Journal of Network Intelligence ©2024 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **9**, Number **2**, May **2024**

An Improved Elephant Clan Optimization Algorithm for Global Function Optimization

Yan-Jiao Wang
School of Electrical Engineering Northeast Electric Power University
169 Changchun Road, Jilin, 132012, China
wangyanjiao1028@126.com

Wen-Yu Liu*
School of Electrical Engineering Northeast Electric Power University
169 Changchun Road, Jilin, 132012, China
18844225768@163.com

*Corresponding author: Wen-Yu Liu

Received May 29, 2023, revised August 12, 2023, accepted November 16, 2023.

Abstract. *The elephant clan optimization algorithm is a meta-heuristic algorithm that was developed by simulating elephant behavior. However, the algorithm has issues with slow convergence speed, poor convergence accuracy, and easy local optimum fall-in. Based on this, this paper proposes an improved elephant clan optimization algorithm. In this paper, we mainly improve the population initialization based on the additional autonomous movement strategy, the Euclidean distance-based community division, and the early maturity suppression mechanism to improve the population diversity to avoid falling into the local optimum. At the same time, the improved individuals balance the convergence speed and diversity with autonomous positional movement and novel updating strategy, in addition, this paper chooses different replacement methods according to different fitness values and improves the next convergence speed and robustness by the simplex method in the late stage of evolution. Finally, experiments are performed on 28 benchmark functions. The findings demonstrate that the IECO algorithm clearly outperforms the original ECO algorithm and other top algorithms in terms of convergence speed, convergence accuracy, and algorithm stability.*
Keywords: Elephant clan optimization, Meta-heuristic, Global optimization, Benchmark functions

1. **Introduction.**

Optimization problems, such as shortest time, shortest path, optimal parameters, etc., are widely found in the fields of Internet advertising, recommender systems, unmanned vehicles, etc. To solve such complex optimization problems, researchers try to solve the problems using meta-heuristic algorithms, which are proposed by simulating nature and human intelligence, among others. Compared with traditional optimization methods, meta-heuristic algorithms are a flexible and gradient-variation-ignoring approach, and their well-studied and exploited capabilities can be applied to a variety of complex global optimization problems.

Meta-heuristic algorithms are used to solve optimization problems by simulating the behavior of collective intelligence. Metaheuristic algorithms may be categorized into three categories in line with distinct heuristic notions. The first group consists of metaheuristic algorithms that are motivated by physical occurrences, such as BB-BC [1], GSA [2], WEO [3], LSA [4], AOA [5], etc. The Atomic Orbital Search (AOS) optimization technique was introduced by Azizi in the same year, and it was motivated by the atomic theory of quantum mechanics [6]. The second category consists of metaheuristic algorithms, such as PSO [7], ACO [8], GWO [9], MA [10], ABC [11], BA [12], WOA [13], etc., that are motivated by the behavior of biological groupings. New metaheuristic algorithms have been proposed in the last few years. The Raccoon Optimization Algorithm (ROA), developed in 2019 by Sina et al., was inspired by how raccoons hunt for high-quality food quickly using their eyesight and touch [14]; By idealizing and simplifying the monarch butterfly movement, Wang et al. created Monarch Butterfly Optimization (MBO) in the same year [15]; Li et al.'s development of the slime mold algorithm (SMA) in 2020 was inspired by the behavioral and physical changes that Physarum polycephalum endured during foraging [16]; The Elephant Clan Optimization (ECO) method was developed in 2021 by Jafari et al. by modeling the fundamental aspects of elephant individual and group behavior [17]; Hashim et al. introduced a snake optimizer (SO) in 2022 based on the foraging or mating behavior of snakes in varied temperature and food availability scenarios [18]. The third category is metaheuristic algorithms inspired by other mechanisms, such as genetic processes. Examples are GA [19], DE [20], IA [21], SOS [22], etc. The Blood Coagulation Algorithm (BCA), motivated by platelet cooperation and the thrombosis process, was devised by Drishti Yadav et al. in the same year [23].

Compared with the traditional metaheuristic algorithms, the new metaheuristic algorithms proposed in recent years often come with more novel and efficient evolution strategies, which are very likely to greatly improve the optimization effect, and further improvement of the new metaheuristic algorithms will inevitably gradually attract the attention of scholars. In this context, this paper improves the Object Family Optimization (ECO) algorithm and proposes an Improved Elephant Family Optimization (IECO) algorithm to further improve its convergence speed, convergence accuracy, and algorithmic stability. The main contributions of this paper are as follows:

(1) Improvement of the initialization phase. Based on the autonomous movement habits of natural elephant populations, a population initialization approach with additional autonomous movement strategies is proposed to increase population diversity: The proposed Euclidean distance-based population partitioning method avoids the blindness of the original random partitioning method and different balances convergence speed and population diversity: A premature suppression method, inspired by ABC, is put forward to enhance the likelihood that the algorithm will leave the local optimum.

(2) To improve the individual renewal methods of family clans. According to different evolutionary stages and member types, learning from different good evolutionary genes, and introducing autonomous positional movement, we propose a new search strategy for individual family clan members and matriarchs based on autonomous positional movement to balance convergence speed and population diversity.

(3) A new male elephant individual renewal strategy is proposed, in which each male moves under the pull of the central position of each female matriarch and its autonomous moving position, avoiding the randomness of the original path and further balancing the algorithm's population diversity and rate of convergence.

(4) To propose new individual replacement strategies for segments of the family clan. Adult elephants among them select various replacement strategies based on their fitness values, which accelerates convergence while retaining population variety and strengthens the algorithm's ability to handle various problems with optimization; The population diversity is maintained and the convergence speed of the algorithm is increased in the early stages of evolution by the inferior individuals learning from the superior individuals while retaining some of their knowledge. In the later stages of evolution, the simplex

method is used to speed up convergence while new superior individuals may be born to further increase population diversity.

Testing on the CEC2013 test set reveals that the IECO reported in this paper exhibits significant improvements over ECO and four more conventional optimization algorithms in terms of convergence time, accuracy, and stability.

This essay's remainder is organized as follows: The operation and workings of the ECO algorithm are explained in Section 3. In Section 4, the flaws of the original ECO approach are discussed and an improved IECO algorithm is proposed. Section 5 compares the IECO algorithm to the original ECO technique and other more widely used contemporary methods using simulation results from the CEC2013 test function set. An overview of the method suggested in this work is provided in Section 6.

2. Related work.

In order to further enhance the optimization effect and improve the existing meta-heuristic algorithms, researchers have conducted a lot of studies. The DLPSO method was created by Zhang et al. [24], which chooses the best vectors from those dispersed throughout the search space, creates a new vector, and moves to the better vector location, increasing the likelihood of leaving the local optimum; By mimicking the human forgetting phenomena, Xia et al. suggested an extended PSO with forgetting capacity (XPSO) [25] and accelerated the algorithm's convergence; Han et al. proposed the IGTOA algorithm [26] in 2022 to improve the convergence speed and accuracy of the algorithm through dynamic partitioning and a new subspace search mode; Song et al. used the hybrid mechanism of complete learning and incomplete learning to propose the INCCLA algorithm to improve the search and development capabilities of the algorithm [27]; Haroon et al. in 2022 proposed a Gaussian Mutation-Spider Monkey Optimization (GM-SMO) algorithm to solve overfitting and unbalanced data in scene classification. Gaussian mutation improves the diversity of the algorithm's population, allowing GM-SMO to better balance exploration and exploitation [28]; Chen et al. in 2023 proposed a genetic algorithm combined with simulated annealing to solve the Vehicle Routing Problem (VRP) [29]; A multi-population individual-based artificial bee colony algorithm (IDABC) was proposed by Zhou et al. [30], which introduces various evolutionary operators into each subpopulation and enhances the evolutionary operators based on individual fitness values, distances, and other information to improve the algorithm's capacity to look for potential solutions; A new algorithm, the Chaos-based Phasmatodea population evolution algorithm (CPPE) [31], was proposed by Wu et al. Chaotic mapping replaces the initialized population of the original PPE algorithm, thus improving the performance and convergence of the algorithm; Ye et al. introduced the RNSABC method [32], which uses a depth-first search strategy to improve the search capabilities of the subsequent bee and a random neighborhood architecture to ensure that each solution has a random neighborhood; Wu et al. proposed a chaos-based tumbleweed optimization algorithm (CTOA) [33], which improves population diversity and global explorability while preventing the algorithm from falling into a local optimum by incorporating 12 common chaotic maps into the optimization process; By enhancing the explosion scheme, including a GS Gaussian explosion operator, and including a depth information sharing mechanism, Chen et al. enhanced the Fireworks algorithm [34]; The quasi-oppositional chaotic symbiotic biological search algorithm (QOCSOS) was developed by Truong and colleagues to increase the algorithm's search exploitation capacity [35]. This algorithm incorporates two search techniques, such as chaotic local search and quasi-oppositional learning; Mahmoud et al. suggested the improved whale optimization salp swarm algorithm (IWOSSA) [36], which mixes salp swarm and the improved whale optimization approach; By introducing velocity and recall to aid persons during the localization phase and a refraction-based developing strategy, a velocity-based butterfly optimization algorithm (VBOA) (proposed by Long et al.) [37] significantly boosts the variety of the population and the algorithm's capacity for exploration.

From the above study on the improvement of the existing meta-heuristic algorithm, it can be seen that compared with the original algorithm, the performance of the improved algorithm has a certain degree of improvement, but there is still a large space for improvement of convergence performance and stability performance when solving complex problems. The advantages of the IECO algorithm proposed in this paper are: (1) It can effectively balance the convergence speed and population diversity, which is conducive to the further improvement of the convergence speed and convergence accuracy of the algorithm. (2) Its better stability makes it more competitive in solving complex problems.

3. Elephant clan optimization algorithm.

In the natural world, elephant populations are typically divided into male herds and family herds. These family herds are led by female matriarchs who demonstrate impressive learning and memory skills. These skills allow them to lead their herds on long-distance migrations in search of suitable resources for survival. Inspired by this behavior, Jafari et al. proposed the elephant clan optimization algorithm (ECO). In ECO, each individual in the algorithm represents the location of an elephant, and the fitness value represents the availability of survival resources in that location. The algorithm consists of three stages: initialization and herd division, individual herd update, and individual herd replacement. Algorithm 1 provides the pseudo-code for ECO, and the key phases of the algorithm are briefly explained below.

3.1. Population initialization and clan delineation.

Suppose the dimension of the problem to optimize is D, and the population X contains a total of N individual elephants. The initial position X_i^0 of each elephant is randomly generated according to Equation (1) at the beginning of the algorithm.

$$X_i^0(j) = X^{\min}(j) + r \times \left(X^{\max}(j) - X^{\min}(j)\right) \tag{1}$$

Where, $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, D, X^{\max}(j)$ and $X^{\min}(j)$ are the upper and lower bounds of the j-th dimension of the search space for the optimization issue, and r is a random integer with uniform distribution over the range $[0, 1]$.

After generating the initial population X, N individual elephants in X were randomly and equally distributed among Nc populations. $N_c - 1$ clans are family clans, and 1 clan is a male elephant clan; each clan contains a total of $N_e(N_e = N/N_c)$ individual elephants.

3.2. Individual renewal of family clans.

In each family clan, the individual elephant with the best fitness score was the matriarch of the family clan, and the other individual elephants were clan members. Clan members and matriarchs use different evolutionary approaches for individual renewal as follows.

(1) Individual renewal of clan members

Each clan member moves toward their family clan matriarch and completes self-renewal according to Equation (2):

$$X_{FCi,m}^{it+1}(j) = X_{FCi,m}^{it}(j) + r \times \alpha \times \left[X_{FCi,M}^{it}(j) - X_{FCi,m}^{it}(j)\right] \tag{2}$$

Where $X_{FCi,M}^{it}$ and $X_{FC,m}^{it}$ represent the matriarch of the i-th $(i = 1, 2, \ldots, N_c - 1)$ family clan X_{FC_i} at it-th iteration and the m-th $(m = 1, 2, \ldots, N_e)$ clan member after sorting, respectively. r is a uniformly distributed random number in the range $[0, 1]$. α is a scaling

```
Algorithm 1: ECO
  Input  : Optimization problem dimension: $D$; the number of populations: $N_c$; the
           number of members per population: $N_e$; the number of mating available:
           $N_m$; scaling factor: $\alpha$, $\beta$; the maximum number of iterations: $Maxiter$
  Output: Optimal solution and its fitness value
1 Variable initialization ($N_d$, $N_c$, $N_e$, $N_m$, $\alpha$, $\beta$, $Maxiter$)
2 Individual elephant locations were initialized as described in Section 3.1, and the
   herd was divided into the $N_c - 1$ family herd and 1 male herd.
3 Calculation of fitness values for each individual in each clan
4 while $it <= Maxiter$ do
5     Update each elephant's position in each family clan as described in Section 3.2.
6     Update each elephant's position in the male herd as described in Section 3.3.
7     Calculate the fitness value for each of the new elephants.
8     Sorting of the elephants in each of the clans
9     for $I = 1 : N_c - 1$ do
10        Determine the worst baby elephant, departing adult elephant, and mating
           elephant in the ith family clan in the manner described in Section 3.4.
11        As described in Section 3.4, two new individuals were created to replace the
           worst baby elephant and the departing adult elephant in this family clan.
12        Departing adult elephants replaced the worst elephant in the male herd in a
           2.4-section manner.
13        Update the worst elephants in the male elephant population
14    end
15 end
16 Output of the optimal global solution
```

factor that controls the balance between algorithm exploration and development. It is advised that α be set to 2 , although different values can be used.

(2) Female elephant matriarch updated

According to Equation (3), the matriarch of the family clan of female elephants completes self-renewal based on the ideal location of the entire population.

$$X_{FCi,M}^{it+1}(j) = X_{FCi,M}^{it}(j) + r \times \beta \times \left[X_{Best}^{it}(j) - X_{FCi,M}^{it}(j)\right] \tag{3}$$

Where $X_{\text{Best}}^{it}(j)$ represents the j-th dimension of the best position obtained by the entire population containing all elephants in i-th generation. r is a uniformly distributed random number in the range $[0,1]$. β is a scaling factor that regulates the balance of algorithm exploration and development. It is advised to set it to 2, however, various numbers can be used depending on the optimization problem.

