	3.217E+06	7.797E+06	2.269E+06	3.002E+08	4.560E+06	6.985E+07	4.389E+06
	Std	**1.708E+05**	2.102E+06	4.576E+06	4.572E+06	1.543E+08	3.787E+06	2.728E+07	4.238E+06
Fun_{15}	Mean	**9.740E+03**	3.062E+07	4.277E+08	9.543E+06	1.903E+10	1.664E+07	7.133E+09	2.197E+06
	Std	**6.648E+03**	1.989E+07	2.015E+08	1.233E+07	4.431E+09	1.709E+07	1.298E+09	2.681E+06
Fun_{16}	Mean	**3.194E+03**	5.353E+03	5.474E+03	3.427E+03	1.389E+04	6.176E+03	9.359E+03	4.507E+03
	Std	**3.097E+02**	3.342E+02	4.562E+02	3.754E+02	2.755E+03	7.690E+02	7.462E+02	5.984E+02
Fun_{17}	Mean	2.952E+03	4.220E+03	4.489E+03	**2.913E+03**	1.021E+05	4.507E+03	9.655E+03	4.479E+03
	Std	3.172E+02	2.743E+02	4.079E+02	**2.712E+02**	2.926E+04	4.197E+02	2.149E+02	4.490E+02
Fun_{18}	Mean	**2.083E+06**	2.713E+07	3.631E+07	1.577E+07	7.967E+08	4.106E+07	1.652E+08	1.273E+07
	Std	**1.321E+06**	1.129E+07	1.284E+07	1.682E+07	2.463E+08	3.126E+07	3.469E+07	7.304E+06
Fun_{19}	Mean	**1.871E+04**	1.905E+07	1.678E+08	1.002E+07	8.853E+09	6.310E+07	3.311E+09	3.428E+06
	Std	**1.077E+04**	1.085E+07	9.345E+07	2.045E+07	1.489E+09	6.642E+07	8.494E+08	4.666E+06
Fun_{20}	Mean	**3.104E+03**	4.166E+03	3.773E+03	3.255E+03	4.729E+03	3.998E+03	4.299E+03	3.504E+03
	Std	3.527E+02	1.986E+02	2.164E+02	5.199E+02	**4.950E+01**	3.944E+02	1.672E+02	3.468E+02
Fun_{21}	Mean	**2.425E+03**	2.790E+03	2.867E+03	2.568E+03	3.530E+03	2.991E+03	3.231E+03	2.856E+03
	Std	**2.371E+01**	2.465E+01	4.740E+01	5.696E+01	1.043E+02	9.804E+01	3.610E+01	9.304E+01
Fun_{22}	Mean	**8.204E+03**	1.670E+04	1.690E+04	1.211E+04	1.875E+04	1.452E+04	1.683E+04	1.157E+04
	Std	1.031E+03	**3.385E+02**	4.674E+02	3.114E+03	4.156E+02	9.224E+02	3.854E+02	8.801E+02
Fun_{23}	Mean	**2.894E+03**	3.230E+03	3.512E+03	3.047E+03	4.934E+03	3.853E+03	4.422E+03	3.397E+03
	Std	5.296E+01	**1.956E+01**	5.994E+01	5.711E+01	1.986E+02	2.119E+02	1.081E+02	9.741E+01
Fun_{24}	Mean	**3.071E+03**	3.370E+03	3.725E+03	3.354E+03	5.588E+03	3.801E+03	4.920E+03	3.579E+03
	Std	4.846E+01	**4.105E+01**	7.928E+01	1.095E+02	4.534E+02	1.538E+02	3.067E+02	1.181E+02
Fun_{25}	Mean	**3.073E+03**	4.425E+03	7.745E+03	3.980E+03	1.887E+04	4.440E+03	1.473E+04	9.876E+03
	Std	**2.376E+01**	3.629E+02	6.800E+02	4.462E+02	4.535E+02	3.830E+02	7.931E+02	2.584E+02
Fun_{26}	Mean	**5.381E+03**	8.877E+03	1.296E+04	7.143E+03	1.924E+04	1.472E+04	1.672E+04	1.050E+04
	Std	5.278E+02	2.846E+02	9.586E+02	6.886E+02	**1.032E+02**	1.599E+03	4.428E+02	8.735E+02
Fun_{27}	Mean	3.426E+03	3.596E+03	4.960E+03	**3.200E+03**	8.694E+03	4.421E+03	7.016E+03	3.931E+03
	Std	9.603E+01	6.505E+01	4.028E+02	**2.730E−04**	1.028E+03	4.375E+02	4.337E+02	2.384E+02
Fun_{28}	Mean	**3.319E+03**	6.553E+03	8.247E+03	3.618E+03	1.904E+04	5.529E+03	1.301E+04	9.335E+03
	Std	**3.026E+01**	1.181E+03	6.629E+02	7.853E+02	7.351E+02	4.314E+02	4.476E+02	1.353E+03
Fun_{29}	Mean	**4.103E+03**	5.974E+03	9.849E+03	4.396E+03	1.061E+06	8.874E+03	4.870E+04	6.853E+03
	Std	**3.362E+02**	4.350E+02	1.311E+03	4.613E+02	9.642E+05	1.856E+03	2.104E+04	1.008E+03

Continued

Fun	Indice	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_{30}	Mean	**1.331E+06**	1.192E+08	5.009E+08	1.493E+07	1.450E+10	3.210E+08	6.982E+09	5.710E+07
	Std	**3.926E+05**	4.297E+07	1.606E+08	1.457E+07	4.505E+09	1.333E+08	1.260E+09	3.764E+07

Table 5. The Statistical results of the different algorithms on the CEC-2017 test suite in dimensions of 50. The bold refers to the best results.

using 10 and 20 dimensions, and each optimizer was given 500 iterations over 20 runs. The parameters for each algorithm remained consistent and are detailed in Table 2.

Tables 12 and 13 summarizes the results of each algorithm, including the average and standard deviation of the 20 runs.

From the table, we can observe that CAO ranks first among 6 functions in the 10 dimensional and 9 functions in the 20 dimensional problems. In addition, CAO achieved the second best result among other functions that were not ranked first. Figures 14, 15, 16, and 17 depict the convergence curve and boxplot of the eight algorithms for different CEC-2020 test suits. The figures show that the CAO algorithm converges rapidly to the optimal solution and achieves lower fitness values, as indicated by the small boxplot height. This demonstrates the robustness and steadiness of the CAO algorithm.

The Friedman test results for eight distinct algorithms are shown in Tables 14 and 15. According to the table, CAO outperformed the other eight algorithms used for comparison in terms of efficiency and received the top score. Therefore, the Friedman rank tests prove that the proposed CAO is effective and reliable. In addition, Tables 16 and 17 provide a comparison of the optimizers using the Wilcoxon rank test for 10 and 20 dimensions. A (-)sign indicates that the CAO algorithm is more efficient than its eight competitors, a (+)sign implies that the eight competitors are more efficient than CAO, and a (=) sign denotes equal performance between CAO and the competitor algorithm. Tables 16 and 17 presents the statistical results of all 20 runs and reveals that the CAO algorithm performed remarkably better than its competitors.

Engineering optimization test problems

In this section we assess the CAO algorithm's performance on five real-world engineering optimization problems, encompassing both constrained and unconstrained issues. Multiple inequality constraints in the constrained issues call for a constraint-handling strategy, like the death penalty approach[71–73]. We used eight more algorithms to evaluate the CAO algorithm's performance against that of other optimization techniques, MA, LSO, GWO, WOA, HHO, EO, and MPA. For every engineering challenge, we ran the CAO and the other optimization algorithms 20 times independently to guarantee unbiased assessments. There are several inequality constraints in these constrained issues. A penalty function technique is used to deal with constraint breaches; if any constraints are broken, the algorithm is penalized heavily. The parameter configurations are still the same as those from earlier studies. The specifics of the confined and unconstrained engineering benchmark tasks are as follows.

Pressure vessel design problem

As shown in Fig. 18, the main goal of this study is to reduce the overall cost of the raw materials needed for pressure vessel design. Finding the ideal values for the four decision variables—the thickness of the head (Th), the thickness of the shell (Ts), the radius of entrance (R), and the length of the cylindrical section (L), excluding the head—is our goal in order to do this. The following is the mathematical formulation for this optimization problem:

Consider variable $\mathbf{x} = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L]$.

Minimize $f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$.

Subject to $g_1(\mathbf{x}) = -x_1 + 0.0193x_3$.

$$g_2(\mathbf{x}) = -x_2 + 0.00954x_3 \leq 0.$$

$$g_3(\mathbf{x}) = -\pi x_3^2 x_4 - 4/3\pi x_3^3 + 1296{,}000 \leq 0.$$

$$g_4(\mathbf{x}) = x_4 - 240 \leq 0.$$

Variable range $0 \leq x_1, x_2 \leq 99$.

$$10 \leq x_3, x_4 \leq 200.$$

We used the suggested CAO algorithm in conjunction with other competing methods to ascertain the ideal values for the pressure vessel design factors. Table 18, which displays the simulation results, demonstrates that the CAO method performs better than the other competing algorithms by offering a more optimal computation of the objective function. Furthermore, statistical data for each algorithm are shown in Table 19, emphasizing the CAO algorithm's superiority in terms of the mean, worst, and standard deviation of the best solutions. We have incorporated a convergence curve, constraint. The CAO algorithm's gradual convergence to the ideal solution is shown in this figure.

Fun	Indice	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_1	Mean	**3.730E+08**	1.238E+11	1.761E+11	6.733E+10	2.898E+11	7.833E+10	2.671E+11	2.375E+11
	Std	**4.041E+08**	1.131E+10	9.255E+09	1.054E+10	3.836E+09	8.924E+09	4.185E+09	2.228E+10
Fun_3	Mean	7.415E+05	8.489E+05	3.616E+05	1.016E+06	3.664E+05	9.828E+05	**3.544E+05**	9.166E+05
	Std	1.469E+05	6.084E+04	3.038E+04	3.094E+05	**1.574E+02**	1.213E+05	5.767E+03	2.113E+05
Fun_4	Mean	**9.145E+02**	1.704E+04	3.076E+04	8.636E+03	1.515E+05	1.341E+04	1.036E+05	4.915E+04
	Std	**9.898E+01**	3.486E+03	3.871E+03	2.397E+03	5.539E+03	2.842E+03	7.146E+03	1.536E+04
Fun_5	Mean	**9.133E+02**	1.819E+03	1.951E+03	1.295E+03	2.275E+03	1.828E+03	2.118E+03	1.992E+03
	Std	7.567E+01	5.281E+01	6.391E+01	5.684E+01	**2.555E+01**	9.701E+01	1.637E+01	1.474E+02
Fun_6	Mean	**6.142E+02**	6.722E+02	7.031E+02	6.463E+02	7.186E+02	7.064E+02	7.146E+02	6.781E+02
	Std	4.936E+00	4.011E+00	4.115E+00	3.885E+00	**6.927E−01**	1.100E+01	2.107E+00	4.848E+00
Fun_7	Mean	**1.466E+03**	5.501E+03	3.619E+03	2.119E+03	4.193E+03	3.759E+03	4.044E+03	5.726E+03
	Std	1.052E+02	3.306E+02	1.279E+02	1.420E+02	2.522E+01	1.340E+02	5.387E+02	5.387E+02
Fun_8	Mean	**1.190E+03**	2.097E+03	2.332E+03	1.618E+03	2.712E+03	2.303E+03	2.614E+03	2.367E+03
	Std	7.758E+01	6.629E+01	5.029E+01	1.178E+02	1.494E+01	1.042E+02	**2.142E+01**	1.346E+02
Fun_9	Mean	**9.000E+03**	5.140E+04	7.820E+04	4.543E+04	9.522E+04	7.081E+04	8.021E+04	4.881E+04
	Std	**3.389E+03**	6.784E+03	5.129E+03	1.372E+04	3.706E+03	1.656E+04	3.128E+03	7.552E+03
Fun_{10}	Mean	**1.425E+04**	3.280E+04	3.260E+04	2.629E+04	3.476E+04	2.873E+04	3.278E+04	2.027E+04
	Std	1.030E+03	7.422E+02	6.130E+02	6.886E+03	2.756E+02	1.415E+03	**8.217E+02**	1.596E+03
Fun_{11}	Mean	4.692E+06	8.290E+07	5.272E+05	1.363E+06	5.721E+05	**3.687E+07**	2.823E+05	1.156E+07
	Std	2.349E+06	1.352E+07	3.989E+05	2.165E+06	2.253E+03	3.552E+07	**3.221E+04**	3.692E+06
Fun_{12}	Mean	**4.433E+07**	1.989E+10	7.077E+10	1.877E+10	2.472E+11	1.578E+10	2.005E+11	4.414E+10
	Std	**1.804E+07**	4.167E+09	1.164E+10	5.875E+09	1.439E+10	2.864E+09	2.545E+09	1.187E+10
Fun_{13}	Mean	**8.000E+03**	1.395E+09	1.023E+10	2.090E+09	6.114E+10	9.438E+08	4.536E+10	2.343E+09
	Std	**5.293E+03**	3.793E+08	3.096E+09	1.782E+09	3.993E+09	3.180E+08	2.839E+09	1.449E+09
Fun_{14}	Mean	**1.792E+06**	5.101E+07	2.009E+07	9.564E+06	2.378E+08	1.672E+07	5.177E+07	3.051E+07
	Std	**1.354E+06**	1.621E+07	7.624E+06	5.052E+06	5.087E+07	6.783E+06	1.139E+07	1.167E+07
Fun_{15}	Mean	**5.472E+03**	2.090E+08	2.841E+09	3.459E+08	3.733E+10	1.951E+08	2.482E+10	4.200E+08
	Std	**5.153E+03**	8.165E+07	2.015E+08	3.862E+08	2.465E+09	1.479E+08	2.607E+09	3.907E+08
Fun_{16}	Mean	**5.495E+03**	1.155E+04	1.289E+04	8.141E+03	3.300E+04	1.619E+04	2.435E+04	9.287E+03
	Std	**7.516E+02**	4.404E+02	7.359E+02	2.019E+03	3.831E+03	2.819E+03	9.906E+02	1.303E+03
Fun_{17}	Mean	4.632E+03	8.737E+03	1.853E+04	**6.592E+03**	3.854E+07	1.444E+04	7.667E+06	1.070E+04
	Std	4.440E+02	2.218E+02	1.650E+04	**1.418E+03**	2.747E+07	1.172E+04	4.458E+06	1.007E+04
Fun_{18}	Mean	**4.642E+06**	7.375E+07	2.965E+07	1.186E+07	9.637E+08	1.384E+07	1.996E+08	5.380E+07
	Std	**2.354E+06**	2.583E+07	1.196E+07	9.704E+06	2.954E+08	7.244E+06	6.070E+07	1.831E+07
Fun_{19}	Mean	**4.517E+03**	2.792E+08	2.272E+09	3.117E+08	3.686E+10	1.926E+08	2.397E+10	4.818E+08
	Std	**2.245E+03**	1.631E+08	1.020E+09	2.268E+08	2.986E+09	1.172E+08	1.958E+09	4.215E+08
Fun_{20}	Mean	**4.693E+03**	7.979E+03	7.141E+03	6.214E+03	9.115E+03	6.904E+03	7.929E+03	6.563E+03
	Std	5.540E+02	2.580E+02	4.440E+02	1.332E+03	**3.510E+02**	7.520E+02	2.214E+02	3.481E+02
Fun_{21}	Mean	**2.696E+03**	3.672E+03	3.870E+03	3.222E+03	5.362E+03	4.360E+03	4.857E+03	3.944E+03
	Std	**5.028E+01**	5.574E+01	9.542E+01	7.814E+01	2.349E+02	2.235E+02	1.187E+02	1.821E+02
Fun_{22}	Mean	**1.683E+04**	3.497E+04	3.498E+04	2.539E+04	3.761E+04	3.132E+04	3.498E+04	2.258E+04
	Std	1.163E+03	**8.374E+02**	8.064E+02	5.242E+03	1.763E+02	1.241E+03	4.275E+02	1.269E+03
Fun_{23}	Mean	**3.308E+03**	4.075E+03	4.958E+03	3.847E+03	8.052E+03	5.245E+03	6.960E+03	4.151E+03
	Std	7.214E+01	**5.627E+01**	1.909E+02	1.249E+02	5.240E+02	2.378E+02	4.994E+02	1.320E+02
Fun_{24}	Mean	**3.809E+03**	4.197E+03	6.919E+03	4.676E+03	1.325E+04	6.575E+03	1.081E+04	5.370E+03
	Std	1.134E+02	**7.010E+02**	2.881E+02	1.304E+02	1.091E+03	4.196E+02	6.395E+02	3.536E+02
Fun_{25}	Mean	**3.557E+03**	2.133E+04	1.684E+04	7.651E+03	3.369E+04	8.616E+03	2.769E+04	2.943E+04
	Std	**8.210E+01**	3.070E+03	1.599E+03	9.899E+02	1.238E+03	8.406E+02	8.952E+02	4.754E+03
Fun_{26}	Mean	**1.065E+04**	2.067E+04	3.462E+04	1.982E+04	5.803E+04	3.633E+04	5.147E+04	2.650E+04
	Std	8.390E+02	9.260E+02	1.406E+03	1.568E+03	**6.288E+02**	3.011E+03	1.230E+03	2.114E+03
Fun_{27}	Mean	3.536E+03	4.396E+03	8.086E+03	**3.200E+03**	1.627E+04	6.116E+03	1.373E+04	4.394E+03
	Std	6.445E+01	2.512E+02	7.738E+02	**3.190E−04**	1.637E+03	1.081E+03	1.050E+03	2.890E+02
Fun_{28}	Mean	**3.752E+03**	2.026E+04	2.206E+04	4.409E+03	3.305E+04	1.223E+04	2.847E+04	2.526E+04
	Std	**1.154E+02**	1.542E+03	1.432E+03	2.778E+03	4.501E+02	1.106E+03	6.801E+02	2.407E+03
Fun_{29}	Mean	**6.122E+03**	1.208E+04	2.549E+04	7.640E+03	5.051E+06	1.894E+04	4.085E+05	1.508E+04
	Std	**3.083E+02**	7.557E+02	6.277E+03	8.900E+02	1.967E+06	2.959E+03	1.734E+05	4.504E+03

