

Research Article

An Improved Chicken Swarm Optimization Algorithm for Solving Multimodal Optimization Problems

Jianhui Liang ^{1,2}, Lifang Wang ¹ and Miao Ma ³

¹School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China

²School of Applied Science and Technology, Hainan University, Dan Zhou 571737, China

³School of Computer Science, Shaanxi Normal University, Xi'an 710062, China

Correspondence should be addressed to Lifang Wang; wanglf@nwpu.edu.cn and Miao Ma; mmthp@snnu.edu.cn

Received 24 August 2022; Revised 12 October 2022; Accepted 28 October 2022; Published 22 November 2022

Academic Editor: Miaolei Zhou

Copyright © 2022 Jianhui Liang 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.

To solve the premature convergence problem of the standard chicken swarm optimization (CSO) algorithm in dealing with multimodal optimization problems, an improved chicken swarm optimization (ICSO) algorithm is proposed by referring to the ideas of bacterial foraging algorithm (BFA) and particle swarm optimization (PSO) algorithm. First, in order to improve the depth search ability of the algorithm, considering that the chicks have the weakest optimization ability in the whole chicken swarm, the replication operation of BFA is introduced. In the role update process of the chicken swarm, the chicks are replaced by the same number of chickens with the strongest optimization ability. Then, to maintain the population diversity, the elimination-dispersal operation is introduced to disperse the chicks that have performed the replication operation to any position in the search space according to a certain probability. Finally, the PSO algorithm is integrated to improve the global optimization ability of the algorithm. The experimental results on the CEC2014 benchmark function test suite show that the proposed algorithm has good performance in most test functions, and its optimization accuracy and convergence performance are also better than BFA, artificial fish swarm algorithm (AFSA), genetic algorithm (GA), and PSO algorithm, etc. In addition, the ICSO is also utilized to solve the welded beam design problem, and the experimental results indicate that the proposed algorithm has obvious advantages over other comparison algorithms. Its disadvantage is that it is not suitable for dealing with large-scale optimization problems.

1. Introduction

The multimodal optimization problem is a complex function optimization problem with one or more local extrema [1]. In practical applications, there are many multimodal optimization problems, such as parameter estimation and identification of models [2, 3], engineering structure optimization, welded beam design [4], and medical diagnosis [5]. It is difficult to find the global optimum because there are many local extrema in the multimodal optimization problem. Therefore, it is of very importance to study efficient and reasonable algorithms [6, 7].

The swarm intelligence optimization algorithm is a kind of bionic random search algorithm which can solve complex optimization problems by imitating the ecosystem mechanism in nature. Because of its strong global search ability,

high efficiency, and insensitivity to initial values, it has been widely concerned by relevant researchers [8–10]. At present, hundreds of algorithms based on swarm intelligence have emerged, such as genetic algorithm (GA) [11], particle swarm optimization (PSO) algorithm [12], bacterial foraging algorithm (BFA) [13], and artificial fish swarm algorithm (AFSA) [14]. In the swarm intelligence optimization algorithm, there are no individuals with centralized control, and the interaction between individuals is extremely simple, but they have the ability to solve complex problems in a short time, which makes them very suitable for solving complex optimization problems in practice. Moreover, the swarm intelligence optimization algorithm does not need the gradient information of the optimization problem, so it belongs to the nongradient optimization algorithm. Therefore, it has a wide range of applications [15, 16]. At

present, it has been applied in optimization calculation [17, 18], workshop scheduling [19, 20], image engineering [21], network structure optimization [22], vehicle routing problem [23], control of teleoperating systems, and other fields [24].

As a kind of swarm intelligent algorithm, the chicken swarm optimization (CSO) algorithm was proposed by Meng et al. in 2014 [25]. Due to its good stability and global search ability, the algorithm has attracted extensive attention since it was proposed and has been successfully applied in some fields [26, 27]. Liang et al. combined the CSO algorithm with the pulse-coupled neural network for image segmentation. The experimental results show that this method has obvious advantages in convergence speed and segmentation accuracy [28]. Cristin et al. combined the CSO algorithm with the deep neural network for the classification of brain tumors and achieved good performance in terms of accuracy, specificity, sensitivity, and so on [29].