3.3. **Individual renewal of male elephant populations.** The elephants in the male herd move aimlessly in the search space, as shown in Equation (4).

$$X_{MC,n}^{it+1}(j) = X_{MC,n}^{it}(j) + r_1 \times p \times \left[X^{\max}(j) - X^{\min}(j)\right] \tag{4}$$

Where $X_{MC,n}^{i+1}(j)$ is the j-th dimension of the n-th ($n = 1, 2, \ldots, N_e$) elephant in the male elephant population X_{MC} of the $it+1$-th iteration. r_l is a uniformly distributed random number in the range $[-1,1]$. As indicated in Equation (5), p is a parameter that gets smaller with each iteration and controls the typical bishop's range of movement in the search space.

$$p = 1 - \left(c \times \frac{it}{it_{\max}} \right) \tag{5}$$

Where $it_{\max}$ and it are the maximum and current repetitions. c is a fixed parameter, usually set to 0.5 for better results, but it can also be modified according to the optimization of your problems.

3.4. **Replacement of specific individuals from each ethnic clan.** In ECO, the worst juvenile elephants in the family clans will be replaced, and an elephant will leave the family clan, while only the worst calves in the male clan may be replaced. The specific way is as follows.

(1) Replacement of specific individuals in the family clan

Each family clan contains mating elephants and adult elephants. Among them, the mating elephant is the Nm elephant with the best fitness value in that family clan (in general, N_m is set to 3, but it can be modified according to the optimization problem). In contrast, an adult elephant is an elephant that leaves its home colony and is randomly selected from the remaining N_e-N_m-1 elephants except for the mating elephants and the elephants with the poorest fitness values.

The worst individual and the adult elephant individual in each family clan are each replaced by two new individuals, generated according to Equation (6), as follows.

$$X_{FCi,Call}^{it+1}(j) = r \times \left(\frac{X_{FCi,Rf}^{i+1}(j) + X_{MC,Rm}^{it+1}(j)}{2} \right) \tag{6}$$

Where $X_{FCi,Calf}^{it+1}$ denotes a newly generated baby elephant in the family colony X_{FCi}^{it+1} in $it+1$-th iteration, $X_{FCi,Rf}^{it+1}$ ($Rf \in [1, \ldots, Nm]$) indicates a randomly selected individual among the mating elephants in $it+1$-th iteration, and $X_{MC,Rm}^{it+1}$ denotes the best elephant in the male settlement X_{MC} in $it+1$ iteration; r is a uniformly distributed random number in the range $[0, 1]$.

(2) Replacement of the worst individuals in the male clan

The worst elephants in the male clan may be replaced by adult elephants leaving the respective family clan, as shown in Equation (7).

$$X_{MC,\text{ worst}}^{it+1} = \begin{cases} X_{FCi,Gm}^{it+1}, & \text{if } f\left(X_{FCi,Gm}^{it+1}\right) < f\left(X_{MC,\text{ worst}}^{it+1}\right) \\ X_{MC,\text{ worst}}^{i++1}, & \text{else} \end{cases} \tag{7}$$

Where $X_{MC,\text{ worst}}^{it+1}$ is the worst elephant in the $it+1$-th iteration of the male clan, and $X_{FCi,Gm}^{it+1}$ is the adult elephant that left the family clan X_{FCi}^{it+} in the $it+1$-th iteration.

4. **Improved elephant clan optimization algorithm.** The elephant clan optimization algorithm has to be enhanced in order to increase convergence speed and accuracy. This paper makes comprehensive improvements to each essential operation of the ECO algorithm. It suggests an improved elephant clan optimization process, represented by the flowchart in Figure 1.

4.1. **Initialization phase improvements.** (1) Method for initializing a population with a strategy for additional autonomous movement

The ECO algorithm simulates the behavior of elephants as they migrate under the leadership of each clan leader. Of course, this fast access to elite evolutionary information inevitably speeds up the algorithm's convergence considerably. However, it also allows for rapid population aggregation, significantly reducing population diversity. If the ideal

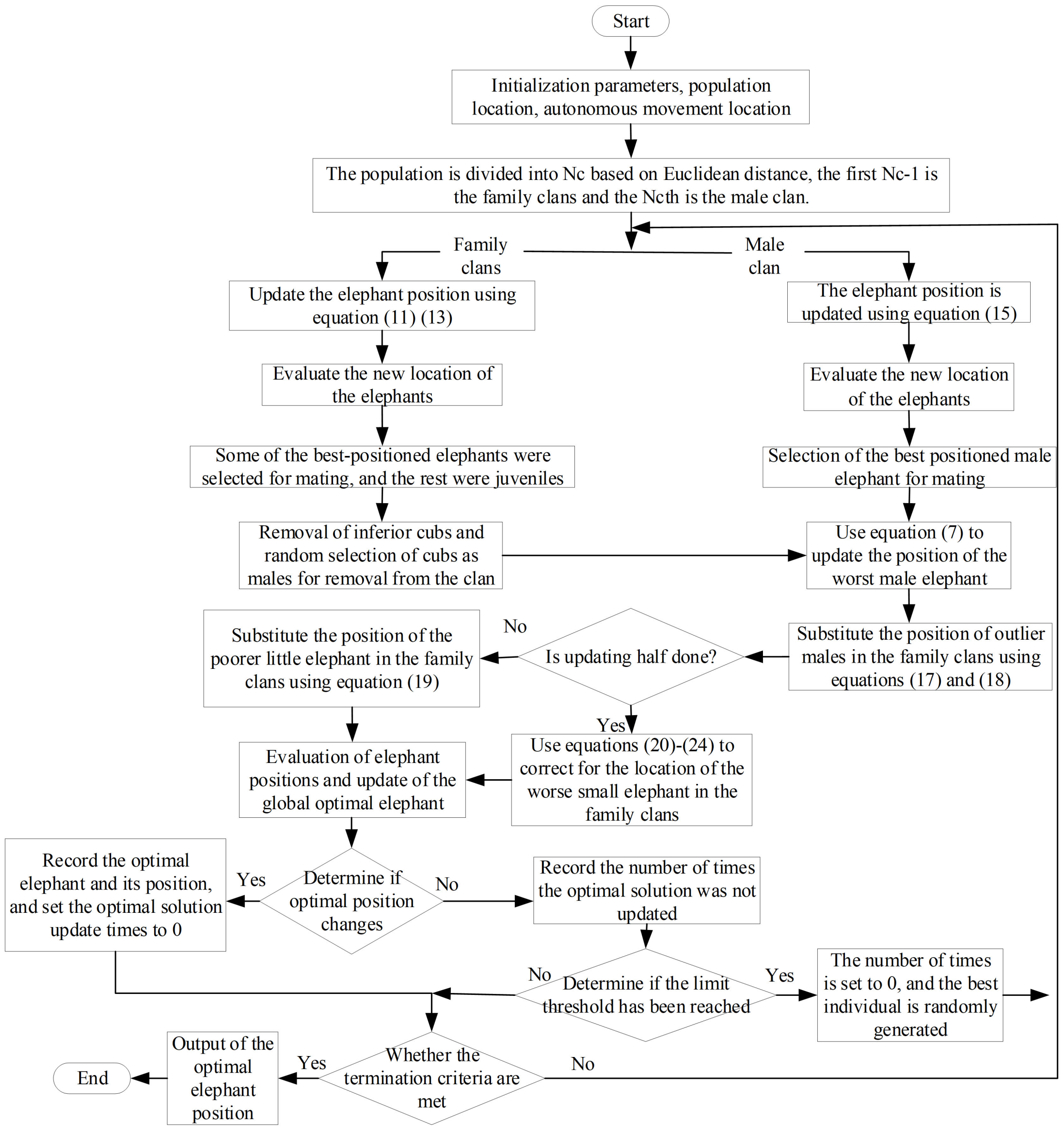

Start
Initialization parameters, population location, autonomous movement location
The population is divided into Nc based on Euclidean distance, the first Nc-1 is the family clans and the Ncth is the male clan.
Family clans
Male clan
Update the elephant position using equation (11) (13)
Evaluate the new location of the elephants
Some of the best-positioned elephants were selected for mating, and the rest were juveniles
Removal of inferior cubs and random selection of cubs as males for removal from the clan
The elephant position is updated using equation (15)
Evaluate the new location of the elephants
Selection of the best positioned male elephant for mating
Use equation (7) to update the position of the worst male elephant
Substitute the position of outlier males in the family clans using equations (17) and (18)
Is updating half done?
No
Yes
Substitute the position of the poorer little elephant in the family clans using equation (19)
Use equations (20)-(24) to correct for the location of the worse small elephant in the family clans
Evaluation of elephant positions and update of the global optimal elephant
Determine if optimal position changes
Yes
No
Record the optimal elephant and its position, and set the optimal solution update times to 0
Record the number of times the optimal solution was not updated
Determine if the limit threshold has been reached
No
Yes
The number of times is set to 0, and the best individual is randomly generated
Whether the termination criteria are met
Yes
No
Output of the optimal elephant position
End

FIGURE 1. Flowchart of the IECO algorithm

individual's location differs from the actual theoretical position and is at a certain local peak, the algorithm may stall and enter a local optimum.

Indeed, all-natural elephants have autonomous consciousness and carry out small independent movements around their environment, which is a significant factor preventing the rapid aggregation of elephant populations. To prevent the algorithm from falling into a local optimum to increase the diversity of the population and simulate the above behaviors based on the original elephant position, this section adds an autonomous movement range and an autonomous movement position for each elephant. This is shown in Equations (8) and (9).

$$\Delta X_i^0(j) = \Delta X^{\min}(j) + r \times \left(\Delta X^{\max}(j) - \Delta X^{\min}(j)\right) \tag{8}$$

Where $\Delta X_i^0(j)(i = 1, 2, \ldots, N, j = 1, 2, \ldots, D)$ is the range of autonomous movement of the i-th elephant in the j-th dimension at the initial time, r is a uniformly distributed random number in the range $[0, 1]$. $\Delta X^{\max}(j) = F \times \left(X^{\max}(j) - X^{\min}(j)\right)$ and $\Delta X^{\min}(j) = -\Delta X^{\max}(j)$ correspond to the upper and lower limits of the j-th dimensional autonomous motion space. In general, F can be taken as 0.005 for better results or can be set by yourself.

$$ZX_i^0(j) = X_i^0(j) + \Delta X_i^0(j) \tag{9}$$

As individuals move closer together as evolution progresses, each elephant's autonomous range of movement should also decrease. Considering this, the independent moving range updating method shown in Equation (10) is proposed.

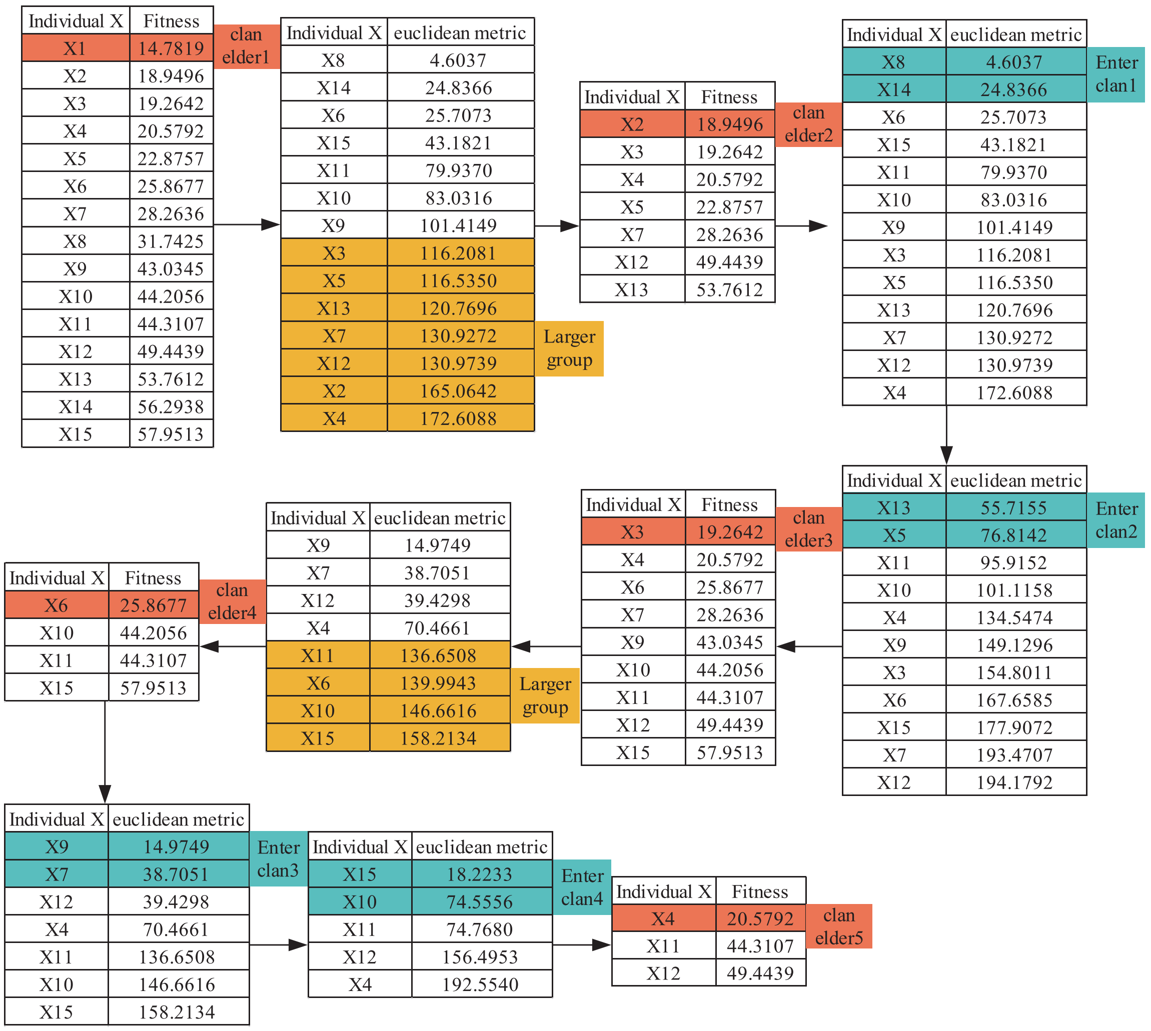
$$\Delta X_i^{it+1}(j) = \left[0.9 - \left(0.8 \times \frac{it}{it_{\max}}\right)\right] \times \Delta X_i^{it}(j) \tag{10}$$

(2) Euclidean distance-based clan delimitation

ECO is susceptible to the following two scenarios when clan divisions are made and each elephant is randomly and equally assigned to each clan: (1) Some clans have a concentration of several quality elephants, while individuals in individual clans are relatively inferior. For those clans composed of relatively poor individuals, their matriarchs will also be poorer, and it will be difficult for the various elephants within that clan to quickly gain a better position under their leadership. This will have an effect on the algorithm's convergence speed. (2) The patriarchs of certain clans are very close to each other, and these clans are very susceptible to aggregation with evolution under the leadership of the patriarch, leading to a rapid loss of population diversity. In summary, the patriarchs of each clan should be relatively good and non-aggregated. With this in mind, the following Euclidean distance-based clan delineation is proposed.