Continued

Fun	Indice	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_{30}	Mean	**4.432E+04**	6.004E+08	9.732E+09	1.170E+09	5.624E+10	1.515E+09	4.032E+10	6.337E+08
	Std	**1.979E+04**	1.555E+08	2.222E+09	1.517E+09	4.107E+09	6.181E+08	1.886E+09	3.915E+08

Table 6. The Statistical results of the different algorithms on the CEC-2017 test suite in dimensions of 100. The bold refers to the best results.

Fig. 9. Boxplot of CAO and competitor algorithms for different CEC-2017 test functions (Dim = 10).

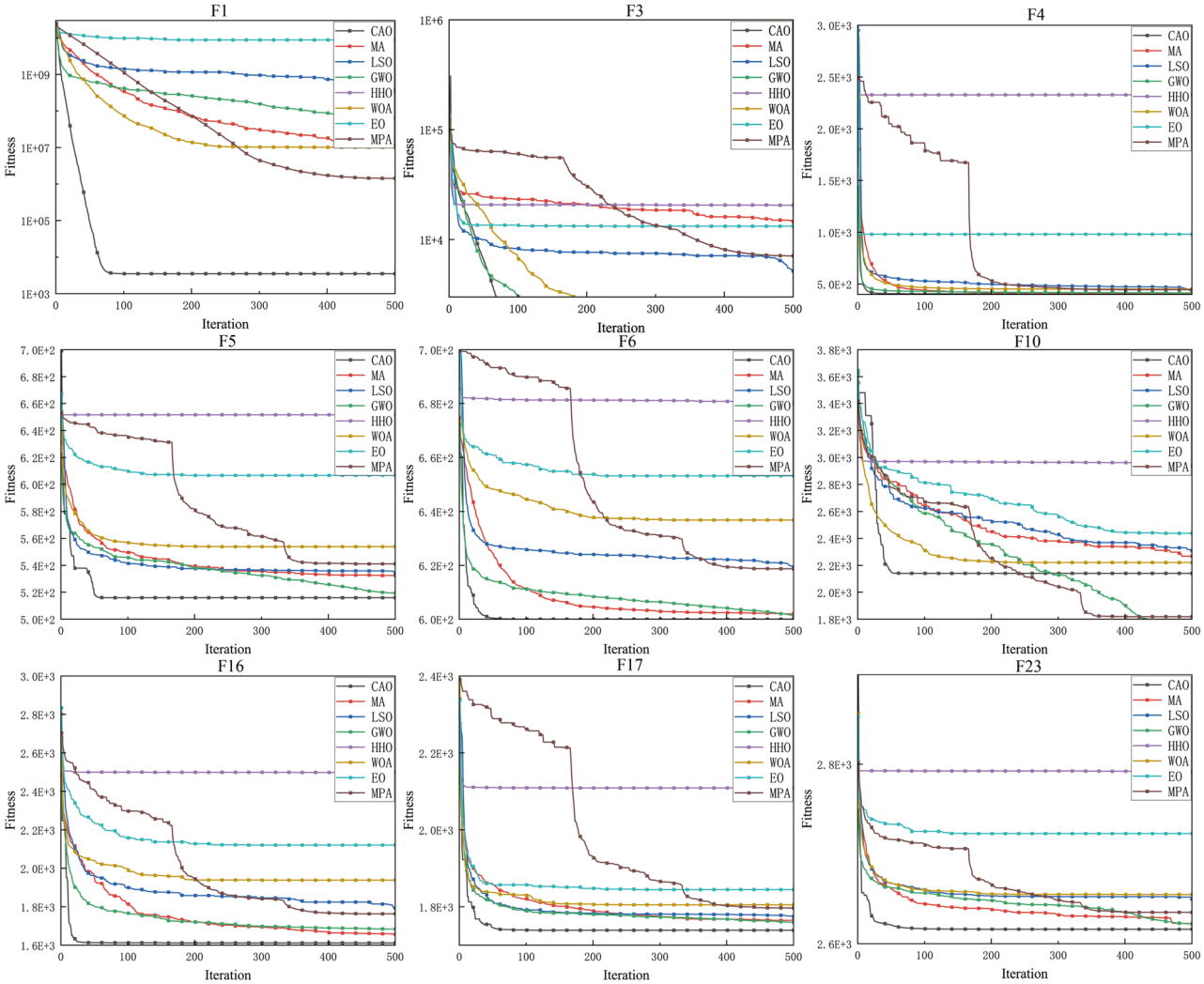

Fig. 10. Convergence curves of CAO and competitor algorithms for different CEC-2017 test functions (Dim = 10).

Welded beam design

The welded beam design problem, a form of composite beam, is one of the most well-known engineering problems used to assess the algorithm's performance. A weld is produced by welding multiple components together with molten metal in the way depicted in Fig. 19.

The best goal is to lower the total cost of the beam by selecting the optimal four design parameters: bar thickness (b), bar length (l), weld thickness (t), and bar height (h). One way to express the optimization model is as.

Already know: $\boldsymbol{x} = [x_1, x_2, x_3, x_4] = [h, m, t, b]$,

Minimize: $f(\boldsymbol{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$,

Variables range: $0.1 \le x_1 \le 2, 0.1 \le x_2 \le 10, 0.1 \le x_3 \le 10, 0.1 \le x_4 \le 2$,

Restriction condition:
$$\begin{cases} h_1(\boldsymbol{x}) = \tau(x) - \tau_{\max} \le 0, h_2(\boldsymbol{x}) = \sigma(x) - \sigma_{\max} \le 0, \\ h_3(\boldsymbol{x}) = \delta(x) - \delta_{\max} \le 0, h_4(\boldsymbol{x}) = x_1 - x_4 \le 0, \\ h_5(\boldsymbol{x}) = P - P_C(x) \le 0, h_6(\boldsymbol{x}) = 0.125 - x_1 \le 0, \\ h_7(\boldsymbol{x}) = 1.1047x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0, \end{cases}$$

where

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{R} + (\tau'')^2},\ \tau' = \frac{P}{\sqrt{2}x_1x_2},\ \tau'' = \frac{MR}{J},$$

$$M = P\left(L + \frac{x_2}{2}\right),\ R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},$$

Fun	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_1	8.858E−05/-	8.858E−05/-	1.034E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	1.700E−04/-
Fun_3	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-
Fun_4	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-
Fun_5	1.204E−04/-	1.034E−04/-	3.330E−02/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-
Fun_6	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-
Fun_7	8.858E−05/-	8.858E−05/-	4.500E−03/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	1.204E−04/-
Fun_8	8.858E−05/-	8.858E−05/-	8.519E−01/=	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-
Fun_9	1.700E−03/-	8.858E−05/-	5.934E−04/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-
Fun_{10}	8.858E−05/-	8.858E−05/-	4.115E−01/=	8.858E−05/-	5.167E−04/-	8.858E−05/-	1.454E−01/=
Fun_{11}	8.858E−05/-	1.890E−04/-	8.858E−05/-	8.858E−05/-	8.918E−04/-	8.858E−05/-	8.858E−05/-
Fun_{12}	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	1.204E−04/-	8.858E−05/-	8.858E−05/-
Fun_{13}	3.703E−01/=	9.702E−01/=	2.043E−01/=	8.858E−05/-	2.043E−01/=	8.858E−05/-	5.755E−01/=
Fun_{14}	8.948E−04/+	1.690E−02/+	4.000E−02/+	1.200E−03/+	6.400E−03/+	2.191E−04/+	1.520E−02/+
Fun_{15}	1.084E−01/=	9.108E−01/=	1.672E−01/=	9.300E−02/=	3.135E−01/=	5.503E−01/=	3.317E−01/=
Fun_{16}	3.660E−02/+	3.330E−02/-	2.322E−01/+	8.858E−05/-	1.034E−04/-	8.858E−05/-	3.703E−01/=
Fun_{17}	8.813E−01/=	5.016E−01/=	5.755E−01/=	8.858E−05/-	6.704E−02/=	3.385E−04/-	1.005E−01/=
Fun_{18}	1.900E−03/+	2.500E−03/-	2.280E−02/+	8.858E−05/-	7.652E−01/=	8.858E−05/-	3.902E−04/=
Fun_{19}	3.660E−02/+	1.084E−01/=	3.317E−01/=	1.034E−04/-	2.760E−02/-	1.204E−04/-	3.507E−01/=
Fun_{20}	8.858E−05/-	6.806E−04/=	2.790E−01/=	8.858E−05/-	4.000E−03/-	1.034E−04/-	2.060E−02/-
Fun_{21}	1.370E−02/-	2.627E−01/=	4.500E−03/+	8.858E−05/-	1.000E−03/-	8.519E−01/=	3.902E−04/+
Fun_{22}	1.370E−02/-	8.858E−05/-	2.760E−02/-	8.858E−05/-	1.300E−03/-	8.858E−05/-	2.500E−03/-
Fun_{23}	4.500E−03/-	8.858E−05/-	4.000E−02/-	8.858E−05/-	1.401E−04/-	8.858E−05/-	2.932E−04/-
Fun_{24}	8.858E−05/-	3.902E−04/-	3.905E−01/=	8.858E−05/-	1.300E−03/-	1.034E−04/-	6.200E−02/=
Fun_{25}	1.169E−01/=	4.493E−04/-	8.228E−01/=	8.858E−05/-	6.200E−02/=	8.858E−05/-	5.167E−04/-
Fun_{26}	3.905E−01/=	1.520E−02/-	9.108E−01/=	8.858E−05/-	2.060E−02/-	1.401E−04/-	5.257E−01/=
Fun_{27}	9.702E−01/=	1.520E−02/+	1.200E−03/+	8.858E−05/-	4.000E−03/-	1.034E−04/-	9.702E−01/=
Fun_{28}	4.553E−01/=	3.703E−02/-	1.370E−02/+	8.858E−05/-	2.043E−01/-	8.858E−05/-	1.169E−01/=
Fun_{29}	9.405E−01/=	1.240E−02/-	6.740E−02/=	8.858E−05/-	1.034E−04/-	8.858E−05/-	1.354E−01/=
Fun_{30}	6.200E−05/=	1.200E−03/-	1.200E−03/+	8.858E−05/-	7.930E−02/=	8.858E−05/-	1.913E−01/=
+/=/−	4/9/16	2/6/21	6/10/13	1/1/27	1/6/22	1/2/26	2/12/15

Table 7. Significance comparisons of WSR test.