Although the CSO algorithm has shown good performance in solving many benchmark problems and practical problems, it also has some inherent shortcomings, such as premature convergence and falling into local extrema. Therefore, researchers have proposed many improved algorithms. At present, the improvement of the CSO algorithm can be mainly classified into three categories as follows:

- (1) *Combination with other swarm intelligent algorithms.* For example, Li et al. improved the search ability of the CSO algorithm by integrating the operation of gray wolf optimization algorithm and PSO algorithm into the CSO algorithm. The experimental results show that the algorithm is superior to other basic swarm intelligent algorithms in accuracy, convergence speed [30], etc. Sampath fine-tuned the solution of the CSO algorithm by introducing the differential evolution algorithm to avoid premature convergence and applied the proposed algorithm to solve routing problems [31].
- (2) *Modification of position update formulas of the chicken swarm.* For example, Gu et al. improved the CSO algorithm by introducing an adaptive update factor and designing the inertia weight and successfully applied it to parameter estimation of the Richards model [32]. Considering that the imbalance between the diversity and intensification of the population may affect the performance of the CSO algorithm, Lin et al. proposed an improved CSO algorithm by modifying the position update formulas of the roosters and hens. Experiments show that this algorithm has obvious advantages over other swarm intelligent algorithms in terms of optimization accuracy and robustness [33].
- (3) *Multiobjective CSO algorithm.* In order to solve the configuration problem of electric vehicle charging stations, Deb et al. proposed a new hybrid multi-objective CSO algorithm based on pareto optimization. The effectiveness of the algorithm is verified

on some multiobjective benchmark problems and the electric vehicle charging station configuration problem [34].

The performance of the aforementioned swarm intelligence optimization algorithms has been improved to a certain extent, but there are still some disadvantages. For example, the literature [18] introduced the differential evolution strategy and quantum behavior into the bird swarm algorithm. Although the convergence speed of the algorithm was enhanced, the problem of premature convergence still existed. In the literature [30], several improved factors are introduced into the position update formulas of the chicken swarm, which improves the ability of the algorithm to jump out of the local extrema to a certain degree. But the optimization accuracy still needs to be further improved.

To solve the abovementioned problems, in this paper, an improved chicken swarm optimization (ICSO) algorithm is proposed which combines the idea of PSO with the replication and elimination-dispersal operations of BFA. More specifically, our contributions are as follows:

- (1) The replication operation of BFA is applied to the chicks with the weakest optimization ability to inherit the optimal food source in the whole chicken swarm, which is profitable to enhance the depth search ability of the algorithm.
- (2) The chicks are dispersed to any position in the search space according to a certain probability by using the elimination-dispersal operation of BFA, which is beneficial to improve the population diversity.
- (3) The idea of PSO is integrated to improve the global search ability of the algorithm. The experimental results on CEC2014 standard function test suite preliminarily show that this algorithm has obvious advantages over other swarm intelligent algorithms in optimization accuracy and convergence performance [35]. In addition, compared with other comparison algorithms, it also obtains competitive results in solving the welded beam design problem.

2. CSO Algorithm

The CSO algorithm is a meta-heuristic optimization algorithm that simulates the foraging behavior of roosters, hens, and chicks in nature. The characteristics of the algorithm are as follows:

- (1) *Correspondence with the optimization problem.* The algorithm regards several randomly generated positions in the solution space of the optimization problem as several chickens, and the food source of each chicken is measured by the fitness function value of the corresponding optimization problem.
- (2) *Hierarchical order, that is, the role assignment of the chicken swarm.* The whole chicken swarm is composed of four roles, namely, the roosters, hens, chicks, and mother hens. Their role assignment is

based on the content of food sources. The rooster has the best food sources, the hens take the second place, the food sources of the chicks are the worst, and the mother hens are randomly selected from the hens.

- (3) *Subgroup division.* During the whole foraging process, the chicken swarm is divided into several subgroups. The number of subgroups is determined by the number of the roosters, because each subgroup is composed of a randomly selected rooster, several hens, and chicks, and there is at least one hen in each subgroup.
- (4) *Relation of dependence.* In the foraging process of the chicken swarm, the chicks follow the chicks' mothers (mother hens) and the hens follow the roosters in their subgroups to forage for food. They can also steal the good food sources found by other subgroups at random.
- (5) *Information exchange.* The hierarchical order of the chicken swarm and the mother-child relationship between the mother hens and the chicks will be updated after several iterations. The information exchange between subgroups will be realized through continuous role assignment.
- (6) *Parallel optimization.* The whole chicken swarm can realize parallel optimization through the division of

labor and cooperation mechanism between subgroups, quickly find the best food sources, and then obtain the solution to the optimization problem. The formulas used by chickens with different roles in foraging are given below.