Step 1, identify the patriarch. Assuming a total of N_c clans, the most outstanding $3N_c$ individuals are selected first, and the most impressive individual is made the patriarch of the first clan. Then, the Euclidean distance between each individual and clan leader 1 is calculated, divided into two groups according to the size of the Euclidean distance, and the most outstanding individual in the group with the more considerable Euclidean distance is selected as the leader of the second clan. Again, the distance between the remaining individuals other than the patriarch and these two patriarchs is calculated, and the two individuals closest to each patriarch are included as members of their clan. Finally, the above patriarchs and clan members are removed, and the above process is repeated to identify other patriarchs and clan members from the remaining members until all individuals are divided into individual clans.

To better understand the process of determining the patriarchs. Figure 2 shows the specific determination process for the optimized 2-dimensional rotated Katsuura function. It is assumed that it is divided into 5 clans. First, Choose the member with the highest fitness value x1 as clan leader 1, and determine the distance in Euclidean space between each of the remaining individuals and x1, where {x8, x14, x6, x15, x11, x10, x9} are closer to x1, while {x3, x5, x13, x7, x12, x2, x4} are farther from x1. Among the further away individuals, x2 has the highest fitness value and is chosen as clan leader 2. Then compute the Euclidean distances to x1 and x2 for the remaining individuals except x1 and x2, where {x8, x14} is closest to patriarch x1 and is split into clan 1, while {x13, x5} is closest to patriarch x2 and is divided into clan 2. Again, except for clan 1 and clan 2, the remaining 9 individuals, including {x3, x4, x6, x7, x9, x10, x11, x12, x15}, The clan leader of clan 3 is chosen as the individual with the highest fitness value x3. and the distance between each individual in {x3, x4, x6, x7, x9, x10, x11, x12, x15} and x3 is calculated, and in the more distant group { x11, x6, x10, x15}, individual x6 has the best fitness value and is selected as clan elder 4, and the Euclidean distances of individuals {x4, x7, x9, x10, x11, x12, x15} from x3 and x6 are calculated, where {x9, x7} is closest to clan elder x3 and is divided to clan 3, while {x15, x10} is most comparable to clan elder x6 and is divided to clan 4. Finally, the best individual from the remaining individuals {x4, x11, x12} is selected as the patriarch of clan 5.



Individual X	Fitness
X1	14.7819
X2	18.9496
X3	19.2642
X4	20.5792
X5	22.8757
X6	25.8677
X7	28.2636
X8	31.7425
X9	43.0345
X10	44.2056
X11	44.3107
X12	49.4439
X13	53.7612
X14	56.2938
X15	57.9513

Individual X	euclidean metric
X8	4.6037
X14	24.8366
X6	25.7073
X15	43.1821
X11	79.9370
X10	83.0316
X9	101.4149
X3	116.2081
X5	116.5350
X13	120.7696
X7	130.9272
X12	130.9739
X2	165.0642
X4	172.6088

Individual X	Fitness
X2	18.9496
X3	19.2642
X4	20.5792
X5	22.8757
X7	28.2636
X12	49.4439
X13	53.7612

Individual X	euclidean metric
X8	4.6037
X14	24.8366
X6	25.7073
X15	43.1821
X11	79.9370
X10	83.0316
X9	101.4149
X3	116.2081
X5	116.5350
X13	120.7696
X7	130.9272
X12	130.9739
X4	172.6088

Individual X	euclidean metric
X13	55.7155
X5	76.8142
X11	95.9152
X10	101.1158
X4	134.5474
X9	149.1296
X3	154.8011
X6	167.6585
X15	177.9072
X7	193.4707
X12	194.1792

Individual X	Fitness
X3	19.2642
X4	20.5792
X6	25.8677
X7	28.2636
X9	43.0345
X10	44.2056
X11	44.3107
X12	49.4439
X15	57.9513

Individual X	euclidean metric
X9	14.9749
X7	38.7051
X12	39.4298
X4	70.4661
X11	136.6508
X6	139.9943
X10	146.6616
X15	158.2134

Individual X	Fitness
X6	25.8677
X10	44.2056
X11	44.3107
X15	57.9513

Individual X	euclidean metric
X9	14.9749
X7	38.7051
X12	39.4298
X4	70.4661
X11	136.6508
X10	146.6616
X15	158.2134

Individual X	euclidean metric
X15	18.2233
X10	74.5556
X11	74.7680
X12	156.4953
X4	192.5540

Individual X	Fitness
X4	20.5792
X11	44.3107
X12	49.4439

FIGURE 2. Schematic illustrating the process used to determine the Patriarch

Step 2, clan partitioning. N_c clan leaders and individual clan members are identified in step 1, and the Euclidean distance between other individuals and each clan leader is calculated. Each clan leader, in turn, selects an individual with the smallest Euclidean distance to join that clan until all individuals are divided.

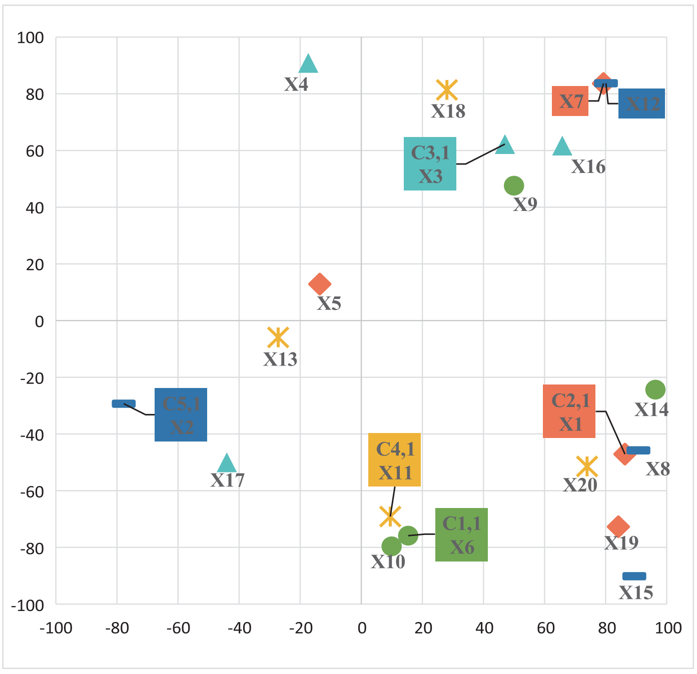
To examine the effect of the Euclidean distance-based clan division method proposed in this section, an optimized 2-dimensional Rotated Katsuura function is used for the clan division of 20 individuals, and the results of random uniform division and Euclidean distance-based division are shown in Figure 3. With the original random division, the clan leaders of clans 1 and 4 are close to each other, and the fitness value of clan leader X11 of clan 4 is 44.3107, which is a poor individual. It can be seen that clan 2 has more excellent individuals gathered in it, while with the Euclidean distance-based clan division, there is no close distance between the clan leaders, and the fitness values of each clan member have evenly distributed. The problem of clustering of better or worse individuals does not occur.

(a) Graph based on random uniform division results

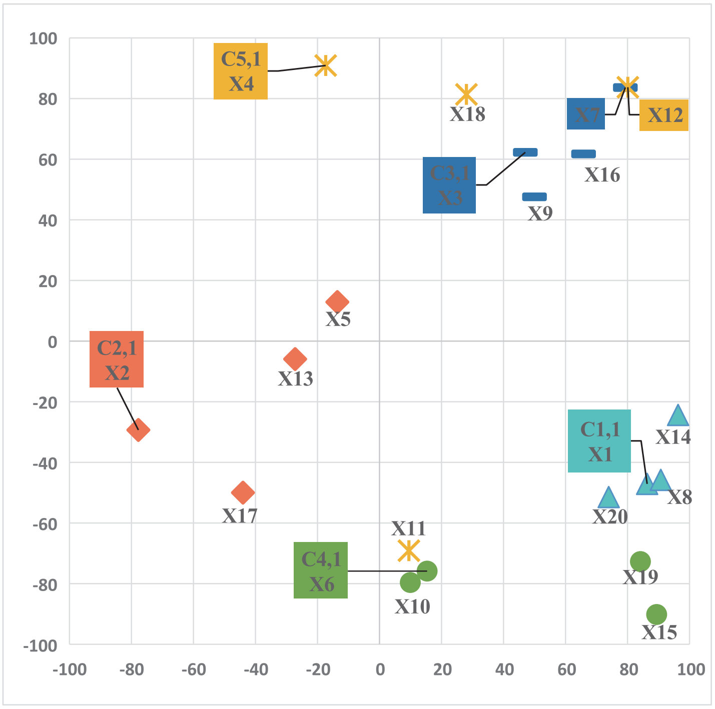
(b) Graph of division results based on Euclidean distance

In summary, compared with the original random uniform division method, each clan leader in this section of community division belongs to relatively superior individuals,

Individual X	Fitness
X1	14.7819
X2	18.9496
X3	19.2642
X4	20.5792
X5	22.8757
X6	25.8677
X7	28.2636
X8	31.7425
X9	43.0345
X10	44.2056
X11	44.3107
X12	49.4439
X13	53.7612
X14	56.2938
X15	57.9513
X16	60.9276
X17	62.8982
X18	71.2812
X19	71.8048
X20	75.0631



(a) Graph based on random uniform division results



(b) Graph of division results based on Euclidean distance

FIGURE 3. Comparison of random uniform division and Euclidean distance division results

ensuring the algorithm's convergence speed. In addition, clan leaders do not cluster among clan leaders within the constraints of Euclidean distance, thus ensuring that clans do not cluster prematurely and thus ensuring population diversity.

(3) Premature suppression strategy

Like other swarm intelligence algorithms, ECO suffers from the phenomenon that as evolution proceeds, the distribution of individuals gradually becomes denser, the population diversity decreases, the motivation for further evolution is lacking, and the population appears to be caught in a local optimum. In the artificial bee colony algorithm, when an individual's fitness value remains constant over successive limit generations, the individual's current position is dropped, and a new position is generated to take part in the evolution of the following generation. This procedure increases the algorithm's ability to depart from the local optimum to some extent. Based on this concept, this study introduces a control parameter limit: if the optimal fitness value does not change in subsequent limit generations, a new individual is created at random in the search space to replace the original optimal individual.

Many experiments have confirmed that if the limit is too large, the time to catch the local optimum is too long. As evolution progresses, all individuals grow more similar to the ideal individual, and the likelihood of leaving the optimum region diminishes considerably. While the limit value is too small, the frequent introduction of other evolutionary information may change the population's evolutionary direction and reduce the algorithm's convergence speed. After several experiments, we found that the general limit is set to 10 to achieve better results, but also, according to the specific problems, set their own.

4.2. **Improvement of individual renewal method of family clans.** In each family clan, the best individual is the mother elephant, from whom all clan members learn. The mother elephant is largely in charge of rapidly exploring the region where the theoretical ideal location is situated, while clan members are in charge of preserving population variety in order to supply the mother elephant with excellent evolutionary information for rapid convergence. While this relatively single-learning strategy can boost convergence time by some amount, it significantly reduces species diversity and raises the chance of the algorithm being caught in a local optimum. Given this, the individual renewal of clan members and maternal elephants in family clans is improved.

(1) An individual update approach for clan members based on autonomous location traction

This section suggests a method to update the individual population members based on autonomous location traction, as indicated in Equation (11), for better maintenance of species diversity and not degrade the algorithm's convergence speed too much.

$$X_{FCi,m}^{it+1}(j) = \begin{cases} ZX_{FCi,m}^{it+1}(j) + r \times \alpha \times \left[X_{FC_{NC-i},M}^{it}(j) - X_{FCi,m}^{it}(j)\right] \\ +r \times \alpha \times \left[X_{MC,Rm}^{it}(j) - X_{FCi,m}^{it}(j)\right], \text{ if } m > Ne/2 \\ ZX_{FCi,m}^{it+1}(j) + r \times \alpha \times \left[X_{FCi,M}^{it}(j) - X_{FCi,m}^{it}(j)\right] \\ +r \times \alpha \times \left[X_{MC,Rm}^{it}(j) - X_{FCi,m}^{it}(j)\right], \text{ else} \end{cases} \tag{11}$$

Where $X_{FC_{Nc-i},M}^{it}$ represents the matriarch of the female elephant in the $Nc - i$ th family group $X_{FC_{Nc-i}}$ at the i-th iteration. $ZX_{FCi,m}^{it+1}$ is the autonomous position of the $m(m = 1, 2, \dots, N_e)$ th clan member after sorting at $it + 1$ iteration, and r is a uniformly distributed random number in the range $[0, 1] \cdot \alpha$ is the improved adaptive scaling factor, as shown in Equation (12).

$$\alpha = 2 - \left(c \times \frac{it}{it_{\max}}\right) \tag{12}$$

Where c as a fixed value is usually set to 0.5 to achieve the best results, but other values can be set depending on the optimization problem itself.

In conclusion, the clan member individual renewal approach based on autonomous location traction described in this paragraph has the following strengths over the original method: To begin, unlike the old method of learning exclusively from this clan's most excellent female elephant, the learning objects in this part include the best male person as well as females from other clans. The females of other clans that each clan has learned from are not identical, so the relatively abundant information on outstanding evolution gives each elephant more outstanding evolutionary directions, which does not slow down the algorithm's convergence speed but unquestionably can further increase population diversity. In addition, those superior clan members with relatively similar evolutionary information to the mother elephant from whom they are learning can quickly move closer to the mother and provide her with more refined information. And those members of the clan who are genetically different from their mothers learn from the mothers of neighboring clans, improving communication with other regions, providing richer evolutionary information for the clan, and avoiding the algorithm's emergence of a local optimum. Second, the small search range does not reduce the convergence speed of the algorithm, and the introduction of autonomous locations significantly improves population diversity. Third, the scale factor controls the search range of individuals, and this section transforms the original constant scale factor into a form that decreases gradually with iteration. In the early stages of evolution, each individual is far from the theoretical optimum, and a more extensive search range allows individuals to move quickly toward the region where the optimum academic lies, thus accelerating the algorithm's convergence. In the late evolutionary stage, the individuals are not very different and are closer to the theoretical optimum, and the smaller search range is more conducive to refined search, thus accurately locking in the theoretical optimum.

