Mathematical Biosciences and Engineering

MBE, 19(11): 10963–11017.
DOI: 10.3934/mbe.2022512
Received: 18 June 2022
Revised: 13 July 2022
Accepted: 18 July 2022
Published: 1 Aug 2022

http://www.aimspress.com/journal/MBE

Research article

IHAOAVOA: An improved hybrid aquila optimizer and African vultures optimization algorithm for global optimization problems

Yaning Xiao, Yanling Guo*, Hao Cui, Yangwei Wang, Jian Li and Yapeng Zhang

College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China

* **Correspondence:** Email: nefugyl@hotmail.com.

Abstract: Aquila Optimizer (AO) and African Vultures Optimization Algorithm (AVOA) are two newly developed meta-heuristic algorithms that simulate several intelligent hunting behaviors of Aquila and African vulture in nature, respectively. AO has powerful global exploration capability, whereas its local exploitation phase is not stable enough. On the other hand, AVOA possesses promising exploitation capability but insufficient exploration mechanisms. Based on the characteristics of both algorithms, in this paper, we propose an improved hybrid AO and AVOA optimizer called IHAOAVOA to overcome the deficiencies in the single algorithm and provide higher-quality solutions for solving global optimization problems. First, the exploration phase of AO and the exploitation phase of AVOA are combined to retain the valuable search competence of each. Then, a new composite opposition-based learning (COBL) is designed to increase the population diversity and help the hybrid algorithm escape from the local optima. In addition, to more effectively guide the search process and balance the exploration and exploitation, the fitness-distance balance (FDB) selection strategy is introduced to modify the core position update formula. The performance of the proposed IHAOAVOA is comprehensively investigated and analyzed by comparing against the basic AO, AVOA, and six state-of-the-art algorithms on 23 classical benchmark functions and the IEEE CEC2019 test suite. Experimental results demonstrate that IHAOAVOA achieves superior solution accuracy, convergence speed, and local optima avoidance than other comparison methods on most test functions. Furthermore, the practicality of IHAOAVOA is highlighted by solving five engineering design problems. Our findings reveal that the proposed technique is also highly competitive and promising when addressing real-world optimization tasks. The source code of the IHAOAVOA is publicly available at https://doi.org/10.24433/CO.2373662.v1.

Keywords: aquila optimizer; African vultures optimization algorithm; hybrid algorithm; composite opposition-based learning; fitness-distance balance; global optimization

1. Introduction

Optimization is essentially the process of determining the optimal solution for a given problem among all potential solutions to achieve maximum profit, productivity, and efficiency [1–4]. Over the past several decades, with the development of human society and modern science, the complexity of optimization problems in the real world has been increasing sharply, thus putting higher demands on the reliability and effectiveness of optimization techniques [5,6]. In general, existing optimization technology can be classified into deterministic algorithms and meta-heuristic algorithms (MAs) [7]. For a deterministic algorithm, candidate solutions are generated using the same initial values according to the analytical properties of problems and converge mechanically toward the global optimum without any randomness. Newton-Raphson method and Conjugate Gradient are two representative deterministic algorithms. Although this type of algorithm can provide satisfactory solutions in solving certain nonlinear problems, it needs the derivative information of the problem and frequently falls into the local optima when confronting the challenges of multimodal, large-scale, and sub-optimal search space [8]. Recently, as an ideal alternative to deterministic algorithms, MAs have attracted the attention of more and more scholars worldwide due to their simple structure, low computational consumption, no need for gradient information, and powerful local optimal avoidance capability. Based on the requirements of the objective function, such algorithms iteratively use different operators to randomly sample the search space to acquire better decision variables [9,10]. Compared with traditional methods, these merits enable MAs to find the global optimal solution for complex optimization problems more effectively. Therefore, MAs have been widely applied in a variety of research areas, such as engineering design [11–14], feature selection [15–17], photovoltaic (PV) parameter extraction [18–21], image segmentation [22–24], and path planning [25].

As their name implies, MAs build optimization models by imitating a series of natural stochastic phenomena. On the basis of different design inspirations, MAs can be divided into four dominant classes (as illustrated in Figure 1) [1]: evolutionary algorithms, physics-based algorithms, swarm-based algorithms, and human-based algorithms. Evolutionary algorithms stem from the mechanisms of biological evolution, such as selection, mutation, recombination, and elimination. One of the most used algorithms in this category is the Genetic Algorithm (GA) [26], which simulates Darwinian evolution theory. Some other well-known evolutionary algorithms include Genetic Programming (GP) [27], Differential Evolution (DE) [28], Evolution Strategy (ES) [29], and Biogeography-Based Optimization (BBO) [30]. Physics-based algorithms are mainly inspired by the physical laws of the surrounding world. Examples of such algorithms contain Simulated Annealing (SA) [31], Gravity Search Algorithm (GSA) [32], Multi-Verse Optimizer (MVO) [33], Atom Search Optimization (ASO) [34], Black Hole Algorithm (BHA) [35], Sine Cosine Algorithm (SCA) [36], Thermal Exchange Optimization (TEO) [37], and Arithmetic Optimization Algorithm (AOA) [38]. Swarm-based algorithms originate from the self-organization and collective behaviors of organisms in nature. Particle Swarm Algorithm (PSO) [39] is considered the most classic embodiment of this branch, which searches for the optimal solution to a problem by emulating the collaborative foraging of bird flocks. Of course, there are many other very famous swarm-based algorithms such as Ant Colony

Optimization (ACO) [40], Dragonfly Algorithm (DA) [41], Ant Lion Optimizer (ALO) [42], Whale Optimization Algorithm (WOA) [43], Grey Wolf Optimizer (GWO) [44], and Salp Swarm Algorithm (SSA) [45]. The fourth category is human-based algorithms, derived from some human activities in the community. Examples of such algorithms are Tabu Search (TS) [46], Harmony Search (HS) [47], Search Group Algorithm (SGA) [48], Imperialist Competitive Algorithm (ICA) [49], and Teaching Learning-Based Optimization (TLBO) [50]. In addition to the above algorithms, more MAs have been proposed in recent years, like Moth-Flame Optimization (MFO) [51], Slime Mould Algorithm (SMA) [52], Tunicate Swarm Algorithm (TSA) [53], Harris Hawks Optimization (HHO) [54], Gorilla Troops Optimizer (GTO) [55], Remora Optimization Algorithm (ROA) [56], Hunger Games Search (HGS) [57], and Reptile Search Algorithm (RSA) [58]. Although these nature-inspired MAs share distinct characteristics, they all have two important phases in the search gradation: exploration and exploitation [59,60]. In the exploration phase, search agents explore the whole target space as much as possible to find the parts that may have the optimal solution. Then, in the exploitation phase, more local searches are conducted to improve the quality and precision of the gained optimal solution. For a well-organized optimizer, it is vital to maintain a proper balance between exploration and exploitation.

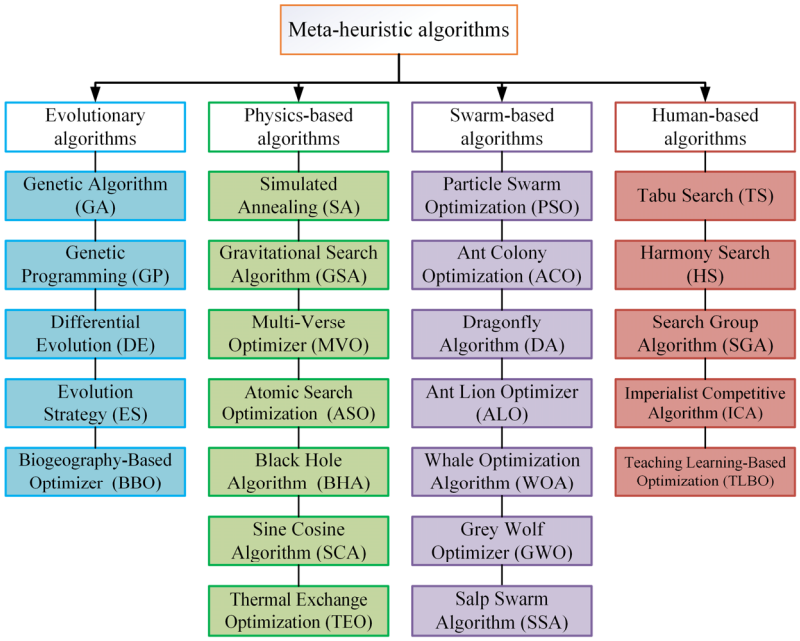


Figure 1. Classification of meta-heuristic algorithms.

Despite the success of MAs in many aspects of computational science, they may still suffer from slow convergence speed, the tendency to fall into the local optima, and premature convergence [61,62]. As stated in the No-Free-Lunch (NFL) theorem [63], no one algorithm can work for all kinds of optimization problems. Therefore, motivated by this theorem, numerous scholars dedicate themselves to designing new MAs or enhancing existing ones. Nowadays, apart from adding some effective search strategies, it has become one popular trend to hybridize the two basic MAs for better comprehensive performance in the improvements of existing algorithms. Unlike the single algorithm, a hybrid algorithm promotes diversity and shares more useful information within the population, which endows

it with a stronger search capability. For example, Zheng et al. [60] introduced AOA into SMA and constructed a new hybrid optimization algorithm called DESMAOA. Compared with basic algorithms, experimental results suggested that DESMAOA has a high superiority on 23 standard benchmark functions and three engineering design problems. Chakraborty et al. [64] integrated WOA and HGS into an efficient hybrid optimizer named HSWOA, which has been successfully applied to solve seven real-world engineering problems and IEEE CEC2019 test set. Pirozmand et al. [65] presented a novel hybrid technique based on GA and GSA to address task scheduling problems in cloud infrastructure. Bao et al. [22] proposed the HHO-DE algorithm for multi-level thresholding color image segmentation by incorporating HHO and DE. Besides, Abdel-Mawgoud et al. [66] combined SCA with MFO and used this hybrid approach to find the optimal allocation of distributed generations and capacitors in distribution networks.

In this paper, we focus on the two latest swarm-based MAs, namely Aquila Optimizer (AO) [67] and African Vultures Optimization Algorithm (AVOA) [68]. The AO algorithm was first proposed in 2021, which simulates four unique hunting methods of Aquila. Since AO has powerful robustness and global exploration capability, it has been extensively applied to lots of scenarios. Guo et al. [69] adopted AO to adjust the proportional-integral-derivative (PID) coefficients of the phase-locked loop (PLL), a key component in the PV inverter, to smooth power fluctuations and improve the quality of grid connection. Experimental results demonstrated that the AO-optimized PLL adjustment strategy could effectively reduce power fluctuations and overshoot with a short response time. Hussan et al. [70] used AO to optimize the selective harmonic elimination equations for the seven-level H-bridge inverter to decrease the component count and total harmonic distortion. Vashishtha et al. [71] applied AO to determine the optimal minimal entropy deconvolution (MED) filter length to boost the recognition accuracy during the bearing fault diagnosis of the Francis turbine. AlRassas et al. [72] adopted AO to identify the optimal parameters of the adaptive neuro-fuzzy inference system (ANFIS) network to increase its prediction accuracy in oil production time series forecasting. In [73], AO is employed to address the stochastic optimal power flow (SCOPF) problem to obtain the best dispatch power from wind farms while minimizing total operating costs. These researches all have proven that AO is a promising optimization tool. However, similar to other MAs, the basic AO algorithm inevitably has the defects of premature convergence and being prone to falling into local minima, mainly caused by its insufficient exploitation phase. As a result, many improved and hybrid attempts have been implemented to enhance the performance of AO. Zhao et al. [74] developed a heterogeneous AO (HAO) based on the multiple updating mechanism to enhance the search capability of the algorithm and alleviate the stagnation in the later exploitation phase. Kandan et al. [75] proposed a novel quasi-oppositional AO called QOAO for solving the issue of resource allocation and management in the internet of things (IoT)-enabled cloud environment. The quasi-oppositional-based learning is used to diversify the initial population and help the algorithm eliminate the local optima. Li et al. [76] proposed an improved variant of AO, namely IAO, to provide the optimal configuration for combined cooling, heating, and power (CCHP) system, which integrated the self-adaptive weight and Logistic chaotic mapping to facilitate the possibility of finding the high-precision solution. In [77], a simplified AO algorithm was developed by removing the equations controlling the exploitation phase and retaining the two exploration tactics. Simulation results on unimodal, multimodal, and the CEC2021 test suite fully validated the superiority of this method. Mahajan et al. [78] blended AO and AOA for complex numerical optimization. The convergence speed and stability of the hybrid algorithm are significantly strengthened in comparison to the basic AO and AOA. Wang et al. [12] presented an excellent hybrid

optimizer known as IHAOHHO by combining the exploration phase of AO and the exploitation phase of HHO. Meanwhile, the random opposition-based learning and nonlinear escaping energy parameter mechanisms are introduced into the hybrid algorithm to further boost its exploration ability and local optima avoidance. The worth of the IHAOHHO algorithm is well reflected in settling industrial engineering optimization tasks. Yao et al. [25] constructed an improved hybrid algorithm named IHSSAO by combining AO with SSA and pinhole imaging opposition-based learning. The IHSSAO algorithm is able to balance exploration and exploitation well and provide the shortest global path for unmanned aerial vehicle (UAV) path planning in complex terrain. Zhang et al. [79] proposed a hybrid AOAAO algorithm for tackling benchmark function optimization and engineering design problems.

For another algorithm concerned in this paper, AVOA was also developed in 2021. This algorithm mimics the foraging and navigation behaviors of African vultures in nature and has drawn many scholars to apply it to resolve real-world optimization problems [80–82]. In contrast to the AO algorithm, AVOA possesses strong exploitation mechanisms, but its exploration capability and convergence speed are not satisfactory [83]. Due to the relatively short time since the algorithm has been proposed, there are few studies on the improvement of AVOA.

Given the above discussion, this paper tries to hybridize the AO and AVOA algorithms to give full play to the advantages of both and achieve better overall optimization performance, and then proposes a novel improved hybrid meta-heuristic algorithm for global optimization, namely IHAOAVOA. To be specific, first, we integrate the exploration phase of AO and the exploitation phase of AVOA, which extracts and inherits the robust exploration and exploitation capabilities of the two basic algorithms. Then, a new composite opposition-based learning (COBL) mechanism is designed and embedded into the hybrid algorithm to avoid the local optima and increase the population diversity. Finally, the fitness-distance balance (FDB) selection method is utilized to select one candidate solution with the highest score from the population to replace the original random individual in the position update formula. This is considered from boosting the search efficiency and balancing the exploration and exploitation trends of the hybrid algorithm. To verify the effectiveness and practicality of IHAOAVOA, 23 classical benchmark functions, IEEE CEC2019 test suite, and five real-world engineering design problems are used for the tests. And the proposed method is compared with the basic AO, AVOA, and six state-of-the-art MAs, including SCA, WOA, GWO, MFO, TSA, and AOA. Experimental results indicate that the proposed IHAOAVOA performs better than other competitors with regard to solution accuracy, convergence speed, stability, and local optima avoidance. The main contributions of this paper are summarized as follows:

- IHAOAVOA, a novel hybrid improved algorithm based on the Aquila Optimizer (AO) and African Vultures Optimization Algorithm (AVOA), is proposed to solve global optimization problems.
- A new mechanism called composite opposition-based learning (COBL) and fitness-distance balance (FDB) selection method are carried out to enhance the searchability of the hybrid algorithm.
- The proposed method (IHAOAVOA) is tested on several optimization problems, including 23 classical benchmark functions, IEEE CEC2019 test suite, and five engineering design problems, and compared with different state-of-the-art MAs.
- Experimental results suggest that IHAOAVOA has more reliable performance than other comparison optimization algorithms.

The structure of this paper is organized as follows. Section 2 presents a brief overview of the basic AO and AVOA algorithms, as well as COBL and FDB strategies. Section 3 describes the proposed IHAOAVOA algorithm in detail. Section 4 evaluates the performance of IHAOAVOA on

benchmark functions and analyzes the obtained experimental results. In Section 5, the proposed IHAOAVOA is applied to solve five real-world engineering design problems. Finally, Section 6 concludes the paper and discusses potential research directions.

2. Preliminaries

2.1. Aquila Optimizer (AO)

Aquila Optimizer (AO) is a new bionic, gradient-free, and swarm-based meta-heuristic algorithm developed by Abualigah et al. [67] in 2021. The main inspiration of this algorithm derives from the hunting behavior of the Aquila, a famous bird of prey found in the Northern Hemisphere. Aquila exerts its fast speed and dexterity, as well as strong feet and sharp talons, to snatch rabbits, marmots, and many other ground animals. During the foraging activities, four different strategies are recognized to be utilized by the Aquila, including: 1) High-altitude soar with vertical stoop; 2) Contour flight along with short glide attack; 3) Low flight along with slow descent attack; 4) Capturing the prey while walking. Thus, the optimization procedure of the AO algorithm can be modeled into four discrete phases, which are briefly described as follows.

In AO, Aquilas are candidate solutions and the best solution in each step is defined as the intended prey. First, as with the fundamental framework of other optimization paradigms, the initial population of AO is generated randomly in the search space of the given problem using Eq (1).

$$X_i = rand \times (ub - lb) + lb, i = 1,2, \dots, N \tag{1}$$

where X_i denotes the position of i-th Aquila in the population, $rand$ denotes a random number within the interval of 0 and 1, N denotes the total number of Aquilas, i.e., population size, ub and lb demonstrate the upper and lower bounds of the search domain, respectively.

To lay a good foundation for the smooth transition from global exploration to local exploitation, AO establishes the following switching condition:

$$\begin{cases} \text{Execution of exploration}, if\ t \le \left(\frac{2}{3}\right) \times T \\ \text{Execution of exploitation}, otherwise \end{cases} \tag{2}$$

where t is the current iteration, and T is the maximum number of iterations. Next, the four phases involved in the mathematical model of AO are presented.

2.1.1. Expanded exploration: high-altitude soar with vertical stoop

In this phase, Aquila flies high over the ground to explore the hunting area extensively, and once the prey is detected, it will make a vertical dive towards the intended prey. This behavior is simulated as in Eq (3).

$$X_i(t+1) = X_{best}(t) \times \left(1 - \frac{t}{T}\right) + X_m(t) - X_{best}(t) \times rand \tag{3}$$

where $X_i(t+1)$ refers to the updated position of i-th Aquila in the next iteration t, $X_{best}(t)$ indicates the location of the prey, i.e., optimal solution found so far, t and T are the current number

of iterations and the maximum iteration, respectively. And $X_m(t)$ represents the average position of all Aquilas in the population, which is calculated as follows:

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \tag{4}$$

where $X_i(t)$ is the current position vector of i-th Aquila, and N is the population size.

2.1.2. Narrowed exploration: contour flight along with short glide attack

In the second phase, Aquila circles above the target prey determined from a high soar, gets ready to land, and then launches an attack. The mathematical model of this behavior is expressed as follows:

$$X_i(t+1) = X_{best}(t) \times \text{Levy}(D) + X_r(t) + (y - x) \times rand \tag{5}$$

where X_r indicates a random position of Aquila selected from the current population $[1, N]$. Levy $(\cdot)$ implies the Lévy flight function, which is presented as follows:

$$\text{Levy}(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(1+\beta) \times \beta \times 2^{(\frac{\beta-1}{2})}}\right)^{\frac{1}{\beta}} \tag{6}$$

where u and v are random numbers within the interval $[0, 1]$, $\Gamma(\cdot)$ denotes the gamma function, and β is a constant value equal to 1.5. In Eq (5), y and x stand for the contour spiral shape during the search, which can be calculated as follows:

$$\begin{cases} x = (r + U \times D_1) \times sin(-\omega \times D_1 + \frac{3\times\pi}{2}) \\ y = (r + U \times D_1) \times cos(-\omega \times D_1 + \frac{3\times\pi}{2}) \end{cases} \tag{7}$$

where r denotes the number of search cycles between 1 and 20, U is a constant fixed to 0.00565, D_1 is a vector of integers from 1 to the dimension size (D), and ω is also a small value equivalent to 0.005.

2.1.3. Expanded exploitation: low flight along with slow descent attack

As the area of the prey is precisely specified, Aquila descends vertically to perform a preliminary attack to probe the prey's response. Here, AO exploits the selected area to approach and attack the prey. The position update formula of Aquila in this phase is described as follows:

$$X_i(t+1) = (X_{best}(t) - X_m(t)) \times \alpha - rand + ((ub - lb) \times rand + lb) \times \delta \tag{8}$$

where α and δ are the exploitation control coefficients set as 0.1.

2.1.4. Narrowed exploitation: walk and catch the prey

In the fourth phase, Aquila comes to the land and pursues the prey according to its random motion trajectory, and finally, Aquila will attack the prey at the appropriate moment. The mathematical representation of this case is given as:

$$X_i(t+1) = QF \times X_{best}(t) - G_1 \times X_i(t) \times rand - G_2 \times \text{Levy}(D) + G_1 \times rand \tag{9}$$

$$QF(t) = t^{\frac{2\times rand-1}{(1-T)^2}} \tag{10}$$

$$\begin{cases} G_1 = 2 \times rand - 1 \\ G_2 = 2 \times (1 - \frac{t}{T}) \end{cases} \tag{11}$$

where QF refers to the quality function used to balance the search strategy, G_1 indicates the movement parameter of Aquila while tracking the prey, which is a random number between -1 and 1, while G_2 denotes the flight slope in the process of Aquila chasing the prey from the first to the last location, which decreases linearly from 2 to 0.

The flow chart of the basic AO is illustrated in Figure 2.

