Journal of Network Intelligence ©2025 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **10**, Number **2**, May **2025**

An Improved Orca Predation Algorithm for Solving Optimization Problems (IOPA)

Yan-Jiao Wang, Yin-Dan Xiong

School of Electrical Engineering
Northeast Electric Power University, Jilin 132012, China
wangyanjiao1028@126.com, 13525789062@163.com

Kadambri Agarwal

Department of Computer Science & Engineering
ABES Engineering College, Ghaziabad 201009, India
kadambri_agarwal@rediffmail.com

*Corresponding author: Yin-Dan Xiong

Received April 3, 2024; revised August 28, 2024; accepted December 1, 2024.

Abstract. *The Orca Predation Algorithm (OPA) is a meta-heuristic algorithm that mimics the predatory behavior of killer whales. However, its effectiveness is limited by its lack of population diversity and poor convergence accuracy. Therefore, we propose a more advanced killer whale predation algorithm (IOPA). A new individual renewal strategy is proposed in the chase phase to meet the requirements of the evolutionary process in terms of convergence speed and diversity. Encircling prey uses a probabilistic model to select basis vectors, while movement step size is controlled by population diversity to allow real-time monitoring and complement of population diversity. The attack phase replaces average learning with adaptive learning. A novel mutation mechanism for learning from opposites is introduced to increase the likelihood of an individual escaping from a locally optimal solution. In addition, it introduces global crossover operations to improve the quality of information sharing. Finally, it also proposes a new individual restart strategy to avoid the problem of stagnant algorithm updates. The results for the 28 test functions at CEC2013 show that IOPA achieves the theoretical optimum on 9 functions. In addition, it has significant advantages in terms of convergence speed and convergence accuracy over the original OPA method and the other four algorithms.*
Keywords: Orca Predation Algorithm, Meta-heuristic algorithm, adaptive learning, global crossover operation, individual restart strategy

1. Introduction.

1.1. Related work.
Optimization difficulties are prevalent throughout various domains. The gradient-based optimization method is the most conventional approach for solving optimization problems [1]. However, it imposes stringent criteria for the objective function and relies significantly on the initial point's position for optimal accuracy. Additionally, it is unable to address non-differentiable problems. On the other hand, meta-heuristic algorithms like GA [2], DE [3], PSO [4], and GWO [5] possess significant adaptability and flexibility, enabling them to effectively address various types of optimization issues. Consequently, an increasing number of academics are engaged in the examination of meta-heuristic algorithms. Currently, meta-heuristic algorithms are the subject of two primary

study categories. (1) Proposed meta-heuristic methods for enhancing strategies. (2) A novel metaheuristic algorithm is proposed.

The proposed meta-heuristic algorithm for strategy enhancement primarily enhances the algorithm's performance by strengthening certain mechanisms [6]. Wu et al. 2024 enhanced the performance of the algorithm by adapting weight factors, probability updates, Gaussian mutation and proposed a neural network model on adaptive Fick's law algorithm. The experimental results verify that the model has the best performance [7]. The most notable algorithms include Yao et al. (2023), who has proposed an enhanced serpentine optimizer for practical engineering problems, which incorporates a novel dynamic mechanism and an adversarial learning strategy to augment the serpentine optimizer (ESO) [8]. Wu et al. in 2023 proposed a chaotic mapping based hay rolling optimization algorithm to enhance the overall performance of the algorithm by introducing chaotic mapping [9]. Similarly, Askari et al. (2021) have introduced an improved political optimizer, which incorporates a new location update strategy and adaptive parameters to strike a balance between algorithm exploration and development [10]. In the year 2021, Aala et al. introduced a hybrid social whale optimization algorithm that integrates the whale optimization algorithm and the social group optimization algorithm [11]. This algorithm leverages the combined exploration of the WOA and convergence capabilities of the SGO to achieve an optimal balance of equilibrium between exploration and development of competence has been successfully achieved [12, 13]. Deng et al. (2022) introduced a multi-strategy improved slime mold algorithm for solving complex optimization problems, introduced a novel search equation, and employed dynamic random search to optimize the system's overall performance [14]. Desuky et al. 2021 proposed an Archimedean Optimization Approach (EAOA) applied to a test set of classification performance, which improves the exploratory function of AOA by proposing a new parameter controlled by the individual step size, and enhances the classification performance [15]. Chen et al. (2023) introduced an enhanced Archimedes optimization algorithm in their study [16]. This algorithm incorporated the moderating factor and simplex approach, resulting in notable improvements in both convergence accuracy and convergence speed.

Currently, several novel meta-heuristic algorithms have been introduced; In 2022, Mohammad Dehghani et al. proposed a bio-inspired Coati Optimization Algorithm (COA) (COA) inspired by the predatory behavior of raccoons in nature, which simulates raccoons attacking and hunting iguanas as well as moving away from predators [17]. In 2022, Braik et al. investigated the behavior of great white sharks in searching for food in nature using their associated senses of hearing and smell, and proposed the White Shark Optimizer (WSO), which achieves continuity optimization by simulating the three behaviors of white sharks in searching for food [18]. In 2022, Zamani et al. introduced the Starling Calling Optimizer (SMO) [19], which simulates the dynamic flight patterns of bird flocks. The SMO algorithm aims to preserve population diversity and prevent local optimums by leveraging the synergistic effects of separation, diving, and rotation strategies. Jiang et al. introduced the Orca Predation Algorithm (OPA) in 2022 as an innovative meta-heuristic algorithm [20]. OPA addresses the optimization problem by emulating the predatory behavior of a collective of orcas. The algorithm represents the predation behavior of orcas using three distinct methods: expulsion, encirclement, and attack of prey. The parameters inside the algorithm are utilized to regulate the magnitude of expulsion and encirclement, hence controlling the exploitation of the algorithm and exploring the weights. As OPA has good performance. But there are also some flaws, so scholars have done a lot of research on them. The latest detailed studies are shown in Table 1.

Table 1. Recent research on OPA algorithms

Latest improved algorithm on OPA	Authors	Times	Main innovations and contributions
Opposition-based learning boosted orca predation algorithm with dimension learning. (IOPA) [21]	Hu et al.	October 2023	Dimensionality learning strategies and contrastive learning strategies are introduced to enhance the ability of the algorithm to jump out of the minimum and are applied for use in the case of multi-degree approximation of NURBS curves.
Multiple Strategies Boosted Orca Predation Algorithm for Engineering Optimization Problems. (LFOPA) [22]	Houssein et al.	April 2023	Integration of the Lévy flight strategy in the chase phase with a greedy selection strategy to enhance the OPA and application of the proposed new algorithm to engineering applications.
Modified orca predation algorithm. (mOPA) [23]	Emam et al.	April 2023	The introduction of the adversarial learning strategy and the Levy flight strategy into OPA improves the algorithmic Ade search efficiency as a way to improve the reliability of the hybrid system and reduce the minimum energy cost.

1.2. **Motivation and contribution.** The above new meta-heuristic algorithms provide new ideas for solving optimization problems, and more scholars have paid attention to the OPA algorithm and proposed improved algorithms. However, these algorithms still have the problems of insufficient convergence accuracy and speed. In light of this context, this study examines the orca predation algorithm and introduces a new enhanced Orca Predation Algorithm (IOPA), which encompasses the following key advancements and contributions:

(1) Improve the way of chasing prey. The weak learning part was replaced by the optimal individual learning, and the Cauchy step factor was used to increase the evolutionary direction to ensure that rapid convergence is achieved in the evolutionary process while meeting the requirements of diversity.

(2) Improve the way of encircling prey. The basis vector is selected in the probabilistic mode, the movement step size is controlled by population diversity, and the population diversity is monitored and supplemented in real-time.

(3) Improve the attack phase. Adaptive learning is used to replace average learning, and the mutation mechanism of individuals is improved, a new type of adversarial learning is adopted to expand the Search Agent's Exploration Space, and intergenerational cross-operation is introduced to improve the quality of solutions.

(4) A new individual restart strategy is proposed. When population diversity is seriously lacking in the evolutionary process, the Gaussian mutation operation is carried out for the poor group of individuals to motivate further global search of the algorithm. The results of the test on the CEC2013 test set show that compared with the other four better algorithms and the latest improved orca algorithm, the IOPA algorithm proposed in this paper has obvious advantages in other aspects such as convergence accuracy and convergence speed.

The remainder of this paper is shown below: Section 2 provides an overview of the fundamental ideas and sequential progression of the original OPA algorithm. Section 3 provides an in-depth analysis of the shortcomings of various aspects of the OPA algorithm and presents a proposed technique to address these limitations. This strategy is referred to as the Enhanced OPA Algorithm (IOPA). In this part, the simulation data and data analysis of the OPA algorithm, the IOPA algorithm, and other notable algorithms are presented on the CEC2013 test set. Section 5 summarizes the IOPA algorithm proposed in this paper.

2. **Orca predation algorithm.**

In the marine world, Orcas are fierce and intelligent top predators in the dolphin family, highly socialized, using sonar navigation for prey search, surroundings exploration, and social interaction, ultimately achieving prey predation. Inspired by the above behaviors, Jiang et al. proposed the Orca Predation Algorithm (OPA) in 2022. In the OPA algorithm, the individual represents the position information of the orca, and the fitness value represents the position of each orca individual, which mainly includes the chase stage and the attack stage. The pseudocode of the OPA algorithm is displayed in Algorithm 1.

Algorithm 1: OPA

Input: Population: N, Dimension of Optimization Problem: D, Minimum Boundary Value of Optimization Problem: lb, Maximum Boundary Value of Optimization Problem: ub, Probability of Chase Selection: $p1$, Probability of Injury or Death: $p2$, Probability of Driving Method Selection q, Maximum Number of Iterations: Max_iter

Output: The optimal solution and its fitness value

Variable initialization (N, D, lb, ub, $p1$, $p2$, q, Max_iter);
An initial population X containing N individuals is randomly generated within a defined domain;
The fitness value of each body X_i is calculated: $Pfit_i$;
while $t <= Max\_iter$ **do**
  Identify the best individuals: $xbest$;
  for X_i *in population* X **do**
    `// Chase phase`;
    if $rand > p1$ **then**
      $x^{t+1}_{chase,i} \leftarrow x^{t}_{chase,i}$ updated in accordance with repelling prey operations.
    else
      $x^{t+1}_{chase,i,k} \leftarrow x^{t}_{chase,i,k}$ updated in accordance with siege prey operation.
    end if
    $x^{t+1}_{chase,i} \leftarrow x^{t}_{chase,i}$ position update according to (3) in 2.1;
    `// Attack phase`;
    $x^{t+1}_{attack,i} \leftarrow x^{t}_{attack,i}$ updated in accordance with attack prey;
    $x^{t+1}_{i} \leftarrow x^{t+1}_{i}$ position update according to (2) in 2.2.
  end for
  $t = t + 1$;
end while
Outputs the global optimal solution;

2.1. **Chase phase.**

Biologically, when a pod of orcas comes across a group of prey, groups of orcas use sonar to collaborate in driving their prey to the surface before enclosing them in a controlled circle. The OPA algorithm simulates the above-mentioned chase process and establishes the operation of expulsion of prey and the operation of encircling the prey, each orca individual selects one of the operations through the parameter $p1$ to update its position, and the detailed strategies are as follows: when the random number and in $[0,1] < p1$, the orca individual performs the encirclement operation; Instead, carry out expulsion of prey operations. where $p1$ can be set as a fixed constant or as a random number within $[0,1]$. The OPA algorithm recommends that $p1$ be set to 0.9.

(1) Expulsion of prey operations

In the expulsion of prey operations of the OPA algorithm, each orca individual realizes the pre-update of its position through velocity according to Equation (1).

$$x_{chase,i}^{t+1} = x_i^t + v_{chase,i}^t \tag{1}$$

Among them, $x_{chase,i}^{t+1}$ indicates the position of the i-th orca after driving away its prey, and $v_{chase,i}^t$ indicates the prey driving speed of the i-th orca. Two methods for calculating the prey-repellent speed are designed in OPA, and each orca individual selects one of the prey-repellent velocity calculation methods through a fixed parameter q, as follows: the random number in $[0,1]$, when $rand > q$, the orca individual uses method 2 to calculate the prey repellent velocity; Otherwise, choose option one. The OPA algorithm recommends that q be set to 0.9. Compared with method 2, there are relatively more individuals who use method 1 to calculate the speed of expulsion of prey, so method 1 and method 2 are called large-population prey expulsion method and small-population prey expulsion method, respectively.