(2) An individual update approach for matriarchs based on autonomous location traction

As mentioned above, the matriarchs in the ECO algorithm are all traversed by the globally optimal individuals and quickly approach the globally optimal region. However, many experiments have confirmed insufficient population diversity for specific complex optimization problems to explore more refined global optimal solutions. Considering that the autonomous movement position of an individual is the result of individual exploration in a small area, which has a weak impact on the convergence rate of the algorithm while providing more evolutionary direction and diversity, this section proposes an autonomous position traction-based matriarch updating approach as shown in Equation (13).

$$X_{FCi,M}^{it+1}(j) = ZX_{FCi,M}^{it+1}(j) + r \times \beta \times \left[X_{Best}^{it}(j) - X_{FCi,M}^{it}(j)\right] \tag{13}$$

Where $ZX_{FCi,M}^{it+1}$ is the autonomous movement position of the matriarch in this clan at the $it + 1$ iteration, obtained similarly to Equation (11); the scale factor β is calculated according to Equation (14).

$$\beta = 3 - \left(c \times \frac{it}{it_{\max}}\right) \tag{14}$$

As can be seen from Equations (13) and (14), the scale factor gradually decreases as the iteration proceeds, which can ensure that in the early evolutionary stage, these matriarchs, which are more different from the optimal global individuals, learn more from the optimal global individuals to make sure that the algorithm converges fast. In contrast, the gap between matriarchs and optimal global individuals is fragile in the late evolutionary stage. A small amount of learning from the optimal global individuals, focusing more on fine exploration near themselves, can further enhance the possibility of locking the optimal global. Furthermore, suppose a female elephant individual is the optimal global individual. In that case, its position does not change at all when the update of the female individual is performed according to Equation (3) in the original ECO, while Equation (13) proposed in this section relocates the individual's position to the autonomous moving place, which provides more evolutionary information and avoids the waste of computational resources caused by the original invalid search.

4.3. **Improvement of the individual renewal method of the male elephant clan.** An in-depth analysis of the ECO algorithm shows that the role of the male elephant clan is mainly twofold: First, the desire to generate globally optimal positions to provide the female clan leader with an incentive to evolve. Second, it is used to replace some members of the family clan to provide more evolutionary information to family members. However, Equation (4) shows that male elephant individuals use a random way to search for new positions in the defined domain, and this blind random search can ensure a good diversity of male elephant populations. However, it is not very easy to produce new individuals better than the matriarchs and obviously cannot provide excellent evolutionary information for the matriarchs. Furthermore, even if evolutionary information from the male elephant population flows into the family clan through replacement operations, the male population is highly susceptible to being eliminated in the evolutionary process due to the extremely low probability of being genetically superior to family clan members and cannot add to the population diversity of the family clan.

Given this, the male elephant individual renewal formula is given in Equation (15) to ensure that the male elephant clan produces as much evolutionary information as feasible and has a specific population diversity.

$$X_{MC,n}^{it+1}(j) = ZX_{MC,n}^{it+1}(j) + r \times p \times \left(X_{\text{Center}}^{it} - X_{MC,n}^{it}(j)\right) \tag{15}$$

Where $ZX_{MC,n}^{it+1}$ is the autonomous movement position of the n-th ($n = 1, 2, \ldots, N_e$) elephant in the $it+1$ st iteration of the male elephant clan X_{MC}. p is calculated according to Equation (5) to control the step size of the male bishop towards the center. X_{Center}^{it} is the center of the maternal elephant patriarch in all family clans of the it-th iteration, as shown in Equation (16).

$$X_{\text{Center}}^{it} = \frac{1}{Nc-1} \times \sum_{i=1}^{Nc-1} X_{FCi,M}^{it} \tag{16}$$

In summary, each male elephant individual learns from the center of the female matriarch in all family clans. Since each female matriarch carries almost the best current evolutionary information, it can provide the most excellent evolutionary motivation to the male individual, which ensures the excellence of the new male individual and increases the possibility of producing individuals that are more excellent than the female matriarch, which in turn promotes the rapid convergence of the algorithm. In addition, since the center of the matriarchs is in a different location from each matriarch, i.e., the male elephant population evolves along a different evolutionary direction from each family group, the gradual reduction of the control step by parameter ensures that the new male individuals are different from the other matriarchs. The algorithm's ability to explore other regions of excellence, aided by the autonomous movement of position, is increased, further complementing the diversity of the population.

4.4. **Improvement of individual replacement strategy for part of the family clan.** As seen in Section 3.4, an adult elephant and the worst baby elephant in each family clan are replaced by two new individuals generated by Equation (6). The unique individual is equivalent to the result of a random search with the origin as the center and the distance from the source to the center of the best male elephant individual and the better individual in this family clan as the search range, as shown in Equation (6).

Although the central position of the best male individual and the better individual in this family clan is highly likely to have a better fitness value, the random search is more blind, and the new individual is most likely inferior to the adult elephant and the worst baby elephant. The direct replacement is likely to cause the evolutionary resources of the family clan to regress, which in turn reduces the convergence speed of the algorithm and even leads to deviation from the evolutionary trajectory of evolutionary excellence and, thus, ineffective merit search.

In view of this, the method of creating new individuals and the process of replacing them will be improved as follows.

(1) Improve the adult elephant replacement strategy

Since adult elephants are not the superior elephants within this clan, the following replacement of adult elephants is proposed to ensure the algorithm's convergence speed and increase the population's diversity. If an adult elephant's fitness value is less than its clan's average fitness value, it is replaced by the central position of all clan members, as shown in Equation (17). Otherwise, the better individual is selected from both the original adult elephant individual and the new individual, as shown in Equation (18), to replace the actual adult elephant.

$$X_{FCi,Gm}^{it+1}(j) = \frac{1}{Ne}\sum_{i=1}^{Ne} X_{FCi}^{it+1}(j) \tag{17}$$

$$X_{FCi,Calf}^{it+1}(j) = X_{FCi,Gm}^{it+1}(j) + r \times \left[\frac{X_{FCi,Rf}^{it+1}(j) + X_{MC,Rm}^{it+1}(j)}{2} - X_{FCi,Gm}^{i+1}(j)\right] \tag{18}$$

An in-depth study reveals that the adult elephant replacement approach described in this section outperforms the previous adult elephant replacement strategy in the following ways: First, if the position evaluation of the adult elephant is inferior to the average position evaluation of the clan in which it is located, the position evaluation of the central position of all members of that clan has a higher probability of being better than the new individual generated according to Equation (6) and the adult elephant itself, and if the position evaluation of the adult elephant is better than the average position evaluation of the clan in which it is located, the new adult individual must not be inferior to the original adult elephant. Obviously, the current section approach has significantly better evolutionary information and, thus, a better convergence speed. Second, from Equation (18), the new individual retains the evolutionary information accumulated by the adult elephants in the evolutionary process and makes full use of the superior evolutionary details of the better elephant individuals and the best male individuals in the clan, which not only preserves the population's variety but also enhances the likelihood of the new individual outperforming the original adult elephants, thus increasing the convergence speed of the algorithm. Third, this section uses different evolutionary approaches to generate unique individuals according to the degree of superiority and inferiority of adult elephants, allowing individuals with a higher-than-average rank to retain their superior evolutionary information while introducing information from other outstanding individuals. At the same time, individuals with inferior rank carry too few adequate evolutionary details, which is not conducive to the evolution of the population, and are replaced by the central position of members, which enriches the evolutionary information of new individuals to a certain extent, increases the population's diversity, and increases the algorithm's ability to handle a variety of optimization challenges.

(2) Improve the inferior small elephant replacement strategy

As described in Section 3.4, except for adult elephants, the ECO algorithm replaces only the worst individuals in family clans. These inferior individuals, while good for maintaining population diversity, significantly reduce the convergence speed of the algorithm. In addition, in the early iterations of the algorithm, individuals are far from the theoretical optimal position and have significant individual differences, which can enhance the convergence of the algorithm. In the late iteration of the algorithm, individuals do not differ much and should focus on providing population diversity to avoid the algorithm getting caught in the local optimum. To boost the algorithm's convergence speed and retain as much population variety as possible, the worst 0.3Ne tiny elephants in each family clan were immediately replaced by creating new individuals in the evolutionary stage in the following manner.

In the pre-evolutionary period, i.e., $it < itmax$, new individuals are generated by Equation (19).

$$X_{FCi,Calf}^{it+1}(j) = X_{FCi,w}^{it+1}(j) + r \times \left[\frac{X_{FCi,Rf}^{it+1}(j) + X_{MC,Rm}^{it+1}(j)}{2} - X_{FCi,w}^{it+1}(j) \right] \tag{19}$$

Where $X_{FCi,w}^{it+1}$ is the worst individual to be replaced in the $it + 1$ iteration of the family clan $X_{FC_i}^{it+1}$.

In the late stage of evolution, i.e., $it \geq itmax$, the position correction is performed using the following simplex method to generate new individuals.

First, the reflection point $X_{FC_i,r}$ of the individual $X_{FC_i,w}^{it+1}$ to be corrected is calculated according to Equation (20).

$$X_{FCi,r} = X_{FCi,c} + a \times (X_{FCi,c} - X_{FCi,w}) \tag{20}$$

The reflectance coefficient, generally set to 1. $X_{FC_i,c}$ is the midpoint of the best individual $X_{FCi,best}$ and the second-best individual $X_{FCi,er}$ that this clan has searched for so far, as shown in Equation (21).

$$X_{FCi,c} = \frac{X_{FCi,best} + X_{FCi,er}}{2} \tag{21}$$

Then a new individual $X'_{FCi,w}$ is generated by the fitness relationship between individuals, replacing $X_{FCi,w}$ if $X'_{FCi,w}$ is superior to $X_{FCi,w}$. Otherwise, leave $X_{FCi,w}$ unchanged. When $Fit\,(X_{FCi,r}) < Fit\,(X_{FCi,best})$, the expansion operation is performed using Equation (22). When Fit $(X_{FCi,r}) > Fit\,(X_{FCi,w})$, the compression operation is performed using Equation (23); otherwise, the contraction operation is performed using Equation (24).

$${X_{FCi,w}}' = X_{FCi,c} + \lambda \times (X_{FCi,r} - X_{FCi,c}) \tag{22}$$

Where λ is the expansion factor, which is generally set to 2.

$${X_{FCi,w}}' = X_{FCi,c} + \eta \times (X_{FCi,w} - X_{FCi,c}) \tag{23}$$

Where η is the compression factor, which is generally set to 0.5.

$${X_{FCi,w}}' = X_{FCi,c} - \mu \times (X_{FCi,w} - X_{FCi,c}) \tag{24}$$

Where μ is the shrinkage factor, which is generally set to 0.5.

In conclusion, the inferior individual replacement technique presented in this section provides the following benefits: First, the population diversity is more adequate and the difference between individuals is more significant in the early stages of algorithm evolution, according to Equation (19). The inferior small elephant learns from the superior individuals in the clan, which preserves some of its genes to a certain extent, maintains population diversity, and improves its evolutionary information quickly to shorten the gap between it and the optimal individuals, thus improving the convergence speed of the algorithm. Second, at later stages of evolution, individuals are less different from each other, and the rate of development of more superior individuals relying on the available evolutionary information slows down significantly. The simplex method can effectively improve the minimum by internal compression, external compression, mapping, and expansion of the minimum within the simplex, which increases the possibility of the new individual outperforming the original one and can enhance the algorithm's convergence speed by some amount. More importantly, the simplex method differs from the authentic method of individual crossover variation and has the potential to generate new good individuals in other new regions, providing new evolutionary information for algorithmic evolution and further complementing population diversity.

5. **Experimental results and analysis.**

Experiments are conducted in two portions of this study to validate the effectiveness of the IECO algorithm. (1) Validate the effectiveness of the four recommended improvement options. (2) Compare the modified IECO algorithm's performance against that of the original ECO algorithm and four additional more typical evolutionary algorithms with good performance.

For the experimental simulation in this part, the CEC2013 test set is utilized. 28 test functions are included in the CEC2013 test set, of which F1 through F5 are single-peaked functions with a single ideal value and are used to assess how well the algorithm achieves convergence. Multi-peak functions like F6–F20 are utilized to show how the method might depart from the local optimum since they have a number of locally optimal solutions. The composite functions are F21 through F28. The CEC2013 test set is described in the literature. [38].

To assure the fairness of the algorithm comparisons, all algorithms were run on a Windows 10 computer with an Intel(R) Core(TM) i7-7700HQ CPU operating at 2.80 GHz and programmed in MATLAB R2021a.

5.1. **Validation of the efficacy of each improvement measure.**

Section 3 demonstrates four ways in which the IECO method outperforms the ECO algorithm. In this study, the appropriate improvement strategy in IECO is eliminated in order to verify the efficacy of the four improvement strategies, and four new improvement algorithms are created in their place. This includes an improved algorithm for removing the initialization phase improvement strategy in Section 3.1 of IECO, an enhanced algorithm for removing the family clan individual update method improvement strategy in Section 3.2 of IECO, an improved algorithm for removing the male elephant clan individual update method improvement strategy in Section 3.3 of IECO, and an improved algorithm for removing the family clan individual replacement strategy in Section 3.4 of IECO. For ease of reference, the four new and improved algorithms above are referred to as IECO1, IECO2, IECO3, and IECO4, respectively, and are compared to the IECO algorithm on the CEC2013 test set.

To ensure fairness, the number of populations in each method is set to N = 50, the dimension of the test function is set to D = 30, and the maximum number of function evaluations is set to MaxFEs = 150000. Table 1 shows how the additional parameters of each algorithm are configured. To exclude the possibility of a single algorithm operation, each algorithm is performed separately 30 times on each test function.