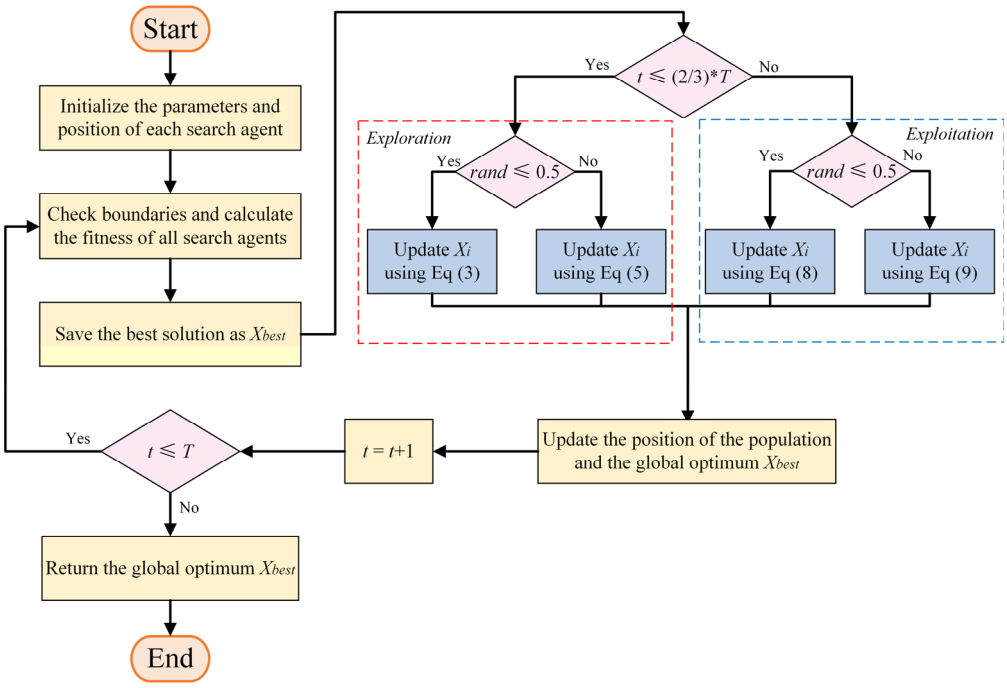


Figure 2. Flow chart of the basic AO algorithm.

2.2. African Vultures Optimization Algorithm (AVOA)

As a novel population-based optimization technique proposed by Abdollahzadeh et al. [68] in 2021, AVOA mimics the living habit and foraging behavior of African vulture. African vultures rarely launch an offensive against healthy animals, but may kill a weak or diseased animal and even feed on the human carcass. One interesting feature of these predatory birds is their bald heads, which play an important role in regulating the body temperature and protecting themselves from bacteria and getting sick. In natural circumstances, vultures continuously travel long distances from one place to another to discover better food sources, and rotational flight is a common mode of flight for them. Frequently, after a food supply is located, the vultures will come into conflict with each other to achieve more allocations. The weak vultures surround the stronger vultures and wait to receive food

until the latter become tired of eating. With the above biological concepts, the mathematical model of the AVOA algorithm is accomplished in four separate phases. A brief description of each step is presented as follows.

2.2.1. Phase one: determining the best vulture in any group

Once the initial random population of the AVOA algorithm is generated, the objective values of all solutions are evaluated, where the best solution is picked as the best vulture in the first group and the vulture corresponding to the second-best solution is placed in the second group. Besides, the rest of the vultures are arranged in the third group. Since these two best vultures have guiding effects, Eq (12) is designed to help the current individual determine which vulture it should move towards in each iteration.

$$X_B = \begin{cases} Bestvulture_1, if\ p_i = L_1 \\ Bestvulture_2, if\ p_i = L_2 \end{cases} \tag{12}$$

where X_B denotes the best vulture selected, $Bestvulture_1$ and $Bestvulture_2$ denote the best vultures of the first group and second group, respectively, L_1 and L_2 represent two parameters between 0 and 1 measured before the optimization operation, where $L_1 + L_2 = 1$. The probability of selecting the best solution from each group p_i is calculated according to the *Roulette Wheel* mechanism, and its formula is as follows.

$$p_i = \frac{f_i}{\sum_{i=1}^{m} f_i} \tag{13}$$

where f_i means the fitness values of vultures, and m is the total number of vultures in the first and second groups.

2.2.2. Phase two: starvation rate of vultures

When vultures feel satiated, they have high energy levels allowing them to go longer distances to seek food. Conversely, if they don't have adequate energy, hungry vultures will become aggressive and thus fight with the nearby stronger vultures to obtain free food. Based on this, the starvation degree of vultures is modeled as follows:

$$F = (2 \times rand + 1) \times z \times (1 - \frac{t}{T}) + g \tag{14}$$

$$g = h \times (sin^w(\frac{\pi}{2} \times \frac{t}{T}) + cos(\frac{\pi}{2} \times \frac{t}{T}) - 1) \tag{15}$$

where F means the hunger degree of vultures, $rand$ is a random number between 0 and 1, z is a random number between -1 and 1, t and T are the current number of iterations and the maximum iteration, respectively, h is a random number within the interval $[-2, 2]$, and w signifies a constant.

As we see from Eq (14), the parameter F shows a decreasing trend with the increasing number of iterations. Therefore, it is also used to construct the transition between the exploration phase and the exploitation phase in the AVOA algorithm. Here, in the case of $|F| \geq 1$, it means the vulture is satiated and searches for new food in different areas, which is also known as the exploration phase. On

the other hand, when $|F| < 1$, the vulture hunts for food in the neighborhood of the solutions, and AVOA enters the exploitation phase.

2.2.3. Phase three: exploration

In nature, vultures have excellent visual skills to spot poor dying animals. When vultures begin foraging, they first spend a lot of time carefully scrutinizing their living environment and then go long distances to search for food. Considering the habits of vultures, two distinct mechanisms are designed in the exploration stage of AVOA so as to explore different random regions as much as possible. Each mechanism is selected by using a parameter called P_1, which must be assigned a value within the interval [0, 1] before the search operation. The mathematical model can be expressed as follows.

If $rand \leq P_1$:

$$X_i(t+1) = X_B(t) - D_i(t) \times F \tag{16}$$

$$D_i(t) = |C \times X_B(t) - X_i(t)| \tag{17}$$

If $rand > P_1$:

$$X_i(t+1) = X_B(t) - F + rand \times \big((ub - lb) \times rand + lb\big) \tag{18}$$

where $X_i(t+1)$ denotes the position vector of i-th vulture in the next iteration t, $X_i(t)$ denotes the current position of i-th vulture, $X_B(t)$ denotes the current best vulture selected according to Eq (12), F describes the hunger rate of vultures calculated by Eq (14), C is a random number in the range $[0, 2]$, ub and lb are the upper and lower bounds of the search range.

2.2.4. Phase four: exploitation

When the value of $|F|$ is less than 1, AVOA performs the exploitation phase, which further contains two stages with two different mechanisms. Likewise, in each internal stage, the selection or not of each mechanism is decided by two parameters, namely P_2 and P_3. The parameter P_2 is used to choose the mechanism available in the first stage and parameter P_3 is utilized to select the mechanism available in the second stage, both of which need to be valued in the range of 0 and 1 before optimization.

- Exploitation (Stage 1)

If the value of $|F|$ is ranged in the interval $[0.5, 1]$, the algorithm proceeds to the first part of exploitation. Here, two behaviors are carried out: siege-fight and rotating flight. When $|F| \geq 0.5$, the vultures are relatively satiated and energetic. At such time, vultures with great physical strength are reluctant to share food with other vultures, while the weaker vultures attempt to get food from the strong ones by gathering together and provoking small conflicts to make them exhausted. This behavior can be simulated as follows:

$$X_i(t+1) = D_i(t) \times (F + rand) - d_i(t) \tag{19}$$

In Eq (19), $d_i(t)$ indicates the distance between the i-th vulture and the current best vulture, which is calculated as follows:

$$d_i(t) = X_B(t) - X_i(t) \tag{20}$$

In addition to the behavior described above, vultures often make a rotational flight, which is similar to Spiral Motion. To model this process, a spiral equation is developed between all vultures and one of the two best vultures. The mathematical expression is given by:

$$X_i(t+1) = X_B(t) - (S_1(t) + S_2(t)) \tag{21}$$

$$S_1(t) = X_B(t) \times \left(\frac{rand \times X_i(t)}{2\pi}\right) \times cos(X_i(t)) \tag{22}$$

$$S_2(t) = X_B(t) \times \left(\frac{rand \times X_i(t)}{2\pi}\right) \times sin(X_i(t)) \tag{23}$$

- Exploitation (Stage 2)

If the value of $|F|$ is less than 0.5, the algorithm enters the second part of exploitation. At this stage, the accumulation of vultures over the food source and violent siege-strife mechanism are implemented. When $|F| < 0.5$, almost all vultures in the population are well full, but the two best vultures become hungry after prolonged exertion. Due to a large amount of food has been consumed at this time, it may happen that many types of vultures gather on a single food resource and compete against each other. In this situation, the position update formula of vultures is expressed as follows:

$$X_i(t+1) = \frac{A_1(t) + A_2(t)}{2} \tag{24}$$

$$A_1(t) = Bestvulture_1(t) - \frac{Bestvulture_1(t) \times X_i(t)}{Bestvulture_1(t) - X_i(t)^2} \times F \tag{25}$$

$$A_2(t) = Bestvulture_2(t) - \frac{Bestvulture_2(t) \times X_i(t)}{Bestvulture_2(t) - X_i(t)^2} \times F \tag{26}$$

On the other hand, in the quest for the little food left, the other vultures will also turn vicious and make their way in various directions toward the head vulture. This movement is simulated as in Eq (27).

$$X_i(t+1) = X_B(t) - |d_i(t)| \times F \times \text{Levy}(D) \tag{27}$$

where $d_i(t)$ is calculated according to Eq (20), D is the problem dimension, and Levy $(\cdot)$ denotes the Lévy flight function used to boost the effectiveness of the AVOA. Same as that in AO, the mathematical expression of Lévy flight is as follows:

$$\text{Levy}(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(1+\beta) \times \beta \times 2^{(\frac{\beta-1}{2})}}\right)^{\frac{1}{\beta}} \tag{28}$$

where u and v are random numbers within the interval $[0, 1]$, $\Gamma(\cdot)$ is the gamma function, and β is a constant fixed to 1.5.

The flow chart of the basic AVOA is illustrated in Figure 3.

2.3. Composite opposition-based learning (COBL)

Opposition-based learning (OBL) [84] is a powerful optimization tool in intelligent computing, which has been successfully used to improve different native meta-heuristic algorithms [11,85–87]. The optimization procedure often starts with an initial stochastic solution. If this initial solution is near the global optimal solution, the algorithm converges quickly. On the contrary, the initial solution may be far from the optimum or just in the opposite direction, which will cause it to take quite a long time to converge or even fall into a stagnant state [88]. The main ideology of OBL is to simultaneously evaluate the fitness values of the current solution as well as its inverse solution, and then the fitter one is retained to participate in the subsequent iterative calculation. Therefore, OBL can effectively increase the probability of finding a better candidate solution. However, it has been indicated that OBL can only generate the inverse solution at a fixed position in optimization, and it still fails to ameliorate the defects of the algorithm when solving complex problems [1,89]. In recent years, more and more enhanced variants of OBL have been proposed, of which lens opposition-based learning (LOBL) [90] and random opposition-based learning (ROBL) [91] are two typical examples. Both methods are effective in improving the ability of the algorithm to avoid falling into local optima, where LOBL can also considerably boost the convergence speed of the algorithm, and ROBL has a unique strength in enriching the population diversity [12]. Considering the superior performance of the two forms of opposition-based learning, we integrate them and propose a novel search strategy: composite opposition-based learning (COBL). As illustrated in Figure 4, the basic principles of LOBL and ROBL will be described first below.

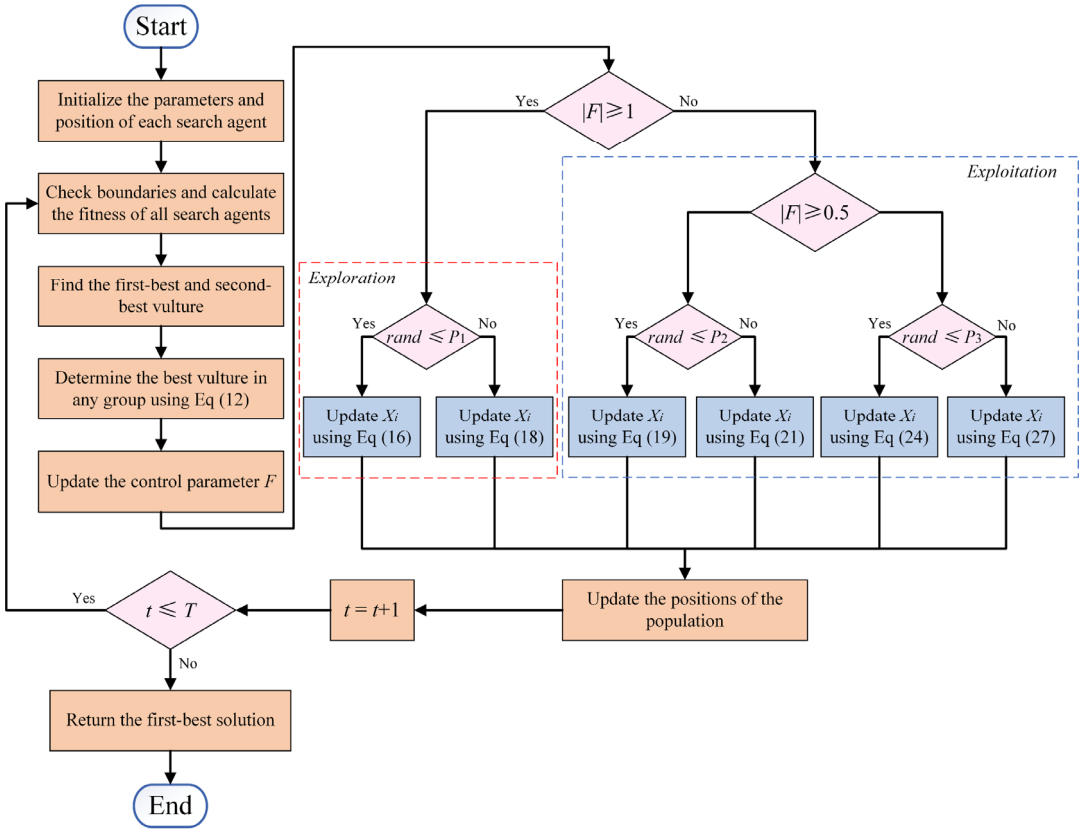
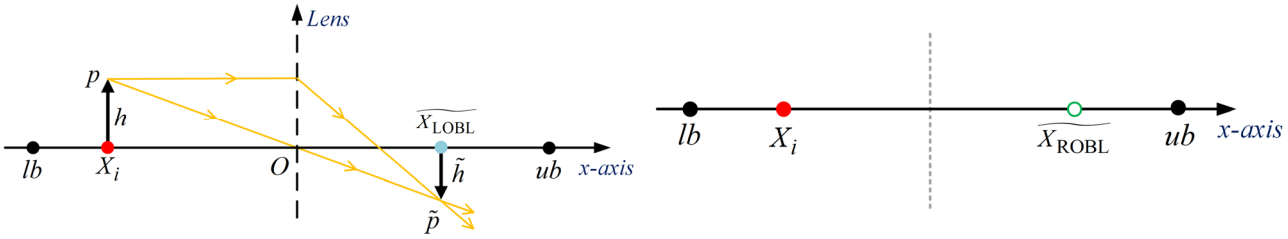


Figure 3. Flow chart of the basic AVOA algorithm.



(a) Lens opposition-based learning (b) Random opposition-based learning

Figure 4. Principle of lens opposition-based learning and random opposition-based learning.

Lens imaging is a common optical phenomenon that specifically refers to when an object is placed at more than twice principal focal lengths away from the convex lens, an inverted and contracted image will be produced on the other side of the lens. Take the one-dimensional search space in Figure 4(a) for instance, the cardinal point O represents the midpoint of the search range $[lb, ub]$, and the y-axis is considered a convex lens. Besides, there is an object p with height h located at the point X_i (X_i is the i-th solution in the population), which is outside twice the lens's focal length. Through lens imaging, the corresponding image $\tilde{p}$ with the height $\tilde{h}$ can be obtained, and its projection on the coordinate axis is $\widetilde{X_{\mathrm{LOBL}}}$. Consequently, the geometric relationship in the figure can be formulated as follows.

$$\frac{(lb+ub)/2-X_i}{\widetilde{X_{\mathrm{LOBL}}}-(lb+ub)/2}=\frac{h}{\tilde{h}} \tag{29}$$

Let $k=h/\tilde{h}$, the opposite solution $\widetilde{X_{\mathrm{LOBL}}}$ based on the theory of lens imaging is calculated by modifying the Eq (29):

$$\widetilde{X_{\mathrm{LOBL}}}=\frac{(lb+ub)}{2}+\frac{(lb+ub)}{2k}-\frac{X_i}{k} \tag{30}$$

Compared to the complex metaphors of the former, ROBL has a much simpler concept. In the search space of Figure 4(b), the point X_i on the x-axis denotes the i-th solution in the population, and its random opposite solution $\widetilde{X_{\mathrm{ROBL}}}$ can be defined by:

$$\widetilde{X_{\mathrm{ROBL}}}=lb+ub-rand\times X_i \tag{31}$$

From Eq (31), it can be seen that the generated inverse solution has good randomness for exploration, which greatly helps to provide more population diversity at the later stage of the search, thus avoiding the algorithm from falling into the local optima.

To make full use of the characteristics of LOBL and ROBL, a probability of 50% is assumed to choose between them in the optimization process. Finally, the mathematical expression of the developed COBL is given as follows.

$$\widetilde{X_{\mathrm{COBL}}}=\begin{cases}lb+ub-rand\times X_i, if\ q<0.5\\ \frac{(lb+ub)}{2}+\frac{(lb+ub)}{2k}-\frac{X_i}{k}, otherwise\end{cases} \tag{32}$$

where X_i is the i-th solution in the population, $\widetilde{X_{\mathrm{COBL}}}$ is the opposite solution of X_i generated by COBL, q illustrates a random number in $[0,1]$, k represents the distance coefficient, ub and lb

are the upper and lower bounds of the search space.

Generally, most optimization problems are multi-dimensional, so the above Eq (32) can also be extended into *D*-dimensional space as follows:

$$\widetilde{X_{\mathrm{COBL},J}} = \begin{cases} lb_j + ub_j - rand \times X_{i,j}, if\ q < 0.5 \\ \frac{(lb_j+ub_j)}{2} + \frac{(lb_j+ub_j)}{2k} - \frac{X_{i,j}}{k}, otherwise \end{cases}, j = 1,2,\cdots,D \tag{33}$$

where $X_{i,j}$ and $\widetilde{X_{\mathrm{COBL},j}}$ are the j-dimensional components of X_i and $\widetilde{X_{\mathrm{COBL}}}$, respectively, lb_j and ub_j are the lower and upper boundaries in the j-th dimension.

2.4. Fitness-Distance Balance (FDB)

Selection methods in the meta-heuristic algorithms are used to identify the individual to be referenced from the whole population to guide future search directions and establish a balance between exploration and exploitation [92]. As a new selection method developed by Kahraman et al. [93] in 2020, the aim of FDB is to discover one or more candidate solutions that will make the most contribution to the algorithm's search process. Since it was first proposed, FDB has been widely applied to many algorithms to improve their exploration capability and overall search performance, such as Symbiotic Organism Search (SOS) [93], Stochastic Fractal Search (SFS) [94], and Coyote Optimization Algorithm (COA) [95]. What distinguishes FDB from other selection methods is that the selection process is executed in accordance with the score of the candidate solution, not just its fitness value. In the score calculation, two traits of candidate solutions, including the fitness function value and their distance from the best solution (X_{best}), are taken into account simultaneously. This guarantees that the candidate solution with the highest score value would be chosen to guide the population search in a more effective way. The implementation steps of the FDB selection method are as follows.

i) Suppose the dimension of the optimization problem is D, and N is the total number of candidate solutions in the population. The i-th candidate solution can be defined as $X_i = \left(x_{i,1}, x_{i,2}, \cdots, x_{i,D}\right), i = 1,2,\cdots,N$. Thus, the Euclidean distance between each solution and the best solution in the population X_{best} is calculated as shown in Eq (34).

$$_{i=1}^{N}\forall X_i, D_{X_i} = \sqrt{(x_{i,1} - x_{best,1})^2 + (x_{i,2} - x_{best,2})^2 + \cdots + (x_{i,D} - x_{best,D})^2} \tag{34}$$

ii) The distance vector D_X for each candidate solution can be expressed as in Eq (35).

$$D_X \equiv \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix}_{N\times 1} \tag{35}$$

iii) After normalization, the fitness and distance values of candidate solutions are used for calculating the score, shown as:

$$_{i=1}^{N}\forall X_i, S_{X_i} = \gamma \cdot normF_{X_i} + (1-\gamma) \cdot normD_{X_i} \tag{36}$$

where γ is a constant equal to 0.5, $normF_{X_i}$ denotes the normalized fitness values of the solution, and $normD_{X_i}$ denotes the normalized distance values.

iv) Finally, the score vector S_X, which stands for the FDB score values of the whole population, is given in Eq (37).

$$S_X \equiv \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix}_{N\times 1} \tag{37}$$

Once S_X is created, the algorithm could select more suitable candidate solutions to direct the search process based on their FDB scores.

3. The proposed IHAOAVOA algorithm

3.1. Detailed design of the proposed IHAOAVOA algorithm

In the exploration phase of the AO algorithm, the predatory behavior of Aquila to detect the potential fast-moving prey over a broad flight area is modeled (see Eqs (3) and (5)), which gives the algorithm robust global search capability and fast convergence rate [12]. Nonetheless, the selected search space cannot be searched thoroughly during the exploitation phase. As Figure 9 in the original paper [67] shows that the convergence curve remains unchanged in the later iterations, and the weak escape effects of the Lévy flight lead the algorithm to converge prematurely. In brief, AO has strong exploration capability, but its exploitation stage is still not sufficient. For the AVOA algorithm, the transition between exploration and exploitation depends on the hunger rate of vultures F. In the early exploration phase, the poor population diversity makes the algorithm exhibit a slow convergence rate. With the increase of iterations, the value of F gradually decreases and the algorithm proceeds to perform the exploitation phase. A total of four different hunting strategies (see Eqs (19), (21), (24), and (27)) are used to achieve various position updating of vultures, which allows the algorithm to effectively exploit the solution information in the search space to approach the global optimum. As a result, AVOA has promising exploitation capability.

