

Research Article

An Improved New Caledonian Crow Learning Algorithm for Global Function Optimization

Yanjiao Wang ¹, Jiaxu Song ¹, and Ziming Teng ²

¹Department of Electrical Engineering, Northeast Electric Power University, Jilin 132012, China

²Department of Communication Engineering, Jilin University, Jilin 130012, China

Correspondence should be addressed to Jiaxu Song; songjiaxu1996@163.com

Received 29 June 2022; Revised 31 July 2022; Accepted 5 August 2022; Published 10 October 2022

Academic Editor: Tongguang Ni

Copyright © 2022 Yanjiao Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The New Caledonian crow learning algorithm (NCCLA) is a novel metaheuristic algorithm inspired by the learning behavior of New Caledonian crows learning to make tools to obtain food. However, it suffers from the problems of easily falling into local optima and insufficient convergence accuracy and convergence precision. To further improve the convergence performance of NCCLA, an improved New Caledonian crow learning algorithm (INCCLA) is proposed in this paper. By determining the parent individuals based on the cosine similarity, the juveniles are guided to search toward different ranges to maintain the population diversity; a novel hybrid mechanism of complete and incomplete learning is proposed to balance the exploration and exploitation capabilities of the algorithm; the update strategy of juveniles and parent individuals is improved to enhance the convergence speed and precision of the algorithm. The test results of the CEC2013 and CEC2020 test suites show that, compared with the original NCCLA algorithm and four of the best metaheuristics to date, INCCLA has significant advantages in terms of convergence speed, convergence precision, and stability.

1. Introduction

A large number of optimization problems exist in real life, engineering design, computer technology and other fields, such as minimum cost, optimal parameters, minimum time, pipeline route design, welded beam design, etc. In order to obtain higher economic efficiency and social value, scholars strive to obtain the optimal solution to optimization problems, thus making the research of optimization methods widely concerned.

Optimization methods usually include traditional optimization methods such as the fastest descent method and metaheuristic methods. Among them, traditional optimization methods usually require the optimization problem to be derivable, and the convergence speed and convergence precision are difficult to meet the practical needs when the optimization problem has multiple extrema. And the metaheuristic algorithms proposed by simulating biological habits in nature, etc., have good exploration and exploitation performance through information exchange among

individuals in the population. Compared with traditional optimization algorithms, metaheuristic algorithms are better in convergence speed, convergence precision, robustness, and stability, with no strict requirements on the form of optimization problems. This makes the metaheuristic algorithm the most effective and widely used optimization method. Researchers have focused on two main aspects of metaheuristics to obtain the best optimization results: improving existing metaheuristics and proposing new ones.

Various metaheuristic algorithms have been proposed, and the more representative methods are as follows: the artificial bee colony algorithm (ABC) was proposed to simulate the behavior of bee colonies to find the optimal nectar source according to different internal divisions of labor [1]; the crow search algorithm (CSA) was proposed to simulate the behavior of crows to hide and search for food [2]; particle swarm optimization (PSO) algorithm was proposed to simulate the foraging behavior of a flock of birds [3]; the firefly algorithm (FA) was proposed to simulate the behavior of fireflies to attract each other [4]; the Marine

Predator Algorithm (MPA) was proposed to simulate the behavior of biological interactions between marine predators and prey [5]; the Manta Ray Foraging Optimization Algorithm (MRFO) was proposed to simulate three unique foraging modes of manta rays: chain foraging, cyclone foraging, and somersault foraging [6]; the dolphin swarm optimization algorithm (DSA) was proposed to simulate the habits of dolphins such as echolocation and information exchange [7]; and the gray wolf optimization algorithm (GWO) was proposed to simulate the hunting behavior of gray wolves [8] and so on. Some of the algorithms mentioned above have been applied to many research articles, such as applying particle swarm optimization to solve the UCP problem with deterministic and stochastic load demands [9], the ocean predator algorithm to solve the optimal reactive power dispatch (ORPD) problem [10], and the manta ray foraging optimization algorithm to solve the economic load dispatch and advance dispatches problems of microgrids [11].

Researchers have done a lot of work on the improvement of existing metaheuristic algorithms, and the more representative research results are given below. Many improvement methods for particle swarm algorithms have been proposed in recent years, and the representative results are as follows: In 2016, Samma et al. proposed the RLMPSO algorithm [12], where each particle performs five operations under the control of the RL algorithm to improve the search performance of the particle swarm algorithm. In 2018, Zhang et al. proposed the DLPSO algorithm [13] to address the shortcomings of the PSO algorithm in multimodal indivisible problems that tend to fall into local optima, which extracts good vectors from the vectors distributed in the search space to form a new vector with a greater possibility of jumping out of local optima. In 2021, Lu et al. proposed the EMCPSO algorithm [14], which takes advantage of the technology of multiple populations in order to overcome the problem of premature convergence of PSO, which divides the population into four identical subpopulations, and the optimal individual in each subpopulation is used to represent the evolutionary state of that subpopulation, and by sharing information among the four populations, the evolutionarily stagnant subpopulations search for the optimal solution again, while introducing an exclusion mechanism to prevent premature convergence of particles further. In 2022, Wang et al. proposed the RLLPSO algorithm for large-scale optimization problems [15], which constructs a level-based population structure to improve population diversity, a reinforcement learning strategy as well as a level competition strategy to improve the search efficiency of the algorithm in order to overcome the complexity of large-scale optimization problems.

Many improvement methods have been proposed for the artificial bee colony algorithm, and the representative results are as follows: In 2018, Cui et al. proposed the DPABC algorithm [16], which uses a dual population framework to divide the population into a convergence population and a diversity population, responsible for developing promising regions as well as maintaining population diversity, respectively, to improve the overall performance of the

algorithm. In 2019, Awadallah et al. improved the onlooker bee stage [17], combining four selection methods, including global optimum, tournament, linear ranking, and exponential ranking, to guide the search process of the onlooker bee in order to determine the impact of the selection scheme on the onlooker bee stage. In 2021, Zhou et al. proposed the ABC-MNT algorithm [18], which applies three different neighborhoods to different individuals, helping the algorithm to achieve a better balance between exploration and exploitation, in addition to employing a global neighborhood search strategy and opposition-based learning that preserves the search experience of the scout bee phase. In 2022, Ye et al. proposed the RNSABC algorithm [19], which uses a random neighborhood structure so that each solution has a random neighborhood, in addition to a depth-first search method to enhance the search capability of the following bee to improve the algorithm's ability to search for the optimal solution.

Representative results of the improvement of other mainstream metaheuristic algorithms are as follows: In 2017, Wang et al. proposed a firefly algorithm with neighborhood attractiveness (NaFA) [20], where each firefly selects attractive individuals from a predefined region instead of the whole population, and the proposed strategy can effectively improve the solution accuracy and reduce the time complexity. In 2018, Sun et al. proposed an improved whale optimization algorithm (MWOA) for solving large-scale optimization problems [21], which uses a nonlinear dynamic strategy based on the cosine function to update the control parameters, balances the exploration and exploitation capabilities of the algorithm, uses a Levy flight strategy to make the algorithm jump out of the local optimum, and uses quadratic interpolation for the optimal individuals of the population to enhance the local exploitation capabilities of the algorithm. In 2019, Zamani et al. proposed a conscious neighborhood-based crow search algorithm (CCSA) [22], which introduces three search strategies, neighborhood-based local search strategy, non-neighborhood global search strategy, and roaming-based search strategy, to enhance the balance between local and global search. In 2020, Gupta et al. proposed a memory-based gray wolf optimization algorithm (mGWO) [23], which modified crossover and greedy selection based on the historical optimum of individuals, enhancing the algorithm's ability to perform the global search, local exploitation, and the balance between the two. In 2022, Long et al. proposed a velocity-based butterfly optimization algorithm (VBOA) [24], which introduced velocity and memory to guide individuals in the local search phase and introduced a refraction-based learning strategy, effectively enhancing the diversity of populations and the exploration ability of the algorithm.

Many excellent metaheuristic algorithms have been proposed in recent years. In 2018, Wang proposed the moth search algorithm (MSA) [25] inspired by the phototropism of moths and Levy flight, which treats moths as individuals. Moths with smaller distances from the optimal individual perform Levy flight, while moths with more considerable distances approach the optimal individual in a straight line. The above two stages optimize the algorithm. In 2019,

Heidari et al. proposed the Harris Hawk optimization algorithm (HHO) [26] based on the inspiration of the collaborative group behavior of Harris hawks during predation, which uses Harris hawks as individuals, and the search process includes three stages: exploration, exploration to exploitation conversion, and exploitation, and the algorithm is characterized by few control parameters and excellent global search capability. In 2020, Li et al. proposed the slime mold algorithm (SMA) [27], inspired by the behavioral and morphological changes in *Physarum polycephalum* during foraging, which creates three different forms to optimize the problem by using weights to simulate the positive and negative feedback generated by slime molds during foraging. In the same year, Al-Sorori and Mohsen proposed the New Caledonian crow learning algorithm (NCCLA) [28] based on the behavior of New Caledonian crows to obtain food by learning to make tools. The advantage of this algorithm is its stochastic nature, which guarantees that the algorithm does not get trapped at the local optimum. In the same year, Mohamed et al. proposed a gaining-sharing knowledge-based algorithm (GSK) [29] inspired by the process of acquiring and sharing knowledge in the human life cycle, which treats people as individuals and improves their knowledge by using junior gaining and sharing phase and senior gaining and sharing phase, i.e., solving optimization problems on continuous space. In 2021, Tu et al. proposed the colony predation algorithm (CPA) [30], inspired by the supportive behavior of herd animals and the behavior of selective hunting, which is based on the coexistence of social animals and focuses on optimizing the problem through five stages: communicating and collaborating, dispersing food, surrounding food, supporting the closest individual, and finding food. In 2022, Hashim et al. proposed the snake optimizer (SO) [31], based on the behavior of snakes to forage or breed under different temperature and food availability conditions, in which individuals explore and exploit the conditions of temperature as well as food.

The various metaheuristic algorithms mentioned above provide new ideas for solving optimization problems and further advance the development of optimization techniques. Compared with the more classical PSO and DE, they have significantly improved in terms of convergence speed and convergence precision. However, unfortunately, for the highly nonlinear and complex optimization problems that emerge one after another in practical engineering, the convergence speed and convergence precision of the existing metaheuristic algorithms are obviously insufficient and even fall into local optimum, making it difficult to obtain highly satisfactory economic and social values. Therefore, improving the optimization performance of each new metaheuristic algorithm has been one of the main research contents in the field of evolution.

In this context, given the literature [28] and a large number of experimental studies, the NCCLA algorithm is a very excellent metaheuristic algorithm because of its simple operation and significantly better convergence capability than optimization algorithms such as GWO, CSA, and WOA, and is highly promising in fields such as engineering optimization. In this paper, we only study NCCLA. In order

to further improve the problems of insufficient convergence precision and convergence speed and easy falling into local optimum when NCCLA deals with very complex optimization problems, we propose an improved New Caledonian Crow Learning Algorithm (INCCLA) in this paper.

The main innovations and contributions of INCCLA are as follows: (1) A cosine similarity-based parent individual selection approach is proposed. The globally optimal individual and another excellent individual with a significant difference in similarity are selected as the parent, and the juvenile crow individuals are guided to search toward different ranges to maintain the population diversity while maintaining the convergence speed of the algorithm. (2) Improving the learning phase of juvenile crows. A new hybrid learning mechanism of complete learning and incomplete learning is set up, in which the individual juvenile crows in the complete learning stage can select learning objects according to their own conditions, which effectively improves the convergence speed of the algorithm while maintaining the population diversity to a certain extent; while the juvenile crows in the incomplete learning stage learn the behavioral attributes of different individuals in order to maintain the population diversity of the algorithm. (3) Improving the reinforcement phase. For the juvenile reinforcement stage, a weighting factor is introduced to enable the algorithm to have a strong exploration ability in the early evolutionary stage and a strong exploitation ability in the late evolutionary stage, and at the same time, a small range of random perturbations is added to increase the possibility of convergence of the algorithm to the global optimum; for the parents' reinforcement stage, the update methods of the two parents' individuals are improved further to respectively balance the exploration and exploitation ability of the algorithm. The results of testing on the CEC2013 and CEC2020 test suites show that the INCCLA proposed in this paper has significant advantages in terms of convergence speed, convergence precision, and stability compared with four other more representative optimization algorithms.

The rest of the paper is organized as follows: Section 2 describes the working principle and flow of the NCCLA algorithm. Section 3 analyzes the defects of the original NCCLA algorithm and further proposes an improved INCCLA algorithm. Section 4 shows the simulation results and analysis of the INCCLA algorithm with the original NCCLA algorithm and other more mainstream improved algorithms on the CEC2013 and CEC2020 test function suites. Section 5 concludes the proposed algorithm in this paper.

2. New Caledonian Crow Learning Algorithm

In nature, New Caledonian crows are divided into the juvenile and the parent crows, which enhance their tool-design skills through learning and their own experience and knowledge, respectively, to obtain food from the pandanus tree. Inspired by the above behavior, Wedad and Abdulqader proposed the New Caledonian crow learning algorithm (NCCLA). In NCCLA, individuals represent the

manufacturing behavior of New Caledonian crows and fitness values represent the behavioral advantage of each crow. The algorithm has three main phases: initialization, learning phase, and reinforcement phase. The pseudo-code of NCCLA is shown in Algorithm 1, and the key steps are briefly described as follows.

2.1. Population Initialization. Suppose the number of individuals in population X is N . Each individual X_i ($i = 1, 2, \dots, N$) represents the behavior of a crow, which can be expressed as $X_i = [X_{i,1}, X_{i,2}, X_{i,3}, \dots, X_{i,D}]$, where D is the dimension of the optimization problem, and $X_{i,j}$ denotes the j -th behavioral attribute of the i -th crow. At the beginning of the algorithm, the initial behavior of each crow is generated randomly according to Equation.

$$X_{i,j}(0) = X_L + U(0, 1) \times (X_U - X_L), \quad (1)$$

where X_U and X_L correspond to the upper and lower bounds of the j -th dimensional search space in the optimization problem, respectively, and $U(0,1)$ is a uniformly distributed random number in the range $[0,1]$.

2.2. Learning Phase. In NCCLA, only juveniles enter the learning phase, and each behavioral attribute $X_{i,j}$ of juveniles will be socially or asocially learned according to the probability SL_{prob} or $1 - SL_{\text{prob}}$, respectively. SL_{prob} is recommended to be set to 0.95, but can be set to other values.

2.2.1. Social Learning. After the j -th behavioral attribute of juvenile crow i , $X_{i,j}$, is determined to require social learning according to the probability SL_{prob} , it is then decided to perform vertical learning or horizontal learning according to the predetermined probability VSL_{prob} or $1 - VSL_{\text{prob}}$. The details are shown in Equation.

$$X_{i,j}(t) = \begin{cases} X_{k_1,j}(t-1), & \text{if } \text{rand} \leq VSL_{\text{prob}}, \\ X_{k_2,j}(t-1), & \text{else,} \end{cases} \quad (2)$$

where $X_{i,j}(t)$ is the j -th new behavioral attribute acquired by juvenile crow i after social learning in iteration t , VSL_{prob} is recommended to be set to 0.99, and can also be set to other values.

In (2), when $\text{rand} \leq VSL_{\text{prob}}$, the juvenile crow X_i performs vertical learning to its parent, i.e., it copies the corresponding behavioral attributes of its parent X_{p1} or X_{p2} with probability $P1_{\text{prob}}$, obviously $k_1 = 1$ or 2 ; Otherwise, the juvenile crow X_i performs horizontal learning, i.e., it randomly selects a sibling k_2 that is more experienced and copies its corresponding behavioral attributes, and the expression formula for k_2 is shown in equation (3). It should be noted that for the juvenile crow with the best fitness value, only vertical learning is performed, not horizontal learning.

$$k_2 = 3 + [\text{rand} \times (i - 3)], \quad (3)$$

where rand is a random number uniformly distributed in the range $[0, 1]$, $[\cdot]$ means rounding is performed.

2.2.2. Asocial Learning. When a crow X_i performs asocial learning, its behavioral attributes are randomly updated using (1) according to the probability TaE_{prob} , or retained the previous behavioral attributes according to $1 - TaE_{\text{prob}}$. This is shown in Equation .

$$X_{i,j}(t) = \begin{cases} X_L + U(0, 1) \times (X_U - X_L), & \text{rand} \leq TaE_{\text{prob}}, \\ X_{i,j}(t-1), & \text{else,} \end{cases} \quad (4)$$

where TaE_{prob} is recommended to be set to 0.99, but can also be set to other values.

2.3. Reinforcement Phase. After completion of the learning phase, certain attributes of the learned juvenile crow behavior and parent behavior are reinforced according to the reinforcement probability RP_{prob} . RP_{prob} is recommended to be set to 0.99, and can be set to other values.

2.3.1. Juvenile Reinforcement. Each behavioral attribute of juvenile crows was reinforced according to Equation.

$$X_{i,j}(t) = X_{i,j}(t) \pm RW, \quad (5)$$

where RW is shown in Equation.

$$RW = \begin{cases} \beta - \alpha, & i < N, \\ r1 \times ((r2 \times \beta) - \alpha), & \text{otherwise,} \end{cases} \quad (6)$$

where $r1$ and $r2$ are random numbers between 0 and 1, α represents the difference between their behavioral attributes before and after learning, as shown in (7), and β represents the social learning effect developed over time, as shown in Equation (8).

$$\alpha = |X_{i,j}(t) - X_{i,j}(t-1)|, \quad (7)$$

$$\beta = X_{i,j}(t-1) \times e^{-lf \times r \times t \times \text{mean}(j)}, \quad (8)$$

where t represents the number of current iterations, $X_{i,j}(t-1)$ is the j -th behavioral attribute of crow i before the current generation of learning, r is a normally distributed random number in the range $[0,1]$, $\text{mean}(j)$ is the average of the j -th behavioral attribute of all individuals in the population, and lf is a learning factor, as shown in Equation .

$$lf = lf_{\min} + \left(\frac{lf_{\max} - lf_{\min}}{\max_t} \right) \times t, \quad (9)$$

where \max_t represents the maximum number of iterations, lf_{\max} and lf_{\min} represent the maximum and minimum values of the learning factor, respectively, and are recommended to be set to 0.02 and 0.0005, respectively, or can be set by oneself.

2.3.2. Parents Reinforcement. The parents X_{p1} and X_{p2} update certain attributes according to the reinforcement probability RP_{prob} . When $\text{rand} \leq RP_{\text{prob}}$, the j -th dimensional behavioral attribute of crow i is reinforced, as shown in (10); otherwise, it retains the original behavioral attribute.

```

Input:  $N, D, RP_{\text{prob}}, SL_{\text{prob}}, VSL_{\text{prob}}, P1_{\text{prob}}, TaE_{\text{prob}}, \text{MaxIter}$ 
Output: best solution and its fitness
(1) Initialization of variables ( $N, D, RP_{\text{prob}}, SL_{\text{prob}}, VSL_{\text{prob}}, P1_{\text{prob}}, TaE_{\text{prob}}, \text{MaxIter}$ )
(2) Generate the initial population  $X$  according to Section 2.1
(3) Calculate the fitness value  $F_i$  of each individual  $X_i$ 
(4) While  $t \leq \text{MaxIter}$  do
(5) Rank all individuals and select the two best individuals,  $X_1$  and  $X_2$ , as parents, noted as  $X_{p1}$  and  $X_{p2}$ , and the rest as juvenile crows
(6) For Each juvenile crow  $X_i$  in population  $X$  do
(7)   For Each behavior attribute  $j$  in  $X_i$  do
      //Learning phase
(8)     If  $\text{rand} \leq SL_{\text{prob}}$  then
(9)       Decide to perform vertical or horizontal learning according to probability  $VSL_{\text{prob}}$  in Section 2.2.1//Social learning
(10)    else
(11)      Random update of behavior attributes or retention of previous behavior attributes using (1) as decided in Section 2.2.2 based on the probability  $TaE_{\text{prob}}$ //Asocial learning
(12)    end If
(13)    Based on the probability  $RP_{\text{prob}}$ , reinforce the attributes of  $X_i$  after learning using (4)//Juvenile reinforcement
(14)  end For
(15) end For
(16) Based on the probability  $RP_{\text{prob}}$ , reinforce the attributes of the parents  $X_{p1}$  and  $X_{p2}$  using (9)//Parents reinforcement
(17)  $t = t + 1$ 
(18) end While
(19) Output the global optimal solution

```

ALGORITHM 1: NCCLA.

$$X_{i,j}(t) = \begin{cases} X_{i,j}(t-1) - \left(X_{1,j}(t-1) + e^{r1 \times (\text{mean}(j) - X_{i,j}(t-1))} \right), & i = p1, \\ X_{i,j}(t-1) - \left(r2 \times \left(X_{1,j}(t-1) - e^{r1 \times (\text{mean}(j) - X_{i,j}(t-1))} \right) \right), & i = p2, \end{cases} \quad (10)$$

where $r1$ is a normally distributed random number, $r2$ is a uniformly distributed random number in the range $[0, 1]$, and $\text{mean}(j)$ is the mean of the j -th behavioral attribute of all individuals in the current population.

3. Proposed Algorithm

To further improve the convergence performance of NCCLA, this section proposes an improved New Caledonian Crow Learning Algorithm (INCCLA), whose pseudo-code is shown in Algorithm 2.

3.1. Determination of Parent Individuals Based on Cosine Similarity. In NCCLA, VSL_{prob} and $P1_{\text{prob}}$ are set to 0.99 and 0.95, respectively, meaning that each juvenile crow will perform vertical learning with a probability of 0.99×0.95 toward the parent individual during the learning phase. The parent individuals are always the two individuals with the best fitness in the population. As evolution proceeds, the two-parent individuals will rapidly approach each other, showing a high degree of similarity, which will lead most of

the juvenile crows to approach them through vertical learning rapidly, and can only search around the parent individuals, lacking exploration of other ranges. Although rapid convergence can be achieved in the early stage of evolution, the loss of population diversity is apparent, and the algorithm is straightforward to fall into the local optimum.