Table 2 displays the running results of each algorithm on 28 test functions in 30 dimensions, where "$\pm$" represents the average and standard deviation of the optimal value in 30 tests, respectively, and excludes an improvement method for the same function. The data for which the enhanced method is clearly inferior to the IECO algorithm is highlighted in black. The Friedman test [39] and the Wilcoxon rank sum test with a significance

Table 1. Algorithm related parameters

Algorithm	Initial parameters
IECO	$numberC = 5$; $Swarmsize = 10$; $Nmat = 3$; $limit = 10$
IECO1	$numberC = 5$; $Swarmsize = 10$; $Nmat = 3$
IECO2	$numberC = 5$; $Swarmsize = 10$; $Nmat = 3$; $limit = 10$; $Alpha = 1.5$; $Beta = 1.5$
IECO3	$numberC = 5$; $Swarmsize = 10$; $Nmat = 3$; $limit = 10$
IECO4	$numberC = 5$; $Swarmsize = 10$; $Nmat = 3$; $limit = 10$

level of 5% were conducted between the IECO algorithm and the improved algorithms to further confirm the distinctions between the two, and the results are displayed in Tables 3 and 4, respectively. When the p-value in Table 3 is greater than 0.05, it means that there is no significant difference between the corresponding enhanced algorithm and the IECO method, as shown by the symbol "=". If the p-value is less than 0.05 and the average value of the optimal solution obtained in 30 trials of the corresponding improved algorithm is better than the IECO algorithm, it indicates that the corresponding improved algorithm is significantly better than the IECO algorithm, which is indicated by the symbol "+". Otherwise, the symbol "−" indicates that the performance of the associated upgraded method is considerably lower than that of the IECO algorithm. The algorithm performs better overall if the average ranking that it correlates to in Table 4 is lower.

Table 2. The comparison results of each of the improved algorithms and the IECO algorithm in the 30-dimensional CEC2013 test set.

	IECO	IECO1	IECO2	IECO3	IECO4
F1	$0.00E+00 \pm 0.00E+00$	$\mathbf{1.26E-27 \pm 3.04E-27}$	$\mathbf{7.15E+00 \pm 3.33E+01}$	$\mathbf{2.28E+03 \pm 7.99E+02}$	$\mathbf{6.33E-12 \pm 1.70E-11}$
F2	$1.45E+06 \pm 1.56E+06$	$1.24E+06 \pm 1.77E+06$	$\mathbf{1.77E+07 \pm 1.49E+07}$	$\mathbf{1.01E+08 \pm 4.06E+07}$	$\mathbf{3.39E+06 \pm 1.58E+06}$
F3	$0.00E+00 \pm 0.00E+00$	$0.00E+00 \pm 0.00E+00$	$\mathbf{4.44E+07 \pm 2.43E+08}$	$0.00E+00 \pm 0.00E+00$	$0.00E+00 \pm 0.00E+00$
F4	$1.75E+04 \pm 6.13E+03$	$1.72E+04 \pm 1.36E+04$	$\mathbf{7.43E+04 \pm 9.28E+03}$	$\mathbf{1.19E+05 \pm 1.95E+04}$	$\mathbf{3.73E+04 \pm 6.68E+03}$
F5	$0.00E+00 \pm 0.00E+00$	$\mathbf{6.60E-19 \pm 3.54E-18}$	$\mathbf{6.72E-08 \pm 1.72E-07}$	$\mathbf{1.63E+03 \pm 6.69E+02}$	$\mathbf{6.80E-09 \pm 1.15E-08}$
F6	$3.23E+01 \pm 2.89E+01$	$\mathbf{4.94E+01 \pm 3.59E+01}$	$\mathbf{9.24E+01 \pm 4.53E+01}$	$\mathbf{3.64E+02 \pm 1.13E+02}$	$\mathbf{5.55E+01 \pm 2.77E+01}$
F7	$0.00E+00 \pm 0.00E+00$	$0.00E+00 \pm 0.00E+00$	$\mathbf{9.02E+02 \pm 1.36E+03}$	$0.00E+00 \pm 0.00E+00$	$0.00E+00 \pm 0.00E+00$
F8	$2.12E+01 \pm 4.96E-02$	$2.10E+01 \pm 4.79E-02$	$2.11E+01 \pm 1.18E-01$	$\mathbf{2.12E+01 \pm 6.36E-02}$	$\mathbf{2.12E+01 \pm 5.88E-02}$
F9	$0.00E+00 \pm 0.00E+00$	$0.00E+00 \pm 0.00E+00$	$\mathbf{1.27E+01 \pm 1.27E+01}$	$0.00E+00 \pm 0.00E+00$	$0.00E+00 \pm 0.00E+00$
F10	$2.96E-01 \pm 1.63E-01$	$\mathbf{3.16E-01 \pm 1.80E-01}$	$\mathbf{2.35E+01 \pm 4.48E+01}$	$\mathbf{5.53E+02 \pm 1.85E+02}$	$\mathbf{3.40E-01 \pm 1.61E-01}$
F11	$0.00E+00 \pm 0.00E+00$	$\mathbf{3.32E-01 \pm 1.64E+00}$	$\mathbf{1.54E+01 \pm 1.60E+01}$	$0.00E+00 \pm 0.00E+00$	$0.00E+00 \pm 0.00E+00$
F12	$0.00E+00 \pm 0.00E+00$	$0.00E+00 \pm 0.00E+00$	$\mathbf{1.51E+02 \pm 3.05E+01}$	$0.00E+00 \pm 0.00E+00$	$0.00E+00 \pm 0.00E+00$
F13	$0.00E+00 \pm 0.00E+00$	$0.00E+00 \pm 0.00E+00$	$\mathbf{1.96E+02 \pm 3.90E+01}$	$0.00E+00 \pm 0.00E+00$	$0.00E+00 \pm 0.00E+00$
F14	$1.41E+03 \pm 3.48E+02$	$\mathbf{1.73E+03 \pm 3.90E+02}$	$\mathbf{2.96E+03 \pm 3.96E+02}$	$\mathbf{7.31E+03 \pm 1.10E+03}$	$1.05E+03 \pm 2.85E+02$
F15	$2.77E+03 \pm 6.91E+02$	$\mathbf{3.26E+03 \pm 8.14E+02}$	$\mathbf{4.02E+03 \pm 5.71E+02}$	$\mathbf{8.68E+03 \pm 3.96E+02}$	$\mathbf{3.19E+03 \pm 7.74E+02}$
F16	$4.35E+00 \pm 6.04E-01$	$2.61E+00 \pm 1.93E-01$	$6.99E-01 \pm 2.61E-01$	$\mathbf{4.44E+00 \pm 7.04E-01}$	$\mathbf{4.70E+00 \pm 5.07E-01}$
F17	$3.40E+01 \pm 1.35E+01$	$\mathbf{5.46E+01 \pm 1.89E+01}$	$\mathbf{9.22E+02 \pm 1.77E+02}$	$\mathbf{4.45E+02 \pm 8.64E+01}$	$\mathbf{5.36E+01 \pm 1.46E+01}$
F18	$9.87E+01 \pm 7.53E+01$	$5.93E+01 \pm 3.42E+01$	$\mathbf{1.00E+03 \pm 2.25E+02}$	$\mathbf{4.81E+02 \pm 1.24E+02}$	$\mathbf{1.68E+02 \pm 6.13E+01}$
F19	$2.97E+00 \pm 7.08E-01$	$\mathbf{3.82E+00 \pm 1.48E+00}$	$\mathbf{1.66E+03 \pm 1.18E+03}$	$\mathbf{1.19E+02 \pm 9.22E+01}$	$\mathbf{1.40E+01 \pm 5.27E+00}$
F20	$0.00E+00 \pm 0.00E+00$	$\mathbf{6.00E+00 \pm 7.47E+00}$	$\mathbf{1.45E+01 \pm 4.53E-03}$	$\mathbf{1.35E+01 \pm 4.57E+00}$	$\mathbf{4.27E+00 \pm 6.59E+00}$
F21	$4.00E+02 \pm 0.00E+00$	$\mathbf{4.00E+02 \pm 1.90E-13}$	$\mathbf{4.12E+02 \pm 5.24E+01}$	$\mathbf{7.95E+02 \pm 1.56E+02}$	$\mathbf{4.00E+02 \pm 1.51E-06}$
F22	$1.44E+03 \pm 4.13E+02$	$\mathbf{1.72E+03 \pm 6.51E+02}$	$\mathbf{4.04E+03 \pm 7.44E+02}$	$\mathbf{8.30E+03 \pm 1.14E+03}$	$1.08E+03 \pm 3.21E+02$
F23	$3.17E+03 \pm 7.89E+02$	$3.14E+03 \pm 6.74E+02$	$\mathbf{5.18E+03 \pm 7.91E+02}$	$\mathbf{8.95E+03 \pm 4.97E+02}$	$\mathbf{3.21E+03 \pm 9.04E+02}$
F24	$2.03E+02 \pm 9.52E+00$	$2.00E+02 \pm 1.35E-01$	$\mathbf{2.60E+02 \pm 3.71E+01}$	$\mathbf{2.17E+02 \pm 1.96E+01}$	$1.97E+02 \pm 1.82E+01$
F25	$2.98E+02 \pm 2.31E+01$	$2.46E+02 \pm 4.93E+01$	$\mathbf{3.13E+02 \pm 2.87E+01}$	$\mathbf{3.07E+02 \pm 2.82E+01}$	$2.30E+02 \pm 1.09E+01$
F26	$2.97E+02 \pm 2.94E+01$	$\mathbf{3.04E+02 \pm 2.64E+01}$	$\mathbf{3.38E+02 \pm 6.34E+01}$	$\mathbf{3.04E+02 \pm 4.62E+01}$	$\mathbf{3.00E+02 \pm 3.16E-06}$
F27	$4.13E+02 \pm 2.06E+02$	$3.34E+02 \pm 8.05E+01$	$\mathbf{1.07E+03 \pm 2.93E+02}$	$\mathbf{6.33E+02 \pm 2.82E+02}$	$3.16E+02 \pm 1.10E+00$
F28	$8.21E+02 \pm 2.00E+01$	$7.80E+02 \pm 1.85E+02$	$\mathbf{4.17E+03 \pm 4.50E+02}$	$\mathbf{2.26E+03 \pm 3.59E+02}$	$\mathbf{9.08E+02 \pm 8.25E+01}$

According to Table 2, there are 13 functions, 26 functions, 22 functions, and 17 functions, respectively, where the performance of the aforementioned four enhanced algorithms is less than that of the IECO method. The Wilcoxon rank sum test results of the IECO1 algorithm and the IECO algorithm on four test functions are "+," suggesting that the IECO1 method outperforms the IECO algorithm on these four functions, as shown in Table 3. The Wilcoxon rank sum test results for eight test functions are "−", indicating that the IECO1 algorithm outperforms the IECO technique on these eight test functions.

Table 3. Wilcoxon rank sum test results of each improved algorithm and IECO algorithm

Function	p-value(vs.IECO)			
	IECO1	IECO2	IECO3	IECO4
F1	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F2	0.122(=)	0.000(−)	0.000(−)	0.000(−)
F3	1.000(=)	0.334(=)	1.000(=)	1.000(=)
F4	0.057(=)	0.000(−)	0.000(−)	0.000(−)
F5	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F6	0.047(−)	0.000(−)	0.000(−)	0.004(−)
F7	1.000(=)	0.000(−)	1.000(=)	1.000(=)
F8	0.000(+)	0.002(+)	0.000(−)	0.000(−)
F9	1.000(=)	0.000(−)	1.000(=)	1.000(=)
F10	0.784(=)	0.000(−)	0.000(−)	0.201(=)
F11	0.161(=)	0.000(−)	1.000(=)	1.000(=)
F12	1.000(=)	0.000(−)	1.000(=)	1.000(=)
F13	1.000(=)	0.000(−)	1.000(=)	1.000(=)
F14	0.002(−)	0.000(−)	0.000(−)	0.000(+)
F15	0.028(−)	0.000(−)	0.000(−)	0.022(−)
F16	0.000(+)	0.000(+)	0.492(=)	0.022(−)
F17	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F18	0.102(=)	0.000(−)	0.000(−)	0.000(−)
F19	0.030(−)	0.000(−)	0.000(−)	0.000(−)
F20	0.000(−)	0.000(−)	0.000(−)	0.001(−)
F21	1.000(=)	0.000(−)	0.000(−)	0.000(−)
F22	0.176(=)	0.000(−)	0.000(−)	0.000(+)
F23	0.842(=)	0.000(−)	0.000(−)	0.807(=)
F24	0.299(+)	0.000(−)	0.000(−)	0.004(+)
F25	0.000(+)	0.003(−)	0.000(−)	0.000(+)
F26	0.893(=)	0.000(−)	0.039(−)	0.000(−)
F27	0.270(=)	0.000(−)	0.001(−)	0.206(=)
F28	0.348(=)	0.000(−)	0.000(−)	0.000(−)
+/ = /−	4/16/8	2/1/25	0/7/21	4/9/15

Table 4. Friedman test results of the five algorithms

	IECO	IECO1	IECO2	IECO3	IECO4
Avg.rank	1.77	2.27	4.39	4.05	2.52
sort	1	2	5	4	3

In comparison, the Wilcoxon rank sum test results on the remaining 16 test functions are "=", indicating that their performance is comparable. The IECO2 algorithm outperforms the IECO algorithm on 2 test functions, is identical to the IECO algorithm on 1 test function, and is inferior to the IECO algorithm on 25 test functions. IECO3 performs similarly to the IECO algorithm on seven tested functions and inferiorly to the IECO algorithm on 21 tested functions. The IECO4 algorithm outperforms the IECO algorithm on 4 test functions, is similar to the IECO algorithm on 9 test functions, and is inferior to the IECO algorithm on 15 test functions. It can be seen from Table 4 that after removing one of the corresponding improved strategies in IECO, the rank mean of the corresponding improved algorithm is larger than that of the IECO algorithm, indicating that all four improved strategies proposed in this paper have a significant impact on the overall performance of the IECO algorithm.

In conclusion, the four improvement tactics described in this research are extremely effective. The improvement techniques in sections 3.2 and 3.3 have a greater influence on the performance of the IECO algorithm, whereas the other two improvement algorithms do not differ much in terms of IECO algorithm performance.

5.2. **IECO's performance in contrast to other algorithms.** To verify whether the performance of the IECO algorithm is superior in terms of convergence accuracy and speed, this section compares it with the ECO algorithm and four previously excellent evolutionary algorithms on the CEC2013 test set. These include the backward-learning enhanced sine cosine algorithm (ESCA) [40], the improved crow search algorithm (ICSA) [41], the enhanced Archimedean classification feature selection algorithm (EAOA) [42], and the dual population-based artificial tree algorithm (IATTP) [43]. The maximum number of function evaluations is MaxFEs = 150000, and the number of populations is N = 50 for all algorithms to guarantee that the comparison is fair. Table 5 shows the additional parameter settings of each method, where the settings of each comparing algorithm are consistent with the original text.