1. Method 1: large-population prey expulsion

For large populations, the OPA algorithm is designed to calculate the rate of expulsion of prey as shown in Equation (2), so that the individual orca and the center of the orca population as a whole are close to the prey.

$$v_{chase,1,i}^{t+1} = a \times (d \times xbest^t - F \times (c \times x_i^t + b \times M^t)) \tag{2}$$

where M^t is the average position of the population calculated by Equation (3); F is a constant taking the value 2; a, b, and d are all random numbers ranging from $[0,1]$; c is calculated as shown in Equation (4).

$$M = \frac{\sum_{i=1}^{N} x_i^t}{N} \tag{3}$$

$$c = 1 - b \tag{4}$$

2. Method 2: small-population prey expulsion

For small populations, the OPA algorithm is designed to calculate the rate of expulsion of prey as shown in Equation (5), so that the orca individuals can quickly approach the prey.

$$v_{chase,2,i}^{t+1} = e \times xbest^t - x_i^t \tag{5}$$

Among them, $xbest^t$ represents the position of the most advantageous orca individual throughout the overall population of orcas; the parameter e is calculated according to Equation (6).

$$e = 2 \times rand \tag{6}$$

(2) Encircle prey operation

As can be seen from Section 2.1. (1), the OPA algorithm designed to drive away prey allows orcas to quickly approach their prey. In contrast, the OPA algorithm designs a prey encirclement operation as shown in Equation (7), in which the prey is surrounded by orcas in a controlled spherical enclosure.

$$x_{chase,i,j}^{t+1} = x_{r1,j}^t + u \times (x_{r2,j}^t - x_{r3,j}^t) \tag{7}$$

Among them, the $x_{chase,i,j}^{t+1}$ denotes spatial coordinates of the j-dimension of the i-th orca after the prey encirclement method is adopted; $r1$, $r2$ and $r3$ represent three distinct orcas that have been randomly chosen from the orca population, namely $r1, r2, r3 \in [1, N]$, the variable u is a randomly generated value derived from Equation (8).

$$u = 2 \times (rand - 0.5) \times \frac{Max\_iter - t}{Max\_iter} \tag{8}$$

where Max_iter is the maximum number of iterations and t is the current number of iterations.

(3) Location Updates

Orcas employ sonar technology to perceive the whereabouts of their prey and subsequently adapt their position in response. In the event that orcas detect a group of fish during a pursuit. They will replace the original position and continue the chase. Otherwise, they will stay in the original position, as shown in Equation (9).

$$x_{chase,i}^{t+1} = \begin{cases} x_{chase,i}^{t+1} & \text{if } f(x_{chase,i}^{t+1}) < f(x_i^t) \\ x_i^t & \text{else} \end{cases} \tag{9}$$

2.2. **Attack phase.** When the group of orcas surrounds their prey, the orcas will enter the enclosure in turn to attack the fish in the circle, slapping the fish with their tails to prey on the stunned fish, and adjust their final position through the sonar system. Simulating the above process, the OPA algorithm designs the prey attack stage and the position update stage, which are as follows.

(1) Attack prey

During the attack, the orcas enter the enclosure in turn to prey on the fish, and when they do so, they jump out of the enclosure so that the next orca can enter the enclosure to hunt. Simulating the above process, the OPA algorithm designs a way to attack prey to ensure that the orca individuals move to the fish and outside the enclosure at the same time, as shown in Equation (10).

$$x_{attack,i}^{t+1} = x_{chase,i}^{t} + g1 \times v_{attack,1,i}^{t} + g2 \times v_{attack,2,i}^{t} \tag{10}$$

where $x_{attack,i}^{t+1}$ indicates the position after the i-th orca individual of the t-generation attacks the prey; $g1$ and $g2$ are the acceleration coefficients, which are random numbers of $[0, 2]$ and $[-2.5, 2.5]$, respectively; $v_{attack,1,i}^{t}$ and $v_{attack,2,i}^{t}$ represent the speed at which the i-th orca preys on the fish and the rate at which it returns to the outside of the enclosure, respectively. Given the fact that the better individuals in the orcas are closer to the shoal of fish and that the fish are located approximately in the center of the better orcas, while the orcas are closer to each other when they randomly return to the outside of the enclosure, OPA designed velocity calculations as shown in Equation (11) and (12), respectively.

$$v_{attack,1,i}^{t+1} = \frac{(x_{first}^t + x_{second}^t + x_{third}^t + x_{four}^t)}{4} - x_i^t \tag{11}$$

$$v_{attack,2,i}^{t+1} = \frac{(x_{r1}^t + x_{r2}^t + x_{r3}^t)}{3} - x_i^t \tag{12}$$

where x_{first}^t, x_{second}^t, x_{third}^t and x_{four}^t represents the four individuals with the best position in the orca population; x_{r1}^t, x_{r2}^t and x_{r3}^t represents the three randomly selected orcas individuals during the chase phase $r1, r2, r3 \in [1, N]$, The variables $r1$, $r2$, and $r3$ are not equal to each other.

(2) Location updates

Unlike the position adjustment method in the chase phase, after the orca individual attacks the prey, if they sense that they are close to the fish, they will enter the enclosure to continue attacking the prey and hunting. Otherwise, the location is shifted depending on the physical condition of the orcas. The specific position adjustment method is shown in Equation (13).

$$x_i^{t+1} = \begin{cases} x_{attack,i}^{t+1} & \text{if } f(x_{attack,i}^{t+1}) < f(x_{chase,i}) \\ x\_new & \text{else} \end{cases} \tag{13}$$

wherein, $x_$new is the position of the orca after the transfer of the physical condition of the orca when it does not approach the fish after attacking the prey, as follows: a random number and is randomly generated, if $rand < p2$, it is considered that the orca is injured or killed, and the orca moves to the minimum boundary; Otherwise, the orca is considered healthy and returns directly to its original location. The updated formula is shown in Equation (14).

$$x\_new = \begin{cases} x_{chase,i,j}^{t+1} & \text{if } rand > p2 \\ lb(k) & \text{else} \end{cases} \tag{14}$$

where $p2$ represents the probability constant for assessing the health status of the orca, which is usually set to 0.005 , $x_{chase,i,j}^{t+1}$ represents the position of the orca at time t during the chase phase, and $lb(j)$ denotes the minimum value on the dimension of the optimization problem.

3. **Improved orca predation algorithm.** To improve the performance of the algorithm by improving inherent flaws of OPA. In this section, an improved Orca Predator Algorithm (IOPA) is proposed, and its flowchart is represented in Figure 1.

3.1. **Improved way to chase prey.** The OPA algorithm incorporates two distinct approaches for prey expulsion: the large-population prey expulsion method and the small-population prey expulsion method. Equation (2) demonstrates that in the prey expulsion mode of large populations, individuals simultaneously acquire knowledge from both the best-positioned individual in the population and from the center position. Learning from the optimal individual expedites the convergence of the algorithm. Additionally, the central position of all individuals integrates the evolutionary genes of all individuals, enabling communication with them to enhance a specific level of population diversity. However, due to the fixed nature of the central position, the potential for diversity supplementation is inherently constrained. Furthermore, in the case of superior individuals, the individual fitness value tends to surpass that of the central position. Consequently, the transition to the central position will impede the evolutionary progression of the superior individuals, consequently impeding the overall rate of population evolution.

It can be seen from Equation (1) and (5) that for the small-population prey expulsion method. The individual exclusively learns from the optimal individual with the aim of enhancing the algorithm's convergence speed. But in fact, the individual will eventually be migrated to $2 \times rand \times xbest^t$, that is a unified random perturbation carried out in all dimensions of the optimal individual, and due to the randomness of randomness, it is difficult for the migrated position to be better than the original position, that is, it causes invalid search. In summary, the primary objective of the prey expulsion phase in the design of the OPA algorithm is to enhance the algorithm's convergence speed, which is different from the small-population prey expulsion, and large-population prey expulsion also expects to maintain a certain population diversity. However, the aforementioned expulsion methods are unable to effectively enhance the convergence speed of the algorithm due to the irrationality of the update strategy design.

In view of the fact that the probability of orca individuals performing the expulsion of prey stage and the encirclement stage are 0.1 and 0.9, respectively. The encirclement stage has a high probability of execution; the population diversity can be well maintained

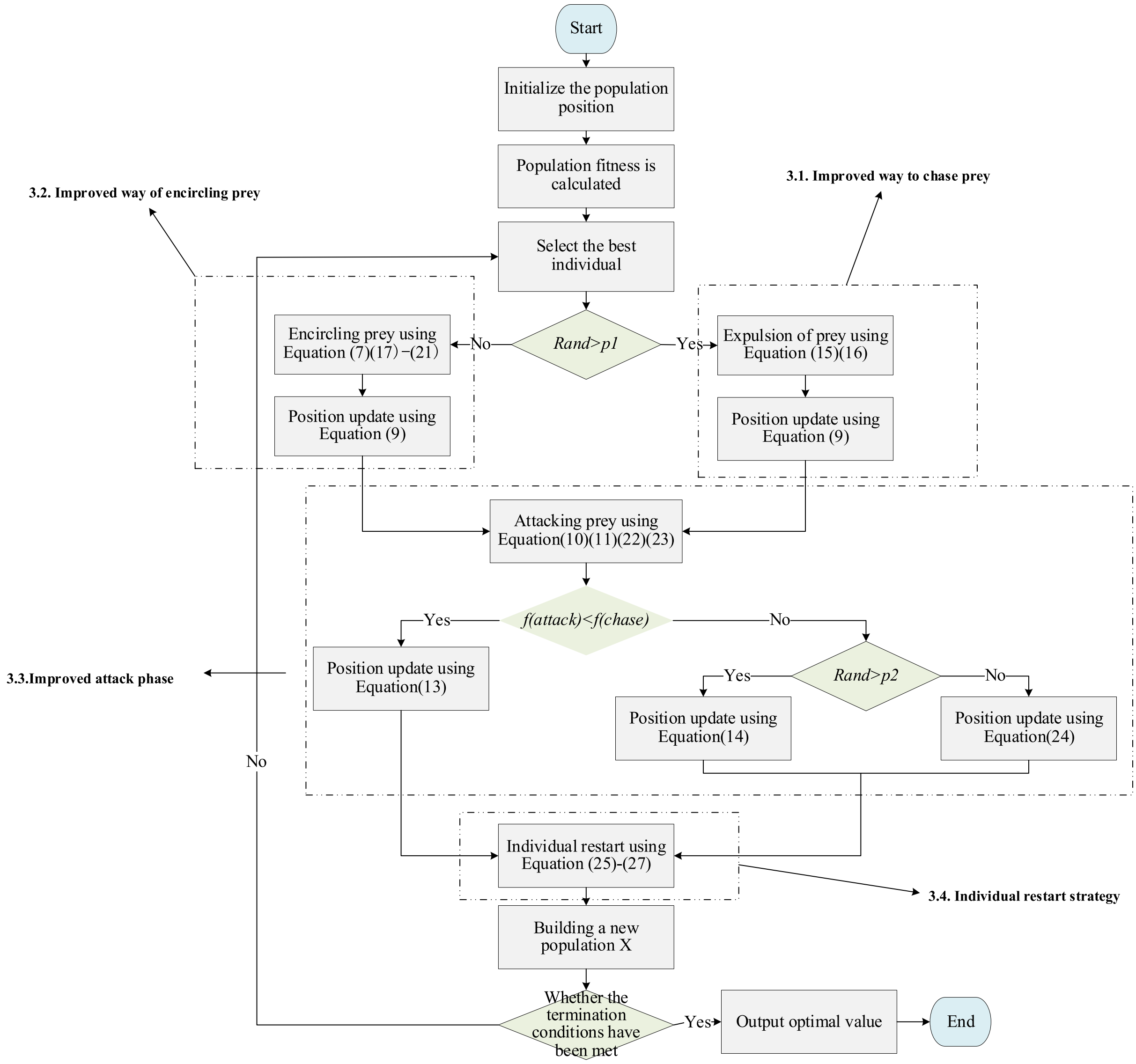


Figure 1. Flowchart of the IOPA algorithm

through communication with other individuals for global search. The expulsion of prey phase with a lower execution probability is primarily employed to accelerate convergence speed without reducing the population diversity too much. In addition, the probability of individuals expelling prey from large populations and small populations was only 0.09 and 0.01, respectively, and the probability of expulsion of prey from small populations was too low to be designed separately. In summary, this section designs a new type of prey repellent as shown in Equation (15).