In order to solve the above problem, the following cosine similarity-based parent individual selection method is proposed in this section. First, the best individual in the population is determined as the parent individual X_{p1} ; then, the cosine similarity of the remaining individuals in the population to X_{p1} is calculated according to (11), and they are arranged in order from smallest to largest and evenly divided into two groups; finally, the individual with the best fitness value is selected as the parent individual X_{p2} from the other group different from the group in which X_{p1} is located. It is important to note that the parent individuals are selected for every P generation above. Generally, $P=50$ is sufficient to achieve good results, but it can also be set according to the optimization problem.

```

Input:  $N, D, RP_{\text{prob}}, SL_{\text{prob}}, \text{MaxIter}$ 
Output: best solution and its fitness
(1) Initialization of variables ( $N, D, RP_{\text{prob}}, SL_{\text{prob}}, \text{MaxIter}$ )
(2) Generate the initial population  $X$  according to Section 2.1
(3) Calculate the fitness value  $F_i$  of each individual  $X_i$ 
(4) While  $t \leq \text{MaxIter}$  do
(5) Rank all individuals and select the parents according to Section 3.1, noted as  $X_{p1}$  and  $X_{p2}$ , and the rest as juvenile crows
(6) Randomly select  $R$  individual juvenile crows to form a subpop
(7) For Each juvenile crow  $X_i$  in population  $X$  do
(8)   if  $X_i \in \text{subpop}$  do
(9)     Perform complete learning of the juvenile crow  $X_i$  according to Section 3.2.1
(10)   else
(11)     Perform incomplete learning of the juvenile crow  $X_i$  according to Section 3.2.2
(12)   end if
(13)   operations on juvenile crows  $X_i$  according to Section 3.3.1
(14) end for
(15) Perform parent reinforcement operations on parents  $X_{p1}$  and  $X_{p2}$  according to Section 3.3.2
(16)  $t = t + 1$ 
(17) end While
(18) Output the global optimal solution

```

ALGORITHM 2: INCCLA.

$$\text{sim}_{i,p1}(t-1) = \frac{\sum_{j=1}^D X_{p1,j}(t-1) \times X_{i,j}(t-1)}{\sqrt{\sum_{j=1}^D (X_{p1,j}(t-1))^2} \times \sqrt{\sum_{j=1}^D (X_{i,j}(t-1))^2}}, \quad i = 1, 2, \dots, N, \quad (11)$$

where $X_{p1,j}(t-1)$ denotes the j -th dimension of the globally optimal individual $X_{p1}(t-1)$ determined by relying on the previous iteration of the population, and $\text{sim}_{i,p1}(t-1)$ denotes the cosine similarity of the i -th individual $X_i(t-1)$ without any evolutionary operation to $X_{p1}(t-1)$ in the current generation.

To further illustrate the above method of determining parent individuals based on cosine similarity, taking the optimized 2-dimensional Rotated High Conditioned Elliptic function as an example, the specific determination process is given in Figure 1. It can be seen that, according to the original parent individual determination method, X_1 and X_2 , which are close to each other, will be selected as parent individuals, while according to the proposed method in this section, X_1 and X_4 , which are farther apart, will be selected as parent individuals. The fitness value of X_4 is not much worse than X_2 .

In summary, compared with the original approach of selecting two more similar optimal individuals as parents, this section retains the optimal individual of the population as a parent individual, which does not affect the convergence speed too much. In contrast, the other parent individual is selected to be less similar to the optimal individual of the population but with better fitness value. This way can guide juveniles in their search across a range and ensure that they learn from more experienced individuals, maintaining good population diversity while not reducing the convergence speed too much.

3.2. Improving Juvenile Crow Learning Phase. As seen in Algorithm 1, the learning phase generates its own learning objects for juvenile crows intending to provide excellent evolutionary directions for the reinforcement phase. In-depth analysis can be found that the various behavioral attributes of the learning objects generated in the learning phase are not almost wholly derived from the same individual. Although the population diversity can be better maintained, it is difficult to ensure the superiority of the learning objects composed of them due to the complete separation of each behavioral attribute. Therefore, it is difficult to guarantee the convergence speed of the reinforcement phase. The excellent individuals themselves are already better integrated with each behavioral attribute, which has a vital role in the rapid convergence of the algorithm but is not conducive to the maintenance of population diversity. Given this, this section proposes a novel juvenile learning approach as shown in Figure 2. R juveniles are randomly selected to perform complete learning, i.e., each behavioral attribute of the corresponding learning object originates from the same individual completely, which promotes fast convergence of the algorithm; while the remaining juveniles perform incomplete learning, i.e., each behavioral attribute of the learning object originates from different individuals, which ensures population diversity.

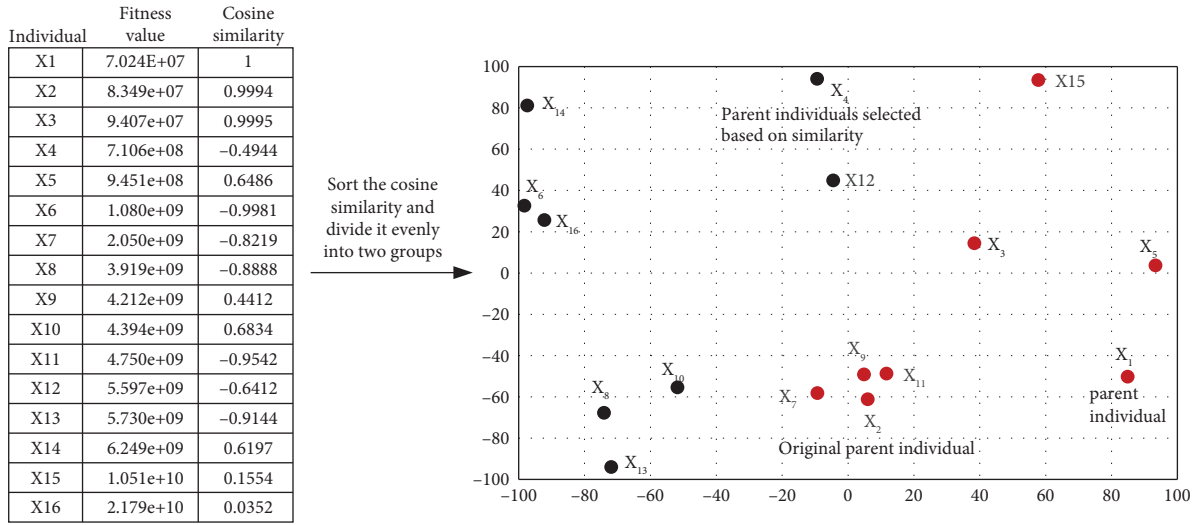


FIGURE 1: Schematic of the process of determining parent individuals based on cosine similarity.

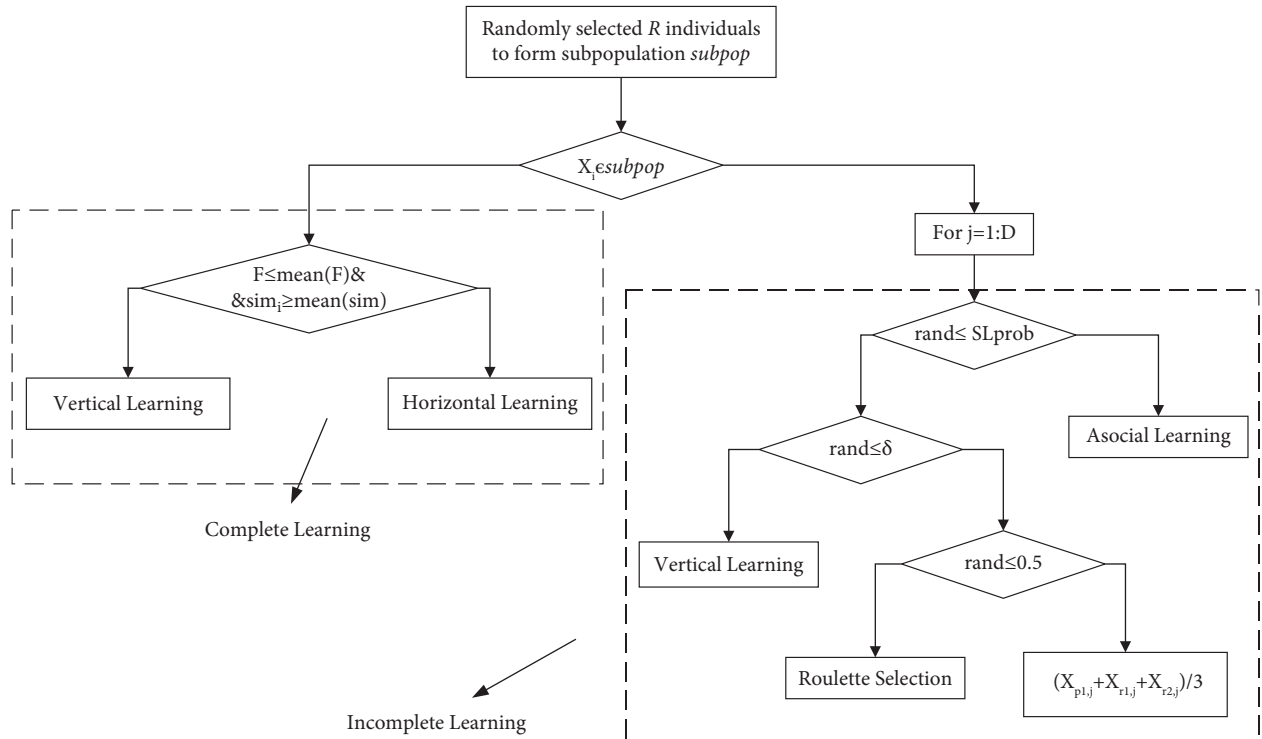


FIGURE 2: Schematic of juvenile crow learning phase.

3.2.1. Complete Learning. As mentioned above, the complete learning phase aims to enhance the convergence speed of the algorithm by copying all the behavioral attributes of a particular outstanding individual. In contrast to social learning, asocial learning focuses on maintaining population diversity; therefore, in the complete learning phase, asocial learning is eliminated, and only social learning is used.

Social learning in NCCLA includes vertical and horizontal learning, where vertical learning is learning from the two best parent individuals in the population, while horizontal learning is learning from other individuals who are

better than oneself. Obviously, compared with vertical learning, horizontal learning is more capable of maintaining population diversity. In NCCLA, vertical or horizontal learning is chosen according to the fixed probability VSL_{prob} . If VSL_{prob} is high, the population will quickly approach the parent individuals, accelerating the algorithm's convergence. However, the rapid loss of population diversity can easily cause the algorithm to fall into local optimum. If VSL_{prob} is low, each individual may have different learning objects, and introducing multiple learning objects makes the evolutionary direction of each individual more diffuse, which is

not conducive to the rapid convergence of the population. The NCCLA recommends that VSL_{prob} be set to 0.95, allowing all juveniles to primarily use vertical learning as their social learning method. In vertical learning, individuals need to choose to learn from the best parent X_{p1} or the second-best parent X_{p2} according to the fixed probability SL_{prob} . If SL_{prob} is large or small, all juvenile crows will search around a parent almost exclusively, which is not conducive to the maintenance of population diversity and increases the possibility of the algorithm falling into local optimum; If SL_{prob} is set to about 0.5, it will learn from the parent

individuals with equal chances, which is similar to random selection and has certain blindness, slowing down the convergence of the algorithm to some extent. In short, the selection methods of vertical learning or social learning by fixed probability and the selection of a certain parent individual for learning by fixed probability in vertical learning are not very reasonable.

Given this, to effectively improve the convergence speed of the algorithm and not destroy the population diversity too much, we propose the complete social learning method as shown in Equation.

$$X_i(t) = \begin{cases} X_{k_1}(t-1), & \text{if } F_i(t-1) \leq \bar{F}_i(t-1) \& \text{sim}_{i,p1}(t-1) \geq \overline{\text{sim}_{i,p1}}(t-1) \\ X_{k_2}(t-1), & \text{else} \end{cases}, \quad (12)$$

where $F_i(t-1)$ denotes the fitness value of juvenile X_i , $\bar{F}_i(t-1)$ denotes the mean of the fitness values of all individuals in the population, $\text{sim}_{i,p1}(t-1)$ denotes the cosine similarity of juvenile X_i to the best individual X_{p1} in the population. $\overline{\text{sim}_{i,p1}}(t-1)$ denotes the mean of the cosine similarity of all individuals in the population to X_{p1} . When both $F_i(t-1) \leq \bar{F}_i(t-1)$ and $\text{sim}_{i,p1}(t-1) \geq \overline{\text{sim}_{i,p1}}(t-1)$ are satisfied, the juvenile X_i performs vertical learning from the parent X_{p1} or X_{p2} according to (13); otherwise, the juvenile X_i will perform horizontal learning in the original way of NCCLA, i.e., it randomly selects a juvenile with a better adaptation value than itself as the learning object.

$$X_i(t-1) = \begin{cases} X_{p1}(t-1), & \text{if } \frac{e^{-\text{sim}_{i,p1}(t-1)}}{1 + e^{-5 \times NF_{p1}}} \geq \frac{e^{-\text{sim}_{i,p2}(t-1)}}{1 + e^{-5 \times NF_{p2}}}, \\ X_{p2}(t-1), & \text{else,} \end{cases} \quad (13)$$

where $\text{sim}_{i,p1}$ and $\text{sim}_{i,p2}$ denote the cosine similarity of juvenile X_i to the parent individuals X_{p1} and X_{p2} , respectively, and NF_i denotes the normalization ability of juvenile X_i as shown in equation (14). The better the individual fitness value, the stronger its normalization ability.

$$NF_i = \begin{cases} \left| \frac{\max(F) - F_i}{\sum_{i=1}^N F_i} \right|, & \text{if } \sum_{i=1}^N F_i \neq 0, \\ \frac{1}{N}, & \text{else.} \end{cases} \quad (14)$$

In summary, this section proposes the complete social learning approach with the following advantages. First, compared with the original fixed-probability selection of vertical or horizontal learning, the new selection approach proposed in this section, as shown in (12), can rely on individuals' conditions to adaptively select the learning mode. Only juvenile individuals with convergence potential, i.e., those who are more similar to the optimal individuals and have better fitness values, will perform vertical learning. In

contrast, the rest of the individuals will perform horizontal learning. Obviously, this approach makes a small number of dominant individuals, who are not far from the optimal parent and are more excellent, focus on mining in the region with more convergence prospects and then quickly determine the more excellent evolutionary direction, driving the rapid convergence of the rest individuals. In turn, most of the remaining individuals learn from other better juvenile crows, which can develop other search areas, facilitating the maintenance of population diversity and reducing the risk of the algorithm falling into a local optimum. Second, the new vertical learning approach as shown in (13) proposed in this section, which relies on a comprehensive judgment of individual fitness values and the degree of similarity with individuals of the two parents, selects individuals learning from the parent with less similarity and excellent performance, and enables the juvenile crows to explore different promising areas as much as possible, which can better maintain the population diversity while ensuring the convergence speed. In short, the complete learning approach proposed in this section selects the learning mode and learning objects in a targeted way according to the juvenile crows' conditions, which effectively improves the convergence speed of the algorithm and maintains the population diversity to a certain extent. In addition, the above process no longer uses the parameters VSL_{prob} and SL_{prob} , which avoids the trouble of parameter debugging.

3.2.2. Incomplete Learning. As mentioned above, the incomplete learning phase will generate new learning objects by copying and absorbing the behavioral attributes of different individuals. To further ensure that the algorithm maintains population diversity and avoids falling into local optimum while improving the convergence speed, this section improves the asocial learning behavior and social learning behavior in the original NCCLA separately and proposes a new incomplete learning approach. The details are as follows.

(1) Improvement of social learning behavior

This section improves the conditions of vertical and horizontal learning in the social learning approach in

NCCLA, as well as the horizontal learning approach, and proposes a new incomplete social learning approach as follows. First, the adaptive selection factor $\delta_i(t)$ is calculated for the i -th juvenile crow individual X_i as shown in (15); then, a random number rand between $[0,1]$, and if $\text{rand} \leq \delta_i(t)$ is generated, vertical learning is performed, i.e., the behavioral attributes corresponding to the j -th dimension of the parent individual X_{p1} or X_{p2} as shown in equation (13) are copied, otherwise, horizontal learning is performed, i.e., the behavioral attributes of the j -th dimension of the individual shown in equation (16) are copied.

$$\delta_i(t) = 1 - e^{-|F_i(t-1) - F_{gbest}(t-1)|}, \quad (15)$$

where $F_{gbest}(t-1)$ and $F_i(t-1)$ represent the fitness values of the globally optimal individual and the juvenile X_i before the current generation's juvenile crow learning operation, respectively.

$$X_{i,j}(t) = \begin{cases} X_{s,j}(t-1), & \text{if } \text{rand} \leq 0.5, \\ \frac{X_{p1,j}(t-1) + X_{r1,j}(t-1) + X_{r2,j}(t-1)}{3}, & \text{else,} \end{cases} \quad (16)$$

where X_{p1} is the optimal parent individual, X_{r1} is a randomly selected individual with a better fitness value than the juvenile crow X_i , X_{r2} is a randomly selected individual in the population, and X_s is the individual selected among all juvenile crow individuals using roulette selection according to the probability corresponding to equation (17).

$$P_k(t) = \frac{e^{-\text{sim}_{i,k}(t-1)}}{1 + e^{-5 \times \text{NF}_k}}, \quad (17)$$

where $\text{sim}_{i,k}$ denotes the cosine similarity between the juvenile crow X_i and the juvenile crow X_k before learning was performed, and NF_k denotes the normalization ability of the juvenile crow X_k .

(2) Improvement of asocial learning behavior

In NCCLA, the probability of asocial learning behavior occurring is only $1 - \text{VSL}_{\text{prob}} \times \text{PI}_{\text{prob}}$, which is only $1 - 0.99 \times 0.95$ by its proposed parameter setting. In essence, such a small probability of asocial learning behavior occurring is intended to provide new evolutionary genes and reduce the possibility of the algorithm falling into a local optimum. Obviously, there is no need to keep certain properties unchanged with a certain probability. For this reason, the asocial learning in the incomplete learning phase only randomly updates each behavioral attribute according to equation (1).

The analysis shows that the incomplete learning approach proposed in this section has the following advantages. First, each behavioral attribute of new juvenile individuals after incomplete learning may

originate from different individuals, including parent individuals X_{p1} and X_{p2} , any other individual in the population, and new genes generated by asocial learning, forming various combinations that can ensure excellent population diversity. Second, the incomplete learning phase mainly uses incomplete social learning and rarely uses asocial learning. Compared with the original social learning approach, the new social incomplete learning proposed in this section better adapts to the needs of the algorithm performance at different stages of the algorithm, as follows: at the early stage of evolution, the individuals in the population are more distributed, with better population diversity, and the difference between individuals and optimal individuals is large. According to the adaptive selection factor, most of the dimensions of the new individuals come from the parent individuals, and a few dimensions come from other juvenile individuals, which further promotes the rapid convergence of the population; while in the late evolutionary stage, all individuals gradually approach the globally optimal individuals, the population shows a certain aggregation, and the population diversity decreases, at this time, most of the dimensions of the new individuals are derived from other more experienced juvenile individuals, which maintains the population diversity without slowing down the convergence of the algorithm. Third, when individuals perform horizontal learning in a certain dimension, they can no longer learn and communicate only with other juveniles who are better than themselves but have the opportunity to exchange information with any individual in the population, which further enhances population diversity, and learning from individuals who are less similar to and better than themselves based on roulette selection does not reduce the convergence speed while maintaining population diversity. In summary, the incomplete learning approach proposed in this section can indeed achieve the goal of maintaining population diversity and maximizing convergence speed.

3.3. Improvement in Reinforcement Phase

3.3.1. Improvements of Juvenile Crows Reinforcement Phase.

In fact, the juvenile crow reinforcement phase in NCCLA is an offset search of RW near the learning objectives identified in the juvenile crow learning phase. The offset range RW is a combination of α and β , where α represents self-perception and β represents social perception, which aims to enhance the exchange of evolutionary information with other individuals and thus explore more search space. The experimental study shows that the unreasonable settings of RW and β make the quality of the reinforced individuals still have much room for improvement, for which they are improved separately.

(1) Improvement of β calculation method

An in-depth analysis of (8) reveals that β is actually a reference to the information of other individuals and scales the individual itself by $\exp(-\text{lf} \times r \times t \times \text{mean}(j))$ times. As the number of iterations t increases, if

mean $(j) < 0$, $\exp(-lf \times r \times t \times \text{mean}(j))$ will be huge, even tending to infinity. The corresponding β and RW are also extremely large, making it extremely easy for the reinforced individuals to exceed the search range, resulting in ineffective reinforcement and making it difficult to provide new individuals with more excellence, leading to slow convergence of the algorithm or even failure to converge to the global optimum. Given this, a new calculation of the social perception factor β is proposed, as shown in equation (18).

$$\beta = (X_{i,j}(t-1) - X_{k,j}(t-1)) \times w(t), \quad (18)$$

where k is a randomly selected individual different from X_i in the unlearned juvenile crow population, i.e., $i \neq k$, and $w(t)$ is a weighting factor, as shown in equation.

$$w(t) = w_{\max} - \frac{w_{\max} - w_{\min}}{1 + e^{-0.1 \times r \times t \times f}}, \quad (19)$$

where w_{\max} and w_{\min} are denoted as the maximum and minimum values of the weighting factors, respectively, generally, when w_{\max} and w_{\min} are 2 and 0, better results can be obtained.