Table 5. Initial parameter settings for each algorithm

Algorithm	Initial Parameters
IECO	numberC = 5; Swarmsize = 10; Nmat = 3; limit = 10
ECO	numberC = 5; Swarmsize = 10; Nmat = 3; Alpha = 1.5; Beta = 1.5
ESCA	Pc = 0.6, r2 = 2*pi*rand(0,1), r3 = 2*rand(0,1), r4 = r5 = r6 = r7 = r8 = rand(0,1)
ICSA	AP = 0.1; FL = 1.5
EAOA	C1 = 2; C2 = 6; C3 = 2; C4 = 0.5
IATTP	h1 = h2 = h3 = 0.5; h4 = 0.8; m = 50, q = 0.8

(1) Comparison of the convergence accuracy of the IECO algorithm with that of other algorithms

In order to fully compare the performance of the IECO algorithm and other algorithms in terms of convergence accuracy, the tests are carried out on the CEC2013 test sets of two different dimensions, $D = 30$ and $D = 100$. On the 30- and 100-dimensional CEC2013 datasets, the mean and standard deviation of 30 individual experiments for each method are displayed in Tables 6 and 7, respectively. The function that achieves the best optimization on the same function is marked in black. A Wilcoxon rank-sum test between the IECO algorithm and the other algorithms was carried out with a 5% level of significance in order to further confirm the distinctions between the two. The results are displayed in Table 8. The Fridman test results are included in Table 9 for a thorough analysis of the overall performance of all methods.

Table 6 indicates that the IECO method achieves the global optimum on nine test functions, including F1, F3, F5, F7, F9, F11, F12, F13, and F20 for the 30-dimensional optimization issue. On the F3, F7, and F9 test functions, the ECO algorithm reached the global optimum. However, the IECO algorithm completed better mean values than the ECO algorithm on the remaining test functions; ESCA only produced theoretically ideal outcomes for F3 and F11; ICSA obtained theoretically ideal values for F3, F7, F9, F12, and F13 functions; EAOA obtained the theoretical optimum on F11 only; Six functions, including F3, F7, F9, F11, F12, and F13, have the theoretical optimum determined via IATTP. As demonstrated in Table 8, the ECO method performs similarly to the IECO algorithm on four functions but considerably worse on 24 tested functions; ESCA performs similarly on seven functions but significantly worse on 21 functions. ICSA's performance was level on 5 of the 23 assessed functions but considerably poorer on the remaining 23 functions. EAOA performs much better on F16 alone but significantly worse on the rest of the 25 functions. Only two of the test functions have a considerably higher performance than the IATTP, while the other eighteen have a significantly lower performance. In conclusion, for the 30-dimensional CEC2013 test set, compared with other representative algorithms, the IECO algorithm proposed in this paper has obvious advantages in convergence accuracy.

Table 6. Data results of the IECO algorithm and other algorithms on the 30-dimensional CEC2013 test set

	IECO		ECO		ESCA		ICSA		EAOA		IATTP	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	**0.00E+00**	**0.00E+00**	2.17E+01	3.55E+00	4.09E+03	7.40E+02	4.10E+01	4.42E+01	1.75E+03	4.88E+02	4.40E+00	1.58E+00
F2	**1.45E+06**	**1.56E+06**	1.01E+07	3.33E+06	1.71E+08	3.81E+07	2.15E+07	6.63E+06	7.53E+07	2.60E+07	1.19E+07	4.54E+06
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**	3.79E+06	1.12E+07	**0.00E+00**	**0.00E+00**
F4	1.75E+04	6.13E+03	1.07E+05	1.77E+04	5.37E+04	5.60E+03	1.31E+04	3.17E+03	5.36E+04	5.91E+03	**1.16E+03**	**5.78E+02**
F5	**0.00E+00**	**0.00E+00**	9.76E+00	1.99E+00	4.80E+03	1.13E+03	2.66E+02	1.50E+02	9.51E+02	2.91E+02	2.03E+01	8.50E+00
F6	**3.23E+01**	**2.89E+01**	5.35E+01	2.62E+01	5.13E+02	1.58E+02	1.29E+02	3.10E+01	2.15E+02	5.80E+01	8.24E+01	2.20E+01
F7	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.95E+00	2.02E+01	**0.00E+00**	**0.00E+00**	1.45E+01	2.02E+01	**0.00E+00**	**0.00E+00**
F8	**2.10E+01**	**3.37E-02**	2.10E+01	5.35E-02	2.10E+01	5.77E-02	2.10E+01	4.35E-02	2.10E+01	6.21E-02	2.10E+01	5.95E-02
F9	**0.00E+00**	**0.00E+00**	2.37E+00	7.03E+00	2.58E+00	9.86E+00	**0.00E+00**	**0.00E+00**	3.98E+00	8.86E+00	**0.00E+00**	**0.00E+00**
F10	**2.96E-01**	**1.63E-01**	7.70E+00	1.86E+00	1.12E+03	1.95E+02	1.06E+02	5.49E+01	6.51E+02	1.50E+02	2.44E+01	9.23E+00
F11	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.98E-01	8.51E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F12	**0.00E+00**	**0.00E+00**	8.70E+01	6.87E+01	7.09E+00	3.88E+01	**0.00E+00**	**0.00E+00**	1.61E+02	4.07E+01	**0.00E+00**	**0.00E+00**
F13	**0.00E+00**	**0.00E+00**	1.07E+01	3.07E+01	1.91E+01	5.83E+01	**0.00E+00**	**0.00E+00**	1.65E+02	5.81E+01	**0.00E+00**	**0.00E+00**
F14	**1.41E+03**	**3.48E+02**	3.92E+03	4.60E+02	7.32E+03	4.48E+02	2.69E+03	7.60E+02	4.19E+03	3.73E+02	6.79E+03	3.03E+02
F15	**2.77E+03**	**6.91E+02**	5.43E+03	4.44E+02	7.61E+03	2.54E+02	6.64E+03	6.17E+02	5.38E+03	6.37E+02	6.86E+03	7.74E+02
F16	4.35E+00	6.04E-01	2.98E+00	3.20E-01	2.77E+00	3.40E-01	2.67E+00	2.93E-01	**2.30E+00**	**3.79E-01**	2.67E+00	3.33E-01
F17	**3.40E+01**	**1.35E+01**	2.17E+02	1.82E+01	5.61E+02	6.35E+01	8.90E+01	2.91E+01	4.41E+02	4.59E+01	2.04E+02	1.34E+01
F18	**9.87E+01**	**7.53E+01**	2.85E+02	1.53E+01	5.64E+02	5.03E+01	2.13E+02	2.78E+01	4.58E+02	6.41E+01	2.22E+02	1.90E+01
F19	**2.97E+00**	**7.08E-01**	1.86E+01	1.40E+00	8.69E+02	5.97E+02	1.15E+01	3.79E+00	8.78E+01	6.06E+01	1.74E+01	2.08E+00
F20	**0.00E+00**	**0.00E+00**	1.45E+01	9.72E-01	2.65E+00	5.41E+00	6.95E+00	4.80E+00	1.41E+01	1.09E+00	7.16E-01	2.48E+00
F21	**4.00E+02**	**0.00E+00**	4.15E+02	6.50E+01	6.02E+02	2.87E+01	4.29E+02	3.09E+01	5.42E+02	4.13E+01	4.02E+02	4.47E-01
F22	**1.44E+03**	**4.13E+02**	4.05E+03	5.42E+02	7.61E+03	3.57E+02	2.27E+03	7.99E+02	4.95E+03	6.16E+02	7.04E+03	3.74E+02
F23	**3.17E+03**	**7.89E+02**	6.07E+03	5.82E+02	7.98E+03	3.24E+02	5.08E+03	1.35E+03	6.60E+03	8.65E+02	7.58E+03	3.89E+02
F24	**1.97E+02**	**1.83E+01**	2.03E+02	1.17E+01	3.03E+02	3.87E+00	2.02E+02	5.23E+00	2.40E+02	3.49E+01	2.00E+02	1.12E-01
F25	2.98E+02	2.31E+01	2.77E+02	3.44E+01	3.02E+02	3.72E+00	2.76E+02	2.76E+01	3.00E+02	1.01E+01	**2.22E+02**	**2.13E+01**
F26	2.97E+02	2.94E+01	2.99E+02	5.71E+01	3.67E+02	5.93E+01	2.96E+02	2.77E+01	3.54E+02	4.24E+01	**2.54E+02**	**5.04E+01**
F27	**3.13E+02**	**1.91E+01**	6.03E+02	2.85E+02	2.60E+03	1.92E+02	3.70E+02	1.35E+02	9.98E+02	1.30E+02	3.20E+02	2.19E+00
F28	**8.21E+02**	**2.00E+01**	1.06E+03	3.68E+02	2.23E+03	1.70E+02	1.70E+03	4.64E+02	3.14E+03	1.09E+03	1.02E+03	2.18E+01

Table 7 shows that the IECO method successfully solves the 100-dimensional optimization issue for the global optimum on three test functions, including F3, F7, and F11. ECO achieves the global optimum on F3 and F7, except on F6, F8, and F16, where ECO obtains better mean values than the IECO algorithm, but on the remaining 23 functions, ECO obtains worse mean values than the IECO algorithm; ESCA and EAOA achieved a global optimum on F11; Neither ICSA nor IATTP achieved the theoretical optimal value on any function. As demonstrated in Table 8, ECO performs much worse than the IECO algorithm on all 20 functions examined, with the exception of three functions, namely F6, F8, and F16; ESCA has the same performance on 2 test functions, significantly better performance on 2 functions, but significantly worse performance on 24 functions; ICSA performs equally well on four functions, noticeably better on four functions, and significantly worse on the other 20 functions examined; EAOA performs much worse on the 24 examined functions, only doing slightly better on F8 and F16; IATTP performs considerably worse on 21 of the assessed functions, while only doing significantly better on three of them. In summary, it demonstrates that the IECO algorithm described in this research has a considerable advantage in convergence accuracy when compared to existing typical approaches for high-dimensional optimization problems. The performance difference between the methods is greater than the performance gap for 30-dimensional optimization problems.

Table 7. Data results of the IECO algorithm and other algorithms on the 100-dimensional CEC2013 test set

	IECO		ECO		ESCA		ICSA		EAOA		IATTP	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	**2.32E-13**	**8.31E-13**	9.99E+01	1.14E+02	6.65E+04	6.76E+03	1.41E+04	4.57E+03	3.61E+04	5.27E+03	3.23E+03	5.81E+02
F2	**1.33E+07**	**8.86E+06**	3.94E+07	7.77E+06	9.62E+08	1.48E+08	2.35E+08	6.43E+07	1.14E+09	1.99E+08	2.28E+08	4.66E+07
F3	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.95E+14	1.06E+15	4.12E+11	3.75E+11	6.97E+14	1.45E+15	9.69E+10	1.80E+11
F4	1.26E+05	4.17E+04	3.53E+05	3.99E+04	2.14E+05	1.46E+04	9.61E+04	1.24E+04	1.55E+05	1.30E+04	**4.45E+04**	**7.90E+03**
F5	**8.13E-10**	**3.76E-09**	1.25E+01	8.42E+00	5.10E+04	8.01E+03	8.22E+03	2.70E+03	1.24E+04	2.54E+03	2.15E+03	3.70E+02
F6	3.30E+02	1.04E+02	2.82E+02	4.60E+01	8.48E+03	1.50E+03	2.04E+03	3.40E+02	5.19E+03	1.23E+03	1.05E+03	1.54E+02
F7	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	7.95E+03	7.54E+03	4.37E+02	2.29E+02	1.41E+04	1.47E+04	5.02E+02	5.01E+02
F8	2.15E+01	2.89E-02	2.13E+01	2.56E-02	**2.13E+01**	**2.31E-02**	2.13E+01	2.52E-02	2.14E+01	2.38E-02	2.14E+01	2.46E-02
F9	9.49E+01	6.17E+00	1.05E+02	7.13E+00	1.26E+02	7.36E+00	**8.09E+01**	**7.89E+00**	1.25E+02	6.23E+00	1.15E+02	7.82E+00
F10	**3.05E+00**	**1.95E+00**	5.29E+01	1.60E+01	8.12E+03	1.02E+03	2.39E+03	4.36E+02	6.92E+03	8.24E+02	1.60E+03	2.39E+02
F11	**0.00E+00**	**0.00E+00**	1.14E+01	1.39E+01	**0.00E+00**	**0.00E+00**	9.49E+01	3.21E+01	**0.00E+00**	**0.00E+00**	1.58E+00	6.32E+00
F12	**4.43E+02**	**3.80E+01**	7.36E+02	7.17E+01	1.17E+03	1.62E+02	5.77E+02	6.70E+01	1.32E+03	2.44E+02	7.23E+02	7.22E+01
F13	**5.31E+02**	**1.50E+02**	6.34E+02	5.01E+01	1.20E+03	1.35E+02	6.43E+02	5.52E+01	1.11E+03	1.38E+02	7.30E+02	6.11E+01
F14	**8.15E+03**	**9.55E+02**	1.90E+04	2.30E+03	2.90E+04	6.83E+02	1.79E+04	2.93E+03	2.44E+04	1.11E+03	3.12E+04	9.73E+02
F15	**1.35E+04**	**1.44E+03**	2.51E+04	1.09E+03	2.92E+04	9.84E+02	3.06E+04	5.50E+02	2.79E+04	1.15E+03	3.09E+04	5.94E+02
F16	5.55E+00	6.80E-01	4.41E+00	3.31E-01	**4.17E+00**	**3.32E-01**	4.40E+00	3.47E-01	4.24E+00	3.05E-01	4.40E+00	1.98E-01
F17	**3.74E+02**	**1.15E+02**	9.87E+02	9.60E+01	4.09E+03	3.33E+02	1.35E+03	1.87E+02	2.67E+03	2.28E+02	1.45E+03	9.02E+01
F18	**1.61E+02**	**2.46E+01**	1.32E+03	9.01E+01	4.14E+03	2.69E+02	1.73E+03	1.86E+02	2.95E+03	2.56E+02	1.45E+03	5.20E+01
F19	**2.57E+01**	**4.87E+00**	9.15E+01	8.79E+00	1.63E+05	8.27E+04	5.32E+03	3.15E+03	1.80E+05	8.90E+04	1.67E+03	9.35E+02
F20	**5.00E+01**	**3.56E-14**	5.00E+01	1.27E-13	5.00E+01	1.66E-09	5.00E+01	2.37E-13	5.00E+01	4.70E-09	5.00E+01	2.01E-11
F21	**3.77E+02**	**4.28E+01**	5.43E+02	1.63E+02	5.14E+03	3.74E+02	4.81E+03	9.74E+02	6.89E+03	4.35E+02	2.36E+03	9.65E+02
F22	**9.48E+03**	**1.51E+03**	2.23E+04	2.28E+03	2.98E+04	7.26E+02	1.64E+04	3.72E+03	2.69E+04	9.13E+02	3.14E+04	8.68E+02
F23	**1.54E+04**	**1.78E+03**	2.91E+04	1.90E+03	3.21E+04	9.67E+02	2.98E+04	1.75E+03	3.22E+04	9.93E+02	3.29E+04	7.42E+02
F24	**3.14E+02**	**9.49E+01**	3.33E+02	1.17E+02	6.11E+02	5.29E+00	4.20E+02	2.91E+01	5.79E+02	2.01E+01	4.01E+02	8.02E+01
F25	**5.51E+02**	**2.59E+01**	5.57E+02	1.74E+01	6.06E+02	4.97E+00	5.49E+02	1.70E+01	5.97E+02	1.70E+01	5.53E+02	1.68E+01
F26	4.46E+02	6.55E+01	5.12E+02	9.79E+01	6.97E+02	6.99E+00	**4.42E+02**	**4.66E+01**	5.98E+02	3.02E+01	4.56E+02	9.48E+01
F27	**1.71E+03**	**9.35E+02**	1.92E+03	1.11E+03	5.67E+03	5.35E+01	1.98E+03	4.00E+02	3.70E+03	2.50E+02	2.65E+03	5.60E+02
F28	**6.61E+03**	**2.70E+03**	4.39E+03	1.78E+03	1.50E+04	8.97E+02	9.75E+03	1.20E+03	1.61E+04	3.07E+03	1.00E+04	2.19E+03

Table 9 shows that the IECO method performs the best overall on the 30- and 100-dimensional test functions among the six optimization techniques and that this advantage becomes more pronounced as the dimensionality of the optimization issue rises. IATTP placed second in total performance on the 30-dimensional test function, with ICSA and ECO performing marginally worse than IATTP. The overall performance of ECO ranks second in the 100-dimensional test function, while ICSA, IATTP, and EAOA perform somewhat worse than ECO. The ESCA algorithm has the poorest overall performance of the six algorithms. In summary, compared with other algorithms, the IECO algorithm proposed in this paper has certain advantages in terms of convergence accuracy.