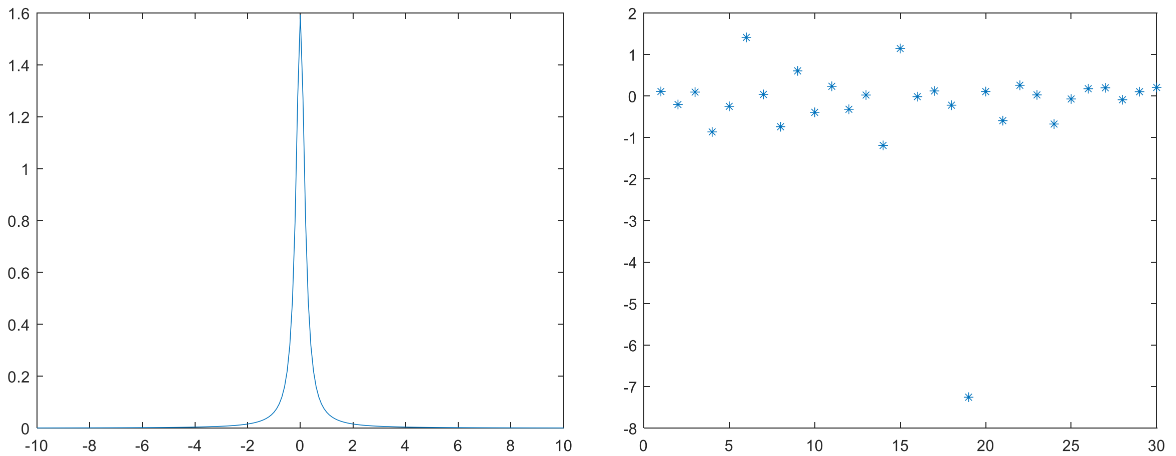
$$x_i^{t+1} = x_i^t + Cauchy \times (xbest^t - x_i^t) \tag{15}$$

In Equation (16) $Cauchy$ represents a random number that adheres to the Cauchy distribution.

$$Cauchy = Cauchy\_rand(\mu F, \gamma, num) \tag{16}$$

where μF is the positional parameter of the Cauchy peak, and its value is set to 0; γ is the scale parameter, and its value is set to 0.2; num represents the number of Cauchy random numbers that are generated, and its value is set to the same dimension as the problem being solved.

In summary, in the new expulsion of prey method designed in this section. Individuals all learn only from the globally optimal individual, and the speed of the algorithm improves significantly. In addition, the step size of the change in each dimension is completely different from that in the original prey-driving approach. Because of the special functional properties of the Cauchy function, its distribution is shown in Figure 2(a): there is a smaller peak at the origin but longer tails at the ends. Therefore, Cauchy is adopted as the step factor for the novel way of driving prey. A Cauchy random number of the same number of dimensions as the dimension of the problem being solved is generated by Equation (16). The value distribution of the generated random numbers is shown in Figure 2(b). The step factor is not the same in all dimensions, and it is highly likely that there will be an unusually variable value in one dimension. The Cauchy distribution property makes most of the individuals will be concentrated near the local optimal solution, and some small number of solutions are distributed far away. It makes the algorithm take care of both exploitation and exploration. It can increase the possibility of the algorithm to converge to the global optimum region, and also can maintain the population diversity to some extent.



(A) Cauchy function distribution (B) D-dimensional Cauchy random numbers

FIGURE 2. Cauchy random number distribution

3.2. **Improved way of encircling prey.** It can be seen from Equation (7) that the prey enveloping method designed by the OPA algorithm can be understood as "basis vector + step × difference vector", and the in-depth analysis shows that the above method has the following shortcomings. First, the optimization problem has a certain correlation in each dimension, and the individual synthesizes the evolutionary information in each dimension in the evolutionary process. While the original prey encirclement method randomly selects the individual as the basis vector and the difference vector in each dimension, although it can provide some population diversity because the internal connection between the dimensions is completely severed, but severely limits the likelihood that new individuals will outperform the original individuals. After the position update in Section 2.2, it is difficult for new individuals to be retained, that is, the diversity is not effectively replenished, and it is difficult to retain it in the iterative population. Second, from Equation (10), it can be seen that as the number of iterations increases, the step size factor will gradually decrease. In the early stage of iteration, a larger search step size can accelerate the convergence speed to the optimal solution. In the later stage of iteration, a small step size brings a very small perturbation. This refined search approach mitigates the risk of persons missing the optimal solution as a consequence of excessively large search steps and is more conducive to searching for the optimal solution. However, as the population converges to the local optimum, that is, the individuals are relatively similar to each other, it is difficult for a small search step size to deviate from the local optimum. In summary, to mitigate the potential for the algorithm to become trapped in a local optimum and better supplement the population diversity, the selection of the basis vector and the setting of the step factor in the original prey encirclement method as shown in Equation (7) are improved, as shown in Equation (17) and (18).

$$x_{r1,k}^{t} = \begin{cases} x_{r1,k}^{t} & \text{if } rand > pb \\ x_{i,k}^{t} & \text{else} \end{cases} \tag{17}$$

Among them, *pb* is the selection probability of the control basis vector, can be assigned a value of 0.5 to attain favorable outcomes. Additionally, it can be adjusted based on the specific optimization problem.

$$u = \begin{cases} 2 \times (rand - 0.5) & \text{if } divers < 10^{-5} \\ 2 \times (rand - 0.5) \times \frac{Max\_iter - t}{Max\_iter} & \text{else} \end{cases} \tag{18}$$

where $divers$ denotes population diversity, measured by the ratio between the population hypervolume and the D-dimensional volume of the search space restriction, as shown in Equation (19). The larger it is, the better the population diversity; On the contrary, the worse.

$$divers = \sqrt{\frac{V_P}{V_{Max}}} \tag{19}$$

where, V_{Max} and V_P are shown in Equation (20) and (21), respectively.

$$V_{Max} = \ln\left(1 + \prod_{j=1}^{D} |ui - li|\right) \tag{20}$$

$$V_P = \sqrt{\prod_{j=1}^{D} y_j} \tag{21}$$

where D is the dimension of the optimization problem, ui and li denote the upper and lower limits on the dimension of the problem, respectively. The variable y_i represents the disparity between the highest and lowest values observed in the j-dimension of the population.

In conclusion, the improved prey encirclement method has the following advantages. First, the basis vector is selected according to probability pb between the individual itself and random individuals in all dimensions. This implies that the new individual will perturb near itself in some dimensions, and introduce the genes of any other individual in the other dimensions. The larger the *pb*, the greater the possibility that the new individual will absorb the evolutionary information of other individuals. Compared with the original method of a completely random selection of base vectors in all dimensions, the improved method retains the connection of evolutionary information between some dimensions based on supplementing population diversity and increases the likelihood of a new individual being superior to the original individual so that the diversity can be effectively supplemented and participate in the subsequent evolution. Second, comparing Equation (18) and (10), it is observed that in cases where the population diversity is extremely low, the influence of the number of iterations in the step factor is canceled, that is, the value of the step factor is amplified, and if the difference vector part is completely 0, there is no change before and after improvement. On the contrary, if the difference vector part is not completely 0, a greater perturbation can potentially enhance the likelihood of the individual deviating from the local peak to a certain degree. It is well known that even when population diversity is low, the probability that individuals will be the same in all dimensions is extremely low, that is, the difference vector part is still not exactly 0 in most dimensions. That is, the step factor shown in Equation (18) is more conducive to supplementing population diversity and avoiding the algorithm falling into the local optimum.

3.3. **Improved attack phase.** The attack phase of the OPA algorithm includes the way of attacking the prey and the location update operation. To improve the performance of the algorithm in an all-around way, the way it attacks prey and the location update operation are comprehensively improved. The pseudo code is shown in Algorithm 2.

Algorithm 2: Improved attack phase

$v_{attack,1,i}^{t+1} \leftarrow$ update according to Equation (11);
$v_{attack,2,i}^{t+1} \leftarrow$ update according to Equation (22);
$v_{attack,i}^{t+1} \leftarrow$ Position update according to Equation (10);
for $j = 1 : D$ **do**
  $x_{attack,i}^{t+1} \leftarrow$ Perform the global crossover strategy according to Equation (23)
end for
Calculate the fitness value $f(x_{attactk}^{t+1})$;
if $f(x_{attactk}^{t+1}) < f(x_{chase}^{t+1})$ **then**
  $x_{attack,i}^{t+i} \leftarrow x_{attack,i}^{t+1}$
else
  if $rand() > p2$ **then**
    $x\_new_{i,k}^{t} \leftarrow$ update according to Equation (24)
  else
    $x\_new_{i,k}^{t} = x_{chase,i,k}^{t+1}$
  end if
end if

(1) Improved way to attack prey

Equation (11) shows that during the attack phase, each individual changes its position according to the speed of the prey ($v_{attack,1,i}^{t}$) and its subsequent return to the outside of the enclosure ($v_{attack,2,i}^{t}$). Both $v_{attack,1,i}^{t}$ and $v_{attack,2,i}^{t}$ can be understood as the product of the acceleration coefficients and the corresponding difference vector, and the shortcomings of each part are analyzed as follows. (1) For the difference vector part, $v_{attack,1,i}^{t}$ shows the movement of individuals to the center of the optimal four individuals, which is conducive to the rapid convergence of the algorithm. $v_{attack,2,i}^{t}$ shows that individuals communicate and learn from each other with three random individuals, which increases the diversity of the population. The fitness's of the optimal 4 individuals are all relatively similar but different, while the fitness values of the 3 random individuals may vary widely. If one of the three random individuals has a good fitness value, learning from the three random individuals to the same extent will inevitably reduce the algorithm's evolutionary speed. (2) For the acceleration coefficient, the accelerations $g1$ and $g2$ for predation on fish and return to the outside of the paddock control the importance of the speed of predation and return to the outside of the paddock, respectively. When the $g1$ range is larger, and the

$g2$ range is smaller, the learning experience provided by good orcas may be larger, and the algorithm may converge faster, but there may be a dramatic loss of population diversity, also increasing the risk of falling into a local optimum. On the contrary, if the $g1$ range is smaller and the $g2$ range is larger, different orcas share more experience with each other, which can better maintain the population diversity but slows down the convergence of the algorithm to some extent. Nevertheless, the values assigned to the random numbers $g1$ and $g2$ in the OPA algorithm fall inside the intervals $[0, 2]$ and $[-2.5, 2.5]$, respectively, which is evidently illogical.

Based on the above analysis. To better balance the convergence speed of the algorithm with population diversity. The following two improvements are made: On the one hand, for the difference vector part of $v^t_{attack,2,i}$, it no longer learns uniformly from each individual, but comprehensively considers the fitness advantages and disadvantages of the three individuals, as shown in Equation (22). On the other hand, the ranges of $g1$ and $g2$ are adjusted to $[0, 2]$ and $[-2, 2]$, respectively, to ensure that they are of equal importance.

$$v^{t+1}_{attack,2,i} = \frac{f(x^t_{j1})x^t_{j1} + f(x^t_{j2})x^t_{j2} + f(x^t_{j3})x^t_{j3}}{f(x^t_{j1}) + f(x^t_{j2}) + f(x^t_{j3}) + \varepsilon} - x^t_i \tag{22}$$

Where, $f(\cdot)$ represents the adaptation value of the corresponding individual, ε is a very small number that prevents the denominator from being zero. In contrast to Equation (12), the calculation of $v^{t+1}_{attack,2,i}$ is related to the individual's own fitness value. Based on its own fitness, adaptive superiority learning is performed to adjust the speed of orcas returning to the paddock. The overall adaptive learning ability of the algorithm can be enhanced. In addition, the randomly selected individuals are more different from each other. The adaptive learning method is better able to identify the differences between individuals, thus ensuring that individuals return to the paddock with a greater return speed to enhance the convergence speed of the algorithm.

In addition, inspired by the DE algorithm, the quality of the solution can be improved by using the global crossover strategy, and the crossover frequency is a key factor: frequent crossover makes the exchange of information frequent, which also accelerates the exchange of misinformation, and if each generation crosses over, the new individuals produced contain more original evolutionary information, resulting in lower evolutionary efficiency and slow evolutionary process.