The change process of the weight factor with iteration is shown in Figure 3. At the early stage of evolution, the weight factor maintains a large value, which disguisedly increases the social learning phase in the reinforcement phase. The communication between individuals is more extensive, which makes individuals search extensively in different regions, further enhancing the global search ability of individuals and reducing the risk of the algorithm falling into the local optimum; as the evolution proceeds, the weight factor gradually decreases, especially at the late stage of evolution, the weight factor maintains a small value for a long time, which weakens the communication between individuals and other individuals, and enhances the local exploration ability of individuals in their neighborhood, which is more conducive to finding the global optimum.

(2) Improvement of RW calculation method

An in-depth analysis of RW calculation, as shown in (6), reveals that when α and β are zero simultaneously, the reinforcement phase is ineffective and fails to provide new individuals. In practical optimization, the possibility of this situation is not low. For example, in the late stage of algorithm evolution, almost all individuals in the population will gather around the optimal individual, and the vast majority of them have similar behavioral attributes, or even some behavioral attributes of some individuals are entirely identical when α and β are likely to be zero at the same time. Obviously, in order to enable individuals to perform a refined search around the obtained optimal value and thus converge to the globally optimal position, when both α and β are zero at the same time, a new stochastic reinforcement

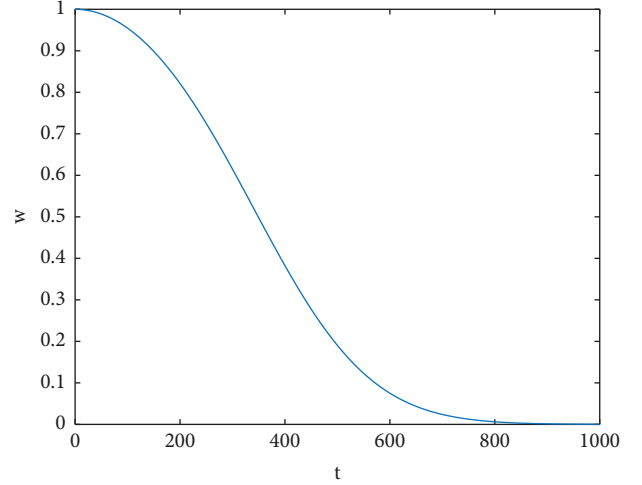


FIGURE 3: Schematic diagram of the variation of weighting factors with iterations.

strategy is proposed to calculate RW as shown in equation (20). Here, it should be noted that when α and β are not zero simultaneously, RW is still calculated according to the original way shown in equation (6).

$$RW = r \times (X_{s1,j}(t-1) - X_{s2,j}(t-1)), \quad (20)$$

where $s1$ and $s2$ are two different individuals randomly selected in the unlearned juvenile crow population, i.e., $s1 \neq s2 \neq i$, and r is a random number uniformly distributed in the range $[0, 1]$.

As seen in (20), when the individuals within the population are more similar, the above random reinforcement strategy makes the individuals perform a small random perturbation near themselves, increasing the possibility of convergence of the algorithm to the global optimum.

3.3.2. Improvements of Parent Crows Reinforcement Phase.

In NCCLA, the parent individuals in this iteration are the two relatively better individuals identified in the previous iteration of the population, which directly guide the evolution of the current generation of juveniles play an important role in the exploration and exploitation of the algorithm. As seen from the parent reinforcement phase in Section 2.3, the two-parent individuals each self-update in their independent reinforcement according to the probabilistic RP_{prob} . Further simplification reveals that the parent individuals X_{p1} and X_{p2} perform the reinforcement operation according to Equations 21 and 22.

$$X_{p1,j}(t) = e^{r1 \times (\text{mean}(j) - X_{p1,j}(t-1))}, \quad (21)$$

$$X_{p2,j}(t) = X_{2,j}(t-1) - r2 \times X_{1,j}(t-1) + e^{r1 \times (\text{mean}(j) - X_{p2,j}(t-1))}. \quad (22)$$

Both of these reinforcement methods include $e^{r1 \times (\text{mean}(j) - X_{p1,j}(t-1))}$. From the properties of the e exponential function, we can find that the effect of $e^{r1 \times (\text{mean}(j) - X_{p1,j}(t-1))}$ is to amplify the gap between $\text{mean}(j)$ and the two-parent individuals, especially when $\text{mean}(j) - X_{i,j}(t-1) > 0$, the gap amplification is more pronounced. In general, early in evolution, individual distribution is extremely dispersed, and the gap between individuals is not small, which is easily exceeded by the search space of the optimization problem after exponential amplification, resulting in ineffective reinforcement and waste of computational resources. Moreover, in the late stage of evolution, although the gap between individuals is not large, the further amplification of the gap by $e^{r1 \times (\text{mean}(j) - X_{i,j}(t-1))}$ will cause the parent individuals to produce a not small offset in this dimension, which is also very easy to deviate from the excellent evolutionary direction and will likewise cause ineffective reinforcement. Obviously, the reinforcement mentioned above of the two-parent individuals cannot effectively meet the needs of algorithm evolution. Given the different roles of the two-parent individuals in the algorithm,

the reinforcement methods of these two-parent individuals are improved separately further to balance the exploration and exploitation capabilities of the algorithm.

(1) Novel reinforcement of parent individual X_{p1}

The parent individual X_{p1} is the optimal individual determined by relying on the previous iteration of the population, which plays a vital role in guiding the algorithm's convergence. However, if its evolutionary direction points directly to the optimum local peak, it will increase the possibility of the algorithm falling into the local optimum. In view of the fact that the juvenile individuals of this generation have already achieved self-improvement based on the two more excellent parent individuals, carrying more excellent evolutionary information, which can provide more excellent reference information for the parent individuals to determine the evolutionary direction, a novel reinforcement was designed for the first parent individual X_{p1} as shown in Equation .

$$X_{p1,j}(t) = \begin{cases} X_{p1,j}(t-1) + G_1 \times (\text{mean}(j) - X_{p1,j}(t-1)), & \text{if } X_{p1,j}(t-1) \neq X_{k,j}(t) \\ X_{p1,j}(t-1) + r1 \times e^{\frac{-16t^2}{\max - t^2}} \times (X_{p1,j}(t-1) - X_{k,j}(t)), & \text{else} \end{cases}, \quad (23)$$

where $r1$ is a random number uniformly distributed in the range $[-1.5, 1.5]$, $X_{k,j}(t)$ is the j -th behavioral attribute of a randomly selected individual different from $X_{p1}(t-1)$ in the current population, and G_1 is a number conforming to the Gaussian distribution $N \sim (X_{p1,j}(t-1), 1)$, as shown in Equation

$$G_1 = \frac{1}{\sqrt{2\pi}} \times e^{\frac{-(\text{mean}(j) - X_{p1,j}(t-1))^2}{2}}. \quad (24)$$

From the above novel reinforcement approach for the parent individual X_{p1} , it can be seen that if for the j -th dimension of X_{p1} relying on the probability RP_{prob} determines that reinforcement is needed, an individual $X_k(t)$ needs to be randomly selected from the current population. By comparing whether $X_{k,j}(t)$ is equal to $X_{p1,j}(t-1)$, it is determined that learning toward the mean (j) or $X_{k,j}(t)$ is performed. At the early stage of evolution, $X_{k,j}(t)$ is almost not equal to $X_{p1,j}(t-1)$. The j -th dimension of X_{p1} is reinforced by learning from $X_{k,j}(t)$. Since the individuals in the current population already carry more excellent evolutionary information, the optimal evolutionary direction can be determined as soon as possible with reference to

their evolutionary direction. As evolution proceeds, the range $r1 \times e^{-16t^2/\max - t^2}$ of X_{p1} tries to explore a more optimal region gradually decreases, and gradually locks in a smaller range near the optimal individual to exploit a more optimal individual. This is in accordance with the process and law of evolutionary algorithms that gradually approach the region where the global optimal solution is located, avoiding the phenomenon of missing the optimal solution due to too large a search range. In addition, because the individuals of X_{p1} reinforcement learning are not the same on each behavioral attribute, it further enriches the range of dominant regions that can be explored by X_{p1} , increasing the possibility of the algorithm locking the actual optimal region and reducing the risk of the algorithm falling into a local optimum due to a single evolutionary direction. Moreover, as evolution proceeds, individuals are more similar by the late stage of evolution, and the j -th dimension of X_{p1} will be reinforced with a high probability by learning from mean (j). Compared with $X_{k,j}(t)$, the j -th dimension of X_{p1} is more likely to be different from mean (j), and the difference between mean (j) and the j -th dimension of X_{p1} is smaller, making the optimal individual X_{p1} able to perform a more

refined search within the current range, making it easier for the individual to find the optimal value of the current dimension, thus speeding up the convergence of the algorithm.

(2) Novel reinforcement of parent individual X_{p2}

Unlike the parent individual X_{p1} , which focuses on determining the evolutionary direction of the algorithm and improving the overall convergence speed, the parent individual X_{p2} focuses on providing excellent evolutionary information to guide the evolution of juveniles while maintaining population diversity. Given this, for the parent individual X_{p2} , let it learn from the optimal individual X_{p1} and other individuals in the population together and propose a novel reinforcement approach as shown in Equation .

$$\begin{aligned} X_{p2,j}(t) &= X_{p2,j}(t-1) \\ &+ r1 \times (X_{p1,j}(t-1) - X_{p2,j}(t-1)) \quad (25) \\ &+ G_2 \times (X_{p2,j}(t-1) - X_{q,j}(t)), \end{aligned}$$

where individual X_q is a juvenile randomly selected from the population of juvenile crows after reinforcement learning, and G_2 is a number that fits the Gaussian distribution $N \sim (X_{p2,j}(t-1), 0.5)$, as shown in Equation .

$$G_2 = \frac{1}{\sqrt{2\pi} \times 0.5} \times e^{-\frac{(\text{mean}(j) - X_{p1,j}(t-1))^2}{2 \times 0.5}}. \quad (26)$$

Compared with the parent individual X_{p1} reinforcement method, the parent individual X_{p2} learns from the optimal individual X_{p1} when performing reinforcement to ensure that it does not deviate from the optimal evolutionary direction, but when referring to the evolutionary information of other individuals, it can explore in a more extensive search range, potentially providing information on other excellent locations that are not in the same region as the parent individual, making it possible for the juvenile individuals learning from it to explore in other regions that are not the same as X_{p1} , further balancing the exploration and exploitation capabilities of the algorithm. In addition, the parent individual X_{p2} controls the scope of learning from the optimal individual X_{p1} and individual X_q with $r1$ and G_2 , respectively. In the early evolutionary stage, G_2 is greater than $r1$ with a greater probability, i.e., compared with learning from the optimal individual, the parent individual X_{p2} learns more from the remaining individuals, which can better maintain population diversity, ensure the global search of the algorithm, and increase the possibility of convergence of the algorithm to the global optimum. In the late evolutionary stage, G_2 is smaller than $r1$ with a greater

probability, i.e., the parent individual X_{p2} learns mainly from the optimal individual, and learns supplementally from the rest of the individuals, further ensuring that individuals perform a refined search near the optimal individual, thus locking the global optimal position.

3.4. Algorithm Complexity Analysis. Assuming that the population size is N , the maximum number of iterations is T , the problem dimension is D , and the numbers of parent and juvenile jays are N_p and N_j , respectively. The INCCLA algorithm mainly consists of the juvenile crow learning phase (T_{JL}), the juvenile crow reinforcement phase (T_{JR}), and the parent reinforcement phase (T_{PR}). The worst time complexity of each stage of the INCCLA algorithm in a single run is analyzed as follows: in the juvenile crow learning stage (T_{JL}), at most $(N_j - R) \times N$ times (1) needs to be computed, then the worst time complexity of this stage is $O((N_j - R) \times N)$; in the juvenile crow reinforcement stage (T_{JR}), at most $N_j \times D$ times (5) needs to be computed, then the worst time complexity of this stage is $O(N_j \times D)$; in the parent reinforcement stage (T_{PR}), at most $N_p \times D$ times (23) or (25) needs to be computed, then the worst time complexity of this stage is $O(N_p \times D)$.

Therefore, the worst time complexity required for a single run of INCCLA is $O((N_j - R) \times N) + O(N_j \times D) + O(N_p \times D) \approx O(T \times N \times (N_j - R + D))$.

4. Experimental Results and Discussion

In this section, to verify the performance of the INCCLA algorithm, the following four parts of experiments will be conducted in this paper: (1) The parameter sensitivity analysis; (2) Verification of the effectiveness of the proposed three improvement strategies; (3) Compare the performance of the improved INCCLA algorithm with the original NCCLA algorithm and four other representative and excellent performance evolutionary algorithms; (4) Comparison of the effectiveness of each algorithm in engineering applications.

This section uses the CEC2013 and CEC2020 test suites for experimental simulation. The CEC2013 test suite contains a total of 28 test functions, among which $F1 \sim F5$ are unimodal functions, which have only one optimal value and are used to verify the convergence performance of the algorithm; $F6 \sim F20$ are multimodal functions, which have multiple locally optimal solutions and are used to verify the ability of the algorithm to escape from the local optimum; $F21 \sim F28$ are composition functions. The CEC2020 test suite contains a total of 10 test functions, of which $F1$ is a unimodal function, $F2 \sim F4$ are multimodal shifted and rotated functions, $F5 \sim F7$ are hybrid functions, and $F8 \sim F10$ are composition functions. The relevant functions for the CEC2013 and CEC2020 test suites can be found in the literature [32, 33], respectively.

In this section, to ensure the fairness of the algorithm comparison, all algorithms were run on a computer with Windows 11 operating system, i5-11400H CPU, and programmed with MATLAB R2021a.

4.1. Sensitivity Analysis of Parameters. The INCCLA algorithm proposed in this paper involves five parameters: RP_{prob} , SL_{prob} , R , lf_{min} , and lf_{max} . Compared with the NCCLA algorithm, the INCCLA algorithm retains the originally proposed settings of RP_{prob} and SL_{prob} , changes the settings of lf_{min} and lf_{max} , and adds the parameter R . Given this, to analyze further the effect of parameters on the performance of the INCCLA algorithm, this section only analyzes the effect of parameters R , lf_{min} and lf_{max} on the performance of the INCCLA algorithm. To ensure fairness of comparison, the population size in each algorithm is $N = 50$, the dimension of the test function is $D = 30$, and the maximum number of function evaluations is $\text{MaxFEs} = 150,000$, $RP_{\text{prob}} = 0.9$, $SL_{\text{prob}} = 0.99$.

Table 1 gives the mean and average values of the optimal results obtained from 30 independent runs of the INCCLA algorithm on the CEC2013 test set when R is set to different parameters. In this experiment, $lf_{\text{min}} = 0.0001$ and $lf_{\text{max}} = 0.09$. Table 2 give the mean and average values of the optimal results of the INCCLA algorithm for 30 independent runs on the CEC2013 test set when lf_{min} and lf_{max} are set to different combinations of parameters. For this experiment, $R = 15$. Tables 1 and 2 blacken the parameters that achieved the best results on each function and count the number of functions that performed best on each parameter set in the last row.

According to the data in Table 1, it can be seen that INCCLA achieves the best convergence on 19 test functions when $R = 15$. When $R = 5, 25$, and 35 , the best convergence is achieved on 9, 11, and 14 test functions, respectively. Thus, the performance of INCCLA is sensitive to the setting of the parameter R , and the algorithm performs best when $R = 15$. Similarly, according to the data in Table 2, the performance of INCCLA is also sensitive to the settings of lf_{min} and lf_{max} , and INCCLA achieves the best convergence among the 19 tested functions when $lf_{\text{min}} = 0.0001$ and $lf_{\text{max}} = 0.09$. In summary, if there is no special requirement, INCCLA with R , lf_{min} , and lf_{max} set to 15, 0.0001, and 0.09, respectively, the algorithm can obtain better optimization results.

4.2. Experiment on Effectiveness of Each Improvement Strategy. According to Section 3, it is known that the INCCLA algorithm improves the NCCLA algorithm in three aspects. In this paper, to verify the effectiveness of the three improvement strategies, the NCCLA algorithm is combined with the three improvement strategies individually to form three new algorithms, namely, the INCCLA algorithm based on cosine similarity, the INCCLA algorithm based on improved juvenile learning phase, and the INCCLA algorithm based on improved reinforcement phase, named as INCCLA1, INCCLA2, and INCCLA3, respectively, and

compared with the original NCCLA algorithm on the CEC2013 test suit.

In this section, to ensure fairness of comparison, the population size in each algorithm is $N = 50$, the dimension of the test function is $D = 30$, and the maximum number of function evaluations is $\text{MaxFEs} = 5000 \times D = 150,000$. The other parameters of each algorithm are set as shown in Table 3. To avoid the contingency of a single operation of the algorithms, each algorithm was run 30 times independently on each test function.

Table 4 presents the running results of each algorithm on 28 test functions in 30 dimensions, where the “ \pm ” before and after represents the mean and standard deviation of the optimal values in 30 experiments, respectively, and the data that outperform the original NCCLA algorithm on the same function are marked in bold. To compare the significance of the performance of each improved strategy with the performance of NCCLA and to verify that the obtained results are not coincidental, Tables 5 and 6 present the results of the Wilcoxon rank sum test and Friedman test [34] between each improved strategy and NCCLA algorithm on 28 test functions, respectively. In Table 5, when the p value is greater than 0.05, it indicates that there is no significant difference between the improvement strategy and NCCLA, which is indicated by the symbol “ $=$.” When the p value is less than 0.05, and the mean value of the optimal solution of the result obtained in 30 experiments of the improvement strategy is better than NCCLA, it indicates that the improvement strategy is significantly better than NCCLA, which is indicated by the symbol “ $+$ ”; otherwise, it indicates that the performance of the improvement strategy is significantly worse than NCCLA, which is indicated by the symbol “ $-$.” In Table 6, the smaller the rank mean value corresponding to the algorithm, the better the algorithm’s overall performance.

Table 4 shows that INCCLA1 obtains better mean values on all the remaining 24 test functions except $F1, F8, F20$, and $F21$ compared to NCCLA. INCCLA2 obtained better mean values on all 24 tested functions except $F4, F8, F15$, and $F23$; INCCLA3 obtained better mean values on all 26 tested functions except $F17$ and $F27$. As can be seen from Table 5, the Wilcoxon rank sum test results for INCCLA1 and NCCLA on the four tested functions of $F4, F5, F10$, and $F13$ are “ $+$,” indicating that INCCLA1 outperforms NCCLA on these four functions. The Wilcoxon rank sum test result of “ $-$ ” on the $F21$ test function indicates the inferior performance of INCCLA1 over NCCLA on $F21$, while the Wilcoxon rank sum test result of “ $=$ ” on the remaining 23 test functions indicates that they perform similarly; INCCLA2 outperforms NCCLA on 14 test functions, has similar performance to NCCLA on 13 test functions, and inferior performance to NCCLA on $F23$ test functions; while INCCLA3 outperforms NCCLA on 15 test functions and has similar performance to NCCLA on 13 test functions. As can be seen from Table 6, the rank means of INCCLA2 is the smallest, indicating that the overall performance of the algorithm is superior, and the rank means of both INCCLA1 and INCCLA2 are smaller than those of NCCLA, indicating

TABLE 1: Effect of parameter R on the performance of the algorithm.

Function	$R = 5$	$R = 15$	$R = 25$	$R = 35$
F1	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F2	7.62E+05 ± 3.02E+05	8.10E+05 ± 4.96E+05	7.76E+05 ± 3.66E+05	2.45E+06 ± 3.89E+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
F4	1.13E+04 ± 3.87E+03	1.10E+04 ± 3.46E+03	1.43E+04 ± 3.64E+03	1.82E+04 ± 4.62E+03
F5	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F6	2.87E+01 ± 2.49E+01	2.40E+01 ± 2.04E+01	2.91E+01 ± 2.50E+01	3.39E+01 ± 2.58E+01
F7	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F8	2.09E+01 ± 4.40E-02	2.10E+01 ± 5.23E-02	2.09E+01 ± 8.22E-02	2.10E+01 ± 5.06E-02
F9	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F10	2.68E-01 ± 9.36E-02	2.10E-01 ± 5.91E-02	1.91E-01 ± 7.21E-02	1.24E-01 ± 5.80E-02
F11	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F12	4.00E+00 ± 2.15E+01	0.00E+00 ± 0.00E+00	4.08E+00 ± 2.23E+01	0.00E+00 ± 0.00E+00
F13	4.86E+00 ± 2.62E+01	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F14	6.82E+00 ± 4.57E+00	6.29E+00 ± 4.92E+00	3.88E+01 ± 7.61E+01	5.26E+01 ± 7.95E+01
F15	4.70E+03 ± 5.73E+02	3.94E+03 ± 6.65E+02	3.69E+03 ± 4.42E+02	3.69E+03 ± 7.34E+02
F16	1.03E+00 ± 3.89E-01	9.33E-01 ± 2.63E-01	9.36E-01 ± 3.55E-01	9.57E-01 ± 2.58E-01
F17	4.28E+00 ± 7.40E+00	5.43E+00 ± 7.92E+00	7.20E+00 ± 9.07E+00	8.64E+00 ± 4.20E+00
F18	1.18E+02 ± 3.47E+01	9.65E+01 ± 2.06E+01	9.90E+01 ± 2.63E+01	8.60E+01 ± 2.39E+01
F19	6.73E+00 ± 1.85E+00	5.32E+00 ± 2.09E+00	5.42E+00 ± 1.54E+00	5.53E+00 ± 1.84E+00
F20	1.94E-01 ± 1.05E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F21	4.15E+02 ± 8.03E+01	4.00E+02 ± 0.00E+00	4.00E+02 ± 0.00E+00	4.00E+02 ± 0.00E+00
F22	9.01E+01 ± 6.02E+01	7.70E+01 ± 5.90E+01	1.10E+02 ± 6.44E+01	1.11E+02 ± 7.74E+01
F23	3.96E+03 ± 5.75E+02	4.06E+03 ± 5.45E+02	3.55E+03 ± 7.00E+02	3.67E+03 ± 6.44E+02
F24	2.00E+02 ± 1.81E-02	2.00E+02 ± 2.44E-02	2.00E+02 ± 2.13E-02	2.00E+02 ± 1.93E-02
F25	2.59E+02 ± 3.90E+01	2.40E+02 ± 3.15E+01	2.42E+02 ± 3.21E+01	2.23E+02 ± 2.13E+01
F26	3.06E+02 ± 1.60E+01	2.87E+02 ± 3.45E+01	2.97E+02 ± 1.82E+01	3.00E+02 ± 5.68E-07
F27	3.27E+02 ± 7.05E+01	3.13E+02 ± 8.90E-01	3.13E+02 ± 9.46E-01	3.13E+02 ± 9.22E-01
F28	9.64E+02 ± 3.39E+01	9.14E+02 ± 2.15E+01	8.83E+02 ± 1.81E+01	8.60E+02 ± 1.80E+01
	9	19	11	14

Bold indicates the best results obtained on each function.

