Application of Improved Crown Porcupine Optimizer in UAV Path Planning Based on Dynamic Weighted JAYA-CPO Attack Strategy

Huanlong Zhang, Chenglin Guo, Denghui Zhai, Yanfeng Wang, Heng Liu, Fuguo Chen, and Dan Xu

***Abstract*—Unmanned aerial vehicle (UAV) path planning plays an important role in power systems. In order to address the challenge in UAV path planning, an improved crested porcupine optimizer (ICPO) combining the Cauchy inverse cumulative distribution function and JAYA algorithm is proposed in this paper. First, the traditional random initialization is replaced by sine chaotic mapping, making the initial population more evenly distributed in the search space and improving the quality of the initial solution. Since the global search ability of the crested porcupine optimizer (CPO) is limited, the Cauchy inverse cumulative distribution strategy is introduced. In addition, as CPO is prone to fall into local optima in later stages, a weighted JAYA-CPO attack strategy is proposed to balance the global exploration and local exploitation, thereby improving the algorithm's ability to escape from local optima. Finally, ICPO is compared with another 10 algorithms on the cec2017 and cec2020 test sets. The experimental results show that ICPO has excellent competitiveness and optimization performance. The ICPO algorithm is applied to the path planning problem of power inspection UAV and is compared with four algorithms. The results show that the algorithm can generate more feasible path trajectories across two terrains with varying complexity, demonstrating the effectiveness and significance of the ICPO algorithm for UAV power inspection path planning.**

***Index Terms*—UAV path planning, power system, Cauchy's inverse cumulative distribution function, JAYA algorithm, crested porcupine optimizer.**

I. Introduction

In recent years, driven by the rapid development of unmanned aerial vehicle (UAV) technology, the application of UAV in power inspections has increasingly become an indispensable part of modern power systems [1]–[3]. Power facilities, particularly high-voltage lines, substations and remote power facilities, are often located in hard-to-reach areas, making manual inspection time-consuming, costly, and associated with significant safety risks. The introduction of UAV provides an efficient, convenient and safe solution for power inspections [4]–[6]. However, power inspection scenes are often complex and changeable, especially over high-voltage power lines and related facilities, and various obstacles are encountered during flights, such as power towers, trees, buildings, etc., making drone path planning challenging. In order to ensure that UAV can complete inspection tasks efficiently and safely, accurate path planning is particularly important [7]–[10].

Traditional path planning algorithms, such as A* algorithm and rapidly-exploring random tree (RRT), perform well in simple environments, but are often difficult to adapt to changing environments and multi-objective inspection requirements in complex and dynamic power inspection tasks [11]–[13]. To solve these problems, researchers have applied swarm intelligence optimization algorithms to UAV path planning. By simulating the collective behavior of nature groups, algorithms such as ant colony optimization (ACO) algorithm [14], particle swarm optimization (PSO) algorithm [15], and gray wolf optimization (GWO) algorithm [16] can effectively solve path planning problems in complex environments.

Swarm intelligent optimization algorithms have the advantages of strong global search capability, fast convergence speed, and strong adaptability, which can find a better path planning scheme in complex environments [17], [18]. In recent years, the combination of swarm intelligence optimization algorithms and UAV path

Received: November 15, 2024
Accepted: September 1, 2025
Published Online: November 1, 2025
Huanlong Zhang, Chenglin Guo, and Yanfeng Wang (corresponding author) are with the College of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450000, China (e-mail: zzuli407@163.com; 1274693417@qq.com; yanfengwang@yeah.net).
Denghui Zhai is with China Electric Equipment Research Institute of Science and Technology Co., Ltd., Shanghai 200000, China (e-mail: 1912361424@qq.com).
Heng Liu is with Henan Pinggao Electric Co., Ltd., Pingdingshan 467001, China (e-mail: liuhengzzu@163.com).
Fuguo Chen is with Dongfang Electric (Hangzhou) Innovation Institute Co., Ltd., Hangzhou 310000, China (e-mail: 2295723959@qq.com).
Dan Xu is with XJ Electric Co., Ltd., Xuchang 461000, China (e-mail: 1244066842@qq.com)
DOI: 10.23919/PCMP.2024.000413

planning has become a research hotspot [19], [20]. These algorithms achieve global path optimization for UAV by simulating group intelligence behavior, overcoming the limitations of traditional algorithms and improving the efficiency and accuracy of path planning. Meanwhile, in order to avoid the algorithms falling into local optimal solutions, researchers have introduced some improvement algorithms, such as hybrid algorithms, multi-objective optimization algorithms and adaptive algorithms, to further enhance the global search performance. For example, reference [21] uses ant colony algorithm to initialize the search path and beetle antenna algorithm to update the particle coordinates of subsequent steps, and subsequently an algorithm based on the combination of ant colony algorithm and beetle antenna algorithm is proposed to solve the UAV path planning problem in mountainous terrain. However, the problem of path planning in complex three-dimensional space needs to be improved. Reference [22] proposes a new spherical vector-based particle swarm optimization (SPSO) algorithm to solve the path planning problem of UAV facing multiple threats in complex environments, but its optimization performance is not tested on different benchmark test functions. In [23], an improved GWO algorithm, GEDGWO, is proposed by combining GWO algorithm with Gaussian distribution strategy, and is applied to solve multi-UAV path planning problem. However, other improved algorithms have broad prospects in UAV path planning. Other improved versions of algorithms used in UAV path planning have also been proposed. Reference [24] proposes an improved hybrid algorithm, IHSSAO, based on salp swarm algorithm (SSA) and Aquila optimizer (AO) to solve the UAV path planning problem, but its performance needs be verified under more constraints. Reference [25] proposes multi-strategy cuckoo search algorithm based on reinforcement learning, which achieves satisfactory results on the UAV path planning problem. However, the established mathematical model does not consider important objectives and constraints such as meteorological factors.

Crested porcupine optimizer (CPO) is an emerging swarm intelligence optimization algorithm that solves the optimization problem by simulating the four defense mechanisms of crested porcupine (CP), i.e., visual, auditory, scent, and physical attacks [26]. CPO performs well in dealing with complex optimization problems, with the advantages of strong global search ability, fast convergence, and high adaptability. However, conventional CPO may still fall into local optimal solutions when facing path planning problems with high dimensionality and complex constraints, affecting its global search performance.

To solve these problems, this paper proposes an improved version of CPO algorithm (ICPO). The random initialization is replaced with sine chaotic mapping initialization in the initialization phase, while the Cauchy inverse cumulative distribution function is added in the first defense phase of the CPO algorithm. JAYA algorithm is integrated into the fourth defense phase of CPO to improve the overall optimization performance and adaptability of the algorithm. The ICPO algorithm is applied to terrain models with different complexity levels, and relevant constraints such as optimality, safety and feasibility are included. The performances of different algorithms in various complex environments are compared and analyzed to provide new concepts and methods for the research and application of UAV path planning. The main research work in this paper are summarized as follows.

1) Sine chaotic mapping initialization is used instead of random initialization, it makes the population distribution more uniform and improves the quality of the initial solution.

2) Incorporate the Cauchy inverse cumulative distribution function into the first defense stage of the CPO algorithm to enhance the global search ability of the algorithm and improve the convergence performance of the algorithm.

3) Incorporate the JAYA algorithm into the fourth defense stage of the CPO to coordinate the global search and local development of the algorithm and enhance its adaptability and robustness.

4) The effectiveness of ICPO is demonstrated by comparing the multiple performance test indexes under cec2017 and cec2020 test functions with other swarm intelligence optimization algorithms.

5) The ICPO algorithm proposed in this paper is applied to the path planning of power inspection UAV, and is compared with CPO, an improved Tasmanian devil optimization algorithm based on sine-cosine strategy with dynamic weighting factors (NTDO), dual fitness particle swarm optimization algorithm(DFPSO), a novel sparrow search algorithm with integrates spawning strategy (NSSA), backtracking search algorithm driven by generalized mean position (GMPBSA) and other algorithms in terms of path length, smoothness and optimization effect, thereby fully demonstrating its superior performance and excellent path planning ability in power inspection tasks.

The remaining structure of this paper is as follows. Section II introduces the core concepts and basic principles of the conventional CPO algorithm, while Section III proposes three strategies to improve the CPO algorithm (i.e., ICPO). Section IV verifies the strong optimization performance of ICPO through the cec2017 and cec2020 test functions, and Section V applies the ICPO algorithm to the power inspection UAV path planning task. Finally, Section VI summarizes the paper and recommends future work.

II. Crested Porcupine Optimizer

CPs are larges rodent found across forests, deserts, rocky outcrops, and mountain slopes on every continent except Antarctica [27]–[29]. They rank third in size among rodents, following capybaras and beavers. Most CPs have a dark brown or black stiff coat, occasionally dotted with white spots. The most striking feature is their broad feather cover, which extends to the head, neck, back and tail, as shown in Fig. 1. As nocturnal herbivores, CPs feed on a variety of plants, including shrubs and leaves [30].

Fig. 1. CP and CP defends itself from a leopard.

CPO simulates various defensive behaviors of CPs, including sight, sound, smell, and physical aggression as ranked from least aggressive to most aggressive [31]–[34]. Based on these four defense strategies, CPO is proposed, consisting of two phases (exploration and exploitation). In CPO, the search space is visualized as shown in Fig. 2, which is divided into four distinct zones, each simulating the defense strategy of CPs. The first zone A represents the first defense zone, and the first defense strategy is implemented when the CP is away from the predator. The second zone B represents the second defense zone, which is activated when the predator is unaffected by the first defense mechanism and continues approaching CP. The third zone C represents the third defense zone, and the third defense strategy will be activated when the predator is still approaching CP despite the first and second defense mechanisms. The last zone D represents the last defense zone where, after all previous defense mechanisms have failed, and the CP will take aggressive measures to incapacitate and possibly kill the predator in order to protect itself.

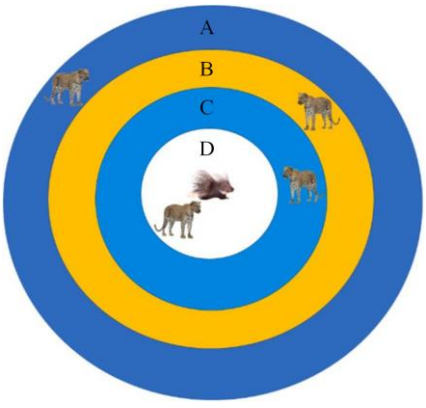


Fig. 2. Four defense mechanisms of CPs.

A. Population Initialization

CPO begins the search process with an initial set of individuals (candidate solutions), as:

$$\boldsymbol{X}_i = \boldsymbol{L} + \boldsymbol{r} \times \boldsymbol{U} - \boldsymbol{L},\ i = 1,\ 2,\ \cdots,\ N' \tag{1}$$

where $\boldsymbol{X}_i$ is the ith candidate solution in the search space; $\boldsymbol{L}$ and $\boldsymbol{U}$ are the lower and upper limits of the search range, respectively; $\boldsymbol{r}$ is a random number between 0 and 1; and N' represents the number of individuals, i.e., N' is population size.

B. Cyclic Population Reduction Technique

The cyclic population reduction (CPR) technique, in addition to accelerating the rate of convergence, can also maintain population diversity. This strategy simulates the idea that not all CPs activate defense mechanisms, but only those that are threatened. The mathematical model of cyclic population size reduction is as follows:

$$N = N_{\min} + \left(N' - N_{\min}\right) \times \left(1 - \frac{g\% \frac{T_{\max}}{T}}{\frac{T_{\max}}{T}}\right) \tag{2}$$

where T is the variable that determines the number of cycles; g is the current function evaluation; $T_{\max}$ is the maximum number evaluated by the function; "%" represents the remainder or modular operator; and $N_{\min}$ is the minimum number of individuals in the newly generated population, so the population size cannot be less than $N_{\min}$. As the number of current function evaluations increases, the population size gradually decreases, reaching as low as 40, which marks the first cycle. Subsequently, the population size is maximized again and then gradually declines until the end of the optimization process, representing the second and final period, as T is set to 2. Overall, the population size initially reaches its maximum and then progressively shrinks, eventually approaching $N_{\min}$.

C. Exploration Phase

1) First Defense Strategy (Vision)

When the CP senses a predator, it raises and waves the quill to make a deeper impression on the predator. In this situation, the predator has two options: approach the CP or maintain a distance from it. In the first option, the predator approaches the CP, reducing the distance between them. This promotes exploration of the space between the predator and CP, accelerating convergence. In the second option, the predator moves away from the CP, maximizing the distance. This encourages exploration of more distant regions, potentially uncovering unexplored areas that may contain the desired solutions. Using a normal distribution to generate random values, these options are mathematically simulated. If these random values are less than 1 or greater than −1, approaching the CP is encouraged. Otherwise, predators

stays away from CP. Typically, this behavior is mathematically simulated as follows:

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \tau_1 \times \left| 2 \times \tau_2 \times \boldsymbol{x}_{\text{cp}}^t - \boldsymbol{y}_i^t \right| \tag{3}$$

where $\boldsymbol{x}_i^t$ is the position of the ith individual at iteration t; $\boldsymbol{x}_{\text{cp}}^t$ is the best solution of the evaluation function t; $\boldsymbol{y}_i^t$ is the vector generated between the current CP and a randomly selected CP from the population to represent the position of the predator when iterating t; τ_1 is a random number based on a normal distribution; and τ_2 is a random value in the interval [0, 1]. The mathematical formula for generating $\boldsymbol{y}_i^t$ is:

$$\boldsymbol{y}_i^t = \frac{\boldsymbol{x}_i^t + \boldsymbol{x}_c^t}{2} \tag{4}$$

where c is a random number in [1, N].

2) Second Defense Strategy (Sounds)

In this strategy, the CP uses sound methods to make noise and threaten predators. When predators approach the CP, the sound becomes louder. To simulate this behavior mathematically, the following formula is proposed:

$$\boldsymbol{x}_i^{t+1} = \left(1 - \boldsymbol{U}_1\right) \times \boldsymbol{x}_i^t + \boldsymbol{U}_1 \times \left(\boldsymbol{y} + \tau_3 \times \left(\boldsymbol{x}_{c1}^t - \boldsymbol{x}_{c2}^t \right) \right) \tag{5}$$

where the subscripts $c1$ and $c2$ are two random integers in [1, N]; and τ_3 is a random value in the interval [0, 1]. In (5), we attempt to simulate the influence of the sound on the distance between the predators and the CPs. The influence of the sound is categorized into three cases: the first case is that the sound is weak and hence the predators continue moving toward the CP; the second case is that the sound is slightly loud and the predators might stand in their position without moving; lastly, the sound is loud and the predators move away due to fear. The three cases may not be exactly applied given the limitations of each rule. The predators might completely move toward the CP even if the noise is so loud or might go away or stand in place even when the sound is low. Therefore, we use a binary vector $\boldsymbol{U}_1$ including 0 and 1 generated randomly to cover all the possible probabilities. When the cells in this vector include 0, the second case is achieved because this means that the sound is slightly loud, and hence the predators stand without moving toward the CP. However, when the cells include 1, this means that the predators might move towards or far from the CP; this covers the first and third cases. The tradeoff between the first and third cases is achieved as follows: the vector $\boldsymbol{y}$ represents the predator's position, which is between the current CP and a CP solution selected randomly from the population. After computing $\boldsymbol{y}$, we compute the difference between two solutions selected randomly from the current population to determine whether the predator will move toward or away from the CP. If this difference is smaller than 0 and the predator's position is greater than the current CP's position, then the predators will move toward the CP; otherwise, the predator moves away due to fear. On the other contrary, if this difference is greater than 0 and the predator's position is smaller than the current CP's position, then the predators will move toward the CP; otherwise, the predator moves away due to fear.

D. Exploitation Phase

1) Third Defense Strategy (Smell)

In this strategy, the CP secretes a foul odor that spreads in its surrounding area to prevent predators from approaching it. To model this behavior mathematically, the following equation is proposed:

$$\boldsymbol{x}_i^{t+1} = \left(1 - \boldsymbol{U}_1\right) \times \boldsymbol{x}_i^t + \boldsymbol{U}_1 \times \left(\boldsymbol{x}_{c1}^t + S_i^t \times \left(\boldsymbol{x}_{c2}^t - \boldsymbol{x}_{c3}^t \right) - \tau_3 \times \boldsymbol{\delta} \times \gamma_t \times S_i^t \right) \tag{6}$$

where the subscript $c3$ is a random number in [1, N]; $\boldsymbol{\delta}$ is a parameter used to control the direction of the search and is defined by (7); $\boldsymbol{x}_i^t$ is the position of the ith individual at iteration t; γ_t is a defense factor defined by (8); S_i^t is an odor diffusion factor defined by (9); and the $\boldsymbol{U}_1$ vector is used to simulate the strategy in the following possible three scenarios:

1) When $\boldsymbol{U}_1$ equals 0, the CP ceases odor diffusion because the predator, fearing the CP, stops moving, causing the distance between them to remain constant;

2) When $\boldsymbol{U}_1$ is equal to 1, the CP emits a strong odor because the predator is nearby;

3) When $\boldsymbol{U}_1$ takes a value between 0 and 1, the predator maintains a safe distance from the CP, so extensively odor emission is unnecessary.

$$\boldsymbol{\delta} = \begin{cases} +1, & \text{if } \boldsymbol{e}_{\text{rand}} \leqslant 0.5 \\ -1, & \text{else} \end{cases} \tag{7}$$

$$\gamma_t = 2 \times \boldsymbol{e}_{\text{rand}} \times \left(1 - \frac{t}{t_{\max}} \right)^{\frac{t}{t_{\max}}} \tag{8}$$

$$S_i^t = \exp\left(\frac{f(x_i^t)}{\sum_{k=1}^{N} f(x_k^t) + \epsilon} \right) \tag{9}$$

where $f(x_i^t)$ denotes the value of the objective function for the ith individual at iteration t; $f(x_k^t)$ denotes the value of the objective function for the kth individual at iteration t; ϵ is a small value that avoids division by zero; $\boldsymbol{e}_{\text{rand}}$ is a vector that includes randomly generated values between 0 and 1; N is the population size; t is the number of current iterations; and $t_{\max}$ is the maximum number of iterations.

2) Fourth Defense Strategy (Physical Aggression)

The final strategy is a physical attack. When a predator approaches and strikes with its short, thick feathers, the CP responds with a physical counterattack. In this

process, the two bodies come into close contact, resembling a one-dimensional inelastic collision. To express the physical attack behavior, the following mathematical formula is proposed:

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_{\mathrm{CP}}^t + \left(\alpha\left(1-\tau_4\right)+\tau_4\right)\times\left(\boldsymbol{\delta}\times\boldsymbol{x}_{\mathrm{CP}}^t - \boldsymbol{x}_i^t\right) - \tau_5\times\boldsymbol{\delta}\times\gamma_t\times\boldsymbol{F}_i^t \tag{10}$$

where $\boldsymbol{x}_{\mathrm{CP}}^t$ is the best-obtained solution and represents the CP; α is the convergence rate factor discussed later in the parameter setting section; τ_4 and τ_5 are the random value in the interval [0, 1]; and $\boldsymbol{F}_i^t$ is the average force of CP affecting the ith predator, it is provided by the inelastic collision law [35] and defined by:

$$\begin{cases} \boldsymbol{F}_i^t = \boldsymbol{\tau}_6 \times \dfrac{m_i \times (\boldsymbol{v}_i^{t+1} - \boldsymbol{v}_i^t)}{\Delta t} \\ m_i = \dfrac{f(\boldsymbol{x}_i^t)}{\mathrm{e}^{\sum_{k=1}^{N} f(\boldsymbol{x}_k^t)+\epsilon}} \\ \boldsymbol{v}_i^t = \boldsymbol{x}_i^t \\ \boldsymbol{v}_i^{t+1} = \boldsymbol{x}_r^t \end{cases} \tag{11}$$

where m_i is the mass of the ith individual (predator) at iteration t; $f(\cdot)$ denotes the objective function; $\boldsymbol{v}_i^{t+1}$ is the final velocity of the ith individual at the next iteration t+1 and is assigned based on the selection of a random solution from the current totality; $\boldsymbol{v}_i^t$ is the initial velocity of the ith individual at iteration t; Δt is the number of the current iteration; and $\boldsymbol{\tau}_6$ is a vector including random values generated between 0 and 1.