Given this, the following generational crossover strategy is proposed, that is, when there are odd iterations, the crossover method as shown in Equation (23) is used to adjust the position after the attack. Conversely, that is, in the case of an even number of iterations, no crossover occurs.

$$x^{t+1}_{attack,i,j} = \begin{cases} x^{t+1}_{attack,i,j} & \text{if } rand < CR \| j = n_j \\ x^{t+1}_{chase,i,j} & \text{else} \end{cases} \tag{23}$$

The crossover probability, denoted as CR, is assigned a value of 0.8; The number n_j is randomly chosen from the range $[1, D]$.

In conclusion. when the population attack phase is completed, it is easier to skip the optimal solution position due to the imbalance between the attack speed of the prey and the return to the paddock speed, and through the generational crossing, the superior position of the orca in the chase stage is fully utilized under the condition of ensuring that the new evolutionary information is sufficient, and the original evolutionary information is not too much obtained. The crossover operation enables the population distribution more extensive in the solution space, which helps to guide the algorithm to search globally in the solution space, thus significantly improving the search efficiency and accuracy of the OPA algorithm.

(2) Improved location update

As can be seen from Section 2.2.2, different from the conventional position update method, when the orca individual produced after attacking the prey is not as good as the original one, the orca individual will not only stay in the original position but also have a certain probability of being transferred to the minimum boundary value $lb(j)$. In addition to the greedy adherence to the survival of the fittest in addition to the conventional position renewal method, the population is provided with an additional new species of population diversity. However, since the location of the optimal solution is generally not located at the boundary of the feasible domain, the population diversity provided by shifting to the minimum boundary of the feasible range usually does not work for the evolution of other individuals. Considering the opposing positions of individuals can not only provide more population diversity but also be of great help to the algorithm. Given this, when the orca individual produced after attacking the prey is inferior to the original individual and it is judged to be injured or killed, it is no longer transferred to the minimum boundary of the feasible range, but its dimensions are studied in opposition according to Equation (24).

$$x\_new_{i,j}^{t} = x_{i,j}^{t} + rand \times (rand \times (Max\_ub_j + Min\_lb_j - x_i^t) - x_{i,j}^{t})) \tag{24}$$

Where Max_ub_j and Min_lb_j denote the maximum and minimum values in the j-dimension of all individuals in the population, respectively.

In summary, when updating dead individuals, replacing the original fixed and unchanging boundary values with real-time changing boundary values increases the utilization of information in each dimension of the population. In addition, the individual moves the position of the individual to the opposite position of that individual through the novel dyadic learning. Compared with the boundary value, its fitness value is better. And it can help the individual to escape from the current position.

3.4. **Individual restart strategy.** Similar to other optimization algorithms, the OPA algorithm also suffers from the flaw of falling into a local optimum solution due to insufficient diversity. Inspired by the literature [24–26], this section proposes a new individual restart strategy to supplement population diversity, Specifically as shown in Figure 3. Determine whether the optimal value of the two generations has changed, and if so, A cumulative operation is then performed on the variable *unchanged_number* to record the number of consecutive times that the global optimum of the population does not change during the evolutionary process. Otherwise, zero out *unchanged_number*. When *unchanged_number* reaches the preset threshold T, *unchanged_number* will be reset to zero, and the individual restart strategy will be started as follows: Firstly, Sorting individuals in a population by fitness value. Then, for the 0.3 N individuals with the worst fitness value, each individual randomly selected *D_num* dimensions to be regenerated according to Equation (25) to form 0.3 N new individuals to directly participate in the next iteration.

$$Worse\_group' = x_{i,rand\_D}^{t} \times (1 + n \times z) \tag{25}$$

where *rand_D* is a randomly selected *D_num* dimension from D, set $D\_num = D/3$, n is the step factor, which is calculated as shown in Equation (26), and z is a Gaussian random number, which is calculated as shown in Equation (27).

$$n = 1 - (t-1)/(Max\_iter - 1) \tag{26}$$

$$z = gauss(\mu, \sigma^2) \tag{27}$$

where μ is the mean and σ^2 is the variance.

The individual restart strategy proposed in this section is different from other ways to supplement diversity, as follows: First, when judging whether the population is stagnant, existing methods are usually based on judging the Euclidean distance between the optimal individuals of the two previous and two subsequent iterations, while this section directly uses the change of optimal fitness to judge. It is well known that when the population falls into a local optimal peak, individuals in different positions may also have the same fitness value. At this time, the Euclidean distance that measures the position relationship cannot effectively determine that the algorithm has fallen into the local optimum. In contrast, the change of optimal fitness can effectively identify the algorithm that has fallen into the local optimum. In addition, compared with the additional calculation of Euclidean distance, the fitness calculated by the algorithm is used for judgment, resulting in a reduction in the algorithm's time complexity and saving time resources. Second, the existing methods generally adopt the method of random mutation in all dimensions when implementing the restart strategy for poor individuals, while this section only carries out Gaussian mutations in some dimensions, which retains the original evolutionary information to a certain extent, and the newly added genes through Gaussian mutation will appear near the original location with a high probability and a small probability to be explored to a distant location. Obviously, compared with other existing methods, although the method in this section injects slightly fewer new genes into the population, in the process of iteration, it is difficult for those excellent individuals who have not undergone individual restart to explore a better position with the help of random positions, while the new individuals restarted in this section are more conducive to the exploration of excellent individuals to a better position, enhance their ability to escape from localized regions to avoid algorithmic lack of motivation that leads to stagnation.

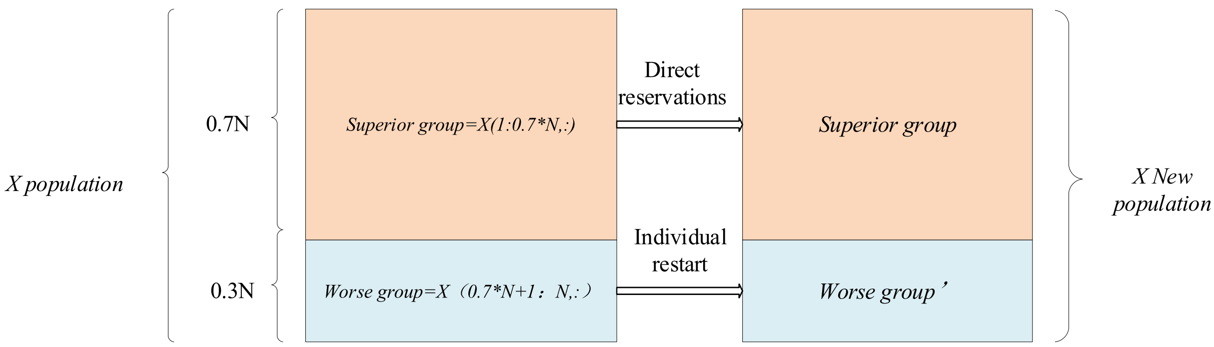


FIGURE 3. Individual restart strategy

3.5. **Comparative analysis with other improved OPA algorithms.** The OPA algorithm has attracted the attention of scholars due to its more outstanding performance. Three enhanced OPA algorithm papers have been published. The three previously improved OPA algorithms (IOPA [21], LFOPA, mOPA) are now compared and analyzed.

Analysis in terms of optimization strategies:

(1) Expulsion phase: IOPA [21] has no improvement on this, LFOPA algorithm introduces Levy flights, and mOPA algorithm also introduces Levy flights. The algorithm proposed in this paper proposes a new way of driving prey in this phase, which can effectively improve the convergence speed of the algorithm and supplement the population diversity at the same time.

(2) Encircling phase: there is no improvement in these three articles. In the improvement strategy of this paper, the base vectors are selected in a probabilistic mode. The

population diversity is used to control the size of the movement step, real-time monitoring and supplementing the population diversity.

(3) Attack phase: none of the three articles has been improved. In this paper, adaptive superiority learning and novel dyadic learning are utilized in this phase to avoid the problem of the algorithm falling into local optimum. In addition, the global crossover strategy is used to improve the quality of the solution.

Analyzing the algorithm architecture: IOPA [21] adds dimension learning (DL) strategy phase and opposition-based learning (OP) strategy phase; LFOPA adds GS strategy phase at the end of the algorithm; MOPA algorithm adds opposition-based learning phase at the beginning of each iteration of the algorithm. The MOPA algorithm adds the opposition-based learning (OP) strategies phase at the beginning of each iteration of the algorithm. The algorithm proposed in this paper adds an individual restart phase at the end of the algorithm.

4. **Experimental results and analysis.** In this section, the following series of experiments are done to comprehensively evaluate the overall performance of the IOPA. (1) Parameter validity analysis. (2) Evaluate the effectiveness of each proposed enhancement strategy. (3) comparison of the performance of the improved algorithm IOPA with the original algorithm and four excellent improved algorithms. All of the above experiments were conducted using the CEC2013 test set.

The CEC2013 test set contains single-peak functions, multi-peak functions, and combinatorial functions, among which F1–F5 has only one optimal value, which is a single-mode function, so enabling the evaluation of the algorithm's convergence performance. F6–F20 has multiple optimal values, which is a multimodal function that verifies the performance of jumping out of local optimum, and F21–F28 is a combination function, and detailed information on the test set can be found in the literature [27]. To ensure a fair comparison of the algorithms, all algorithms are run on a Windows 10 computer with an i5-7200U CPU and programmed with MATLAB R2020a.

4.1. **Parameter validity analysis.** In the IOPA algorithm, the use of Cauchy's random number as a step factor is proposed. In order to verify its validity and its impact on the performance of the algorithm, this section analyses the validity of the Cauchy random number as well as the parameters. To ensure the fairness of the validation, the number of populations in this section of the experiment $N = 50$, the dimension of the test function $D = 30$, the maximum number of function evaluations $Max\_FE = 100000$ and the maximum number of iterations $Max\_iter = 1000$. The rest of the parameters involved in the algorithms are as follows: $p1 = q = 0.9$, $p2 = 0.005$, $T = 50$, $pb = 0.5$, $CR = 0.8$, $D\_num = 10$.

To verify the validity of Cauchy's random numbers, the setting constant 2 and the Levy flight factor replacing Cauchy's random numbers will be analyzed in a comparative experiment. To verify the validity of the values taken, the performance of the IOPA algorithm is verified when the values are taken to be 0.1, 0.5 and 1, respectively. The function that achieves the best result is black labelled on each function.

From the results in Table 2, the use of constant and Lévy flight as the step factor are not as effective as the use of the Cauchy step factor, and the difference is large. So the Cauchy random number is chosen as the step factor. From the analysis of the parameters of the scale function γ in Table 3, the overall performance of the algorithm decreases when γ increases or decreases, and the algorithm performs best when $\gamma = 0.2$. Without special requirements, the Cauchy random number is taken as the step factor in IOPA and $\gamma = 0.2$.