The formula used by the roosters in foraging is as follows:

$$X_{i,j}(t+1) = X_{i,j}(t) \times (1 + \text{Randn}(0, \sigma^2)) \quad j \in [1, \text{dim}], \quad (1)$$

$$\sigma^2 = \begin{cases} 1 & f_i \leq f_k \\ \exp\left(\frac{(f_k - f_i)}{|f_i| + \varepsilon}\right) & f_i > f_k \end{cases} \quad i, k \in [1, rNum], k \neq i, \quad (2)$$

where $X_{i,j}(t)$ represents the position of the i th rooster in the t -th iteration, and dim is the dimension of the optimization problem. $\text{Randn}(0, \sigma^2)$ is a normal distribution with a mean value of 0 and a standard deviation of σ^2 . ε is a very small number that can be expressed by the computer, which is used to avoid the situation that the denominator is 0 in the formula. f_k represents the content of the food source of any other rooster which is different from the i th rooster. $rNum$ represents the number of roosters.

The formula used by the hens in foraging is as follows:

$$X_{i,j}(t+1) = X_{i,j}(t) + c_1 \times \text{Rand} \times (X_{r_1,j}(t) - X_{i,j}(t)) + c_2 \times \text{Rand} \times (X_{r_2,j}(t) - X_{i,j}(t)), \quad (3)$$

$$c_1 = \exp\left(\frac{(f_i - f_{r_1})}{(\text{abs}(f_i) + \varepsilon)}\right), \quad (4)$$

$$c_2 = (f_{r_2} - f_i), \quad (5)$$

where $X_{i,j}(t)$ is the position of the i th hen in the t -th iteration. Rand is a random number function with a value range of $(0, 1)$. $X_{r_1,j}(t)$ is the position of the rooster which is in the same subgroup as the i th hen. $X_{r_2,j}(t)$ is any chicken randomly selected from the whole chicken swarm, which is different from the i th hen, and $r_1 \neq r_2$.

The formula used by the chicks in foraging is as follows:

$$X_{i,j}(t+1) = X_{i,j}(t) + FL \times (X_{m,j}(t) - X_{i,j}(t)), \quad (6)$$

where $X_{i,j}(t)$ represents the i th chick, and $X_{m,j}(t)$ is the position of the chick's mother. $FL \in (0, 2)$ is a following coefficient that the chick follows the chick's mother to search for food.

The corresponding basic flowchart is shown in Figure 1.

3. The Proposed Algorithm

Aiming at the premature convergence of the standard CSO algorithm in solving multimodal optimization problems, an ICSO algorithm is proposed in this paper. Considering that the chicks have the weakest optimization ability in the whole

chicken swarm, the algorithm improves the foraging behavior of the chicks by referring to the reproduction and elimination-dispersal operations of BFA. At the same time, considering that the PSO algorithm has good global search ability, on the basis of individual division of labor and cooperation optimization mechanism of the CSO algorithm, a hybrid CSO algorithm is constructed to enhance the global search ability of the algorithm by integrating the PSO algorithm.

3.1. CSO Algorithm with Reproduction and Elimination-Dispersal Operations (RECSO Algorithm). In the standard CSO algorithm, chicks are the most vulnerable group. Therefore, in order to improve the depth search ability of the algorithm, this paper introduces the reproduction and elimination-dispersal operations of BFA into the chicks' foraging behavior, and a RECSO algorithm is proposed. Firstly, in the role update process of the chicken swarm, the replication operation of BFA is introduced to replace the chicks with the same number of chickens with the strongest