In (11), the average force of the CP is calculated based on dividing the numerator by the current iteration, which increases linearly during the optimization process. As a result, the effect of the average force of the CP is gradually minimized. In fact, small values of this factor do not harm the performance of the CPO, since they contribute little to exploring alternative regions around the current best solution. Therefore, the numerator is removed and only the denominator is retained as shown in (12). This approach helps to create a wide range of values within the search space, enabling a more comprehensive examination of the regions around the current best solution.

$$\boldsymbol{F}_i^t = \boldsymbol{\tau}_6 \times m_i \times \left(\boldsymbol{v}_i^{t+1} - \boldsymbol{v}_i^t\right) \tag{12}$$

III. Improved Crested Porcupine Optimizer

A. Sine Chaotic Mapping Strategy

Random initialization generates solutions randomly within the feasible space to cover a wide area and increase the chances of finding a global optimum [36]. However, it may result in uneven distribution, affecting algorithm efficiency and solution diversity. To address this issue, this paper replaces random initialization with sine chaotic mapping to enhance global search by avoiding local optima through its inherent randomness and unpredictability [37]. Additionally, sine chaotic mapping improves path selection, reduces collision risk, and ensures efficiency and safety in UAV path planning [38]. The two initialization methods are compared in Fig. 3. The expression of sine chaotic mapping is given as:

$$x_{i+1} = \frac{a}{4}\sin\left(\pi x_i\right) \tag{13}$$

where a is the control parameter, and is typically 4, while the chaotic orbit state values range from (0, 1).

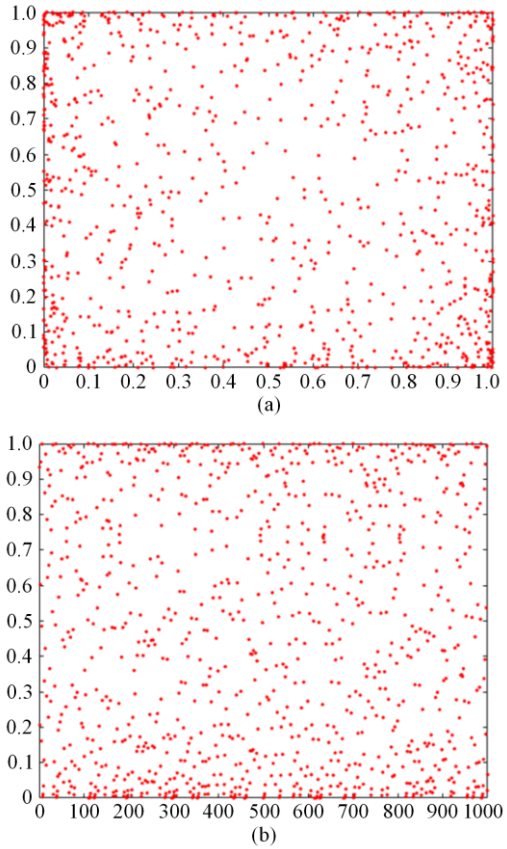


Fig. 3. Initialization method. (a) Random initialization. (b) Initialization of the sine chaotic mapping.

B. Cauchy Inverse Cumulative Distribution Function Strategy

The Cauchy inverse cumulative distribution function is a mapping from the probability values to the corresponding observations in the Cauchy distribution. The probability density function of the Cauchy distribution is defined as:

$$f(x;x_0,\gamma) = \frac{1}{\pi\gamma\left[1+\left(\dfrac{x-x_0}{\gamma}\right)^2\right]} \tag{14}$$

where x is a random variable that can take on any real number; x_0 is the positional parameter, which determines the central position of the distribution; and γ is the scale parameter, which determines the brightness of

the distribution. The characteristic of this distribution is that both its mean and variance are undefined, unlike other common distributions such as the normal distribution. The Cauchy cumulative distribution function (CDF) describes the probability that a random variable is less than or equal to a certain value. The CDF expression of the Cauchy distribution is:

$$F(x,x_0,\gamma)=\frac{1}{2}+\frac{1}{\pi}\arctan\left(\frac{x-x_0}{\gamma}\right) \tag{15}$$

The Cauchy inverse cumulative distribution function (ICDF) is an inverse function of CDF used to determine the value of a random variable corresponding to a probability value. ICDF of the Cauchy distribution is:

$$F^{-1}(p,x_0,\gamma)=x_0+\gamma\tan\left[\pi\left(p-\frac{1}{2}\right)\right] \tag{16}$$

where p is a probability value between 0 and 1. The position parameter x_0 is assigned to 0, which is consistent with the standard Cauchy distribution. And the scale parameter γ is reduced by 100 times to 0.01, which can improve the convergence speed of the algorithm through reasonable variation step size. The formula for incorporating the inverse Cauchy cumulative distribution function into the first defense phase of the CPO is as follows:

$$\boldsymbol{x}_i^{t+1}=\boldsymbol{x}_i^t+\tau_1\times\left|2\tau_2\boldsymbol{x}_{\mathrm{CP}}^t-\boldsymbol{y}_i^t\right|\times 0.01\tan\left(\pi\left(p-1/2\right)\right) \tag{17}$$

ICDF can be used to increase the exploratory nature of the algorithm by generating random perturbations or redirecting the search. Specifically, the long-tailed nature of the Cauchy distribution makes it capable of generating larger perturbations, which helps the algorithm to escape the local optimal solution in the search space [39] and explore the potential global optimal solution more extensively. Therefore, the inclusion of ICDF in the first defense stage of the CPO algorithm can enhance its global search capability and improve the quality of its convergence and solution, especially when dealing with complex, multimodal problems.

C. Weighted JAYA-CPO Attack Strategy

1) JAYA Algorithm

The core idea of the JAYA algorithm is to approach the optimal solution and avoid the worst solution. In each iteration, the individual moves closer to the excellent individual while distancing itself from the poor individual, thereby improving the solution quality [40]. This strategy of "seeking advantages and avoiding disadvantages" enables the algorithm to gradually approach the optimal solution.

The implementation process of the JAYA algorithm is relatively simple and easy to understand. Compared with other meta-heuristic algorithms, JAYA requires adjusting only the parameters (such as random numbers) of the iterative process for specific problems, thereby avoiding the complexity and implementation difficulties associated with tuning numerous parameters [41]. In the JAYA algorithm, the new solution is obtained by the iterative evolution of an equation while the individual is close to the excellent individual and away from the poor individual. This equation controls the updating direction and step size of the individual, so that the individual can gradually approach the optimal solution. The specific update method is to control the scale size through random numbers to realize the movement of individuals, shown as:

$$X'_{ij}=X_{ij}+r_{1j}\left(X_{bj}-\left|X_{ij}\right|\right)-r_{2j}\left(X_{wj}-\left|X_{ij}\right|\right) \tag{18}$$

where $r_{1j}\left(X_{bj}-\left|X_{ij}\right|\right)$ indicates the tendency of the solution to move towards the optimal solution; $-r_{2j}\left(X_{wj}-\left|X_{ij}\right|\right)$ indicates the tendency of the solution to move away from the worst solution; $X_{\mathrm{b}j}$ and $X_{\mathrm{w}j}$ represent the value of the jth dimension variable in the optimal and worst solution of the population respectively. If the objective function $f(x_{ij})$ corresponding to X'_{ij} is better, X'_{ij} is accepted and the original solution is replaced; otherwise, the original solution is maintained.

2) Weight Factor

Weight factor β regulates the algorithm's tradeoff between global search (provided by JAYA) and local search (provided by CPO's fourth defense strategy). In the early iteration stage of the algorithm, a larger β value helps the algorithm to jump out of the local optimal solution and explore a wider solution space [42]. As the algorithm gradually approaches the potential optimal solution, smaller β values can enhance local search, refine the current solution, and improve the accuracy of the solution. In addition, the dynamic adjustment mechanism of β makes the algorithm more adaptive and robust. By monitoring the performance of the algorithm and adjusting the β value accordingly, the algorithm is able to optimize its search behavior according to the characteristics of different problems and the dynamic changes in the search process. This flexibility not only improves the algorithm's performance on a variety of problems, but also enhances its stability in the face of complex search environments. The formula for β is given as:

$$\beta(t)=\beta_{\max}-\left(\frac{t}{T}\right)^{\alpha}\times\left(\beta_{\max}-\beta_{\min}\right) \tag{19}$$

where $\beta_{\max}$ and $\beta_{\min}$ are random numbers between 0 and 1, after extensive testing, the best results are obtained when $\beta_{\max}=0.9$ and $\beta_{\min}=0.1$.

In (19), an appropriate selection of α can balance the global search and local exploitation so that the algorithm can effectively explore the solution space across

different stages [43]. When $\alpha = 1$, the weight factor β decreases linearly, limiting the algorithm's flexibility in adjusting the weights between global and local search across different stages. This may lead to premature convergence or entrapment in a local optimum, while overlooking potentially better solutions. when $\alpha = 1.5$ and $\alpha = 2$, at the beginning of the iterations, the algorithm converges more slowly, allowing the individuals to move in larger steps and thus better explore the solution space to approach the global optimum; in the middle and late iterations, convergence accelerates, individuals take smaller steps, and the search for the optimal solution becomes more precise. When $\alpha = 2.5$ and $\alpha = 3$, the algorithm converges too slowly in the early stage, leading to excessive exploration of the solution space, which delays the discovery of the global optimum and increases runtime; and in contrast, in the middle and late iterations, convergence becomes too rapid, which may cause the algorithm to fall into a local optimum prematurely, miss the global solution, and restrict its search capability. Therefore, this paper chooses a more suitable value between $\alpha = 1.5$ and $\alpha = 2$. Compared with $\alpha = 1.5$, setting $\alpha = 2$ better reflects the actual situation and achieves a balance between exploration and accuracy at different stages. The convergence curve for α values is shown in Fig. 4.

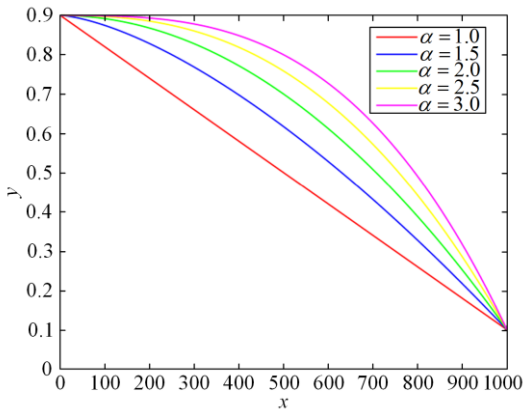


Fig. 4. Value curve of α.

In summary, the weight factor β plays a crucial role in fusion algorithms, as it regulates the balance between exploration and exploitation and directly affects convergence speed and overall performance.

3) Fusing the JAYA Algorithm into the Fourth Defense Strategy

The JAYA algorithm is combined with the fourth defense strategy of the CPO algorithm to leverage the strengths of both methods, enhancing the overall search efficiency and the quality of reconciliation. First, JAYA is a parameter-free optimization algorithm that guides the search process by identifying the best and worst solutions to the current solution set. This method helps the algorithm to avoid the local optimal solution and improve the global search ability. By introducing the JAYA algorithm, fusion strategy can make the algorithm search effectively in a wider solution space and increase the probability of finding the global optimal solution. Second, CPO's fourth defense strategy simulates the defense behavior of the CP and focuses on local search. This allows the algorithm to perform a more precise search near the optimal solution, improving solution the accuracy. Integrating the two allows further local development when needed, thereby achieving better results in terms of solution quality. In addition, the introduction of the weight factor β enables the algorithm to dynamically balance between global search and local search. By adjusting the value of β at different search stages, the algorithm can flexibly adjust its search behavior according to the current demand and the nature of the problem. This dynamic adjustment mechanism improves the adaptability and robustness of the algorithm, allowing it to deal with various kinds of optimization problems effectively. The specific fusion formula is as follows:

$$\boldsymbol{X}_i^{t+1} = \beta\left[\boldsymbol{x}_i^t + r_{1j}\left(X_{\mathrm{bj}} - \left|\boldsymbol{x}_i^t\right|\right) - r_{2j}\left(X_{\mathrm{wj}} - \left|\boldsymbol{x}_i^t\right|\right)\right] + (1-\beta)\boldsymbol{x}_i^{t+1} \quad (20)$$

where r_{1j} and r_{2j} are random values in [0, 1]; and $\boldsymbol{X}_i^{t+1}$ is the defense position calculated by individual i according to the fourth defense strategy of the CPO.

In summary, the mathematical formula of the fourth defense strategy, combining JAYA and CPO, leverages the strengths of both methods to enhance global search capability, local search efficiency, and overall adaptability and robustness. This fusion strategy represents an important trend in optimization algorithm design, contributing to the development of more efficient and robust optimization tools.

D. Complexity Analysis of ICPO

Similar to other algorithms, the improved ICPO in this paper mainly depends on the initialization, fitness evaluation and algorithm update of the Coronus optimizer. These three processes are related to population number (N), maximum number of iterations ($T_{\max}$), and dimension (D). The termination condition of ICPO is based on the maximum number of function evolution rather than the maximum number of iterations. As population size has no effect on time complexity, its time complexity depends on two factors: the maximum number of iterations of the function ($T_{\max}$) and the number of dimensions (D). In general, for the evaluation of all functions, the time complexity of ICPO is: $O(T_{\max} \times D)$.

IV. Simulation Experiment and Result Analysis

To verify the performance of ICPO, optimization simulation experiments for solving the UAV inspection path planning problem are conducted and compared

with NTDO [44], DFPSO [45], NSSA [46], GMPBSA [47], whale optimization algorithm (WOA) [48], dung beetle optimizer (DBO) [49], coati optimization algorithm (COA) [50], GWO [51], and butterfly optimization algorithm (BOA) [52]. The experimental environment is Windows 10 64-bit operating system with Intel(R) Core(TM) i3-8100 CPU @3.60GHz 16GB MATLAB R2022a. In this paper, 29 (F1, and F3–F30) test functions of cec2017 (F2 has been deleted in the official statement) [53] and 10 test functions of cec2020 [54] are selected to validate the effectiveness of the improved ICPO algorithm. Due to the inherent randomness and instability of the algorithms, and to ensure the authenticity and fairness of the experiments, each algorithm is independently tested 30 times for each type of experiments, while the corresponding optimal value, standard deviation and average value are recorded [55]. In this paper, the average values are analyzed as it reflects the overall performance of the algorithm in multiple runs and provides reliable results. The maximum number of iterations is 500 times, the maximum number of iterations of the algorithm is 1000, and the number of populations N is set to 30. The parameter settings of each algorithm are shown in Table I.

TABLE I
PARAMETER SETTINGS OF DIFFERENT SWARM INTELLIGENT ALGORITHMS

Algorithm	Main parameters settings	Value
ICPO	Population size	30
	Maximum iterations	500 or 1000
	Convergence rate factor	0.1
	T_f	0.5
	T	2
CPO	Population size	30
	Maximum iterations	500 or 1000
	Convergence rate factor	0.1
	T_f	0.5
	T	2
NTDO	Population size	30
	Maximum iterations	500 or 1000
	random value	{1, 2}
	$r2$	{0, 2π}
	$r3$	{0, ∞}
DFPSO	Population size	30
	Maximum iterations	500 or 1000
	$r1$	[0, 1]
	$r2$	[0, 1]
NSSA	Population size	30
	Maximum iterations	500 or 1000
	Proportion of discoverers	0.2
	Safety limit	0.8
	Safety value	0.2
GMPBSA	Population size	30
	Maximum iterations	500 or 1000
	D	1
WOA	Convergence constant	Decreases linearly from 2 to 0
	Spiral factor	1
	Population size	30
DBO	deflection factor	(0, 0.2]
	Natural coefficient	1 or –1
	constant value	[0, 1]
COA	Population size	30
	Maximum iterations	1000
GWO	Convergence constant	Decreases linearly from 2 to 0
	Population size	30
BOA	Power exponent	0.1
	Sensory modality	0.01
	p	0.8

A. CEC2017 Test Functions

As previously described, this paper uses 29 test functions of cec2017, of which F1 and F3 are single-peak functions, F4–F10 are multi-peak functions, F11–F20 are hybrid functions, and F21–F30 are composite functions, as shown in Table II. The test dimensions include 2/10/30/50/100 [56], and the test function with the dimension of 10 is chosen. The value range of all test functions is [–100, 100], and $f_{\min}$ denotes the optimal solution of the function.

TABLE II
EXPERIMENTAL DATA UNDER DIFFERENT IMPROVED ALGORITHMS (CEC2017)