TABLE 2: Effect of parameters lf_{\min} and lf_{\max} on the performance of the algorithm.

Function	$lf_{\min} = 0.0001, lf_{\max} = 0.005$	$lf_{\min} = 0.0001, lf_{\max} = 0.09$	$lf_{\min} = 0.0005, lf_{\max} = 0.02$	$lf_{\min} = 0.005, lf_{\max} = 0.2$
F1	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F2	1.03E+06 ± 6.20E+05	8.10E+05 ± 4.96E+05	8.62E+05 ± 5.43E+05	9.26E+05 ± 5.10E+05
F3	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F4	1.42E+04 ± 4.01E+03	1.10E+04 ± 3.46E+03	1.25E+04 ± 3.36E+03	1.28E+04 ± 3.47E+03
F5	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F6	2.99E+01 ± 2.55E+01	2.40E+01 ± 2.04E+01	2.61E+01 ± 2.29E+01	2.60E+01 ± 2.23E+01
F7	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F8	2.10E+01 ± 4.85E-02	2.10E+01 ± 5.23E-02	2.10E+01 ± 4.17E-02	2.10E+01 ± 5.84E-02
F9	8.85E-01 ± 4.85E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F10	2.12E-01 ± 9.61E-02	2.10E-01 ± 5.91E-02	2.16E-01 ± 6.67E-02	2.15E-01 ± 7.12E-02
F11	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F12	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F13	4.75E+00 ± 2.60E+01	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F14	6.81E+02 ± 8.45E+02	6.29E+00 ± 4.92E+00	1.38E+02 ± 3.17E+02	9.30E+00 ± 2.29E+01
F15	4.21E+03 ± 5.84E+02	3.94E+03 ± 6.65E+02	3.89E+03 ± 7.58E+02	3.78E+03 ± 6.28E+02
F16	1.02E+00 ± 3.13E-01	9.33E-01 ± 2.63E-01	9.73E-01 ± 2.67E-01	9.77E-01 ± 3.63E-01
F17	3.17E+01 ± 4.06E+01	5.43E+00 ± 7.92E+00	1.05E+01 ± 1.10E+01	4.87E+00 ± 8.60E+00
F18	1.29E+02 ± 2.59E+01	9.65E+01 ± 2.06E+01	1.19E+02 ± 3.04E+01	9.69E+01 ± 2.65E+01
F19	7.11E+00 ± 2.59E+00	5.32E+00 ± 2.09E+00	6.10E+00 ± 2.32E+00	5.27E+00 ± 1.94E+00
F20	2.72E-01 ± 1.49E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F21	4.00E+02 ± 0.00E+00	4.00E+02 ± 0.00E+00	3.93E+02 ± 3.65E+01	4.00E+02 ± 0.00E+00
F22	5.18E+02 ± 8.92E+02	7.70E+01 ± 5.90E+01	1.25E+02 ± 7.26E+01	1.16E+02 ± 5.89E+01
F23	4.06E+03 ± 7.22E+02	4.06E+03 ± 5.45E+02	4.29E+03 ± 6.34E+02	4.02E+03 ± 6.07E+02
F24	2.00E+02 ± 2.30E-02	2.00E+02 ± 2.44E-02	2.00E+02 ± 2.24E-02	2.00E+02 ± 2.46E-02
F25	2.34E+02 ± 3.04E+01	2.40E+02 ± 3.15E+01	2.51E+02 ± 3.59E+01	2.56E+02 ± 3.80E+01
F26	3.01E+02 ± 7.13E+00	2.87E+02 ± 3.45E+01	2.97E+02 ± 2.91E+01	2.98E+02 ± 2.03E+01
F27	3.14E+02 ± 1.49E+00	3.13E+02 ± 8.90E-01	3.13E+02 ± 1.03E+00	3.13E+02 ± 1.14E+00
F28	9.10E+02 ± 2.50E+01	9.14E+02 ± 2.15E+01	9.07E+02 ± 1.73E+01	9.19E+02 ± 2.33E+01
	7	19	13	13

Bold indicates the best results obtained on each function.

TABLE 3: Algorithm-related parameters.

Algorithm	Parameter
NCCLA	$RP_{\text{prob}} = 0.9; SL_{\text{prob}} = 0.99; VSL_{\text{prob}} = 0.99; P1_{\text{prob}} = 0.95; TaE_{\text{prob}} = 0.3; lf_{\text{min}} = 0.0005; lf_{\text{max}} = 0.02$
INCCLA1	$RP_{\text{prob}} = 0.9; SL_{\text{prob}} = 0.99; VSL_{\text{prob}} = 0.99; P1_{\text{prob}} = 0.95; TaE_{\text{prob}} = 0.3; lf_{\text{min}} = 0.0005; lf_{\text{max}} = 0.02$
INCCLA2	$RP_{\text{prob}} = 0.9; SL_{\text{prob}} = 0.99; R = 15; lf_{\text{min}} = 0.0005; lf_{\text{max}} = 0.02$
INCCLA3	$RP_{\text{prob}} = 0.9; SL_{\text{prob}} = 0.99; VSL_{\text{prob}} = 0.99; P1_{\text{prob}} = 0.95; TaE_{\text{prob}} = 0.3; lf_{\text{min}} = 0.0001; lf_{\text{max}} = 0.09$

TABLE 4: Results of each improvement strategy in the 30-dimensional CEC2013 test suite.

Function	NCCLA	INCCLA1	INCCLA2	INCCLA3
F1	1.99E-05 ± 5.60E-05	4.22E-05 ± 1.59E-04	4.19E-06 ± 1.30E-05	2.10E-30 ± 9.42E-30
F2	7.59E+06 ± 4.32E+06	6.38E+06 ± 3.75E+06	6.37E+06 ± 3.13E+06	7.13E+05 ± 2.69E+05
F3	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F4	7.59E+03 ± 2.43E+03	6.33E+03 ± 2.47E+03	8.13E+03 ± 2.51E+03	3.28E+03 ± 1.98E+03
F5	8.91E-04 ± 3.77E-03	3.42E-05 ± 8.93E-05	6.86E-05 ± 2.37E-04	0.00E+00 ± 0.00E+00
F6	6.64E+01 ± 3.21E+01	5.82E+01 ± 2.90E+01	5.15E+01 ± 2.35E+01	3.12E+01 ± 2.78E+01
F7	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F8	2.10E+01 ± 5.25E-02	2.10E+01 ± 5.42E-02	2.10E+01 ± 4.17E-02	2.10E+01 ± 3.49E-02
F9	5.26E+00 ± 9.82E+00	2.23E+00 ± 6.87E+00	0.00E+00 ± 0.00E+00	2.52E+00 ± 7.71E+00
F10	9.26E+00 ± 7.28E+00	6.80E+00 ± 5.65E+00	6.67E+00 ± 3.25E+00	2.16E-01 ± 1.98E-01
F11	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
F12	1.16E+02 ± 3.57E+01	8.96E+01 ± 6.22E+01	0.00E+00 ± 0.00E+00	1.02E+02 ± 7.05E+01
F13	1.31E+02 ± 6.87E+01	1.18E+02 ± 6.15E+01	0.00E+00 ± 0.00E+00	1.07E+02 ± 8.38E+01
F14	1.78E+01 ± 3.12E+01	8.96E+00 ± 4.34E+00	4.24E+00 ± 2.81E+00	1.15E+01 ± 2.49E+01
F15	4.79E+03 ± 1.50E+03	4.33E+03 ± 1.14E+03	7.17E+03 ± 2.72E+02	3.65E+03 ± 5.90E+02
F16	2.68E+00 ± 3.22E-01	2.50E+00 ± 3.43E-01	2.62E+00 ± 2.98E-01	1.06E+00 ± 3.87E-01
F17	9.59E+00 ± 1.21E+01	7.98E+00 ± 1.19E+01	5.44E+00 ± 6.87E+00	1.10E+01 ± 1.37E+01
F18	1.77E+02 ± 4.90E+01	1.66E+02 ± 4.61E+01	1.43E+02 ± 4.88E+01	1.54E+02 ± 3.64E+01
F19	1.57E+01 ± 7.70E+00	1.47E+01 ± 6.78E+00	7.74E+00 ± 2.36E+00	9.11E+00 ± 2.73E+00
F20	1.31E+01 ± 3.16E+00	1.34E+01 ± 2.31E+00	0.00E+00 ± 0.00E+00	1.20E+01 ± 3.58E+00
F21	4.00E+02 ± 2.73E-01	4.00E+02 ± 3.12E-01	4.00E+02 ± 1.97E-01	4.00E+02 ± 7.62E-12
F22	1.17E+02 ± 6.11E+01	7.99E+01 ± 7.61E+01	9.09E+01 ± 6.46E+01	1.04E+02 ± 8.25E+01
F23	4.21E+03 ± 6.46E+02	4.07E+03 ± 1.11E+03	6.90E+03 ± 4.60E+02	3.87E+03 ± 9.03E+02
F24	2.04E+02 ± 1.37E+01	2.01E+02 ± 7.70E-01	2.00E+02 ± 4.13E-02	2.02E+02 ± 1.10E+01
F25	2.88E+02 ± 3.40E+01	2.79E+02 ± 3.98E+01	2.48E+02 ± 4.21E+01	2.85E+02 ± 3.25E+01
F26	3.08E+02 ± 4.25E+01	2.99E+02 ± 4.17E+01	2.97E+02 ± 1.82E+01	2.97E+02 ± 3.89E+01
F27	4.46E+02 ± 2.75E+02	3.94E+02 ± 1.81E+02	3.22E+02 ± 2.68E+00	6.16E+02 ± 3.16E+02
F28	1.46E+03 ± 8.95E+02	1.21E+03 ± 5.33E+02	1.08E+03 ± 5.76E+01	1.05E+03 ± 3.53E+02

Bold indicates the best results obtained on each function.

that all three improvement strategies proposed in this paper achieve better results.

In summary, all three improvement strategies proposed in this paper have certain improvement effects on NCCLA. The improvement strategies in the juvenile learning phase and the improvement strategies in the reinforcement phase have the most obvious improvement effects.

4.3. Performance Comparison of INCCLA with Other Algorithms. In this section, to verify the superior performance of the INCCLA algorithm in terms of convergence precision and convergence speed, this section compares the INCCLA with the NCCLA algorithm and the four better evolutionary algorithms on the CEC2013 and CEC2020 test suites, including the artificial bee colony algorithm based on new neighborhood selection mechanism (NSABC) [35], the sine cosine algorithm based on transition parameters and mutation operators (MSCA) [36], artificial tree algorithm

based on two populations (IATTP) [37] and the improved crow search algorithm (ICSA) [38]. To ensure the fairness of the comparison, the population size in each algorithm is $N = 50$, and the maximum number of function evaluations is $\text{MaxFEs} = 150,000$. The other parameters of each algorithm are set as shown in Table 7, where the parameter values of each comparison algorithm are taken as in the original paper.

4.3.1. Comparison of INCCLA with Other Algorithms on Convergence Precision

(1) *Testing at CEC 2013.* In order to fully compare the performance of INCCLA with other algorithms in terms of convergence precision, tests were conducted on the CEC2013 test suite with three different dimensions, $D = 10$, $D = 30$, and $D = 100$, respectively. Tables 8–10 give the mean and standard deviation of 30 independent experiments for each algorithm

TABLE 5: Wilcoxon rank sum test results between NCCLA and each improvement strategy.

Function	p value (vs.NCCLA)		
	INCCLA1	INCCLA2	INCCLA3
F1	0.876 (=)	0.053 (=)	0.000 (+)
F2	0.304 (=)	0.379 (=)	0.000 (+)
F3	1.000 (=)	1.000 (=)	1.000 (=)
F4	0.029 (+)	0.363 (=)	0.000 (+)
F5	0.003 (+)	0.004 (+)	0.000 (+)
F6	0.559 (=)	0.082 (=)	0.000 (+)
F7	1.000 (=)	1.000 (=)	1.000 (=)
F8	0.067 (=)	0.340 (=)	0.077 (=)
F9	0.179 (=)	0.005 (+)	0.218 (=)
F10	0.035 (+)	0.270 (=)	0.000 (+)
F11	1.000 (=)	1.000 (=)	1.000 (=)
F12	0.302 (=)	0.000 (+)	0.385 (=)
F13	0.025 (+)	0.000 (+)	0.616 (=)
F14	0.491 (=)	0.000 (+)	0.012 (+)
F15	0.099 (=)	0.000 (+)	0.000 (+)
F16	0.057 (=)	0.395 (=)	0.000 (+)
F17	0.090 (=)	0.118 (=)	0.137 (=)
F18	0.420 (=)	0.006 (+)	0.045 (+)
F19	0.706 (=)	0.000 (+)	0.000 (+)
F20	0.264 (=)	0.000 (+)	0.273 (=)
F21	0.019 (-)	0.994 (=)	0.000 (+)
F22	0.067 (=)	0.050 (=)	0.911 (=)
F23	0.112 (=)	0.000 (-)	0.108 (=)
F24	0.482 (=)	0.000 (+)	0.000 (+)
F25	0.807 (=)	0.000 (+)	0.473 (=)
F26	0.529 (=)	0.000 (+)	0.000 (+)
F27	0.599 (=)	0.000 (+)	0.510 (=)
F28	0.258 (=)	0.002 (+)	0.000 (+)
+/-/-	4/23/1	14/13/1	15/13/0

TABLE 6: Friedman test results for four algorithms.

	NCCLA	INCCLA1	INCCLA2	INCCLA3
Avg.rank	3.48	2.48	1.96	2.07
Sort	4	3	1	2

TABLE 7: Related parameter settings of each algorithm.

Algorithm	Parameter
NCCLA	$RP_{\text{prob}} = 0.9; SL_{\text{prob}} = 0.99; VSL_{\text{prob}} = 0.99; P1_{\text{prob}} = 0.95; TaE_{\text{prob}} = 0.3; lf_{\text{min}} = 0.0005; lf_{\text{max}} = 0.02$
NSABC	$C = 1.5; \text{limit} = 100; k = 10$
MSCA	$a = 2; b = 0.5$
IATTP	$h1 = h2 = h3 = 0.5, h4 = 0.8, m = 50, q = 0.8$
ICSA	$AP = 0.1; FL = 1.5$
INCCLA	$RP_{\text{prob}} = 0.9; SL_{\text{prob}} = 0.99; R = 15; lf_{\text{min}} = 0.0001; lf_{\text{max}} = 0.09$

on the 10-dimension, 30-dimension, and 100-dimension CEC2013 datasets. The functions that achieved the best optimization results on the same functions are bold. To further verify the differences between INCCLA and each algorithm, the Wilcoxon rank sum test with a 5% significance level was performed between INCCLA and each algorithm, and the results are shown in Table 11. To comprehensively evaluate the overall performance of all algorithms, the results of the Friedman test are given in Table 12.

For the 10-dimensional optimization problem, as shown in Table 8, INCCLA achieves better mean values in all 28 test functions compared with NCCLA, and INCCLA achieves global optimum on 9 test functions, including F1, F3, F5, F7, F9, F11, F12, F13, and F20. Both the INCCLA algorithm and NSABC achieved the global optimum on nine test functions, including F1, F3, F5, F7, F9, F11, F12, F13, and F20, and the two algorithms achieved comparable mean values on functions F25 and F26. On five functions, including F19,

TABLE 8: Data result of INCCLA with other algorithms on the 10-dimensional CEC2013 test suite.

Fun.	NSABC		MSCA		IATTP		ICSA		NCCLA		INCCLA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	0.00E+00	0.00E+00	2.17E+02	2.79E+02	8.91E-03	1.05E-02	4.53E-07	2.34E-06	1.39E-08	4.82E-08	0.00E+00	0.00E+00
F2	7.16E+05	4.42E+05	2.62E+06	2.16E+06	5.11E+04	4.09E+04	3.07E+05	2.59E+05	7.28E+05	1.16E+06	4.01E+04	3.74E+04
F3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F4	7.16E+03	2.30E+03	5.31E+03	1.98E+03	7.64E+00	3.64E+00	1.04E+03	6.49E+02	1.55E+03	2.11E+03	5.52E+02	4.35E+02
F5	0.00E+00	0.00E+00	4.85E+01	1.34E+01	4.64E-02	3.16E-02	7.30E-01	3.80E+00	5.24E-07	1.43E-06	0.00E+00	0.00E+00
F6	5.71E+00	4.78E+00	2.03E+01	2.04E+01	7.76E+00	3.97E+00	2.09E+01	2.60E+01	2.82E+01	3.25E+01	4.03E+00	4.80E+00
F7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F8	6.78E-01	3.71E+00	4.44E-16	3.01E-31	4.44E-16	3.01E-31	6.77E-01	3.71E+00	1.22E+01	1.01E+01	4.44E-16	3.01E-31
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	7.48E-01	2.40E-01	2.86E+01	2.44E+01	7.28E-01	1.61E-01	5.27E+00	4.75E+00	5.90E+00	4.26E+00	1.31E-01	8.01E-02
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F13	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F14	1.92E+00	5.25E+00	1.68E+03	2.04E+02	1.38E+03	1.70E+02	1.20E+02	1.10E+02	2.60E+00	3.27E+00	7.04E-01	1.56E+00
F15	6.26E+02	1.55E+02	1.54E+03	1.68E+02	1.03E+03	1.78E+02	5.33E+02	2.85E+02	7.31E+02	2.81E+02	4.43E+02	2.14E+02
F16	5.16E-01	1.79E-01	1.20E+00	2.36E-01	1.11E+00	1.50E-01	1.13E+00	2.30E-01	6.28E-01	2.82E-01	4.10E-01	1.85E-01
F17	4.98E+00	5.10E+00	7.16E+01	1.88E+01	3.12E+01	5.59E+00	6.15E+00	4.54E+00	4.89E+00	5.12E+00	4.43E+00	4.92E+00
F18	2.06E+01	3.94E+00	6.70E+01	1.84E+01	3.41E+01	6.04E+00	2.52E+01	6.02E+00	2.51E+01	1.10E+01	1.79E+01	5.21E+00
F19	7.05E-01	1.89E-01	5.42E+00	1.34E+00	2.23E+00	3.18E-01	8.89E-01	2.04E-01	1.65E+00	1.01E+00	8.08E-01	2.49E-01
F20	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.50E-01	8.23E-01	0.00E+00	0.00E+00
F21	3.19E+02	9.82E+01	4.09E+02	2.50E+01	4.00E+02	2.56E-04	3.84E+02	5.14E+01	3.98E+02	7.01E+00	3.90E+02	5.59E+01
F22	7.17E+01	5.88E+01	1.32E+03	2.90E+02	1.34E+03	2.20E+02	1.20E+02	8.23E+01	8.72E+01	6.55E+01	6.54E+01	6.12E+01
F23	6.37E+02	1.59E+02	1.65E+03	2.05E+02	1.43E+03	1.52E+02	3.02E+02	1.39E+02	8.88E+02	3.61E+02	3.96E+02	2.06E+02
F24	1.00E+02	8.14E-03	1.73E+02	4.50E+01	1.00E+02	1.15E-02	1.17E+02	3.79E+01	1.69E+02	4.73E+01	1.50E+02	5.09E+01
F25	1.00E+02	1.28E-04	1.17E+02	3.76E+01	1.01E+02	1.41E-01	1.03E+02	1.98E+00	1.00E+02	1.08E-04	1.00E+02	0.00E+00
F26	1.00E+02	4.96E-13	1.17E+02	3.79E+01	1.00E+02	4.28E-13	1.00E+02	4.65E-12	1.23E+02	4.30E+01	1.20E+02	4.07E+01
F27	3.09E+02	3.35E-01	3.16E+02	4.02E+01	3.08E+02	1.45E-01	3.10E+02	5.89E-01	3.21E+02	4.33E+01	3.09E+02	3.19E-01
F28	3.31E+02	2.48E+02	6.65E+02	1.29E+02	3.37E+02	2.77E+02	4.43E+02	3.01E+02	5.18E+02	2.82E+02	5.06E+02	2.53E+02

Bold indicates the best results obtained on each function.

TABLE 9: Data result of INCCLA with other algorithms on the 30-dimensional CEC2013 test suite.