In view of the above analysis, we hybridize the exploration phase of AO and the exploitation phase of AVOA to make full use of the advantages of the two basic algorithms. First, the AVOA algorithm is considered as the core framework, and we replace its original position updating rule in the exploration phase with Eqs (3) and (5) from AO, as follows:

If $rand \leq 0.5$:

$$X_i(t+1) = X_B(t) \times \left(1 - \frac{t}{T}\right) + X_m(t) - X_B(t) \times rand \tag{38}$$

If $rand > 0.5$:

$$X_i(t+1) = X_B(t) \times \text{Levy}(D) + X_r(t) + (y - x) \times rand \tag{39}$$

This hybrid operation preserves the algorithm's stronger global and local search capabilities, as well as faster convergence speed. Then, to further improve the overall search performance of the preliminary hybrid algorithm, we introduce the COBL and FDB strategies. As described in Section 2.3, COBL is beneficial to enrich the population diversity and escape from the local optima. Hence, the

COBL strategy is employed to find better candidate solutions before each iterative calculation. Meanwhile, it can be seen from Eq (39) in the hybrid algorithm that the next generation position of the i-th search agent primarily relies on the current best individual X_B and one individual X_r randomly selected from the whole population. Such reference individual obtained through the random selection method may not properly guide the algorithm to explore and exploit. To boost the search efficiency and maintain a better balance between the exploration and exploitation stages, we adopt the FDB selection strategy to identify one candidate X_{FDB} that will make the most contribution to the search process to replace X_r, as shown in Eq (40). All these strategies significantly enhance the convergence speed, solution quality, and robustness of the hybrid algorithm. Finally, this improved hybrid Aquila Optimizer and African Vultures Optimization Algorithm developed in this paper is named IHAOAVOA.

$$X_i(t+1) = X_B(t) \times \text{Levy}(D) + X_{FDB}(t) + (y - x) \times rand \tag{40}$$

Figure 5 depicts the flow chart of the proposed IHAOAVOA algorithm, and its pseudo-code is summarized in Algorithm 1.

3.2. Computational complexity of IHAOAVOA

The computational complexity of the proposed IHAOAVOA is associated with three components: initialization, fitness evaluation, and updating of positions. In the initialization phase, the positions of all search agents are generated randomly in the search space, which needs computational complexity $O(N)$, where N is the population size. Then in the iteration procedure, the algorithm evaluates the fitness value of each individual and updates the population positions sequentially, so the computational complexity is $O(2 \times T \times N + 2 \times T \times N \times D)$, where T denotes the maximum number of iterations and D denotes the dimension of specific problems. Thus, the total computational complexity of IHAOAVOA should be $O(N \times (1 + 2T + 2TD))$. As per the references [67,68], the computational complexity of both AO and AVOA is $O(N \times (1 + T + TD))$. Compared with the basic algorithms, the computational complexity of IHAOAVOA increases to some extent as a consequence of the introduced COBL and FDB strategies. However, these extra time costs can greatly improve the search performance of the algorithm, which is acceptable based on the NFL theorem [63].

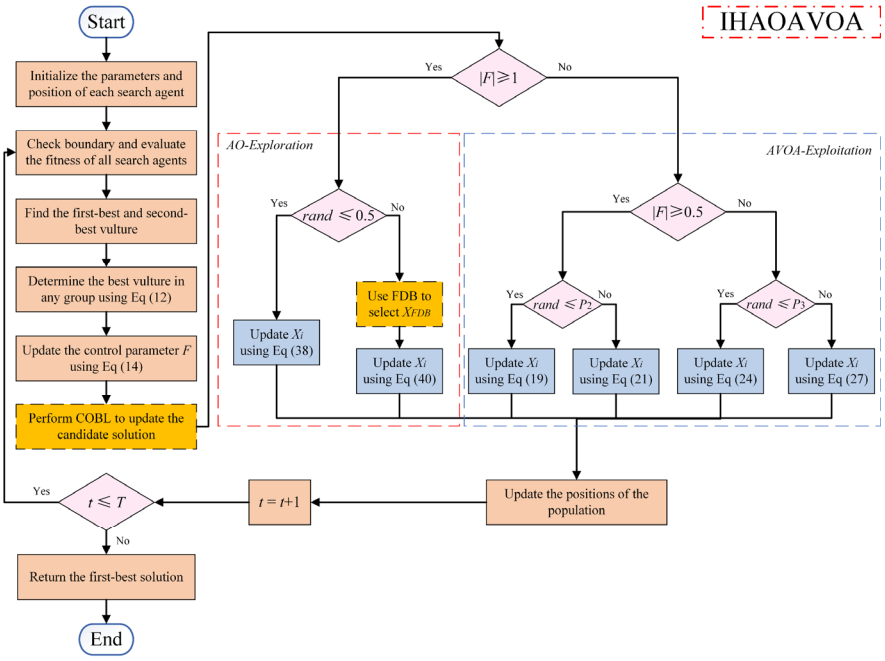


Figure 5. Flow chart of the proposed IHAOAVOA algorithm.

Algorithm 1 Pseudo-code of the proposed IHAOAVOA

Initialization

Initialize the population size N and the maximum iterations T

Initialize the positions of each search agent $X_i(i = 1,2,\cdots,N)$

Iteration

While $t \leq T$

  Check if the position goes beyond the search space boundary and then adjust it

  Evaluate the fitness values of all search agents

  Set $Bestvulture_1$ and $Bestvulture_2$ as the first-best solution and second-best solution respectively

  For each search agent X_i do

    Select the best vulture X_B according to Eq (12)

    Update the parameter F according to Eq (14)

    Perform COBL to generate the opposite solution $\widetilde{X_{\mathrm{COBL}}}$ of X_i using Eq (32) //COBL

    If the fitness of the opposite solution $f(\widetilde{X_{\mathrm{COBL}}})$ < the fitness of candidate solution $f(X_i)$ then

      $X_i = \widetilde{X_{\mathrm{COBL}}}, f(X_i) = f(\widetilde{X_{\mathrm{COBL}}})$

    End If

    If $|F| \geq 1$ then //AO-Exploration

      If $rand \leq 0.5$ then

        Update the position using Eq (38)

      Else

        Use FDB to select one candidate solution with the highest score X_{FDB} from the whole population //FDB

        Update the position using Eq (40)

      End If

    Else if $|F| < 1$ then //AVOA-Exploitation

Continued on next page

22. If $|F| \geq 0.5$ then
23. If $rand \leq P_2$ then
24. Update the position using Eq (19)
25. Else
26. Update the position using Eq (21)
27. End If
28. Else
29. If $rand \leq P_3$ then
30. Update the position using Eq (24)
31. Else
32. Update the position using Eq (27)
33. End If
34. End If
35. End If
36. End For
37. $t = t + 1$
38. End While

Output

39. Return the first-best solution $Bestvulture_1$

4. Experimental results and discussion

In this section, the effectiveness and feasibility of the proposed IHAOAVOA are thoroughly validated on two groups of optimization functions. The classical benchmark functions are first employed to estimate the performance of the algorithm in solving 23 simple numerical problems. Afterward, 10 IEEE CEC2019 benchmark functions are used to assess the algorithm with respect to addressing complex numerical problems. To illustrate the advantage of the proposed algorithm, IHAOAVOA is compared with the native AO [67], AVOA [68], and six other state-of-the-art algorithms, namely Sine Cosine Algorithm (SCA) [36], Whale Optimization Algorithm (WOA) [43], Grey Wolf Optimizer (GWO) [44], Moth-Flame Optimization algorithm (MFO) [51], Tunicate Swarm Algorithm (TSA) [53], and Arithmetic Optimization Algorithm (AOA) [38]. For consistency and fairness of the comparison, the maximum iteration and population size are set as 500 and 30, respectively. All the mentioned algorithms run independently 30 times to decrease random errors, and the average fitness (Avg) and standard deviation (Std) of experimental results are adopted as two evaluation metrics, where the average fitness represents the searchability of the algorithm, and the closer the average fitness is to the theoretical optimum value indicates the higher convergence accuracy of the algorithm, while the standard deviation characterizes the deviation degree of the experimental data, and the smaller the standard deviation indicates the better robustness of the algorithm. Moreover, the Wilcoxon rank-sum test [96], Friedman ranking test [97], and mean absolute error (MAE) test are used to determine whether there are significant differences between IHAOAVOA and other competitors in a statistical sense. Table 1 lists the important parameter values of each algorithm, which are set the same as those recommended in the original literature. The proposed IHAOAVOA succeeds the parameter settings for each stage of AO and AVOA algorithms, and the distance coefficient k for the COBL mechanism is fixed to 12,000 according to the literature [89] as well as extensive trials.

All the experimental series are implemented in MATLAB R2017a software (version 9.2.0) with Microsoft Windows 10 system, and the hardware platform of the computer is configured as Intel (R) Core (TM) i5-10300H CPU @ 2.50GHz and 16GB RAM.

Table 1. Parameter settings of different algorithms.

algorithm	parameter setting
AO [67]	$U = 0.00565; r = 10; \omega = 0.005; \alpha = 0.1; \delta = 0.1; G_1 \in [-1,1]; G_2 = [2,0]$
SCA [36]	$a = 2$
WOA [43]	$b = 1; a_1 = [2,0]; a_2 = [-2,-1]$
GWO [44]	$a = [2,0]$
MFO [51]	$b = 1; t = [-1,1]; a \in [-1,-2]$
TSA [53]	$P_{min} = 1; P_{max} = 4$
AOA [38]	$\alpha = 5; \mu = 0.499; Min = 0.2; Max = 0.9$
AVOA [68]	$L_1 = 0.8; L_2 = 0.2; w = 2.5; P_1 = 0.6; P_2 = 0.4; P_3 = 0.6$
IHAOAVOA	$L_1 = 0.8; L_2 = 0.2; w = 2.5; P_2 = 0.4; P_3 = 0.6; U = 0.00565; r = 10; \omega = 0.005; k = 12{,}000$

4.1. Experiment 1: classical benchmark functions

In this subsection, a set of 23 classical benchmark functions selected from the reference [68] are utilized to evaluate the performance of the proposed IHAOAVOA. The 23 benchmark functions can be classified into three different categories on the basis of their properties: unimodal, multimodal, and fix-dimension multimodal. The unimodal benchmark functions (F_1–F_7) have only one global optimal value and are usually applied to check the algorithm's exploitation competence. By contrast, the multimodal benchmark functions (F_8–F_{13}) are characterized by multiple local minima. This kind of function is designed to examine the exploration capability and the local optima avoidance of the algorithm. It is worth mentioning here that the dimensions of the unimodal and multimodal benchmark functions (F_1–F_{13}) can be set as required, so they can optionally be used to see the performance of the proposed algorithm on high-dimensional problems. The fix-dimension multimodal benchmark functions (F_{14}–F_{23}) can be regarded as a combination of the first two categories of functions but with a lower dimension. They are used to study the stability of the algorithm in the transition between exploration and exploitation. The formula, dimension size (D), variable range, and theoretical minimum ($F_{\min}$) of each function are outlined in Tables 2–4. Figure 6 intuitively shows the search space of some representative benchmark functions.

In the experiments of classical benchmark functions, the impacts of two introduced strategies are first examined. Then, IHAOAVOA, AO, AVOA, and six state-of-the-art meta-heuristic algorithms are tested on these 23 functions concurrently. Several aspects of the obtained results are analyzed, including exploitation capability, exploration capability, boxplot, convergence curve, average computational time, and statistical differences. In addition, the scalability of IHAOAVOA for large-scale optimization is also investigated.

Table 2. Unimodal benchmark functions.

function	D	range	$F_{\min}$
$F_1(x) = \sum_{i=1}^{D} x_i^2$	30	[-100, 100]	0
$F_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	30	[-10, 10]	0
$F_3(x) = \sum_{i=1}^{D} (\sum_{j=1}^{D} x_j)^2$	30	[-100, 100]	0
$F_4(x) = max_i\{\lvert x_i \rvert, 1 \le i \le D\}$	30	[-100, 100]	0
$F_5(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30, 30]	0
$F_6(x) = \sum_{i=1}^{D} (\lvert x_i + 0.5 \rvert)^2$	30	[-100, 100]	0
$F_7(x) = \sum_{i=1}^{D} i x_i^4 + random[0,1)$	30	[-1.28, 1.28]	0

Table 3. Multimodal benchmark functions.

function	D	range	$F_{\min}$
$F_8(x) = \sum_{i=1}^{D} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	[-500, 500]	-418.9829 × *Dim*
$F_9(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12, 5.12]	0
$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{D} \cos(2\pi x_i)) + 20 + e$	30	[-32, 32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[-600, 600]	0
$F_{12}(x) = \frac{\pi}{D}\{10\sin(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\} + \sum_{i=1}^{D} u(x_i, 10,100,4)$ $y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a < x_i < a \\ k(-x_i - a)^m, x_i < -a \end{cases}$	30	[-50, 50]	0
$F_{13}(x) = 0.1\{\sin^2(3\pi x_i) + \sum_{i=1}^{D}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_D - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{D} u(x_i, 5,100,4)$	30	[-50, 50]	0

Table 4. Fix-dimension multimodal benchmark functions.

function	D	range	F_{min}
$F_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25}(j + \sum_{i=1}^{2}(x_i - a_{ij})^6)^{-1})^{-1}$	2	[-65, 65]	0.998
$F_{15}(x) = \sum_{i=1}^{11}[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}]^2$	4	[-5, 5]	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	[-5, 5]	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_2 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2, 2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[-1, 2]	-3.8628
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0, 1]	-3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.1532
$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.5363

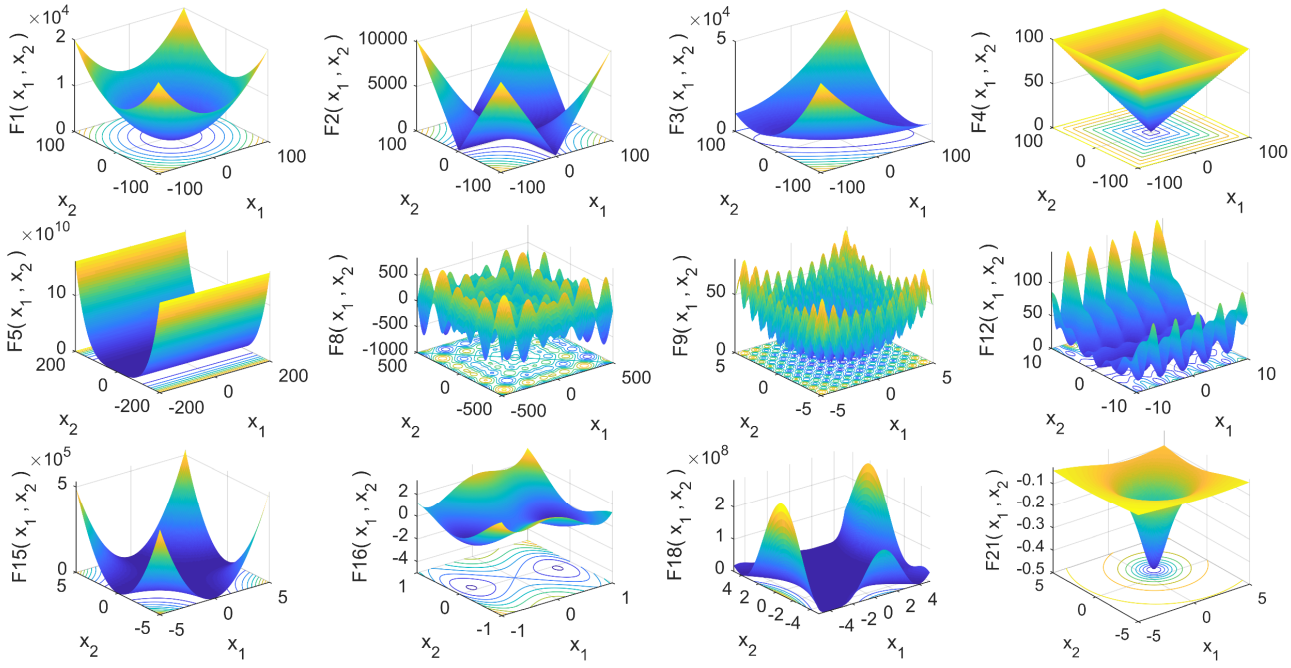


Figure 6. 3D view of some typical benchmark functions.

4.1.1. Impacts of components

To overcome the defects in the single algorithm, this paper proposes a novel improved hybrid optimizer. First, the exploration phase of AO is hybridized with the exploitation phase of AVOA to achieve better convergence performance. Then, we introduce the COBL mechanism into the preliminary hybrid algorithm to help search agents escape from the local optima. Besides, to maintain a good balance between exploration and exploitation, the FDB method is adopted to select one more suitable reference individual for the population search. Hence, the proposed IHAOAVOA can be regarded as a hybrid of AO and AVOA integrated with COBL and FDB strategies. To evaluate the effectiveness of each component, three IHAOAVOA-derived variants are designed individually for comparison study in this subsection, which are listed below:

- IHAOAVOA-1 (Hybrid of AO and AVOA only);
- IHAOAVOA-2 (Hybrid of AO and AVOA integrated with COBL);
- IHAOAVOA-3 (Hybrid of AO and AVOA integrated with FDB).

Under the same experimental setting, IHAOAVOA-1, IHAOAVOA-2, IHAOAVOA-3, and IHAOAVOA are tested on 23 different types of benchmark functions in Tables 2–4 concurrently. The obtained average fitness (Avg) and standard deviation (Std) results are listed in

Table **5**. Based on the results, we can find that IHAOAVOA-2, IHAOAVOA-3, and IHAOAVOA always obtain better convergence accuracy and standard deviation values than IHAOAVOA-1 on test functions F_2–F_8, F_{12}–F_{15}, and F_{20}. For F_{16}–F_{19} and F_{21}–F_{23}, four algorithms could obtain the same optimal fitness, but IHAOAVOA-2, IHAOAVOA-3, and IHAOAVOA still slightly outperform IHAOAVOA-1 regarding the standard deviation. These demonstrate that the introduced COBL and FDB strategies are indeed effective in improving the search breadth and robustness of the hybrid algorithm to some extent; in particular, the role of COBL is more important and irreplaceable. Compared to IHAOAVOA-2 and IHAOAVOA-3, which have one single strategy, it is clear that IHAOAVOA wins on F_5–F_8 and F_{12}–F_{15}. In addition, IHAOAVOA shows a higher level of stability in solving almost all test issues. Thus, we can conclude that the reasonable combination of COBL and FDB has a significant synergistic effect on boosting the comprehensive performance of IHAOAVOA, enabling it to provide very excellent solutions. After validation, IHAOAVOA is selected as the final version for further comparison and discussion.

Table 5. Comparison results of IHAOAVOA-1, IHAOAVOA-2, IHAOAVOA-3, and IHAOAVOA on 23 benchmark functions.

F_n	IHAOAVOA-1		IHAOAVOA-2		IHAOAVOA-3		IHAOAVOA	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
F_1	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F_2	1.51E-157	8.24E-157	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F_3	2.28E-260	4.31E-260	**0.00E+00**	**0.00E+00**	2.90E-272	**0.00E+00**	**0.00E+00**	**0.00E+00**
F_4	1.84E-163	7.64E-163	**0.00E+00**	**0.00E+00**	1.06E-172	**0.00E+00**	**0.00E+00**	**0.00E+00**
F_5	3.23E-05	4.31E-05	2.41E-06	1.17E-05	2.91E-05	3.61E-05	**8.30E-07**	**2.43E-06**
F_6	8.27E-08	7.19E-08	3.04E-08	3.42E-08	5.08E-08	4.52E-08	**2.00E-08**	**2.52E-08**
F_7	1.20E-04	1.08E-04	4.12E-05	4.16E-05	9.66E-05	9.39E-05	**3.41E-05**	**3.28E-05**
F_8	-11372.005	1.85E+03	-12045.671	1.33E+03	-11455.163	1.82E+03	**-12115.001**	**1.07E+03**

Continued on next page

F_n	IHAOAVOA-1		IHAOAVOA-2		IHAOAVOA-3		IHAOAVOA	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
F_9	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F_{10}	**8.88E-16**	**0.00E+00**	**8.88E-16**	**0.00E+00**	**8.88E-16**	**0.00E+00**	**8.88E-16**	**0.00E+00**
F_{11}	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F_{12}	8.17E-09	7.84E-09	3.95E-09	5.99E-09	5.27E-09	5.70E-09	**3.83E-09**	**4.34E-09**
F_{13}	4.10E-08	6.52E-08	5.27E-09	6.91E-09	2.31E-08	2.09E-08	**3.36E-09**	**4.30E-09**
F_{14}	3.05E+00	3.76E+00	1.49E+00	8.54E-01	2.79E+00	3.47E+00	**1.23E+00**	**6.21E-01**
F_{15}	3.17E-04	1.54E-05	3.12E-04	6.51E-06	3.15E-04	1.41E-05	**3.11E-04**	**5.06E-06**
F_{16}	**-1.0316**	4.77E-16	**-1.0316**	4.32E-16	**-1.0316**	4.59E-16	**-1.0316**	**4.36E-16**
F_{17}	**3.98E-01**	3.24E-16	**3.98E-01**	**0.00E+00**	**3.98E-01**	**0.00E+00**	**3.98E-01**	**0.00E+00**
F_{18}	**3.00E+00**	7.45E-07	**3.00E+00**	2.44E-09	**3.00E+00**	2.86E-08	**3.00E+00**	**3.63E-10**
F_{19}	**-3.8628**	4.34E-12	**-3.8628**	2.01E-11	**-3.8628**	2.97E-12	**-3.8628**	**1.32E-12**
F_{20}	-3.2669	6.56E-02	-3.2784	6.62E-02	-3.2850	5.77E-02	**-3.2903**	**5.35E-02**
F_{21}	**-10.1532**	8.49E-14	**-10.1532**	4.20E-13	**-10.1532**	1.99E-13	**-10.1532**	**6.68E-14**
F_{22}	**-10.4029**	3.32E-13	**-10.4029**	2.93E-13	**-10.4029**	3.18E-13	**-10.4029**	**1.89E-13**
F_{23}	**-10.5364**	6.78E-13	**-10.5364**	7.57E-13	**-10.5364**	4.53E-13	**-10.5364**	**8.64E-14**

Note: The best results obtained have been marked in bold.

4.1.2. Evaluation of exploitation and exploration capabilities

According to the previously described unimodal, multimodal, and fix-dimension multimodal benchmark functions, in this part, we give a complete assessment of the exploitation and exploration capabilities of the proposed algorithm. Table 6 lists the average fitness and standard deviation results obtained by IHAOAVOA and other algorithms for each function F_1–F_{23} in the dimension $D = 30$. As can be seen from this table, the proposed IHAOAVOA outperforms its peers on the majority of the test problems.