(2) Comparison of the IECO algorithm with other algorithms in terms of convergence speed

The convergence curves of each algorithm on the 28 test functions of the CEC2013 test set are shown in Figure 4 in order to more easily examine the variations in the algorithms' rates of convergence when the dimension is 30. The abscissa is the number of function evaluations, and the ordinate is the logarithm of the fitness value.

It can be seen from Figure 4 that for the single-mode functions F1–F5, the IECO algorithm can obtain the global optimal value on the F1, F3, and F5 test functions and at the same time show the fastest convergence speed. Compared to the other five algorithms on F2, the IECO method has the highest convergence speed and accuracy. The IECO algorithm's convergence accuracy and speed on F4 are second only to ICSA and IATTP. The IECO method can find the global optimal value on the F7, F9, F11, F12, F13, and F20 test functions for the multi-mode functions F6–F20, and its convergence speed on F12 is second only to ECO and ICSA. On F6, F8, F10, F14, F17, and F18, the IECO method outperforms the other five algorithms in terms of convergence accuracy and speed. The convergence accuracy and convergence speed of the IECO algorithm only on F16 are second only to EAOA and ICSA; On the F15 and F19 test functions, the IECO algorithm exhibits better convergence accuracy and speed than other algorithms. While the IECO

Table 8. Wilcoxon rank sum test results for IECO and other algorithms on the CEC2013 test set

Function	D = 30 p-value(vs.IECO)					D = 100 p-value(vs.IECO)				
	ECO	ESCA	ICSA	EAOA	IATTP	ECO	ESCA	ICSA	EAOA	IATTP
F1	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F2	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F3	1.000(=)	1.000(=)	1.000(=)	0.006(−)	1.000(=)	1.000(=)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F4	0.000(−)	0.000(−)	0.003(−)	0.000(−)	0.000(+)	0.000(−)	0.000(−)	0.002(+)	0.000(−)	0.000(+)
F5	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F6	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.017(+)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F7	1.000(=)	0.161(=)	1.000(=)	0.000(−)	1.000(=)	1.000(=)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F8	0.569(=)	0.003(−)	0.000(−)	0.000(−)	0.000(−)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)
F9	0.042(−)	0.161(=)	1.000(=)	0.001(−)	1.000(=)	0.000(−)	0.000(−)	0.000(+)	0.000(−)	0.000(−)
F10	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F11	1.000(=)	1.000(=)	0.003(−)	1.000(=)	1.000(=)	0.000(−)	1.000(=)	0.000(−)	1.000(=)	0.082(=)
F12	0.000(−)	0.334(=)	1.000(=)	0.000(−)	1.000(=)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F13	0.022(−)	0.082(=)	1.000(=)	0.000(−)	1.000(=)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F14	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F15	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F16	0.000(−)	0.000(−)	0.000(−)	0.000(+)	0.000(−)	0.000(+)	0.000(+)	0.000(+)	0.000(+)	0.000(+)
F17	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F18	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F19	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F20	0.000(−)	0.011(−)	0.000(−)	0.000(−)	0.082(=)	1.000(=)	0.334(=)	1.000(=)	0.334(=)	1.000(=)
F21	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F22	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F23	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
F24	0.001(−)	0.000(−)	0.000(−)	0.000(−)	0.001(−)	0.290(−)	0.000(−)	0.000(−)	0.000(−)	0.001(−)
F25	0.003(−)	0.340(=)	0.000(−)	0.234(=)	0.000(+)	0.631(=)	0.000(−)	0.420(=)	0.000(−)	0.796(=)
F26	0.009(−)	0.000(−)	0.000(−)	0.000(−)	0.583(=)	0.000(−)	0.000(−)	0.438(=)	0.000(−)	0.348(=)
F27	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.001(−)	0.158(=)	0.000(−)	0.673(=)	0.000(−)	0.000(−)
F28	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)	0.000(−)
+/=/−	0/4/24	0/7/21	0/5/23	1/2/25	2/8/18	3/5/20	2/2/24	4/4/20	2/2/24	3/4/21

Table 9. Friedman test for each algorithm

		IECO	ECO	ESCA	ICSA	EAOA	IATTP
D=30	**Avg. rank**	1.82	3.50	5.27	2.98	4.70	2.73
	sort	1	4	6	3	5	2
D=100	**Avg. rank**	1.77	2.63	4.95	3.18	4.79	3.70
	sort	1	2	6	3	5	4

algorithm's convergence speed is a little slower in the early stages of evolution than other algorithms', by the late stages of evolution, when other algorithms' search performance has decreased, the IECO algorithm is still able to find a better solution than other algorithms. For the composite functions F21–F28, the IECO method outperforms the other five algorithms in terms of convergence accuracy and speed on the test functions F22, F26, and F28; The IECO algorithm exhibits higher convergence accuracy and speedier convergence on F23 and F25. The convergence speed of the IECO algorithm on F21, F24, and F27 is similar to that of ECO, ICSA, and IATTP, where the convergence accuracy of the IECO algorithm is higher. To summarize, the IECO method provides certain benefits in terms of convergence speed when compared to the other five algorithms.

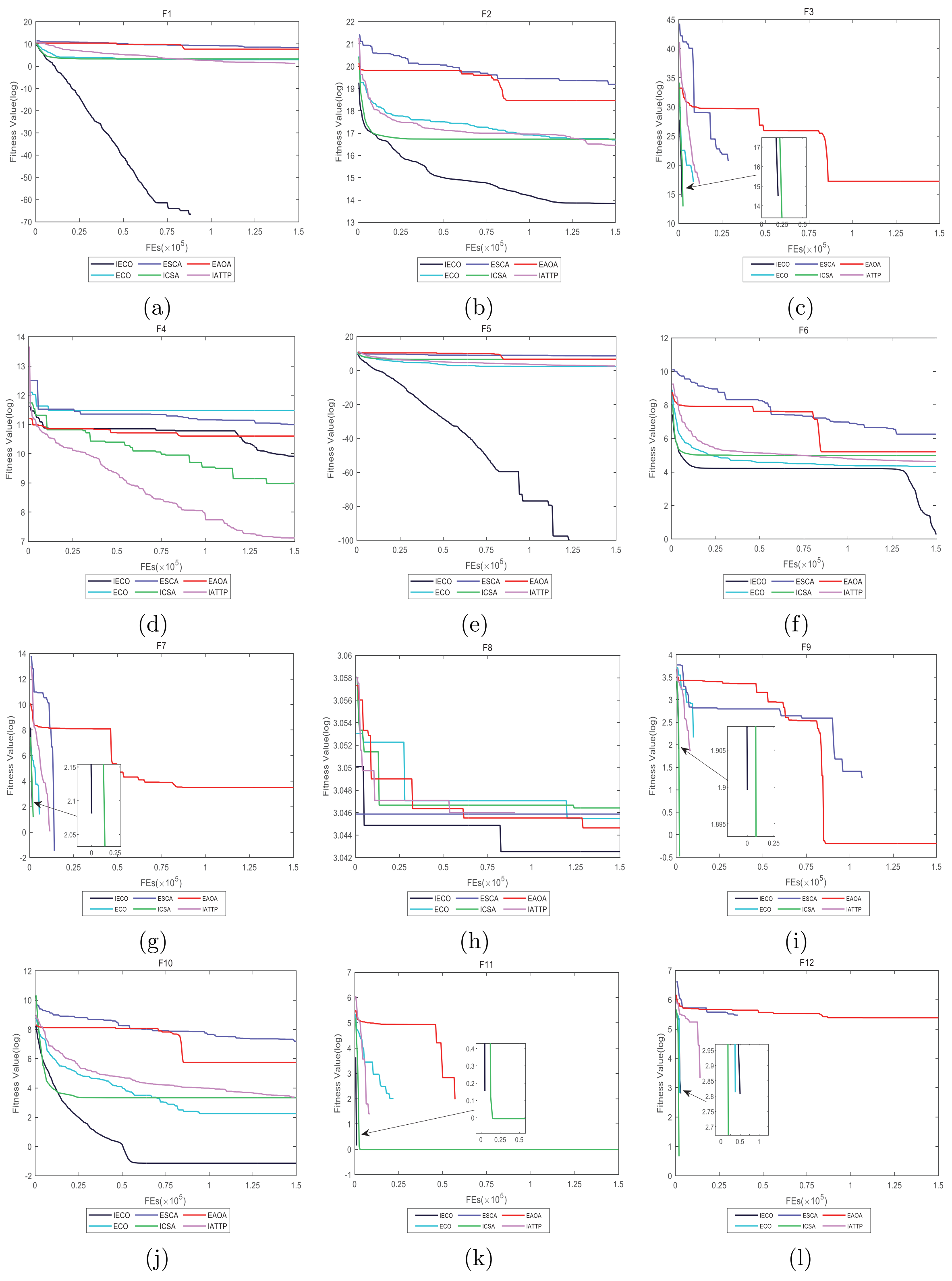
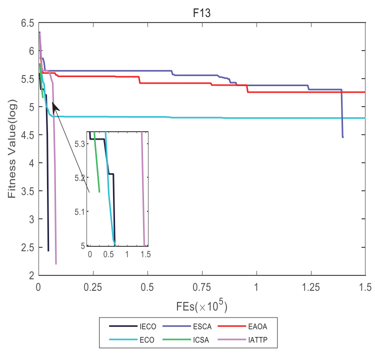
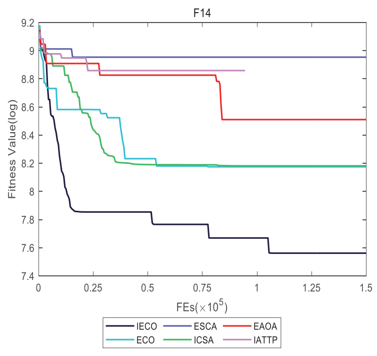


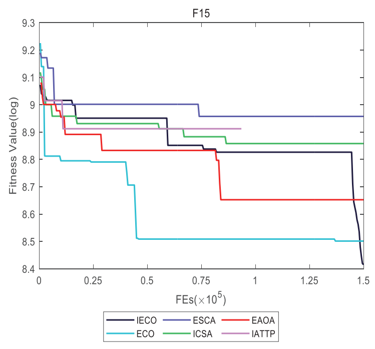
FIGURE 4. Each algorithm's convergence curves on the CEC2013 test suite. The 28 test functions F1F28 in the CEC2013 test set are denoted by (a)–(ab)



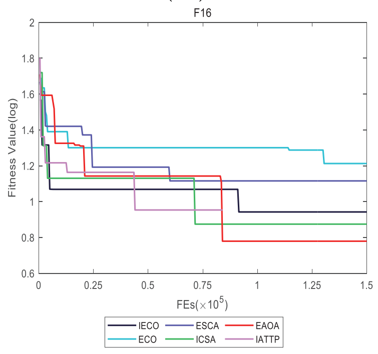
(m)



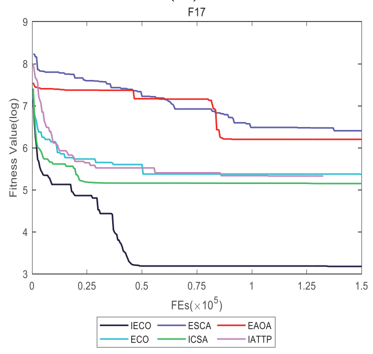
(n)



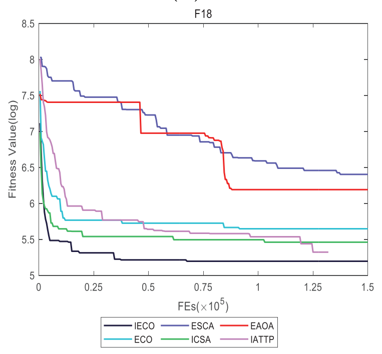
(o)



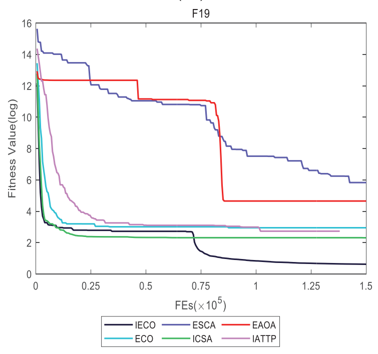
(p)



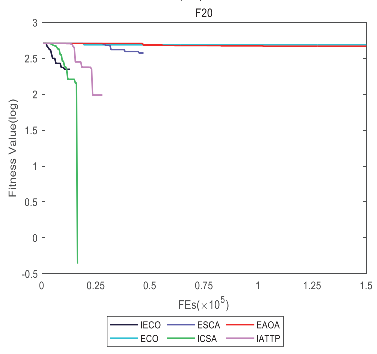
(q)