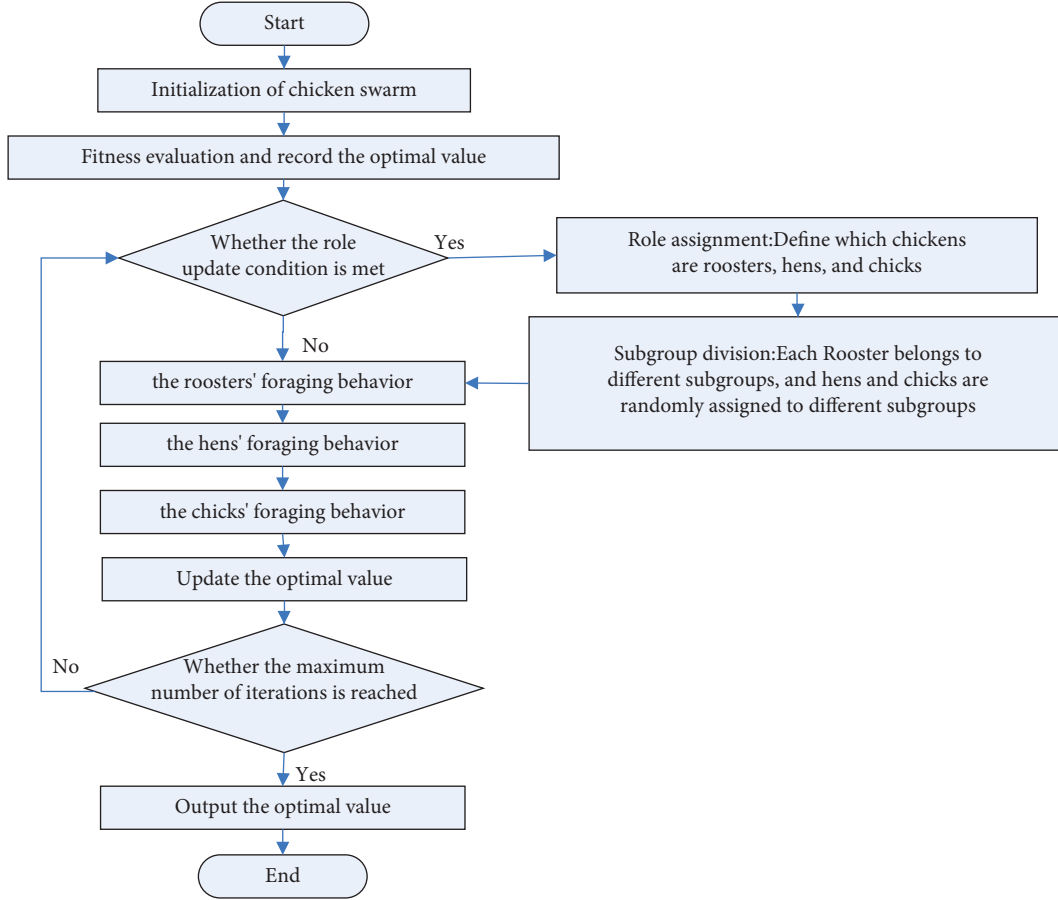


FIGURE 1: Flowchart of the standard CSO algorithm.

optimization ability. Through this behavior, the depth optimization speed of the chicken swarm can be accelerated. At the same time, in order to maintain the population diversity, the elimination-dispersal operation is introduced to disperse the chicks that have performed the replication operation to any position in the search space according to a certain probability. Through this operation, the chicken swarm can avoid falling into the local extrema. The specific reproduction and elimination-dispersal operations are as follows:

3.1.1. Reproduction Operation. According to the fitness function values of the chicken swarm, let the chicks with poor optimization ability inherit the position of the chickens with the strongest optimization ability. The formula is as follows:

$$X_{i,j}(t+1) = X_{i-rNum-hNum,j}(t), \quad (7)$$

where $rNum$ is the number of roosters and $hNum$ is the number of hens.

3.1.2. Elimination-Dispersal Operation. The chicks are dispersed to any position in the search space with the probability P_{ed} , where $P_{ed} = 0.25$. The concrete contents are given as follows:

for $i = (rNum + hNum + 1): pop$ do.

if $p > rand$ then

$$X_{i,j}(t+1) = lb + (ub - lb) \times rand. \quad (8)$$

end.

end.

Here, pop is the population size. lb and ub are the lower and upper bounds of search range, respectively.

3.2. CSO Algorithm Based on PSO (CSO-PSO Algorithm). To improve the global optimization ability of the CSO algorithm, in this section, we construct a CSO-PSO algorithm to enhance the ability of the algorithm to jump out of the local extrema by integrating PSO into the CSO algorithm. The flowchart of the CSO-PSO algorithm is shown in Figure 2.