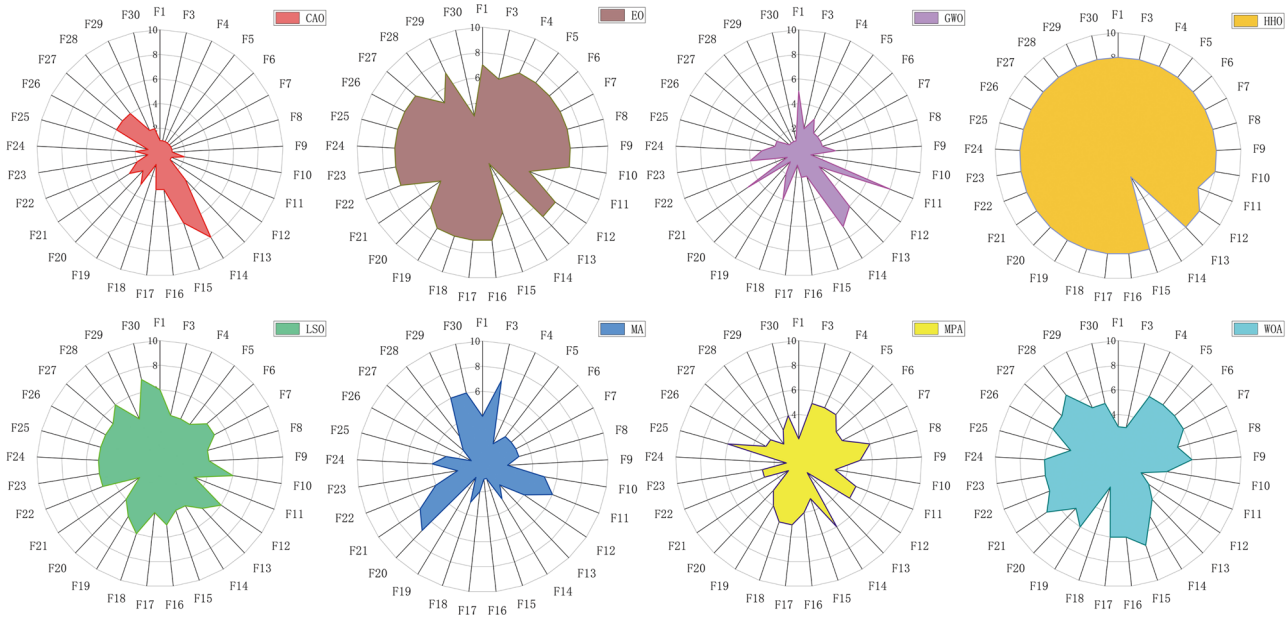

Fig. 11. Radar maps of the eight algorithms on 29 benchmark functions.

Fun	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_1	1	4	6	5	8	3	7	2
Fun_3	1	7	4	2	8	3	6	5
Fun_4	1	2	4	3	8	6	7	5
Fun_5	1	3	4	2	8	6	7	5
Fun_6	1	3	5	2	8	6	7	4
Fun_7	1	3	5	2	8	6	7	4
Fun_8	1	3	4	2	8	5	7	6
Fun_9	1	2	4	3	8	6	7	5
Fun_{10}	2	5	6	1	8	4	7	3
Fun_{11}	1	6	3	8	7	2	4	5
Fun_{12}	1	4	6	2	8	3	7	5
Fun_{13}	3	2	5	6	8	4	7	1
Fun_{14}	8	3	4	7	2	5	1	6
Fun_{15}	6	1	4	2	8	7	5	3
Fun_{16}	3	1	5	2	8	6	7	4
Fun_{17}	3	2	4	1	8	6	7	5
Fun_{18}	1	3	6	4	8	2	7	5
Fun_{19}	3	1	5	2	8	6	7	4
Fun_{20}	2	7	4	1	8	5	6	3
Fun_{21}	3	6	2	5	8	7	4	1
Fun_{22}	2	4	5	1	8	6	7	3
Fun_{23}	1	2	5	4	8	6	7	3
Fun_{24}	2	4	5	3	8	6	7	1
Fun_{25}	1	3	5	2	8	4	7	6
Fun_{26}	4	1	5	2	8	6	7	3
Fun_{27}	4	2	5	1	8	6	7	3
Fun_{28}	4	3	6	1	8	7	5	2
Fun_{29}	2	6	4	1	8	5	7	3
Fun_{30}	2	6	7	1	8	5	3	4
Mean rank	**2.276**	**3.414**	**4.724**	**2.690**	**7.759**	**5.138**	**6.241**	**3.759**
Result	**1**	**3**	**5**	**2**	**8**	**6**	**7**	**4**

Table 8. Friedman ranks for 29 functions for CAO and seven other optimizers.

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\},\ \sigma(\boldsymbol{x}) = \frac{6PL}{x_4x_3^2},\ \delta(\boldsymbol{x}) = \frac{6PL^3}{Ex_4x_3^2},$$

$$P_C(x) = \frac{4.013\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$$

$P = 6000\,\text{lb}$, $L = 14\,\text{in}$, $\delta_{\max} = 0.25\,\text{in}$, $E = 3 \times 10^6\,\text{psi}$,
$G = 12 \times 10^6\,\text{psi}$, $\tau_{\max} = 13600\,\text{psi}$, $\sigma_{\max} = 30000\,\text{psi}$.

After 20 runs, all results are gathered in Table 20. The results for CAO are the best across all four indicators. CAO is more stable, as seen by its superior average value and STD. The variable values for each algorithm's optimal result are displayed in Table 21.

Three bar truss design

The three-bar truss design seeks to achieve the lowest weight feasible while building a truss with multiple limitations, including deflection, buckling, and stress. As shown in Fig. 20, this optimization problem has two design parameters, and. It is displayed mathematically as.

Consider: $\boldsymbol{x} = [x_1, x_2] = [A_1, A_2]$,
Minimize: $f(\boldsymbol{x}) = \left(2\sqrt{2x_1} + x_2\right) * l$,

$$\text{Restriction condition: } \begin{cases} h_1(\boldsymbol{x}) = \frac{\sqrt{2x_1}+x_2}{\sqrt{2x_1^2+2x_1x_2}}P - \sigma \le 0, \\ h_2(\boldsymbol{x}) = \frac{x_2}{\sqrt{2x_1^2+2x_1x_2}}P - \sigma \le 0, \\ h_3(\boldsymbol{x}) = \frac{1}{\sqrt{2x_1^2+x_1}}P - \sigma \le 0, \end{cases}$$

Variables range: $0 \le x_1, x_2 \le 1$,

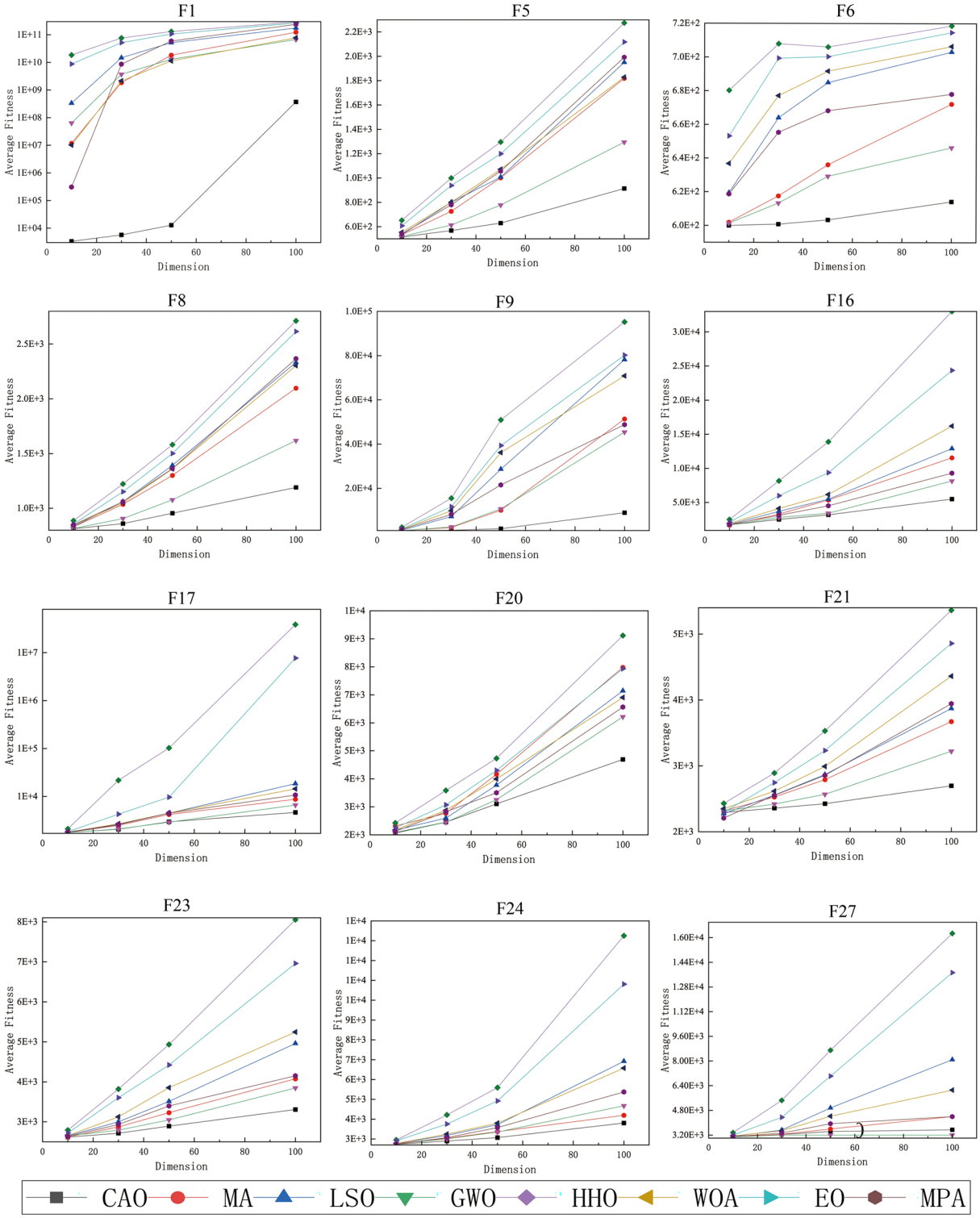

Fig. 12. Scalability analysis of different methods.

where $l = 100$ cm, $P = 2$ kN/cm^2, $\sigma = 2$ kN/cm^2.

The CAO algorithm was employed to solve the three-bar truss design problem, and its effectiveness was compared with that of other optimization techniques. A comprehensive comparison of the capabilities of the CAO algorithm versus other methods is provided in Table 22, and the statistical results are shown in Table 23. The CAO algorithm outperforms other approaches, particularly in terms of the 'Std', 'Worst', 'Mean', and 'Best' metrics.

Fun	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_1	0.0374	0.0801	0.0315	0.0298	0.0648	0.0286	0.0396	0.0725
Fun_3	0.0378	0.0845	0.0324	0.0296	0.0662	0.0280	0.0395	0.0710
Fun_4	0.0373	0.0806	0.0305	0.0288	0.0631	0.0276	0.0383	0.0705
Fun_5	0.0431	0.0789	0.0303	0.0340	0.0624	0.0326	0.0443	0.0761
Fun_6	0.0686	0.0928	0.0354	0.0498	0.0724	0.0484	0.0610	0.0920
Fun_7	0.0454	0.1308	0.0513	0.0356	0.1008	0.0337	0.0458	0.0768
Fun_8	0.0439	0.0957	0.0367	0.0344	0.0748	0.0330	0.0454	0.0766
Fun_9	0.0467	0.0943	0.0359	0.0363	0.0735	0.0350	0.0470	0.0782
Fun_{10}	0.0495	0.0977	0.0375	0.0388	0.0762	0.0366	0.0492	0.0813
Fun_{11}	0.0427	0.1055	0.0408	0.0324	0.0799	0.0312	0.0432	0.0747
Fun_{12}	0.0443	0.0886	0.0341	0.0334	0.0690	0.0318	0.0445	0.0755
Fun_{13}	0.0437	0.0914	0.0345	0.0334	0.0703	0.0322	0.0445	0.0755
Fun_{14}	0.0495	0.0920	0.0348	0.0374	0.0775	0.0354	0.0479	0.0789
Fun_{15}	0.0414	0.0997	0.0382	0.0318	0.0678	0.0305	0.0424	0.0739
Fun_{16}	0.0460	0.0881	0.0330	0.0353	0.0737	0.0332	0.0455	0.0777
Fun_{17}	0.0614	0.0955	0.0363	0.0474	0.0945	0.0444	0.0573	0.0879
Fun_{18}	0.0455	0.0921	0.0466	0.0347	0.0721	0.0323	0.0446	0.0758
Fun_{19}	0.1668	0.2881	0.0348	0.1165	0.2234	0.1142	0.1336	0.1587
Fun_{20}	0.0621	0.1213	0.1176	0.0480	0.0960	0.0448	0.0591	0.0888
Fun_{21}	0.0593	0.1241	0.0483	0.0479	0.0922	0.0430	0.0564	0.0913
Fun_{22}	0.0714	0.1193	0.0458	0.0544	0.1079	0.0518	0.0656	0.1002
Fun_{23}	0.0745	0.1410	0.0562	0.0572	0.1104	0.0539	0.0683	0.1044
Fun_{24}	0.0779	0.1438	0.0581	0.0579	0.1147	0.0557	0.0711	0.1056
Fun_{25}	0.0644	0.1476	0.0501	0.0495	0.0989	0.0469	0.0602	0.0982
Fun_{26}	0.0848	0.1277	0.0635	0.0633	0.1240	0.0600	0.0742	0.1118
Fun_{27}	0.0870	0.1611	0.0650	0.0645	0.1277	0.0620	0.0766	0.1161
Fun_{28}	0.0783	0.1640	0.0579	0.0569	0.1124	0.0545	0.0675	0.1079
Fun_{29}	0.0774	0.1447	0.0594	0.0594	0.1168	0.0552	0.0691	0.1049
Fun_{30}	0.1845	0.3143	0.1268	0.1277	0.2424	0.1247	0.1476	0.1745

Table 9. Computational time of CAO compared to other algorithms.

		Functions			
OPI	Index	Fun3	Fun6	Fun12	Fun21
0.1	Mean	3.124E + 02	6.243E + 02	1.957E + 04	3.019E + 03
	SD	1.126E + 01	2.491E + 01	1.682E + 04	1.380E + 02
0.15	Mean	3.087E + 02	6.186E + 02	1.703E + 04	2.703E + 03
	SD	4.523E + 00	1.953E + 01	1.401E + 04	8.198E + 01
0.2	Mean	3.000E + 02	6.000E + 02	1.335E + 04	2.292E + 03
	SD	0	3.641E−03	1.006E + 04	4.728E + 01
0.25	Mean	3.095E + 02	6.072E + 02	1.594E + 04	2.816E + 03
	SD	8.177E + 00	1.845E + 01	1.382E + 04	6.480E + 01
0.3	Mean	3.169E + 02	6.183E + 02	1.706E + 04	2.984E + 03
	SD	1.640E + 01	2.049E + 01	1.403E + 04	1.156E + 01

Table 10. The results obtained for diverse test functions at varying numbers of the original position inertia(OPI) parameter.