Specifically, for the unimodal functions (F_1–F_7), IHAOAVOA can effectively search for the global optimum (0) on F_1–F_4, and the solution accuracy of the proposed improved hybrid algorithm is greatly increased compared to the basic AO and AVOA. On functions F_5–F_7, though IHAOAVOA does not obtain the theoretical optimal values, its solution accuracy is still marginally higher than that of AO and AVOA by several orders of magnitude, ranking first in all comparison algorithms. As far as the standard deviation is concerned, IHAOAVOA also provides the best performance on these problems. The goal of unimodal functions is to evaluate the exploitation capability. From the above results, we can confirm that IHAOAVOA has competitive local exploitation potential.

For the multimodal functions (F_8–F_{13}), the average fitness and standard deviation of IHAOAVOA on F_8, F_{12}, and F_{13} are completely superior to other competitor algorithms. On functions F_9 and F_{10}, the proposed algorithm obtains the same performance as AO, AOA, and AVOA, but much better than SCA, WOA, GWO, MFO, and TSA. On function F_{11}, AO, AVOA, and IHAOAVOA show no difference, and all provide the most satisfactory results. The purpose of multimodal functions is to measure the exploration ability. Therefore, these results prove that IHAOAVOA possesses excellent global exploration capability. This is mainly attributed to the fact that the designed COBL strategy can efficiently expand the unknown search region and help the algorithm bypass the local optima to find higher-quality solutions.

When solving fix-dimension multimodal functions (F_{14}–F_{23}), IHAOAVOA could comfortably outperform others in terms of the average fitness and standard deviation on F_{14}, F_{15}, F_{20}, and F_{23}. For the remaining functions, whereas some comparison algorithms can achieve the same best average fitness values as IHAOAVOA, the standard deviation of the proposed algorithm is the smallest among them. This reveals the superior robustness of IHAOAVOA. In light of the properties of the fix-dimension multimodal functions, these results indicate that IHAOAVOA is capable of better balancing the exploration and exploitation, which benefits from the FDB selection method.

4.1.3. Boxplots analysis

Since the boxplot can visualize the data distribution, it is a well-suited diagram for describing the agreement between the data. Based on the results obtained through 30 independent runs in Table 6, to better understand the algorithm's distribution characteristics, the boxplots of IHAOAVOA and other algorithms on 12 representative benchmark functions are depicted in Figure 7. In this figure, the center marker of each box denotes the median value, the bottom and top fringes of the box respectively represent the first and third quartiles, and the notation "+" represents the outliers. From Figure 7, it can be seen that the proposed IHAOAVOA shows great consistency and produces no outliers during the optimization process for almost all test cases. At the same time, the median, maximum, and minimum values achieved by IHAOAVOA are more concentrated compared with competitor algorithms. On function F_8, despite individual outliers, the overall distribution of IHAOAVOA remains superior to that of others. The above demonstrates that the IHAOAVOA proposed in this paper has strong stability.

Table 6. Comparison results of different algorithms on 23 benchmark functions.

F_n	criteria	AO	SCA	WOA	GWO	MFO	TSA	AOA	AVOA	IHAOAVOA
F_1	Avg	1.65E-101	8.60E+00	2.21E-71	6.79E-28	3.00E+03	9.17E-21	3.44E-26	9.28E-301	**0.00E+00**
	Std	9.04E-101	1.18E+01	1.20E-70	9.42E-28	5.35E+03	4.50E-20	1.88E-25	**0.00E+00**	**0.00E+00**
F_2	Avg	2.30E-55	3.44E-02	1.12E-49	1.13E-16	3.38E+01	7.45E-14	**0.00E+00**	1.34E-149	**0.00E+00**
	Std	1.26E-54	4.97E-02	6.06E-49	9.49E-17	1.86E+01	6.22E-14	**0.00E+00**	7.32E-149	**0.00E+00**
F_3	Avg	9.21E-106	9.04E+03	4.55E+04	8.29E-05	2.01E+04	3.38E-04	2.81E-03	9.87E-208	**0.00E+00**
	Std	4.03E-105	5.83E+03	1.20E+04	4.16E-04	1.01E+04	6.64E-04	7.69E-03	**0.00E+00**	**0.00E+00**
F_4	Avg	1.60E-51	3.35E+01	5.03E+01	1.03E-06	6.70E+01	3.73E-01	2.57E-02	1.53E-146	**0.00E+00**
	Std	8.76E-51	1.11E+01	2.63E+01	2.00E-06	9.60E+00	3.50E-01	2.07E-02	8.23E-146	**0.00E+00**
F_5	Avg	5.17E-03	4.92E+04	2.79E+01	2.70E+01	5.36E+06	3.12E+01	2.85E+01	4.82E-05	**5.83E-07**
	Std	1.82E-02	1.02E+05	4.10E-01	6.82E-01	2.03E+07	1.54E+01	2.80E-01	4.40E-05	**9.72E-07**

Continued on next page

F_n	criteria	AO	SCA	WOA	GWO	MFO	TSA	AOA	AVOA	IHAOAVOA
F_6	Avg	9.93E-05	2.07E+01	4.33E-01	7.79E-01	2.01E+03	3.72E+00	3.20E+00	5.27E-07	**2.17E-08**
	Std	1.60E-04	3.48E+01	2.02E-01	3.94E-01	4.08E+03	5.83E-01	3.19E-01	5.06E-07	**4.06E-08**
F_7	Avg	1.31E-04	1.17E-01	3.65E-03	1.93E-03	2.99E+00	9.99E-03	6.07E-05	1.29E-04	**3.22E-05**
	Std	1.28E-04	1.14E-01	4.94E-03	8.63E-04	4.42E+00	5.40E-03	6.64E-05	8.95E-05	**2.53E-05**
F_8	Avg	-7666.078	-3710.356	-9574.657	-6086.846	-8436.128	-6084.151	-5267.714	-12365.251	**-12514.211**
	Std	3.57E+03	3.60E+02	1.59E+03	7.35E+02	7.67E+02	3.51E+02	3.89E+02	4.16E+02	**3.03E+02**
F_9	Avg	**0.00E+00**	3.37E+01	5.67E+00	2.57E+00	1.59E+02	1.88E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	2.62E+01	3.11E+01	3.49E+00	3.75E+01	3.95E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**
F_{10}	Avg	**8.88E-16**	1.49E+01	4.80E-15	1.02E-13	1.29E+01	1.59E+00	**8.88E-16**	**8.88E-16**	**8.88E-16**
	Std	**0.00E+00**	8.32E+00	2.35E-15	1.86E-14	8.44E+00	1.53E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
F_{11}	Avg	**0.00E+00**	9.45E-01	1.64E-02	4.96E-03	2.20E+01	1.07E-02	1.50E-01	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	3.33E-01	5.30E-02	8.06E-03	3.87E+01	1.55E-02	1.24E-01	**0.00E+00**	**0.00E+00**
F_{12}	Avg	3.45E-06	8.42E+04	2.87E-02	4.32E-02	1.27E+04	7.94E+00	5.15E-01	2.41E-08	**3.35E-09**
	Std	5.68E-06	2.82E+05	4.43E-02	2.43E-02	5.07E+04	3.61E+00	4.35E-02	1.32E-08	**2.99E-09**
F_{13}	Avg	2.03E-05	2.23E+05	5.36E-01	6.71E-01	1.37E+07	3.08E+00	2.81E+00	4.15E-08	**8.32E-09**
	Std	3.39E-05	7.25E+05	2.35E-01	2.14E-01	7.49E+07	7.42E-01	9.95E-02	3.59E-08	**2.30E-08**
F_{14}	Avg	2.92E+00	1.66E+00	2.96E+00	5.14E+00	2.68E+00	8.88E+00	9.42E+00	1.36E+00	**1.26E+00**
	Std	3.76E+00	9.51E-01	3.23E+00	4.41E+00	2.01E+00	5.51E+00	4.23E+00	1.79E+00	**6.86E-01**
F_{15}	Avg	4.71E-04	1.03E-03	5.65E-04	5.05E-03	1.23E-03	3.93E-03	1.18E-02	4.08E-04	**3.25E-04**
	Std	1.21E-04	3.92E-04	2.15E-04	8.60E-03	1.39E-03	7.54E-03	1.41E-02	1.99E-04	**6.11E-05**
F_{16}	Avg	-1.0313	**-1.0316**	**-1.0316**	**-1.0316**	**-1.0316**	-1.0253	**-1.0316**	**-1.0316**	**-1.0316**
	Std	3.78E-04	4.67E-05	4.89E-09	2.53E-08	6.78E-16	1.29E-02	1.18E-07	4.46E-16	**4.34E-16**
F_{17}	Avg	**3.98E-01**	4.00E-01	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**	4.10E-01	**3.98E-01**	**3.98E-01**
	Std	3.19E-04	1.87E-03	8.49E-05	2.96E-06	**0.00E+00**	5.51E-05	1.06E-02	5.42E-16	**0.00E+00**
F_{18}	Avg	3.04E+00	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	6.60E+00	1.24E+01	**3.00E+00**	**3.00E+00**
	Std	3.90E-02	7.14E-05	9.90E-05	3.22E-05	**1.55E-15**	9.34E+00	1.95E+01	6.67E-06	2.81E-08
F_{19}	Avg	-3.8546	-3.8552	-3.8491	-3.8617	**-3.8628**	-3.8620	-3.8527	**-3.8628**	**-3.8628**

Continued on next page

F_n	Criteria	AO	SCA	WOA	GWO	MFO	TSA	AOA	AVOA	IHAOAVOA
	Std	6.88E-03	3.23E-03	4.07E-02	2.26E-03	1.36E-11	1.96E-03	3.00E-03	7.01E-11	**2.71E-15**
F_{20}	Avg	-3.1375	-2.8605	-3.2402	-3.2656	-3.2266	-3.2513	-3.0903	-3.2704	**-3.2824**
	Std	1.00E-01	3.92E-01	9.56E-02	7.69E-02	6.40E-02	6.86E-02	7.36E-02	6.00E-02	**5.70E-02**
F_{21}	Avg	-10.1434	-2.1289	-8.2722	-9.5607	-5.8923	-6.3812	-3.8602	**-10.1532**	**-10.1532**
	Std	1.33E-02	1.77E+00	2.49E+00	1.84E+00	3.42E+00	2.94E+00	1.35E+00	7.59E-13	**6.03E-13**
F_{22}	Avg	-10.3894	-3.1252	-8.1968	-10.4010	-6.9732	-6.9165	-3.5839	**-10.4029**	**-10.4029**
	Std	1.56E-02	1.80E+00	3.16E+00	1.47E-03	3.58E+00	3.45E+00	1.08E+00	6.78E-13	**1.00E-13**
F_{23}	Avg	-10.5293	-3.8926	-7.2253	-10.5345	-7.3135	-6.9438	-4.1219	-10.5360	**-10.5364**
	Std	8.12E-03	1.69E+00	3.44E+00	9.63E-04	3.58E+00	3.71E+00	1.79E+00	4.42E-08	**3.35E-13**

Note: The best results obtained have been marked in bold.

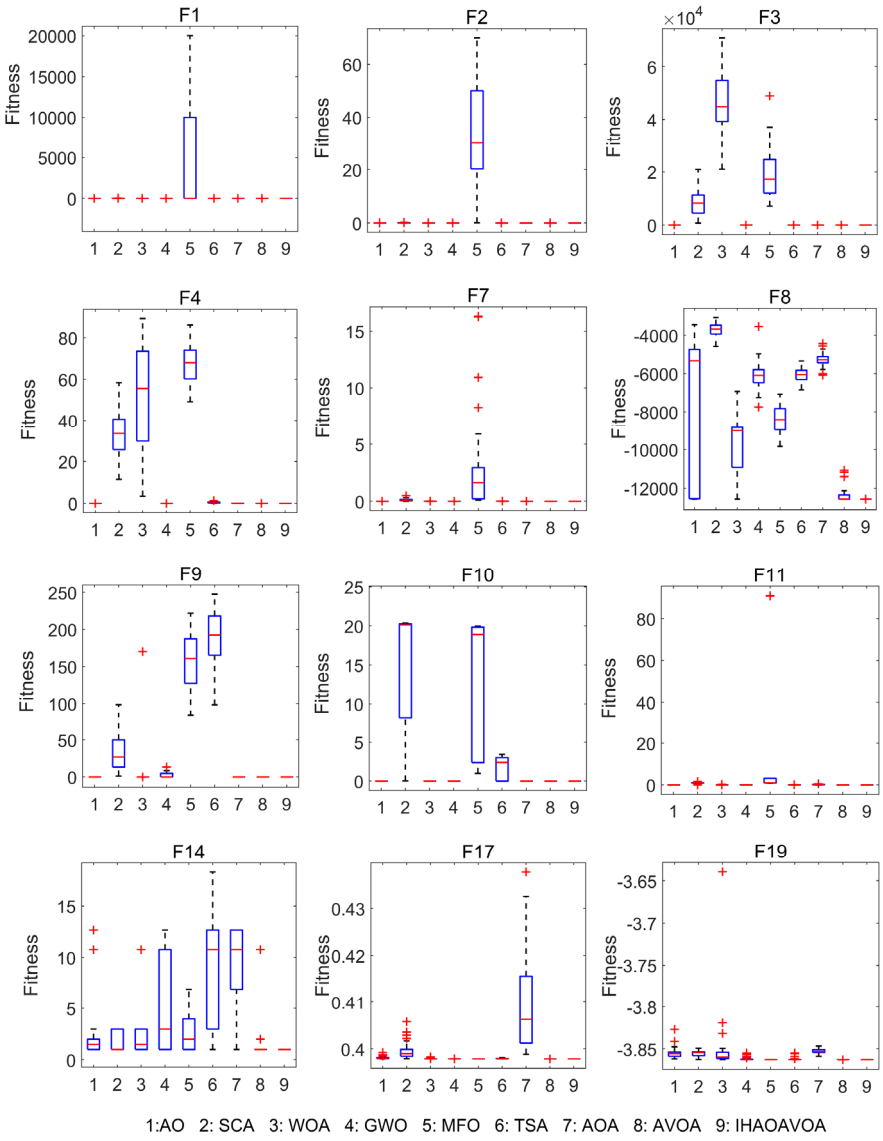


Figure 7. Boxplots of different algorithms on some benchmark functions.

4.1.4. Convergence behavior analysis

Normally, search agents tend to change dramatically in the early iterations to explore the promising area of the search space as much as possible, then exploit it at length and converge gradually with the number of iterations. To analyze the convergence behavior of the algorithm in the search for the optimal solution, Figure 8 plots the convergence curves of AO, SCA, WOA, GWO, MFO, TSA, AOA, AVOA, and IHAOAVOA on 23 benchmark functions throughout the iterations.

As we can observe from this figure, the proposed IHAOAVOA has superior and competitive convergence performance compared with other state-of-the-art algorithms. For unimodal benchmark functions (F_1–F_7), the proposed IHAOAVOA can rapidly converge to the global optimum in the initial phase of functions F_1–F_4, and its convergence curve displays the fastest decay rate; however, other algorithms suffer from significant lag and are slow to search. This phenomenon is because the designed COBL mechanism can provide better randomness and population diversity at the initial stage, thus

deeply extending the search range of IHAOAVOA. On functions F_5 and F_6, IHAOAVOA presents a similar but better convergence trend than AO at the beginning of the iteration, and later it progressively follows the same trend as AVOA. Eventually, IHAOAVOA obtains the highest convergence accuracy among these algorithms with a considerable improvement over the basic AO and AVOA. These behaviors exactly validate the general framework of the proposed algorithm. The combination of the exploration phase of AO and the exploitation phase of AVOA contributes to effectively enhancing the search performance and accelerating the convergence. On function F_7, IHAOAVOA also obtains the best convergence accuracy with the least number of iterations compared with its peers.

Since the multimodal benchmark functions (F_8–F_{13}) consist of several local optima, it becomes more challenging to solve them. Nevertheless, IHAOAVOA still maintains excellent convergence behavior in these test cases. In particular, on functions F_9 and F_{11}, IHAOAVOA can achieve the global optimum within ten iterations. On functions F_8 and F_{10}, although the theoretical optimal value is not obtained, the convergence speed and final solution accuracy of the proposed method again rank first among all algorithms. On function F_{12}, IHAOAVOA lags behind AO at the beginning of the search process. Yet, during the later iterations, AO falls into the local optima, but the proposed method commences to show its advantages and accelerate convergence to yield higher-quality results. Furthermore, the superior local optima avoidance capability of IHAOAVOA is well demonstrated on F_{13}. These convergence behaviors of IHAOAVOA on multimodal functions present strong evidence that the hybrid operation and COBL mechanism are beneficial to help get rid of the local optima. For fix-dimension multimodal benchmark functions (F_{14}–F_{23}), it can be noted that IHAOAVOA quickly shifts from exploration to exploitation phases, converges towards the global optimum in the early stages of the iterations, and gradually determines the optimal value. Compared with the AO, AVOA, and other competitor algorithms, the calculation accuracy and operating efficiency of the proposed algorithm on these functions are also improved to some extent, which mainly owes to the role of the FDB selection method in guiding the search process.

In short, the proposed IHAOAVOA can provide a better convergence pattern no matter for unimodal or multimodal functions.

4.1.5. Computational time analysis

To investigate the computational cost of the proposed IHAOAVOA, Table 7 reports the average computational time obtained by each algorithm on 23 benchmark functions. For a more intuitive overview of the results, the total runtime of the nine methods has been calculated and sorted as follows: IHAOAVOA > AO > AVOA > GWO > MFO > TSA > SCA > AOA > WOA. It can be noticed that IHAOAVOA consumes more computational time than AO and AVOA, which ranks last among all algorithms. One of the main reasons for this is the high time consumption of AO and AVOA themselves. Furthermore, IHAOAVOA employs COBL to generate the opposite candidate solution to boost the algorithm's local optima avoidance capability and extend the unknown search space, and the FDB selection method is used to better guide the search procedure. These introduced strategies also increase the steps of the hybrid algorithm and extra computational time cost. However, on the whole, considering the NFL theorem and the substantial time consumption of function evaluation in resolving real-life optimization tasks, it is acceptable to sacrifice some runtime to achieve more reliable and accurate solutions.

Table 7. Average computational time of different algorithms on 23 benchmark functions (unit: s).

F_n	AO	SCA	WOA	GWO	MFO	TSA	AOA	AVOA	IHAOAVOA
F_1	2.42E-01	1.08E-01	**7.88E-02**	1.29E-01	1.08E-01	1.15E-01	9.67E-02	2.06E-01	3.23E-01
F_2	2.64E-01	1.13E-01	**8.63E-02**	1.40E-01	1.17E-01	1.24E-01	1.01E-01	1.96E-01	3.29E-01
F_3	9.72E-01	4.66E-01	**4.32E-01**	5.00E-01	4.73E-01	4.80E-01	4.52E-01	5.49E-01	1.40E+00
F_4	2.36E-01	1.08E-01	**7.49E-02**	1.26E-01	1.08E-01	1.10E-01	9.17E-02	1.84E-01	2.92E-01
F_5	2.80E-01	1.24E-01	**9.37E-02**	1.47E-01	1.31E-01	1.35E-01	1.13E-01	2.12E-01	3.60E-01
F_6	2.31E-01	1.06E-01	**7.40E-02**	1.23E-01	1.02E-01	1.07E-01	8.54E-02	1.75E-01	2.86E-01
F_7	3.55E-01	1.63E-01	**1.30E-01**	1.83E-01	1.65E-01	1.72E-01	1.51E-01	2.44E-01	4.68E-01
F_8	2.85E-01	1.30E-01	**9.53E-02**	1.45E-01	1.28E-01	1.38E-01	1.13E-01	2.04E-01	3.62E-01
F_9	2.42E-01	1.13E-01	**7.94E-02**	1.28E-01	1.12E-01	1.18E-01	9.58E-02	1.80E-01	2.98E-01
F_{10}	2.70E-01	1.29E-01	**9.02E-02**	1.37E-01	1.26E-01	1.27E-01	1.02E-01	1.95E-01	3.39E-01
F_{11}	2.93E-01	1.39E-01	**1.06E-01**	1.48E-01	1.38E-01	1.34E-01	1.20E-01	2.13E-01	3.69E-01
F_{12}	6.28E-01	3.01E-01	**2.70E-01**	3.15E-01	3.03E-01	3.01E-01	2.88E-01	3.71E-01	8.72E-01
F_{13}	6.44E-01	3.09E-01	**2.72E-01**	3.22E-01	3.03E-01	3.11E-01	2.81E-01	3.79E-01	8.91E-01
F_{14}	1.34E+00	6.22E-01	6.14E-01	**6.10E-01**	6.28E-01	6.16E-01	6.25E-01	7.15E-01	1.95E+00
F_{15}	1.96E-01	6.40E-02	**6.27E-02**	6.66E-02	6.87E-02	6.61E-02	6.78E-02	1.51E-01	2.64E-01
F_{16}	1.61E-01	4.82E-02	**4.74E-02**	4.94E-02	5.52E-02	4.81E-02	5.08E-02	1.36E-01	2.13E-01
F_{17}	1.62E-01	4.41E-02	**4.25E-02**	4.47E-02	4.81E-02	4.38E-02	4.52E-02	1.30E-01	1.97E-01
F_{18}	1.59E-01	4.30E-02	**4.23E-02**	4.60E-02	5.05E-02	4.45E-02	4.32E-02	1.33E-01	1.99E-01
F_{19}	2.64E-01	9.55E-02	**9.03E-02**	9.51E-02	9.92E-02	8.95E-02	9.55E-02	1.79E-01	3.48E-01
F_{20}	2.62E-01	9.71E-02	**9.21E-02**	1.03E-01	1.05E-01	9.96E-02	9.42E-02	1.84E-01	3.56E-01
F_{21}	3.85E-01	1.55E-01	**1.54E-01**	1.55E-01	1.59E-01	1.57E-01	1.58E-01	2.43E-01	5.33E-01
F_{22}	4.68E-01	1.99E-01	**1.91E-01**	2.00E-01	2.04E-01	1.96E-01	1.92E-01	2.89E-01	6.52E-01
F_{23}	5.85E-01	2.55E-01	**2.48E-01**	2.51E-01	2.60E-01	2.52E-01	2.54E-01	3.51E-01	8.41E-01

Note: The best results obtained have been marked in bold.