The main steps are described as follows:

- (1) *Population initialization.* It mainly involves the parameter settings and determination of initial individual between CSO and PSO algorithms.
- (2) *Fitness evaluation.* The initial swarm optimal value is recorded on the bulletin board.

- (3) *Subgroup division*. The initial population is divided into two parts with the same scale as follows: subgroup 1 and subgroup 2.
- (4) *CSO*. We assign roles to subgroup 1 according to the CSO algorithm and perform the foraging behavior of the chicken swarm to search for the global optimal value.
- (5) *PSO*. The velocity and position of particles in subgroup 2 are updated according to the PSO algorithm to search for the global optimal value.
- (6) *Information exchange*. We merge subgroup 1 and subgroup 2 to realize information exchange.
- (7) *Updating swarm optimal value*. The swarm optimal value is updated according to the subgroup optimal values obtained in steps (4) and (5).
- (8) We repeat steps (3)–(7) until the maximum number of iterations is reached and the optimal value is output.

3.3. ICSO Algorithm. In view of the premature convergence problem of the standard CSO algorithm in dealing with multimodal optimization problems, an ICSO algorithm is proposed in this section. Considering that the chicks have the weakest optimization ability in the whole chicken swarm, the RECSO algorithm in Section 3.1 is introduced to improve the depth search ability of the algorithm. At the same time, the CSO-PSO algorithm in Section 3.2 is introduced to improve the global search ability of the algorithm. The flowchart of the ICSO algorithm is shown in Figure 3. The green part is the improvement strategy of reproduction and elimination-dispersal operations introduced in Section 3.1 and the yellow part is the improvement strategy of the PSO algorithm integrated in Section 3.2.

The corresponding detailed steps are as follows:

- (1) *Parameter settings*. It mainly involves the maximum number of iterations M , the population size pop , the dimension of solution space dim , and the limited number of role updates G .
- (2) *Population initialization*. pop solutions are randomly generated in the solution space of the optimization problem, which are used as the initial positions of the population. According to the fitness function of the optimization problem, the fitness value of each position is calculated as a food source.
- (3) *Fitness evaluation*. By comparing the food source content of the whole initial population, the optimal food source and corresponding individual position of the population are recorded.
- (4) *Subgroup division*. The whole population is randomly divided into two parts, namely, subgroup 1 and subgroup 2.
- (5) *RECSO*. The subgroup 1 is optimized according to the RECSO algorithm. The details are summarized as follows:
 - (a) *Judgment of the role update condition*. The role update condition of the whole algorithm is $\text{mod}(-t, G) = 1$, where t is the current iteration number,

mod is a remainder function. If the condition is false, we jump to step (d); otherwise, we judge whether it is the first iteration of the algorithm, if so, we go to step (c), if not, we go to step (b).

- (b) *Reproduction and elimination-dispersal operations*. We perform the reproduction and elimination-dispersal operations which are described in Section 3.1 on the chicks.
- (c) *Role assignment and subgroup division*. According to the fitness function of the current chicken swarm, we update the hierarchical order and mother-child relationship of the whole chicken swarm. After that, we divide the subgroups and determine the number of subgroups according to the number of the roosters. Each rooster belongs to different subgroups. The hens and chicks are randomly assigned to different subgroups, but it is necessary to ensure that there is at least one hen in each subgroup.
- (d) *Foraging behavior*. According to (1), (3), and (6), the foraging behaviors are performed by chickens with different roles.
- (e) *Update of the optimal food source*. At the end of each iteration, the optimization situation of chickens with different roles will change accordingly. We calculate the food source content of the current chicken swarm according to the fitness function and record the optimal food source and its corresponding position by comparing them with the previous situation.
- (6) *PSO*. The velocity and position of particles in subgroup 2 are updated according to the PSO algorithm. The optimal food source and its corresponding position are recorded.
- (7) *Information interaction*. We merge subgroups 1 and 2 to realize the information interaction.
- (8) *Update of swarm optimal value*. The swarm optimal value is updated according to the subgroup optimal values obtained in steps (e) and (6).
- (9) *Judgment of the algorithm's termination conditions*. If the maximum number of iterations specified by the algorithm is reached, the optimal value is output and the program operation is ended; otherwise, we jump to step (4).