Spring design

The construction of a tension/compression spring with the design depicted in Fig. 21 is used as another acknowledged standard engineering design problem to assess the viability of the suggested CAO method in conventional engineering applications. Reducing the strain on a tension/compression spring design is the aim of this optimization work. Shear stress, minimum deflection, and minimum surge frequency are the only limitations on this engineering design problem. The diameter of the wire (d), the diameter of the mean coil (D), and the total number of active coils (N) are the optimization choice criteria for the design case. A vector representing the optimization parameters for this design scenario looks like this: $x = [\times 1, \times 2, \times 3]$, where the variables $\times 1, \times 2$,

	No	Function	Range	Fun_{Best}
Unimodal functions	$C2020_1$	Shifted & Rotated Bent Cigar Function	[-100 100]	100
Basic functions	$C2020_2$	Shifted & Rotated Schwefel's Function	[-100 100]	1100
	$C2020_3$	Shifted & Rotated Lunacek Bi_Rastrigin Function	[-100 100]	700
	$C2020_4$	Expand Rosenbrock's plus Griewangk's Function	[-100 100]	1900
Hybrid functions	$C2020_5$	Hybrid-Function-1(N = 3)	[-100 100]	1700
	$C2020_6$	Hybrid-Function-2(N = 4)	[-100 100]	1600
	$C2020_7$	Hybrid-Function-3(N = 5)	[-100 100]	2100
Composition functions	$C2020_8$	Composition-Function-1(N = 3)	[-100 100]	2200
	$C2020_9$	Composition-Function-2(N = 4)	[-100 100]	2400
	$C2020_{10}$	Composition-Function-3(N = 5)	[-100 100]	2500

Table 11. Summary of the CEC2020 benchmark functions.

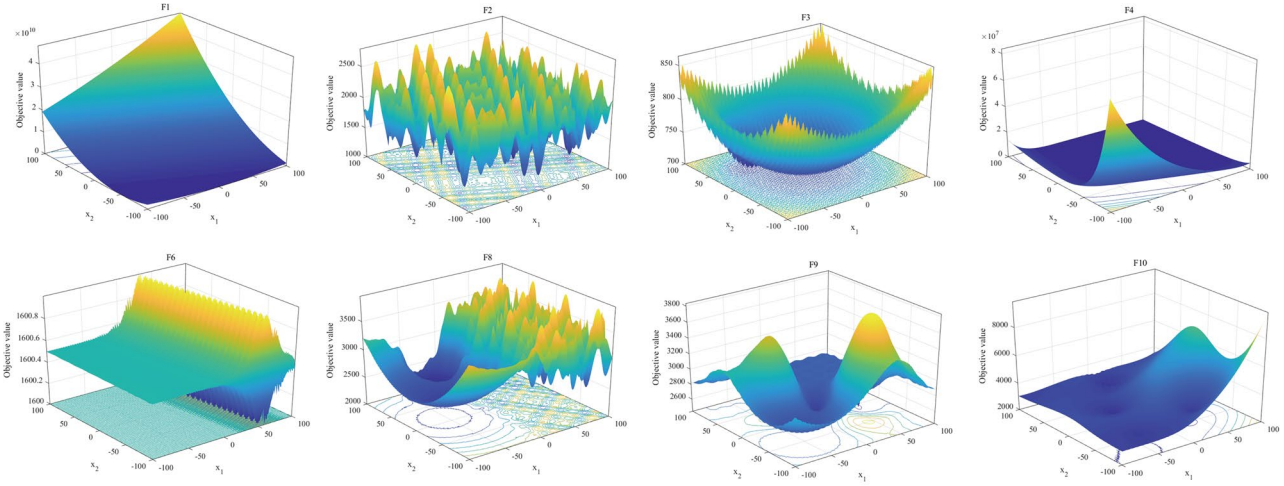

Fig. 13. Three-dimensional representation of some CEC-2020 test functions.

and × 3 stand in for the constants d, D, and N. The following is a description of the mathematical formula for this optimization design problem:

Minimize: $f(\boldsymbol{x}) = (x_3 + 2)\, x_2 x_1^2$,

$$\text{Restriction condition:} \begin{cases} h_1(\boldsymbol{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0, \\ h_2(x) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} \le 0, \\ h_3(\boldsymbol{x}) = 1 - \frac{104.45 x_1}{x_2^2 x_3} \le 0, \\ h_4(\boldsymbol{x}) = \frac{x_1 + x_2}{1.5} - 1 \le 0, \end{cases}$$

Variables range: $0.05 \le x_1 \le 2, 0.25 \le x_2 \le 1.30, 2.00 \le x_3 \le 15$.

Regarding the values of design variables and objective cost values, the suggested CAO algorithm is contrasted with other prospective competing algorithms for the tension/compression spring design problem in Table 24. The results presented in Table 24 show that the suggested CAO algorithm can be used to optimally design the tension spring problem with an optimum cost of 0.0127. In many instances, this cost is marginally lower than that of competing optimization techniques.

An overview of the statistical results of this design challenge as decided by the CAO algorithm and various competing meta-heuristics is shown in Table 25. Once again, when considering the best, average, worst, and standard deviation statistical results, the CAO algorithm fared better than other optimization methods, as shown in Table 25. According to this, CAO is more reliable and effective than many of its competitors at solving this design problem for the same number of iterations and search agents.

Speed reducer design problem

The design of a speed reducer, whose structural schematic is shown in Fig. 22, is another real-world example that is frequently used as a reference benchmark for evaluating optimization techniques. There are seven decision parameters in this design challenge, which makes it difficult. The following four limitations have an impact on the weight that needs to be decreased in this design problem: bending stress of the gear teeth, surface stress,

Fun	Indice	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_1	Mean	**2.050E + 03**	1.653E + 07	3.258E + 08	2.478E + 07	1.730E + 10	1.766E + 07	9.048E + 09	5.181E + 05
	Std	**3.389E + 03**	1.111E + 07	1.744E + 08	7.551E + 07	5.031E + 09	2.301E + 07	1.771E + 09	2.025E + 06
Fun_2	Mean	1.702E + 03	2.329E + 03	2.445E + 03	**1.626E + 03**	2.757E + 03	2.146E + 03	2.693E + 03	1.906E + 03
	Std	3.018E + 02	2.024E + 02	2.110E + 02	3.112E + 02	4.718E + 02	4.134E + 02	**1.529E + 02**	2.122E + 02
Fun_3	Mean	**7.201E + 02**	7.464E + 03	7.691E + 02	7.290E + 02	8.713E + 02	7.846E + 02	8.217E + 02	7.640E + 02
	Std	**3.565E + 00**	5.920E + 00	9.527E + 00	1.075E + 01	9.421E + 00	3.019E + 01	1.049E + 01	3.057E + 03
Fun_4	Mean	**1.901E + 03**	1.903E + 03	1.917E + 03	1.902E + 03	3.089E + 05	1.907E + 03	1.147E + 04	1.903E + 03
	Std	**2.750E–01**	5.539E−01	1.254E + 01	1.082E + 00	1.764E + 05	5.006E + 00	8.399E + 03	1.006E + 00
Fun_5	Mean	**6.860E + 03**	5.101E + 04	1.108E + 05	5.239E + 04	7.134E + 05	3.615E + 05	4.198E + 05	1.842E + 04
	Std	4.845E + 03	3.648E + 04	9.591E + 04	1.049E + 04	**2.570E + 03**	6.944E + 05	1.221E + 05	8.108E + 03
Fun_6	Mean	1.731E + 03	1.682E + 03	**1.382E + 03**	1.742E + 03	2.597E + 03	1.807E + 03	2.220E + 03	1.835E + 03
	Std	**8.655E–03**	6.038E + 01	4.907E + 01	9.682E + 01	2.296E + 02	8.377E + 01	8.042E + 01	1.060E + 01
Fun_7	Mean	1.042E + 04	9.179E + 03	**8.964E + 03**	1.208E + 04	2.353E + 06	2.651E + 05	1.426E + 05	1.440E + 04
	Std	6.698E + 03	5.485E + 03	7.475E + 03	**4.973E + 03**	1.571E + 06	6.252E + 05	1.205E + 05	1.262E + 04
Fun_8	Mean	**2.301E + 03**	2.312E + 03	2.346E + 03	2.310E + 03	3.717E + 03	2.380E + 03	2.918E + 03	2.344E + 03
	Std	**5.358E–01**	1.463E + 00	2.375E + 01	8.094E + 00	4.034E + 02	2.620E + 02	2.064E + 02	1.723E + 02
Fun_9	Mean	2.746E + 03	2.759E + 03	**2.346E + 03**	2.751E + 03	2.957E + 03	2.777E + 03	2.914E + 03	2.730E + 03
	Std	**7.024E + 00**	1.267E + 01	2.375E + 01	1.333E + 01	4.723E + 01	2.247E + 01	7.617E + 01	8.836E + 01
Fun_{10}	Mean	**2.933E + 03**	2.947E + 03	2.969E + 03	**2.933E + 03**	4.402E + 03	2.951E + 03	3.367E + 03	2.972E + 03
	Std	2.290E + 01	**1.036E + 01**	1.078E + 01	1.857E + 01	3.981E + 02	3.184E + 01	9.727E + 01	3.639E + 01

Table 12. The results of the different methods for CEC-2020 test functions (Dim = 10). The bold refers to the best results.

Fun	Indice	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_1	Mean	**2.254E + 03**	2.582E + 08	4.264E + 09	1.005E + 09	4.407E + 10	3.901E + 08	2.938E + 10	8.575E + 08
	Std	**3.105E + 03**	6.905E + 07	1.415E + 09	1.316E + 09	3.153E + 09	3.515E + 08	2.448E + 09	8.035E + 08
Fun_2	Mean	2.415E + 03	4.760E + 03	5.379E + 03	**2.411E + 03**	6.326E + 03	4.137E + 03	5.323E + 03	3.246E + 03
	Std	4.056E + 02	3.230E + 02	3.219E + 02	7.336E + 02	2.705E + 02	6.210E + 02	**1.809E + 02**	5.796E + 02
Fun_3	Mean	**7.507E + 02**	8.617E + 02	9.269E + 02	7.794E + 02	1.122E + 03	9.703E + 02	1.042E + 03	9.693E + 02
	Std	**1.099E + 01**	9.569E + 00	2.395E + 01	3.150E + 01	8.134E + 00	4.411E + 01	2.016E + 01	9.298E + 01
Fun_4	Mean	**1.903E + 03**	1.915E + 03	2.745E + 03	2.604E + 03	3.082E + 06	2.058E + 03	2.779E + 05	2.234E + 03
	Std	**1.141E + 00**	2.234E + 00	7.402E + 02	4.841E + 02	1.828E + 06	1.656E + 02	1.432E + 05	1.134E + 03
Fun_5	Mean	**3.210E + 05**	1.819E + 06	1.042E + 06	6.825E + 05	2.644E + 07	1.767E + 06	4.680E + 06	1.780E + 06
	Std	2.445E + 05	9.060E + 05	6.802E + 05	8.420E + 05	**1.113E + 07**	1.110E + 06	1.543E + 06	1.300E + 06
Fun_6	Mean	1.813E + 03	1.974E + 03	**2.651E + 03**	1.949E + 03	4.078E + 03	2.493E + 03	3.340E + 03	2.222E + 03
	Std	**1.762E + 02**	1.023E + 02	1.973E + 02	2.377E + 02	3.824E + 02	3.045E + 02	1.448E + 02	1.953E + 02
Fun_7	Mean	1.474E + 05	6.588E + 05	**1.173E + 05**	1.638E + 05	3.665E + 07	1.209E + 06	9.255E + 05	5.617E + 05
	Std	2.141E + 05	4.405E + 05	1.184E + 05	**8.627E + 04**	2.730E + 07	1.126E + 06	4.050E + 05	4.181E + 05
Fun_8	Mean	**3.379E + 03**	3.044E + 03	4.167E + 03	3.784E + 03	7.412E + 03	4.211E + 03	5.634E + 03	3.884E + 03
	Std	**1.251E + 03**	1.616E + 03	1.016E + 03	2.057E + 03	5.074E + 02	1.933E + 03	4.330E + 02	1.560E + 03
Fun_9	Mean	2.842E + 03	2.913E + 03	**2.983E + 03**	2.894E + 03	3.344E + 03	2.999E + 03	3.316E + 03	2.926E + 03
	Std	**1.412E + 01**	1.274E + 01	2.499E + 01	3.798E + 01	1.253E + 02	5.347E + 01	7.614E + 01	4.080E + 01
Fun_{10}	Mean	**2.949E + 03**	2.944E + 03	3.918E + 03	**2.991E + 03**	8.688E + 03	3.056E + 03	5.471E + 03	3.063E + 03
	Std	2.927E + 01	**2.615E + 01**	8.494E + 01	3.839E + 01	1.563E + 03	3.156E + 01	3.714E + 02	1.120E + 02

Table 13. The results of the different methods for CEC-2020 test functions (Dim = 20). The bold refers to the best results.

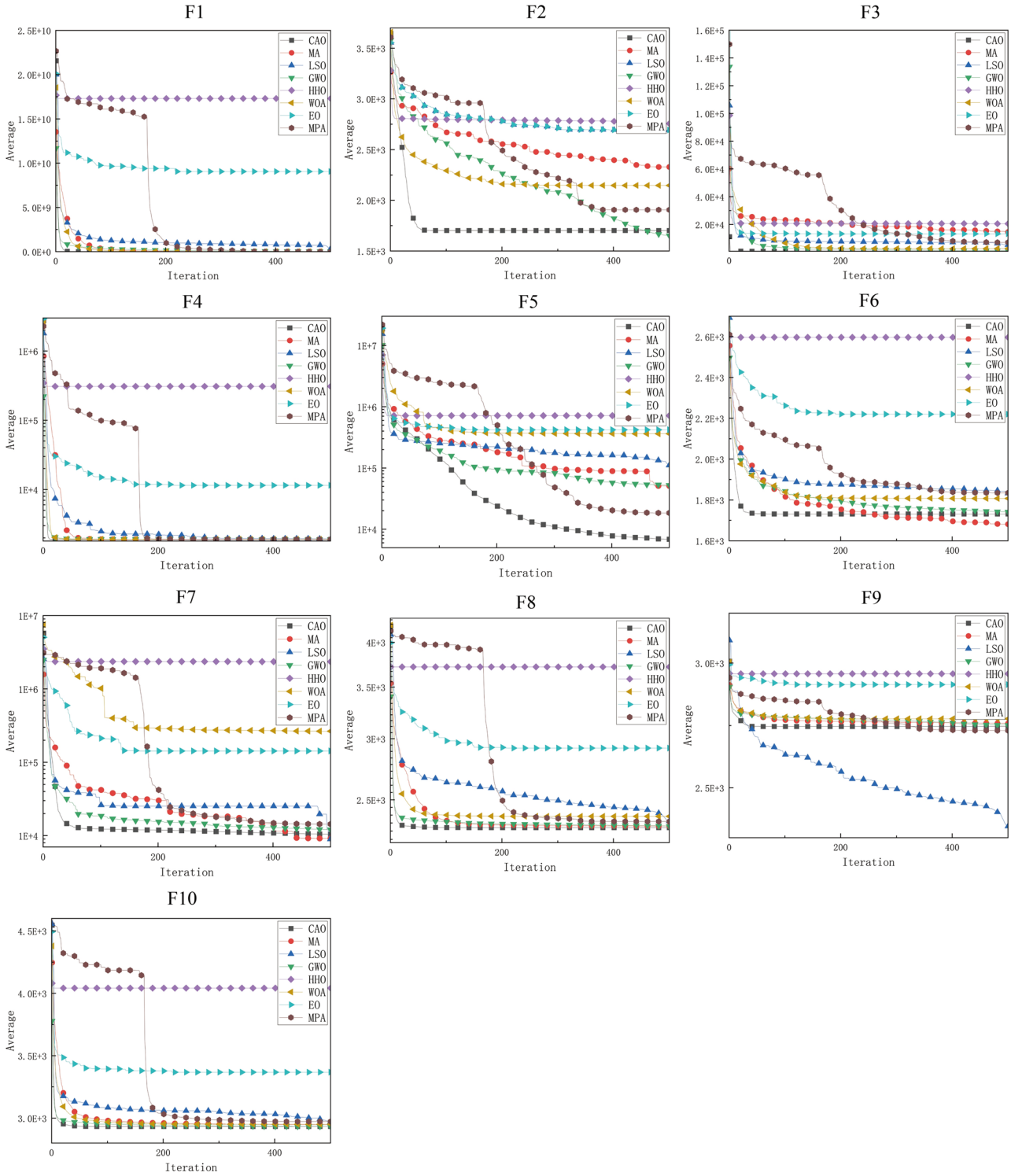

Fig. 14. Convergence curves of algorithms for different CEC-2020 functions (Dim = 10).