4.1.6. Statistical test

Because the results attained by each algorithm are random, it is usually not sufficient to evaluate the relevant performance based only on the average fitness and standard deviation values. To statistically validate whether there is a significant difference between the proposed IHAOAVOA and the comparison algorithm, the Wilcoxon rank-sum test [96], Friedman ranking test [97], and mean absolute error (MAE) test are conducted in this subsection.

For Wilcoxon rank-sum test, a non-parametric statistical method, the significance level is set as 5%. Specifically, if the p-value is less than 0.05, it means that IHAOAVOA performs better than the comparison algorithm; otherwise, IHAOAVOA performs worse than the comparison algorithm. Additionally, NaN indicates that IHAOAVOA performs consistently with the comparison algorithm. The obtained p-values of the Wilcoxon rank-sum test on each benchmark function are recorded in Table 9. For convenience, in the last line of this table, we use the letter symbols (W/T/L) to denote the number of winner times, the number of tie times, and the number of loss times for IHAOAVOA, respectively. As shown in Table 9, IHAOAVOA is able to outperform AO on 20 functions, SCA on 23 functions, WOA on 23 functions, GWO on 23 functions, MFO on 22

functions, TSA on 23 functions, AOA on 20 functions, and AVOA on 18 functions, which proves the significant superiority of the proposed work.

To reveal the overall performance ranking of each algorithm on 23 benchmark functions, another non-parametric comparison method: the Friedman ranking test, is used to assess the average fitness data "Avg" obtained in Table 6. As presented in Figure 9, the proposed IHAOAVOA achieves the best Friedman mean ranking value of 1.6522 among these algorithms. Thus, based on the theories of statistical analysis, we can consider that IHAOAVOA has a noticeable improvement over the basic AO and AVOA, and it can provide the best performance in all comparative algorithms.

At last, each algorithm's mean absolute error (MAE) on these classical test functions is also evaluated and ranked. The statistical MAE is a measure to reveal the gap between estimates and the theoretical values, which is formulated as follows:

$$\mathrm{MAE} = \frac{1}{NF}\sum_{i=1}^{NF}|f_i - f_i^*| \tag{41}$$

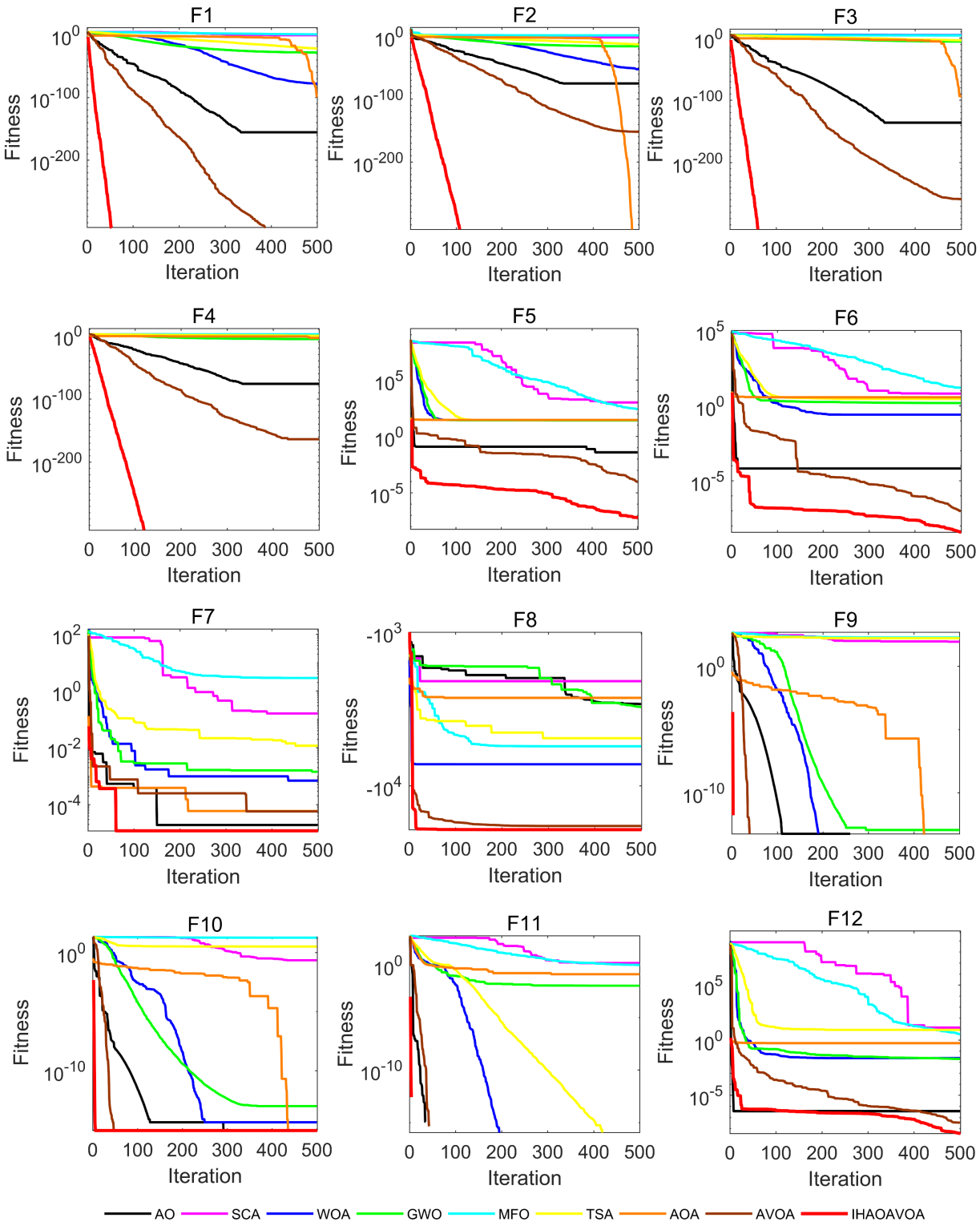
where NF is the number of test functions, f_i denotes the optimization result of the i-th function obtained by the algorithm, and f_i^* denotes the global optimum of the i-th function.

Table 8. Mean absolute error of different algorithms on 23 benchmark functions.

algorithms	MAE	rank
AO	2.13E+02	3
SCA	1.63E+04	8
WOA	2.11E+03	7
GWO	2.83E+02	4
MFO	8.31E+05	9
TSA	2.93E+02	5
AOA	3.21E+02	6
AVOA	8.90E+00	2
IHAOAVOA	**2.42E+00**	**1**

Note: The best results obtained have been marked in bold.

Table 8 records the MAE and ranking of all algorithms. From this table, IHAOAVOA has the smallest MAE value with a reduction of 98.87 and 72.84% compared to AO and AVOA respectively, and it ranks first among all algorithms. These results once again prove the superiority of the proposed method statistically.



Continued on next page

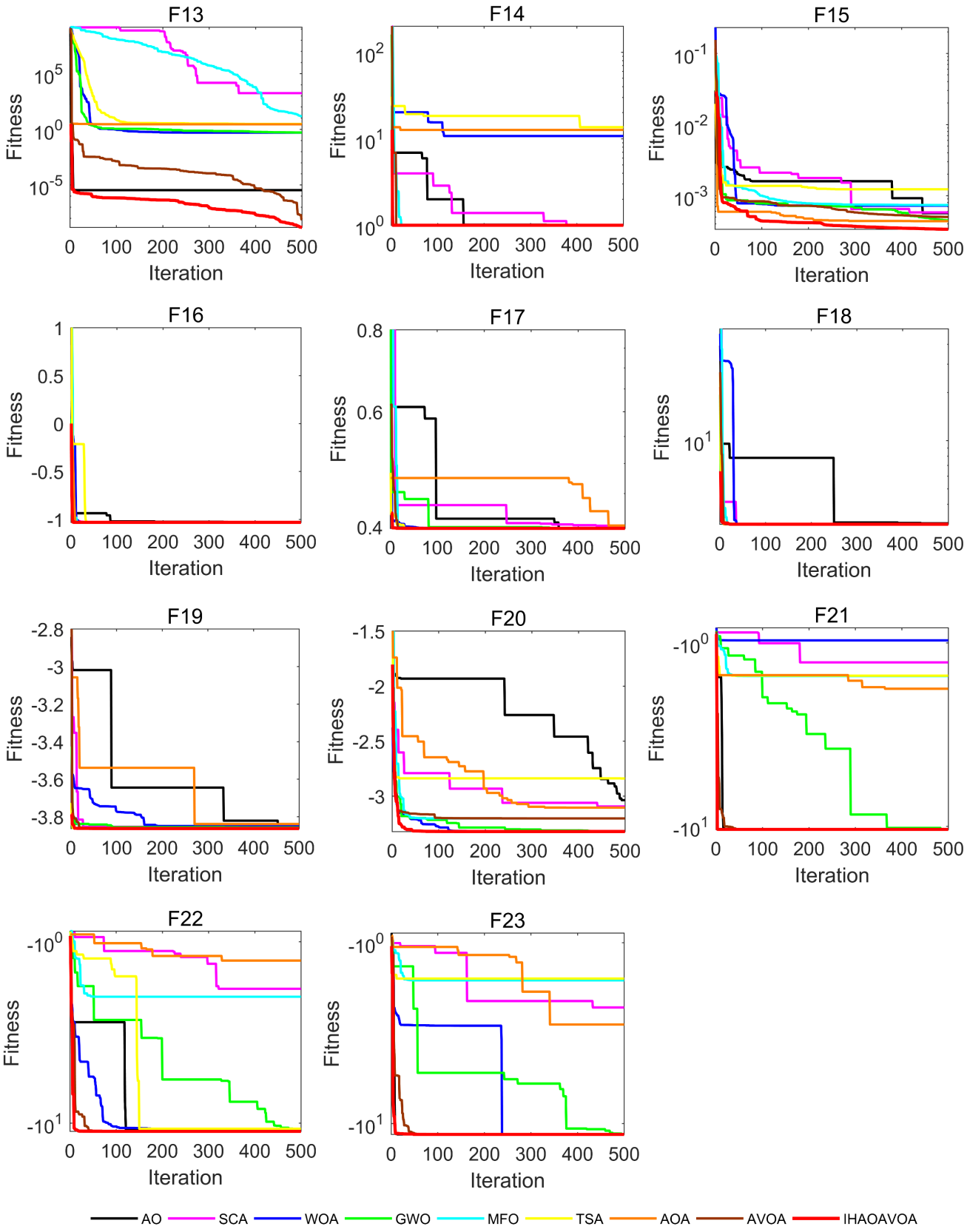


Figure 8. Convergence curves of different algorithms on 23 benchmark functions.

Table 9. Statistical results of the Wilcoxon rank-sum test between IHAOAVOA and other algorithms on 23 benchmark functions.

F_n	IHAOAVOA *VS.*							
	AO	SCA	WOA	GWO	MFO	TSA	AOA	AVOA
F_1	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	8.87E-07
F_2	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NaN	1.21E-12
F_3	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F_4	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F_5	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.34E-11
F_6	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	6.07E-11
F_7	8.84E-07	3.02E-11	2.37E-10	3.02E-11	3.02E-11	3.02E-11	1.15E-02	7.74E-06
F_8	7.39E-11	3.02E-11	6.70E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	7.30E-04
F_9	NaN	1.21E-12	1.61E-11	1.17E-12	1.21E-12	1.21E-12	NaN	NaN
F_{10}	NaN	1.21E-12	3.32E-10	1.15E-12	1.21E-12	1.21E-12	NaN	NaN
F_{11}	NaN	1.21E-12	4.19E-02	1.37E-03	1.21E-12	5.37E-06	1.21E-12	NaN
F_{12}	4.98E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.21E-10
F_{13}	6.72E-10	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.82E-09
F_{14}	1.20E-07	3.16E-08	1.12E-08	8.71E-10	6.35E-05	8.53E-11	5.73E-11	7.83E-03
F_{15}	6.12E-10	4.98E-11	5.09E-08	2.53E-04	3.67E-11	7.74E-06	9.92E-11	2.25E-04
F_{16}	3.15E-12	3.15E-12	3.15E-12	3.15E-12	6.78E-16	3.15E-12	3.15E-12	**8.12E-01**
F_{17}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NaN	1.21E-12	1.21E-12	5.14E-06
F_{18}	3.02E-11	3.34E-11	7.39E-11	3.02E-11	2.48E-11	3.02E-11	3.35E-08	6.72E-10
F_{19}	2.99E-11	2.99E-11	2.99E-11	2.99E-11	1.20E-12	2.99E-11	2.99E-11	**9.88E-01**
F_{20}	8.48E-09	3.02E-11	6.77E-05	6.77E-05	7.69E-08	4.74E-06	8.15E-11	3.92E-02
F_{21}	3.01E-11	3.01E-11	3.01E-11	3.01E-11	7.17E-10	3.01E-11	3.01E-11	6.52E-05
F_{22}	3.00E-11	3.00E-11	3.00E-11	3.00E-11	3.00E-11	3.00E-11	3.00E-11	4.18E-03
F_{23}	2.98E-11	2.98E-11	2.98E-11	2.98E-11	2.98E-11	2.98E-11	2.98E-11	1.22E-07
(W|T|L)	20/3/0	23/0/0	23/0/0	23/0/0	22/1/0	23/0/0	20/3/0	18/3/2

Note: The obtained *p*-values greater than 0.5 have been marked in bold.

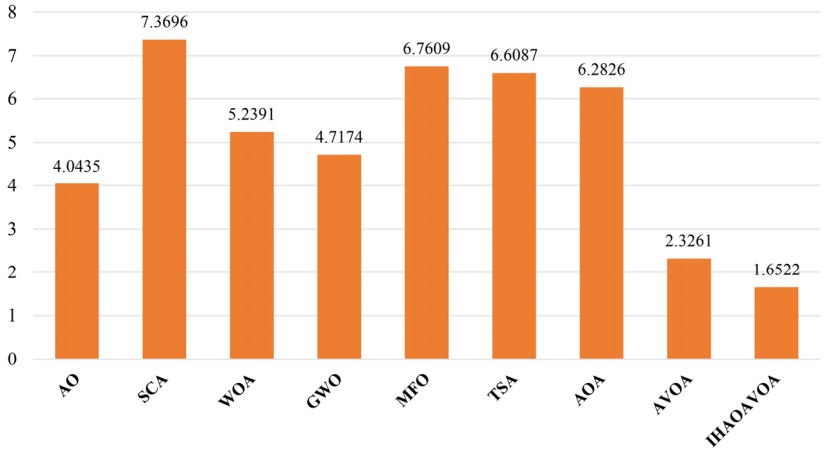


Figure 9. Friedman mean rank of different algorithms on 23 benchmark functions.

4.1.7. Scalability analysis

Scalability is a critical metric that represents the impact of dimension expansion on performance fluctuations of the algorithm. From the above experimental results, it can be seen that IHAOAVOA achieves good convergence on low-dimensional benchmark functions. However, complex high-dimensional optimization problems prevail in practical applications, and many algorithms are prone to failure when dealing with such problems. To further verify the effectiveness of the proposed method for high-dimensional optimization, IHAOAVOA is applied to solve 13 benchmark functions F_1–F_{13} in different dimensions $D = \{100, 500, 1000\}$. The parameter settings remain the same as in the previous experiments, and the obtained results of the nine algorithms after 30 independent runs are reported in Table 10.

From the data comparison in Table 10, it can be seen that IHAOAVOA also performs well in the condition of high dimensions. For functions F_1–F_4, F_9, and F_{11}, IHAOAVOA always finds the global optimal solution to the problem, regardless of whether the dimensions change. For functions F_5–F_8, F_{12}, and F_{13}, like its peers, the optimization accuracy of IHAOAVOA decreases as the number of dimensions increases. The main reason for this is that the larger the dimension of the data, the more complex the search space and the more elements that need to be optimized. However, the performance of IHAOAVOA does not deteriorate significantly. Compared with AO, AVOA, and other optimizers, the proposed method still provides superior outcomes. For function F_{10}, the scalable results of IHAOAVOA are consistent with those of AO and AVOA. Meanwhile, it is worth noting from Table 10 that these comparison algorithms (SCA, WOA, GWO, MFO, TSA, and AOA) show poor search capability for some issues, especially in higher dimensions. In order to better illustrate the overall performance of IHAOAVOA on scalable test functions, a Friedman ranking test based on the average fitness values is carried out and presented in Figure 10. It is clear from this figure that the proposed IHAOAVOA ranks first among all algorithms independent of dimensionality.

These results prove that IHAOAVOA doesn't suffer from the so-called “curse of dimension.” It can not only easily resolve low-dimensional problems, but also high-dimensional problems stably.

Table 10. Comparison results of IHAOAVOA and other algorithms on 13 benchmark functions in different dimensions ($D = 100/500/1000$).

F_n	dimension	criteria	AO	SCA	WOA	GWO	MFO	TSA	AOA	AVOA	IHAOAVOA
F_1	100	Avg	3.65E-109	1.14E+04	8.16E-73	2.23E-12	6.02E+04	4.57E-10	2.46E-02	1.24E-286	**0.00E+00**
		Std	1.96E-108	8.83E+03	2.92E-72	2.08E-12	1.32E+04	5.31E-10	7.55E-03	**0.00E+00**	**0.00E+00**
	500	Avg	2.02E-98	2.10E+05	4.83E-68	1.75E-03	1.16E+06	3.15E-02	6.31E-01	4.82E-291	**0.00E+00**
		Std	1.10E-97	8.92E+04	2.65E-67	7.17E-04	4.14E+04	2.74E-02	4.73E-02	**0.00E+00**	**0.00E+00**
	1000	Avg	2.01E-98	4.28E+05	6.52E-70	2.62E-01	2.74E+06	5.49E+00	1.72E+00	7.28E-275	**0.00E+00**

Continued on next page

F_n	dimension	criteria	AO	SCA	WOA	GWO	MFO	TSA	AOA	AVOA	IHAOAVOA
F_2	100	Std	1.10E-97	1.33E+05	2.59E-69	5.37E-02	6.28E+04	5.29E+00	7.99E-02	**0.00E+00**	**0.00E+00**
		Avg	3.15E-52	8.44E+00	4.10E-50	4.64E-08	2.50E+02	1.86E-07	2.37E-47	2.71E-151	**0.00E+00**
		Std	1.70E-51	6.84E+00	1.61E-49	1.89E-08	3.84E+01	1.89E-07	1.30E-46	1.46E-150	**0.00E+00**
	500	Avg	9.17E-54	1.15E+02	1.39E-48	1.12E-02	1.39E+131	8.00E-03	1.17E-03	2.68E-149	**0.00E+00**
		Std	5.02E-53	5.59E+01	4.89E-48	1.86E-03	7.61E+131	4.27E-03	1.42E-03	1.47E-148	**0.00E+00**
	1000	Avg	3.62E-56	Inf	1.13E-47	6.64E-01	Inf	2.78E-02	1.37E-02	2.15E-167	**0.00E+00**
		Std	1.98E-55	NaN	5.75E-47	3.48E-01	NaN	1.81E-02	4.73E-03	**0.00E+00**	**0.00E+00**
F_3	100	Avg	1.10E-100	2.39E+05	1.02E+06	7.84E+02	2.33E+05	1.39E+04	8.92E-01	7.42E-190	**0.00E+00**
		Std	5.21E-100	7.24E+04	3.34E+05	1.04E+03	5.44E+04	5.82E+03	5.54E-01	**0.00E+00**	**0.00E+00**
	500	Avg	8.08E-98	6.86E+06	3.08E+07	3.10E+05	5.15E+06	1.40E+06	2.89E+01	1.76E-141	**0.00E+00**
		Std	4.43E-97	1.48E+06	9.14E+06	7.34E+04	1.07E+06	2.08E+05	1.46E+01	9.66E-141	**0.00E+00**
	1000	Avg	7.08E-99	3.00E+07	1.24E+08	1.54E+06	1.81E+07	6.00E+06	1.33E+02	6.26E-137	**0.00E+00**
		Std	3.88E-98	5.35E+06	3.97E+07	2.78E+05	3.51E+06	8.85E+05	6.30E+01	3.43E-136	**0.00E+00**
F_4	100	Avg	2.32E-55	9.05E+01	7.68E+01	1.19E+00	9.29E+01	5.39E+01	9.23E-02	2.79E-145	**0.00E+00**
		Std	1.04E-54	2.87E+00	2.18E+01	1.54E+00	1.81E+00	1.17E+01	1.12E-02	1.53E-144	**0.00E+00**
	500	Avg	5.78E-66	9.91E+01	8.14E+01	6.58E+01	9.88E+01	9.92E+01	1.76E-01	3.17E-135	**0.00E+00**
		Std	3.08E-65	2.26E-01	2.16E+01	5.56E+00	4.27E-01	1.95E-01	1.17E-02	1.74E-134	**0.00E+00**
	1000	Avg	2.11E-52	9.96E+01	8.07E+01	7.95E+01	9.95E+01	9.96E+01	2.11E-01	1.73E-142	**0.00E+00**
		Std	1.14E-51	1.07E-01	2.13E+01	3.10E+00	1.35E-01	1.06E-01	1.24E-02	5.85E-142	**0.00E+00**
F_5	100	Avg	1.55E-02	1.26E+08	9.81E+01	9.78E+01	1.64E+08	9.79E+01	9.89E+01	4.40E-04	**9.38E-06**
		Std	2.97E-02	5.75E+07	2.92E-01	7.70E-01	6.41E+07	7.90E-01	6.46E-02	4.35E-04	**2.26E-05**
	500	Avg	1.96E-01	1.92E+09	4.96E+02	4.98E+02	5.02E+09	1.68E+05	4.99E+02	9.13E-03	**5.02E-03**
		Std	5.22E-01	4.07E+08	3.87E-01	3.50E-01	2.17E+08	2.93E+05	9.35E-02	2.40E-02	**3.41E-03**
	1000	Avg	1.76E-01	4.68E+09	9.94E+02	1.05E+03	1.25E+10	4.31E+07	9.99E+02	2.25E-02	**2.34E-03**
		Std	3.39E-01	8.24E+08	8.99E-01	1.64E+01	2.21E+08	2.80E+07	1.36E-01	3.21E-02	**4.73E-03**
F_6	100	Avg	4.75E-04	1.08E+04	4.31E+00	1.02E+01	6.12E+04	1.45E+01	1.81E+01	1.11E-03	**7.11E-06**
		Std	1.29E-03	5.64E+03	8.66E-01	1.09E+00	1.47E+04	1.09E+00	4.87E-01	4.33E-03	**1.03E-05**