Fun.	NSABC			MSCA			IATTP			ICSA			NCCLA			INCCLA		
	Mean	Std		Mean	Std		Mean	Std		Mean	Std		Mean	Std		Mean	Std	
F1	0.00E+00	0.00E+00	9.60E+03	4.32E+03	3.00E+00	3.36E+01	3.41E+01	3.41E+01	1.99E-05	5.60E-05	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F2	1.01E+07	3.38E+06	7.18E+07	2.22E+07	3.71E+06	2.11E+07	1.04E+07	1.04E+07	7.59E+06	4.32E+06	8.10E+05	8.10E+05	4.96E+05	4.96E+05	4.96E+05	4.96E+05	4.96E+05	
F3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F4	8.13E+04	1.31E+04	3.74E+04	7.65E+03	4.01E+02	1.33E+04	2.91E+03	2.91E+03	7.59E+03	2.43E+03	1.10E+04	1.10E+04	3.46E+03	3.46E+03	3.46E+03	3.46E+03	3.46E+03	
F5	0.00E+00	0.00E+00	2.72E+03	1.54E+03	1.58E+01	1.58E+01	9.80E+00	3.09E+02	8.91E-04	3.77E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F6	2.27E+01	1.20E+01	6.45E+02	3.67E+02	8.24E+01	1.35E+02	3.38E+01	3.38E+01	6.64E+01	3.21E+01	2.40E+01	2.40E+01	2.04E+01	2.04E+01	2.04E+01	2.04E+01	2.04E+01	
F7	0.00E+00	0.00E+00	8.51E-02	3.51E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F8	2.10E+01	5.54E-02	2.10E+01	5.62E-02	2.10E+01	2.10E+01	6.35E-02	6.35E-02	2.10E-01	5.25E-02	2.10E+01	2.10E+01	5.23E-02	5.23E-02	5.23E-02	5.23E-02	5.23E-02	
F9	0.00E+00	0.00E+00	2.37E+00	6.82E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.26E+00	9.82E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F10	2.20E+00	4.88E-01	1.06E+03	2.58E+02	2.14E+01	1.56E+02	7.69E+01	7.69E+01	9.26E+00	7.28E+00	2.10E-01	2.10E-01	5.91E-02	5.91E-02	5.91E-02	5.91E-02	5.91E-02	
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F12	2.03E+01	4.62E+01	3.05E+01	6.95E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.16E+02	3.57E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F13	8.17E+00	3.11E+01	4.51E+01	8.33E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.31E+02	6.87E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F14	6.38E+00	2.25E+01	7.02E+03	4.08E+02	6.87E+03	2.73E+03	6.75E+02	6.75E+02	1.78E+01	3.12E+01	6.29E+00	6.29E+00	4.92E+00	4.92E+00	4.92E+00	4.92E+00	4.92E+00	
F15	3.99E+03	3.54E+02	7.97E+03	2.26E+02	6.90E+03	6.11E+03	1.01E+03	1.01E+03	4.79E+03	1.50E+03	3.94E+03	3.94E+03	6.65E+02	6.65E+02	6.65E+02	6.65E+02	6.65E+02	
F16	1.08E+00	2.70E-01	2.79E+00	2.88E-01	2.57E+00	2.60E+00	2.71E-01	2.71E-01	2.68E+00	3.22E-01	9.33E-01	9.33E-01	2.63E-01	2.63E-01	2.63E-01	2.63E-01	2.63E-01	
F17	6.77E+00	1.19E+01	6.47E+02	1.18E+02	2.05E+02	1.91E+01	3.07E+01	3.07E+01	9.59E+00	1.21E+01	5.43E+00	5.43E+00	7.92E+00	7.92E+00	7.92E+00	7.92E+00	7.92E+00	
F18	1.66E+02	2.57E+01	6.17E+02	1.94E+02	2.21E+02	2.22E+02	3.41E+01	3.41E+01	1.77E+02	4.90E+01	9.65E+01	9.65E+01	2.06E+01	2.06E+01	2.06E+01	2.06E+01	2.06E+01	
F19	8.23E+00	1.70E+00	3.10E+03	4.81E+03	1.73E+01	1.58E+01	8.31E+00	8.31E+00	1.57E+01	7.70E+00	5.32E+00	5.32E+00	2.09E+00	2.09E+00	2.09E+00	2.09E+00	2.09E+00	
F20	8.44E+00	6.04E+00	3.10E+00	5.26E+00	5.56E-01	2.13E+00	4.69E+00	4.69E+00	1.31E+01	3.16E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F21	3.97E+02	1.69E+01	1.01E+03	2.52E+02	4.02E+02	5.22E-01	2.11E+01	2.11E+01	4.00E+02	2.73E-01	4.00E+02	4.00E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F22	8.12E+01	6.21E+01	7.45E+03	5.44E+02	7.20E+03	3.25E+02	7.62E+02	7.62E+02	1.17E+02	6.11E+01	7.70E+01	7.70E+01	5.90E+01	5.90E+01	5.90E+01	5.90E+01	5.90E+01	
F23	4.40E+03	3.98E+02	7.87E+03	3.16E+02	7.60E+03	4.00E+02	1.21E+03	1.21E+03	4.21E+03	6.46E+02	4.06E+03	4.06E+03	5.45E+02	5.45E+02	5.45E+02	5.45E+02	5.45E+02	
F24	2.00E+02	4.34E-02	2.07E+02	1.04E+01	2.00E+02	9.22E-02	2.95E-01	2.95E-01	2.04E+02	1.37E+01	2.00E+02	2.00E+02	2.44E-02	2.44E-02	2.44E-02	2.44E-02	2.44E-02	
F25	2.27E+02	2.13E+01	2.78E+02	3.44E+01	2.42E+02	3.48E+01	2.12E+01	2.12E+01	2.88E+02	3.40E+01	2.40E+02	2.40E+02	3.15E+01	3.15E+01	3.15E+01	3.15E+01	3.15E+01	
F26	2.01E+02	1.77E-01	3.06E+02	5.46E+01	2.58E+02	4.81E+01	3.01E+01	3.01E+01	3.08E+02	4.25E+01	2.87E+02	2.87E+02	3.45E+01	3.45E+01	3.45E+01	3.45E+01	3.45E+01	
F27	3.31E+02	2.04E+01	6.23E+02	2.42E+02	3.37E+02	9.82E+01	8.59E+01	8.59E+01	4.46E+02	2.75E+02	3.13E+02	3.13E+02	8.90E-01	8.90E-01	8.90E-01	8.90E-01	8.90E-01	
F28	9.39E+02	1.59E+02	2.58E+03	4.31E+02	1.01E+03	2.17E+01	3.69E+02	3.69E+02	1.46E+03	8.95E+02	9.14E+02	9.14E+02	2.15E+01	2.15E+01	2.15E+01	2.15E+01	2.15E+01	

Bold indicates the best results obtained on each function.

TABLE 10: Data result of INCCCLA with other algorithms on the 100-dimensional CEC2013 test suite.

Fun.	NSABC			MSCA			IATTP			ICSA			NCCLA			INCCCLA		
	Mean	Std		Mean	Std		Mean	Std		Mean	Std		Mean	Std		Mean	Std	
F1	1.85E-20	2.55E-20	8.85E+04	9.37E+03	2.84E+02	1.36E+04	3.64E+03	1.73E+03	2.84E+02	1.36E+04	3.64E+03	1.15E+00	1.85E+00	7.80E-12	1.18E-11			
F2	6.23E+07	1.21E+07	9.81E+08	2.70E+08	4.82E+07	2.20E+08	7.15E+07	2.20E+08	4.82E+07	2.20E+08	7.15E+07	3.58E+07	8.52E+06	2.02E+07	4.71E+06			
F3	0.00E+00	0.00E+00	1.28E+15	3.11E+15	2.34E+10	1.53E+10	1.45E+12	1.53E+10	2.34E+10	7.89E+11	1.45E+12	0.00E+00	0.00E+00	0.00E+00	0.00E+00			
F4	3.11E+05	3.36E+04	2.21E+05	1.57E+04	6.28E+03	3.95E+04	1.66E+04	3.95E+04	6.28E+03	9.43E+04	1.66E+04	6.76E+04	1.36E+04	1.19E+05	1.52E+04			
F5	2.19E-12	2.45E-12	4.07E+04	1.22E+04	2.17E+02	8.53E+03	3.07E+03	1.63E+03	2.17E+02	8.53E+03	3.07E+03	1.81E+00	4.28E+00	1.74E-06	1.22E-06			
F6	2.27E+02	3.41E+01	1.37E+04	2.93E+03	1.28E+02	2.03E+03	3.77E+02	8.88E+02	1.28E+02	2.03E+03	3.77E+02	3.20E+02	5.22E+01	2.36E+02	5.69E+01			
F7	4.26E+02	2.99E+02	1.26E+04	9.57E+03	1.51E+02	5.76E+02	4.53E+02	2.49E+02	1.51E+02	5.76E+02	4.53E+02	3.93E+02	4.94E+02	0.00E+00	0.00E+00			
F8	2.13E+01	2.11E-02	2.13E+01	3.44E-02	1.82E-02	2.13E+01	2.34E-02	2.13E+01	1.82E-02	2.13E+01	2.34E-02	2.13E+01	2.89E-02	2.13E+01	2.14E-02			
F9	1.28E+02	3.51E+00	1.24E+02	8.02E+00	6.41E+00	1.18E+02	1.05E+01	1.18E+02	6.41E+00	7.86E+01	1.05E+01	1.01E+02	8.79E+00	7.02E+01	3.63E+01			
F10	2.49E+01	1.08E+01	9.81E+03	1.29E+03	2.26E+02	2.29E+03	4.02E+02	1.41E+03	2.26E+02	2.29E+03	4.02E+02	1.60E+02	4.54E+01	4.63E+00	1.45E+00			
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.96E-01	3.81E+00	8.47E+01	2.79E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00			
F12	8.02E+02	7.60E+01	1.25E+03	1.38E+02	7.86E+01	5.77E+02	7.69E+01	7.08E+02	7.86E+01	5.77E+02	7.69E+01	5.71E+02	1.02E+02	5.21E+02	7.37E+01			
F13	9.18E+02	7.95E+01	1.22E+03	1.55E+02	6.97E+02	6.97E+02	4.48E+01	6.97E+02	6.93E+01	6.27E+02	4.48E+01	8.15E+02	1.51E+02	6.24E+02	4.86E+01			
F14	1.80E+02	1.49E+02	3.14E+04	7.61E+02	1.24E+03	1.89E+04	2.67E+03	3.15E+04	1.24E+03	1.89E+04	2.67E+03	2.65E+02	2.15E+02	7.58E+03	5.27E+03			
F15	1.80E+04	9.42E+02	3.21E+04	5.71E+02	6.01E+02	3.05E+04	5.18E+02	3.10E+04	6.01E+02	3.05E+04	5.18E+02	3.01E+04	2.86E+03	2.03E+04	1.71E+03			
F16	2.56E+00	3.13E-01	4.53E+00	2.88E-01	2.58E-01	4.30E+00	2.67E-01	4.36E+00	2.58E-01	4.30E+00	2.67E-01	4.27E+00	2.93E-01	2.06E+00	3.75E-01			
F17	4.28E+00	9.78E+00	4.72E+03	3.43E+02	7.39E+01	1.28E+03	2.28E+02	1.31E+03	7.39E+01	1.28E+03	2.28E+02	4.60E+01	8.90E+00	9.60E+01	2.45E+01			
F18	1.75E+03	1.44E+02	4.90E+03	4.89E+02	8.63E+01	1.74E+03	1.58E+02	1.32E+03	8.63E+01	1.74E+03	1.58E+02	1.14E+03	2.04E+02	7.03E+02	1.24E+02			
F19	1.10E+02	1.27E+01	4.60E+05	1.90E+05	2.81E+02	7.19E+03	4.08E+03	6.99E+02	2.81E+02	7.19E+03	4.08E+03	1.80E+02	4.93E+01	7.04E+01	1.88E+01			
F20	5.00E+01	0.00E+00	5.00E+01	0.00E+00	2.63E-09	5.00E+01	1.50E-04	5.00E+01	2.63E-09	5.00E+01	1.50E-04	5.00E+01	1.83E-02	4.99E+01	1.65E-01			
F21	3.79E+02	3.43E+01	7.31E+03	2.84E+02	4.26E+02	5.05E+03	7.92E+02	1.29E+03	4.26E+02	5.05E+03	7.92E+02	4.46E+02	6.81E+00	3.49E+02	4.88E+01			
F22	2.54E+02	1.24E+02	3.16E+04	6.60E+02	1.33E+03	1.78E+04	3.63E+03	3.11E+04	1.33E+03	1.78E+04	3.63E+03	2.89E+02	1.27E+02	7.59E+03	6.74E+03			
F23	2.23E+04	1.73E+03	3.35E+04	6.13E+02	6.55E+02	3.07E+04	8.30E+02	3.31E+04	6.55E+02	3.07E+04	8.30E+02	3.11E+04	3.75E+03	2.22E+04	1.48E+03			
F24	4.72E+02	7.75E+01	5.65E+02	4.14E+01	6.67E+01	4.12E+02	3.62E+01	4.40E+02	6.67E+01	4.12E+02	3.62E+01	4.06E+02	1.25E+02	2.42E+02	8.01E+01			
F25	6.06E+02	1.35E+01	6.73E+02	2.88E+01	1.70E+01	5.51E+02	1.74E+01	5.57E+02	1.70E+01	5.51E+02	1.74E+01	5.76E+02	2.28E+01	5.09E+02	4.82E+01			
F26	2.19E+02	7.44E+01	5.45E+02	4.62E+01	1.06E+02	4.27E+02	5.66E+01	4.52E+02	1.06E+02	4.27E+02	5.66E+01	4.47E+02	9.21E+01	3.10E+02	3.85E+01			
F27	3.30E+03	2.41E+02	3.30E+03	4.34E+02	6.09E+02	1.97E+03	4.12E+02	2.49E+03	6.09E+02	1.97E+03	4.12E+02	2.16E+03	7.89E+02	5.76E+02	5.95E+02			
F28	4.58E+03	1.28E+03	1.47E+04	1.01E+03	3.02E+03	1.02E+04	1.02E+03	9.16E+03	3.02E+03	1.02E+04	1.02E+03	7.16E+03	3.07E+03	4.08E+03	1.19E+03			

Bold indicates the best results obtained on each function.

TABLE 11: Wilcoxon rank sum test results of INCCLA with other algorithms on the 10-dimensional and 30-dimensional CEC2013 test suite.

Function	$D = 10$ p value (vs.INCCLA)					$D = 30$ p value (vs.INCCLA)					$D = 100$ p value (vs.INCCLA)				
	NSABC	MSCA	IATTP	ICSA	NCCLA	NSABC	MSCA	IATTP	ICSA	NCCLA	NSABC	MSCA	IATTP	ICSA	NCCLA
F1	1.000 (=)	0.000 (-)	0.000 (-)	0.000 (-)	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 (-)	0.290 (=)	0.000 (-)	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 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	0.000 (-)	1.000 (=)	
F4	0.000 (-)	0.000 (-)	0.000 (+)	0.000 (-)	0.002 (-)	0.000 (-)	0.000 (-)	0.000 (+)	0.004 (-)	0.000 (+)	0.000 (-)	0.000 (-)	0.000 (+)	0.000 (+)	
F5	1.000 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	1.000 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (+)	0.000 (-)	0.000 (-)	0.000 (-)	
F6	0.030 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (+)	0.000 (-)	0.000 (-)	0.000 (-)	0.190 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	
F7	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	0.160 (=)	1.000 (=)	1.000 (=)	1.000 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	
F8	0.333 (=)	1.000 (=)	1.000 (=)	0.333 (=)	0.000 (-)	0.865 (=)	0.935 (=)	0.258 (=)	0.654 (=)	0.695 (=)	0.970 (=)	0.137 (=)	0.728 (=)	0.180 (=)	
F9	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	0.011 (-)	1.000 (=)	1.000 (=)	0.005 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.200 (=)	
F10	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	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 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	0.081 (=)	1.000 (=)	1.000 (=)	0.333 (=)	0.000 (-)	1.000 (=)	
F12	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	0.021 (-)	0.021 (-)	1.000 (=)	1.000 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.006 (-)	0.065 (=)	
F13	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	0.160 (=)	0.002 (-)	1.000 (=)	1.000 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.706 (=)	0.000 (-)	
F14	0.745 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.004 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	
F15	0.000 (-)	0.000 (-)	0.000 (-)	0.195 (=)	0.000 (-)	0.807 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.039 (-)	0.000 (+)	0.000 (-)	0.000 (-)	0.000 (-)	
F16	0.030 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.002 (-)	0.046 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	
F17	0.075 (=)	0.000 (-)	0.000 (-)	0.037 (-)	0.176 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.007 (-)	0.000 (+)	0.000 (-)	0.000 (-)	0.000 (+)	
F18	0.015 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.004 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	
F19	0.055 (=)	0.000 (-)	0.000 (-)	0.411 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	
F20	1.000 (=)	1.000 (=)	1.000 (=)	1.000 (=)	0.333 (=)	0.000 (-)	0.002 (-)	0.160 (=)	0.000 (-)	0.000 (-)	0.041 (-)	0.041 (-)	0.151 (=)	0.043 (=)	
F21	0.001 (+)	0.000 (-)	0.000 (-)	0.038 (-)	0.000 (-)	0.333 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	
F22	0.711 (=)	0.000 (-)	0.000 (-)	0.001 (-)	0.013 (-)	0.166 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.028 (-)	0.000 (+)	0.000 (-)	0.000 (-)	0.000 (+)	
F23	0.000 (-)	0.000 (-)	0.000 (-)	0.102 (=)	0.000 (-)	0.011 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.297 (=)	0.652 (=)	0.000 (-)	0.000 (-)	0.000 (-)	
F24	0.003 (+)	0.000 (-)	0.061 (=)	0.395 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	
F25	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.888 (=)	0.000 (-)	0.455 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	
F26	0.011 (+)	1.000 (=)	0.011 (+)	0.011 (+)	0.479 (=)	0.000 (+)	0.000 (-)	0.911 (=)	0.000 (-)	0.000 (-)	0.000 (+)	0.000 (-)	0.000 (-)	0.000 (-)	
F27	0.000 (-)	0.000 (-)	0.000 (+)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	
F28	0.002 (+)	0.000 (-)	0.317 (=)	0.771 (=)	0.055 (=)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	0.000 (-)	
+/-/-	4/14/10	0/9/19	3/11/14	1/13/14	0/10/18	2/12/14	0/4/24	1/10/17	0/7/21	1/5/22	7/5/16	0/2/26	1/3/24	1/4/23	4/4/20

TABLE 12: Friedman test results of 6 algorithms.

Dimension		NSABC	MSCA	IATTP	ICSA	NCCLA	INCCLA
$D = 10$	Avg.rank	2.68	4.96	3.54	3.50	4.07	2.25
	sort	2	6	3	4	5	1
$D = 30$	Avg.rank	2.50	5.34	3.57	4.07	3.73	1.79
	sort	2	6	3	5	4	1
$D = 100$	Avg.rank	2.91	5.59	4.04	3.93	2.86	1.68
	sort	3	6	5	4	2	1

F_{21} , F_{24} , F_{26} , and F_{28} , the INCCLA algorithm had inferior mean values to NSABC, and on the remaining 12 functions, the INCCLA algorithm obtained better mean values than NSABC. MSCA, IATTP, and ICSA all achieved theoretical optima on only seven test functions, including F_3 , F_7 , F_9 , F_{11} , F_{12} , F_{13} , and F_{20} . As seen in Table 11, NCCLA has similar performance on 10 functions compared to INCCLA, but significantly inferior performance on the 18 tested functions; NSABC performs significantly better on 4 functions but significantly inferior on 10 functions; MSCA has similar performance on 9 of the tested functions, but significantly inferior performance on the remaining 19 tested functions; IATTP has significantly better performance on F_4 , F_{26} , and F_{27} only, but significantly inferior performance on the 14 tested functions; ICSA has significantly better performance on F_{26} only, but significantly inferior performance on all 14 functions. In summary, it shows that for low-dimensional optimization problems, the INCCLA proposed in this paper has some advantages in terms of convergence precision compared with other representative methods.

For the 30-dimensional optimization problem, it can be seen from Table 9 that INCCLA achieves the global optimum on nine test functions, including F_1 , F_3 , F_5 , F_7 , F_9 , F_{11} , F_{12} , F_{13} , and F_{20} . Like INCCLA, NCCLA also achieves the global optimum on these nine test functions, and INCCLA achieves worse mean values than NCCLA only on F_4 , but better mean values on the remaining test functions; NSABC achieved theoretical optima on six functions, including F_1 , F_3 , F_5 , F_7 , F_9 , and F_{11} ; MSCA obtained the theoretical optimal results on F_3 and F_{11} only; IATTP obtained the theoretical optimal for six functions, including F_3 , F_7 , F_9 , F_{11} , F_{12} , and F_{13} ; ICSA obtained the theoretical optimal for five functions, including F_3 , F_7 , F_9 , F_{12} , and F_{13} . As seen in Table 11, compared to INCCLA, NCCLA has similar performance on 5 functions and significantly better performance on F_4 only, but significantly inferior performance on the 22 tested functions; NSABC has significantly better performance on F_6 and F_{26} , but significantly inferior performance on 14 functions; MSCA has similar performance on 4 tested functions, but significantly inferior performance on the remaining 24 tested functions; IATTP had significantly better performance on F_4 only, but significantly inferior performance on 17 test functions; ICSA has similar performance on 7 test functions only, but significantly inferior performance on the remaining 21 test functions. In summary, it shows that for the 30-dimensional CEC2013 test suite, the INCCLA proposed in this paper has a significant advantage in

convergence accuracy compared with other representative methods, and the performance gap between algorithms is significantly more significant than that of the 10-dimensional optimization problem.