		ICPO	CPO	NTDO	DFPSO	NSSA	GMPBSA
F1	Minimum	1.1036×10^{2}	1.2317×10^{2}	1.0624×10^{2}	2.7337×10^{5}	1.0158×10^{2}	1.0984×10^{2}
	Standard	3.2459×10^{2}	4.6785×10^{2}	4.3801×10^{3}	1.2174×10^{6}	3.8124×10^{3}	1.2995×10^{3}
	Average	$\mathbf{4.0390\times10^{2}}$	4.9653×10^{2}	3.1476×10^{3}	1.8046×10^{6}	4.7608×10^{3}	1.7903×10^{3}
F3	Minimum	3.0548×10^{2}	3.0021×10^{2}	4.3222×10^{2}	3.2538×10^{2}	3.2512×10^{2}	5.6821×10^{2}
	Standard	9.4039×10^{1}	3.9081×10^{1}	5.0337×10^{2}	1.2139×10^{2}	3.5633×10^{2}	9.0801×10^{2}
	Average	3.8912×10^{2}	$\mathbf{3.2515\times10^{2}}$	1.1974×10^{3}	4.6997×10^{2}	6.8410×10^{2}	1.8446×10^{3}
F4	Minimum	4.0181×10^{2}	4.0242×10^{2}	4.0004×10^{2}	4.0357×10^{2}	4.0000×10^{2}	4.0384×10^{2}
	Standard	8.8481×10^{-1}	7.4643×10^{-1}	1.3294×10^{1}	1.0497×10^{1}	1.1299×10^{1}	5.4682×10^{-1}
	Average	$\mathbf{4.0388\times10^{2}}$	4.0442×10^{2}	4.0769×10^{2}	4.0892×10^{2}	4.0747×10^{2}	4.0488×10^{2}
F5	Minimum	5.0306×10^{2}	5.0855×10^{2}	5.3030×10^{2}	5.1287×10^{2}	5.1194×10^{2}	5.1382×10^{2}
	Standard	3.1127×10^{0}	4.4699×10^{0}	2.8475×10^{1}	9.2224×10^{0}	1.5046×10^{1}	5.0496×10^{0}
	Average	$\mathbf{5.1098\times10^{2}}$	5.1804×10^{2}	5.9572×10^{2}	5.2954×10^{2}	5.3147×10^{2}	5.2071×10^{2}
F6	Minimum	6.0000×10^{2}	6.0000×10^{2}	6.2836×10^{2}	6.0041×10^{2}	6.0002×10^{2}	6.0001×10^{2}
	Standard	7.6372×10^{-3}	2.1766×10^{-3}	7.1246×10^{0}	1.8021×10^{0}	1.8389×10^{1}	2.5942×10^{-2}
	Average	6.0001×10^{2}	$\mathbf{6.0000\times10^{2}}$	6.5201×10^{2}	6.0246×10^{2}	6.1344×10^{2}	6.0003×10^{2}
F7	Minimum	7.1616×10^{2}	7.2136×10^{2}	7.7915×10^{2}	7.2155×10^{2}	7.3797×10^{2}	7.1916×10^{2}
	Standard	2.9316×10^{0}	5.3634×10^{0}	6.4109×10^{0}	5.6956×10^{0}	2.5984×10^{1}	4.7996×10^{0}
	Average	$\mathbf{7.2264\times10^{2}}$	7.3128×10^{2}	8.1000×10^{2}	7.3531×10^{2}	7.8779×10^{2}	7.3580×10^{2}
F8	Minimum	8.0563×10^{2}	8.0877×10^{2}	8.2337×10^{2}	8.0953×10^{2}	8.1691×10^{2}	8.1093×10^{2}
	Standard	2.7004×10^{0}	3.6916×10^{0}	3.0439×10^{0}	5.9028×10^{0}	5.2250×10^{0}	4.9393×10^{0}
	Average	$\mathbf{8.1003\times10^{2}}$	8.1571×10^{2}	8.3066×10^{2}	8.1707×10^{2}	8.3011×10^{2}	8.1900×10^{2}
F9	Minimum	9.0000×10^{2}	9.0000×10^{2}	1.7431×10^{3}	9.0010×10^{2}	9.0000×10^{2}	9.0000×10^{2}
	Standard	3.6278×10^{-4}	2.9805×10^{-5}	9.3761×10^{0}	3.2595×10^{-1}	3.3522×10^{2}	1.1476×10^{-1}
	Average	9.0000×10^{2}	$\mathbf{9.0000\times10^{2}}$	1.7688×10^{3}	9.0053×10^{2}	1.1222×10^{3}	9.0004×10^{2}
F10	Minimum	1.2058×10^{3}	1.3859×10^{3}	1.8241×10^{3}	1.3830×10^{3}	1.2404×10^{3}	1.6424×10^{3}
	Standard	1.4110×10^{2}	1.8014×10^{2}	2.1331×10^{2}	3.5851×10^{2}	3.3428×10^{2}	1.8187×10^{2}
	Average	$\mathbf{1.5474\times10^{3}}$	1.8567×10^{3}	2.2035×10^{3}	2.0597×10^{3}	1.8392×10^{3}	1.9684×10^{3}
F11	Minimum	1.1014×10^{3}	1.1029×10^{3}	1.1036×10^{3}	1.1109×10^{3}	1.1068×10^{3}	1.1033×10^{3}
	Standard	1.1678×10^{0}	1.7146×10^{0}	6.5702×10^{0}	7.7321×10^{0}	9.0202×10^{0}	2.3864×10^{0}
	Average	$\mathbf{1.1037\times10^{3}}$	1.1055×10^{3}	1.1131×10^{3}	1.1245×10^{3}	1.1191×10^{3}	1.1073×10^{3}
F12	Minimum	2.3286×10^{3}	2.5192×10^{3}	6.5949×10^{3}	2.7906×10^{4}	2.9412×10^{3}	5.9776×10^{3}
	Standard	1.9802×10^{3}	3.8381×10^{3}	2.1988×10^{5}	6.4609×10^{5}	8.5348×10^{4}	1.8271×10^{4}
	Average	$\mathbf{4.4426\times10^{3}}$	7.9755×10^{3}	9.1112×10^{4}	1.1629×10^{6}	2.8205×10^{4}	3.0499×10^{4}
F13	Minimum	1.3034×10^{3}	1.3117×10^{3}	1.6215×10^{3}	6.3632×10^{3}	2.3267×10^{3}	1.3246×10^{3}
	Standard	7.5213×10^{0}	5.2925×10^{0}	1.9760×10^{3}	2.2805×10^{3}	7.2240×10^{3}	8.0906×10^{0}
	Average	$\mathbf{1.3193\times10^{3}}$	1.3206×10^{3}	4.4031×10^{3}	1.1033×10^{4}	1.0682×10^{4}	1.3360×10^{3}
F14	Minimum	1.4053×10^{3}	1.4068×10^{3}	1.4797×10^{3}	1.8395×10^{3}	1.4435×10^{3}	1.4129×10^{3}
	Standard	4.0637×10^{0}	4.7730×10^{0}	6.8690×10^{1}	1.4574×10^{3}	5.2428×10^{2}	3.9000×10^{0}
	Average	$\mathbf{1.4142\times10^{3}}$	1.4158×10^{3}	1.5548×10^{3}	4.1797×10^{3}	1.7642×10^{3}	1.4243×10^{3}
F15	Minimum	1.5013×10^{3}	1.5014×10^{3}	1.6439×10^{3}	4.8403×10^{3}	1.5885×10^{3}	1.5024×10^{3}
	Standard	8.9786×10^{-1}	1.2534×10^{0}	7.2090×10^{2}	3.6032×10^{3}	9.9553×10^{2}	3.4519×10^{0}
	Average	$\mathbf{1.5030\times10^{3}}$	1.5037×10^{3}	2.5293×10^{3}	1.1729×10^{4}	2.7269×10^{3}	1.5079×10^{3}
F16	Minimum	1.6009×10^{3}	1.6018×10^{3}	1.9230×10^{3}	1.7814×10^{3}	1.6038×10^{3}	1.6047×10^{3}
	Standard	2.3015×10^{1}	6.8326×10^{0}	8.5566×10^{1}	7.3164×10^{1}	1.3246×10^{2}	3.9049×10^{1}
	Average	1.6092×10^{3}	$\mathbf{1.6064\times10^{3}}$	2.0514×10^{3}	1.9465×10^{3}	1.8850×10^{3}	1.6578×10^{3}
F17	Minimum	1.7063×10^{3}	1.7082×10^{3}	1.7253×10^{3}	1.7502×10^{3}	1.7194×10^{3}	1.7372×10^{3}
	Standard	5.5389×10^{0}	6.2159×10^{0}	1.3014×10^{1}	1.7107×10^{1}	1.8681×10^{1}	1.1877×10^{1}
	Average	$\mathbf{1.7154\times10^{3}}$	1.7239×10^{3}	1.7459×10^{3}	1.7673×10^{3}	1.7473×10^{3}	1.7576×10^{3}

Continued

F18	Minimum	1.8087×10^{3}	1.8083×10^{3}	1.9399×10^{3}	2.7562×10^{3}	2.5535×10^{3}	1.8301×10^{3}
	Standard	3.6729×10^{0}	4.8055×10^{0}	3.1318×10^{3}	4.4393×10^{3}	1.0481×10^{4}	1.8561×10^{1}
	Average	**1.8146×10^{3}**	1.8174×10^{3}	4.0245×10^{3}	9.3511×10^{3}	1.4653×10^{4}	1.8501×10^{3}
F19	Minimum	1.9014×10^{3}	1.9011×10^{3}	1.9185×10^{3}	2.5019×10^{3}	1.9135×10^{3}	1.9036×10^{3}
	Standard	4.1790×10^{-1}	5.3030×10^{-1}	1.5045×10^{3}	2.2095×10^{3}	6.4974×10^{3}	1.2144×10^{0}
	Average	**1.9022×10^{3}**	1.9023×10^{3}	2.8190×10^{3}	7.3985×10^{3}	7.1079×10^{3}	1.9055×10^{3}
F20	Minimum	2.0001×10^{3}	2.0008×10^{3}	2.1487×10^{3}	2.0419×10^{3}	2.0290×10^{3}	2.0349×10^{3}
	Standard	3.5247×10^{0}	4.4942×10^{0}	4.5390×10^{1}	5.7618×10^{1}	5.5675×10^{1}	1.2512×10^{1}
	Average	**2.0050×10^{3}**	2.0071×10^{3}	2.1907×10^{3}	2.1608×10^{3}	2.1671×10^{3}	2.0552×10^{3}
F21	Minimum	2.2013×10^{3}	2.2003×10^{3}	2.2005×10^{3}	2.2019×10^{3}	2.2006×10^{3}	2.2043×10^{3}
	Standard	3.9979×10^{1}	5.8139×10^{1}	2.5097×10^{1}	5.0636×10^{1}	6.7547×10^{1}	3.9004×10^{1}
	Average	2.2298×10^{3}	2.2518×10^{3}	**2.2128×10^{3}**	2.3044×10^{3}	2.2777×10^{3}	2.3052×10^{3}
F22	Minimum	2.2116×10^{3}	2.2027×10^{3}	2.3014×10^{3}	2.2029×10^{3}	2.3003×10^{3}	2.2564×10^{3}
	Standard	2.4316×10^{1}	1.8141×10^{1}	5.5917×10^{2}	2.4693×10^{1}	1.4843×10^{0}	1.1637×10^{1}
	Average	**2.2928×10^{3}**	2.2986×10^{3}	2.4868×10^{3}	2.3005×10^{3}	2.3024×10^{3}	2.2993×10^{3}
F23	Minimum	2.6067×10^{3}	2.6065×10^{3}	2.6235×10^{3}	2.3160×10^{3}	2.6106×10^{3}	2.6142×10^{3}
	Standard	3.4077×10^{0}	5.8561×10^{0}	2.0994×10^{2}	5.9378×10^{1}	1.3082×10^{1}	5.0330×10^{0}
	Average	**2.6140×10^{3}**	2.6178×10^{3}	3.0859×10^{3}	2.6256×10^{3}	2.6266×10^{3}	2.6222×10^{3}
F24	Minimum	2.5001×10^{3}	2.5000×10^{3}	2.5000×10^{3}	2.5008×10^{3}	2.5000×10^{3}	2.6198×10^{3}
	Standard	1.0516×10^{2}	1.0062×10^{2}	8.0898×10^{1}	1.0643×10^{2}	1.0737×10^{2}	2.5225×10^{1}
	Average	2.6111×10^{3}	2.6976×10^{3}	**2.5291×10^{3}**	2.6913×10^{3}	2.5527×10^{3}	2.7494×10^{3}
F25	Minimum	2.8977×10^{3}	2.8977×10^{3}	2.8978×10^{3}	2.9000×10^{3}	2.8979×10^{3}	2.8978×10^{3}
	Standard	2.2879×10^{1}	2.2976×10^{1}	2.0711×10^{1}	1.7936×10^{1}	2.2390×10^{1}	2.2463×10^{1}
	Average	**2.9185×10^{3}**	2.9242×10^{3}	2.9236×10^{3}	2.9360×10^{3}	2.9282×10^{3}	2.9200×10^{3}
F26	Minimum	2.6013×10^{3}	2.8000×10^{3}	2.8000×10^{3}	2.8149×10^{3}	2.8000×10^{3}	2.9000×10^{3}
	Standard	5.4543×10^{1}	3.0513×10^{1}	2.3518×10^{2}	3.9740×10^{1}	1.2338×10^{2}	1.7585×10^{2}
	Average	2.8900×10^{3}	2.8900×10^{3}	2.9161×10^{3}	**2.8714×10^{3}**	2.8842×10^{3}	2.9390×10^{3}
F27	Minimum	3.0884×10^{3}	3.0920×10^{3}	3.0895×10^{3}	3.0906×10^{3}	3.0905×10^{3}	3.0894×10^{3}
	Standard	2.1523	1.6027×10^{0}	1.4397×10^{2}	2.7806×10^{1}	1.6936×10^{1}	9.5004×10^{-1}
	Average	3.0946×10^{3}	3.0948×10^{3}	3.1321×10^{3}	3.1085×10^{3}	3.1029×10^{3}	**3.0911×10^{3}**
F28	Minimum	2.8914×10^{3}	3.1000×10^{3}	3.1004×10^{3}	3.1013×10^{3}	3.1000×10^{3}	3.1791×10^{3}
	Standard	4.3527×10^{1}	1.0264×10^{2}	1.1126×10^{2}	1.0623×10^{2}	1.3039×10^{2}	7.6704×10^{1}
	Average	**3.0994×10^{3}**	3.1414×10^{3}	3.2700×10^{3}	3.3377×10^{3}	3.2994×10^{3}	3.2543×10^{3}
F29	Minimum	3.1480×10^{3}	3.1589×10^{3}	3.1868×10^{3}	3.1903×10^{3}	3.1623×10^{3}	3.1617×10^{3}
	Standard	2.0001×10^{1}	1.6517×10^{1}	4.2942×10^{1}	2.5773×10^{1}	4.7772×10^{1}	2.1709×10^{1}
	Average	**3.1829×10^{3}**	3.1968×10^{3}	3.2440×10^{3}	3.2389×10^{3}	3.2335×10^{3}	3.2000×10^{3}
F30	Minimum	4.1669×10^{3}	4.1856×10^{3}	5.9233×10^{3}	1.1286×10^{5}	4.1500×10^{3}	4.7180×10^{3}
	Standard	1.5470×10^{4}	9.7373×10^{3}	2.1114×10^{5}	6.0840×10^{5}	3.2872×10^{5}	2.5040×10^{5}
	Average	1.2770×10^{4}	1.3662×10^{4}	1.3667×10^{5}	6.2355×10^{5}	1.5150×10^{5}	1.1949×10^{5}

Note: The best performance values of the algorithm under different test functions are shown in bold.

1) Performance Comparison Between ICPO and Other Improved Algorithms (cec2017)

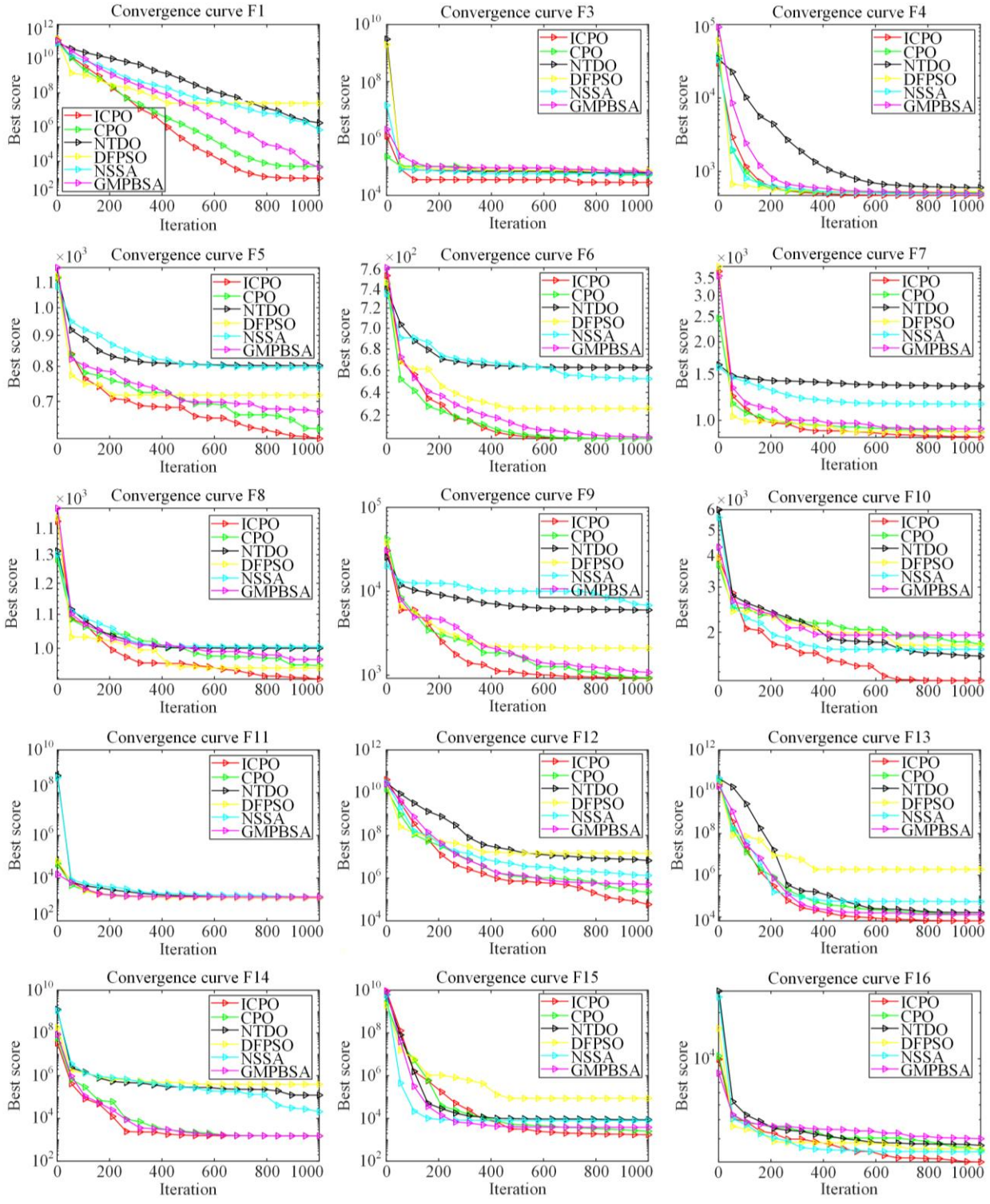
In this section, the improved versions of TDO algorithm, PSO algorithm, SSA algorithm and BSA algorithm are used to compare with CPO and ICPO proposed in this paper. As described, the dimensionality of the function is 10, the maximum number of iterations of the algorithms is 500, and the number of populations is 30. To ensure the reliability and fairness of the experiments, and to analyze the experimental results more accurately, each algorithm is run for 30 times. Table II compares the optimal values (Minimum), standard deviations (Standard) and average values (Average) of the different algorithms, while Fig. 5 compares the convergence diagrams of the algorithms with the maximum number of iterations of each algorithm being 1000.

From Table II, it can be seen that the average optimal fitness values of the proposed ICPO in most of the test functions in cec2017 are better than other algorithms. From the results, the following conclusions can be drawn: the average optimal fitness values of the ICPO in test functions F1, F4–F5, F7–F8, F10–F15, F17–F20, F22–F23, F25, F28–F30 are the best; the average optimal fitness of CPO in functions F3, F6, F9, F16 are the best; for functions F21, F24, the average optimal fitness values of NTDO are the best; the average optimal fitness of DFPSO in function F26 is the best; and for function F27, the average value of GMPBSA is the best. From the experimental results, it can be seen that the improved ICPO is the most effective, followed by the original CPO algorithm and the improved NTDO algorithm. Therefore, the proposed ICPO in this paper is

very effective and can greatly improve the overall performance of the algorithm.

From Fig. 5, it can be seen that in functions F1–F3, F5, F7–F10, F12–F13, F15–F21, F23–F24, F26, F29–F30, the convergence speed and convergence accuracy of ICPO are significantly better than the other improved algorithms. For functions F6 and F25, ICPO achieves the fastest convergence speed, although its convergence accuracy is not the highest. For the remaining test functions, its overall performance is satisfactory. These results demonstrate that the improved ICPO significantly improves local search ability, enabling it to reach the optimal value more quickly and accurately.

Based on the above results and analysis, it can be concluded that the overall performance of the proposed ICPO in the cec2017 test functions is excellent, with both high effectiveness and competitiveness.



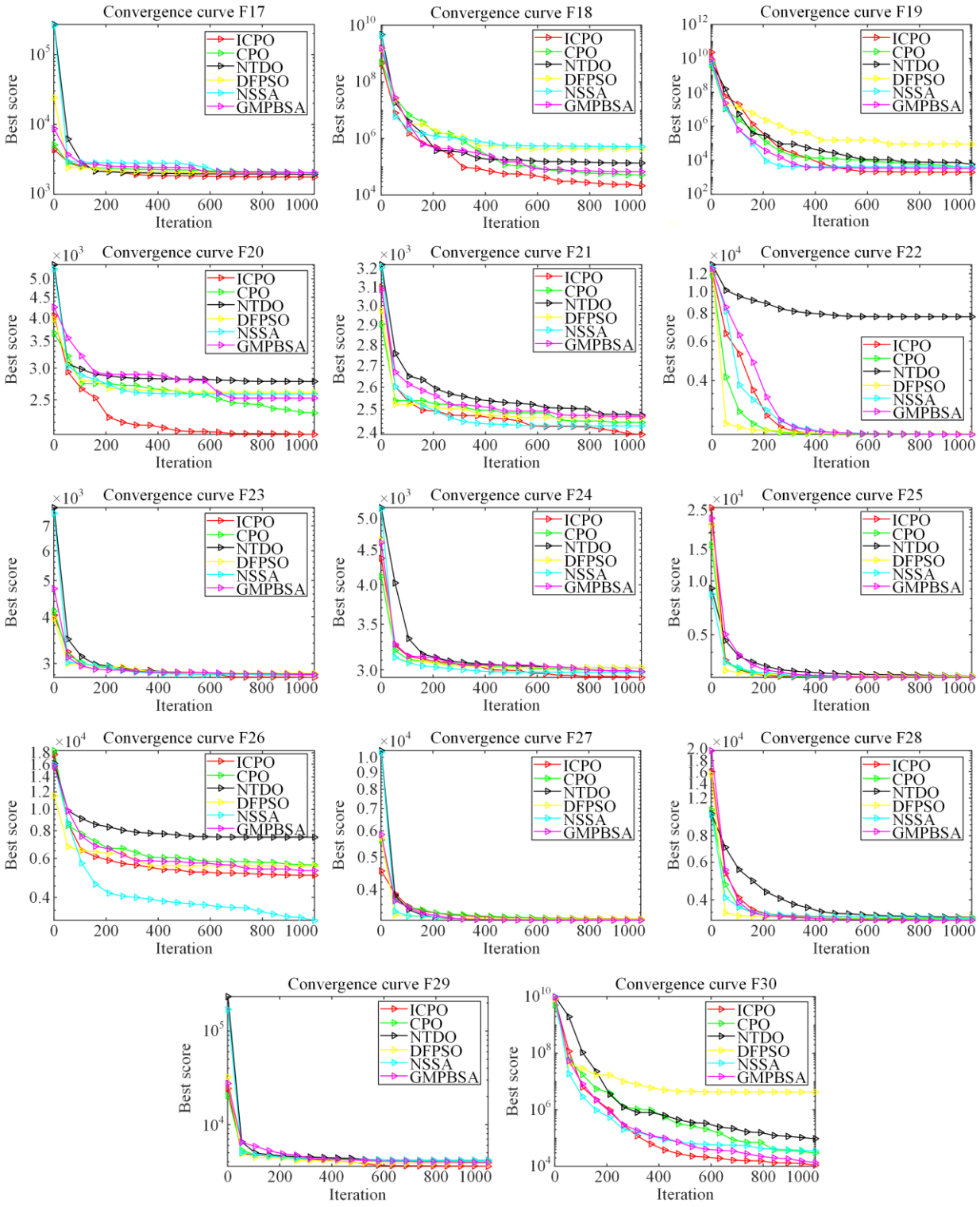


Fig. 5. Convergence curves of each improved algorithm under the cec2017 test sets.