Table 2. Step factor analysis

step factor	*Cauchy*		*Levy*		*Constant*	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**
F1	**0.00E+00**	**0.00E+00**	5.05E-30	1.51E-29	**0.00E+00**	**0.00E+00**
F2	**1.21E+05**	**5.82E+04**	1.91E+05	1.51E+05	1.57E+05	9.07E+04
F3	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F4	**3.19E+01**	**4.84E+01**	9.83E+01	1.35E+02	5.03E+01	8.45E+01
F5	**0.00E+00**	**0.00E+00**	1.82E-28	9.81E-28	**0.00E+00**	**0.00E+00**
F6	1.11E+01	1.50E+01	**6.54E+00**	**6.01E+00**	8.40E+00	1.20E+01
F7	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F8	**2.10E+01**	**4.78E-02**	2.10E+01	5.81E-02	2.10E+01	5.66E-02
F9	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F10	7.48E-02	5.13E-02	**6.40E-02**	**6.48E-02**	7.77E-02	6.47E-02
F11	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F12	**0.00E+00**	**0.00E+00**	8.12E+00	3.05E+01	4.73E+00	2.55E+01
F13	**0.00E+00**	**0.00E+00**	4.73E+00	2.55E+01	5.26E+00	2.83E+01
F14	**1.27E+03**	**3.27E+02**	1.33E+03	2.99E+02	1.39E+03	4.92E+02
F15	**6.47E+03**	**1.33E+03**	6.74E+03	1.03E+03	6.63E+03	1.11E+03
F16	**2.68E+00**	**3.37E-01**	2.79E+00	2.78E-01	2.84E+00	2.83E-01
F17	**1.72E+01**	**3.76E+00**	2.02E+01	6.69E+00	1.76E+01	3.93E+00
F18	**1.78E+02**	**2.24E+01**	1.84E+02	2.01E+01	1.82E+02	2.19E+01
F19	**3.77E+00**	**1.19E+00**	3.80E+00	1.03E+00	4.17E+00	1.19E+00
F20	**0.00E+00**	**0.00E+00**	7.93E-02	4.27E-01	**0.00E+00**	**0.00E+00**
F21	**4.00E+02**	**0.00E+00**	**4.00E+02**	**0.00E+00**	**4.00E+02**	**0.00E+00**
F22	**6.40E+02**	**7.13E+02**	9.96E+02	1.17E+03	8.36E+02	1.01E+03
F23	**4.34E+03**	**2.06E+03**	4.78E+03	2.36E+03	4.42E+03	2.18E+03
F24	**2.00E+02**	**0.00E+00**	2.00E+02	4.65E-02	2.00E+02	6.60E-02
F25	2.63E+02	3.38E+01	2.60E+02	3.56E+01	2.49E+02	3.61E+01
F26	2.86E+02	4.04E+01	**2.84E+02**	**4.26E+01**	2.96E+02	2.84E+01
F27	**3.13E+02**	**9.66E-01**	3.33E+02	1.04E+02	3.31E+02	9.11E+01
F28	**8.37E+02**	**1.38E+02**	9.42E+02	4.01E+02	9.42E+02	3.58E+02

4.2. **Effectiveness experiments of various improvement strategies.** As can be seen from the third section, the IOPA algorithm has been improved in four parts on the OPA algorithm. In this section. To comprehensively assess the efficacy of each enhancement approach, this section eliminates one enhancement approach from the IOPA algorithms to obtain the four new improved algorithms that are obtained accordingly. Includes Improved algorithm in IOPA after removing the improved prey expulsion operation of section 3.1 (abbreviated as IOPA1), improved algorithm in IOPA after removing the improved encircling prey operation of section 3.2 (abbreviated as IOPA2), Improved algorithm in IOPA after removing the improved attack phase of section 3.3 (abbreviated as IOPA3), Improved algorithm after removing the 3.4 population individual restart strategy in IOPA (abbreviated as IOPA4). The IOPA algorithm was compared using the CEC2013 test set. To guarantee an equitable comparison, each algorithm has a population size of $N = 50$, a test function dimension of $D = 30$, and a maximum limit of $Max\_FE = 100000$ for function evaluations. The remaining parameters involved in each algorithm are as follows: $p1 = q = 0.9$, $p2 = 0.005$, $T = 50$, $pb = 0.5$, $CR = 0.8$, $D\_num = 10$.

Table 4 shows the test results of each improved algorithm on 28 objective functions with a spatial dimension of 30 dimensions. The data indicates the mean and standard deviation of the corresponding algorithm, and data that are worse than the IOPA algorithm are clearly labeled. Additionally, the Wilcoxon rank sum test as well as the Friedman test are performed on each improved algorithm with the IOPA algorithm [28]. The principle is to

TABLE 3. Cauchy parameter analysis

parameters	$\gamma = 0.2$		$\gamma = 0.1$		$\gamma = 0.5$		$\gamma = 1$	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
F1	**0.00E+00**	**0.00E+00**	1.68E-30	9.06E-30	3.37E-30	1.26E-29	8.41E-30	3.71E-29
F2	**1.21E+05**	**5.82E+04**	1.82E+05	1.29E+05	1.97E+05	1.08E+05	1.98E+05	9.64E+04
F3	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F4	**3.19E+01**	**4.84E+01**	1.03E+02	2.11E+02	6.54E+01	7.58E+01	1.14E+02	1.95E+02
F5	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F6	1.11E+01	1.50E+01	1.34E+01	1.48E+01	1.54E+01	1.77E+01	**1.09E+01**	**1.67E+01**
F7	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F8	2.10E+01	4.78E-02	2.10E+01	5.81E-02	2.10E+01	5.76E-02	**2.10E+01**	**3.97E-02**
F9	**0.00E+00**	**0.00E+00**	7.53E-01	4.05E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F10	7.48E-02	5.13E-02	**5.55E-02**	**3.64E-02**	6.20E-02	5.59E-02	6.64E-02	4.95E-02
F11	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F12	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.60E+01	4.15E+01	3.97E+00	2.14E+01
F13	**0.00E+00**	**0.00E+00**	5.51E+00	2.97E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F14	**1.27E+03**	**3.27E+02**	1.28E+03	4.21E+02	1.42E+03	4.65E+02	1.32E+03	2.95E+02
F15	6.47E+03	1.33E+03	6.43E+03	1.15E+03	6.68E+03	9.20E+02	**6.32E+03**	**1.67E+03**
F16	**2.68E+00**	**3.37E-01**	2.75E+00	2.66E-01	2.69E+00	3.26E-01	2.71E+00	3.29E-01
F17	1.72E+01	3.76E+00	1.64E+01	2.90E+00	**1.60E+01**	**2.91E+00**	2.01E+01	9.79E+00
F18	**1.78E+02**	**2.24E+01**	1.81E+02	2.24E+01	1.84E+02	2.01E+01	1.84E+02	1.86E+01
F19	3.77E+00	1.19E+00	3.75E+00	1.06E+00	3.99E+00	1.83E+00	**3.66E+00**	**9.29E-01**
F20	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F21	4.00E+02	0.00E+00	4.00E+02	0.00E+00	4.00E+02	0.00E+00	4.00E+02	0.00E+00
F22	**6.40E+02**	**7.13E+02**	7.44E+02	8.01E+02	7.11E+02	8.23E+02	1.05E+03	1.23E+03
F23	4.34E+03	2.06E+03	4.62E+03	2.01E+03	**4.32E+03**	**2.16E+03**	5.95E+03	1.65E+03
F24	**2.00E+02**	**0.00E+00**	2.00E+02	8.33E-02	2.00E+02	6.82E-02	2.00E+02	9.25E-02
F25	2.63E+02	3.38E+01	2.45E+02	3.73E+01	2.52E+02	3.87E+01	**2.43E+02**	**3.61E+01**
F26	2.86E+02	4.04E+01	2.97E+02	1.79E+01	3.00E+02	2.21E+01	**2.85E+02**	**3.85E+01**
F27	**3.13E+02**	**9.66E-01**	3.57E+02	1.65E+02	3.14E+02	8.42E-01	3.31E+02	9.05E+01
F28	**8.37E+02**	**1.38E+02**	8.63E+02	1.56E+01	8.47E+02	1.41E+02	8.70E+02	1.52E+01

mix the two samples and sort them in ascending order, and then convert the values of the samples into ordinal numbers. Wilcoxon rank sum test compares the size of the ordinal numbers of the two samples. When large ordinal numbers are concentrated in one sample, the difference between that sample and the other sample is more significant, resulting in a p-value of less than 0.05. The Friedman test solves for the mean of the ordinal numbers of the samples. The smaller the corresponding ordinal mean of the sample, the better the overall performance. The specific results are shown in Tables 5 and 6. When the p-value is greater than 0.05, the comparison algorithm is obviously better than IOPA is indicated by '+', otherwise, it is indicated by '$-$', while when the p-value is greater than 0.05, it indicates that the difference between each improved algorithm and IOPA is insufficient, which is indicated by '='.

As can be seen from Table 5, The OPA1 algorithm significantly outperforms the IOPA algorithm in 3 functions, is significantly worse than the IOPA algorithm in 9 test functions, and has similar performance in the remaining 17 test functions. The IOPA2 algorithm significantly outperforms the IOPA algorithm on 3 test functions, significantly outperforms the IOPA algorithm on 13 test functions, and has similar performance on 12 test functions. The IOPA3 algorithm significantly outperforms the IOPA algorithm on 1 test function, significantly outperforms the IOPA algorithm on 9 test functions, and performs similarly on 18 test functions. The IOPA4 algorithm performs significantly worse than the IOPA algorithm on 6 test functions, and the algorithm performs similarly on 22 test functions. As can be seen from Table 6, after removing the corresponding one strategy in the IOPA algorithm, the corresponding improved algorithms all have a larger rank-mean

Table 4. Running results of each improvement strategy in a 30-dimensional CEC2013 test

	IOPA		**IOPA1**		**IOPA2**		**IOPA3**		**IOPA4**	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.38E-28	2.61E-28	1.53E-28	2.01E-28	3.37E-29	7.06E-29
F2	1.21E+05	5.82E+04	1.84E+05	8.81E+04	1.92E+05	1.06E+05	1.56E+05	9.63E+04	1.77E+05	1.12E+05
F3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F4	3.19E+01	4.84E+01	5.86E+01	1.20E+02	1.28E+02	1.78E+02	1.75E+01	3.28E+01	7.63E+01	7.68E+01
F5	0.00E+00	0.00E+00	1.82E-28	9.81E-28	2.02E-27	2.83E-27	1.36E-17	4.35E-17	3.64E-28	1.36E-27
F6	1.11E+01	1.50E+01	1.18E+01	1.57E+01	1.97E+01	2.18E+01	9.44E+00	8.34E+00	9.65E+00	1.35E+01
F7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F8	2.10E+01	4.78E-02	2.10E+01	3.77E-02	2.10E+01	5.83E-02	2.10E+01	4.97E-02	2.10E+01	5.41E-02
F9	0.00E+00	0.00E+00	7.83E-01	4.22E+00	7.52E-01	4.05E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	7.48E-02	5.13E-02	5.64E-02	3.04E-02	7.51E-02	5.21E-02	6.25E-02	5.27E-02	6.81E-02	5.68E-02
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.32E-02	1.79E-01
F12	0.00E+00	0.00E+00	3.02E+01	5.49E+01	1.18E+01	3.57E+01	3.04E+00	1.63E+01	0.00E+00	0.00E+00
F13	0.00E+00	0.00E+00	6.00E+00	3.23E+01	2.65E+01	5.98E+01	8.62E+00	3.27E+01	1.49E+01	4.46E+01
F14	1.27E+03	3.27E+02	6.43E+02	2.10E+02	2.34E+03	5.92E+02	2.07E+03	3.13E+02	1.23E+03	3.55E+02
F15	6.47E+03	1.33E+03	4.02E+03	5.65E+02	3.09E+03	5.25E+02	7.02E+03	8.76E+02	6.53E+03	1.45E+03
F16	2.68E+00	3.37E-01	2.70E+00	2.85E-01	2.52E+00	4.59E-01	2.70E+00	2.91E-01	2.70E+00	3.42E-01
F17	1.72E+01	3.76E+00	2.51E+01	1.23E+01	4.15E+01	1.47E+01	2.16E+01	6.46E+00	2.00E+01	1.13E+01
F18	1.78E+02	2.24E+01	1.45E+02	5.91E+01	5.15E+01	1.81E+01	1.91E+02	1.80E+01	1.83E+02	2.06E+01
F19	3.77E+00	1.19E+00	4.73E+00	2.25E+00	5.20E+00	1.91E+00	5.06E+00	3.13E+00	4.30E+00	1.67E+00
F20	0.00E+00	0.00E+00	1.99E+00	3.36E+00	3.12E+00	3.62E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F21	4.00E+02	0.00E+00	4.00E+02	0.00E+00	4.00E+02	3.07E-13	4.00E+02	0.00E+00	4.00E+02	0.00E+00
F22	6.40E+02	7.13E+02	6.77E+02	3.63E+02	2.07E+03	5.79E+02	2.02E+03	1.59E+03	1.11E+03	1.05E+03
F23	4.34E+03	2.06E+03	3.81E+03	6.32E+02	2.74E+03	7.37E+02	5.93E+03	1.89E+03	4.79E+03	2.13E+03
F24	2.00E+02	0.00E+00	2.00E+02	8.17E-02	2.00E+02	7.43E-02	2.00E+02	1.23E-01	2.00E+02	7.18E-02
F25	2.63E+02	3.38E+01	2.72E+02	3.41E+01	2.65E+02	3.30E+01	2.65E+02	3.83E+01	2.55E+02	3.64E+01
F26	2.86E+02	4.04E+01	2.95E+02	3.51E+01	3.00E+02	2.20E+01	2.94E+02	2.59E+01	2.93E+02	3.36E+01
F27	3.13E+02	9.66E-01	3.55E+02	1.51E+02	3.82E+02	2.05E+02	3.64E+02	1.54E+02	3.62E+02	1.48E+02
F28	8.37E+02	1.38E+02	9.40E+02	3.58E+02	1.01E+03	4.57E+02	8.71E+02	2.04E+01	8.60E+02	1.44E+02

than the IOPA algorithm, indicating that the four improved strategies proposed in this paper have an impact on the performance of the IOPA algorithm and that the improved encircling prey strategy in Section 3.2 has the greatest impact on the IOPA algorithm.