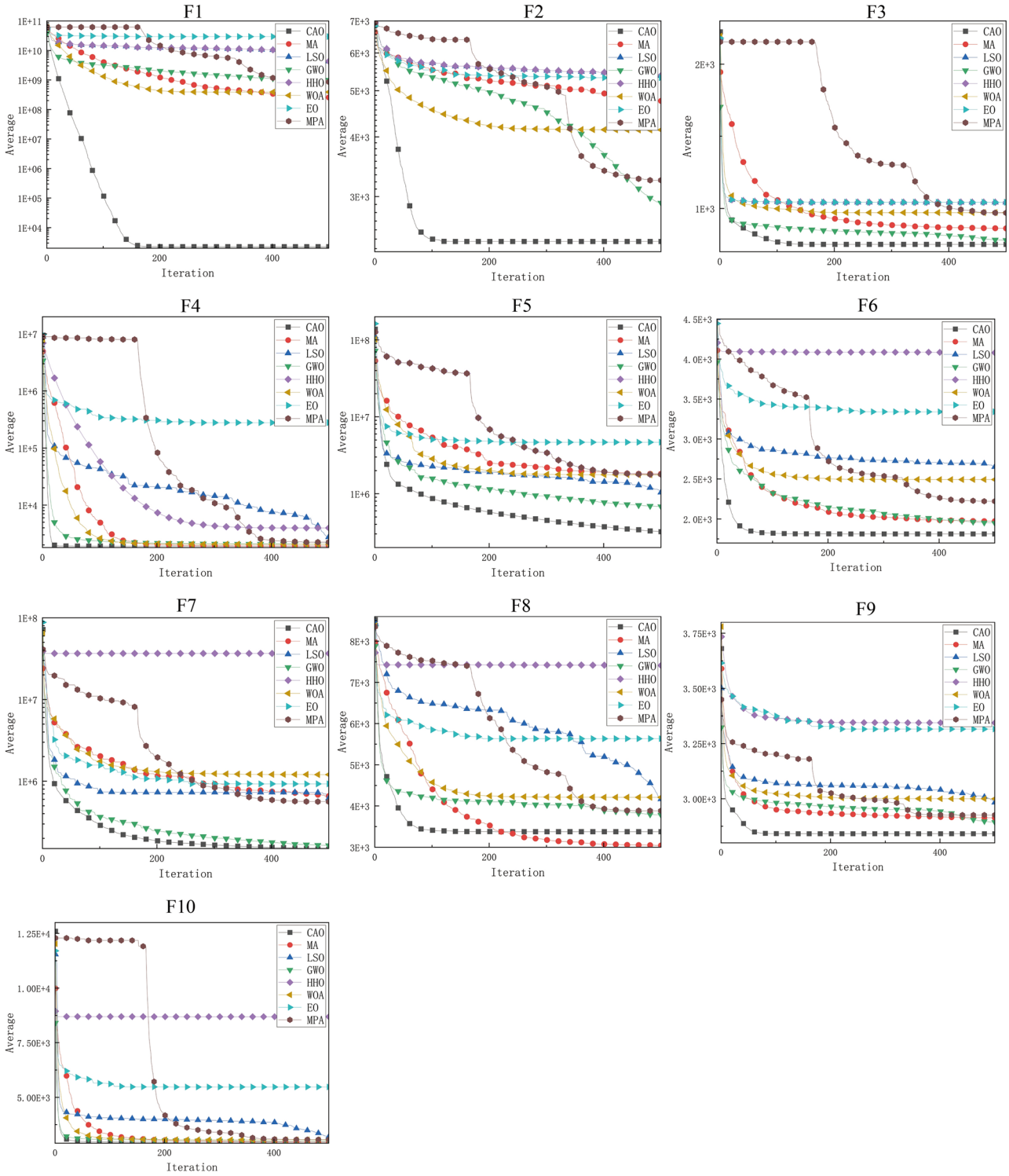

Fig. 15. Convergence curves of algorithms for different CEC-2020 functions (Dim = 20).

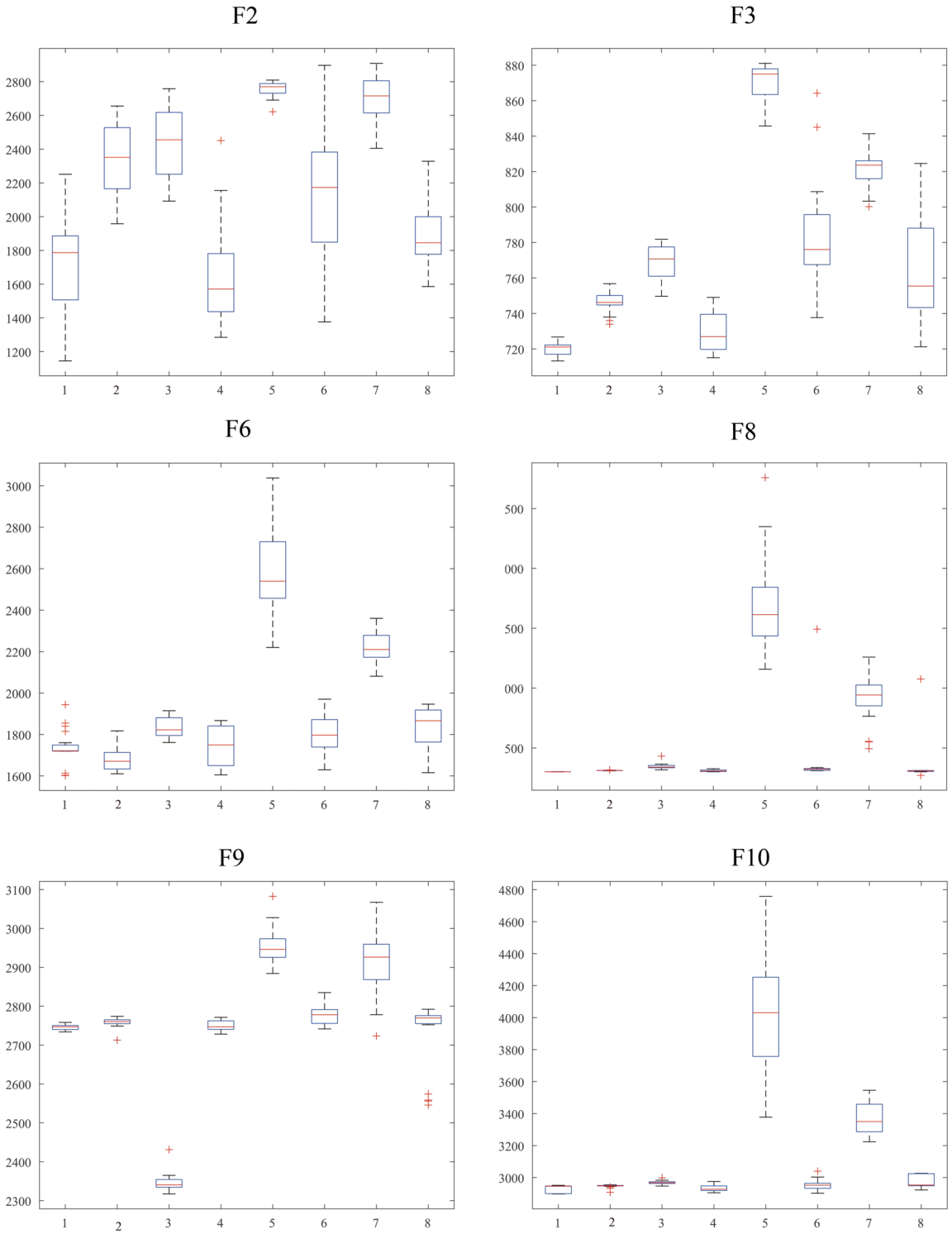

Fig. 16. Boxplot of algorithms for different CEC-2020 functions (Dim = 10).

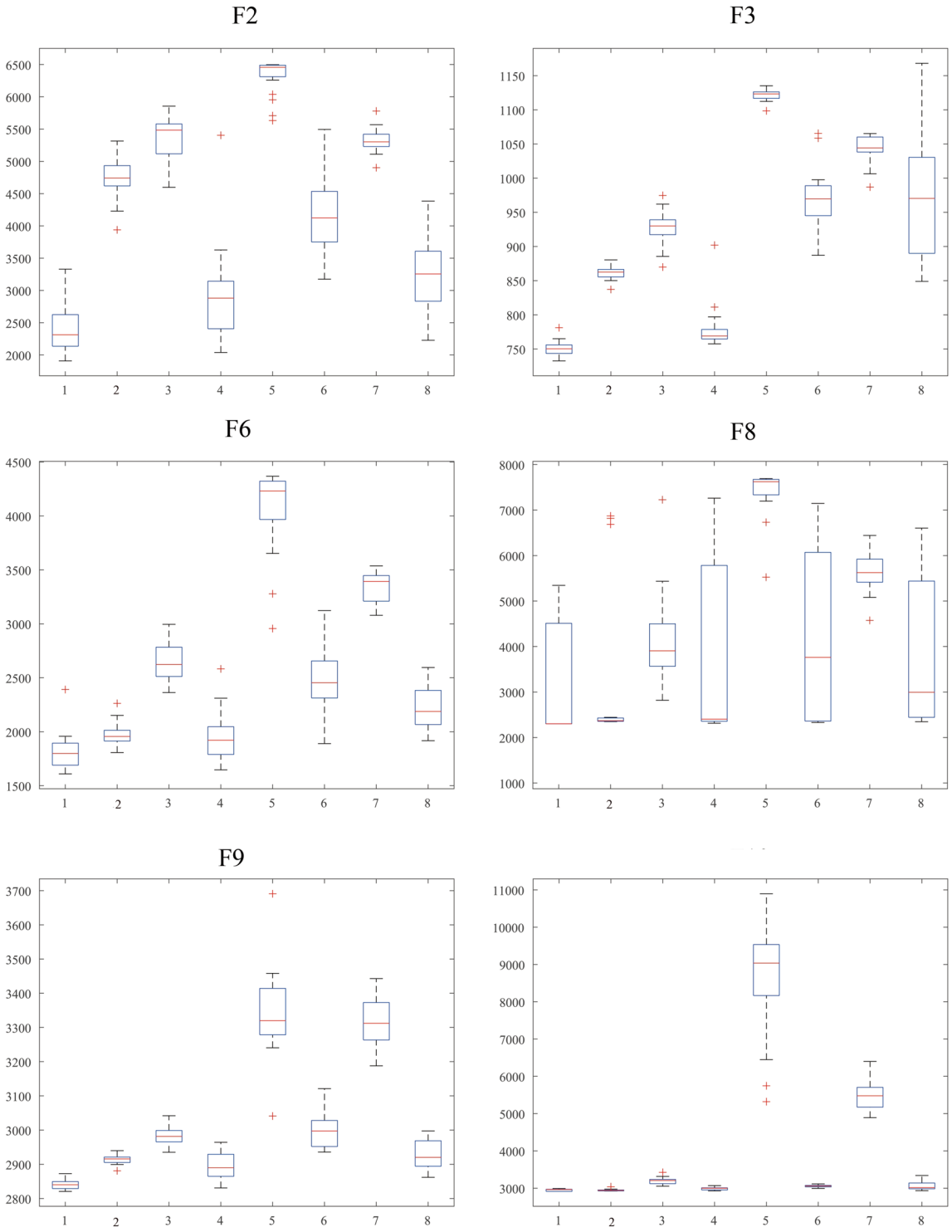

Fig. 17. Boxplot of algorithms for different CEC-2020 functions (Dim = 20).

Fun	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_1	1	3	6	5	8	4	7	2
Fun_2	2	5	6	1	8	4	7	3
Fun_3	1	8	4	2	7	5	6	3
Fun_4	1	3	6	2	8	5	7	4
Fun_5	1	3	5	4	8	6	7	2
Fun_6	3	2	1	4	8	5	7	6
Fun_7	3	2	1	4	8	7	6	5
Fun_8	1	3	5	2	8	6	7	4
Fun_9	3	5	1	4	8	6	7	2
Fun_{10}	2	3	5	1	8	4	7	6
Mean rank	**1.8**	**3.7**	**4.0**	**2.9**	**7.9**	**5.2**	**6.8**	**3.7**
Result	**1**	**3**	**5**	**2**	**8**	**6**	**7**	**4**

Table 14. Friedman ranks for CEC-2020 benchmark functions(Dim = 10).

Fun	CAO	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_1	1	2	6	5	8	3	7	4
Fun_2	1	5	7	2	8	4	6	3
Fun_3	1	3	4	2	8	6	7	5
Fun_4	1	2	7	5	8	3	6	4
Fun_5	1	6	3	2	8	4	7	5
Fun_6	1	3	6	2	8	5	7	4
Fun_7	1	5	4	2	8	7	6	3
Fun_8	2	1	5	3	8	6	7	4
Fun_9	1	3	5	2	8	6	7	4
Fun_{10}	2	1	6	3	8	4	7	5
Mean rank	**1.2**	**3.1**	**5.3**	**2.8**	**8.0**	**4.8**	**4.7**	**4.1**
Result	**1**	**3**	**7**	**2**	**8**	**6**	**5**	**4**

Table 15. Friedman ranks for CEC-2020 benchmark functions(Dim = 20).