Continued on next page

F_n	dimension	criteria	AO	SCA	WOA	GWO	MFO	TSA	AOA	AVOA	IHAOAVOA
	500	Avg	9.26E-04	1.84E+05	3.08E+01	9.14E+01	1.16E+06	1.03E+02	1.16E+02	6.68E-02	**9.07E-05**
		Std	1.88E-03	8.36E+04	9.20E+00	1.92E+00	3.36E+04	1.89E+00	1.04E+00	1.66E-01	**1.67E-04**
	1000	Avg	9.62E-03	4.58E+05	6.47E+01	2.04E+02	2.72E+06	2.33E+02	2.42E+02	1.96E-01	**7.29E-05**
		Std	4.81E-02	1.72E+05	1.55E+01	2.66E+00	5.21E+04	4.54E+00	1.42E+00	3.90E-01	**1.38E-04**
	100	Avg	1.04E-04	1.73E+02	3.86E-03	7.35E-03	2.55E+02	4.92E-02	6.81E-05	1.45E-04	**3.93E-05**
		Std	1.11E-04	1.04E+02	3.61E-03	2.74E-03	1.09E+02	1.99E-02	8.37E-05	1.60E-04	**3.02E-05**
F_7	500	Avg	6.38E-05	1.59E+04	4.32E-03	4.72E-02	3.84E+04	2.74E+00	6.75E-05	2.15E-04	**3.60E-05**
		Std	6.13E-05	3.40E+03	5.14E-03	1.66E-02	1.97E+03	1.03E+00	3.76E-05	2.50E-04	**3.32E-05**
	1000	Avg	1.04E-04	6.92E+04	5.33E-03	1.53E-01	1.97E+05	3.00E+02	9.35E-05	2.00E-04	**4.50E-05**
		Std	8.32E-05	1.11E+04	6.57E-03	3.13E-02	7.79E+03	1.53E+02	6.11E-05	2.07E-04	**5.80E-05**
	100	Avg	-9328.482	-7056.794	-34621.557	-15216.923	-23112.332	-12933.983	-10052.935	-40933.164	**-41793.696**
		Std	1.86E+03	7.12E+02	6.31E+03	3.35E+03	1.92E+03	1.07E+03	7.33E+02	1.81E+03	**4.30E+02**
F_8	500	Avg	-41534.395	-15465.639	-185802.018	-55265.276	-61273.898	-31127.517	-22505.588	-200569.062	**-203821.173**
		Std	1.29E+04	**1.28E+03**	2.68E+04	1.27E+04	4.38E+03	2.30E+03	1.63E+03	1.39E+04	1.99E+04
	1000	Avg	-60526.163	-22031.394	-354116.605	-84858.638	-89734.384	-45407.853	-32295.698	-403963.628	**-409246.624**
		Std	1.13E+04	**1.50E+03**	5.61E+04	1.81E+04	5.97E+03	3.24E+03	2.22E+03	2.19E+04	4.02E+04
	100	Avg	**0.00E+00**	2.96E+02	3.79E-15	1.10E+01	8.55E+02	9.59E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**
		Std	**0.00E+00**	1.09E+02	2.08E-14	7.55E+00	6.78E+01	1.25E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**
F_9	500	Avg	**0.00E+00**	1.13E+03	3.03E-14	7.78E+01	6.96E+03	5.84E+03	6.08E-06	**0.00E+00**	**0.00E+00**
		Std	**0.00E+00**	5.21E+02	1.66E-13	2.20E+01	1.75E+02	6.14E+02	5.47E-06	**0.00E+00**	**0.00E+00**
	1000	Avg	6.06E-14	1.87E+03	6.06E-14	1.88E+02	1.55E+04	9.86E+03	6.05E-05	**0.00E+00**	**0.00E+00**
		Std	3.32E-13	6.87E+02	3.32E-13	4.00E+01	2.02E+02	1.81E+03	1.52E-05	**0.00E+00**	**0.00E+00**
	100	Avg	**8.88E-16**	1.84E+01	3.97E-15	1.28E-07	1.98E+01	1.00E-01	5.51E-04	**8.88E-16**	**8.88E-16**
		Std	**0F.00E+00**	4.10E+00	2.59E-15	5.39E-08	1.98E-01	5.50E-01	9.61E-04	**0.00E+00**	**0.00E+00**
F_{10}	500	Avg	**8.88E-16**	1.97E+01	3.85E-15	1.99E-03	2.03E+01	1.22E-02	7.92E-03	**8.88E-16**	**8.88E-16**
		Std	**0.00E+00**	2.79E+00	2.10E-15	3.92E-04	1.64E-01	6.72E-03	3.73E-04	**0.00E+00**	**0.00E+00**
	1000	Avg	**8.88E-16**	1.84E+01	4.32E-15	1.78E-02	2.04E+01	9.91E-02	9.26E-03	**8.88E-16**	**8.88E-16**

Continued on next page

F_n	dimension	criteria	AO	SCA	WOA	GWO	MFO	TSA	AOA	AVOA	IHAOAVOA
F_{11}	100	Std	**0.00E+00**	4.62E+00	2.18E-15	2.54E-03	2.02E-01	4.31E-02	2.76E-04	**0.00E+00**	**0.00E+00**
		Avg	**0.00E+00**	1.05E+02	1.05E-02	4.31E-03	5.50E+02	1.16E-02	6.17E+02	**0.00E+00**	**0.00E+00**
	500	Std	**0.00E+00**	5.66E+01	5.74E-02	1.01E-02	1.87E-02	1.87E-02	1.94E+02	**0.00E+00**	**0.00E+00**
		Avg	**0.00E+00**	1.76E+03	**0.00E+00**	1.04E-02	1.02E+04	3.18E-02	1.09E+04	**0.00E+00**	**0.00E+00**
		Std	**0.00E+00**	6.34E+02	**0.00E+00**	2.73E-02	2.76E+02	7.30E-02	2.71E+03	**0.00E+00**	**0.00E+00**
	1000	Avg	**0.00E+00**	3.92E+03	**0.00E+00**	4.35E-02	2.46E+04	3.33E-01	2.82E+04	**0.00E+00**	**0.00E+00**
		Std	**0.00E+00**	1.37E+03	**0.00E+00**	6.94E-02	3.99E+02	2.20E-01	2.97E+02	**0.00E+00**	**0.00E+00**
F_{12}	100	Avg	1.87E-06	3.30E+08	4.86E-02	2.97E-01	2.61E+08	1.22E+01	9.06E-01	9.92E-07	**5.66E-08**
		Std	3.02E-06	1.71E+08	2.88E-02	6.58E-02	1.57E+08	4.24E+00	2.21E-02	2.35E-06	**7.68E-08**
	500	Avg	7.58E-07	6.06E+09	9.05E-02	7.39E-01	1.21E+10	3.34E+06	1.09E+00	7.86E-06	**9.84E-08**
		Std	1.50E-06	1.13E+09	4.06E-02	5.92E-02	7.34E+08	3.63E+06	1.14E-02	3.58E-05	**1.47E-07**
	1000	Avg	1.25E-06	1.29E+10	9.69E-02	1.24E+00	3.02E+10	5.25E+08	1.11E+00	9.31E-06	**8.78E-08**
		Std	2.49E-06	2.33E+09	3.70E-02	2.99E-01	1.44E+09	2.37E+08	5.40E-03	2.15E-05	**1.65E-07**
F_{13}	100	Avg	4.67E-05	4.96E+08	3.00E+00	7.00E+00	6.68E+08	1.28E+01	9.98E+00	1.46E-07	**4.88E-08**
		Std	7.09E-05	2.77E+08	1.01E+00	3.94E-01	3.66E+08	1.59E+00	4.17E-02	1.65E-07	**8.26E-08**
	500	Avg	2.79E-04	9.65E+09	1.96E+01	5.09E+01	2.19E+10	1.15E+06	5.02E+01	8.40E-07	**6.80E-07**
		Std	4.18E-04	2.02E+09	6.16E+00	1.38E+00	1.36E+09	1.24E+06	3.62E-02	1.60E-06	**1.26E-06**
	1000	Avg	5.17E-04	2.21E+10	3.81E+01	1.23E+02	5.60E+10	3.07E+08	1.01E+02	5.04E-03	**3.41E-06**
		Std	8.71E-04	4.63E+09	1.17E+01	8.57E+00	1.89E+09	2.23E+08	5.93E-02	2.01E-02	**6.58E-06**

Note: The best results obtained have been marked in bold.

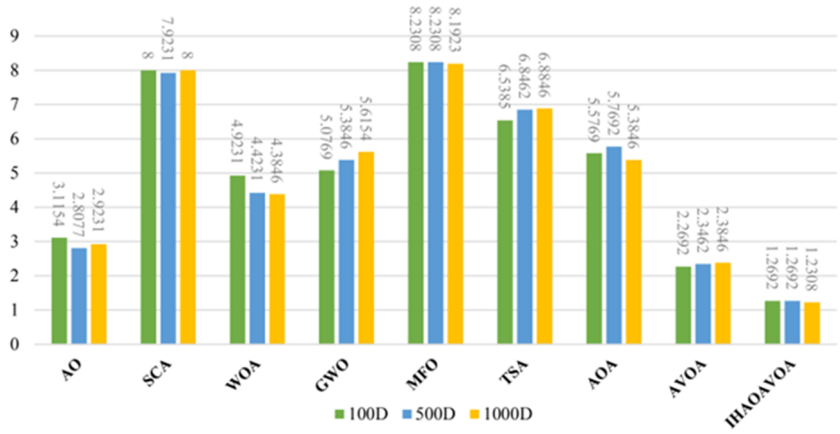


Figure 10. Friedman mean rank of different algorithms in 100/500/1000 dimensions.

4.2. Experiment 2: IEEE CEC2019 test suite

Classical benchmark function experiments have proven the prominent performance of IHAOAVOA with respect to solving simple optimization problems. To further emphasize the superiority of the improved algorithm in this paper, this subsection uses the IEEE CEC2019 test suite [98], also known as *100-Digit Challenge*, to estimate the performance of IHAOAVOA in solving complex numerical problems. This test suite comprises ten complicated and latest benchmark functions, the profiles of which are listed in Table 11. As stated in the previous subsection, the proposed IHAOAVOA and other eight comparison algorithms run independently 30 times on each function with the maximum iteration and population size fixed to 500 and 30, respectively. The obtained average value and standard deviation results of this test are presented in Table 12. Meanwhile, the Friedman mean rank values used for statistical analysis of algorithms are included in the last line of this table.

Table 11. Details of IEEE CEC2019 test suite.

function	name	*D*	range	F_{min}
CEC-01	Storn's Chebyshev Polynomial Fitting Problem	9	[-8192, 8192]	1
CEC-02	Inverse Hilbert Matrix Problem	16	[-16384, 16384]	1
CEC-03	Lennard-Jones Minimum Energy Cluster	18	[-4, 4]	1
CEC-04	Rastrigin's Function	10	[-100, 100]	1
CEC-05	Griewangk's Function	10	[-100, 100]	1
CEC-06	Weierstrass Function	10	[-100, 100]	1
CEC-07	Modified Schwefel's Function	10	[-100, 100]	1
CEC-08	Expanded Schaffer's F6 Function	10	[-100, 100]	1
CEC-09	Happy Cat Function	10	[-100, 100]	1
CEC-10	Ackley Function	10	[-100, 100]	1

From Table 12, it is clear that IHAOAVOA outperforms the other eight algorithms on 6 out of 10 test functions. For CEC-2, although GWO and MFO achieve the same average fitness as IHAOAVOA, the proposed algorithm has a smaller standard deviation, which demonstrates the better stability of IHAOAVOA. For CEC-5 and CEC-9, the performance of IHAOAVOA is slightly worse than that of MFO, but it still ranks second among all algorithms. For CEC-7, AOA provides the most satisfactory solutions, whereas IHAOAVOA also performs quite competitively. Besides, compared with its peers, IHAOAVOA obtains the best Friedman mean ranking value of 1.8000 followed by the MFO algorithm. These findings demonstrate that the proposed IHAOVAOA is capable of tackling various challenging optimization problems as well.

To summarize, the effectiveness and superiority of the proposed method are thoroughly verified in this section through a series of experiments on classical benchmark functions and the IEEE CEC2019 test suite. Whether solving simple or complex numerical problems, IHAOAVOA can give satisfactory results in most cases. IHAOAVOA inherits the merits of the basic AO and AVOA and makes use of the COBL and FDB strategies to compensate for the defects of poor population diversity, the tendency to fall into local optima, and the imbalance between exploration and exploitation. Of course, a good algorithm needs to be applied in practice to show its value. In the next section, IHAOAVOA will be used to address five constrained industrial engineering problems.

Table 12. Comparison results of different algorithms on IEEE CEC2019 test suite.

F_n	criteria	AO	SCA	WOA	GWO	MFO	TSA	AOA	AVOA	IHAOAVOA
CEC-1	Avg	5.58E+04	3.76E+09	3.77E+10	1.75E+08	1.96E+10	2.03E+08	7.86E+09	4.50E+04	**4.14E+04**
	Std	8.33E+03	4.69E+09	4.58E+10	3.03E+08	2.96E+10	3.54E+08	2.74E+10	3.35E+03	**2.60E+03**
CEC-2	Avg	1.74E+01	1.75E+01	1.74E+01	**1.73E+01**	**1.73E+01**	1.85E+01	1.93E+01	1.74E+01	**1.73E+01**
	Std	1.13E-02	5.17E-02	1.55E-02	3.06E-04	7.14E-12	6.14E-01	4.85E-01	6.10E-02	**9.47E-14**
CEC-3	Avg	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	**1.27E+01**
	Std	6.95E-06	1.05E-04	1.54E-06	8.00E-06	2.77E-04	9.66E-04	1.23E-03	5.21E-09	**1.26E-09**
CEC-4	Avg	7.15E+02	1.48E+03	3.66E+02	1.38E+02	1.82E+02	4.14E+03	1.31E+04	1.56E+02	**1.28E+02**
	Std	4.63E+02	6.40E+02	1.18E+02	4.30E+02	1.91E+02	2.76E+03	5.83E+03	6.60E+01	**4.03E+01**
CEC-5	Avg	1.59E+00	2.19E+00	1.86E+00	1.39E+00	**1.28E+00**	2.79E+00	4.22E+00	1.52E+00	1.36E+00
	Std	2.74E-01	4.15E-01	3.64E-01	2.21E-01	**1.38E-01**	7.72E-01	1.00E+00	3.45E-01	2.15E-01
CEC-6	Avg	1.07E+01	1.09E+01	9.70E+00	1.10E+01	6.21E+00	1.12E+01	8.97E+00	6.23E+00	**5.77E+00**
	Std	7.52E-01	7.05E-01	1.26E+00	6.08E-01	2.21E+00	6.12E-01	2.00E+00	1.87E+00	**1.79E+00**
CEC-7	Avg	4.32E+02	8.03E+02	4.92E+02	4.22E+02	3.52E+02	6.96E+02	**2.03E+02**	3.60E+02	3.07E+02
	Std	2.10E+02	1.78E+02	2.35E+02	2.95E+02	1.91E+02	1.88E+02	**1.15E+02**	2.01E+02	1.67E+02

Continued on next page

F_n	criteria	AO	SCA	WOA	GWO	MFO	TSA	AOA	AVOA	IHAOAVOA
CEC-8	Avg	5.45E+00	5.97E+00	6.09E+00	5.41E+00	5.67E+00	6.20E+00	5.47E+00	5.55E+00	**5.32E+00**
	Std	5.66E-01	4.66E-01	5.10E-01	9.01E-01	6.79E-01	6.03E-01	5.28E-01	5.77E-01	**5.03E-01**
CEC-9	Avg	5.00E+00	9.82E+01	5.09E+00	4.76E+00	**2.87E+00**	6.03E+02	7.66E+02	3.62E+00	3.54E+00
	Std	7.93E-01	6.64E+01	9.13E-01	9.75E-01	**3.89E-01**	6.71E+02	4.36E+02	7.50E-01	6.52E-01
CEC-10	Avg	2.04E+01	2.05E+01	2.03E+01	2.02E+01	2.02E+01	2.05E+01	2.01E+01	2.03E+01	**2.00E+01**
	Std	1.16E-01	7.95E-02	1.19E-01	1.48E+00	1.51E-01	8.28E-02	6.61E-02	6.71E-02	**5.41E-02**
Friedman mean ranking		5.1000	7.0500	6.1500	3.8500	3.5500	7.6500	5.9000	3.9500	**1.8000**

Note: The best results obtained have been marked in bold.

5. IHAOAVOA for solving engineering design problems

In this section, five common engineering design problems from the structural field are utilized to highlight the applicability and black-box nature of the proposed IHAOAVOA in real-world constrained optimization, which are tension/compression spring design problem, welded beam design problem, cantilever beam design problem, speed reducer design problem, and rolling element bearing design problem. For convenience, the death penalty function [99] is introduced here to handle those infeasible candidate solutions subject to equality and inequality constraints. In the same way, we set the maximum number of iterations and population size as 500 and 30, respectively. The detailed comparison results of IHAOAVOA and other algorithms after 30 times of independent runs on each project are presented and discussed below.

5.1. Tension/compression spring design problem

As shown in Figure 11, the goal of this optimization problem is to find three optimal design variables, namely diameter of the wire (d), average coil diameter (D), and active coils number (N), to reduce the weight of a tension/compression spring as much as possible. Meanwhile, the constraints of shear stress, surge frequency, and minimum deflection should be satisfied in the minimization process. The mathematical model of this design is formulated as follows.

Consider

$$\vec{z} = [z_1, z_2, z_3] = [d, D, N]$$

Minimize

$$f(\vec{z}) = (z_3 + 2)z_2 z_1^2$$

Subject to

$$g_1(\vec{z}) = 1 - \frac{z_2^3 z_3}{71785 z_1^4} \leq 0, g_2(\vec{z}) = \frac{4z_2^2 - z_1 z_2}{12566(z_2 z_1^3 - z_1^4)} + \frac{1}{5108 z_1^2} \leq 0,\ g_3(\vec{z}) = 1 - \frac{140.45 z_1}{z_2^2 z_3} \leq 0, g_4(\vec{z}) = \frac{z_1 + z_2}{1.5} - 1 \leq 0$$

Variable range

$$0.05 \leq z_1 \leq 2, 0.25 \leq z_2 \leq 1.30, 2.00 \leq z_3 \leq 15.00$$

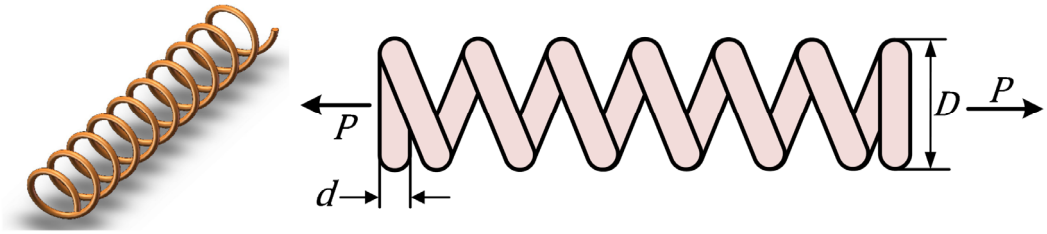


Figure 11. Schematic view of tension/compression spring design problem.

Table 13. Comparison results of different algorithms for tension/compression spring design problem.

algorithm	optimal values for variables			minimum weight
	$d(z_1)$	$D(z_2)$	$N(z_3)$	
AO	0.0505978	0.330908	13.1244	0.012708
SCA	0.0500431	0.318378	13.7796	0.012757
WOA	0.0500000	0.310414	15.0000	0.013193
GWO	0.0545730	0.430150	8.1728	0.012811
MFO	0.0571830	0.503870	6.2155	0.013181
TSA	0.0536750	0.405150	9.4851	0.012840
AOA	0.0526750	0.380910	9.8629	0.012683
AVOA	0.0500150	0.317762	14.3427	0.012718
IHAOAVOA	0.0518973	0.361749	10.5783	**0.012666**

Note: The best results obtained have been marked in bold.

The performance evaluation of the optimal solution obtained by the proposed IHAOAVOA on this application is compared with those of AO, SCA, WOA, GWO, MFO, TSA, AOA, and AVOA, as listed in Table 13. It can be observed from this table that IHAOAVOA outperforms all other comparison algorithms and reveals the minimum weight $f_{min}(\vec{z}) = 0.012666$ corresponding to the best solution $\vec{z} = [0.0518973, 0.361749, 10.5783]$, which demonstrates the merits of IHAOAVOA in resolving the tension/compression spring design problem.

5.2. Welded beam design problem

Just as its name implies, this well-known engineering case first proposed by Coello [99] aims at

minimizing the overall fabrication cost of a welded beam under the constraints on shear stress, bending stress in the beam, buckling load, and end deflection. As illustrated in Figure 12, there are four decision parameters that need to be considered in this problem such as the weld thickness (h), the length of the joint beam (l), the height of the beam (t), and the thickness of the beam (b). The mathematical representation of this optimization is described as follows.