4.3. **Comparison of the performance of the IOPA algorithm with other algorithms.** To verify the performance of the IOPA algorithm in all aspects, this section conducts a comparative analysis with the original algorithm and other four excellent algorithms on the CEC2013 test set, including ESO [6], MSMA [11], EAOA [12], IAOA [13], and IOPA [21]. To ensure a fair comparison of each algorithm, the number of populations as $N = 50$, the optimization dimension as $D = 30$, and the maximum number of function evaluations is $Max\_FE = 100000$. If the algorithm involves a number of iterations, it needs to be converted according to the number of evaluations. Table 7 shows the other parameter settings of the algorithm.

The statistical findings of the IOPA algorithm and the other algorithms after conducting 30 separate runs on CEC2013 are presented in Table 8. The black-labeled data represents the optimal outcomes achieved on each test function. Table 9 and Table 10 show the results of the Wilcoxon rank sum test and Friedman test conducted on the IOPA algorithm in comparison to the remaining enhanced algorithms.

According to Table 8, the theoretical optimal values for nine functions, namely F1, F3, F5, F7, F9, F11, F12, F13, and F20, were reached using IOPA and IAOA. The theoretical optimum was attained by OPA and the algorithm in IOPA [21] exclusively for the F3,

TABLE 5. Wilcoxon rank sum test results of each improved algorithm and IOPA algorithm

	IOPA1	IOPA2	IOPA3	IOPA4
F1	1.000(=)	0.000(−)	0.000(−)	0.001(−)
F2	0.000(−)	0.002(−)	0.185(=)	0.017(−)
F3	1.000(=)	1.000(=)	1.000(=)	1.000(=)
F4	0.051(−)	0.000(−)	0.000(+)	0.000(−)
F5	0.333(=)	0.000(−)	0.000(−)	0.160(=)
F6	0.515(=)	0.008(−)	0.651(=)	0.923(=)
F7	1.000(=)	1.000(=)	1.000(=)	1.000(=)
F8	0.065(=)	0.364(=)	0.078(=)	0.024(−)
F9	0.333(=)	0.333(=)	1.000(=)	1.000(=)
F10	0.293(=)	0.864(=)	0.190(=)	0.468(=)
F11	1.000(=)	1.000(=)	1.000(=)	0.333(=)
F12	0.005(−)	0.082(=)	0.333(=)	1.000(=)
F13	0.333(=)	0.022(−)	0.160(=)	0.081(=)
F14	0.000(+)	0.000(−)	0.000(−)	0.678(=)
F15	0.000(+)	0.000(+)	0.003(−)	0.251(=)
F16	0.988(=)	0.239(=)	0.761(=)	0.750(=)
F17	0.003(−)	0.000(−)	0.000(−)	0.917(=)
F18	0.047(+)	0.000(+)	0.033(−)	0.446(=)
F19	0.050(−)	0.000(−)	0.050(−)	0.115(=)
F20	0.003(−)	0.000(−)	1.000(=)	1.000(=)
F21	1.000(=)	1.000(=)	1.000(=)	1.000(=)
F22	0.050(−)	0.000(−)	0.000(−)	0.058(=)
F23	0.988(=)	0.015(+)	0.000(−)	0.549(=)
F24	0.160(=)	0.334(=)	0.081(=)	1.000(=)
F25	0.157(=)	0.657(=)	0.657(=)	0.584(=)
F26	0.410(=)	0.205(=)	0.523(=)	0.559(=)
F27	0.000(−)	0.0269(−)	0.365(=)	0.010(−)
F28	0.007(−)	0.001(−)	0.145(=)	0.000(−)
(+/ = /−)	(3/17/9)	(3/12/13)	(1/18/9)	(0/22/6)

TABLE 6. Friedman test results of each improved algorithm

	IOPA	**IOPA1**	**IOPA2**	**IOPA**	**IOPA4**
Avg.rank	2.09	2.98	3.82	3.25	2.88
sort	1	3	5	4	2

TABLE 7. Initialization settings for each algorithm parameter

Algorithms	**Related parameters**
IOPA	$p1 = q = 0.9$; $p2 = 0.005$; $T = 50$; $pb = 0.5$; $CR = 0.8$; $D\_num = 10$
OPA	$p1 = q = 0.9$; $p2 = 0.005$; $F = 2$
ESO	$Threshold = 0.25$; $Thresold2 = 0.6$; $T = 900$; $vec\_flag = [1, -1]$
IAOA	$C2 = 6$; $C3 = 2$; $C4 = 0.5$
MSMA	$z = 0.03$; $l = 2$
EAOA	$C1 = 2$; $C2 = 6$; $C3 = 2$; $C4 = 0.5$
IOPA [21]	$p1 = q = 0.9$; $p2 = 0.005$; $F = 2$

F7, and F11 functions, by ESO exclusively for the F3 and F11 functions, and by MSMA only for the F3 function. In conclusion, the findings indicate that the IOPA algorithm

Table 8. Data results of IOPA and other algorithms on the 30-dimensional CEC2013 test set

		IOPA	**OPA**	**ESO**	**IAOA**	**MSMA**	**EAOA**	**IOPA** [21]
F1	Mean	**0.00E+00**	2.32E-26	8.77E+00	**0.00E+00**	9.23E+00	2.62E+03	1.30E-21
	Std	**0.00E+00**	3.95E-26	4.71E+00	**0.00E+00**	3.37E+01	6.73E+02	5.69E-21
F2	Mean	**1.21E+05**	1.78E+05	1.46E+07	1.27E+06	3.53E+07	9.34E+07	3.86E+05
	Std	**5.82E+04**	9.96E+04	3.81E+06	1.24E+06	1.50E+07	2.84E+07	266E+05
F3	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.87E+08	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	6.96E+08	**0.00E+00**
F4	Mean	**3.19E+01**	1.41E+02	3.92E+04	8.04E+03	5.09E+04	5.41E+04	6.89E+02
	Std	**4.84E+01**	2.23E+02	7.77E+03	9.17E+03	5.17E+03	8.83E+03	7.96E+02
F5	Mean	**0.00E+00**	2.33E-15	1.73E+00	**0.00E+00**	1.12E+01	1.53E+03	4.40E-15
	Std	**0.00E+00**	1.56E-15	1.63E+00	**0.00E+00**	1.10E+01	4.81E+02	2.88E-15
F6	Mean	**1.11E+01**	2.18E+01	8.57E+01	2.76E+01	9.64E+01	2.63E+02	3.73E+01
	Std	**1.50E+01**	2.03E+01	3.40E+01	2.38E+01	2.74E+01	6.51E+01	2.73E+01
F7	Mean	**0.00E+00**	**0.00E+00**	3.30E+00	**0.00E+00**	9.40E+00	1.30E+01	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	1.51E+01	**0.00E+00**	2.72E+01	2.05E+01	**0.00E+00**
F8	Mean	2.10E+01	2.10E+01	2.10E+01	2.10E+01	2.10E+01	**2.10E+01**	2.10E+01
	Std	6.00E-02	6.41E-02	5.25E-02	6.53E-02	7.16E-02	**5.20E-02**	4.97E-02
F9	Mean	**0.00E+00**	2.32E+00	7.43E+00	**0.00E+00**	4.37E+00	4.74E+00	7.24E-01
	Std	**0.00E+00**	7.01E+00	9.89E+00	**0.00E+00**	9.88E+00	8.40E+00	3.90E+00
F10	Mean	**7.48E-02**	8.26E-02	9.63E+00	8.05E-02	9.15E+01	7.79E+02	8.06E-02
	Std	**5.13E-02**	6.23E-02	3.04E+00	7.42E-02	8.30E+01	2.50E+02	7.67E-02
F11	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.00E+00	0.00E+00	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.00E+00	0.00E+00	**0.00E+00**
F12	Mean	**0.00E+00**	8.98E+01	6.77E+01	**0.00E+00**	1.20E+02	1.62E+02	5.16E+01
	Std	**0.00E+00**	6.56E+01	6.53E+01	**0.00E+00**	7.79E+01	6.05E+01	6.86E+01
F13	Mean	**0.00E+00**	8.07E+01	8.16E+01	**0.00E+00**	1.34E+02	1.15E+02	6.50E+01
	Std	**0.00E+00**	8.15E+01	7.82E+01	**0.00E+00**	9.06E+01	8.80E+01	7.56E+01
F14	Mean	**1.27E+03**	2.74E+03	1.81E+03	2.17E+03	1.56E+03	4.46E+03	2.20E+03
	Std	**3.27E+02**	7.27E+02	3.83E+02	4.13E+02	4.33E+02	5.45E+02	4.58E+02
F15	Mean	6.47E+03	**3.83E+03**	6.58E+03	6.92E+03	4.57E+03	5.79E+03	4.09E+03
	Std	1.33E+03	**5.80E+02**	3.90E+02	4.94E+02	8.93E+02	5.35E+02	5.59E+02
F16	Mean	2.68E+00	**1.11E+00**	2.22E+00	2.80E+00	2.38E+00	2.51E+00	1.32E+00
	Std	3.37E-01	**5.07E-01**	2.86E-01	3.61E-01	5.33E-01	4.53E-01	7.95E-01
F17	Mean	**1.72E+01**	5.67E+01	4.09E+02	4.36E+01	2.89E+02	4.39E+02	6.21E+01
	Std	**3.76E+00**	1.82E+01	6.90E+01	2.06E+01	1.47E+02	4.68E+01	2.28E+01
F18	Mean	1.78E+02	7.64E+01	4.09E+02	1.81E+02	3.88E+02	4.85E+02	**7.31E+01**
	Std	2.24E+01	2.50E+01	7.04E+01	2.66E+01	1.25E+02	6.41E+01	**2.74E+01**
F19	Mean	**3.77E+00**	6.04E+00	2.46E+01	1.18E+01	3.01E+01	2.84E+02	5.21E+00
	Std	**1.19E+00**	2.05E+00	3.82E+00	2.88E+00	9.79E+00	4.78E+02	2.29E+00
F20	Mean	**0.00E+00**	4.08E+00	7.65E+00	**0.00E+00**	1.50E+01	1.31E+01	2.15E+00
	Std	**0.00E+00**	4.48E+00	4.48E+00	**0.00E+00**	3.29E-07	2.68E+00	3.93E+00
F21	Mean	4.00E+02	4.00E+02	4.02E+02	**3.93E+02**	4.00E+02	6.26E+02	4.00E+02
	Std	0.00E+00	0.00E+00	6.91E-01	**3.59E+01**	1.37E-01	7.01E+01	2.36E-11
F22	Mean	**7.14E+02**	3.04E+03	2.02E+03	1.93E+03	1.60E+03	5.41E+03	2.27E+03
	Std	**1.01E+03**	7.42E+02	4.46E+02	5.04E+02	4.00E+02	5.10E+02	4.80E+02
F23	Mean	4.34E+03	**4.16E+03**	7.13E+03	6.65E+03	5.12E+03	6.74E+03	4.27E+03
	Std	2.06E+03	**9.08E+02**	4.52E+02	5.54E+02	7.73E+02	6.35E+02	9.02E+02
F24	Mean	**2.00E+02**	2.03E+02	2.11E+02	2.29E+02	2.49E+02	2.38E+02	2.02E+02
	Std	**0.00E+00**	1.08E+01	2.25E+01	3.38E+01	3.96E+01	2.57E+01	1.04E+01
F25	Mean	2.63E+02	2.75E+02	2.70E+02	2.82E+02	2.97E+02	3.02E+02	**2.54E+02**
	Std	3.38E+01	2.97E+01	2.54E+01	1.94E+01	8.18E+00	1.10E+01	**3.64E+01**
F26	Mean	**2.86E+02**	3.06E+02	3.27E+02	3.23E+02	3.57E+02	3.46E+02	2.94E+02
	Std	**4.04E+01**	3.62E+01	3.11E+01	2.78E+01	4.02E+01	4.44E+01	4.13E+01
F27	Mean	**3.13E+02**	4.60E+02	7.45E+02	8.68E+02	9.14E+02	9.65E+02	4.46E+02
	Std	**9.66E-01**	2.52E+02	2.60E+02	2.38E+02	3.25E+02	1.83E+02	2.46E+02
F28	Mean	**8.37E+02**	1.08E+03	3.00E+03	1.05E+03	1.22E+03	3.32E+03	9.89E+02
	Std	**1.38E+02**	6.16E+02	5.46E+02	5.09E+02	4.22E+02	1.06E+03	3.44E+02