For the 100-dimensional optimization problem, it can be seen from Table 10 that INCCLA achieves the global optimum on three test functions, including F_3 , F_7 , and F_{11} . NCCLA achieves the global optimum on F_3 and F_{11} , except that INCCLA and NCCLA obtain the same mean value on F_8 , NCCLA obtains a better mean value than INCCLA on F_4 , F_{14} , F_{17} , and F_{22} , but NCCLA obtains a worse mean value than INCCLA on the rest of the 21 functions; NSABC obtained the global optimum on F_3 and F_{11} ; MSCA obtained the theoretical optimum on F_{11} only; neither IATTP nor ICSA obtained the theoretical optimum on any function. As seen in Table 11, NCCLA performs significantly better on only four functions compared to INCCLA, including F_4 , F_{14} , F_{17} , and F_{22} , but performs significantly inferior on the 20 test functions; NSABC has similar performance on 5 tested functions, significantly better performance on 7 functions, but significantly inferior performance on 16 functions; MSCA has similar performance on F_8 and F_{11} , but significantly inferior performance on the remaining 26 test functions; IATTP has significantly better performance on F_4 only, but significantly inferior performance on the 24 test functions; ICSA has significantly better performance on F_4 only, but significantly inferior performance on the 23 test functions. In summary, it shows that for high-dimensional optimization problems, the INCCLA proposed in this paper has a significant advantage in convergence precision compared with other representative methods, and the performance gap between the algorithms is more significant than that for optimization problems in 10 and 30 dimensions.

Table 12 shows that the overall performance of INCCLA is the best among the six optimization algorithms for the 10-dimensional, 30-dimensional, and 100-dimensional test functions, and its advantage is more obvious as the dimensionality of the optimization problem increases. In both 10-dimensional and 30-dimensional test functions, NSABC ranks second in overall performance, and IATTP, ICSA, and NCCLA are slightly worse than NSABC in both dimensions. In the 100-dimensional test function, the overall performance of NCCLA ranked second, and NSABC, ICSA, and IATTP were slightly worse than NCCLA in this dimension. The overall performance of MSCA was the worst among the six algorithms. In summary, compared with other algorithms, the INCCLA proposed in this paper has certain advantages in terms of convergence precision.

TABLE 13: Data result of INCCLA with other algorithms on the CEC2020 test suite.

Function	NSABC		MSCA		IATTP		ICSA		NCCLA		INCCLA	
	Mean (std)	<i>p</i> value (vs. INCCLA)	Mean (std)	<i>p</i> value (vs. INCCLA)	Mean (std)	<i>p</i> value (vs. INCCLA)	Mean (std)	<i>p</i> value (vs. INCCLA)	Mean (std)	<i>p</i> value (vs. INCCLA)	Mean (std)	<i>p</i> value (vs. INCCLA)
F1	1.53E+03(1.89E+03)	0.000 (-)	1.42E+08 (9.70E+07)	0.000 (-)	2.66E+04 (1.66E+04)	0.000 (-)	4.17E+04 (1.28E+05)	0.000 (-)	3.25E+03 (2.75E+03)	0.000 (-)	2.38E-01 (5.98E-01)	
F2	7.31E+01 (6.93E+01)	0.739 (=)	1.15E+03 (2.35E+02)	0.000 (-)	1.17E+03 (1.42E+02)	0.000 (-)	1.71E+02 (1.47E+02)	0.145 (=)	3.23E+02 (2.38E+02)	0.000 (-)	9.33E+01(8.63E+01)	
F3	1.25E+01 (1.87E+00)	0.000 (+)	5.63E+01 (9.38E+00)	0.000 (-)	4.01E+01 (4.60E+00)	0.000 (-)	1.53E+01 (2.38E+00)	0.112 (=)	3.12E+01 (7.79E+00)	0.000 (-)	1.63E+01(3.44E+00)	
F4	6.49E-01(1.19E-01)	0.684 (=)	7.65E+00 (2.43E+00)	0.000 (-)	2.03E+00 (3.89E-01)	0.000 (-)	5.21E+00 (1.86E+01)	0.008 (-)	1.28E+00 (5.88E-01)	0.000 (-)	6.20E-01 (2.53E-01)	
F5	3.53E+04 (2.75E+04)	0.000 (-)	9.21E+03 (6.13E+03)	0.000 (-)	4.94E+02 (2.68E+02)	0.000 (-)	4.68E+03 (1.74E+04)	0.000 (-)	6.71E+03 (1.07E+04)	0.000 (-)	1.34E+02 (1.04E+02)	
F6	2.03E+01 (4.13E+01)	0.994 (=)	1.21E+02 (5.85E+01)	0.000 (-)	4.16E+01 (4.74E+01)	0.011 (+)	4.63E+01 (6.54E+01)	0.008 (-)	1.17E+02 (8.56E+01)	0.000 (-)	4.53E+01(6.66E+01)	
F7	4.11E+03 (3.95E+03)	0.000 (-)	2.83E+03 (1.39E+03)	0.000 (-)	1.42E+02 (6.70E+01)	0.000 (-)	1.02E+02 (8.29E+01)	0.000 (-)	4.84E+03 (5.59E+03)	0.000 (-)	2.86E+01 (3.67E+01)	
F8	1.07E+02 (1.49E+01)	0.570 (=)	1.21E+02 (6.45E+00)	0.000 (-)	1.07E+02 (1.52E+01)	0.000 (-)	1.10E+02 (2.20E-01)	0.000 (-)	1.10E+02 (1.10E-04)	0.000 (-)	1.04E+02 (2.43E+01)	
F9	2.41E+02(1.10E+02)	0.000 (+)	3.74E+02 (6.78E+00)	0.000 (-)	2.08E+02 (1.18E+02)	0.007 (+)	3.13E+02 (7.19E+01)	0.013 (+)	3.47E+02 (6.53E+00)	0.001 (-)	3.18E+02 (7.44E+01)	
F10	4.15E+02 (2.27E+01)	0.100 (=)	4.33E+02 (2.81E+01)	0.000 (-)	4.17E+02 (2.25E+01)	0.610 (=)	4.31E+02 (2.23E+01)	0.000 (-)	4.18E+02 (2.33E+01)	0.085 (=)	4.14E+02 (2.21E+01)	
+/-	2/5/3		0/0/10		2/1/7		1/2/7		0/1/9			
Avg.rank	2.45		5.60		3.25		3.65		4.25		1.80	
Sort	2		6		3		4		5		1	

(2) *Testing at CEC 2020.* To further examine the performance of the INCCLA algorithm, the INCCLA algorithm and four other algorithms are tested on the 10-dimensional CEC2020 test suite. The parameters of each algorithm were set as above. Table 13 counts the results of 30 independent experiments for each algorithm, including the mean and standard deviation, the results of the Wilcoxon rank sum test results for each algorithm with a significance level of 5% with the INCCLA algorithm, the number of functions for which each algorithm is significantly better than INCCLA, significantly worse than INCCLA, and not significantly different from INCCLA, and the results of the Friedman test and ranking results for each algorithm.

As shown in Table 13, for the 10-dimensional CEC2020 test function set, compared with NCCLA, INCCLA showed similar performance only on $F10$, achieved better mean values on the remaining 9 test functions, and performed significantly better on them. Compared with NSABC, INCCLA performs significantly inferior only on $F9$ and $F3$, achieves similar performance on 5 test functions, and performs significantly better on 3 test functions; compared to MSCA, INCCLA achieves better mean values on all 10 tested functions as well as showing significantly better performance. Compared to IATTP, INCCLA only showed inferior performance on $F6$ and $F9$, similar performance on $F10$, and significantly better performance on 7 test functions; compared to ICSA, INCCLA only showed inferior performance on $F9$, similar performance on $F2$ and $F3$, and significantly better performance on 7 test functions. It is obvious from the rank sum calibration results that INCCLA has an advantage over several other comparative algorithms on the CEC2020 test suite.

In summary, compared with NCCLA and the other four superior optimization algorithms, INCCLA shows a significant advantage in convergence precision. The advantages become more obvious as the dimension of the optimization problem increases.

4.3.2. Comparison of INCCLA with Other Algorithms on Convergence Speed. In order to compare the differences in convergence speed among the algorithms more intuitively, Figure 4 shows the convergence curves of each algorithm on the 28 test functions of the CEC2013 test suite when the dimension is 30, where the horizontal coordinate is the number of function evaluations and the vertical coordinate is the logarithm of the fitness value.

As shown in Figure 4, for the unimodal functions $F1\sim F5$, INCCLA can obtain the global optimum on $F1$, $F3$, and $F5$ test functions, but the convergence speed is slightly slower than NSABC on $F1$ and $F5$ test functions; INCCLA shows better convergence precision and the fastest convergence speed than the other five algorithms on $F2$; INCCLA is second only to IATTP in terms of convergence precision and convergence speed on $F4$. For multimodal functions $F6\sim F20$, INCCLA can obtain the global optimum on $F7$, $F9$, $F11$, $F12$, $F13$, and $F20$ test functions, and the convergence speed is second only to MSCA on $F11$; INCCLA shows better convergence precision and the fastest convergence speed

than the other five algorithms on $F6$, $F10$, and $F19$. INCCLA shows better convergence precision and relatively better convergence speed on $F8$, $F14$, $F15$, $F16$, $F17$, and $F18$ test functions. In the early evolutionary stage, the convergence speed of INCCLA is slightly slower than the other algorithms. When it reaches the late evolutionary stage, INCCLA can continue searching for better solutions compared to the reduced search performance of the other algorithms. For the composition functions $F21\sim F28$, INCCLA shows better convergence precision and fastest convergence speed than the other five algorithms on the test functions $F25$, $F27$, and $F28$; INCCLA shows higher convergence precision and relatively faster convergence speed on $F22$ and $F23$. The convergence speed of INCCLA on $F21$ is similar to that of the three algorithms NSABC, NCCLA and IATTP, where the convergence precision of INCCLA and NSABC is slightly better than that of the other algorithms; the convergence speed of INCCLA on $F24$ is similar to that of the four algorithms NCCLA, ICSA, IATTP, and NSABC, with the convergence precision of INCCLA being higher. On $F26$, the convergence speed and precision of INCCLA are similar to those of ICSA and IATTP, and the convergence precision of INCCLA is second only to that of NSABC. In summary, it shows that INCCLA has some advantages in terms of convergence speed compared with NCCLA and the other four superior optimization algorithms.

In summary, compared with NCCLA and the other four superior optimization algorithms, the INCCLA proposed in this paper is superior in the overall performance in terms of convergence precision and convergence speed, although it has the shortcoming of not converging fast enough in the early stage for individual complex functions. The essential reasons for this are all caused by the three improved evolutionary strategies in Sections 3.1–3.3 of this paper, which are analyzed in depth as follows. The update strategy of the INCCLA algorithm has a large step size in the early evolutionary stage, which makes individuals search a farther distance around themselves, thus ensuring the global search of the algorithm and reducing the possibility of the algorithm falling into a local optimum, for more complex optimization problems, which also inevitably slows down the rapid aggregation of the population around a dominant region. Therefore, it makes INCCLA not converge fast enough in the early stage on individual complex functions. However, as the iteration proceeds, the individual search step size gradually decreases, and it is easier to search to have the advantage of the area, easy to refine the search, which improves the convergence speed in the late evolutionary stage and facilitates the algorithm to obtain a more excellent convergence precision. In addition, the parental selection mechanism can guide the evolutionary direction of juveniles during the learning phase, allowing the algorithm to increase algorithmic diversity while maintaining convergence. The juvenile crow hybrid learning mechanism can determine the state of the algorithm in the current evolutionary stage according to the individual's attributes and select the learning mode in a targeted manner so that the algorithm can effectively maintain the balance of convergence speed and diversity in the evolutionary process. The interplay and

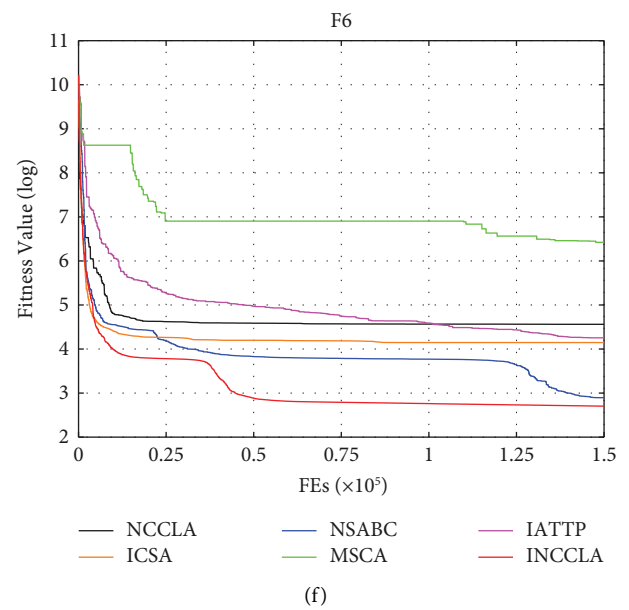
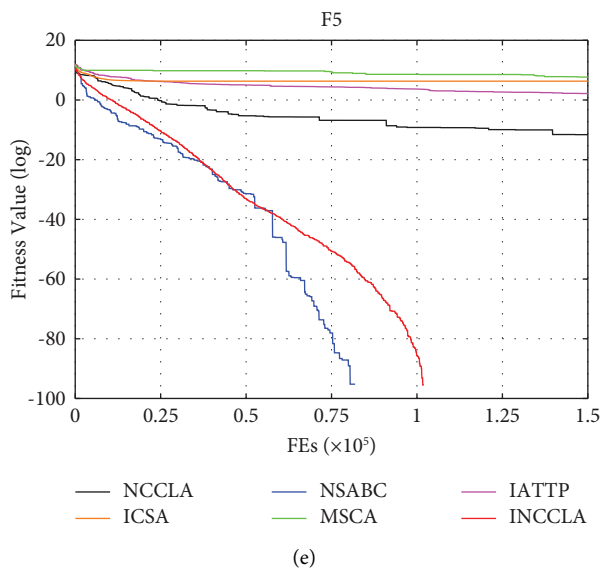
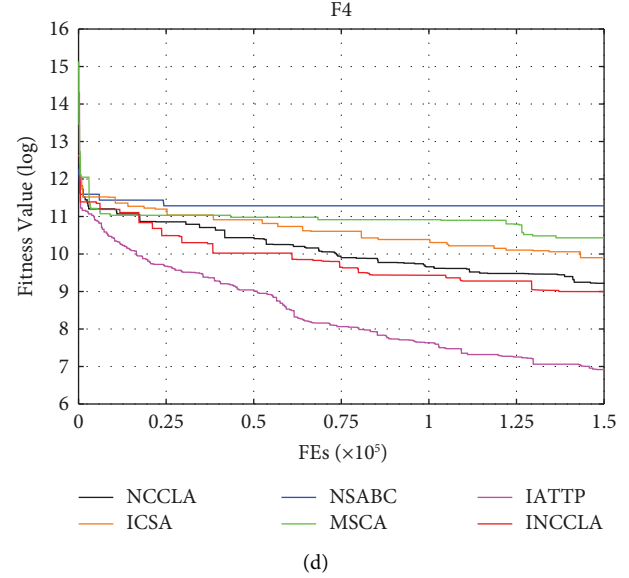
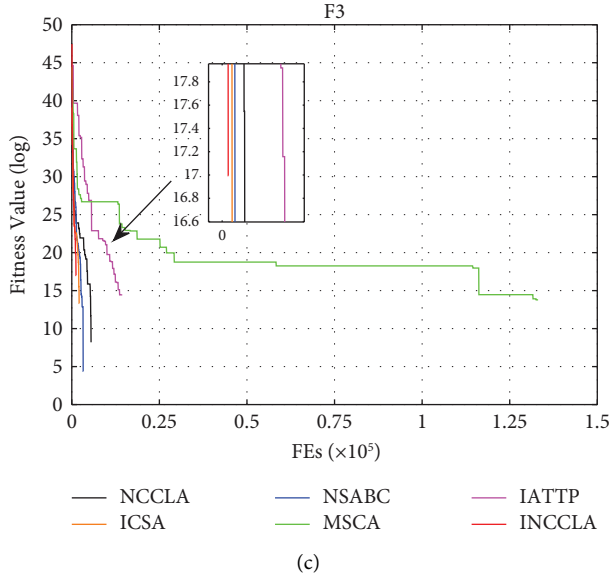
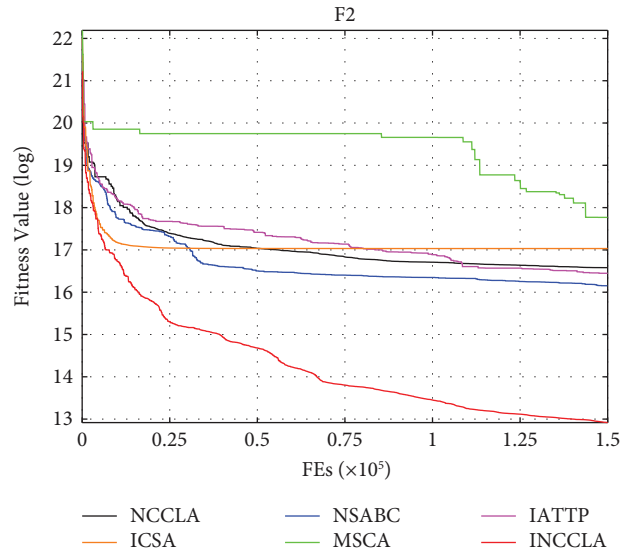
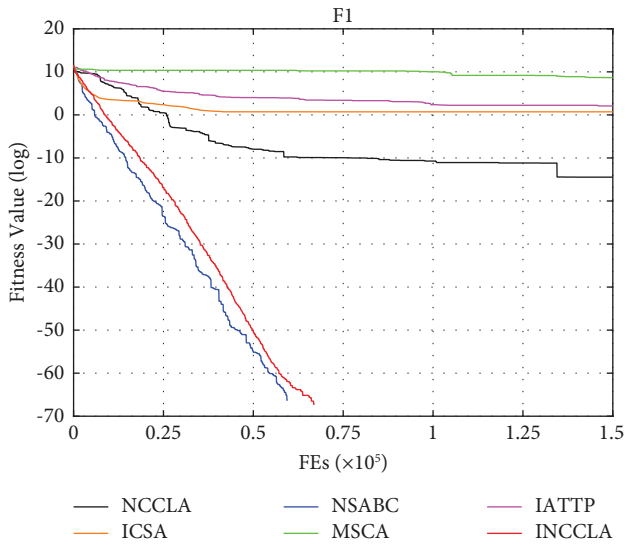


FIGURE 4: Continued.