2) Wilcoxon Rank Sum Test Analysis (cec2017)

To ensure the reliability of the experimental results and highlight the differences between the proposed algorithm and other improved algorithms, this section adopts the Wilcoxon rank-sum test at a significance level of p-value equaling 0.05 [57]. When p-value is greater than 0.5, the two algorithms are considered to have comparable performance; when p-value is less than 0.5, the difference between the algorithms is statistically significant, indicating that one algorithm has a clear advantage. In Table III, the values with p-value being greater than 0.5 are shown in bold. It can be seen that the

values with p-value being greater than 0.5 account for only a small portion, indicating that the improved ICPO differs from the other improved algorithms and demonstrates superior performance in most instances.

TABLE III
WILCOXON RANK SUM TEST RESULTS (CEC2017)

	CPO	NTDO	DFPSO	NSSA	GMPBSA
F1	**7.2827×10^{1}**	2.3168×10^{6}	3.0199×10^{-11}	5.8587×10^{-6}	7.7387×10^{-6}
F3	3.8307×10^{-5}	9.9186×10^{-11}	2.1566×10^{-3}	2.4913×10^{-6}	4.0772×10^{-11}
F4	3.9167×10^{-2}	4.8413×10^{-2}	1.1737×10^{-9}	9.5139×10^{-6}	3.8349×10^{-6}
F5	1.7294×10^{-7}	3.0199×10^{-11}	1.4643×10^{-10}	8.9934×10^{-11}	2.6099×10^{-10}
F6	2.0152×10^{-8}	3.0199×10^{-11}	3.0199×10^{-11}	3.6897×10^{-11}	7.2208×10^{-6}
F7	3.9648×10^{-8}	3.0199×10^{-11}	9.7555×10^{-10}	3.0199×10^{-11}	4.1997×10^{-10}
F8	5.0922×10^{-8}	3.0199×10^{-11}	1.3594×10^{-7}	3.3384×10^{-11}	8.1014×10^{-10}
F9	1.5581×10^{-8}	3.0199×10^{-11}	3.0199×10^{-11}	2.1544×10^{-10}	1.8916×10^{-4}
F10	4.3106×10^{-8}	3.3384×10^{-11}	3.6459×10^{-8}	9.7917×10^{-5}	4.6159×10^{-10}
F11	3.8349×10^{-6}	6.1210×10^{-10}	3.0199×10^{-11}	3.0199×10^{-11}	1.8500×10^{-8}
F12	3.3681×10^{-5}	1.0937×10^{-10}	3.0199×10^{-11}	2.3897×10^{-8}	8.9934×10^{-11}
F13	**3.3285×10^{-1}**	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	5.9673×10^{-9}
F14	**1.3732×10^{-1}**	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	1.4110×10^{-9}
F15	8.3146×10^{-3}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	4.9980×10^{-9}
F16	**1.0869×10^{-1}**	3.0199×10^{-11}	3.0199×10^{-11}	2.1544×10^{-10}	2.9215×10^{-9}
F17	3.5708×10^{-6}	3.0199×10^{-11}	3.0199×10^{-11}	5.4941×10^{-11}	3.0199×10^{-11}
F18	1.6955×10^{-2}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F19	**3.1830×10^{-1}**	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F20	4.2446×10^{-2}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F21	**5.2978×10^{-1}**	6.6689×10^{-3}	4.4205×10^{-6}	**1.6687×10^{-1}**	7.6950×10^{-8}
F22	**6.7350×10^{-1}**	5.6073×10^{-5}	8.4848×10^{-9}	**5.3685×10^{-2}**	5.5699×10^{-3}
F23	1.4412×10^{-2}	3.0199×10^{-11}	9.7555×10^{-10}	1.0666×10^{-07}	2.0152×10^{-8}
F24	4.7138×10^{-4}	1.2860×10^{-6}	1.7666×10^{-3}	6.7650×10^{-5}	2.8716×10^{-10}
F25	**7.7312×10^{-1}**	4.3584×10^{-2}	2.6784×10^{-6}	1.3703×10^{-3}	**2.0095×10^{-1}**
F26	1.4932×10^{-4}	1.9112×10^{-2}	**1.3345×10^{-1}**	**3.7904×10^{-1}**	3.0199×10^{-11}
F27	**5.7929×10^{-1}**	**6.7350×10^{-1}**	4.9426×10^{-5}	3.9881×10^{-4}	7.6950×10^{-8}
F28	2.7548×10^{-3}	4.1997×10^{-10}	1.6132×10^{-10}	2.3855×10^{-8}	3.0199×10^{-11}
F29	6.3772×10^{-3}	2.6695×10^{-9}	3.1589×10^{-10}	1.7290×10^{-6}	4.0330×10^{-3}
F30	**1.1536×10^{-1}**	7.0881×10^{-8}	3.0199×10^{-11}	2.0283×10^{-7}	3.9881×10^{-4}

Note: Bold entries indicates the values with insignificant p-value in the non-parametric test.

3) Performance Comparison Between ICPO and Other Swarm Intelligence Optimization Algorithms (cec2017)

In order to visualize the advantages of ICPO, this section compares CPO and ICPO with other swarm intelligent optimization algorithms, including WOA, DBO, COA, GWO, and BOA. As in previous cases, the dimension of the function is 10, the maximum number of iterations for each algorithm is 500, and the number of populations is 30. Table IV demonstrates the Minimumimum, standard deviation, and mean values of each algorithm, and Fig. 6 compares the convergence curves of each swarm intelligent optimization algorithm with ICPO.

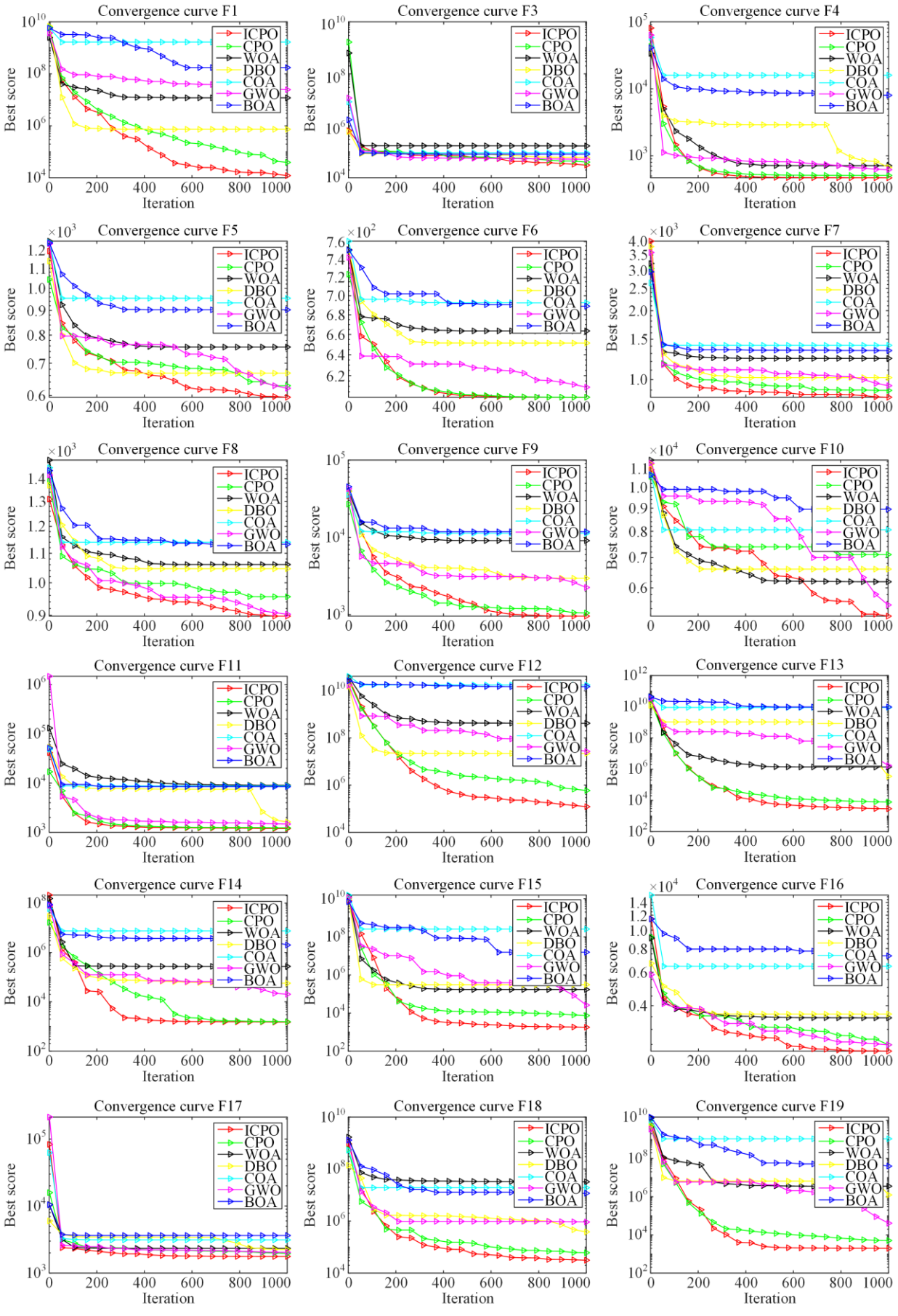
TABLE IV
EXPERIMENTAL DATA UNDER DIFFERENT ORIGINAL ALGORITHMS (CEC2017)

		ICPO	CPO	WOA	DBO	COA	GWO	BOA
F1	Minimum	1.1752×10^{2}	1.2201×10^{2}	4.5472×10^{6}	1.6826×10^{2}	4.5410×10^{9}	3.1758×10^{4}	1.9173×10^{9}
	Standard	1.7789×10^{2}	5.9281×10^{2}	9.2567×10^{7}	8.3451×10^{6}	3.2364×10^{9}	1.2996×10^{8}	2.7197×10^{9}
	Average	**3.2707×10^{2}**	6.0591×10^{2}	6.7430×10^{7}	2.6983×10^{6}	1.1086×10^{10}	5.2519×10^{7}	6.0411×10^{9}
F3	Minimum	3.0453×10^{2}	3.0069×10^{2}	8.5784×10^{2}	3.0004×10^{2}	4.3721×10^{3}	3.7242×10^{2}	3.2304×10^{3}
	Standard	6.5879×10^{1}	2.2280×10^{1}	4.0397×10^{3}	3.8101×10^{2}	3.0411×10^{3}	2.5740×10^{3}	2.8673×10^{3}
	Average	3.6221×10^{2}	**3.1772×10^{2}**	5.5531×10^{3}	6.5199×10^{2}	9.8784×10^{3}	3.4002×10^{3}	9.2329×10^{3}
F4	Minimum	4.0067×10^{2}	4.0093×10^{2}	4.0485×10^{2}	4.0161×10^{2}	6.3035×10^{2}	4.0390×10^{2}	4.1554×10^{2}
	Standard	1.1796×10^{0}	1.1555×10^{0}	5.8845×10^{1}	3.8811×10^{1}	4.7697×10^{2}	2.2534×10^{1}	5.3765×10^{2}
	Average	**4.0367×10^{2}**	4.0380×10^{2}	4.6319×10^{2}	4.3306×10^{2}	1.1536×10^{3}	4.2153×10^{2}	1.4209×10^{3}
F5	Minimum	5.0478×10^{2}	5.1161×10^{2}	5.1845×10^{2}	5.1610×10^{2}	5.5112×10^{2}	5.0810×10^{2}	5.6646×10^{2}
	Standard	3.7148×10^{0}	4.4030×10^{0}	2.5505×10^{1}	1.5389×10^{1}	1.9363×10^{1}	1.0470×10^{1}	1.4305×10^{1}
	Average	**5.1134×10^{2}**	5.1922×10^{2}	5.6006×10^{2}	5.4138×10^{2}	5.9449×10^{2}	5.2030×10^{2}	5.9677×10^{2}
F6	Minimum	6.0000×10^{2}	6.0000×10^{2}	6.1481×10^{2}	6.0004×10^{2}	6.2758×10^{2}	6.0010×10^{2}	6.2136×10^{2}
	Standard	7.4090×10^{-3}	1.5334×10^{-3}	1.1553×10^{1}	8.9637×10^{0}	1.1138×10^{1}	2.8663×10^{0}	8.2651×10^{0}
	Average	6.0001×10^{2}	**6.0000×10^{2}**	6.3886×10^{2}	6.1391×10^{2}	6.4858×10^{2}	6.0210×10^{2}	6.3820×10^{2}
F7	Minimum	7.1513×10^{2}	7.2157×10^{2}	7.2162×10^{2}	7.1753×10^{2}	7.5703×10^{2}	7.1829×10^{2}	7.8159×10^{2}
	Standard	3.7365×10^{0}	4.2091×10^{0}	2.5625×10^{1}	1.8241×10^{1}	1.5246×10^{1}	9.9726×10^{0}	1.0265×10^{1}
	Average	**7.2169×10^{2}**	7.3074×10^{2}	7.8438×10^{2}	7.5042×10^{2}	7.9704×10^{2}	7.3946×10^{2}	8.0049×10^{2}