Table 9. Wilcoxon rank sum test results for IOPA and other algorithms on the CEC2013 test set

	OPA	**ESO**	**IAOA**	**MSMA**	**EAOA**	**IOPA** [21]
F1	0.000(−)	0.000(−)	1.000(=)	0.000(−)	0.000(−)	0.000(−)
F2	0.003(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.001(−)
F3	1.000(=)	1.000(=)	1.000(=)	1.000(=)	0.009(−)	1.000(=)
F4	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F5	0.000(−)	0.000(−)	1.000(=)	0.000(−)	0.000(−)	0.000(−)
F6	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F7	1.000(=)	0.153(=)	1.000(=)	0.009(−)	0.000(−)	1.000(=)
F8	0.341(=)	0.530(=)	0.005(−)	0.001(−)	0.614(=)	0.037(−)
F9	0.076(=)	0.000(−)	1.000(=)	0.019(−)	0.000(−)	0.325(=)
F10	0.593(=)	0.000(−)	0.971(=)	0.000(−)	0.000(−)	0.834(=)
F11	1.000(=)	1.000(=)	1.000(=)	1.000(=)	1.000(=)	1.000(=)
F12	0.000(−)	0.000(−)	1.000(=)	0.000(−)	0.000(−)	0.000(−)
F13	0.000(−)	0.000(−)	1.000(=)	0.000(−)	0.000(−)	0.000(−)
F14	0.000(−)	0.000(−)	0.000(−)	0.001(−)	0.000(−)	0.000(−)
F15	0.000(+)	0.254(=)	0.133(=)	0.000(−)	0.000(−)	0.000(−)
F16	0.000(+)	0.000(−)	0.135(=)	0.052(=)	0.133(=)	0.000(−)
F17	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F18	0.000(+)	0.000(−)	0.525(=)	0.000(−)	0.000(−)	0.000(−)
F19	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F20	0.000(−)	0.000(−)	1.000(=)	0.000(−)	0.000(−)	0.005(−)
F21	0.341(=)	0.000(−)	1.000(=)	0.000(−)	0.000(−)	0.341(=)
F22	0.000(+)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F23	0.729(=)	0.000(−)	0.000(−)	0.251(=)	0.000(−)	0.649(=)
F24	0.003(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.086(=)
F25	0.088(=)	0.772(=)	0.035(−)	0.000(−)	0.000(−)	0.457(=)
F26	0.018(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.299(=)
F27	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.001(−)
F28	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
+/ = /−	4/9/15	0/6/22	0/14/14	0/4/24	0/3/25	0/10/18

Table 10. Friedman's test for each algorithm

		IOPA	**OPA**	**ESO**	**IAOA**	**MSMA**	**EAOA**	**IOPA** [21]
$D = 30$	*Avg.rank*	2.05	3.27	4.84	3.39	5.14	6.39	2.91
	sort	1	3	5	4	6	7	2

has a higher likelihood of achieving convergence to the theoretical optimum compared to alternative algorithms. Table 9 demonstrates that OPA outperforms IOPA in just 4 functions while doing significantly worse on 15 tasks. The remaining 9 functions remain equivalent. ESO exhibited no substantial improvement in any of the functions, as its performance was comparable on only 6 functions, but markedly inferior on 22 functions. IAOA demonstrates comparable performance on 14 functions while exhibiting notably inferior performance on the remaining 14 functions. The IOPA [21] are equal on 10 functions, but significantly worse than IOPA on 18 functions; Neither MSMA nor EAOA outperforms IOPA on any function and demonstrates significantly inferior performance on 24 or 25 functions. The data presented in Table 10 demonstrates that rank-mean of IOPA is lower than the remaining six algorithms, suggesting that the IOPA algorithm exhibits superior performance.

In conclusion, when compared to alternative algorithms, the IOPA method suggested in this study exhibits certain advantages in terms of convergence accuracy.

In order to visually compare the differences between the convergence speeds of the algorithms, Figure 4 represents the iterative evolution curve of each algorithm running randomly once in 30 dimensions of the test optimization problem, the number of evaluations of the function and the logarithm of the fitness value of the related function are represented by the horizontal and vertical coordinates, respectively.

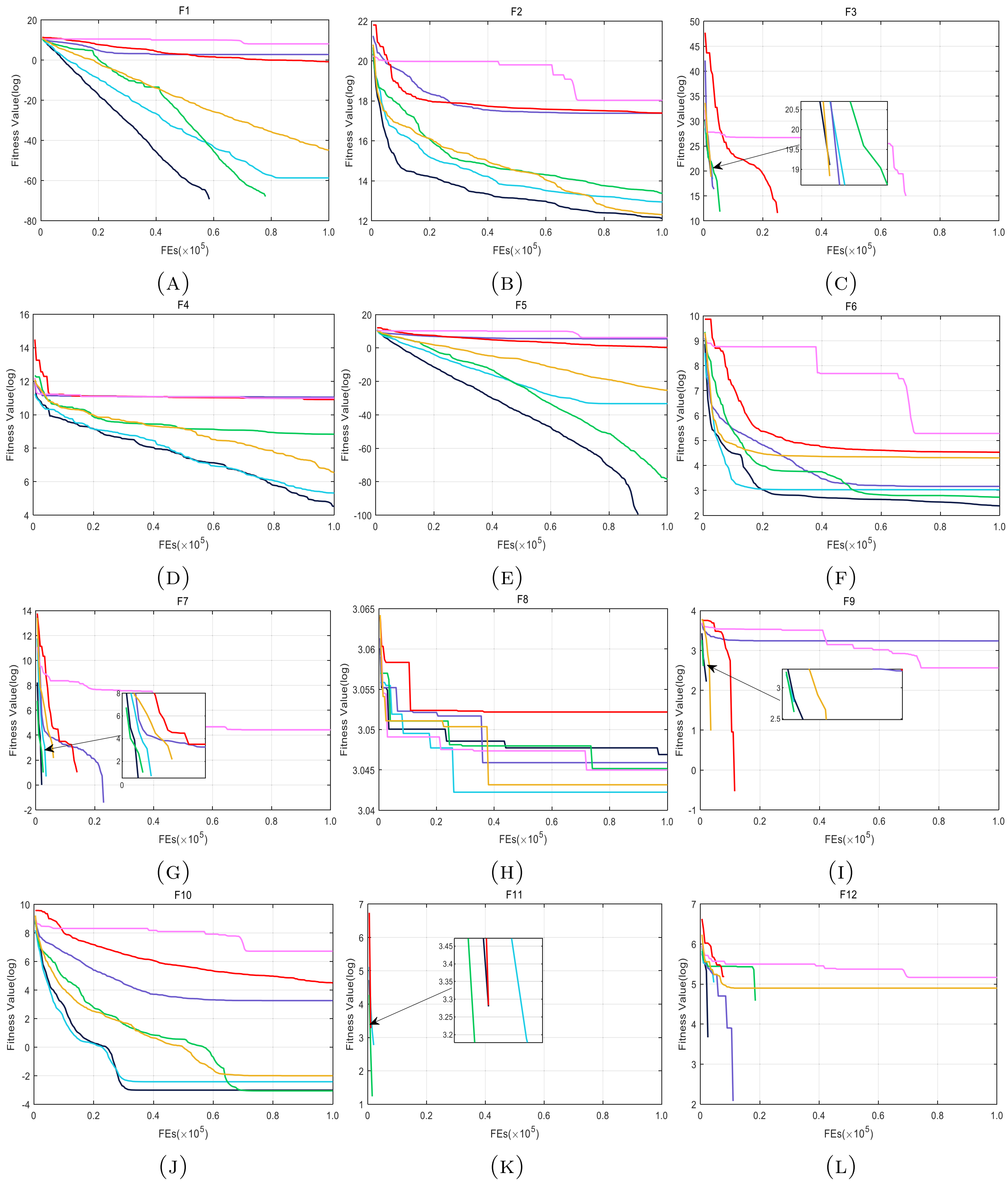
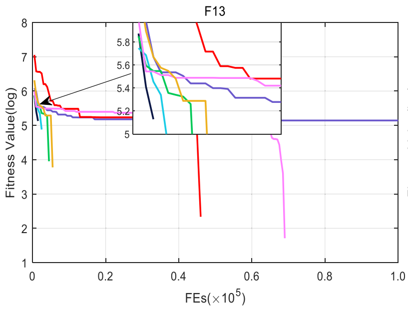
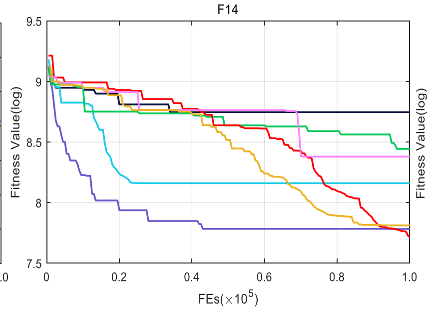


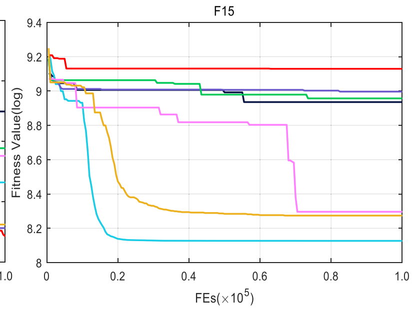
Figure 4. Convergence curves for each algorithm on 28 functions



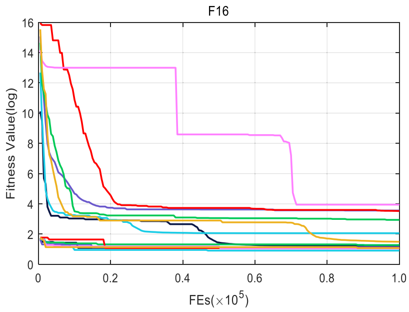
(M)



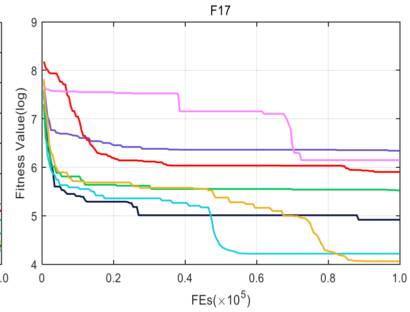
(N)



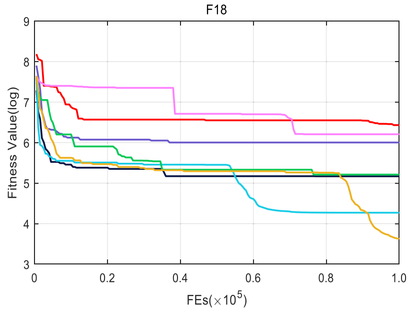
(O)



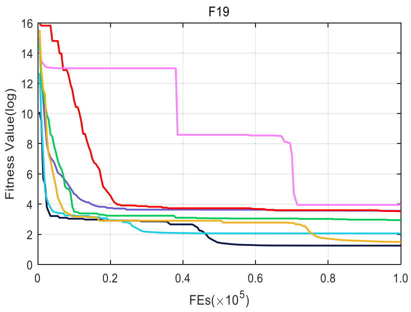
(P)



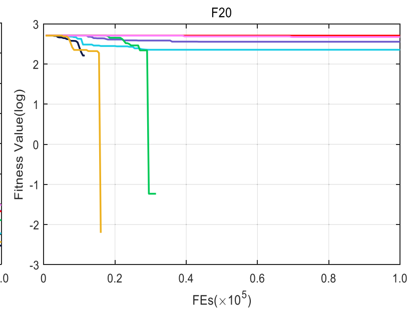
(Q)



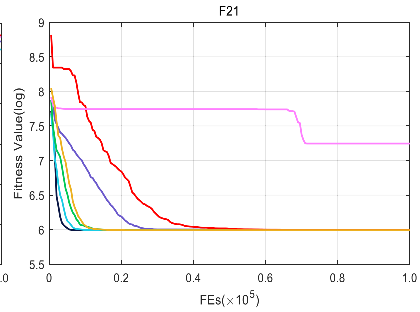
(R)