Consider

$$\vec{z} = [z_1, z_2, z_3, z_4] = [h, l, t, b]$$

Minimize

$$f(\vec{z}) = 1.10471 z_1^2 z_2 + 0.04811 z_3 z_4 (14 + z_2)$$

Subject to

$$g_1(\vec{z}) = \tau(\vec{z}) - \tau_{\max} \le 0, g_2(\vec{z}) = \sigma - \sigma_{\max} \le 0, g_3(\vec{z}) = \delta - \delta_{\max} \le 0,$$
$$g_4(\vec{z}) = z_1 - z_4 \le 0, g_5(\vec{z}) = P - P_c(\vec{z}) \le 0, g_6(\vec{z}) = 0.125 - z_1 \le 0,$$
$$g_7(\vec{z}) = 1.10471 z_1^2 + 0.04811 z_3 z_4 (14 + z_2) - 5 \le 0$$

Variable range

$$0.1 \le z_1, z_4 \le 2, 0.1 \le z_2, z_3 \le 10$$

where

$$\tau(\vec{z}) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{z_2}{2R} + (\tau'')^2}, \tau' = \frac{P}{\sqrt{2} z_1 z_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{z_2}{2}\right), R = \sqrt{\frac{z_2^2}{4} + \left(\frac{z_1 + z_3}{2}\right)^2},$$

$$J = 2\left\{\sqrt{2} z_1 z_2 \left[\frac{z_2^2}{4} + \left(\frac{z_1 + z_3}{2}\right)^2\right]\right\}, \sigma(\vec{z}) = \frac{6PL}{E z_3^2 z_4}\ \delta(\vec{z}) = \frac{6PL^3}{E z_3^2 z_4}, P_C(\vec{z}) = \frac{4.013E\sqrt{\frac{z_3^2 z_4^6}{36}}}{L^2}\left(1 - \frac{z_3}{2L}\sqrt{\frac{E}{4G}}\right), P = 6000\text{lb},$$

$$L = 14\text{in}, E = 30 \times 10^6\text{psi}, G = 12 \times 10^6\text{psi}, \delta_{\max} = 0.25\text{in}, \tau_{\max} = 13600\text{psi}, \sigma_{\max} = 30000\text{psi}.$$

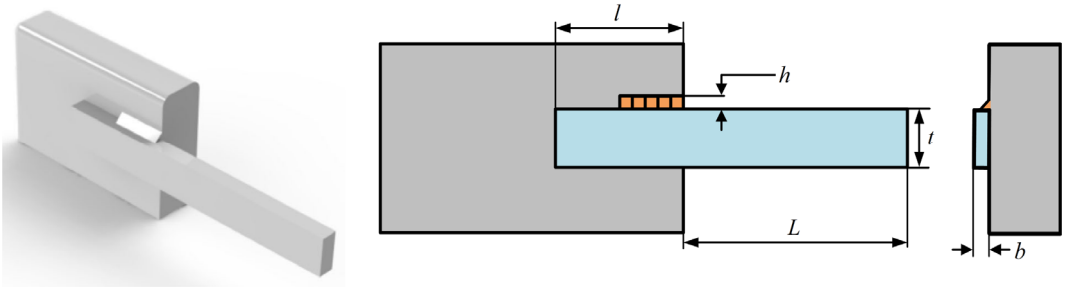


Figure 12. Schematic view of welded beam design problem.

This problem has been figured out using IHAOAVOA and the remaining eight methods. The optimal solutions are summarized in Table 14. It can be seen that the minimum manufacturing cost of IHAOAVOA is 1.7249 when the four variables h, l, t, and b are set as 0.20573, 3.4705, 9.0366, and 0.20573, respectively. In this comparison, IHAOAVOA attains a superior outcome to all the other optimization techniques, which suggests that the proposed hybrid algorithm in this paper can be regarded as a promising tool to deal with the welded beam design problem.

Table 14. Comparison results of different algorithms for welded beam design problem.

algorithm	optimal values for variables				minimum cost
	$h(z_1)$	$l(z_1)$	$t(z_3)$	$b(z_4)$	
AO	0.20355	3.5170	9.0392	0.20572	1.7281
SCA	0.19979	3.6142	9.0393	0.20572	1.7352
WOA	0.20241	3.5518	9.0302	0.20609	1.7322
GWO	0.20476	3.4958	9.0409	0.20576	1.7277
MFO	0.20288	3.5332	9.0359	0.20576	1.7290
TSA	0.19894	3.6141	9.0584	0.20562	1.7364
AOA	0.20628	3.4652	9.0199	0.20649	1.7279
AVOA	0.20638	3.4721	9.0212	0.20661	1.7301
IHAOAVOA	0.20573	3.4705	9.0366	0.20573	**1.7249**

Note: The best results obtained have been marked in bold.

5.3. Cantilever beam design problem

The design of the cantilever beam is also a popular research concern in real-life engineering optimization. Its main intention is to locate five optimal structural variables to dwindle the total weight of a cantilever beam when meeting the load capacity requirements. Figure 13 illustrates the architecture of the cantilever beam, which is made up of several hollow square-shaped sections. Mathematically, this problem is stated as follows.

Consider

$$\vec{z} = [z_1, z_2, z_3, z_4, z_5]$$

Minimize

$$f(\vec{z}) = 0.6224(z_1 + z_2 + z_3 + z_4 + z_5)$$

Subject to

$$g(\vec{z}) = \frac{61}{z_1^3} + \frac{27}{z_2^3} + \frac{19}{z_3^3} + \frac{7}{z_4^3} + \frac{1}{z_5^3} - 1 \leq 0$$

Variable range

$$0.01 \leq z_1, z_2, z_3, z_4, z_5 \leq 100$$

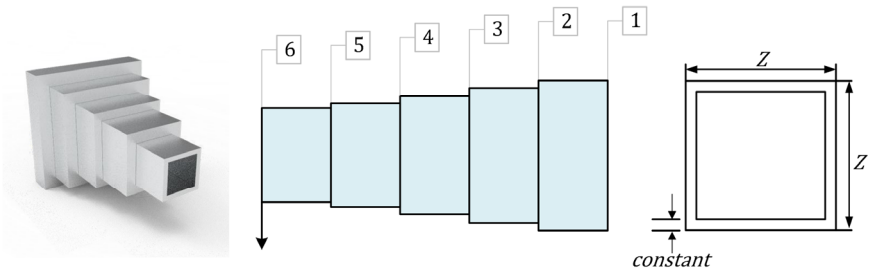


Figure 13. Schematic view of cantilever beam design problem.

The best results obtained by all algorithms for the cantilever beam design problem are recorded in Table 15. As can be seen from this table, the proposed IHAOAVOA has achieved the best design assurance with the lowest weight compared with the other optimizers. Besides, the basic AO and AVOA come in the second and seventh ranks, respectively. Therefore, it is reasonable to believe that IHAOAVOA has good potential for solving such a problem.

Table 15. Comparison results of different algorithms for cantilever beam design problem.

algorithm	optimal values for variables					minimum weight
	z_1	z_1	z_3	z_4	z_5	
AO	6.3708	5.0687	4.5731	3.4451	2.0924	1.3447
SCA	5.9121	5.0872	4.9220	3.4056	2.2583	1.3469
WOA	6.3667	5.2580	3.8672	4.0987	2.3313	1.3679
GWO	5.7371	5.5770	4.4891	3.5928	2.1354	1.3436
MFO	5.8949	5.4072	4.5087	3.4724	2.2016	1.3407
TSA	5.8791	5.2745	4.5557	3.5769	2.2081	1.3412
AOA	5.7563	5.3133	4.4690	3.7889	2.2155	1.3443
AVOA	5.9292	5.3891	4.4645	3.5405	2.1566	1.3403
IHAOAVOA	6.0108	5.3170	4.4678	3.5324	2.1466	**1.3400**

Note: The best results obtained have been marked in bold.

5.4. Speed reducer design problem

The purpose of this speed reducer design problem is to minimize the mass of a reducer by optimizing seven decision variables, which are the face width (z_1), module of teeth (z_2), the number of teeth in the pinion (z_3), length of the shafts between bearings (z_4, z_5), and diameter of the shafts (z_6, z_7). In addition, the design is subject to the limitations of the gear teeth' bending stress, the transverse deflection of the shafts, surface stress, and stresses in the shafts. The structure of this problem is depicted in Figure 14, and the related mathematical description can be specified as follows.

Consider

$$\vec{z} = [z_1, z_2, z_3, z_4, z_5, z_6, z_7]$$

Minimize

$$f(\vec{z}) = 0.7854 z_1 z_2^2 (3.3333 z_3^2 + 14.9334 z_3 - 43.0934) - 1.508 z_1 (z_6^2 + z_7^2) + 7.4777 (z_6^3 + z_7^3)$$

Subject to

$$g_1(\vec{z}) = \frac{27}{z_1 z_2^2 z_3} - 1 \le 0, g_2(\vec{z}) = \frac{397.5}{z_1 z_2^2 z_3^2} - 1 \le 0, g_3(\vec{z}) = \frac{1.93 z_4^3}{z_2 z_3 z_6^4} - 1 \le 0,\ g_4(\vec{z}) = \frac{1.93 z_5^3}{z_2 z_3 z_7^4} - 1 \le 0, g_5(\vec{z}) = \frac{\sqrt{\left(\frac{745 z_4}{z_2 z_3}\right)^2 + 16.9 \times 10^6}}{110.0 z_6^3} - 1 \le 0,$$

$$g_6(\vec{z}) = \frac{\sqrt{\left(\frac{745 z_4}{z_2 z_3}\right)^2 + 157.5 \times 10^6}}{85.0 z_6^3} - 1 \le 0, g_7(\vec{z}) = \frac{z_2 z_3}{40} - 1 \le 0,$$

$$g_8(\vec{z}) = \frac{5z_2}{z_1} - 1 \le 0, g_9(\vec{z}) = \frac{z_1}{12z_2} - 1 \le 0,\ g_{10}(\vec{z}) = \frac{1.5z_6+1.9}{z_4} - 1 \le 0, g_{11}(\vec{z}) = \frac{1.1z_7+1.9}{z_5} - 1 \le 0$$

Variable range

$$2.6 \le z_1 \le 3.6, 0.7 \le z_2 \le 0.8, 17 \le z_3 \le 28, 7.3 \le z_4 \le 8.3, 7.8 \le z_5 \le 8.3, 2.9 \le z_6 \le 3.9, 5.0 \le z_7 \le 5.5$$

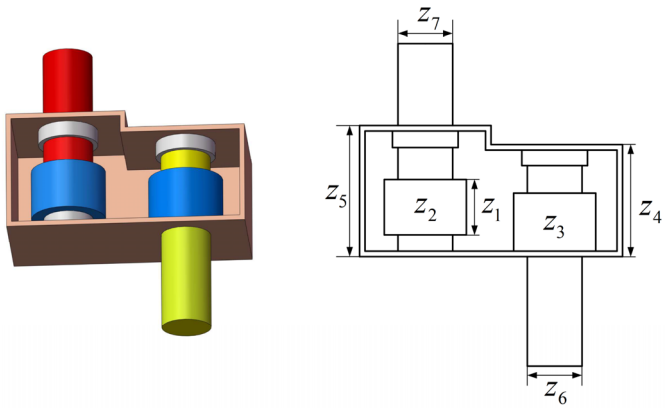


Figure 14. Schematic view of speed reducer design problem.

Table 16 shows the comparison results between different algorithms when solving the speed reducer design problem. Compared to AO, SCA, WOA, GWO, MFO, TSA, AOA, and AVOA, the proposed IHAOAVOA effectively provides higher-quality results. The optimal solution of IHAOAVOA is attained at $\vec{z} = [3.5, 0.7, 17, 7.30000, 7.71532, 3.35021, 5.28665]$ with the minimum weight $f_{min}(\vec{z}) = 2994.4711$. This example again showcases the excellent performance of IHAOAVOA at the practical application level.

Table 16. Comparison results of different algorithms for speed reducer design problem.

algorithm	optimal values for variables							minimum weight
	z_1	z_2	z_3	z_4	z_5	z_6	z_7	
AO	3.5	0.7	17	7.30091	7.82289	3.35022	5.28669	2996.8676
SCA	3.52991	0.7	17	7.64007	7.73596	3.38152	5.28666	3017.7743
WOA	3.5	0.7	17	8.27222	7.99218	3.35215	5.28675	3009.6826
GWO	3.50242	0.7	17.0123	7.46923	7.86195	3.35336	5.28693	3003.2403
MFO	3.5	0.7	17	7.53278	7.73890	3.35066	5.28666	2997.1598
TSA	3.50693	0.7	17	8.00463	8.11691	3.35393	5.28683	3013.2930
AOA	3.5	0.7	17	7.35758	7.78101	3.35032	5.28668	2996.4624
AVOA	3.5	0.7	17.0002	7.36821	7.72042	3.35034	5.28666	2995.2443
IHAOAVOA	3.5	0.7	17	7.30000	7.71532	3.35021	5.28665	**2994.4711**

Note: The best results obtained have been marked in bold.

5.5. Rolling element bearing design problem

The last constrained engineering problem is the design of the rolling element bearing, as

illustrated in Figure 15. In contrast to the optimization tasks mentioned above, the final objective of this test case is to maximize the dynamic loading carrying capacity of rolling element bearings. In this optimum design, a total of ten geometric parameters need to be taken into account, including pitch diameter (D_m), ball diameter (D_b), the number of balls (Z), the inner and outer raceway curvature radius coefficient (f_i and f_o), K_{dmin}, K_{dmax}, δ, e, and ζ. The problem has nine constraints and its mathematical formula is as follows.

Maximize

$$C_d = \begin{cases} f_c Z^{2/3} D_b^{1.8} & ,if\ D_b \le 25.4\text{mm} \\ 3.647 f_c Z^{2/3} D_b^{1.4}, & otherwise \end{cases}$$

Subject to

$$g_1(\vec{z}) = \frac{\phi_0}{2\sin^{-1}(D_b/D_m)} - Z + 1 \le 0, g_2(\vec{z}) = 2D_b - K_{dmin}(D-d) > 0,\ g_3(\vec{z}) = K_{dmax}(D-d) - 2D_b \ge 0, g_4(\vec{z}) = \zeta B_w - D_b \le 0,\ g_5(\vec{z}) = D_m - 0.5(D+d) \ge 0, g_6(\vec{z}) = (0.5+e)(D+d) - D_m \ge 0, g_7(\vec{z}) = 0.5(D - D_m - D_b) - \delta D_b \ge 0, g_8(\vec{z}) = f_i \ge 0.515,\ g_9(\vec{z}) = f_o \ge 0.515$$

where

$$f_c = 37.91\left[1 + \left\{1.04\left(\frac{1-\gamma}{1+\gamma}\right)^{1.72}\left(\frac{f_i(2f_o-1)}{f_o(2f_i-1)}\right)^{0.41}\right\}^{10/3}\right]^{-0.3} \times \left[\frac{\gamma^{0.3}(1-\gamma)^{1.39}}{(1+\gamma)^{1/3}}\right]\left[\frac{2f_i}{2f_i-1}\right]^{0.41}, x = [\{(D-d)/2 - 3(T/4)\}^2 + \{D/2 - T/4 - D_b\}^2 - \{d/2 + T/4\}^2], y = 2\{(D-d)/2 - 3(T/4)\}\{D/2 - T/4 - D_b\}, \phi_0 = 2\prod - cos^{-1}(\frac{x}{y}), \gamma = \frac{D_b}{D_m}, f_i = \frac{r_i}{D_b}, f_o = \frac{r_o}{D_b}, T = D - d - 2D_b\ \ D = 160, d = 90, B_w = 30, r_i = r_o = 11.033, 0.5(D+d) \le D_m \le 0.6(D+d)\ \ 0.15(D-d) \le D_b \le 0.45(D-d), 4 \le Z \le 50, 0.515 \le f_i \text{ and } f_o \le 0.6\ \ 0.4 \le K_{dmin} \le 0.5, 0.6 \le K_{dmax} \le 0.7, 0.3 \le \delta \le 0.4, 0.02 \le e \le 0.1, 0.6 \le \zeta \le 0.85.$$

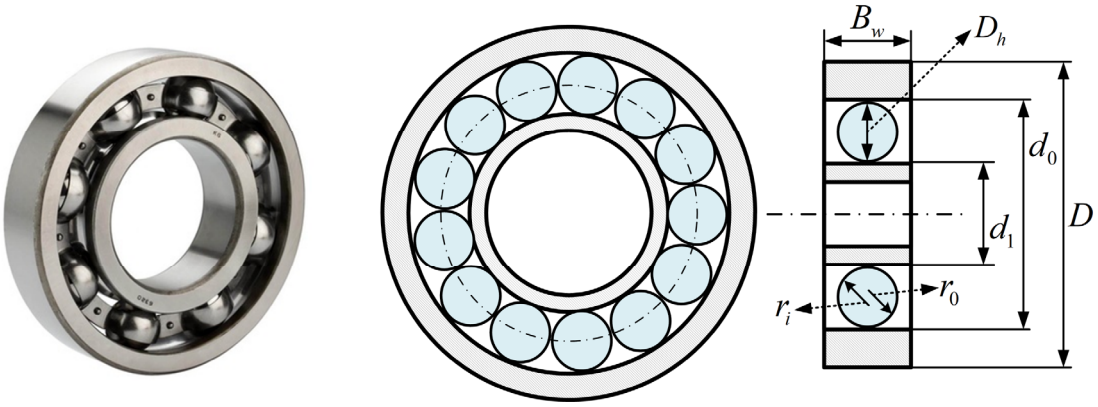


Figure 15. Schematic view of rolling element bearing design problem.

The detailed results of the optimum variables and cost for this problem are presented in Table 17. By examining the data in this table, it is evident that the proposed IHAOAVOA is capable of detecting a much better cost than its competitors, which is 85549.1628. And the results of MFO and AVOA are also very competitive.

In summary, the findings of this section strongly demonstrate that IHAOAVOA is equally

effective and feasible for practical engineering design applications. Attributed to the hybrid operation, COBL, and FDB, the exploration and exploitation capabilities of the algorithm developed in this paper have been dramatically improved. It is highly hopeful to apply IHAOAVOA to solve more real-life problems in various scenarios.

Table 17. Comparison results of different algorithms for rolling element bearing design problem.

algorithm	AO	SCA	WOA	GWO	MFO	TSA	AOA	AVOA	IHAOAVOA
D_m	125.5912	125	126.3104	125.6319	125.7191	125.3916	125	125.7186	125.7191
D_b	21.39605	21.15739	21.03404	21.39261	21.42559	21.28729	21.27301	21.42548	21.42559
Z	11.13631	10.90745	10.95836	11.03602	11.0039	10.78929	11.36323	10.7434	10.62575
f_i	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515
f_o	0.5240512	0.5359205	0.515	0.58441	0.5343591	0.5628563	0.515	0.5200027	0.5154035
K_{dmin}	0.4096218	0.4	0.4064247	0.4138138	0.4223729	0.4159249	0.4997957	0.4766428	0.4129783
K_{dmax}	0.657841	0.6423859	0.6011677	0.6946858	0.6512409	0.6156199	0.6964319	0.6598702	0.6282836
δ	0.301808	0.3	0.3008351	0.3021904	0.3	0.3	0.3	0.3000088	0.3
e	0.05709046	0.02251181	0.02775154	0.09110937	0.02285913	0.04570604	0.08114373	0.02112543	0.02000012
ζ	0.6169172	0.6	0.6	0.6154725	0.6001039	0.6	0.6	0.6041762	0.6507823
Maximum cost	85336.3471	83645.9245	82812.5215	85298.9012	85545.3137	84558.2474	84459.7849	85547.5187	**85549.1628**

Note: The best results obtained have been marked in bold.

6. Conclusions and future work

Considering the characteristics of Aquila Optimizer and African Vultures Optimization Algorithm, this paper proposed a novel improved hybrid meta-heuristic algorithm, namely IHAOAVOA, for solving global optimization problems. First, the exploration phase of AO and the exploitation phase of AVOA were integrated to accomplish superior overall search performance and alleviate the weaknesses existing in the single algorithm. Second, we designed a new composite opposition-based learning mechanism to enhance population diversity and increase the probability of obtaining the global optimal solution. Meanwhile, the fitness-distance balance selection method was used to choose one candidate solution that contributes most to the search process to replace the original random individual in the position update rules, which helps to better balance the exploration and development capabilities of the hybrid algorithm. To fully evaluate the function optimization performance, the proposed IHAOAVOA was compared with the basic AO, AVOA, and six advanced metaheuristics based on 23 classical benchmark functions and the IEEE CEC2019 test suite. The significance of obtained results was

verified through the Wilcoxon rank-sum test, Friedman test, and mean absolute error test. Numerical and statistical results indicate that IHAOAVOA significantly outperforms the other algorithms in terms of accuracy, convergence speed, stability, and local optima avoidance. Moreover, the proposed algorithm also shows stable performance in high-dimensional cases ($D = 100/500/1000$). To demonstrate the applicability of IHAOAVOA in practice, five engineering design problems were employed. It has been found that the proposed IHAOAVOA can effectively provide very competitive solutions in solving such real-life optimization issues as well.

Even though the proposed IHAOAVOA has remarkable improvements over the AO and AVOA algorithms, its computational cost is a potential limitation, and the performance on partial CEC2019 benchmark functions still has room to be further enhanced. In the subsequent research works, we will: 1) introduce some parallel strategies in IHAOAVOA, such as the co-evolutionary mechanism or cell model to reduce the time consumption under the guarantee of ensuring performance; 2) strengthen the exploration and exploitation capabilities of IHAOAVOA through other hybrid and general modification techniques to remove barriers on the IEEE CEC2019 test suite; 3) evaluate the performance differences between IHAOAVOA and some improved variants of AO on more challenging engineering application problems; 4) integrate the designed composite opposition-based learning mechanism into more MAs to enhance their search capabilities; 5) apply IHAOAVOA to solve different optimization problems in a wider range of disciplines, like feature selection, path planning, PID parameters self-tuning, forecast modeling, and image segmentation. Meanwhile, the identification of optimal process parameters for selective laser sintering (SLS) would also be a meaningful research topic.

Acknowledgments

The authors are grateful to the editor and reviewers for their constructive comments and suggestions, which have improved the presentation. And this work is financially supported by the National Natural Science Foundation of China under Grant 52075090, Key Research and Development Program Projects of Heilongjiang Province under Grant GA21A403, and Natural Science Foundation of Heilongjiang Province under Grant YQ2021E002.

Conflict of interest

All authors declare there is no conflict of interest.