Continued

F8	Minimum	8.0487×10^{2}	8.0808×10^{2}	8.2604×10^{2}	8.1592×10^{2}	8.4512×10^{2}	8.0306×10^{2}	8.3830×10^{2}
	Standard	2.8646×10^{0}	4.2075×10^{0}	1.3619×10^{1}	1.1155×10^{1}	5.6155×10^{0}	6.0325×10^{0}	7.1401×10^{0}
	Average	$\mathbf{8.0956\times10^{2}}$	8.1689×10^{2}	8.4696×10^{2}	8.3411×10^{2}	8.5597×10^{2}	8.1574×10^{2}	8.5680×10^{2}
F9	Minimum	9.0000×10^{2}	9.0000×10^{2}	9.5987×10^{2}	9.0071×10^{2}	1.1814×10^{3}	9.0011×10^{2}	9.6044×10^{2}
	Standard	6.4250×10^{-4}	3.7997×10^{-5}	2.7763×10^{2}	1.7370×10^{2}	1.8373×10^{2}	4.6180×10^{1}	2.1402×10^{2}
	Average	9.0000×10^{2}	$\mathbf{9.0000\times10^{2}}$	1.4580×10^{3}	9.9128×10^{2}	1.5160×10^{3}	9.2075×10^{2}	1.3474×10^{3}
F10	Minimum	1.1985×10^{3}	1.4367×10^{3}	1.6072×10^{3}	1.4890×10^{3}	1.9396×10^{3}	1.1233×10^{3}	2.1584×10^{3}
	Standard	1.3892×10^{2}	2.0474×10^{2}	2.8318×10^{2}	2.8064×10^{2}	2.4535×10^{2}	4.2315×10^{2}	1.8608×10^{2}
	Average	$\mathbf{1.4995\times10^{3}}$	1.8509×10^{3}	2.1803×10^{3}	2.0409×10^{3}	2.6121×10^{3}	1.6911×10^{3}	2.7239×10^{3}
F11	Minimum	1.1022×10^{3}	1.1039×10^{3}	1.1260×10^{3}	1.1077×10^{3}	1.2543×10^{3}	1.1103×10^{3}	1.2498×10^{3}
	Standard	1.1649×10^{0}	1.3011×10^{0}	9.1751×10^{1}	1.1877×10^{2}	1.0811×10^{3}	3.4093×10^{1}	7.2399×10^{2}
	Average	$\mathbf{1.1045\times10^{3}}$	1.1060×10^{3}	1.2410×10^{3}	1.2261×10^{3}	2.2672×10^{3}	1.1448×10^{3}	1.7725×10^{3}
F12	Minimum	1.7770×10^{3}	1.9641×10^{3}	7.3886×10^{4}	3.7112×10^{3}	1.2757×10^{7}	1.1670×10^{4}	4.4974×10^{6}
	Standard	1.5818×10^{3}	9.7629×10^{3}	6.5100×10^{6}	3.6355×10^{6}	2.2735×10^{8}	9.0105×10^{5}	5.0425×10^{7}
	Average	$\mathbf{3.8970\times10^{3}}$	1.0302×10^{4}	6.5913×10^{6}	2.5510×10^{6}	2.9313×10^{8}	8.3257×10^{5}	6.1220×10^{7}
F13	Minimum	1.3070×10^{3}	1.3092×10^{3}	1.6634×10^{3}	1.7908×10^{3}	8.4143×10^{3}	2.2862×10^{3}	3.0721×10^{4}
	Standard	7.6577×10^{0}	5.6133×10^{0}	1.2563×10^{4}	1.5431×10^{4}	3.3564×10^{6}	8.8415×10^{3}	2.7226×10^{5}
	Average	$\mathbf{1.3192\times10^{3}}$	1.3221×10^{3}	1.4081×10^{4}	1.5371×10^{4}	1.0272×10^{6}	1.2311×10^{4}	2.4703×10^{5}
F14	Minimum	1.4076×10^{3}	1.4084×10^{3}	1.4473×10^{3}	1.4416×10^{3}	1.4738×10^{3}	1.4689×10^{3}	1.5138×10^{3}
	Standard	4.6058×10^{0}	4.0158×10^{0}	1.7265×10^{3}	6.0404×10^{2}	6.2115×10^{1}	2.2039×10^{3}	1.9082×10^{3}
	Average	1.4146×10^{3}	$\mathbf{1.4141\times10^{3}}$	3.1140×10^{3}	1.9032×10^{3}	1.5734×10^{3}	3.9379×10^{3}	3.9268×10^{3}
F15	Minimum	1.5007×10^{3}	1.5011×10^{3}	1.7675×10^{3}	1.6491×10^{3}	1.7037×10^{3}	1.5861×10^{3}	2.7546×10^{3}
	Standard	1.1145×10^{0}	1.0145×10^{0}	1.0756×10^{4}	4.4548×10^{3}	3.4536×10^{3}	3.6032×10^{3}	5.2499×10^{3}
	Average	$\mathbf{1.5029\times10^{3}}$	1.5029×10^{3}	1.0973×10^{4}	4.8314×10^{3}	9.4596×10^{3}	4.9623×10^{3}	1.0554×10^{4}
F16	Minimum	1.6013×10^{3}	1.6014×10^{3}	1.6869×10^{3}	1.6006×10^{3}	1.7388×10^{3}	1.6176×10^{3}	1.7317×10^{3}
	Standard	3.7332×10^{1}	9.8717×10^{0}	1.4494×10^{2}	1.5715×10^{2}	1.0725×10^{2}	1.5251×10^{2}	1.1549×10^{2}
	Average	1.6157×10^{3}	$\mathbf{1.6068\times10^{3}}$	1.9616×10^{3}	1.7624×10^{3}	2.0697×10^{3}	1.7630×10^{3}	1.9832×10^{3}
F17	Minimum	1.7072×10^{3}	1.7103×10^{3}	1.7421×10^{3}	1.7298×10^{3}	1.7436×10^{3}	1.7151×10^{3}	1.7783×10^{3}
	Standard	6.8154×10^{0}	5.7430×10^{0}	8.2149×10^{1}	4.2137×10^{1}	4.6620×10^{1}	3.9170×10^{1}	2.1117×10^{1}
	Average	$\mathbf{1.7175\times10^{3}}$	1.7236×10^{3}	1.8361×10^{3}	1.7970×10^{3}	1.8084×10^{3}	1.7735×10^{3}	1.8159×10^{3}
F18	Minimum	1.8072×10^{3}	1.8051×10^{3}	2.0017×10^{3}	2.1596×10^{3}	6.4471×10^{3}	3.5823×10^{3}	5.4095×10^{4}
	Standard	4.0937×10^{0}	5.0900×10^{0}	1.3989×10^{4}	1.4797×10^{4}	4.1950×10^{7}	1.3674×10^{4}	4.8781×10^{6}
	Average	$\mathbf{1.8149\times10^{3}}$	1.8163×10^{3}	1.7434×10^{4}	2.0934×10^{4}	1.5942×10^{7}	2.3202×10^{4}	3.4012×10^{6}
F19	Minimum	1.9010×10^{3}	1.9010×10^{3}	2.0319×10^{3}	2.0321×10^{3}	2.1665×10^{3}	1.9109×10^{3}	7.5502×10^{3}
	Standard	5.2415×10^{-1}	5.8451×10^{-1}	4.9002×10^{5}	1.7781×10^{4}	2.1928×10^{4}	4.2146×10^{4}	6.1545×10^{4}
	Average	$\mathbf{1.9020\times10^{3}}$	1.9022×10^{3}	1.4641×10^{5}	1.0654×10^{4}	1.4734×10^{4}	1.8285×10^{4}	7.0345×10^{4}
F20	Minimum	2.0004×10^{3}	2.0020×10^{3}	2.0775×10^{3}	2.0236×10^{3}	2.1503×10^{3}	2.0251×10^{3}	2.0839×10^{3}
	Standard	3.0430×10^{0}	2.9995×10^{0}	8.6472×10^{1}	7.2468×10^{1}	3.8203×10^{1}	6.6148×10^{1}	4.7900×10^{1}
	Average	$\mathbf{2.0046\times10^{3}}$	2.0060×10^{3}	2.1945×10^{3}	2.1196×10^{3}	2.2312×10^{3}	2.0979×10^{3}	2.1838×10^{3}
F21	Minimum	2.2009×10^{3}	2.2002×10^{3}	2.2118×10^{3}	2.2040×10^{3}	2.2683×10^{3}	2.2055×10^{3}	2.2100×10^{3}
	Standard	3.8709×10^{1}	6.0697×10^{1}	5.3337×10^{1}	5.0007×10^{1}	3.8014×10^{1}	2.9830×10^{1}	2.7934×10^{1}
	Average	$\mathbf{2.2330\times10^{3}}$	2.2572×10^{3}	2.3156×10^{3}	2.2362×10^{3}	2.3701×10^{3}	2.3132×10^{3}	2.2279×10^{3}
F22	Minimum	2.2192×10^{3}	2.2000×10^{3}	2.2663×10^{3}	2.3012×10^{3}	2.4992×10^{3}	2.3011×10^{3}	2.3249×10^{3}
	Standard	2.0115×10^{1}	2.1065×10^{1}	4.2921×10^{2}	9.2416×10^{0}	4.0624×10^{2}	9.3159×10^{0}	3.6715×10^{1}
	Average	2.2966×10^{3}	$\mathbf{2.2966\times10^{3}}$	2.4559×10^{3}	2.3116×10^{3}	3.1818×10^{3}	2.3095×10^{3}	2.3861×10^{3}
F23	Minimum	2.6087×10^{3}	2.6097×10^{3}	2.6263×10^{3}	2.6231×10^{3}	2.6603×10^{3}	2.6050×10^{3}	2.6467×10^{3}
	Standard	3.5123×10^{0}	4.6988×10^{0}	2.4385×10^{1}	1.4150×10^{1}	3.1890×10^{1}	1.2049×10^{1}	1.9705×10^{1}
	Average	$\mathbf{2.6142\times10^{3}}$	2.6187×10^{3}	2.6546×10^{3}	2.6462×10^{3}	2.7184×10^{3}	2.6256×10^{3}	2.6744×10^{3}
F24	Minimum	2.5002×10^{3}	2.5000×10^{3}	2.6023×10^{3}	2.5000×10^{3}	2.6956×10^{3}	2.7327×10^{3}	2.5637×10^{3}
	Standard	1.0431×10^{2}	1.0016×10^{2}	4.0910×10^{1}	8.5669×10^{1}	6.6694×10^{1}	1.3252×10^{1}	7.3289×10^{1}
	Average	$\mathbf{2.6489\times10^{3}}$	2.6966×10^{3}	2.7766×10^{3}	2.7380×10^{3}	2.8888×10^{3}	2.7505×10^{3}	2.6564×10^{3}
F25	Minimum	2.6069×10^{3}	2.8977×10^{3}	2.9038×10^{3}	2.9012×10^{3}	3.0756×10^{3}	2.8980×10^{3}	3.0474×10^{3}
	Standard	6.0478×10^{1}	2.2867×10^{1}	3.1316×10^{1}	2.3664×10^{1}	2.8219×10^{2}	2.3510×10^{1}	2.0661×10^{2}
	Average	$\mathbf{2.9056\times10^{3}}$	2.9183×10^{3}	2.9624×10^{3}	2.9453×10^{3}	3.5340×10^{3}	2.9365×10^{3}	3.5262×10^{3}
F26	Minimum	2.6005×10^{3}	2.6003×10^{3}	2.8273×10^{3}	2.8056×10^{3}	3.2449×10^{3}	2.8379×10^{3}	2.9973×10^{3}
	Standard	7.5613×10^{1}	5.4724×10^{1}	6.0706×10^{2}	1.8025×10^{2}	4.0348×10^{2}	3.0713×10^{2}	1.3918×10^{2}
	Average	$\mathbf{2.8801\times10^{3}}$	2.8900×10^{3}	3.5544×10^{3}	3.1054×10^{3}	4.1497×10^{3}	3.0770×10^{3}	3.2035×10^{3}
F27	Minimum	3.0920×10^{3}	3.0904×10^{3}	3.0937×10^{3}	3.0933×10^{3}	3.1240×10^{3}	3.0908×10^{3}	3.1031×10^{3}
	Standard	1.6981×10^{0}	2.0752×10^{0}	4.7643×10^{1}	2.3226×10^{1}	4.0741×10^{1}	1.6985×10^{1}	3.1345×10^{1}
	Average	$\mathbf{3.0953\times10^{3}}$	3.0954×10^{3}	3.1474×10^{3}	3.1170×10^{3}	3.1910×10^{3}	3.1002×10^{3}	3.1570×10^{3}
F28	Minimum	2.8797×10^{3}	3.1000×10^{3}	3.2163×10^{3}	2.8000×10^{3}	3.3174×10^{3}	3.1005×10^{3}	3.2758×10^{3}
	Standard	1.0047×10^{2}	1.1543×10^{2}	1.4035×10^{2}	1.8368×10^{2}	1.3896×10^{2}	1.0506×10^{2}	2.1535×10^{2}
	Average	$\mathbf{3.1268\times10^{3}}$	3.1592×10^{3}	3.4584×10^{3}	3.3446×10^{3}	3.7610×10^{3}	3.3556×10^{3}	3.7014×10^{3}
F29	Minimum	3.1474×10^{3}	3.1698×10^{3}	3.1782×10^{3}	3.1531×10^{3}	3.2608×10^{3}	3.1584×10^{3}	3.2175×10^{3}
	Standard	1.6967×10^{1}	1.5833×10^{1}	1.0893×10^{2}	8.2172×10^{1}	1.0464×10^{2}	4.5708×10^{1}	6.5528×10^{1}
	Average	$\mathbf{3.1834\times10^{3}}$	3.1969×10^{3}	3.4037×10^{3}	3.2763×10^{3}	3.4236×10^{3}	3.2148×10^{3}	3.3397×10^{3}
F30	Minimum	4.4840×10^{3}	4.6103×10^{3}	2.0531×10^{4}	7.5853×10^{3}	8.4219×10^{5}	8.2804×10^{3}	5.5468×10^{5}
	Standard	5.0758×10^{3}	9.0256×10^{3}	1.6856×10^{6}	1.3457×10^{6}	8.6065×10^{6}	1.3541×10^{6}	4.5837×10^{6}
	Average	$\mathbf{8.2190\times10^{3}}$	1.3348×10^{4}	1.6443×10^{6}	1.1139×10^{6}	7.3779×10^{6}	9.4144×10^{5}	5.4187×10^{6}

Note: The best performance values of the algorithm under different test functions are shown in bold.


Convergence curve F1
Convergence curve F3
Convergence curve F4
Convergence curve F5
Convergence curve F6
Convergence curve F7
Convergence curve F8
Convergence curve F9
Convergence curve F10
Convergence curve F11
Convergence curve F12
Convergence curve F13
Convergence curve F14
Convergence curve F15
Convergence curve F16
Convergence curve F17
Convergence curve F18
Convergence curve F19
Best score
Iteration
ICPO
CPO
WOA
DBO
COA
GWO
BOA

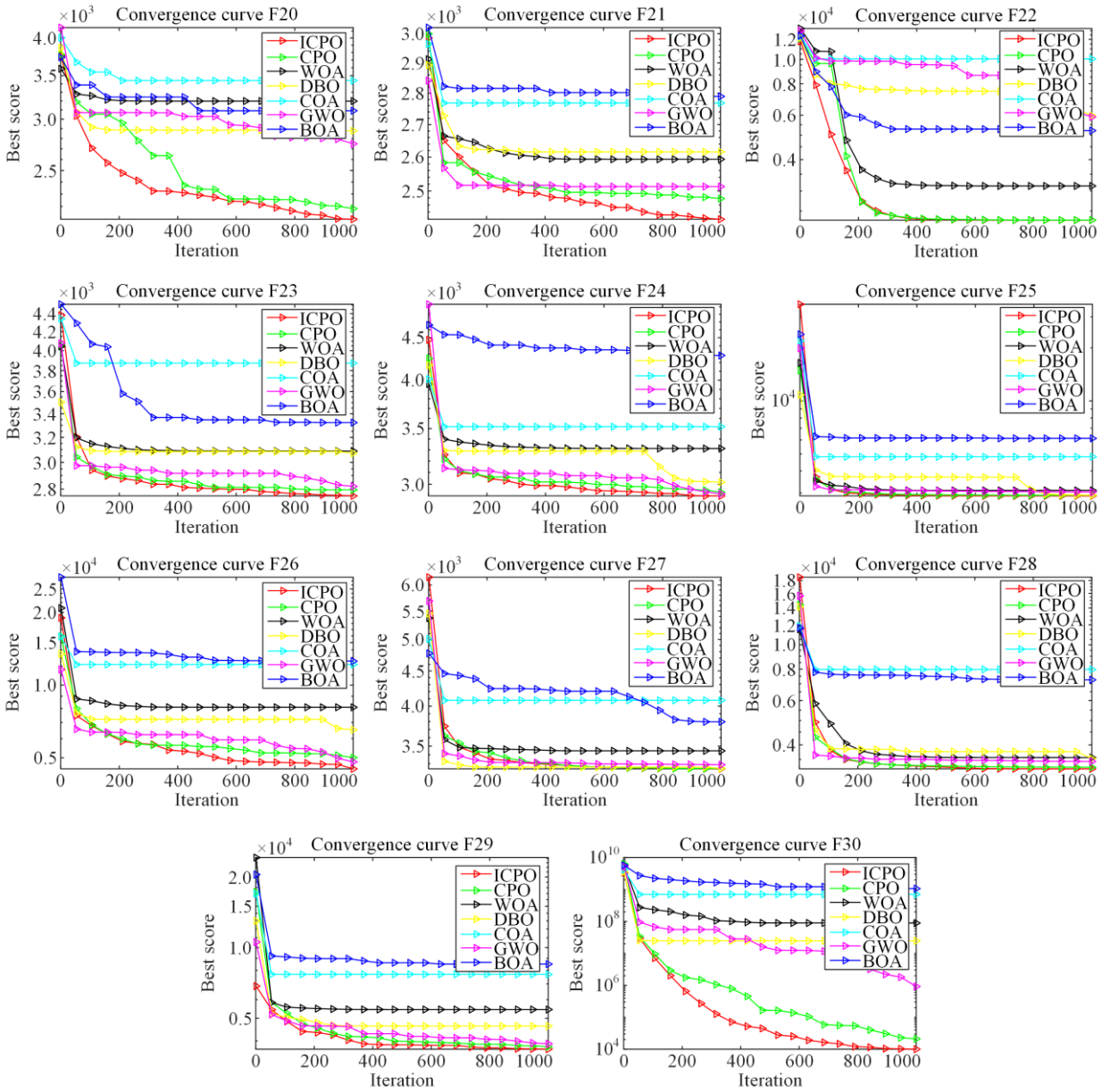


Fig. 6. Convergence curves of each algorithm under the cec2017 test sets.

From Table IV, it can be seen that the average optimal fitness values of the proposed ICPO are better than other swarm intelligent optimization algorithms for most of the functions in the cec2017 test functions. From the results, the following conclusions can be drawn: the average values of ICPO are optimal for the test functions F1, F4–F5, F7–F8, F10–F13, F15, F17–F21, and F23–F30, whereas the average values of CPO in functions F3, F6, F9, F14, F16, F22 are optimal. Therefore, the proposed ICPO shows clear advantages, achieving optimal average values across 23 test functions, which confirms its high effectiveness.

From Fig. 6, it can be seen that the convergence speed and convergence accuracy of ICPO in the test functions F1–F5, F7–F26, and F28–F30 are significantly better than other algorithms, and the convergence speed of ICPO is the fastest in function F6. ICPO can identify the optimal value more quickly and accurately, thereby greatly improving the optimization performance of the algorithm.

In summary, the proposed ICPO demonstrates significant advantages in solving the three metrics across the 29 cec2017 test functions, as well as in the convergence curves, with overall performance surpassing that of other intelligent optimization algorithms.

4) Wilcoxon Rank Sum Test Analysis (cec2017)

This subsection uses the Wilcoxon rank sum test to better validate the excellence of ICPO. Table V shows the Wilcoxon rank sum test results of ICPO with CPO, WOA, DBO, COA, GWO, and BOA algorithms run

independently for 30 times under dimension dim being 10 for the 29 test functions of cec2017. From Table V, it can be seen that: the ICPO and WOA, DBO, COA, BOA algorithms have p-value which is less than 0.5 under all functions, i.e., there is a significant difference with these algorithms under all test functions of cec2017; ICPO is significantly different from the CPO and GWO algorithms compared to the vast majority of functions. In summary, it shows that ICPO is more advantageous compared to the remaining six algorithms in terms of significant difference in the performance of optimization search.

TABLE V
WILCOXON RANK SUM TEST RESULTS (CEC2017)

	CPO	WOA	DBO	COA	GWO	BOA
F1	3.8481×10^{-3}	3.0199×10^{-11}	3.3461×10^{-8}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F3	6.5261×10^{-7}	3.6897×10^{-11}	2.9205×10^{-2}	3.0199×10^{-11}	2.2273×10^{-9}	3.0199×10^{-11}
F4	$\mathbf{2.5805\times10^{-1}}$	4.1997×10^{-10}	6.7183×10^{-10}	3.0199×10^{-11}	5.5727×10^{-10}	3.0199×10^{-11}
F5	1.6947×10^{-9}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	2.3800×10^{-3}	3.0199×10^{-11}
F6	8.8829×10^{-6}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F7	5.8587×10^{-6}	3.0199×10^{-11}	2.9215×10^{-9}	3.0199×10^{-11}	6.2828×10^{-06}	3.0199×10^{-11}
F8	3.4971×10^{-9}	3.0199×10^{-11}	5.4941×10^{-11}	3.0199×10^{-11}	1.2362×10^{-3}	3.0199×10^{-11}
F9	9.0632×10^{-8}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F10	1.1567×10^{-7}	2.1544×10^{-10}	2.8314×10^{-8}	3.0199×10^{-11}	1.2732×10^{-2}	3.0199×10^{-11}
F11	4.9426×10^{-5}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F12	7.6171×10^{-3}	3.0199×10^{-11}	7.7725×10^{-9}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F13	$\mathbf{3.3285\times10^{-1}}$	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F14	$\mathbf{8.6499\times10^{-1}}$	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F15	$\mathbf{1.9073\times10^{-1}}$	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F16	9.8834×10^{-3}	3.0199×10^{-11}	4.0772×10^{-11}	3.0199×10^{-11}	7.3891×10^{-11}	3.0199×10^{-11}
F17	$\mathbf{1.0547\times10^{-1}}$	3.0199×10^{-11}	1.2057×10^{-10}	3.0199×10^{-11}	4.9752×10^{-11}	3.0199×10^{-11}
F18	1.5969×10^{-3}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F19	3.5137×10^{-2}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F20	4.2067×10^{-2}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F21	$\mathbf{3.0418\times10^{-1}}$	1.2023×10^{-8}	6.7350×10^{-1}	1.6132×10^{-10}	2.6695×10^{-9}	1.0763×10^{-2}
F22	$\mathbf{3.8710\times10^{-1}}$	4.6159×10^{-10}	9.8329×10^{-8}	3.0199×10^{-11}	8.1527×10^{-11}	3.0199×10^{-11}
F23	7.2208×10^{-6}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	1.1058×10^{-4}	3.0199×10^{-11}
F24	8.6844×10^{-3}	3.0199×10^{-11}	1.0666×10^{-7}	1.8500×10^{-8}	2.2780×10^{-5}	1.1711×10^{-2}
F25	$\mathbf{5.9969\times10^{-1}}$	3.0939×10^{-6}	1.8731×10^{-7}	3.0199×10^{-11}	2.4327×10^{-5}	3.0199×10^{-11}
F26	1.5178×10^{-3}	3.0199×10^{-11}	8.3520×10^{-8}	3.0199×10^{-11}	3.0199×10^{-11}	5.0723×10^{-10}
F27	2.1506×10^{-2}	3.0199×10^{-11}	7.3763×10^{-10}	3.0199×10^{-11}	6.3772×10^{-3}	3.0199×10^{-11}
F28	$\mathbf{4.8252\times10^{-1}}$	8.9934×10^{-11}	1.5401×10^{-9}	3.0199×10^{-11}	7.3891×10^{-11}	4.0772×10^{-11}
F29	6.9724×10^{-3}	4.0772×10^{-11}	3.8053×10^{-7}	3.0199×10^{-11}	$\mathbf{7.9782\times10^{-2}}$	3.0199×10^{-11}
F30	$\mathbf{1.2597\times10^{-1}}$	3.0199×10^{-11}	2.5711×10^{-7}	3.0199×10^{-11}	2.3768×10^{-7}	3.0199×10^{-11}

Note: Bold entries indicates the values with insignifcant p-value in the non-parametric test.

B. CEC2020 Test Functions

There are 10 single-objective test functions in the cec2020 test functions, composed of several functions selected from the cec2014 and cec2017 test functions, and the search intervals are all between [−100, 100]. All the test functions solve the minimization problem with optional dimensions: 2/5/10/15/20, whereas the dimension of the test function is selected to be 20 in this study.