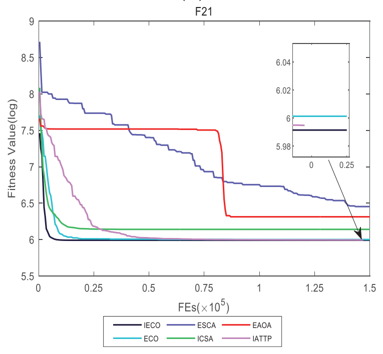
(r)



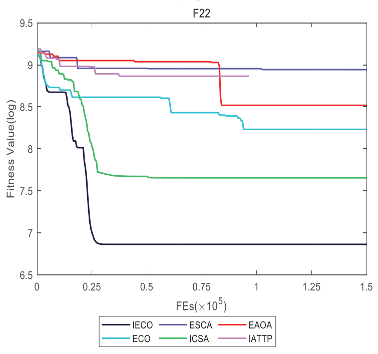
(s)



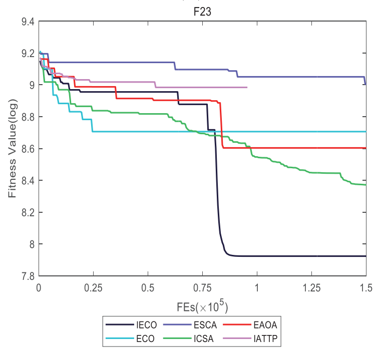
(t)



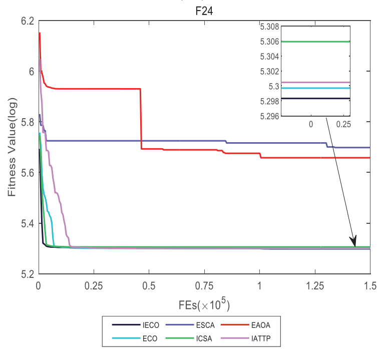
(u)



(v)



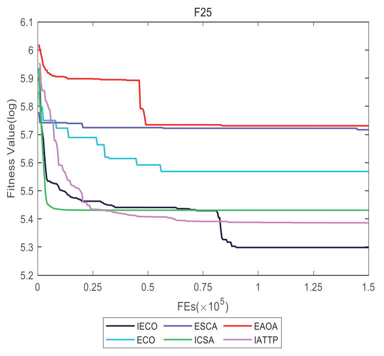
(w)



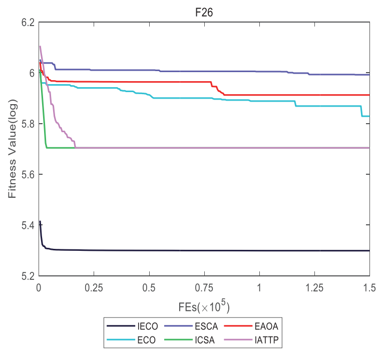
(x)

FIGURE 4. Each algorithm's convergence curves on the CEC2013 test suite. The 28 test functions F1F28 in the CEC2013 test set are denoted by (a)–(ab)

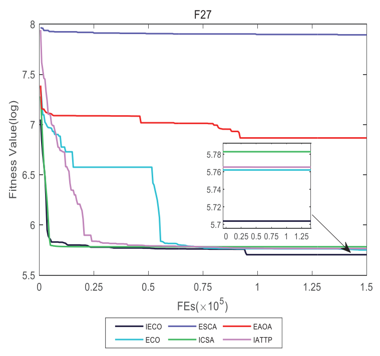
6. **Conclusion.** In this paper, an improved IECO algorithm is proposed to enhance the convergence speed, convergence accuracy, and algorithm stability of the ECO algorithm.



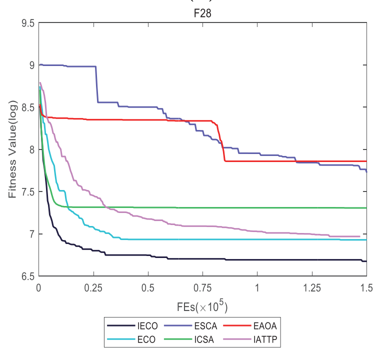
(y)



(z)



(aa)



(bb)

FIGURE 4. Each algorithm's convergence curves on the CEC2013 test suite. The 28 test functions F1F28 in the CEC2013 test set are denoted by (a)–(ab)

The main contributions include the following: The algorithm proposes a population initialization method with an additional autonomous movement strategy to increase population diversity with reference to the autonomous movement habits of natural elephant populations. At the same time, Euclidean distance-based clade division is proposed to avoid the blindness of the original random division method and further balance convergence speed and population diversity. Also inspired by ABC, a premature suppression mechanism is proposed to increase the possibility of the algorithm jumping out of the local optimum. In addition, a novel search strategy for individual family clan members and matriarchs based on autonomous positional movements is proposed to balance convergence speed and population diversity. Then, a new strategy for updating male elephant individuals is proposed, which furthers the algorithm's convergence speed and population diversity. Finally, adult elephants select alternative replacement techniques based on their fitness values, which enhance convergence speed while retaining population variety as well as the algorithm's robustness to cope with varied optimization challenges. In the early stages of evolution, the inferior individuals learn from the superior individuals while retaining some of their information, which maintains population diversity and improves the convergence speed of the algorithm. In later stages of evolution, the simplex approach is employed to improve the worst point in order to raise the algorithm's convergence speed, while new exceptional individuals may be produced to further supply the population's variety. The simulation results of multiple experiments on this algorithm in the CEC2013 test set show that the IECO algorithm has a stronger competitiveness in single-mode and multi-mode problems compared to several other comparative algorithms, but the complexity of this algorithm is slightly high and lacks application to practical engineering problems. In future work, the complexity of the IECO algorithm can be further reduced while applying it to practical engineering problems.

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

REFERENCES

[1] O.-K. Erol, I. Eksin, "A new optimization method: Big Bang- Big Crunch," *Advances in Engineering Software*, vol. 37, no. 2, pp. 106–111, 2006.

[2] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, "GSA: a Gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[3] A. Kaveh, T. Bakhshpoori, "Water Evaporation Optimization: A novel physically inspired optimization algorithm," *Computers and Structures*, vol. 167, no. 15, pp. 69–85, 2016.

[4] Y.-J. Wang, X.-T. Jiang, "An enhanced lightning attachment procedure optimization algorithm," *Algorithms*, vol. 12, no. 7, pp. 134, 2019.

[5] F.-A. Hashim, K. Hussain, E.-H. Houssein, M.-S. Mabrouk, and W.-A. Atabany, "Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, no. 3, pp. 1531–1551, 2021.

[6] M. Azizi, "Atomic orbital search: A novel metaheuristic algorithm," *Applied Mathematical Modelling*, vol. 93, pp. 657–683, 2021.

[7] Q. Zhao, C.-W. Li, "Two-stage multi-swarm particle swarm optimizer for unconstrained and constrained global optimization," *IEEE Access*, vol. 8, pp. 124905–124927, 2020.

[8] T.-J. Liao, K. Socha, M.-M. Oca, T. Stützle, M. Dorigo, "Ant Colony Optimization for Mixed-Variable Optimization Problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 503–518, 2014.

[9] A. Routray, R.-K. Singh, R. Mahanty, "Harmonic Reduction in Hybrid Cascaded Multilevel Inverter Using Modified Grey Wolf Optimization," *IEEE Transactions on Industry Applications*, vol. 56, no. 2, pp. 1827–1838, 2020.

[10] Y.-J. Cui, "Application of the Improved Chaotic Self-Adapting Monkey Algorithm Into Radar Systems of Internet of Things," *IEEE Access*, vol. 6, pp. 54270–54281, 2018.

[11] L. Wang, X. Zhang, X. Zhang, "Antenna Array Design by Artificial Bee Colony Algorithm With Similarity Induced Search Method," *IEEE Transactions on Magnetics*, vol. 55, no. 6, pp. 1–4, 2019.

[12] J. Senthilnath, S. Kulkarni, J.-A. Benediktsson, X.-S. Yang, "A Novel Approach for Multispectral Satellite Image Classification Based on the Bat Algorithm," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 4, pp. 599–603, 2016.

[13] Q. Zhang, L.-J. Liu, "Whale optimization algorithm based on Lamarckian learning for global optimization problems," *IEEE Access*, vol. 7, no. 1, pp. 36642–36666, 2019.

[14] S.-Z. Koohi, N.-A. W.-A. Hamid, M. Othman, and G. Ibragimov, "Raccoon optimization algorithm," *IEEE Access*, vol. 7, pp. 5383–5399, 2019.

[15] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing & Applications*, vol. 31, no. 7, pp. 1995–2014, 2019.

[16] S.-M. Li, H.-L. Chen, M.-J. Wang, "Slime mould algorithm: A new method for stochastic optimization,"*Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.

[17] M. Jafari, E. Salajegheh, J. Salajegheh, "Elephant clan optimization: A nature-inspired metaheuristic algorithm for the optimal design of structures," *Applied Soft Computing*, vol. 113, Part A, December 2021, 107892, 2021.

[18] F.-A. Hashim, A.-G. Hussien, "Snake Optimizer: A novel meta-heuristic optimization algorithm," *Knowledge-Based Systems*, vol. 242, 108320, 2022.

[19] J.-J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 122–128, 1986.

[20] A.-K. Qin, V.-L. Huang, and P.-N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[21] H.-Y. Zhang, "An improved immune algorithm for simple assembly line balancing problem of type 1," *Journal of Algorithms and Computational Technology*, vol. 11, no. 4, pp. 317–326, 2017.

[22] M.-Y. Cheng, D. Prayogo, “Symbiotic organisms search: a new metaheuristic optimization algorithm,” *Computers & Structures*, vol. 139, pp. 98–112, 2014.

[23] Y. Drishti, “Blood Coagulation Algorithm: A Novel Bio-Inspired Meta-Heuristic Algorithm for Global Optimization,” *Mathematics*, vol. 9, no. 23, 3011, 2021.

[24] G. Zhang, Y. Li, Y. Shi, “Distributed learning particle swarm optimizer for global optimization of multimodal problems,” *Frontiers of Computer Science*, vol. 12, no. 1, pp. 122–134, 2018.

[25] X.-W. Xia, L. Gui, G. He, B. Wei, Y . Zhang, F. Yu, H. Wu, and Z.-H. Zhan, “An expanded particle swarm optimization based on multi-exemplar and forgetting ability,” *Information Sciences*, vol. 508, pp. 105–120, 2019.

[26] Y.-J. Wang, J.-R. Han, and Z.-M. Teng, “An improved group teaching optimization algorithm for global function optimization,” *Scientific Reports*, vol. 12, 11267, 2022.

[27] Y.-J. Wang, J.-X. Song, Z.-M. Teng, “An Improved New Caledonian Crow Learning Algorithm for Global Function Optimization,” *Computational Intelligence and Neuroscience*, vol. 2022, 9248771, 2022.

[28] A.-L.-H.-P. Shaik, M.-K. Manoharan, A.-K. Pani, R.-R. Avala, C.-M. Chen, “Gaussian Mutation–Spider Monkey Optimization (GM-SMO) Model for Remote Sensing Scene Classification.” *Remote Sensing*, vol. 14, no. 24, 6279, 2022.

[29] C.-M. Chen, S. Lv, J. Ning, J.-M.-T. Wu, “A Genetic Algorithm for the Waitable Time-Varying Multi-Depot Green Vehicle Routing Problem,” *Symmetry*, vol. 15, no. 1, 124, 2023.

[30] J. Zhou, X. Yao, F.-T.-S. Chan, Y. Lin, H. Jin, L. Gao, and X. Wang, “An individual dependent multi-colony artificial bee colony algorithm,” *Information Sciences*, vol. 485, pp. 114–140, 2019.

[31] T.-Y. Wu, H.-N. Li, S.-C. Chu, “CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps,” *Mathematics*, vol. 11, no. 9, 1977, 2023.

[32] T.-Y. Ye, W.-J. Wang, H. Wang, “Artificial bee colony algorithm with efficient search strategy based on random neighborhood structure,” *Knowledge-Based Systems*, vol. 241, 108306, 2022.

[33] T.-Y. Wu, A.-K. Shao, J.-S. Pan, “CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm,” *Mathematics*, vol. 11, no. 10, 2339, 2023.

[34] R. Cheng, Y. Bai, Y. Zhao, X. Tan, and T. Xu, “Improved fireworks algorithm with information exchange for function optimization,” *Knowledge-Based Systems*, vol. 163, pp. 82–90, 2019.

[35] K.-H. Truong, P. Nallagownden, I. Elamvazuthi, “An improved meta-heuristic method to maximize the penetration of distributed generation in radial distribution networks,” *Neural Computing and Applications*, vol. 32, no. 14, pp. 10159–10181, 2020.

[36] M.-M. Saafan, E.-M. El-Gendy, “IWOSSA: An improved whale optimization salp swarm algorithm for solving optimization problems,” *Expert Systems with Applications*, vol. 176, no. 2, 114901, 2021.

[37] W. Long, M. Xu, J.-J. Jiao, “A velocity-based butterfly optimization algorithm for high-dimensional optimization and feature selection,” *Expert Systems with Applications*, vol. 201, 117217, 2022.

[38] J.-J. Liang, B.-Y. Qu, P.-N. Suganthan, and A.-G. Hernández-Díaz, “Problem definitions an evaluation criteria for the CEC2013 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.

[39] J. Derrac, S. Garcíab, D. Molina, “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.

[40] Z.-K. Feng, J.-F. Duan, W.-J. Niu, Z.-Q. Jiang, Y. Liu, “Enhanced sine cosine algorithm using opposition learning, adaptive evolution and neighborhood search strategies for multivariable parameter optimization problems,” *Applied Soft Computing*, vol. 119, 108562, 2022.

[41] J. Gholami, F. Mardukhi and H. M. Zawbaa, ”An improved crow search algorithm for solving numerical optimization functions,” *Soft Computing*, vol. 25, no.14, pp. 9441–9454, 2021.

[42] A.-S. Desuky, S. Hussain, S. Kausar, Md.-A. Islam, L.-M. El Bakrawy, “EAOA: an enhanced archimedes optimization algorithm for feature selection in classification,” *IEEE Access*, vol. 9, pp. 120795–120814, 2021.

[43] Y.-P. Xiao, H.-B. Chi, and Q.-Q. Li, “An improved artificial tree algorithm with two populations (IATTP),” *Engineering Applications of Artificial Intelligence*, vol. 104, 104324, 2021.