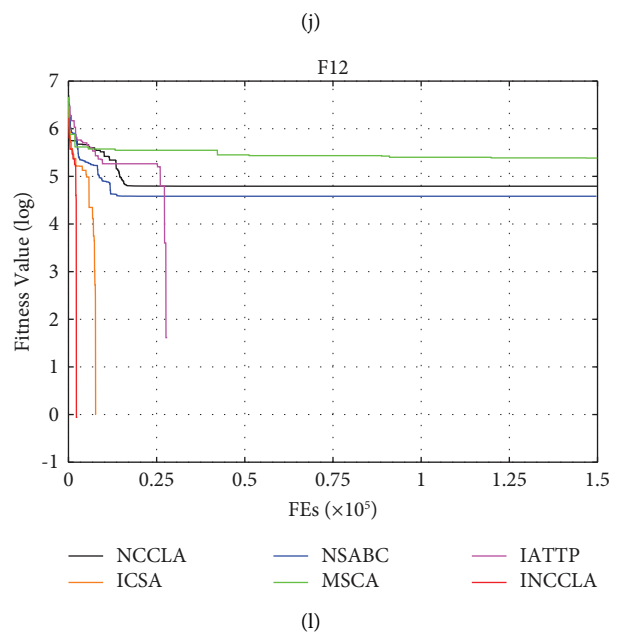
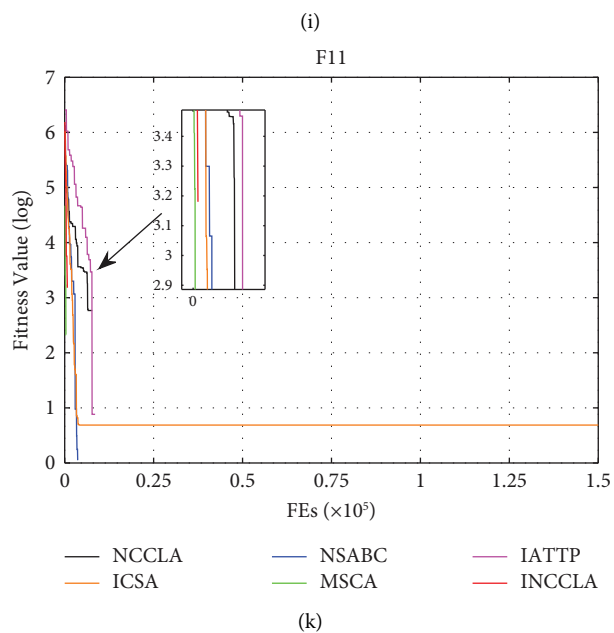
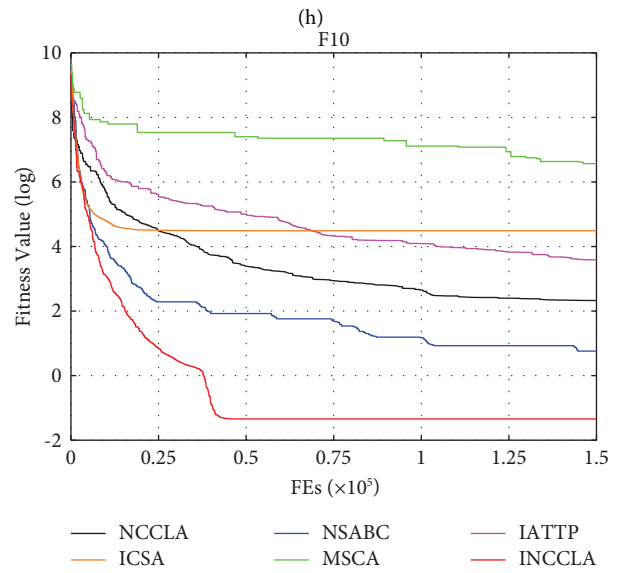
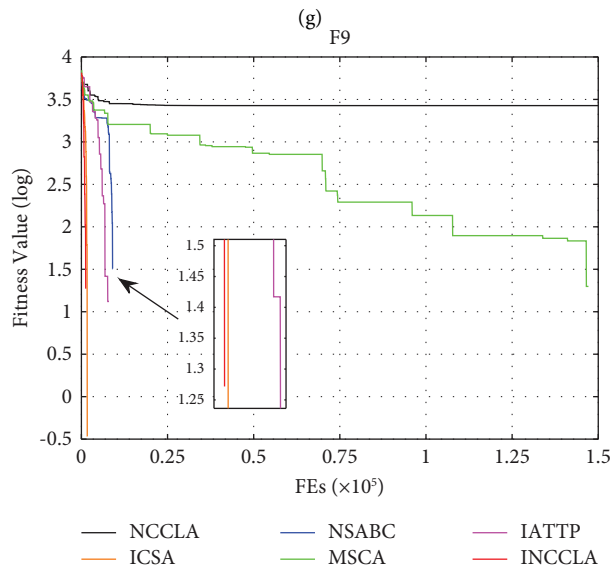
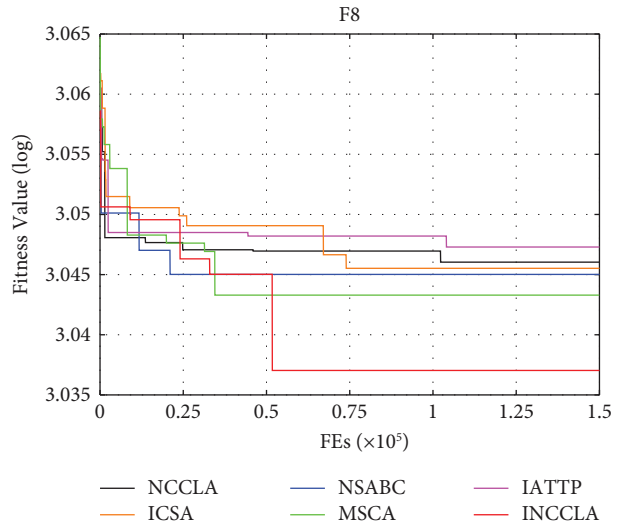
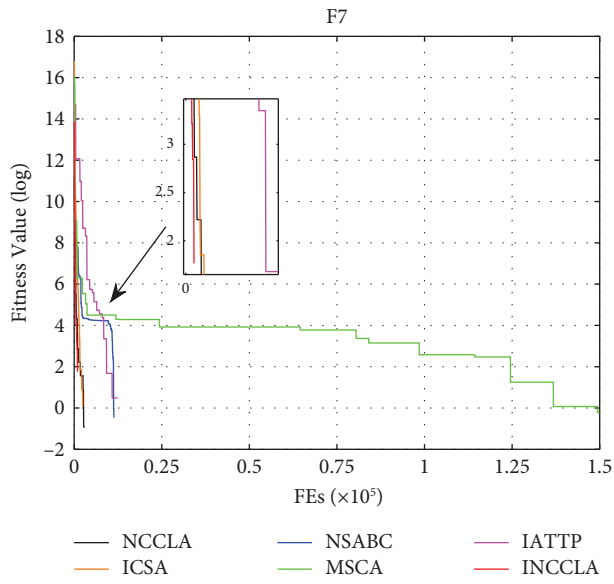
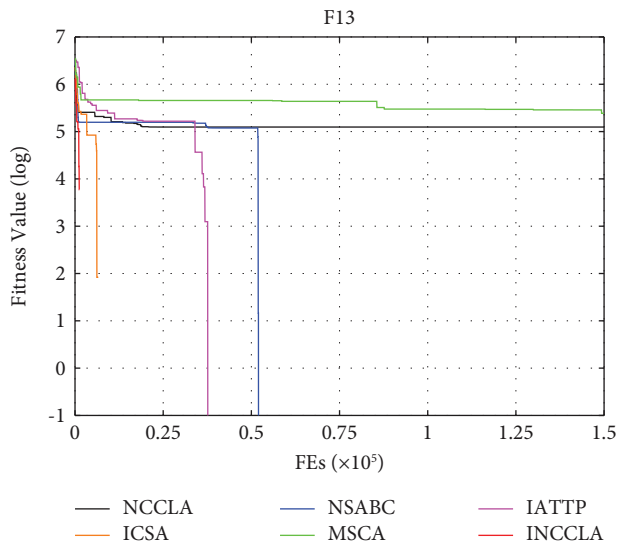
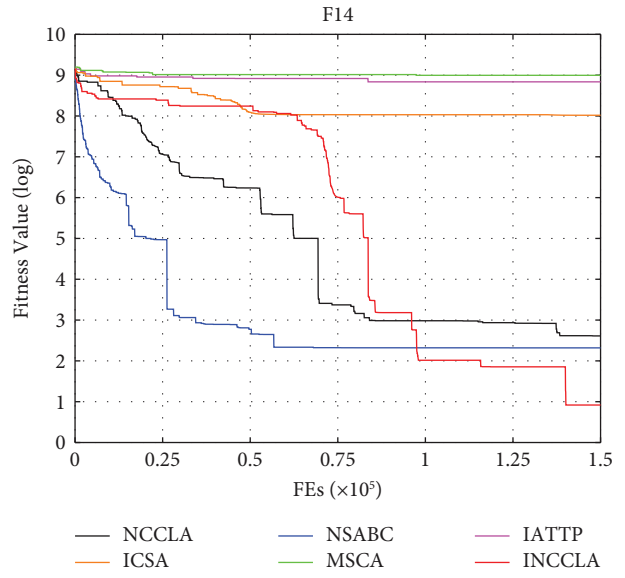


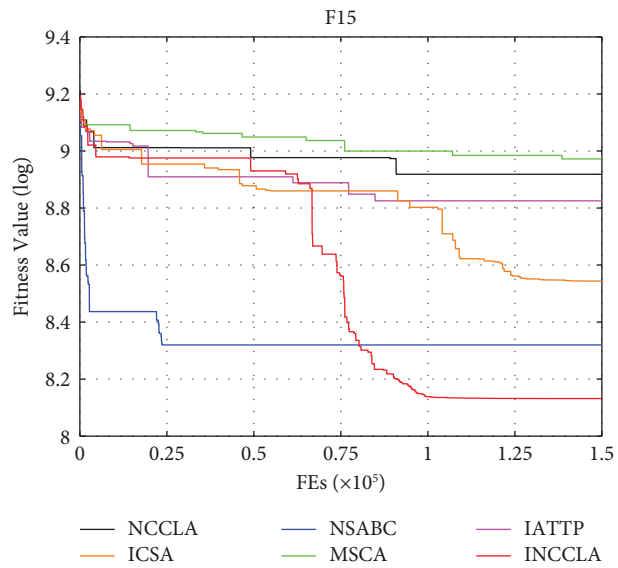
FIGURE 4: Continued.



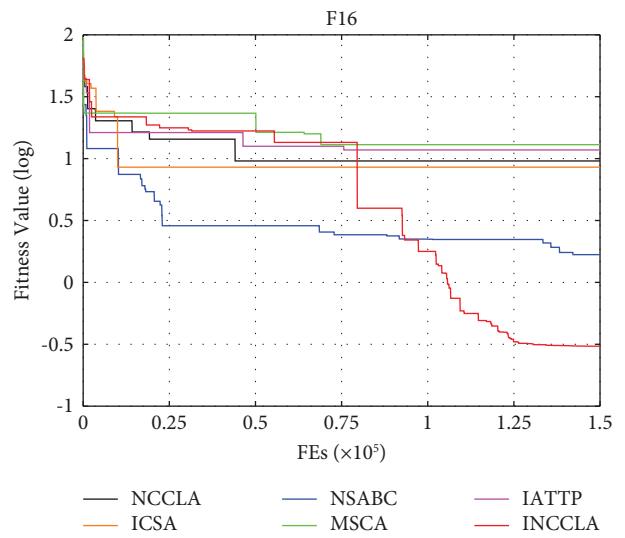
(m)



(n)

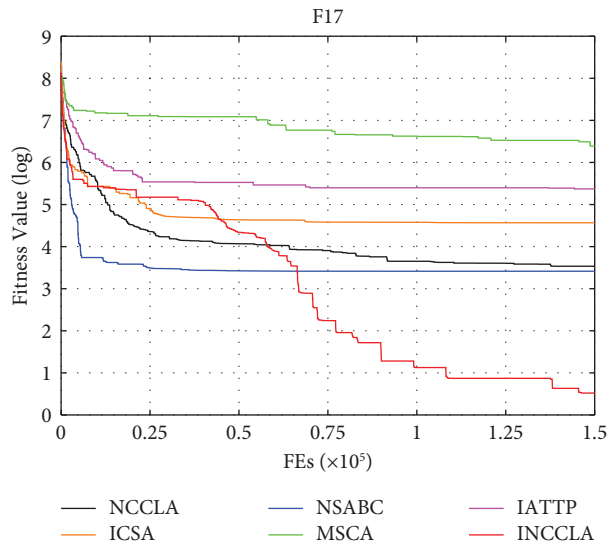


(o)

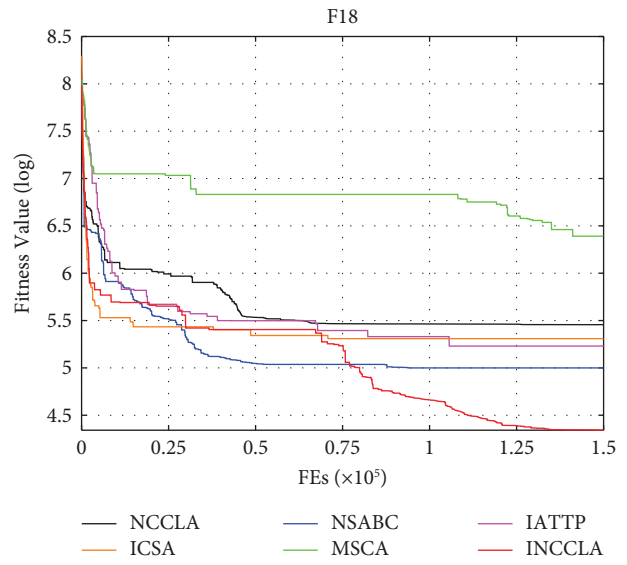


(p)

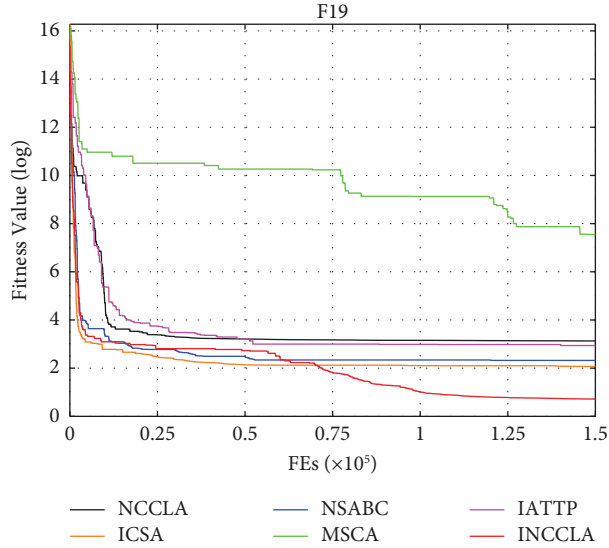
FIGURE 4: Continued.



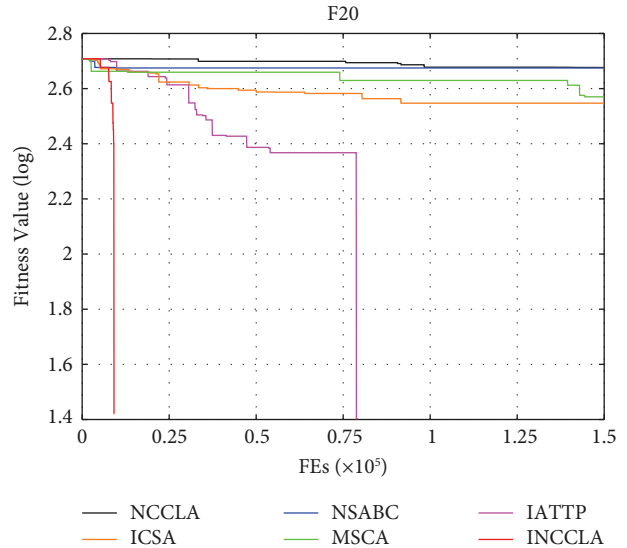
(q)



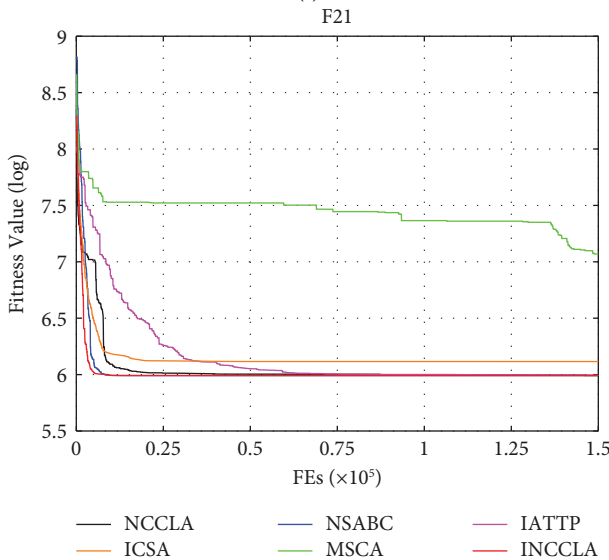
(r)



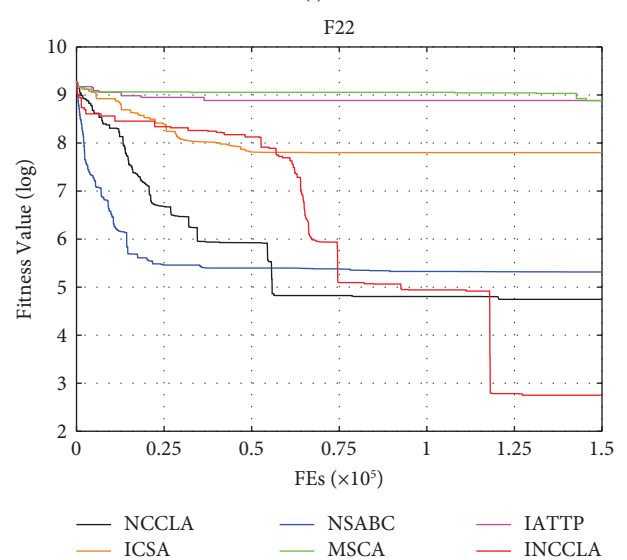
(s)



(t)



(u)



(v)

FIGURE 4: Continued.

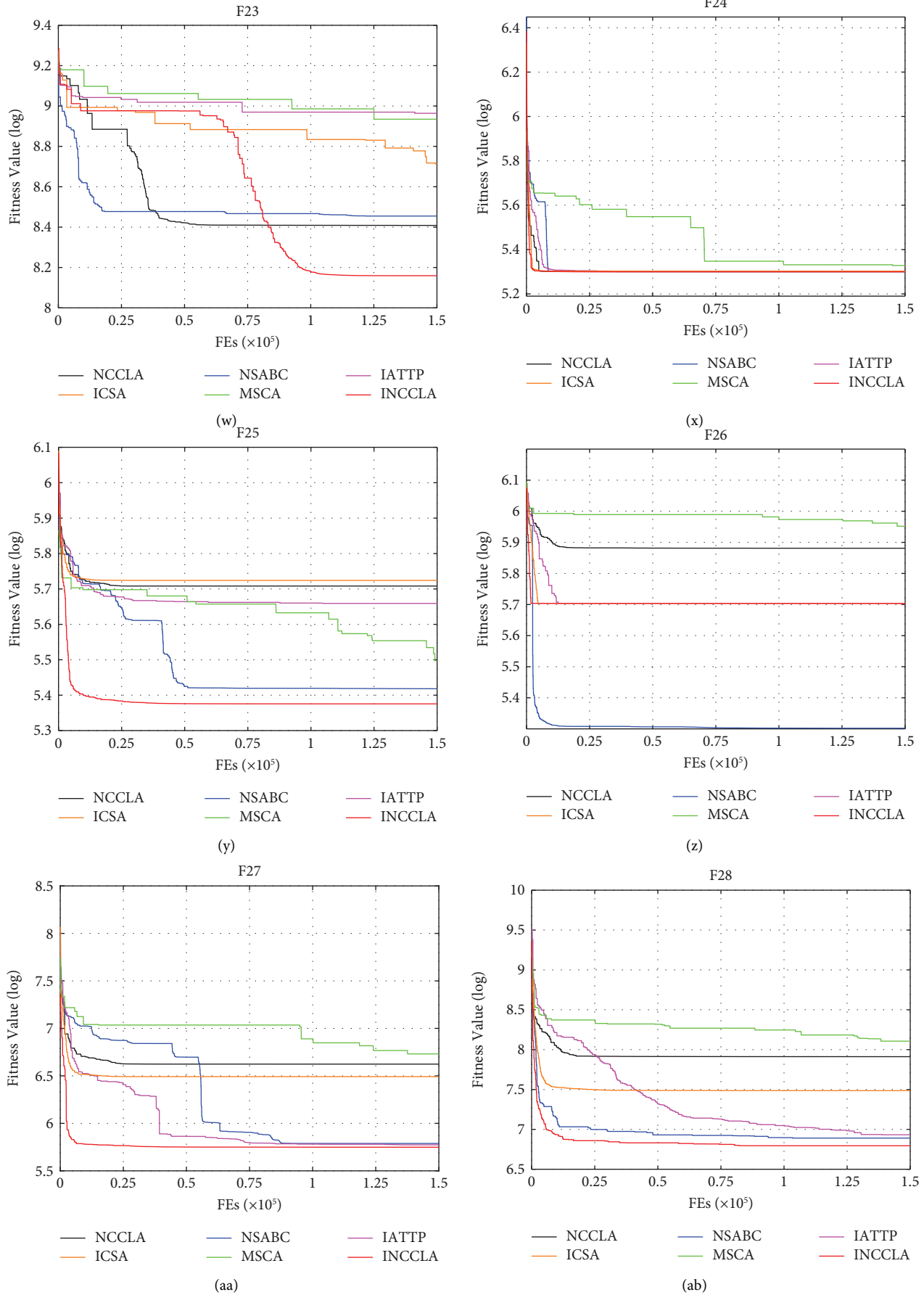


FIGURE 4: Convergence curves of each algorithm on the CEC2013 test suite. (a)-(ab) denote the 28 test functions $F1 \sim F28$ in the CEC2013 test set, respectively.

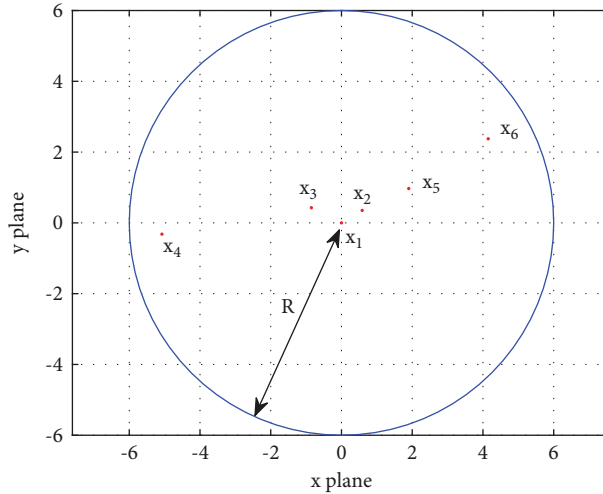


FIGURE 5: Distribution of synergy nodes.

influence of these three mechanisms ensures the overall advantage of INCCLA in terms of convergence precision and convergence speed. In the next step, we will work on further improving the convergence speed of the INCCLA algorithm in the pre-evolutionary stage while ensuring no significant change in the convergence precision.

4.4. Comparison of the Effects of Engineering Applications. To further compare the effectiveness of INCCLA with each comparison algorithm for engineering applications, this section will be validated by dealing with the collaborative beamforming optimization problem. The collaborative beamforming optimization problem is a typical problem in antenna arrays. The amplitude $\xi \in [0, 1]$ and phase $\alpha \in [-\pi, \pi]$ of the transmit signal weights of each collaborative node are used as decision variables in the algorithm optimization process, and the peak side valve level PSL minimization as shown in (27) is achieved by algorithm optimization.