Fun	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_1	8.858E−05/-	8.858E−05/-	1.034E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	1.401E−04/-
Fun_2	1.034E−04/-	8.858E−05/-	7.089E−01/=	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-
Fun_3	8.858E−05/-	8.858E−05/-	4.000E−03/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	1.034E−04/-
Fun_4	8.858E−05/-	8.858E−05/-	1.890E−04/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	1.034E−04/-
Fun_5	1.036E−04/-	8.858E−05/-	4.553E−01/=	8.858E−05/-	1.401E−04/-	8.858E−05/-	4.493E−04/-
Fun_6	8.590E−02/=	1.900E−03/-	6.813E−01/=	8.858E−05/-	5.700E−03/-	8.858E−05/-	7.200E−03/-
Fun_7	4.553E−01/=	3.905E−01/=	3.905E−01/=	8.858E−05/-	8.858E−05/-	1.034E−04/-	3.135E−01/=
Fun_8	8.858E−05/-	8.858E−05/-	1.401E−04/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	1.900E−03/-
Fun_9	2.500E7−03/-	8.858E−05/-	1.560E−01/=	8.858E−05/-	2.536E−04/-	5.700E−03/-	2.627E−05/=
Fun_{10}	6.400E−03/-	1.034E−04/-	9.405E−01/=	8.858E−05/-	3.040E−02/-	8.858E−05/-	1.629E−04/-
+/=/−	0/2/8	0/1/9	0/6/4	0/0/10	0/0/10	0/0/10	0/2/10

Table 16. Statistical results of WSR test obtained by CAO on CEC-2020 benchmark (Dim = 10).

transverse shaft deflections, and shaft stresses. To address this optimization problem, seven decision design parameters were used, which are as follows: b, m, z, l1, l2, d1, and d2. These characteristics are as follows: the diameter of the first and second shafts, the distance between the bearings between the first and second shafts, the tooth module, the number of teeth in the pinion, and the face width. In order to solve this optimization problem, these parameters were represented by a vector, which is provided as follows: x = [×1 × 2 × 3 × 4 × 5 × 6 × 7] = [b m z l1 l2 d1 d2]. The following is a description of the mathematical formula for this problem:

The cost function that needs to be optimized can be described as follows:

$$f(\vec{x}) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$$

Fun	MA	LSO	GWO	HHO	WOA	EO	MPA
Fun_1	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-
Fun_2	8.858E−05/-	8.858E−05/-	1.240E−02/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	1.900E−03/-
Fun_3	8.858E−05/-	8.858E−05/-	2.536E−04/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-
Fun_4	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-
Fun_5	1.034E−04/-	8.918E−05/-	2.043E−01/=	8.858E−05/-	2.932E−04/-	8.858E−05/-	1.036E−04/-
Fun_6	2.500E−03/-	8.858E−05/-	4.790E−02/-	8.858E−05/-	1.034E−04/-	8.858E−05/-	1.401E−04/-
Fun_7	1.401E−04/-	1.401E−04/-	1.672E−01/=	8.858E−05/-	1.401E−04/-	1.034E−04/-	6.806E−04/-
Fun_8	8.228E−01/=	1.690E−02/-	4.330E−01/=	8.858E−05/-	5.220E−02/=	8.858E−05/-	2.180E−01/=
Fun_9	8.858E−05/-	8.858E−05/-	4.493E−04/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	8.858E−05/-
Fun_{10}	5.503E−01/=	8.858E−05/-	5.100E−03/-	8.858E−05/-	8.858E−05/-	8.858E−05/-	3.385E−04/-
+/=/−	0/2/8	0/0/10	0/3/7	0/0/10	0/1/9	0/0/10	0/1/9

Table 17. Statistical results of WSR test obtained by CAO on CEC-2020 benchmark (Dim = 20).

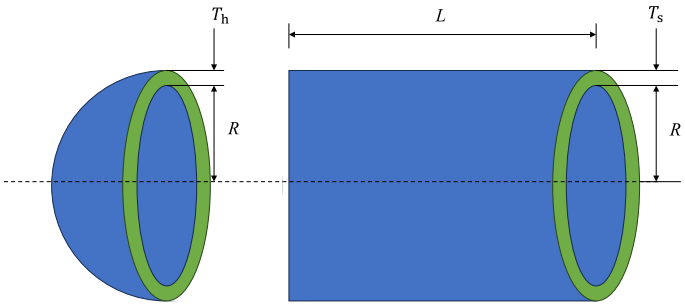

Fig. 18. Pressure vessel design problem.

Optimzers	*Ts*	*Th*	*R*	*L*	Best value
CAO	0.7781686398	0.3846491701	40.31961874	200	5885.332774
MA	0.7781687871	0.3846498138	40.31961875	200	5885.335716
LSO	12.51406454	6.18743224	40.51910215	197	5885.335475
GWO	0.7784057149	0.3854974361	40.32307458	200	5890.070967
HHO	0.8412681315	0.4185983078	43.56479801	159	6014.425412
WOA	0.8158984081	0.4491581143	41.3425583	186	6213.457215
EO	1.234901903	0.6104126501	63.98455452	96	11,546.52898
MPA	0.9006497489	0.4448263208	46.62749694	127	6132.188742

Table 18. Optimum results of the different methods for the pressure vessel design problem.

Algorithms	Min	Max	Mean	Std
CAO	5885.332774	5885.405814	5885.343451	1.83E−01
MA	5885.335716	6195.382028	5932.787433	8.51E+01
LSO	5885.335475	6069.067614	6032.000702	3.02E+02
GWO	5890.070967	6822.450373	6255.804986	2.22E+02
HHO	6014.425412	7527.978657	6718.945634	3.80E+02
WOA	6213.457215	9563.098978	7510.098338	7.31E+02
EO	11,546.52898	198,316.9584	106,631.9927	4.87E+04
MPA	6132.188742	7321.126927	6785.659595	3.79E+02

Table 19. Statistical results from various meta-heuristic algorithms for the pressure vessel problem.

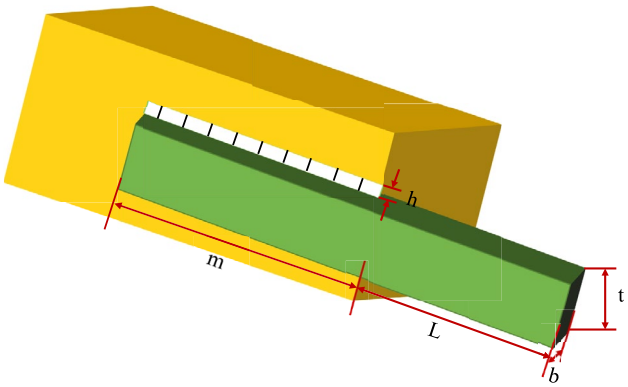

Fig. 19. The welded beam design drawing map.

Algorithms	Best	Mean	Worst	Std	Rank
CAO	**1.724527**	1.9142249	2.1056474	0.1175823	**1**
MA	1.724536	2.0246610	2.2271271	0.2375921	2
LSO	1.724718	1.9528762	2.1444756	0.1802129	4
GWO	1.733462	2.6589611	3.1274315	0.7812475	7
HHO	1.726240	1.9964298	2.8731091	0.5209634	5
WOA	1.733487	2.6589687	3.1495628	0.7801672	8
EO	1.729843	1.9275891	2.4980137	0.3089531	6
MPA	1.724852	2.2614932	2.8091743	0.4308599	3

Table 20. Results of the welded beam design problem.

Algorithms	Optimal values for variables				Optimum cost
	h	l	t	b	
CAO	0.187156	3.470488	9.036623	0.205730	**1.724527**
MA	0.187156	3.744548	9.173438	0.205057	1.724536
LSO	0.203687	3.470489	9.036624	0.205729	1.724718
GWO	0.205701	3.479005	9.036874	0.205732	1.733462
HHO	0.203137	3.744548	9.173438	0.205057	1.726240
WOA	0.205700	3.657587	9.176331	0.205111	1.733487
EO	0.204368	3.856979	3.856979	0.212148	1.729843
MPA	0.205729	3.512662	8.997062	0.207548	1.724852

Table 21. The best variables the welded beam design problem.

The following limitations apply to this engineering design:

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

$$g_2(\overrightarrow{x}) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \leq 0$$

$$g_3(\overrightarrow{x}) = \frac{1.9 x_4^3}{x_2 x_6^4 x_3} - 1 \leq 0$$

$$g_4(\overrightarrow{x}) = \frac{1.93 x_5^3}{x_2 x_7^4 x_3} - 1 \leq 0$$

$$g_5(\overrightarrow{x}) = \frac{\left[(745(x_4/x_2 x_3))^2 + 16.9 \times 10^6\right]^{1/2}}{110 x_6^3} - 1 \leq 0$$

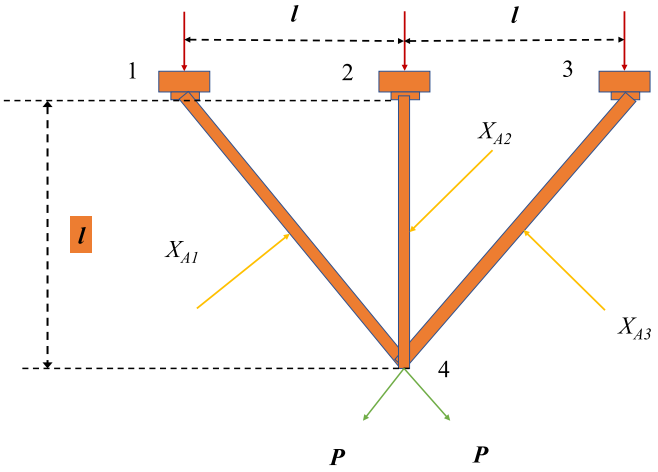

Fig. 20. Schematic of the three-bar truss design.

Algorithms	Best	Mean	Worst	Std	Rank
CAO	291.7724	291.7877	291.7899	0.02849	**1**
MA	303.4433	303.5451	303.5905	0.02964	8
LSO	292.1323	292.1392	292.1399	0.03027	3
GWO	292.2584	292.2797	292.2842	0.03351	4
HHO	292.5509	292.5610	292.6141	0.12096	6
WOA	292.1138	292.9142	293.0172	0.54898	2
EO	292.4797	292.5026	292.8821	0.49729	5
MPA	292.8279	292.8301	292.8491	0.39421	7

Table 22. Results of the three-bar truss design.

Algorithms	Optimal values for variables				Optimum cost
	Ta	Tb	Td	Tf	
CAO	42.5715	19.3577	15.9691	48.9495	**2.7009E−12**
MA	42.6953	15.5460	19.4798	49.2722	2.7009E−12
LSO	49.0442	16.2459	18.7594	42.7506	2.7009E−12
GWO	43.2800	18.6127	15.6179	48.5065	2.7009E−12
HHO	42.6953	15.5460	19.4798	49.2722	2.7009E−12
WOA	42.6477	18.8069	15.8717	49.3500	2.7009E−12
EO	43.2426	19.2179	16.0007	49.3223	2.7009E−12
MPA	43.4553	16.3714	18.5838	48.7162	2.7009E−12

Table 23. The best variables of the three-bar truss design.

$$g_6(\overrightarrow{x}) = \frac{\left[(745(x_5/x_2x_3))^2 + 157.5 \times 10^6\right]^{1/2}}{85x_7^3} - 1 \leq 0$$

$$g_7(\overrightarrow{x}) = \frac{x_2x_3}{40} - 1 \leq 0$$

$$g_8(\overrightarrow{x}) = \frac{5x_2}{x_1} - 1 \leq 0$$

$$g_9(\overrightarrow{x}) = \frac{x_1}{12x_2} - 1 \leq 0$$

$$g_{10}(\overrightarrow{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0$$

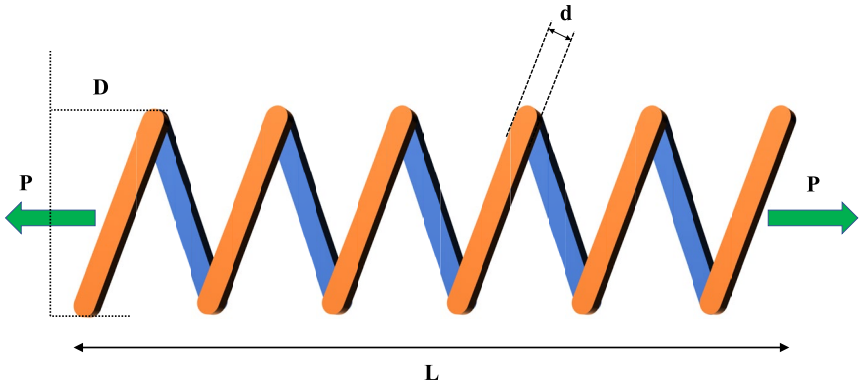

Fig. 21. A schematic structural diagram of a tension spring.

Algorithms	Optimal values for variables			Optimum cost
	d	D	N	
CAO	0.0543	0.4316	7.9901	**0.0127**
MA	0.0740	0.8482	4.8851	0.0130
LSO	0.0533	0.3960	9.5781	0.0130
GWO	0.0525	0.3775	10.168	0.0128
HHO	0.0539	0.4025	9.4517	0.0134
WOA	0.0623	0.6408	0.6408	0.6408
EO	0.0598	0.0598	0.0598	0.0598
MPA	0.0610	0.0610	0.0610	0.0610

Table 24. Results of the spring design problem.

Algorithms	Best	Mean	Worst	Std	Rank
CAO	**0.0127**	0.0134	0.0142	0.0005	**1**
MA	0.0130	0.0194	0.0203	0.0098	4
LSO	0.0130	0.0140	0.0161	0.0072	3
GWO	**0.0127**	0.0142	0.0198	0.0024	2
HHO	0.0134	0.0153	0.0184	0.0016	5
WOA	0.6408	0.7241	0.7803	0.2459	8
EO	0.0598	0.0893	0.1035	0.0852	6
MPA	0.0610	0.0704	0.0830	0.0281	7

Table 25. Results of the spring design problem.

$$g_{11}\overrightarrow{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0$$

where, for the variables b, m, z, l1, l2, d1, and d2, the ranges of the design variables are $2.6 \leq \times 1 \leq 3.6$, $0.7 \leq \times 2 \leq 0.8$, $17 \leq \times 3 \leq 28$, $7.3 \leq \times 4 \leq 8.3$, $7.3 \leq \times 5 \leq 8.3$, $2.9 \leq \times 6 \leq 3.9$, and $5.0 \leq \times 4 \leq 5.5$, respectively.

Table 26 shows a comparison between the designs and cost solutions for the speed reducer design challenge that CAO came up with and the other optimization methods mentioned above. Table 26 shows that the suggested CAO algorithm works better than other competing optimization methods because it has the best design cost for this problem, which is around 2994.471. This suggests that CAO can be used to determine the best design for this issue.