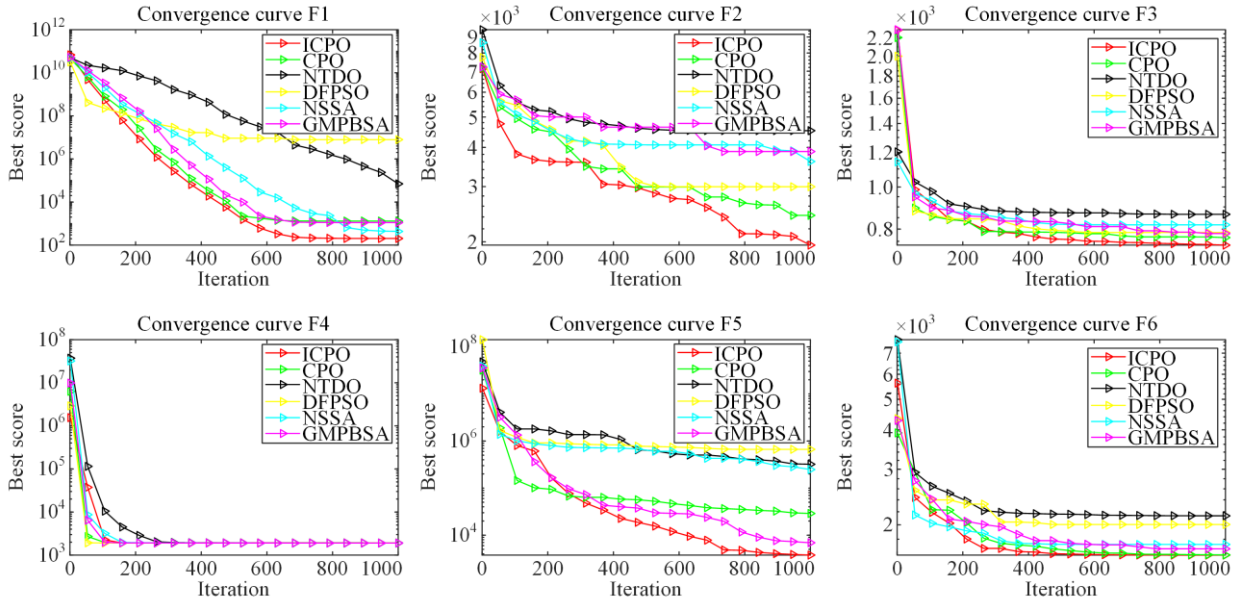
1) Color/Grayscale Figures

In this subsection, the improved versions of the NTDO, DFPSO, NSSA, GMPBSA algorithms are compares with the proposed ICPO and the original CPO algorithm under the cec2020 test functions. The number of dimensions is 20, the number of populations is 30, and the maximum number of iterations is 500 [58]. In order to ensure the stability and reliability of the experimental results, each algorithm is run independently for 30 times. Table VI compares the four improved algorithms, the proposed ICPO, and the original CPO for the three common indicators: the optimal value, the standard deviation and the average value. Figure 7 demonstrates the convergence curves of each comparative algorithm, and the maximum number of iterations of each algorithm in the convergence curve is 1000.

TABLE VI
EXPERIMENTAL DATA UNDER DIFFERENT ORIGINAL ALGORITHMS (CEC2020)

		ICPO	CPO	NTDO	DFPSO	NSSA	GMPBSA
F1	Minimum	4.4453×10^{2}	9.3671×10^{2}	4.0604×10^{6}	5.3811×10^{6}	1.3402×10^{5}	5.5685×10^{3}
	Standard	7.0683×10^{3}	8.6176×10^{3}	1.3659×10^{9}	8.9487×10^{6}	1.7662×10^{6}	1.7191×10^{5}
	Average	**7.3325×10^{3}**	7.8077×10^{3}	6.2552×10^{8}	1.5502×10^{7}	1.2258×10^{6}	1.2340×10^{5}
F2	Minimum	2.0768×10^{3}	2.6666×10^{3}	3.1392×10^{3}	2.1827×10^{3}	2.2929×10^{3}	3.6054×10^{3}
	Standard	1.9466×10^{2}	2.5691×10^{2}	5.2050×10^{2}	6.6907×10^{2}	4.1130×10^{2}	2.8901×10^{2}
	Average	**2.5570×10^{3}**	3.2385×10^{3}	3.9910×10^{3}	2.9427×10^{3}	2.9047×10^{3}	4.3317×10^{3}
F3	Minimum	7.5573×10^{2}	7.6893×10^{2}	8.5606×10^{2}	7.5351×10^{2}	7.7026×10^{2}	7.9121×10^{2}
	Standard	6.3090×10^{0}	8.4978×10^{0}	3.4044×10^{1}	1.8479×10^{1}	4.8204×10^{1}	1.1491×10^{1}
	Average	**7.6805×10^{2}**	7.8430×10^{2}	9.1183×10^{2}	8.0742×10^{2}	8.6253×10^{2}	8.1345×10^{2}
F4	Minimum	1.9044×10^{3}	1.9057×10^{3}	1.9093×10^{3}	1.9071×10^{3}	1.9041×10^{3}	1.9063×10^{3}
	Standard	9.5039×10^{-1}	1.1203×10^{0}	1.8473×10^{1}	1.2422×10^{0}	2.9244×10^{0}	1.4272×10^{0}
	Average	**1.9064×10^{3}**	1.9084×10^{3}	1.9247×10^{3}	1.9098×10^{3}	1.9096×10^{3}	1.9093×10^{3}
F5	Minimum	4.4031×10^{3}	6.0871×10^{3}	1.1596×10^{5}	4.9551×10^{5}	2.7893×10^{5}	5.9553×10^{3}
	Standard	2.5944×10^{4}	3.0154×10^{4}	1.4265×10^{5}	1.5589×10^{5}	1.5227×10^{5}	2.8530×10^{4}
	Average	**2.4110×10^{4}**	3.9137×10^{4}	2.5742×10^{5}	7.1698×10^{5}	4.7517×10^{5}	3.4794×10^{4}
F6	Minimum	1.6065×10^{3}	1.6151×10^{3}	1.7808×10^{3}	1.8420×10^{3}	1.6085×10^{3}	1.7476×10^{3}
	Standard	2.0454×10^{1}	2.1497×10^{1}	2.3513×10^{2}	1.7687×10^{2}	1.6138×10^{2}	9.5728×10^{1}
	Average	**1.6337×10^{3}**	1.6424×10^{3}	2.1739×10^{3}	2.0877×10^{3}	1.8739×10^{3}	1.9118×10^{3}
F7	Minimum	2.7807×10^{3}	3.0850×10^{3}	2.2043×10^{4}	1.0306×10^{5}	1.7826×10^{4}	2.9672×10^{3}
	Standard	3.1079×10^{2}	2.2695×10^{3}	2.8101×10^{4}	8.0289×10^{4}	5.5312×10^{4}	2.0949×10^{3}
	Average	**3.3823×10^{3}**	4.3505×10^{3}	6.4033×10^{4}	2.5153×10^{5}	9.7811×10^{4}	4.9878×10^{3}
F8	Minimum	2.3001×10^{3}	2.3001×10^{3}	2.3135×10^{3}	2.3124×10^{3}	2.3032×10^{3}	2.3050×10^{3}
	Standard	2.7419×10^{-1}	8.5559×10^{-1}	1.0422×10^{3}	9.0256×10^{2}	2.7038×10^{0}	1.4024×10^{3}
	Average	**2.3004×10^{3}**	2.3005×10^{3}	2.7903×10^{3}	2.6028×10^{3}	2.3095×10^{3}	2.9456×10^{3}
F9	Minimum	2.8439×10^{3}	2.8598×10^{3}	2.8613×10^{3}	2.7523×10^{3}	2.5104×10^{3}	2.8595×10^{3}
	Standard	1.2532×10^{1}	1.5421×10^{1}	3.3586×10^{1}	4.5182×10^{1}	1.0002×10^{2}	9.8036×10^{0}
	Average	2.8657×10^{3}	2.8926×10^{3}	2.9177×10^{3}	2.8779×10^{3}	**2.8616×10^{3}**	2.8818×10^{3}
F10	Minimum	2.9024×10^{3}	2.9143×10^{3}	2.9174×10^{3}	2.9279×10^{3}	2.9139×10^{3}	2.9143×10^{3}
	Standard	2.9546×10^{1}	3.2613×10^{1}	3.1242×10^{1}	1.6794×10^{1}	3.7157×10^{1}	3.2035×10^{1}
	Average	2.9628×10^{3}	2.9648×10^{3}	2.9837×10^{3}	2.9904×10^{3}	2.9712×10^{3}	**2.9557×10^{3}**

Note: The best performance values of the algorithm under different test functions are shown in bold.



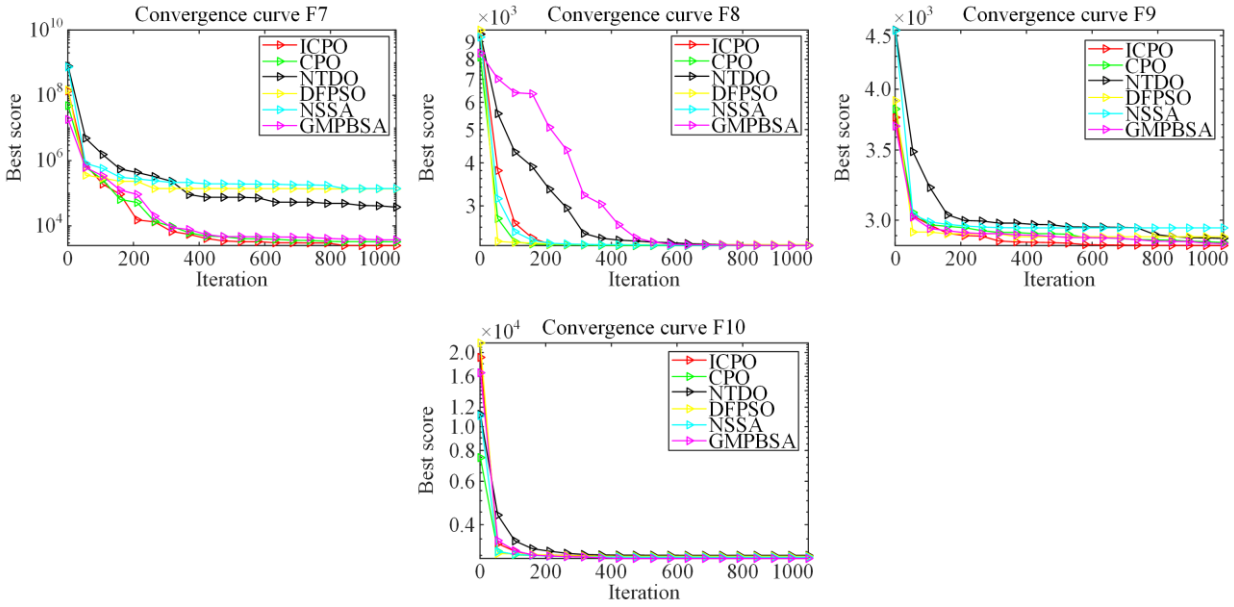


Fig. 7. Convergence curves of each improved algorithm under the cec2020 test sets.

As shown in Table VI, the proposed ICPO achieves the optimal average values for F1–F8 under the cec2020 test functions, while F9 and F10 are optimized best by NSSA and GMPBSA, respectively. Thus, ICPO attains the best average values in 8 out of 10 functions, demonstrating its excellent optimization search performance.

From Fig. 7, it can be seen that ICPO achieves optimal convergence accuracy and speed on functions F1–F3 and F5–F9. For functions F4 and F10, although its convergence accuracy is not the highest, the convergence speed remains highly competitive. Overall, the improvement proposed in this paper make the advantages of ICPO evident, significantly improving the overall performance of CPO.

In summary, the comparison of the proposed ICPO with other improved algorithms on the cec2020 test functions highlights its strong ability to find optimal solutions quickly and accurately.

2) Wilcoxon Rank Sum Test Analysis (cec2020)

As with the cec2017 test functions, Wilcoxon rank sum test is applied to the cec2020 test functions [59] to evaluate differences in optimization performance between the algorithms.

In Table VII, the values with p-value being more than 0.5 are shown in bold. As can be seen, compared with CPO, only F1, F8, and F10 functions show no significant difference, while the remaining functions exhibit significant differences. Compared with DFPSO, NSSA, and GMPBSA, ICPO has only one function without a significant difference, with the rest showing significant differences. Compared with NTDO, all functions differ significantly. These results indicate that, across the cec2020 test functions, ICPO demonstrates significant improvement over most other enhanced algorithms, confirming the effectiveness of the proposed improvements.

TABLE VII
WILCOXON RANK SUM TEST ANALYSIS (CEC2020)

	CPO	NTDO	DFPSO	NSSA	GMPBSA
F1	**9.8231×10^{-1}**	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.8249×10^{-9}
F2	1.6132×10^{-10}	3.0199×10^{-11}	8.6844×10^{-3}	6.5486×10^{-4}	3.0199×10^{-11}
F3	2.6695×10^{-9}	3.0199×10^{-11}	6.7220×10^{-10}	8.9934×10^{-11}	3.0199×10^{-11}
F4	5.0922×10^{-8}	3.0199×10^{-11}	2.6099×10^{-10}	2.1540×10^{-6}	1.0702×10^{-9}
F5	1.7649×10^{-2}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	2.4157×10^{-2}
F6	4.5671×10^{-2}	3.0199×10^{-11}	3.0199×10^{-11}	1.4733×10^{-7}	3.0199×10^{-11}
F7	9.5139×10^{-6}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	8.3520×10^{-8}
F8	**4.2896×10^{-1}**	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F9	7.0881×10^{-8}	7.7725×10^{-9}	**1.9579×10^{-1}**	4.3584×10^{-2}	8.1975×10^{-7}
F10	**6.6273×10^{-1}**	5.0842×10^{-3}	3.5923×10^{-5}	**2.2823×10^{-1}**	**4.2039×10^{-1}**

Note: Bold entries indicates the values with insignificant p-value in the non-parametric test.

3) Performance Comparison Between ICPO and Other Swarm Intelligence Optimization Algorithms (cec2020)

In order to reflect the performance of ICPO more comprehensively, this subsection compares the performances of ICPO, CPO, WOA, DBO, COA, GWO, and BOA under the cec2020 test functions. The number of function dimension is 20, the maximum number of iterations is 500, and the number of populations is 30.

From Table VIII, it can be seen that ICPO achieves the best average optimal fitness value under the cec2020

test functions except F8, demonstrating excellent optimality finding performance under the cec2020 test functions.

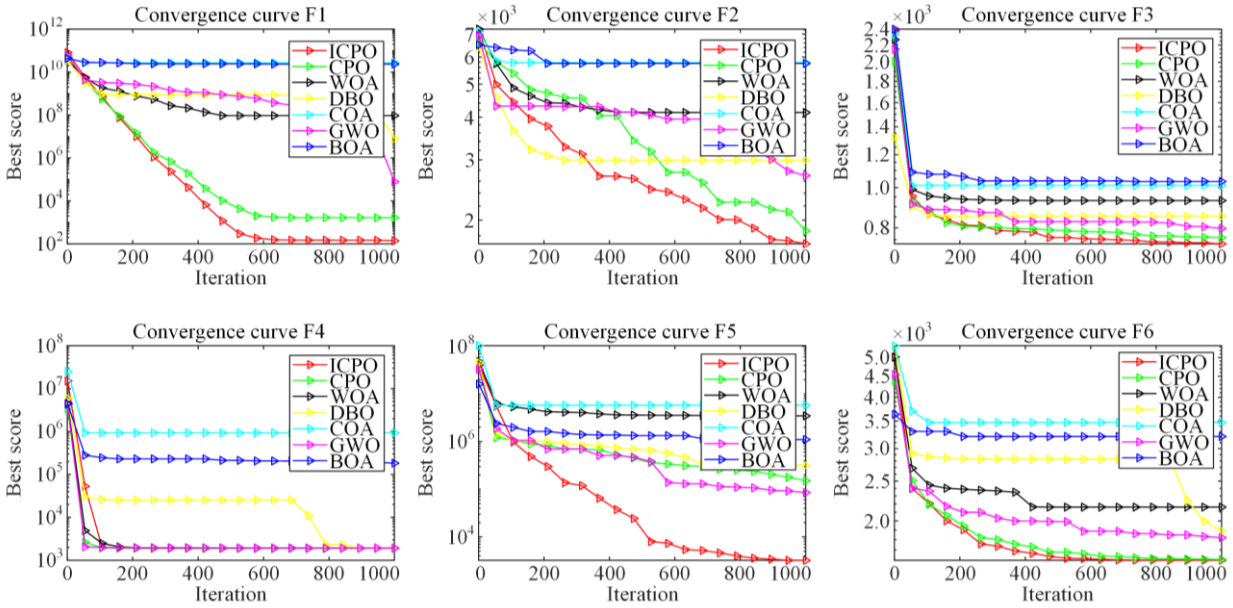
From Fig. 8, it can be seen that ICPO leads to significantly improved convergence speed and accuracy when compared to other optimization algorithms under the cec2020 test functions F1–F3 and F5–F10. For F4, although its convergence accuracy is not the best, its convergence speed is still satisfactory. Therefore, ICPO shows clear advantages under the convergence graph of the cec2020 test functions, confirming the effectiveness of the improvement.

In summary, the proposed ICPO achieves excellent optimization performance and high competitiveness and effectiveness on solving the three indexes and the convergence curve on the 10 test functions in cec2020.

TABLE VIII
EXPERIMENTAL DATA UNDER DIFFERENT IMPROVED ALGORITHMS (CEC2020)

		JCPO	CPO	WOA	DBO	COA	GWO	BOA
F1	Minimum	5.6471×10^{2}	1.3728×10^{3}	4.0313×10^{8}	6.7435×10^{3}	1.8093×10^{10}	3.5002×10^{5}	1.4922×10^{10}
	Standard	5.4831×10^{3}	9.3979×10^{3}	5.5346×10^{8}	6.0280×10^{7}	4.9192×10^{9}	9.7933×10^{8}	4.9060×10^{9}
	Average	$\mathbf{6.0091\times10^{3}}$	9.2609×10^{3}	1.2909×10^{9}	5.4033×10^{7}	2.9923×10^{10}	8.5010×10^{8}	2.5192×10^{10}
F2	Minimum	2.0113×10^{3}	2.8295×10^{3}	3.2376×10^{3}	2.2867×10^{3}	4.8943×10^{3}	1.7727×10^{3}	4.8942×10^{3}
	Standard	2.3876×10^{2}	2.5327×10^{2}	5.6667×10^{2}	6.0375×10^{2}	$3.4258\mathrm{E}\times10^{2}$	7.1476×10^{2}	2.5706×10^{2}
	Average	$\mathbf{2.5871\times10^{3}}$	3.4061×10^{3}	4.4690×10^{3}	3.4571×10^{3}	5.6416×10^{3}	2.9297×10^{3}	5.6675×10^{3}
F3	Minimum	7.5013×10^{2}	7.6483×10^{2}	8.7277×10^{2}	7.5935×10^{2}	9.5498×10^{2}	7.5519×10^{2}	9.5583×10^{2}
	Standard	7.3003×10^{0}	7.9776×10^{0}	5.4172×10^{1}	4.0130×10^{1}	2.6948×10^{1}	1.7879×10^{1}	2.4729×10^{1}
	Average	$\mathbf{7.6821\times10^{2}}$	7.8239×10^{2}	9.6289×10^{2}	8.2933×10^{2}	1.0284×10^{3}	7.8625×10^{2}	1.0027×10^{3}
F4	Minimum	1.9040×10^{3}	1.9047×10^{3}	1.9135×10^{3}	1.9092×10^{3}	1.1928×10^{4}	1.9054×10^{3}	9.5657×10^{4}
	Standard	1.0319×10^{0}	1.0603×10^{0}	9.5732×10^{2}	8.3674×10^{1}	4.5984×10^{5}	1.1070×10^{2}	5.2792×10^{5}
	Average	$\mathbf{1.9065\times10^{3}}$	1.9081×10^{3}	2.7090×10^{3}	1.9513×10^{3}	6.6916×10^{5}	1.9577×10^{3}	6.5623×10^{5}
F5	Minimum	3.9555×10^{3}	6.2887×10^{3}	6.1036×10^{4}	6.5940×10^{4}	4.6082×10^{5}	9.2406×10^{4}	8.2425×10^{5}
	Standard	1.2707×10^{4}	3.1838×10^{4}	2.7628×10^{6}	1.0442×10^{6}	3.4787×10^{6}	8.2957×10^{5}	2.4097×10^{6}
	Average	$\mathbf{1.4929\times10^{4}}$	3.7010×10^{4}	3.3341×10^{6}	1.2197×10^{6}	5.3584×10^{6}	1.0039×10^{6}	3.7631×10^{6}
F6	Minimum	1.6075×10^{3}	1.6108×10^{3}	2.0706×10^{3}	1.7427×10^{3}	2.6194×10^{3}	1.6767×10^{3}	2.8461×10^{3}
	Standard	2.1678×10^{1}	3.0100×10^{1}	3.3646×10^{2}	2.5436×10^{2}	3.1216×10^{2}	1.8445×10^{2}	2.7063×10^{2}
	Average	$\mathbf{1.6321\times10^{3}}$	1.6445×10^{3}	2.6615×10^{3}	2.2398×10^{3}	3.2744×10^{3}	2.0453×10^{3}	3.2672×10^{3}
F7	Minimum	2.8986×10^{3}	2.9023×10^{3}	1.3747×10^{5}	2.7292×10^{4}	3.6130×10^{5}	5.3155×10^{4}	2.6319×10^{5}
	Standard	3.9214×10^{2}	1.0249×10^{3}	1.3581×10^{6}	7.4354×10^{5}	1.6451×10^{6}	5.9817×10^{5}	2.5896×10^{6}
	Average	$\mathbf{3.5203\times10^{3}}$	3.9591×10^{3}	1.7737×10^{6}	4.6922×10^{5}	2.1327×10^{6}	4.2197×10^{5}	2.0684×10^{6}
F8	Minimum	2.3001×10^{3}	2.2881×10^{3}	2.3601×10^{3}	2.3000×10^{3}	4.3675×10^{3}	2.3120×10^{3}	2.7446×10^{3}
	Standard	5.6723×10^{-1}	2.3456×10^{0}	1.9316×10^{3}	1.0208×10^{3}	8.1659×10^{2}	1.3080×10^{3}	5.7622×10^{2}
	Average	2.3005×10^{3}	$\mathbf{2.3001\times10^{3}}$	4.3831×10^{3}	2.9423×10^{3}	5.7913×10^{3}	3.0431×10^{3}	3.8950×10^{3}
F9	Minimum	2.8493×10^{3}	2.8644×10^{3}	2.8960×10^{3}	2.8891×10^{3}	3.1962×10^{3}	2.8201×10^{3}	3.1808×10^{3}
	Standard	1.0400×10^{1}	1.2690×10^{1}	6.6265×10^{1}	6.1426×10^{1}	9.9801×10^{1}	4.0185×10^{1}	1.2002×10^{2}
	Average	$\mathbf{2.8746\times10^{3}}$	2.8930×10^{3}	3.0111×10^{3}	2.9972×10^{3}	3.4004×10^{3}	2.8829×10^{3}	3.3965×10^{3}
F10	Minimum	2.9144×10^{3}	2.9058×10^{3}	3.0392×10^{3}	2.9148×10^{3}	3.5026×10^{3}	2.9426×10^{3}	4.3998×10^{3}
	Standard	3.0202×10^{1}	2.6066×10^{1}	9.2418×10^{1}	6.4321×10^{1}	9.1287×10^{2}	2.6789×10^{1}	5.4992×10^{2}
	Average	$\mathbf{2.9616\times10^{3}}$	2.9634×10^{3}	3.1446×10^{3}	2.9919×10^{3}	5.2896×10^{3}	2.9882×10^{3}	5.4529×10^{3}