References

1. Y. Xiao, X. Sun, Y. Guo, S. Li, Y. Zhang, Y. Wang, An improved gorilla troops optimizer based on lens opposition-based learning and adaptive β-Hill climbing for global optimization, *CMES-Comput. Model. Eng. Sci.*, **131** (2022), 815–850. https://doi.org/10.32604/cmes.2022.019198
2. Y. Xiao, X. Sun, Y. Guo, H. Cui, Y. Wang, J. Li, et al., An enhanced honey badger algorithm based on Lévy flight and refraction opposition-based learning for engineering design problems, *J. Intell. Fuzzy Syst.*, (2022), 1–24. https://doi.org/10.3233/JIFS-213206
3. Q. Liu, N. Li, H. Jia, Q. Qi, L. Abualigah, Y. Liu, A hybrid arithmetic optimization and golden sine algorithm for solving industrial engineering design problems, *Mathematics*, **10** (2022), 1567. https://doi.org/10.3390/math10091567

4. A. S. Sadiq, A. A. Dehkordi, S. Mirjalili, Q. V. Pham, Nonlinear marine predator algorithm: a cost-effective optimizer for fair power allocation in NOMA-VLC-B5G networks, *Expert Syst. Appl.*, **203** (2022), 117395. https://doi.org/10.1016/j.eswa.2022.117395
5. G. Hu, J. Zhong, B. Du, G. Wei, An enhanced hybrid arithmetic optimization algorithm for engineering applications, *Comput. Methods Appl. Mech. Eng.*, **394** (2022), 114901. https://doi.org/10.1016/j.cma.2022.114901
6. A. A. Dehkordi, A. S. Sadiq, S. Mirjalili, K. Z. Ghafoor, Nonlinear-based Chaotic harris hawks optimizer: algorithm and internet of vehicles application, *Appl. Soft Comput.*, **109** (2021), 107574. https://doi.org/10.1016/j.asoc.2021.107574
7. W. Zhao, L. Wang, S. Mirjalili, Artificial hummingbird algorithm: a new bio-inspired optimizer with its engineering applications, *Comput. Methods Appl. Mech. Eng.*, **388** (2022), 114194. https://doi.org/10.1016/j.cma.2021.114194
8. K. Sun, H. Jia, Y. Li, Z. Jiang, Hybrid improved slime mould algorithm with adaptive β hill climbing for numerical optimization, *J. Intell. Fuzzy Syst.*, **40** (2021), 1667–1679. https://doi.org/10.3233/jifs-201755
9. K. Zhong, G. Zhou, W. Deng, Y. Zhou, Q. Luo, MOMPA: multi-objective marine predator algorithm, *Comput. Methods Appl. Mech. Eng.*, **385** (2021), 114029. https://doi.org/10.1016/j.cma.2021.114029
10. Q. Fan, H. Huang, K. Yang, S. Zhang, L. Yao, Q. Xiong, A modified equilibrium optimizer using opposition-based learning and novel update rules, *Expert Syst. Appl.*, **170** (2021), 114575. https://doi.org/10.1016/j.eswa.2021.114575
11. L. Abualigah, A. Diabat, M. A. Elaziz, Improved slime mould algorithm by opposition-based learning and Levy flight distribution for global optimization and advances in real-world engineering problems, *J. Ambient Intell. Humanized Comput.*, (2021), https://doi.org/10.1007/s12652-021-03372-w
12. S. Wang, H. Jia, L. Abualigah, Q. Liu, R. Zheng, An improved hybrid aquila optimizer and harris hawks algorithm for solving industrial engineering optimization problems, *Processes*, **9** (2021), 1551. https://doi.org/10.3390/pr9091551
13. L. Abualigah, A. A. Ewees, M. A. A. Al-qaness, M. A. Elaziz, D. Yousri, R. A. Ibrahim, et al., Boosting arithmetic optimization algorithm by sine cosine algorithm and levy flight distribution for solving engineering optimization problems, *Neural Comput. Appl.*, **34** (2022), 8823–8852. https://doi.org/10.1007/s00521-022-06906-1
14. Y. Zhang, Y. Wang, L. Tao, Y. Yan, J. Zhao, Z. Gao, Self-adaptive classification learning hybrid JAYA and Rao-1 algorithm for large-scale numerical and engineering problems, *Eng. Appl. Artif. Intell.*, **114** (2022), 105069. https://doi.org/10.1016/j.engappai.2022.105069
15. D. Wu, H. Jia, L. Abualigah, Z. Xing, R. Zheng, H. Wang, et al., Enhance teaching-learning-based optimization for tsallis-entropy-based feature selection classification approach, *Processes*, **10** (2022), 360. https://doi.org/10.3390/pr10020360
16. H. Jia, W. Zhang, R. Zheng, S. Wang, X. Leng, N. Cao, Ensemble mutation slime mould algorithm with restart mechanism for feature selection, *Int. J. Intell. Syst.*, **37** (2021), 2335–2370. https://doi.org/10.1002/int.22776
17. H. Jia, K. Sun, Improved barnacles mating optimizer algorithm for feature selection and support vector machine optimization, *Pattern Anal. Appl.*, **24** (2021), 1249–1274. https://doi.org/10.1007/s10044-021-00985-x

18. C. Kumar, T. D. Raj, M. Premkumar, T. D. Raj, A new stochastic slime mould optimization algorithm for the estimation of solar photovoltaic cell parameters, *Optik*, **223** (2020), 165277. https://doi.org/10.1016/j.ijleo.2020.165277
19. Y. Zhang, Y. Wang, S. Li, F. Yao, L. Tao, Y. Yan, et al., An enhanced adaptive comprehensive learning hybrid algorithm of Rao-1 and JAYA algorithm for parameter extraction of photovoltaic models, *Math. Biosci. Eng.*, **19** (2022), 5610–5637. https://doi.org/10.3934/mbe.2022263
20. M. Eslami, E. Akbari, S. T. Seyed Sadr, B. F. Ibrahim, A novel hybrid algorithm based on rat swarm optimization and pattern search for parameter extraction of solar photovoltaic models, *Energy Sci. Eng.*, (2022). https://doi.org/10.1002/ese3.1160
21. J. Zhao, Y. Zhang, S. Li, Y. Wang, Y. Yan, Z. Gao, A chaotic self-adaptive JAYA algorithm for parameter extraction of photovoltaic models, *Math. Biosci. Eng.*, **19** (2022), 5638–5670. https://doi.org/10.3934/mbe.2022264
22. X. Bao, H. Jia, C. Lang, A novel hybrid harris hawks optimization for color image multilevel thresholding segmentation, *IEEE Access*, **7** (2019), 76529–76546. https://doi.org/10.1109/access.2019.2921545
23. S. Lin, H. Jia, L. Abualigah, M. Altalhi, Enhanced slime mould algorithm for multilevel thresholding image segmentation using entropy measures, *Entropy*, **23** (2021), 1700. https://doi.org/10.3390/e23121700
24. M. Abd Elaziz, D. Mohammadi, D. Oliva, K. Salimifard, Quantum marine predators algorithm for addressing multilevel image segmentation, *Appl. Soft Comput.*, **110** (2021), 107598. https://doi.org/10.1016/j.asoc.2021.107598
25. J. Yao, Y. Sha, Y. Chen, G. Zhang, X. Hu, G. Bai, et al., IHSSAO: An improved hybrid salp swarm algorithm and aquila optimizer for UAV path planning in complex terrain, *Appl. Sci.*, **12** (2022), 5634. https://doi.org/10.3390/app12115634
26. J. H. Holland, Genetic algorithms, *Sci. Am.*, **267** (1992), 66–72. https://doi.org/10.1038/scientificamerican0792-66
27. P. J. Angeline, Genetic programming: On the programming of computers by means of natural selection, *Biosystems*, **33** (1994), 69–73. https://doi.org/10.1016/0303-2647(94)90062-0
28. R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.*, **11** (1997), 341–359. https://doi.org/10.1023/A:1008202821328
29. H. G. Beyer, H. P. Schwefel, Evolution strategies-A comprehensive introduction, *Nat. Comput.*, **1** (2002), 3–52. https://doi.org/10.1023/A:1015059928466
30. D. Simon, Biogeography-based optimization, *IEEE Trans. Evol. Comput.*, **12** (2008), 702–713. https://doi.org/10.1109/TEVC.2008.919004
31. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, *Science*, **220** (1983), 671–680. https://doi.org/10.1126/science.220.4598.671
32. E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, *Inf. Sci.*, **179** (2009), 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
33. S. Mirjalili, S. M. Mirjalili, A. Hatamlou, Multi-Verse Optimizer: a nature-inspired algorithm for global optimization, *Neural Comput. Appl.*, **27** (2016), 495–513. https://doi.org/10.1007/s00521-015-1870-7
34. W. Zhao, L. Wang, Z. Zhang, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, *Knowl.-Based Syst.*, **163** (2019), 283–304. https://doi.org/10.1016/j.knosys.2018.08.030

35. A. Hatamlou, Black hole: a new heuristic optimization approach for data clustering, *Inf. Sci.*, **222** (2013), 175–184. https://doi.org/10.1016/j.ins.2012.08.023
36. S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, *Knowl.-Based Syst.*, **96** (2016), 120–133. https://doi.org/10.1016/j.knosys.2015.12.022
37. A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: thermal exchange optimization, *Adv. Eng. Software*, **110** (2017), 69–84. https://doi.org/10.1016/j.advengsoft.2017.03.014
38. L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A. H. Gandomi, The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Eng.*, **376** (2021), 113609. https://doi.org/10.1016/j.cma.2020.113609
39. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95 - International Conference on Neural Networks*, (1995), 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
40. M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE Comput. Intell. Mag.*, **1** (2006), 28–39. https://doi.org/10.1109/MCI.2006.329691
41. S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, *Neural Comput. Appl.*, **27** (2015), 1053–1073. https://doi.org/10.1007/s00521-015-1920-1
42. S. Mirjalili, The ant lion optimizer, *Adv. Eng. Software*, **83** (2015), 80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010
43. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Software*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
44. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software*, **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
45. S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, S. M. Mirjalili, Salp swarm algorithm: a bio-inspired optimizer for engineering design problems, *Adv. Eng. Software*, **114** (2017), 163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002
46. F. Glover, Tabu search—Part I, *ORSA J. Comput.*, **1** (1989), 190–206. https://doi.org/10.1287/ijoc.1.3.190
47. D. Manjarres, I. Landa-Torres, S. Gil-Lopez, J. Del Ser, M. N. Bilbao, S. Salcedo-Sanz, et al., A survey on applications of the harmony search algorithm, *Eng. Appl. Artif. Intell.*, **26** (2013), 1818–1831. https://doi.org/10.1016/j.engappai.2013.05.008
48. M. S. Gonçalves, R. H. Lopez, L. F. F. Miguel, Search group algorithm: a new metaheuristic method for the optimization of truss structures, *Comput. Struct.*, **153** (2015), 165–184. https://doi.org/10.1016/j.compstruc.2015.03.003
49. E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, *2007 IEEE Congr. Evol. Comput.*, (2007), 4661–4667. https://doi.org/10.1109/CEC.2007.4425083
50. R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, *Comput.Aided Des.*, **43** (2011), 303–315. https://doi.org/10.1016/j.cad.2010.12.015
51. S. Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, *Knowl.-Based Syst.*, **89** (2015), 228–249. https://doi.org/10.1016/j.knosys.2015.07.006
52. S. Li, H. Chen, M. Wang, A. A. Heidari, S. Mirjalili, Slime mould algorithm: a new method for stochastic optimization, *Future Gener. Comput. Syst.*, **111** (2020), 300–323. https://doi.org/10.1016/j.future.2020.03.055

53. S. Kaur, L. K. Awasthi, A. L. Sangal, G. Dhiman, Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization, *Eng. Appl. Artif. Intell.*, **90** (2020), 103541. https://doi.org/10.1016/j.engappai.2020.103541
54. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: algorithm and applications, *Future Gener. Comput. Syst.*, **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028
55. B. Abdollahzadeh, F. Soleimanian Gharehchopogh, S. Mirjalili, Artificial gorilla troops optimizer: a new nature‐inspired metaheuristic algorithm for global optimization problems, *Int. J. Intell. Syst.*, **36** (2021), 5887–5958. https://doi.org/10.1002/int.22535
56. H. Jia, X. Peng, C. Lang, Remora optimization algorithm, *Expert Syst. Appl.*, **185** (2021), 115665. https://doi.org/10.1016/j.eswa.2021.115665
57. Y. Yang, H. Chen, A. A. Heidari, A. H. Gandomi, Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts, *Expert Syst. Appl.*, **177** (2021), 114864. https://doi.org/10.1016/j.eswa.2021.114864
58. L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, A. H. Gandomi, Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer, *Expert Syst. Appl.*, **191** (2022), 116158. https://doi.org/10.1016/j.eswa.2021.116158
59. Y. Xiao, X. Sun, Y. Zhang, Y. Guo, Y. Wang, J. Li, An improved slime mould algorithm based on Tent chaotic mapping and nonlinear inertia weight, *Int. J. Innovative Comput. Inf. Control*, **17** (2021), 2151–2176. https://doi.org/10.24507/ijicic.17.06.2151
60. R. Zheng, H. Jia, L. Abualigah, Q. Liu, S. Wang, Deep ensemble of slime mold algorithm and arithmetic optimization algorithm for global optimization, *Processes*, **9** (2021), 1774. https://doi.org/10.3390/pr9101774
61. H. Jia, K. Sun, W. Zhang, X. Leng, An enhanced chimp optimization algorithm for continuous optimization domains, *Complex Intell. Syst.*, **8** (2022), 65–82. https://doi.org/10.1007/s40747-021-00346-5
62. A. S. Sadiq, A. A. Dehkordi, S. Mirjalili, J. Too, P. Pillai, Trustworthy and efficient routing algorithm for IoT-FinTech applications using non-linear Lévy brownian generalized normal distribution optimization, *IEEE Internet Things J.*, (2021), 1–16. https://doi.org/10.1109/jiot.2021.3109075
63. D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.*, **1** (1997), 67–82. https://doi.org/10.1109/4235.585893
64. S. Chakraborty, A. K. Saha, R. Chakraborty, M. Saha, S. Nama, HSWOA: an ensemble of hunger games search and whale optimization algorithm for global optimization, *Int. J. Intell. Syst.*, **37** (2022), 52–104. https://doi.org/10.1002/int.22617
65. P. Pirozmand, A. Javadpour, H. Nazarian, P. Pinto, S. Mirkamali, F. Ja'fari, GSAGA: A hybrid algorithm for task scheduling in cloud infrastructure, *J. Supercomput.*, (2022). https://doi.org/10.1007/s11227-022-04539-8
66. H. Abdel-Mawgoud, S. Kamel, A. A. A. El-Ela, F. Jurado, Optimal allocation of DG and capacitor in distribution networks using a novel hybrid MFO-SCA method, *Electr. Power Compon. Syst.*, **49** (2021), 259–275. https://doi.org/10.1080/15325008.2021.1943066
67. L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. A. Al-qaness, A. H. Gandomi, Aquila optimizer: a novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.*, **157** (2021), 107250. https://doi.org/10.1016/j.cie.2021.107250

68. B. Abdollahzadeh, F. S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems, *Comput. Ind. Eng.*, **158** (2021), 107408. https://doi.org/10.1016/j.cie.2021.107408
69. Z. Guo, B. Yang, Y. Han, T. He, P. He, X. Meng, et al., Optimal PID tuning of PLL for PV inverter based on aquila optimizer, *Front. Energy Res.*, **9** (2022), 812467. https://doi.org/10.3389/fenrg.2021.812467
70. M. R. Hussan, M. I. Sarwar, A. Sarwar, M. Tariq, S. Ahmad, A. Shah Noor Mohamed, et al., Aquila optimization based harmonic elimination in a modified H-bridge inverter, *Sustainability*, **14** (2022), 929. https://doi.org/10.3390/su14020929
71. G. Vashishtha, R. Kumar, Autocorrelation energy and aquila optimizer for MED filtering of sound signal to detect bearing defect in Francis turbine, *Meas. Sci. Technol.*, **33** (2021), 015006. https://doi.org/10.1088/1361-6501/ac2cf2
72. A. M. AlRassas, M. A. A. Al-qaness, A. A. Ewees, S. Ren, M. Abd Elaziz, R. Damaševičius, et al., Optimized ANFIS model using Aquila optimizer for oil production forecasting, *Processes*, **9** (2021), 1194. https://doi.org/10.3390/pr9071194
73. A. K. Khamees, A. Y. Abdelaziz, M. R. Eskaros, A. El-Shahat, M. A. Attia, Optimal power flow solution of wind-integrated power system using novel metaheuristic method, *Energies*, **14** (2021), 6117. https://doi.org/10.3390/en14196117
74. J. Zhao, Z. M. Gao, The heterogeneous Aquila optimization algorithm, *Math. Biosci. Eng.*, **19** (2022), 5867–5904. https://doi.org/10.3934/mbe.2022275
75. M. Kandan, A. Krishnamurthy, S. A. M. Selvi, M. Y. Sikkandar, M. A. Aboamer, T. Tamilvizhi, Quasi oppositional Aquila optimizer-based task scheduling approach in an IoT enabled cloud environment, *J. Supercomput.*, **78** (2022), 10176–10190. https://doi.org/10.1007/s11227-022-04311-y
76. X. Li, S. Mobayen, Optimal design of a PEMFC ‐ based combined cooling, heating and power system based on an improved version of Aquila optimizer, *Concurrency Comput. Pract. Exper.*, **34** (2022), e6976. https://doi.org/10.1002/cpe.6976
77. J. Zhao, Z. M. Gao, H. F. Chen, The simplified aquila optimization algorithm, *IEEE Access*, **10** (2022), 22487–22515. https://doi.org/10.1109/access.2022.3153727
78. S. Mahajan, L. Abualigah, A. K. Pandit, M. Altalhi, Hybrid Aquila optimizer with arithmetic optimization algorithm for global optimization tasks, *Soft Comput.*, **26** (2022), 4863–4881. https://doi.org/10.1007/s00500-022-06873-8
79. Y. Zhang, Y. Yan, J. Zhao, Z. Gao, AOAAO: The hybrid algorithm of arithmetic optimization algorithm with aquila optimizer, *IEEE Access*, **10** (2022), 10907–10933. https://doi.org/10.1109/access.2022.3144431
80. G. Vashishtha, S. Chauhan, A. Kumar, R. Kumar, An ameliorated African vulture optimization algorithm to diagnose the rolling bearing defects, *Meas. Sci. Technol.*, **33** (2022), 075013. https://doi.org/10.1088/1361-6501/ac656a
81. M. R. Kaloop, B. Roy, K. Chaurasia, S. M. Kim, H. M. Jang, J. W. Hu, et al., Shear strength estimation of reinforced concrete deep beams using a novel hybrid metaheuristic optimized SVR models, *Sustainability*, **14** (2022), 5238. https://doi.org/10.3390/su14095238
82. M. Manickam, R. Siva, S. Prabakeran, K. Geetha, V. Indumathi, T. Sethukarasi, Pulmonary disease diagnosis using African vulture optimized weighted support vector machine approach, *Int. J. Imaging Syst. Technol.*, **32** (2022), 843–856. https://doi.org/https://doi.org/10.1002/ima.22669

83. J. Fan, Y. Li, T. Wang, An improved African vultures optimization algorithm based on tent chaotic mapping and time-varying mechanism, *PLoS One*, **16** (2021), e0260725. https://doi.org/10.1371/journal.pone.0260725
84. H. R. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, (2005), 695–701. https://doi.org/10.1109/CIMCA.2005.1631345
85. N. A. Alawad, B. H. Abed-alguni, Discrete island-based cuckoo search with highly disruptive polynomial mutation and opposition-based learning strategy for scheduling of workflow applications in cloud environments, *Arabian J. Sci. Eng.*, **46** (2021), 3213–3233. https://doi.org/10.1007/s13369-020-05141-x
86. T. T. Nguyen, H. J. Wang, T. K. Dao, J. S. Pan, J. H. Liu, S. Weng, An improved slime mold algorithm and its application for optimal operation of cascade hydropower stations, *IEEE Access*, **8** (2020), 226754–226772. https://doi.org/10.1109/access.2020.3045975
87. Y. Zhang, Y. Wang, Y. Yan, J. Zhao, Z. Gao, LMRAOA: an improved arithmetic optimization algorithm with multi-leader and high-speed jumping based on opposition-based learning solving engineering and numerical problems, *Alexandria Eng. J.*, **61** (2022), 12367–12403. https://doi.org/10.1016/j.aej.2022.06.017
88. S. Wang, H. Jia, Q. Liu, R. Zheng, An improved hybrid Aquila optimizer and Harris Hawks optimization for global optimization, *Math. Biosci. Eng.*, **18** (2021), 7076–7109. https://doi.org/10.3934/mbe.2021352
89. Q. Fan, Z. Chen, W. Zhang, X. Fang, ESSAWOA: Enhanced whale optimization algorithm integrated with salp swarm algorithm for global optimization, *Eng. Comput.*, **38** (2022), 797–814. https://doi.org/10.1007/s00366-020-01189-3
90. F. Yu, Y. Li, B. Wei, X. Xu, Z. Zhao, The application of a novel OBL based on lens imaging principle in PSO, *Acta Electron. Sin.*, **42** (2014), 230–235. https://doi.org/10.3969/j.issn.0372-2112.2014.02.004
91. W. Long, J. Jiao, X. Liang, S. Cai, M. Xu, A random opposition-based learning grey wolf optimizer, *IEEE Access*, **7** (2019), 113810–113825. https://doi.org/10.1109/access.2019.2934994
92. H. T. Kahraman, H. Bakir, S. Duman, M. Katı, S. Aras, U. Guvenc, Dynamic FDB selection method and its application: modeling and optimizing of directional overcurrent relays coordination, *Appl. Intell.*, **52** (2022), 4873–4908. https://doi.org/10.1007/s10489-021-02629-3
93. H. T. Kahraman, S. Aras, E. Gedikli, Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms, *Knowl.-Based Syst.*, **190** (2020), 105169. https://doi.org/10.1016/j.knosys.2019.105169
94. S. Aras, E. Gedikli, H. T. Kahraman, A novel stochastic fractal search algorithm with fitness-distance balance for global numerical optimization, *Swarm Evol. Comput.*, **61** (2021), 100821. https://doi.org/10.1016/j.swevo.2020.100821
95. S. Duman, H. T. Kahraman, U. Guvenc, S. Aras, Development of a Lévy flight and FDB-based coyote optimization algorithm for global optimization and real-world ACOPF problems, *Soft Comput.*, **25** (2021), 6577–6617. https://doi.org/10.1007/s00500-021-05654-z

96. S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, *Inf. Sci.*, **180** (2010), 2044–2064. https://doi.org/10.1016/j.ins.2009.12.010
97. E. Theodorsson-Norheim, Friedman and Quade tests: basic computer program to perform nonparametric two-way analysis of variance and multiple comparisons on ranks of several related samples, *Comput. Biol. Med.*, **17** (1987), 85–99. https://doi.org/10.1016/0010-4825(87)90003-5
98. K. V. Price, N. H. Awad, M. Z. Ali, P. N. Suganthan, The 100-digit challenge: problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization. Technical Report Nanyang Technological University, Singapore, (2018).
99. C. A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Comput. Methods Appl. Mech. Eng.*, **191** (2002), 1245–1287. https://doi.org/10.1016/S0045-7825(01)00323-1

AIMS Press