The statistical results of the CAO algorithm and other optimization strategies for the speed reducer design problem are tabulated in Table 27. The statistics shown in Table 27 show that the CAO algorithm is superior to other meta-heuristic techniques. This indicates that the CAO algorithm produced the best optimal solutions among all the competing algorithms. This demonstrates how, based on these statistical findings, the CAO algorithm performs better than rival algorithms.

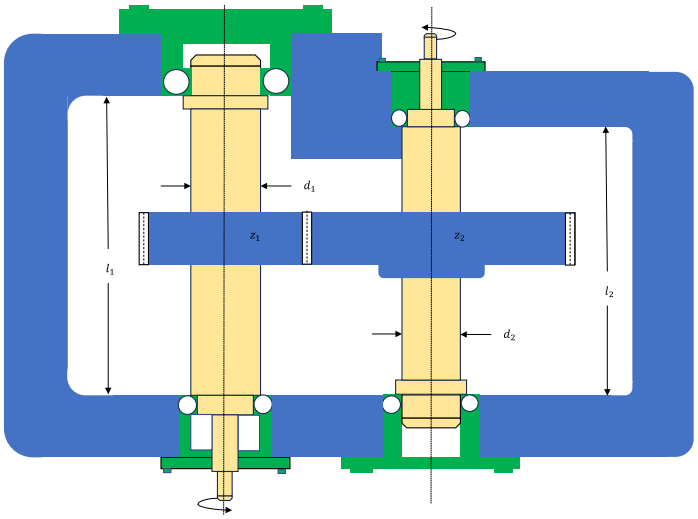

Fig. 22. A schematic structural diagram of a speed reducer design.

Algorithms	Optimal values for variables							Optimum cost
	b	m	z	l_1	l_2	d_1	d_2	
CAO	3.5	0.7	17	7.3	7.715	3.350	5.287	**2994.471**
MA	3.599	0.7	17	7.923	8.099	3.379	5.340	3141.817
LSO	3.545	0.7	17	7.807	8.3	3.353	5.303	3040.644
GWO	3.5	0.7	17	7.403	7.753	3.358	5.287	2998.204
HHO	3.502	0.7	17	7.927	7.737	3.374	5.287	3007.477
WOA	3.5	0.7	17	7.950	7.963	3.351	5.291	3008.738
EO	3.5	0.7	17	7.312	7.715	3.350	5.287	2994.834
MPA	3.546	0.7	17	7.834	8.257	3.356	5.299	3040.749

Table 26. Comparison of the optimization results from several meta-heuristic algorithms for the speed reducer problem.

Algorithms	Best	Mean	Worst	Std
CAO	2994.471	2994.471	2994.471	2.12E−12
MA	3141.817	3564.938	4283.717	3.22E + 02
LSO	3040.644	3107.051	3176.963	3.08E + 01
GWO	2998.204	3005.095	3013.756	4.73E + 00
HHO	3007.477	3134.660	4047.006	2.28E + 02
WOA	3008.738	3117.951	3470.068	9.38E + 01
EO	2994.834	2994.903	3007.437	1.08E + 00
MPA	3040.749	3107.051	3176.963	3.24E + 02

Table 27. The Statistical results obtained from various methods for the speed reducer design problem.

Cantilever beam design problem

Despite the similarities to the previous problem, the objective of this one is to lower the weight of a cantilever beam composed of five components, each of which has a hollow cross section that gradually thickens. As seen in Fig. 23, the beam is securely supported and the free end of the cantilever is subject to an external vertical force.

The goal of this design challenge is to reduce the weight of a cantilever beam while placing a maximum limit on the vertical displacement of the free end. The design variables are each part's cross-sectional heights and widths. Because the upper and lower bounds are too large and little, respectively, these variables do not become operational in the issue. Finding workable combinations of the five structural design parameters is the first step in solving the cantilever beam design challenge. These design parameters could be represented by a vector like this: $\overrightarrow{x} = [x_1, x_2, x_3, x_4, x_5]$. The objective cost function for this design problem can be written as follows:

f(x) = 0.0624(× 1 + × 2 + × 3 + × 4 + × 5), where the optimization constraint that follows is applicable.

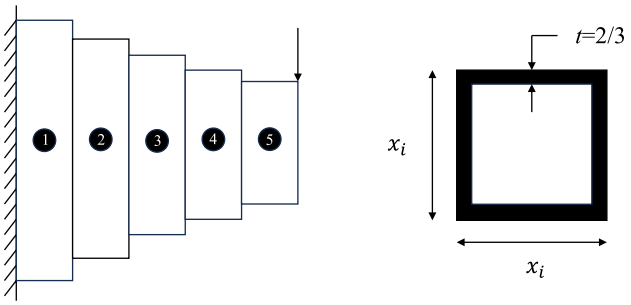

Fig. 23. A schematic diagram of a cantilever beam design problem.

Algorithms	Optimal values for variables					Optimum cost
	X_1	X_2	X_3	X_4	X_5	
CAO	3.350	5.286	6.016	5.309	4.494	263.896
MA	3.485	5.328	5.999	5.329	4.499	263.896
LSO	3.352	5.301	6.015	5.309	4.494	263.896
GWO	3.350	5.287	6.016	5.309	4.494	263.896
HHO	3.350	5.287	6.016	5.309	4.495	263.896
WOA	3.351	5.287	6.015	5.310	4.494	263.896
EO	3.339	3.434	6.015	5.310	4.494	263.896
MPA	3.415	5.351	6.015	5.309	4.496	263.896

Table 28. Comparison of the optimization results from several meta-heuristic algorithms for the cantilever design problem.

Algorithms	Best	Mean	Worst	Std
CAO	263.896	263.896	263.896	1.06E−14
MA	263.896	263.896	263.896	8.15E−10
LSO	263.896	263.896	263.900	7.89E−04
GWO	263.896	263.896	263.896	1.01E−07
HHO	263.896	263.896	263.896	1.46E−04
WOA	263.896	263.896	263.896	6.69E−12
EO	263.896	263.896	263.896	2.96E−09
MPA	263.896	263.900	263.919	1.12E−02

Table 29. Statistical results from various meta-heuristic algorithms for the cantilever design problem.

$$g_1(\overrightarrow{x}) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3}$$

The variables were assumed to be in the range $1 \leq xi \leq 10$ for this design issue, where $i \in 1, 2, 3, 4, 5$.

The optimization results of the suggested CAO technique and other similar competing meta-heuristics used to address this issue are shown in Table 28. Based on the cost weight values given in Table 28, the suggested CAO algorithm yielded the optimum solution for the cantilever design problem, with an ideal cost of around 263.896. When compared to other competing algorithms, this result showed incredibly competitive results for CAO and outperformed the majority of them.

The statistical optimization results of the CAO method and the other optimization strategies covered above are compiled in Table 29 with regard to the mean, standard deviation, best, and worst scores for the cantilever design problem across 20 different runs.

As demonstrated by the solutions in Table 29, the suggested CAO approach fared better statistically than many other competing algorithms for this design problem. This implies that CAO is superior to other meta-heuristics, exceeding most algorithms in producing results that are on par with them.

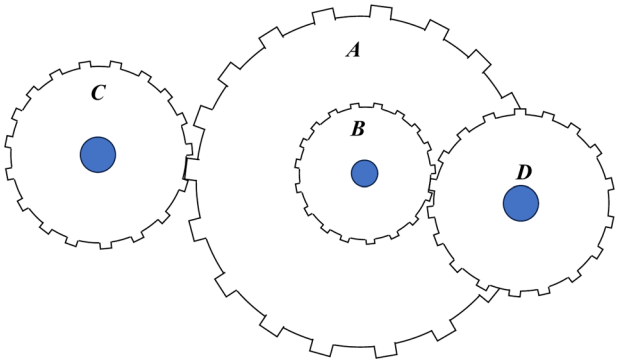

Fig. 24. A schematic diagram of a gear train design problem.

Algorithms	Optimal values for variables				Optimum cost
	Ta	Tb	Td	Tf	
CAO	42.571	19.358	19.357	48.949	**2.701E−12**
MA	33.619	13.008	20.181	52.847	2.308 E−11
LSO	49.044	16.246	18.759	42.751	**2.701E−12**
GWO	43.280	18.613	15.618	48.506	**2.701E−12**
HHO	42.695	15.546	19.480	49.272	**2.701E−12**
WOA	42.648	18.807	15.872	49.350	**2.701E−12**
EO	43.456	16.374	18.584	48.718	2.702E−12
MPA	43.243	19.218	16.001	49.322	**2.701E−12**

Table 30. Optimum results of the different methods for the gear train design problem.

Algorithms	Best	Mean	Worst	Std
CAO	**2.701E−12**	**8.750E−11**	1.362E−09	**2.45E−10**
MA	2.308 E−11	3.995E−09	2.726E−08	8.15E−09
LSO	**2.701E−12**	7.370E−10	2.358E−09	8.08E−10
GWO	**2.701E−12**	6.131E−10	9.922E−10	4.53E−10
HHO	**2.701E−12**	1.734E−09	1.827E−08	3.48E−09
WOA	**2.701E−12**	1.669E−09	1.827E−08	3.44E−09
EO	2.702E−12	3.125E−10	2.358E−09	6.96E−10
MPA	**2.701E−12**	2.299E−09	1.827E−08	4.16E−09

Table 31. The Statistical results obtained from various methods for the gear train design problem.

Gear train design problem

As shown in Fig. 24, the primary goal of this challenge is to reduce the cost of the gear ratio in a gear train, which comprises four design variables, including the number of gear teeth. It is possible to formulate this unconstrained discrete design problem quantitatively.

Consider variable $\mathbf{z} = [z_1, z_2, z_3, z_4] = [T_a, T_b, T_d, T_f]$.

$$\text{Minimize} f(\mathbf{z}) = \left(\frac{1}{6.931} - \frac{T_b.T_d}{T_a.T_f}\right)^2.$$

Variable range: $0.01 \leq z_i \leq 60 i = 1, ..., 4$.

Several techniques, including the proposed CAO algorithm, were used to determine the optimal parameter values for the gear train design problem. The results are shown in Table 30 and show that, with the exception of the MFO algorithm, the CAO algorithm yields the same minimum cost as the other approaches. The statistical results for each method are shown in Table 31, which shows that the CAO algorithm is the most effective in terms of both the "Mean" and "Std" metrics.

Discussion

This part summarizes and discusses the experimental results, which are categorized into four groups to fully demonstrate the competitiveness and efficacy of the CAO algorithm suggested in this work. First, qualitative study shows that CAO well balances exploration and exploitation, avoids local optima, shows good convergence, and demonstrates swarm intelligence traits. Second, the comparison results show that CAO exhibits better convergence accuracy across the majority of benchmark functions (CEC-2017 and CEC-2020) when compared to popular algorithms. Its growing benefit as the problem dimension grows is especially notable. Convergence curves further demonstrate CAO's potential for optimization, indicating robustness against local optima problems for more promising solutions and demonstrating sustained convergence behavior in the late iteration. CAO's result distribution is more centralized, as shown by box plots. Additionally, statistical research confirms CAO's outstanding performance on benchmarks of many dimensions, demonstrating thorough and efficient issue optimization capabilities.

Third, CAO demonstrates its advantage in handling complicated issues by achieving expected numerical optimization outcomes even when compared to two benchmark-winning techniques and six sophisticated algorithms.

Lastly, compared to several state-of-the-art methods, CAO ranks among the top optimizers and obtains the best solutions for eight well-known industrially limited issues. These results highlight how the suggested CAO effectively handles issues related to local optima and immature convergence across many problem classes with its exploratory and exploitative methods, demonstrating a greater possibility for avoiding local optima stagnation.

The better performance of CAO over current optimization algorithms is firmly supported by our experimental data. This benefit is due to a number of important factors:

- CAO employs a hunting search strategy that allows it to solve optimization problems with different characteristics, including single-peak landscapes, numerous variables, and constraints.
- The escape strategy in CAO is designed for exploration and helps the algorithm to carry out global search.
- By successfully avoiding local extremes and premature convergence, the commensalism phase adds to the algorithm's resilience throughout exploration and exploitation.
- CAO is distinguished by its simplicity and ease of use.
- CAO offers benefits in terms of computational cost and complexity while guaranteeing the best outcomes.

The analysis presented above leads to the following conclusions: the suggested algorithm performs exceptionally well in terms of optimization, particularly when dealing with multimodal and composite functions. Three main qualities are responsible for this effectiveness: its extraordinary ability to avoid local optima, its excellent exploration capabilities, and its skillful balance between exploration and exploitation.

These search features are closely related to the algorithm's multi-strategy search mechanism, which guarantees solution variety, promotes thorough exploration, and reduces the possibility of becoming trapped in local optima. It is ideal for resolving industrial optimization issues because of these qualities.

Conclusion and future work

This study introduces a new biologically inspired optimizer inspired by the hunting and escape behavior of Channa Argus in their natural environment. The CAO's performance is evaluated using a broad range of 39 benchmark functions, including unimodal, multimodal, hybrid, and composite functions. To underscore its optimization capabilities, CAO is compared against state-of-art meta-heuristics, results from actual engineering challenges show that CAO is particularly competitive when it comes to solving engineering jobs with uncertain and limited search spaces.

Even with its exceptional efficiency, CAO has limitations when it comes to solving some discrete optimization problems, especially in high-dimensional binary spaces where many optimization algorithms are challenged by the solution space's exponential development. Furthermore, CAO may exhibit relative inefficiency on particular engineering challenges, just like any optimizer. These shortcomings offer insightful guidelines for further study, creating chances for notable breakthroughs in optimization algorithms for challenging, real-world issues. Combining CAO with other algorithms for synergistic enhancement to overcome its shortcomings is also a future development direction. Meanwhile, CAO also has a wide range of application Spaces, such as bearing fault diagnosis[74], defect identification[75], complex machinery applications[76], waste management techniques[77], forecasting production[78], predictive analysis[79].

Data availability

The code used to evaluate the algorithm CAO is available with the paper. The datasets used and analysed during the current study available from the corresponding author on reasonable request.

Received: 7 March 2025; Accepted: 23 June 2025
Published online: 01 July 2025

References

1. Braik, M. et al. Tornado optimizer with Coriolis force: A novel bio-inspired meta-heuristic algorithm for solving engineering problems. *Artif. Intell. Rev.* **58**, 123 (2025).
2. Ghasemi, M. et al. An efficient bio-inspired algorithm based on humpback whale migration for constrained engineering optimization. *Results Eng.* **25**, 104215 (2025).
3. Kaur, S., Awasthi, L. K., Sangal, A. L. & Dhiman, G. Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Eng. Appl. Artif. Intell.* **90**, 103541 (2020).