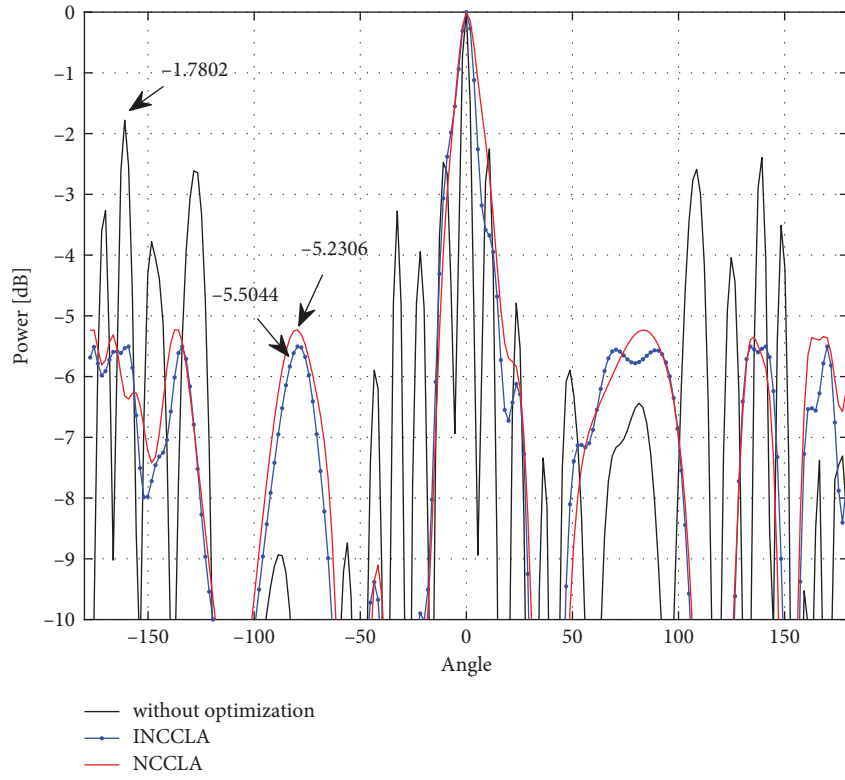
$$PSL = 20 \log_{10} \frac{\max |AF(\theta_{SL}, w)|}{AF(\phi, w)}, \quad (27)$$

where, $AF(\theta, w)$ represents the array factor, as shown in (28) and ϕ is the main beam direction. The positions of θ_{SL} can be found by finding all the peak points of the array factor (other than the main lobe's peak) for the domain $\theta \in [-\pi, \phi) \cup (\phi, \pi]$, the denominator $AF(\phi, w)$ is the main beam power, which can be calculated as described in the literature [39], and molecule $\max |AF(\theta_{SL}, w)|$ is the maximum beam power in the side flap.

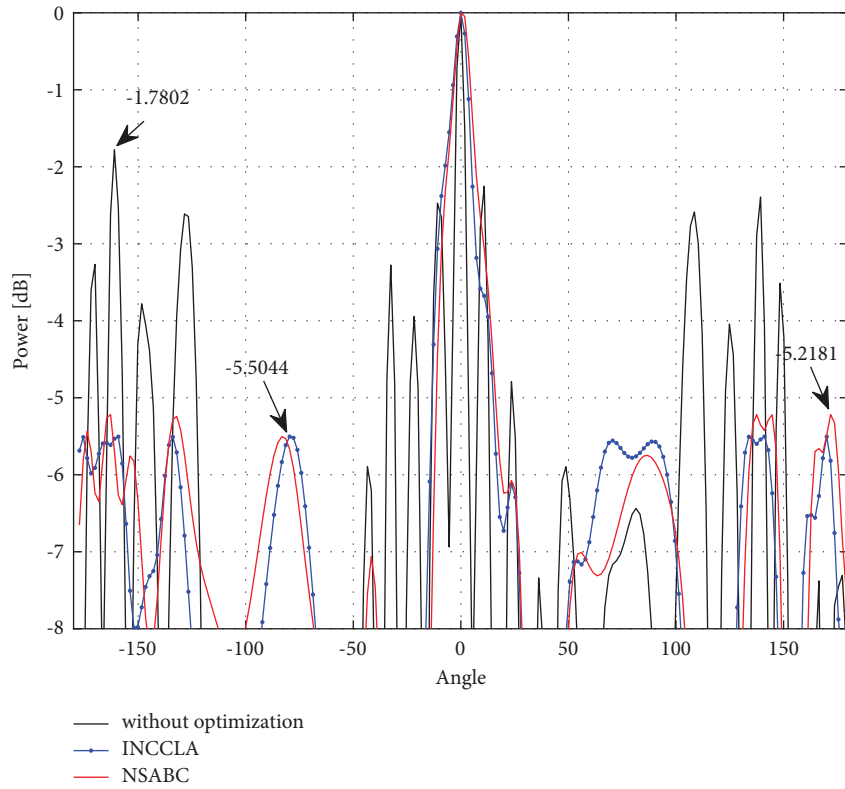
$$AF(\theta, w) = \sum_{k=1}^K w_k e^{j(2\pi/\lambda)r_k [\cos(\theta - \psi_k)]}. \quad (28)$$

This section examines the practical engineering optimization of INCCLA and the other comparative algorithms by optimizing the collaborative beamforming problem shown in Figure 5, in which the wavelength of the transmitting signal is λ . The six synergistic nodes are distributed in a circular domain of radius 6λ , where one synergistic node is located in the center of the circular domain.

The parameter settings of each algorithm are shown in Table 7. In order to avoid adverse effects of chance on algorithm evaluation, each algorithm was run 10 times

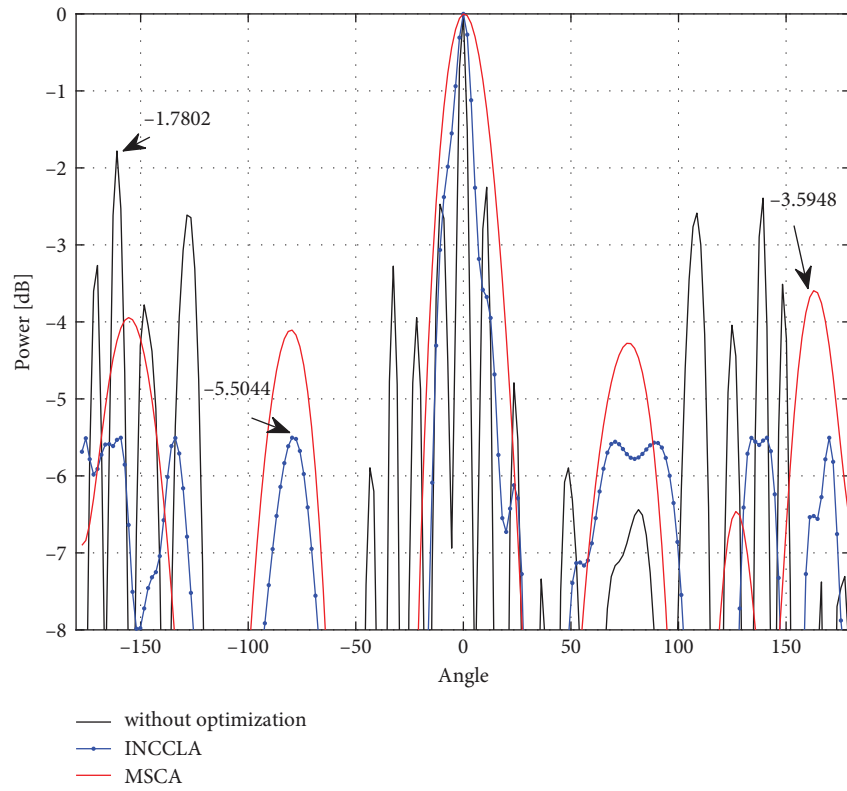


(a)

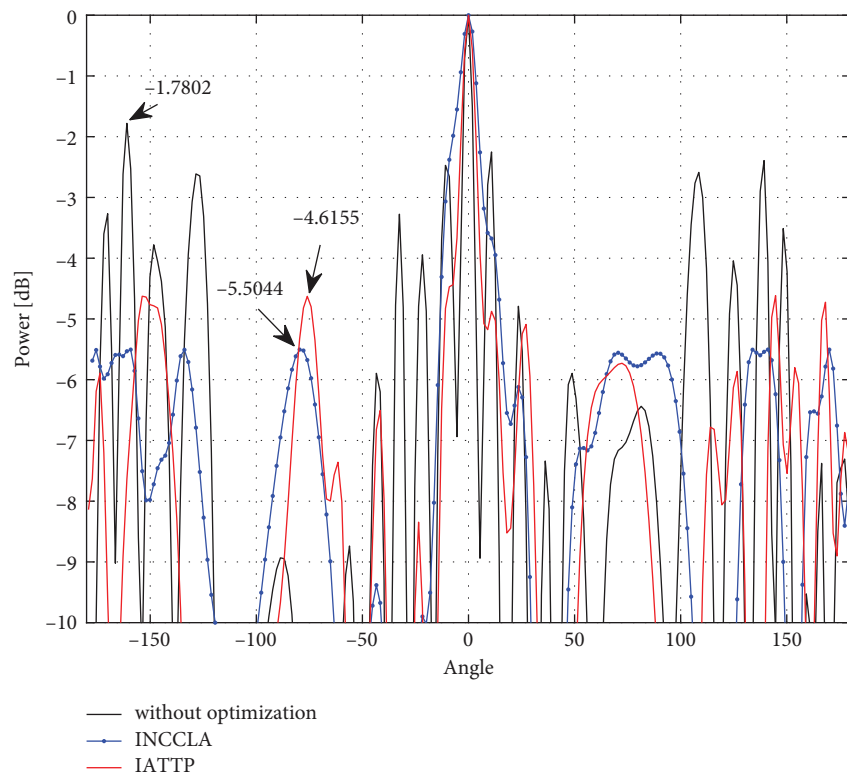


(b)

FIGURE 6: Continued.



(c)



(d)

FIGURE 6: Continued.

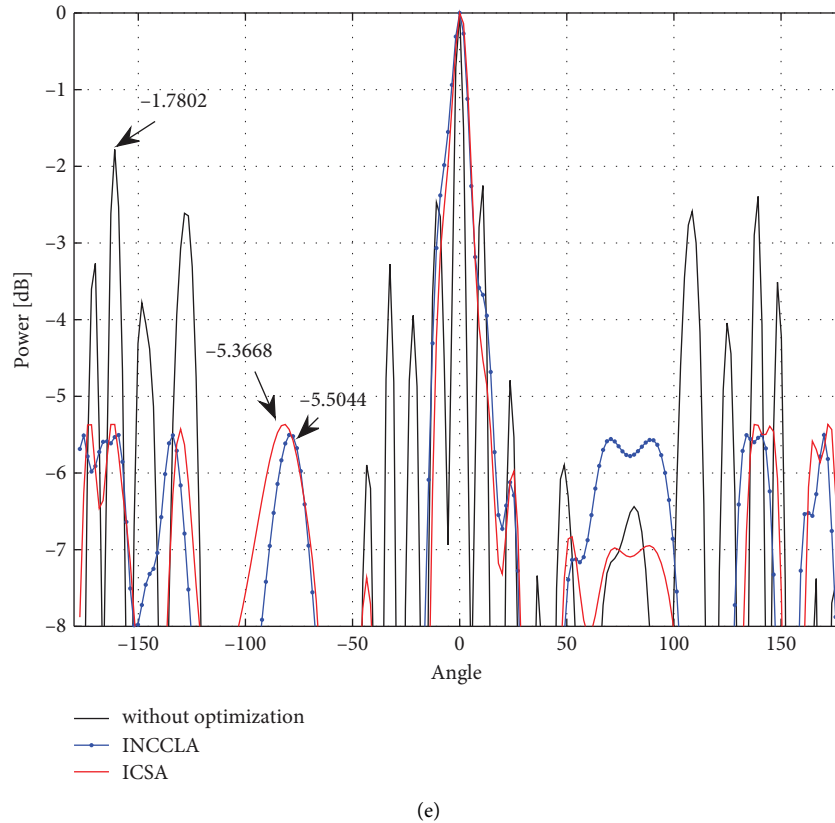


FIGURE 6: Beam diagram of INCCLA and each comparison algorithm in the cartesian coordinate system. (a)–(e) are the beam diagram of INCCLA with NCCLA, NSABC, MSCA, IATTP, and ICSA, respectively.

independently. The best median PSL obtained by each algorithm corresponding to the collaborative beam optimization scheme was selected for comparison. The beam diagram of INCCLA with each comparison algorithm in the Cartesian coordinate system is visually presented in Figure 6. The PSL obtained by each algorithm is labeled within the figure.

According to Figure 6, it can be seen that each algorithm achieves better collaborative beam optimization compared to the unoptimized algorithm, where the best PSL values obtained by INCCLA, NCCLA, NSABC, MSCA, IATTP, and ICSA are -5.5044 dB, -5.2306 dB, -5.2181 dB, -3.5948 dB, -4.1655 dB, and -5.3668 dB, respectively. Compared with the other five algorithms, the best PSL value obtained by INCCLA is the smallest and achieves the best collaborative beam optimization among the five algorithms. In summary, the proposed INCCLA also performs better in engineering applications.

5. Conclusions

This paper proposes an improved New Caledonian Crow Learning Algorithm (INCCLA) to improve further the convergence performance and the ability to escape from the local optimum of the NCCLA algorithm. First, INCCLA introduces cosine similarity in the parent selection phase. The selected parent can guide the juvenile crow to exploit different regions, maintaining the balance between population diversity and the convergence speed of the algorithm

in the evolutionary process. Second, INCCLA sets up a mixed learning mechanism of complete learning and incomplete learning, with the complete learning phase accelerating individual convergence and improving the convergence speed of the algorithm, and the incomplete learning phase increasing population diversity and enhancing the ability of the algorithm to escape from the local optimum. Finally, the juvenile crow reinforcement phase introduces weight factors and random perturbations to increase the global search and local exploration ability of the algorithm in the evolutionary process. The proposed parent reinforcement phase enhances the individual search capability of the parent and improves the overall performance of the algorithm. Experimental results on the CEC2013 and CEC2020 test suites show that the improvement strategy proposed in this paper effectively improves the overall performance of INCCLA, enabling the algorithm to maintain a balance between convergence speed and population diversity during the evolutionary process, and can achieve better results in most of the test functions. In addition, INCCLA is also more competitive in unimodal, multimodal, and composition problems compared with the other four optimization algorithms. INCCLA is also applied to the collaborative beamforming optimization problem, demonstrating the usefulness of INCCLA in engineering applications.

Although the INCCLA algorithm proposed in this paper has obvious advantages in terms of convergence accuracy, its

time complexity is slightly higher. In future work, the time complexity of the algorithm needs to be further reduced. In addition, it further broadens the application of the INCCLA algorithm in practical engineering fields.

Data Availability

The labeled datasets used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

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

References

- [1] S. Aslan, H. Karaboga, and D. Karaboga, "Improved quick artificial bee colony (iqABC) algorithm for global optimization," *Soft Computing*, vol. 23, no. 24, pp. 13161–13182, 2019.
- [2] S. Shekhawat and A. Saxena, "Development and applications of an intelligent crow search algorithm based on opposition based learning," *ISA Transactions*, vol. 99, pp. 210–230, 2020.
- [3] D. Y. Tang, S. B. Dong, X. F. Cai, and J. Zhao, "A two-stage quantum-behaved particle swarm optimization with skipping search rule and weight to solve continuous optimization problem," *Neural Computing & Applications*, vol. 27, no. 8, pp. 2429–2440, 2016.
- [4] J. S. Liu, Y. N. Mao, X. Z. Liu, and Y. Li, "A dynamic adaptive firefly algorithm with globally orientation," *Mathematics and Computers in Simulation*, vol. 174, pp. 76–101, 2020.
- [5] A. Faramarzi, M. Heidarinejad, and S. Mirjalili, "Marine predators algorithm: a nature-inspired metaheuristic," *Expert Systems with Applications*, vol. 152, Article ID 113377, 2020.
- [6] W. G. Zhao, Z. X. Zhang, and L. Y. Wang, "Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications," *Engineering Applications of Artificial Intelligence*, vol. 87, Article ID 103300, 2020.
- [7] Y. J. Wang, P. Sun, X. M. Shi, and L. Zhang, "Green deployment method of micro base station for ultra-dense heterogeneous cellular networks based on constrained dolphin swarm algorithm," *IEEE Access*, vol. 8, pp. 5349–5362, 2020.
- [8] S. Gupta and K. Deep, "A novel random walk grey wolf optimizer," *Swarm and Evolutionary Computation*, vol. 44, pp. 101–112, 2019.
- [9] A. Sayed, M. Ebeed, Z. M. Ali et al., "A hybrid optimization algorithm for solving of the unit commitment problem considering uncertainty of the load demand," *Energies*, vol. 14, no. 23, p. 8014, Article ID 8014, 2022.
- [10] N. Habib Khan, R. Jamal, M. Ebeed, S. Kamel, H. Zeinoddini-Meymand, and H. M. Zawbaa, "Adopting Scenario-Based approach to solve optimal reactive power Dispatch problem with integration of wind and solar energy using improved Marine predator algorithm," *Ain Shams Engineering Journal*, vol. 13, no. 5, p. 101726, Article ID 101726, 2022.
- [11] M. Bastawy, M. Ebeed, A. Ali, M. F. Shaaban, B. Khan, and S. Kamel, "Optimal day-ahead scheduling in micro-grid with renewable based DGs and smart charging station of EVs using an enhanced manta-ray foraging optimisation," *IET Renewable Power Generation*, vol. 16, no. 11, pp. 2413–2428, 2022.
- [12] H. Samma, C. P. Mohamad Saleh, and J. Lim, "A new reinforcement learning-based memetic particle swarm optimizer," *Applied Soft Computing*, vol. 43, pp. 276–297, 2016.
- [13] G. Zhang, Y. M. Li, and Y. H. Shi, "Distributed learning particle swarm optimizer for global optimization of multimodal problems," *Frontiers of Computer Science*, vol. 12, no. 1, pp. 122–134, 2018.
- [14] J. W. Lu, J. Zhang, and J. N. Sheng, "Enhanced multi-swarm cooperative particle swarm optimizer," *Swarm and Evolutionary Computation*, vol. 69, p. 100989, Article ID 100989, 2022.
- [15] F. Wang, X. J. Wang, and S. L. Sun, "A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization," *Information Sciences*, vol. 602, pp. 298–312, 2022.
- [16] L. Z. Cui, G. H. Li, Y. L. Luo et al., "An enhanced artificial bee colony algorithm with dual-population framework," *Swarm and Evolutionary Computation*, vol. 43, pp. 184–206, 2018.
- [17] M. A. Awadallah, M. A. Al-Betar, A. L. Bolaji, E. M. Alsukhni, and H. Al-Zoubi, "Natural selection methods for artificial bee colony with new versions of onlooker bee," *Soft Computing*, vol. 23, no. 15, pp. 6455–6494, 2019.
- [18] X. Y. Zhou, Y. L. Wu, M. S. Zhong, and M. W. Wang, "Artificial bee colony algorithm based on multiple neighborhood topologies," *Applied Soft Computing*, vol. 111, p. 107697, Article ID 107697, 2021.
- [19] T. Y. Ye, W. J. Wang, H. Wang et al., "Artificial bee colony algorithm with efficient search strategy based on random neighborhood structure," *Knowledge-Based Systems*, vol. 241, p. 108306, Article ID 108306, 2022.
- [20] H. Wang, W. J. Wang, X. Y. Zhou et al., "Firefly algorithm with neighborhood attraction," *Information Sciences*, vol. 382–383, pp. 374–387, 2017.
- [21] Y. J. Sun, X. L. Wang, Y. H. Chen, and Z. J. Liu, "A modified whale optimization algorithm for large-scale global optimization problems," *Expert Systems with Applications*, vol. 114, pp. 563–577, 2018.
- [22] H. Zamani, M. H. Nadimi-Shahraki, and A. H. Gandomi, "CCSA: conscious neighborhood-based crow search algorithm for solving global optimization problems," *Applied Soft Computing*, vol. 85, p. 105583, Article ID 105583, 2019.
- [23] S. Gupta and K. Deep, "A memory-based grey wolf optimizer for global optimization tasks," *Applied Soft Computing*, vol. 93, p. 106367, Article ID 106367, 2020.
- [24] W. Long, M. Xu, J. J. Jiao, T. B. Wu, M. Z. Tang, and S. H. Cai, "A velocity-based butterfly optimization algorithm for high-dimensional optimization and feature selection," *Expert Systems with Applications*, vol. 201, p. 117217, Article ID 117217, 2022.
- [25] G. G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2018.
- [26] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. L. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
- [27] S. M. Li, H. L. Chen, M. J. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.

- [28] W. Al-Sorori and A. M. Mohsen, "New Caledonian crow learning algorithm: a new metaheuristic algorithm for solving continuous optimization problems," *Applied Soft Computing*, vol. 92, Article ID 106325, 2020.
- [29] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 7, pp. 1501–1529, 2020.
- [30] J. Z. Tu, H. L. Chen, M. J. Wang, and A. H. Gandomi, "The colony predation algorithm," *Journal of Bionics Engineering*, vol. 18, no. 3, pp. 674–710, 2021.
- [31] F. A. Hashim and A. G. Hussien, "Snake Optimizer: a novel meta-heuristic optimization algorithm," *Knowledge-Based Systems*, vol. 242, p. 108320, Article ID 108320, 2022.
- [32] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions an evaluation criteria for the CEC2013 special session on real-parameter optimization," *Technical Report*, vol. 2012, 2013.
- [33] C. Yue, K. Price, and P. Suganthan, "Problem definitions and evaluation criteria for the CEC 2020 special session and competition on single objective bound constrained numerical optimization," *Zhengzhou Univ., Zhengzhou, China, Tech. Rep.*, Article ID 201912, 2019.
- [34] J. Derrac, S. García, D. Herrera, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
- [35] H. Wang, W. J. Wang, S. Y. Xiao, Z. H. Cui, M. Y. Xu, and X. Y. Zhou, "Improving artificial Bee colony algorithm using a new neighborhood selection mechanism," *Information Sciences*, vol. 527, pp. 227–240, 2020.
- [36] S. Gupta, K. Deep, S. Mirjalili, and J. H. Kim, "A modified sine cosine algorithm with novel transition parameter and mutation operator for global optimization," *Expert Systems with Applications*, vol. 154, p. 113395, Article ID 113395, 2020.
- [37] 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, p. 104324, Article ID 104324, 2021.
- [38] 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.
- [39] S. Jayaprakasam, S. K. A. Rahim, and C. Y. Leow, "PSOGSA-Explore: a new hybrid metaheuristic approach for beam-pattern optimization in collaborative beamforming," *Applied Soft Computing*, vol. 30, pp. 229–237, 2015.