Note: The best performance values of the algorithm under different test functions are shown in bold.



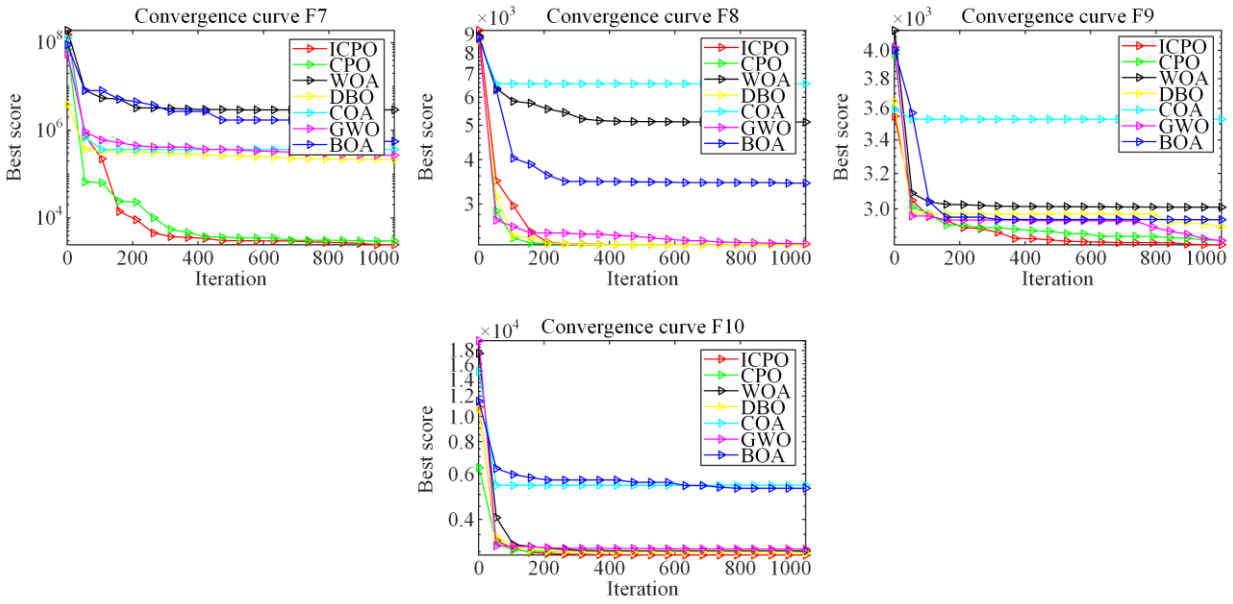


Fig. 8. Convergence curves of each algorithm under the cec2020 test sets.

4) Wilcoxon Rank Sum Test Analysis (cec2020)

This subsection employs the Wilcoxon rank sum test under the cec2020 test functions to further validate the effectiveness of ICPO. Table IX compares the Wilcoxon rank sum test results of ICPO with CPO, WOA, DBO, COA, GWO, and BOA. The algorithms run independently for 30 times under the cec2020 test functions with dimension of 20. It can be seen that ICPO shows significant differences from CPO for the vast majority of functions, with only a few exceptions. Compared with WOA, COA and BOA, ICPO exhibits significant differences across all test functions. Compared with DBO and GWO, only one function shows no significant difference, while the remaining functions do. These results indicates that the proposed ICPO in this paper is generally more competitive than the other algorithms, showing significant differences in most cases.

TABLE IX
WILCOXON RANK SUM TEST ANALYSIS (CEC2020)

	CPO	WOA	DBO	COA	GWO	BOA
F1	**2.7071×10^{-1}**	3.0199×10^{-11}	1.2057×10^{-10}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F2	1.2057×10^{-10}	8.1527×10^{-11}	3.1967×10^{-09}	3.0199×10^{-11}	**1.9579×10^{-1}**	3.0199×10^{-11}
F3	7.1186×10^{-9}	3.0199×10^{-11}	8.1527×10^{-11}	3.0199×10^{-11}	4.6390×10^{-5}	3.0199×10^{-11}
F4	5.5611×10^{-4}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	1.1567×10^{-7}	3.0199×10^{-11}
F5	1.9527×10^{-3}	3.0199×10^{-11}	3.6897×10^{-11}	3.0199×10^{-11}	4.9752×10^{-11}	3.0199×10^{-11}
F6	5.8737×10^{-4}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F7	2.5974×10^{-5}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F8	**6.7350×10^{-1}**	3.0199×10^{-11}	4.9752×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}
F9	1.1077×10^{-6}	3.0199×10^{-11}	3.3384×10^{-11}	3.0199×10^{-11}	1.2967×10^{-1}	3.0199×10^{-11}
F10	**2.3399×10^{-1}**	3.0199×10^{-11}	**9.3341×10^{-2}**	3.0199×10^{-11}	1.1674×10^{-5}	3.0199×10^{-11}

Note: Bold entries indicates the values with insignificant p-value in the non-parametric test.

V. UAV PATH PLANNING

Using the DEM map of Christmas Island, Australia, this paper simulates UAV path planning for a power inspection task. Various threat factors, such as fire hazards, terrain obstacles, and radar detection zones, are to create terrains of varying complexity. In such environments, the UAV must navigate from the starting point to the target inspection site while accounting for obstacles and operational constraints such as fuel consumption, climbing capability, and turning restrictions. To ensure safe and efficient flight, the path planning algorithm must optimize the flight path [60], considering factors such as flight distance, altitude, threat avoidance, and path smoothness, thereby achieving the optimal route for the power inspection task. The mathematical expression is given as follows:

$$F(X_i^{'}) = \sum_{k=1}^{4} b_k F_k(X_i^{'}) \tag{21}$$

where $X_i^{'}$ is the decision variable of a list of n terminal points (x, y, z); b_k is the weight of each cost function; and F_k is the kth cost function.

A. Voyage Length Costs

To reduce fuel consumption, the UAV's flight path during power inspection should be minimized, with path length calculated using the Euclidean distance of each segment to ensure efficient task completion. The path length cost for a particular path is calculated as:

$$F_1(X_i^{'}) = \sum_{j=1}^{n-1} \left\| \boldsymbol{P}_{ij} \boldsymbol{P}_{i,j+1} \right\| \tag{22}$$

where $\boldsymbol{P}_{ij}$ and $\boldsymbol{P}_{i,j+1}$ represent path nodes in the search map; and the Euclidean distance between two nodes is expressed as $\left\| \boldsymbol{P}_{ij} \boldsymbol{P}_{i,j+1} \right\|$.

B. Threat Costs

In power inspection missions, in addition to the optimal path length, path planning must also ensure that the UAV avoids threat areas (such as radar detection, air defense systems, or inclement weather). Due to the complexity of threat modeling, the threat area is simplified into a cylinder with a fixed radius to simulate the flight environment of the UAV. This ensures its safe bypass of these no-fly zones and successful completion of the power inspection task. Its image is described as Fig. 9.

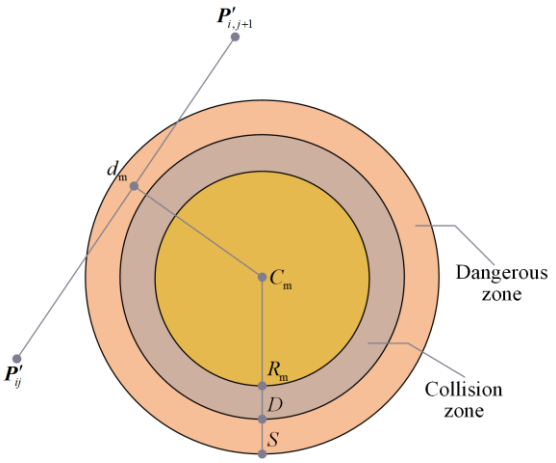


Fig. 9. Threat cost.

Let M be the set of threatening obstacles, each obstacle is modelled as a cylinder with the center C_m and radius R_m. The diameter of the UAV is D; and the vertical distance between the adjacent path nodes and the origin is d_m. The danger area S is affected by positioning accuracy and flight environment. When the static environment and GPS signal are good, S may be tens of meters. In contrast, in the presence of moving objects or poor GPS signals, S may exceed 100 m. The threat cost for a path is calculated as follows:

$$\begin{cases} F_2(J_i) = \sum_{j=1}^{n-1} \sum_{m=1}^{M} T_m \ (P_{i,j} P_{i,j+1}) \\ T_m(P_{i,j} P_{i,j+1}) = \begin{cases} 0, \ \text{if} \ d_m > S+D+R_m \\ (S+D+R_m) - d_m, \ \text{if} \ D+R_m \leqslant d_m \leqslant S+D+R_m \\ \infty, \ \text{if} \ d_m \leqslant D+R_m \end{cases} \end{cases} \tag{23}$$

In power inspection tasks, when the UAV flies outside the dangerous area, there is no collision risk and the threat is zero. As the UAV approaches the danger zone, the threat increases with decreased distance (d_m). If the drone enters a collision area, such as near a power facility or high-voltage power line, a collision will occur, causing the mission to fail and the path's threat value set to infinity. Therefore, path planning needs to avoid these dangerous areas to ensure flight safety.

C. Height Costs

The flight height of UAV is usually constrained by the maximum altitude h_{max} and the minimum altitude h_{min}. The height cost calculation formula is as follows:

$$H_{ij} = \begin{cases} \left| h_{ij} - \dfrac{(h_{max} + h_{min})}{2} \right|, \ \text{if} \ h_{min} \leqslant h_{ij} \leqslant h_{max} \\ \infty, \ \text{otherwise} \end{cases} \tag{24}$$

$$F_3(X_i) = \sum_{j=1}^{n} H_{i,j} \tag{25}$$

D. Smoothing Costs

The flight angle control of the UAV mainly consists of horizontal steering angle and vertical pitch angle. These two parameters must conform to the actual angle constraints of the UAV, otherwise the trajectory planning model will not be able to generate a feasible flight path, as shown in Fig. 10.

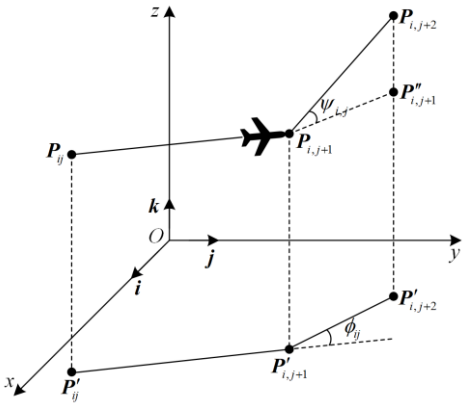


Fig. 10. Calculation of turn climbing angle.

The horizontal steering angle ϕ_{ij} is the angle between two consecutive path segments projected on the horizontal plane O_{xy}, and is calculated as:

$$\phi_{ij} = \arctan\left(\frac{\left\|\boldsymbol{P}'_{ij}\boldsymbol{P}'_{i,j+1} \times \boldsymbol{P}'_{i,j+1}\boldsymbol{P}'_{i,j+2}\right\|}{\boldsymbol{P}'_{ij}\boldsymbol{P}'_{i,j+1} \times \boldsymbol{P}'_{i,j+1}\boldsymbol{P}'_{i,j+2}}\right) \tag{26}$$

The vertical pitch angle ψ_{ij} is the angle between two consecutive path segments projected on the vertical axis, calculated as:

$$\psi_{ij} = \arctan\left(\frac{z_{i,j+1} - z_{ij}}{\left\|\boldsymbol{P}'_{ij}\boldsymbol{P}'_{i,j+1}\right\|}\right) \tag{27}$$

The smooth costing formula is given as follows:

$$F_4(X_i) = a_1 \sum_{j=1}^{n-2} \phi_{ij} + a_2 \sum_{j=1}^{n-1} \left|\psi_{ij} - \psi_{i,j-1}\right| \tag{28}$$

where a_1 and a_2 are penalty coefficients of horizontal steering angle and vertical pitch angle, respectively.

E. Experimental Verification

In power inspection tasks, the path planning of UAV is faced with the influence of complex terrain and threat objects, and traditional mathematical methods struggle to effectively solve the global optimal path. Therefore, meta-heuristic algorithm is particularly suitable for this type of problem because of its simple, flexible and non-derivative characteristics. To solve the path planning problem in power inspection, this paper applies the ICPO algorithm to the path planning experiments based on the DEM map of Christmas Island, Australia.

Multiple threat objects (represented by blue bars) are set up in the experimental environment, and two terrains of different complexity are simulated. Ten waypoints are selected, and the flight starting point is (10, 10, 200) and the end point is (400, 400, 150), as shown in Fig. 11. The experimental platform is developed in Windows 10 64-bit operating system, Intel(R) Core(TM) i3-8100 CPU@3.60GHz 16GB MATLAB R2022a.

The six different improved algorithms including ICPO, CPO, NTDO, DFPSO, NSSA, and GMPBSA, are compared using a population size of 500 and a maximum number of iterations of 100. From the comparative analysis of the topographic map ad three-dimensional graph (Figs. 12 and 13) and the convergence curve (Fig. 14), it can be seen that the ICPO algorithm not only generates the shortest path and the lowest fitness value but also produces a smoother trajectory with fewer inflection points. Compared with other algorithms, it reduces UAV flight resistance during power inspection and provides a more efficient inspection path.

Therefore, ICPO demonstrates strong performance in UAV path planning for power inspection, effectively handling complex terrain and threat objects while optimizing the flight path.

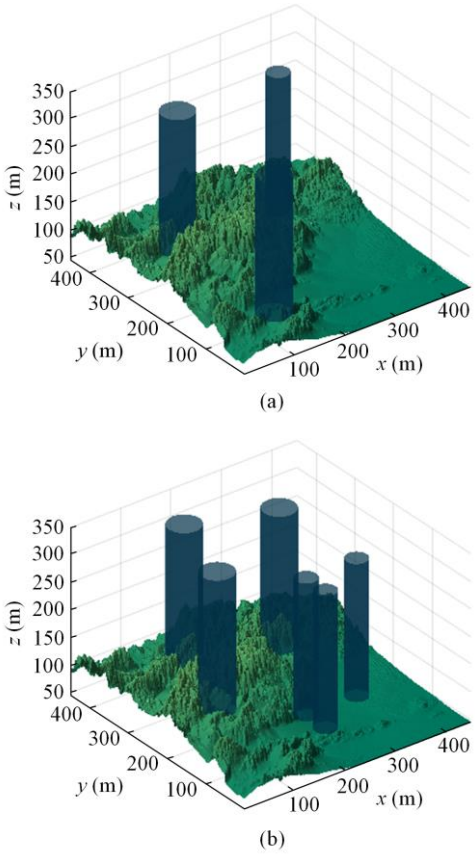


Fig. 11. Topographic map of different complexities. (a) Three threat objects. (b) Six threat objects.

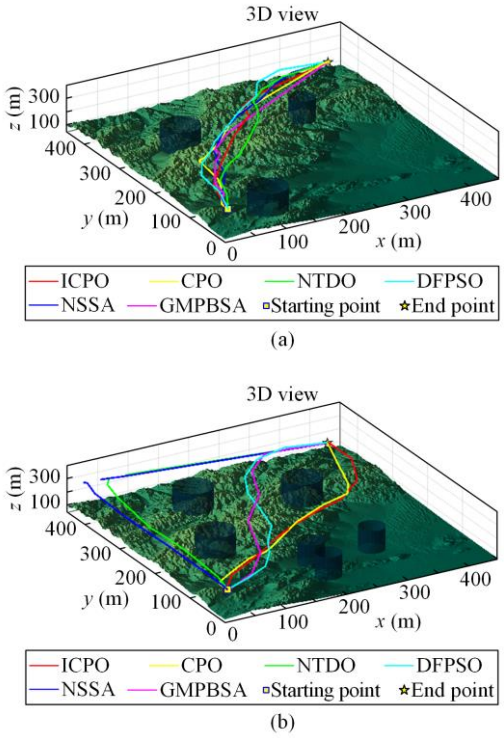


Fig. 12. The three dimensional graph comparison of each algorithm. (a) Three-dimensional terrain map of three threat objects. (b) Three-dimensional terrain map of six threat objects.

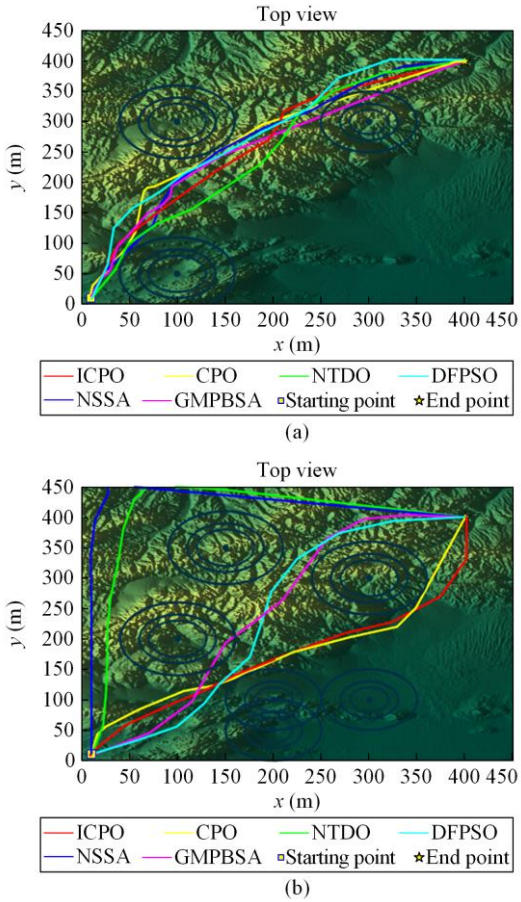


Fig. 13. Comparison of top view of each algorithm. (a) Top view of three threat objects. (b) Top view of six threat objects.