4. Ge, Y., Chen, Z. & Liu, Y. An efficient keywords search in temporal social networks. *Data Sci. Eng.* **8**, 368–384 (2023).
5. Zhong, C. Starfish optimization algorithm (SFOA): A bio-inspired metaheuristic algorithm for global optimization compared with 100 optimizers. *Neural Computing and Applications*.
6. Połap, D. & Woźniak, M. Red fox optimization algorithm. *Expert Syst. Appl.* **166**, 114107 (2021).
7. Deng, L. & Liu, S. Snow ablation optimizer: A novel metaheuristic technique for numerical optimization and engineering design. *Expert Syst. Appl.* **225**, 120069 (2023).
8. Li, K. et al. A multi-strategy enhanced northern goshawk optimization algorithm for global optimization and engineering design problems. *Comput. Methods Appl. Mech. Eng.* **415**, 116199 (2023).
9. Hemeida, M. G., Alkhalaf, S., Mohamed, A.-A.A., Ibrahim, A. A. & Senjyu, T. Distributed generators optimization based on multi-objective functions using manta rays foraging optimization algorithm (MRFO). *Energies* **13**, 3847 (2020).
10. Zhao, W., Zhang, Z. & Wang, L. Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Eng. Appl. Artif. Intell.* **87**, 103300 (2020).
11. Alghamdi, T. A. H., Anayi, F. & Packianather, M. Optimal design of passive power filters using the MRFO algorithm and a practical harmonic analysis approach including uncertainties in distribution networks. *Energies* **15**, 2566 (2022).
12. Gao, Y., Zhang, J., Wang, Y., Wang, J. & Qin, L. Love evolution algorithm: A stimulus–value–role theory-inspired evolutionary algorithm for global optimization. *J. Supercomput.* https://doi.org/10.1007/s11227-024-05905-4 (2024).
13. Javed, S. T., Zafar, K. & Younas, I. Kids learning optimizer: Social evolution and cognitive learning-based optimization algorithm. *Neural Comput. Appl.* https://doi.org/10.1007/s00521-024-10009-4 (2024).
14. Gandomi, A. H. & Alavi, A. H. Krill herd: A new bio-inspired optimization algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **17**, 4831–4845 (2012).
15. Abdel-Basset, M., Mohamed, R., Azeem, S. A. A., Jameel, M. & Abouhawwash, M. Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion. *Knowl.-Based Syst.* **268**, 110454 (2023).
16. Amiri, M. H., Mehrabi Hashjin, N., Montazeri, M., Mirjalili, S. & Khodadadi, N. Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm. *Sci. Rep.-uk* **14**, 5032 (2024).
17. Peraza-Vázquez, H., Peña-Delgado, A., Merino-Treviño, M., Morales-Cepeda, A. B. & Sinha, N. A novel metaheuristic inspired by horned lizard defense tactics. *Artif. Intell. Rev.* **57**, 59 (2024).
18. Joshi, S. K. Chaos embedded opposition based learning for gravitational search algorithm. *Appl. Intell.* https://doi.org/10.1007/s10489-022-03786-9 (2022).
19. Zolfi, K. Gold rush optimizer: A new population-based metaheuristic algorithm. *Oper. Res. Decis.* **33**, 113–150 (2023).
20. Mohammadi-Balani, A., Dehghan Nayeri, M., Azar, A. & Taghizadeh-Yazdi, M. Golden eagle optimizer: A nature-inspired metaheuristic algorithm. *Comput. Ind. Eng.* **152**, 107050 (2021).
21. Hamadneh, T. et al. Builder Optimization Algorithm: An effective human-inspired metaheuristic approach for solving optimization problems. *IJIES* **18**, 928–937 (2025).
22. Hamadneh, T. et al. Candle Flame Optimization: A physics-based metaheuristic for global optimization. *IJIES* **18**, 826–837 (2025).
23. El-kenawy, E.-S.M. et al. Greylag Goose Optimization: Nature-inspired optimization algorithm. *Expert Syst. Appl.* **238**, 122147 (2024).
24. Hamadneh, T. et al. MAO Algorithm: A novel approach for engineering design challenges. *IJIES* **18**, 484–493 (2025).
25. Hamadney, T. et al. On the application of tailor optimization algorithm for solving real-world optimization application. *IJIES* **18**, 1–12 (2025).
26. Hamadneh, T. et al. Orangutan Optimization Algorithm: An innovative bio-inspired metaheuristic approach for solving engineering optimization problems. *IJIES* **18**, 47–57 (2025).
27. Hamadneh, T. et al. PPB optimization: A new human-based metaheuristic approach for solving optimization tasks. *IJIES* **18**, 504–519 (2025).
28. Hamadneh, T. et al. Perfumer optimization algorithm: A novel human-inspired metaheuristic for solving optimization tasks. *IJIES* **18**, 633–643 (2025).
29. Hamadneh, T. et al. Revolution optimization algorithm: A new human-based metaheuristic algorithm for solving optimization problems. *IJIES* **18**, 520–531 (2025).
30. Hamadneh, T. et al. Singer optimization algorithm: An effective human-based approach for solving optimization tasks. *IJIES* **18**, 114–126 (2025).
31. Hamadneh, T. et al. Spider-tailed horned viper optimization: An effective bio-inspired metaheuristic algorithm for solving engineering applications. *IJIES* **18**, 25–35 (2025).
32. Tan, Y. & Zhu, Y. *Fireworks Algorithm for Optimization*.
33. Faramarzi, A., Heidarinejad, M., Stephens, B. & Mirjalili, S. Equilibrium optimizer: A novel optimization algorithm. *Knowl.-Based Syst.* **191**, 105190 (2020).
34. Azizi, M., Aickelin, U., Khorshidi, A., Baghalzadeh, H. & Shishehgarkhaneh, M. Energy valley optimizer: A novel metaheuristic algorithm for global and engineering optimization. *Sci. Rep.* **13**, 226 (2023).
35. Shamsaldin, A. S., Rashid, T. A., Al-Rashid Agha, R. A., Al-Salihi, N. K. & Mohammadi, M. Donkey and smuggler optimization algorithm: A collaborative working approach to path finding. *J. Comput. Des. Eng.* **6**, 562–583 (2019).
36. Hasan, N. M. et al. An enhanced donkey and smuggler optimization algorithm for choosing the precise job applicant. *Iran J. Comput. Sci.* **6**, 233–243 (2023).
37. Zhang, M. & Wen, G. Duck swarm algorithm: Theory, numerical optimization, and applications. *Clust. Comput.* **27**, 6441–6469 (2024).
38. Marinaki, M., Taxidou, A. & Marinakis, Y. A hybrid Dragonfly algorithm for the vehicle routing problem with stochastic demands. *Intell. Syst. Appl.* **18**, 200225 (2023).
39. Singh, H., Sawle, Y., Dixit, S., Malik, H. & García Márquez, F. P. Optimization of reactive power using dragonfly algorithm in DG integrated distribution system. *Electr. Power Syst. Res.* **220**, 109351 (2023).
40. Lang, Y. & Gao, Y. Dream Optimization Algorithm (DOA): A novel metaheuristic optimization algorithm inspired by human dreams and its applications to real-world engineering problems. *Comput. Methods Appl. Mech. Eng.* **436**, 117718 (2025).
41. Azizi, M., Talatahari, S. & Gandomi, A. H. Fire Hawk Optimizer: A novel metaheuristic algorithm. *Artif. Intell. Rev.* **56**, 287–363 (2023).
42. Abdollahzadeh, B. et al. Puma optimizer (PO): a novel metaheuristic optimization algorithm and its application in machine learning. *Clust. Comput.* https://doi.org/10.1007/s10586-023-04221-5 (2024).
43. Dewi, S. K. & Utama, D. M. A new hybrid whale optimization algorithm for green vehicle routing problem. *Syst. Sci. Control Eng.* **9**, 61–72 (2021).
44. Deng, H., Liu, L., Fang, J., Qu, B. & Huang, Q. A novel improved whale optimization algorithm for optimization problems with multi-strategy and hybrid algorithm. *Math. Comput. Simul.* **205**, 794–817 (2023).
45. Weng, S., Liu, Z., Fan, Z. & Zhang, G. A whale optimization algorithm-based ensemble model for power consumption prediction. *Electr. Eng.* https://doi.org/10.1007/s00202-024-02611-5 (2024).
46. Zervoudakis, K. & Tsafarakis, S. A mayfly optimization algorithm. *Comput. Ind. Eng.* **145**, 106559 (2020).
47. Yang, K. & Pan, D. An improved mayfly optimization algorithm for type-2 multi-objective integrated process planning and scheduling. *Mathematics* **11**, 4384 (2023).

48. Zhao, Y., Huang, C., Zhang, M. & Lv, C. COLMA: A chaos-based mayfly algorithm with opposition-based learning and Levy flight for numerical optimization and engineering design. *J. Supercomput.* **79**, 19699–19745 (2023).
49. Pham, V. H. S. & Nguyen, V. N. Cement transport vehicle routing with a hybrid sine cosine optimization algorithm. *Adv. Civ. Eng.* **2023**, 1–15 (2023).
50. Hassan, M. H., Daqaq, F., Selim, A., Domínguez-García, J. L. & Kamel, S. MOIMPA: multi-objective improved marine predators algorithm for solving multi-objective optimization problems. *Soft Comput.* https://doi.org/10.1007/s00500-023-08812-7 (2023).
51. Kuşoğlu, M. & Yüzgeç, U. *Multi-Objective Harris Hawks Optimizer for Multiobjective Optimization Problems* (2020).
52. Zulfiqar, R., Javed, T., Anwar Ali, Z. H., Alkhammash, E. & Hadjouni, M. Selection of metaheuristic algorithm to design wireless sensor network. *Intell. Autom. Soft Comput.* **37**, 985–1000 (2023).
53. Dhal, K. G., Sasmal, B., Das, A., Ray, S. & Rai, R. A comprehensive survey on arithmetic optimization algorithm. *Arch. Comput. Methods Eng.* **30**, 3379–3404 (2023).
54. Mahajan, S., Abualigah, L. & Pandit, A. K. Hybrid arithmetic optimization algorithm with hunger games search for global optimization. *Multimed. Tools Appl.* **81**, 28755–28778 (2022).
55. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M. & Gandomi, A. H. The arithmetic optimization algorithm. *Comput. Methods Appl. Mech. Eng.* **376**, 113609 (2021).
56. Jia, Y., Wang, S., Liang, L., Wei, Y. & Wu, Y. A flower pollination optimization algorithm based on cosine cross-generation differential evolution. *Sensors* **23**, 606 (2023).
57. Liu, H., Duan, S. & Luo, H. A hybrid engineering algorithm of the seeker algorithm and particle swarm optimization. *Mater. Test.* **64**, 1051–1089 (2022).
58. Shehadeh, H. A., Mustafa, H. M. J. & Tubishat, M. A hybrid genetic algorithm and sperm swarm optimization (HGASSO) for multimodal functions. *Int. J. Appl. Metaheurist. Comput.* **13**, 1–33 (2022).
59. Tao, L., Yang, X., Zhou, Y. & Yang, L. A novel transformers fault diagnosis method based on probabilistic neural network and bio-inspired optimizer. *Sensors* **21**, 3623 (2021).
60. Rehman, H. et al. Driving training-based optimization (DTBO) for global maximum power point tracking for a photovoltaic system under partial shading condition. *IET Renew. Power Gen.* **17**, 2542–2562 (2023).
61. Mohamed, A.-A.A. et al. Parasitism – Predation algorithm (PPA): A novel approach for feature selection. *Ain Shams Eng. J.* **11**, 293–308 (2020).
62. Nandi, A. & Kamboj, V. K. A Canis lupus inspired upgraded Harris hawks optimizer for nonlinear, constrained, continuous, and discrete engineering design problem. *Numer. Methods Eng.* **122**, 1051–1088 (2021).
63. Tu, B., Wang, F., Huo, Y. & Wang, X. A hybrid algorithm of grey wolf optimizer and Harris Hawks optimization for solving global optimization problems with improved convergence performance. *Sci. Rep.-uk* **13**, 22909 (2023).
64. Fathiamuthal Rajeena, P. P., Ismail, W. N. & Ali, M. A. S. A Metaheuristic Harris Hawks optimization algorithm for weed detection using drone images. *Appl. Sci.* **13**, 7083 (2023).
65. Too, J., Abdullah, A. R. & Mohd Saad, N. A new quadratic binary harris hawk optimization for feature selection. *Electronics* **8**, 1130 (2019).
66. Chong, G. & Yuan, Y. An improved Harris Hawk optimization algorithm. *Mech. Eng. Sci.* **6**, 13224 (2024).
67. Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Computat.* **6**, 182–197 (2002).
68. Shang, Q. et al. A multi-stage competitive swarm optimization algorithm for solving large-scale multi-objective optimization problems. *Expert Syst. Appl.* **260**, 125411 (2025).
69. Yang, X.-S. & Deb, S. Multiobjective cuckoo search for design optimization. *Comput. Oper. Res.* **40**, 1616–1624 (2013).
70. Price, K. V., Awad, N. H., Ali, M. Z. & Suganthan, P. N. *Problem Definitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization*.
71. Kudela, J. A critical problem in benchmarking and analysis of evolutionary computation methods. *Nat. Mach. Intell.* **4**, 1238–1245 (2022).
72. Huang, X., Xu, R., Yu, W. & Wu, S. Evaluation and analysis of heuristic intelligent optimization algorithms for PSO, WDO, GWO and OOBO. *Mathematics* **11**, 4531 (2023).
73. Dhiman, G. & Kaur, A. STOA: A bio-inspired based optimization algorithm for industrial engineering problems. *Eng. Appl. Artif. Intell.* **82**, 148–174 (2019).
74. Peraza-Vázquez, H. et al. A bio-inspired method for engineering design optimization inspired by dingoes hunting strategies. *Math. Probl. Eng.* **2021**, 1–19 (2021).
75. Vashishtha, G., Chauhan, S., Singh, M. & Kumar, R. Bearing defect identification by swarm decomposition considering permutation entropy measure and opposition-based slime mould algorithm. *Measurement* **178**, 109389 (2021).
76. Chauhan, S. & Vashishtha, G. A synergy of an evolutionary algorithm with slime mould algorithm through series and parallel construction for improving global optimization and conventional design problem. *Eng. Appl. Artif. Intell.* **118**, 105650 (2023).
77. Mahmoud, M. A review on waste management techniques for sustainable energy production. *MOR* **3**, 47–58 (2025).
78. Mishra, P. et al. Forecasting production of potato for a sustainable future: global market analysis. *Potato Res.* **67**, 1671–1690 (2024).
79. Khaled, K. & Singla, M. K. Predictive analysis of groundwater resources using random forest regression. *JAIM* **09**, 11–19 (2025).

Author contributions

Da Fang performed the data analysis; Jun Yan performed the formal analysis; Quan Zhou performed the validation; Da Fang wrote the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Ethical approval

This material is the authors' own original work, which has not been previously published elsewhere. The paper is not currently being considered for publication elsewhere. The paper reflects the authors' own research and analysis in a truthful and complete manner.

Additional information

Correspondence and requests for materials should be addressed to D.F.

Reprints and permissions information is available at www.nature.com/reprints.