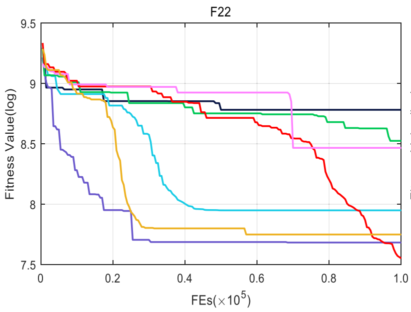
(S)



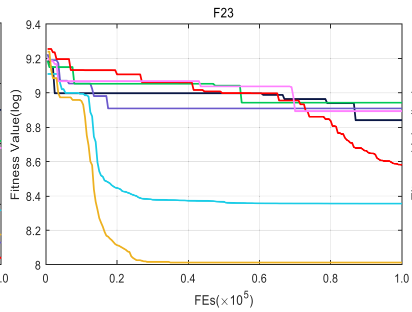
(T)



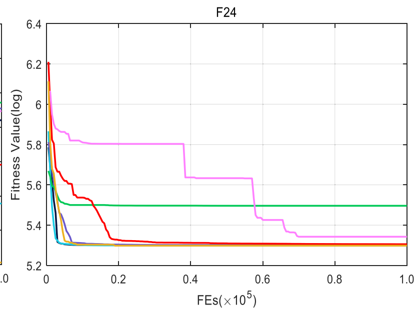
(U)



(V)



(W)



(X)

FIGURE 4. Convergence curves for each algorithm on 28 functions

Upon examination of Figure 4, From F1–F5, IOPA outperforms the rest of the algorithms in terms of convergence speed as well as convergence accuracy. Furthermore, IOPA attains the theoretical optimum for the F1, F3, and F5 functions. From F6-F10, exhibits faster convergence in the early stage of evolution. However, it slows down during the middle stage. In the late stage, IOPA surpasses other methods in terms of both convergence speed and convergence accuracy. The convergence speed of OPA was second only to that of IAOA on the F9 and F11 functions. Additionally, OPA had superior convergence speed and accuracy on the F13 and F20 functions. However, it exhibited worse performance

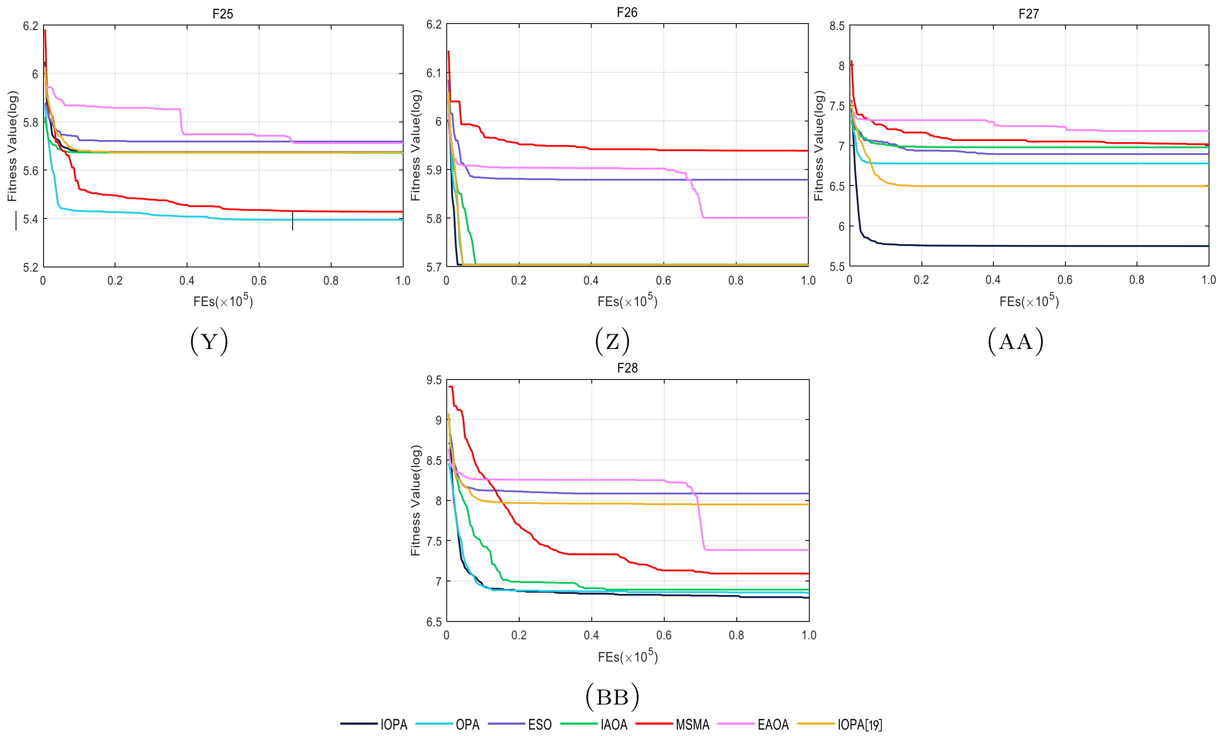


Figure 4. Convergence curves for each algorithm on 28 functions

on the F14, F15, and F16 convergence speeds and accuracies. The initial stages of IOPA exhibit accelerated convergence on the F17 and F18 functions. However, in subsequent stages, IOPA demonstrates superior convergence speed, ranking second only to OPA. Similarly, the convergence accuracy of IOPA is second only to OPA. From F21-F28, it achieves the fastest convergence speed and the best convergence accuracy on functions F21, F26, F27, and F28. IOPA [21] has the best convergence performance on F17, F18, and F23, and IOPA converges at a similar rate as OPA, ESO, MSMA, and the IOPA [21] algorithm on the F24 function. To summarize, IOPA has greater performance across many functions when compared to OPA and the other four algorithms, demonstrating advantages in terms of convergence speed and accuracy.

5. **Conclusion.** This work presents an improved version of the Orca Predation Algorithm (IOPA). The aim is to enhance its convergence performance and address issues related to lack of diversity and susceptibility to local optima. First, the chasing phase introduces a novel approach to individual renewal. It uses the Cauchy step factor to enhance the rate of evolution and meet the requirements for both rapid convergence and diversity in the evolutionary process. Second, base vectors are probabilistically chosen as the population envelops its prey. This enhances the probability that new individuals will utilize the evolutionary knowledge of others. Additionally, population diversity is employed to regulate the magnitude of the step size. This enables real-time monitoring and restoration of population diversity. Third, during the attack phase, adaptive merit learning is employed instead of average learning. Additionally, a global compartmentalized crossover strategy is introduced to enhance the quality of information sharing. This improves the efficiency and accuracy of the OPA search. Furthermore, a mutual opposites learning approach is used to enhance the mutation process of deceased individuals. This empowers the algorithm to overcome local optima. Finally, a novel approach for population individual restart is introduced. It involves modifying specific genes of individuals

using Gaussian mutation. This technique aims to address population variety and enhance the algorithm's ability to conduct further optimization searches. The results of the CEC2013 test set show that the IOPA algorithm has a better overall performance, with faster convergence speed and higher convergence accuracy. However, it is found during the experiment that the performance of the algorithm decreases as the dimension of the optimization problem increases and the time complexity rises. Therefore, reducing the complexity and improving the performance of the algorithm in large-scale optimization is an important part of the research. IOPA can be further extended and applied in the engineering field to solve the optimization problems in real engineering and maximize the benefits. For example, welded beam design problem, compression spring design problem.

Acknowledgment. This work was supported by the Project of Scientific and Technological Innovation Development of Jilin in China under Grant 20210103090.

REFERENCES

[1] L. Nurbekyan, W. Lei, and Y. Yang, "Efficient natural gradient descent methods for large-scale PDE-based optimization problems," *SIAM Journal on Scientific Computing*, vol. 45, no. 4, pp. A1621–A1655, 2023.
[2] K. De Jong, "Learning with genetic algorithms: An overview," *Machine Learning*, vol. 3, pp. 121–138, 1988.
[3] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.
[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.
[5] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.
[6] T.-Y. Wu, H. Li, and S.-C. Chu, "Cppe: An improved phasmatodea population evolution algorithm with chaotic maps," *Mathematics*, vol. 11, no. 9, p. 1977, 2023.
[7] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A spectral convolutional neural network model based on adaptive fick's law for hyperspectral image classification," *Computers, Materials & Continua*, vol. 79, no. 1, 2024.
[8] L. Yao, P. Yuan, C.-Y. Tsai, T. Zhang, Y. Lu, and S. Ding, "Eso: An enhanced snake optimizer for real-world engineering problems," *Expert Systems with Applications*, vol. 230, p. 120594, 2023.
[9] T.-Y. Wu, A. Shao, and J.-S. Pan, "Ctoa: toward a chaotic-based tumbleweed optimization algorithm," *Mathematics*, vol. 11, no. 10, p. 2339, 2023.
[10] Q. Askari and I. Younas, "Improved political optimizer for complex landscapes and engineering optimization problems," *Expert Systems with Applications*, vol. 182, p. 115178, 2021.
[11] V. K. R. A. Kalananda and V. L. N. Komanapalli, "A combinatorial social group whale optimization algorithm for numerical and engineering optimization problems," *Applied Soft Computing*, vol. 99, p. 106903, 2021.
[12] S. Satapathy and A. Naik, "Social group optimization (sgo): a new population evolutionary optimization technique," *Complex & Intelligent Systems*, vol. 2, no. 3, pp. 173–203, 2016.
[13] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
[14] L. Deng and S. Liu, "A multi-strategy improved slime mould algorithm for global optimization and engineering design problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 404, p. 115764, 2023.
[15] A. S. Desuky, S. Hussain, S. Kausar, M. A. Islam, and L. M. El Bakrawy, "Eaoa: an enhanced archimedes optimization algorithm for feature selection in classification," *IEEE Access*, vol. 9, pp. 120 795–120 814, 2021.
[16] Y.-J. Wang, M.-C. Chen, and C. S. Ku, "An improved archimedes optimization algorithm (iaoa)," *Journal of Network Intelligence*, vol. 8, no. 3, pp. 693–709, 2023.
[17] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovskỳ, "Coati optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 259, p. 110011, 2023.

[18] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar, and M. A. Awadallah, “White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems,” *Knowledge-Based Systems*, vol. 243, p. 108457, 2022.

[19] H. Zamani, M. H. Nadimi-Shahraki, and A. H. Gandomi, “Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization,” *Computer Methods in Applied Mechanics and Engineering*, vol. 392, p. 114616, 2022.

[20] Y. Jiang, Q. Wu, S. Zhu, and L. Zhang, “Orca predation algorithm: A novel bio-inspired algorithm for global optimization problems,” *Expert Systems with Applications*, vol. 188, p. 116026, 2022.

[21] G. Hu, W. Jing, G. Wei, and M. Abbas, “Opposition-based learning boosted orca predation algorithm with dimension learning: a case study of multi-degree reduction for nurbs curves,” *Journal of Computational Design and Engineering*, vol. 10, no. 2, pp. 722–757, 2023.

[22] E. H. Houssein, M. R. Saad, A. A. Ali, and H. Shaban, “Multiple strategies boosted orca predation algorithm for engineering optimization problems,” *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, p. 67, 2023.

[23] M. M. Emam, H. A. El-Sattar, E. H. Houssein, and S. Kamel, “Modified orca predation algorithm: developments and perspectives on global optimization and hybrid energy systems,” *Neural Computing and Applications*, vol. 35, no. 20, pp. 15 051–15 073, 2023.

[24] A. Taheri, K. RahimiZadeh, and R. V. Rao, “An efficient balanced teaching-learning-based optimization algorithm with individual restarting strategy for solving global optimization problems,” *Information Sciences*, vol. 576, pp. 68–104, 2021.

[25] Q. Zhang and Z. Meng, “Adaptive differential evolution algorithm based on deeply-informed mutation strategy and restart mechanism,” *Engineering Applications of Artificial Intelligence*, vol. 126, p. 107001, 2023.

[26] Y. Wang, J. Han, and Z. Teng, “An improved group teaching optimization algorithm for global function optimization,” *Scientific Reports*, vol. 12, no. 1, p. 11267, 2022.

[27] J. J. Liang, B. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, “Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization,” *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, vol. 201212, no. 34, pp. 281–295, 2013.

[28] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.