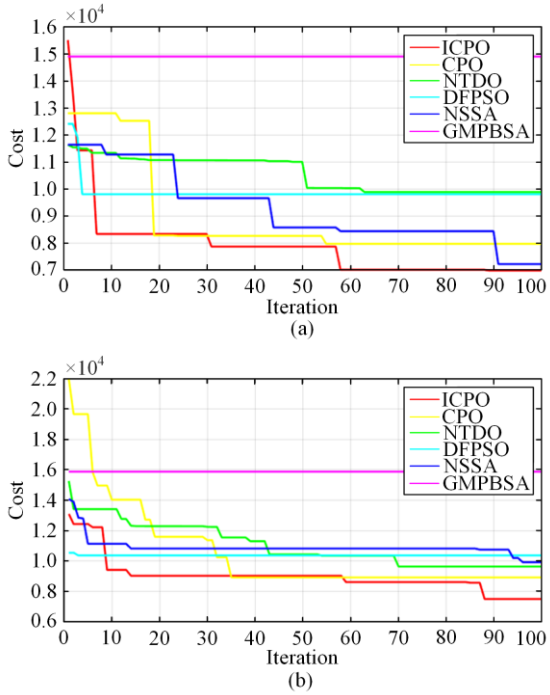


Fig. 14. Comparison of convergence curves of each algorithm. (a) Convergence curves of topographic map for three threat objects. (b) Convergence curves of topographic map for six threat objects.

VI. Conclusion

To address the challenge of UAV path planning for power inspection, a new path planning algorithm, ICPO, is proposed in this paper. The algorithm is designed to generate feasible and safe paths, and through an integrated cost function, ICPO simultaneously considers the key constraints of optimality, feasibility, and safety. The innovation of ICPO lies in its integration of the UAV's motion constraints with the environmental conditions of tasks such as power inspection, optimizing the path to ensure it aligns with the drone's physical characteristics and can be effectively executed in complex environments. First, the ICPO algorithm is tested on the cec2017 and cec2020 datasets, and is compared with 10 population intelligent optimization algorithms. The results show that ICPO performs best in terms of convergence accuracy and optimization performance. Then, ICPO is applied to UAV path planning for power inspection. Using the DEM map of Christmas Island in Australia and comparing two power inspection scenarios, the results show that ICPO produces the best path in both scenarios, verifying its effectiveness in power inspection.

Although the ICPO algorithm makes significant contributions to UAV path planning and power inspection tasks, it still has limitations. In particular, power system equipment maintenance and fault detection often require multiple UAVs to operate collaboratively. Future research will focus on developing multi-UAV collaborative path planning strategies in power systems.

Acknowledgment

Not applicable.

Authors' Contributions

Huanlong Zhang and Chenglin Guo: proposing the improved CPO for UAV path planning and writing the manuscript. Denghui Zhai and Yanfeng Wang: designing the fitness function and optimization framework. Heng Liu, Fuguo Chen, and Dan Xu: implementing dynamic constraints and obstacle avoidance. All authors read and approved the final manuscript.

Funding

This work is supported by the National Natural Science Foundation of China (No. 62102373, No. 62273243, and No. 62473341); Henan Province Key R&D Project (No. 241111210400); and Joint Fund Key Project of science and Technology R&D Plan of Henan Province (No. 235200810022).

Availability of Data and Materials

Not applicable.

DECLARATIONS

Competing interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

AUTHORS' INFORMATION

Huanlong Zhang received the Ph.D. degree from the School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai, China, in 2015. He is currently an associate professor with the Zhengzhou University of Light Industry, Zhengzhou, China. His research interests include signal processing, nonlinear systems, pattern recognition, machine learning, and computer vision.

Chenglin Guo received the master's degree from Zhengzhou University of Light Industry in 2025. Her research field is swarm intelligence algorithms and path planning.

Denghui Zhai received the Master degree from Shenyang Ligong University, Shenyang, China, in 2010. He is currently pursuing the Ph.D. degree with the College of Electrical Engineering, Zhejiang University, Hangzhou, China. He is a senior engineer currently serving at China Electrical Equipment Research Institute of Science and Technology Co., Ltd. His research interests include the micro-grid operation control and power artificial intelligence.

Yanfeng Wang received his Doctor of Science degree and is now a professor and doctoral supervisor. He is an expert with the State Council's Special Allowance, an outstanding expert of Henan Province, an academic and technical leader of Henan Province, and currently serves as the vice president of Zhengzhou Light Industry University and the director of the Key Laboratory of Informatization Electrical Appliances of Henan Province. His main research fields include the construction and analysis of memristor models, synchronous control of nonlinear circuits, design and application of dynamic cascaded circuits.

Heng Liu received the master's degree and is now a senior electrical engineer. He has been engaged in the research and development of high voltage products and related fields of intelligence equipment. Has served as Pinggao Group technology center designer, chief engineer and other positions, currently director and general manager of Guangzhou Pinggao Electric Power Technology Co., Ltd.

Fuguo Chen received his Ph.D in electrical engineering from Xi'an Jiaotong University. He is a professor level senior engineer and master's supervisor, a high-level talent in Hangzhou, a senior expert in live detection at the China Electric Power Equipment Management Association, and currently serves as the vice president of the Yangtze River Delta (Hangzhou) Innovation Research Institute of Dongfang Electric.

Dan Xu received the Master's degree from the School of Electronic Information Engineering, Tianjin University, China in 2015. He is the current manager of the Artificial Intelligence R&D Department at Xuji Electric Science and Technology Research Institute. His research interests include machine learning, deep learning, and computer vision.

REFERENCES

[1] L. Fahmani and S. Benhadou, "Optimizing 2D path planning for unmanned aerial vehicle inspection of electric transmission lines," *Scientific African*, vol. 24, Jun. 2024.

[2] M. Zhan and Y. Zhao, "Research on three-dimensional path planning of UAV-AGV cooperative electric power patrol based on improved firefly algorithm," in *Fourth International Conference on Mechanical, Electronics, and Electrical and Automation Control (METMS 2024)*, Xi'an, China, Jun. 2024. pp. 1994-1999.

[3] K. Li, X. Yan, and Y. Han, "Multi-mechanism swarm optimization for multi-UAV task assignment and path planning in transmission line inspection under multi-wind field," *Applied Soft Computing*, vol. 150, Jan. 2024.

[4] A. Caballero, F. J. Roman-Escorza, and I. Maza *et al.*, "A multi-UAV route planning method for fast inspection of electric power transmission lines," in *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, Crete, Greece, Jun. 2024. pp. 835-842.

[5] S. Singh, M. M. H. Mukit, and V. H. Chu, "Integrating robotic process automation into power system control: a framework for adaptive change management and operational resilience," *Power System Protection and Control*, vol. 53, no. 3, pp. 119-133, Aug. 2025. (in Chinese)

[6] W. Zheng, Y. Li, and S. Li *et al.*, "Research on task path planning method for substation inspection robot and UAV based on genetic algorithm," in *Ninth International Symposium on Sensors, Mechatronics, and Automation System (ISSMAS 2023)*, Nanjing, China, Jun. 2024. pp. 1120-1125.

[7] X. Yu, N. Jiang, and X. Wang *et al.*, "A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning," *Expert Systems with Applications*, vol. 215, Apr. 2023.

[8] S. Sundarraj, R. V. K. Reddy, and M. B. Basam *et al.*, "Route planning for an autonomous robotic vehicle employing a weight-controlled particle swarm-optimized Dijkstra algorithm," *IEEE Access*, vol. 11, pp. 92433-92442, Aug. 2023.

[9] Z. Zhu, Y. Yin, and H. Lü, "Automatic collision avoidance algorithm based on route-plan-guided artificial potential field method," *Ocean Engineering*, vol. 271, Mar. 2023.
[10] R. K. Dewangan and P. Saxena, "Three-dimensional route planning for multiple unmanned aerial vehicles using salp swarm algorithm," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 35, no. 7, pp. 1059-1078, Oct. 2023.
[11] Q. Liang, L. Zhou, and H. Zhou *et al.*, "An adaptive weight A* algorithm applied to UAV path planning," in *the International Conference on Natural Computation*, *Fuzzy Systems and Knowledge Discovery*, Guiyang, China, Jan. 2021. pp. 239-246.
[12] L. Asqui, V. Andaluz, and J. Sánchez *et al.*, "Path planning based in algorithm rapidly-exploring random tree RRT," *Advanced Science Letters*, vol. 24*,* no. 11, pp. 8831-8836, Nov. 2018.
[13] H.-Y. Hsieh, K. H. Chen, and C.-J. Shieh *et al.*, "Urban mobile robot routing using fast search random tree method (RRT) in obstacle environments," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 46*,* no. 10, Sep. 2024.
[14] D. Zhang, Y. Xian, and J. Li *et al.*, "UAV path planning based on chaos ant colony algorithm," in *2015 International Conference on Computer Science and Mechanical Automation (CSMA)*, Hangzhou, China, Oct. 2015. pp. 81-85.
[15] L. Tan, H. Zhang, and Y. Liu *et al.*, "An adaptive Q-learning based particle swarm optimization for multi-UAV path planning," *Soft Computing*, vol. 28, no. 13, pp. 7931-7946, Jul. 2024.
[16] J. Shi, L. Tan, and H. Zhang *et al.*, "Adaptive multi-UAV path planning method based on improved gray wolf algorithm," *Computers and Electrical Engineering*, vol. 104, Dec. 2022.
[17] K. Sethanan and W. Neungmatcha, "Multi-objective particle swarm optimization for mechanical harvester route planning of sugarcane field operations," *European Journal of Operational Research*, vol. 252*,* no. 3, pp. 969-984, Aug. 2016.
[18] S. Maske, "An efficient genetic algorithm based method for finding k shortest and longest paths in an undirected graph," *Power System Protection and Control*, vol. 51, no. 1, pp. 6-12, Feb. 2023. (in Chinese)
[19] Y. Ji, X. Zhao, and J. Hao, "A novel UAV path planning algorithm based on double-dynamic biogeography-based learning particle swarm optimization," *Mobile Information Systems*, vol. 2022*,* no. 1, Jan. 2022.
[20] L. Liu, Y. Lu, and B. Yang *et al.*, "Research on a multi-strategy improved sand cat swarm optimization algorithm for three-dimensional UAV trajectory path planning," *World Electric Vehicle Journal*, vol. 15, no. 6, May 2024.
[21] X. Jiang, Z. Lin, and T. He *et al.*, "Optimal path finding with beetle antennae search algorithm by using ant colony optimization initialization and different searching strategies," *IEEE Access*, vol. 8, pp. 15459-15471, Jan. 2020.
[22] M. D. Phung and Q. P. Ha, "Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization," *Applied Soft Computing*, vol. 107, Aug. 2021.
[23] X. Wang, H. Zhao, and T. Han *et al.*, "A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem," *Applied Soft Computing*, vol. 78, pp. 240-260, May 2019.
[24] J. Yao, Y. Sha, and Y. Chen *et al.*, "IHSSAO: an improved hybrid salp swarm algorithm and Aquila optimizer for UAV path planning in complex terrain," *Applied Sciences*, vol. 12, no. 11, Jun. 2022.
[25] X. Yu and W. Luo, "Reinforcement learning-based multi-strategy cuckoo search algorithm for 3D UAV path planning," *Expert Systems with Applications*, vol. 223, Aug. 2023.
[26] M. Abdel-Basset, R. Mohamed, and M. Abouhawwash, "Crested porcupine optimizer: a new nature-inspired metaheuristic," *Knowledge-Based Systems*, vol. 284, Jan. 2024.
[27] F. Coppola, M. Maestrini, and F. Berrilli *et al.*, "First report of giardia duodenalis infection in the crested porcupine," *International Journal for Parasitology*: *Parasites and Wildlife*, vol. 11, pp. 108-113, Apr. 2020.
[28] L. Monetti, A. Massolo, and A. Sforzi *et al.*, "Site selection and fidelity by crested porcupines for denning," *Ethology Ecology & Evolution*, vol. 17, no. 2, pp. 149-159, Feb. 2005.
[29] F. Akram, O. Ilyas, and A. Haleem, "Food and feeding habits of Indian crested porcupine in pench tiger reserve, Madhya Pradesh, India," *Ambient Science*, vol. 4, no. 1, pp. 62-66, Jan. 2017.
[30] S. Lovari, M. T. Corsini, and B. Guazzini *et al.*, "Suburban ecology of the crested porcupine in a heavily poached area: a global approach," *European Journal of Wildlife Research*, vol. 63, no. 1, pp. 1-10, Jan. 2017.
[31] M. Sarwar, "Some observations on species composition and deterioration of crop plantations and forest flora by porcupines in consort with control techniques," *International Multidisciplinary Research Journal*, vol. 8, pp. 8-14, Apr. 2018.
[32] L. Lazzeri, C. Senini, and E. Mori, "Interspecific aggressions between crested porcupines and roe deer," *Animals*, vol. 10, no. 4, Apr. 2020.
[33] E. Mori, I. Maggini, and M. Menchetti, "When quills kill: the defense strategy of the crested porcupine hystrix cristata," *Mammalia*, vol. 78, no. 2, pp. 229-234, Oct. 2014.
[34] S. Chou and R. Overfelt, "Tensile deformation and failure of north American porcupine quills," *Materials Science and Engineering*, vol. 31, no. 8, pp. 1729-1736, Dec. 2011.
[35] S. Alrasheed, "Impulse, momentum, and collisions," in Principles of Mechanics: Fundamental University Physics, Berlin, Germany: Springer, 2019, pp. 73-82.
[36] J. O. Agushaka, A. E. Ezugwu, and L. Abualigah *et al.*, "Efficient initialization methods for population-based metaheuristic algorithms: a comparative study," *Archives of Computational Methods in Engineering*, vol. 30, no. 3, pp. 1727-1787, Dec. 2023.
[37] L. Zhang and X. Chen, "Social coevolution and sine chaotic opposition learning chimp optimization algorithm for feature selection," *Scientific Reports*, vol. 14, no. 1, Jul. 2024.

[38] M. Ma, J. Wu, and Y. Shi *et al.*, "Chaotic random opposition-based learning and Cauchy mutation improved moth-flame optimization algorithm for intelligent route planning of multiple UAVs," *IEEE Access*, vol. 10, pp. 49385-49397, May 2022.

[39] M. Wang, J. Wang, and X. Li *et al.*, "Harris hawk optimization algorithm based on cauchy distribution inverse cumulative function and tangent flight operator," *Applied Intelligence*, vol. 52, no. 10, pp. 10999-11026, Jan. 2022.

[40] A. Annamraju and S. Nandiraju, "Coordinated control of conventional power sources and PHEVs using jaya algorithm optimized PID controller for frequency control of a renewable penetrated power system," *Protection and Control of Modern Power Systems*, vol. 4, no. 4, pp. 1-13, Oct. 2019.

[41] Y. Cheng, X. Lü, and S. Mao, "Optimization design of brushless DC motor based on improved JAYA algorithm," *Scientific Reports*, vol. 14, no. 1, Mar. 2024.

[42] S. Choudhary, S. Sugumaran, and A. Belazi *et al.*, "Linearly decreasing inertia weight PSO and improved weight factor-based clustering algorithm for wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 6661-6679, Oct. 2023.

[43] Y. Zhao, S. Bi, and H. Zhang *et al.*, "Dynamic weight and mapping mutation operation-based salp swarm algorithm for global optimization," *Applied Sciences*, vol. 13, no. 15, Aug. 2023.

[44] H. Zhang, C. Guo, and J. Zhang *et al.*, "An improved Tasmanian devil optimization algorithm based on sine-cosine strategy with dynamic weighting factors," *Cluster Computing*, vol. 27, no. 9, pp. 12875-12897, Jun. 2024.

[45] F. Rezaei, and H. R. Safavi, "Sustainable conjunctive water use modeling using dual fitness particle swarm optimization algorithm," *Water Resources Management*, vol. 36, no. 3, pp. 989-1006, Jan. 2022.

[46] J. Zhang, L. Li, and H. Zhang *et al.*, "A novel sparrow search algorithm with integrates spawning strategy," *Cluster Computing*, vol. 27, no. 2, pp. 1753-1773, Jun. 2024.

[47] Y. Zhang, "Backtracking search algorithm driven by generalized mean position for numerical and industrial engineering problems," *Artificial Intelligence Review*, vol. 56, no. 10, pp. 11985-12031, Apr. 2023.

[48] B. Dasu, S. Mangipudi, and S. Rayapudi, "Small signal stability enhancement of a large scale power system using a bio-inspired whale optimization algorithm," *Protection and Control of Modern Power Systems*, vol. 6, no. 4, pp. 1-17, Oct. 2021.

[49] W. Wang, Q. Zhang, and S. Guo *et al.*, "Hazard evaluation of goaf based on DBO algorithm coupled with BP neural network," *Heliyon*, vol. 10, no. 13, Jul. 2024.

[50] S. Osama, A. A. Ali, and H. Shaban, "A hybrid of information gain and a coati optimization algorithm for gene selection in microarray gene expression data classification," *Kafrelsheikh Journal of Information Sciences*, vol. 4, no. 1, pp. 1-16, Jun. 2023.

[51] J. Chen, G. He, and Y. Wang *et al.*, "Adaptive PID control for hydraulic turbine regulation systems based on INGWO and BPNN," *Protection and Control of Modern Power Systems*, vol. 9, no. 4, pp. 126-146, Jul. 2024.

[52] M. Badi, S. Mahapatra, and S. Raj, "Hybrid BOA-GWO-PSO algorithm for mitigation of congestion by optimal reactive power management," *Optimal Control Applications and Methods*, vol. 44, no. 2, pp. 935-966, Nov. 2023.

[53] M. Yu, J. Xu, and W. Liang *et al.*, "Improved multi-strategy adaptive grey wolf optimization for practical engineering applications and high-dimensional problem solving," *Artificial Intelligence Review*, vol. 57, no. 10, Sep. 2024.

[54] H. Parmaksiz, U. Yuzgec, and E. Dokur *et al.*, "Mutation based improved dragonfly optimization algorithm for a neuro-fuzzy system in short term wind speed forecasting," *Knowledge-Based Systems*, vol. 268, May. 2023.

[55] F. Wang, S. Feng, and Y. Pan *et al.*, "Dynamic spiral updating whale optimization algorithm for solving optimal power flow problem," *Journal of Supercomputing*, vol. 79, no. 17, pp. 19959-20000, Jun. 2023.

[56] M. Qaraad, A. Aljadania, and M. Elhosseini, "Large-scale competitive learning-based salp swarm for global optimization and solving constrained mechanical and engineering design problems," *Mathematics*, vol. 11, no. 6, Mar. 2023.

[57] Y. K. Saheed, B. F. Balogun, and B. J. Odunayo *et al.*, "Microarray gene expression data classification via Wilcoxon sign rank sum and novel grey wolf optimized ensemble learning models," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 20, no. 6, pp. 3575-3587, Aug. 2023.

[58] G. Li, T. Zhang, and C.-Y. Tsai *et al.*, "Modified crayfish optimization algorithm with adaptive spiral elite greedy opposition-based learning and search-hide strategy for global optimization," *Journal of Computational Design and Engineering*, vol. 11, no. 4, pp. 249-305, Jul. 2024.

[59] Y. Wang, N. Wang, and T. Gao *et al.*, "An improved prairie dog optimization algorithm integrating multiple strategies and its application," *Engineering Research Express*, vol. 6, no. 3, Aug. 2024.

[60] Y. Sun, Q. Yuan, and Q. Gao *et al.*, "A multiple environment available path planning based on an improved A* algorithm," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, Jul. 2024.