4. Simulation Experiment

4.1. Experimental Setup

4.1.1. The Experimental Environment. The experimental environment of this paper is described as follows: Windows 7 operating system, CPU: 3.5 GHz RAM: 12 GB and the programming environment is MATLAB R2016a. In order to verify the effectiveness and superiority of the ICSO algorithm, we conducted experiments on CEC2014 function test suite, which provides 30 test functions, including 3 unimodal functions: $f_1 \sim f_3$, 13 simple multimodal functions: $f_4 \sim f_{16}$, 6 hybrid functions: $f_{17} \sim f_{22}$, and 8

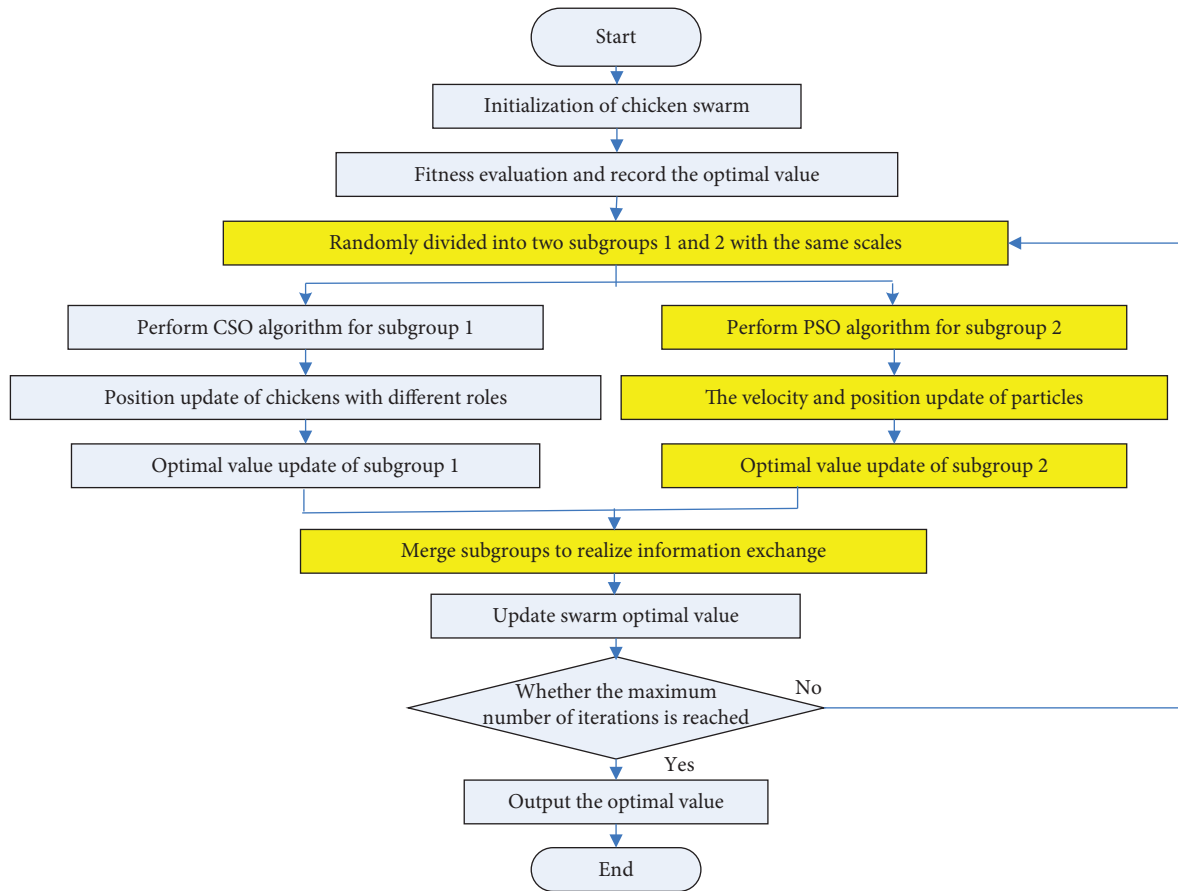


FIGURE 2: Flowchart of CSO-PSO algorithm.

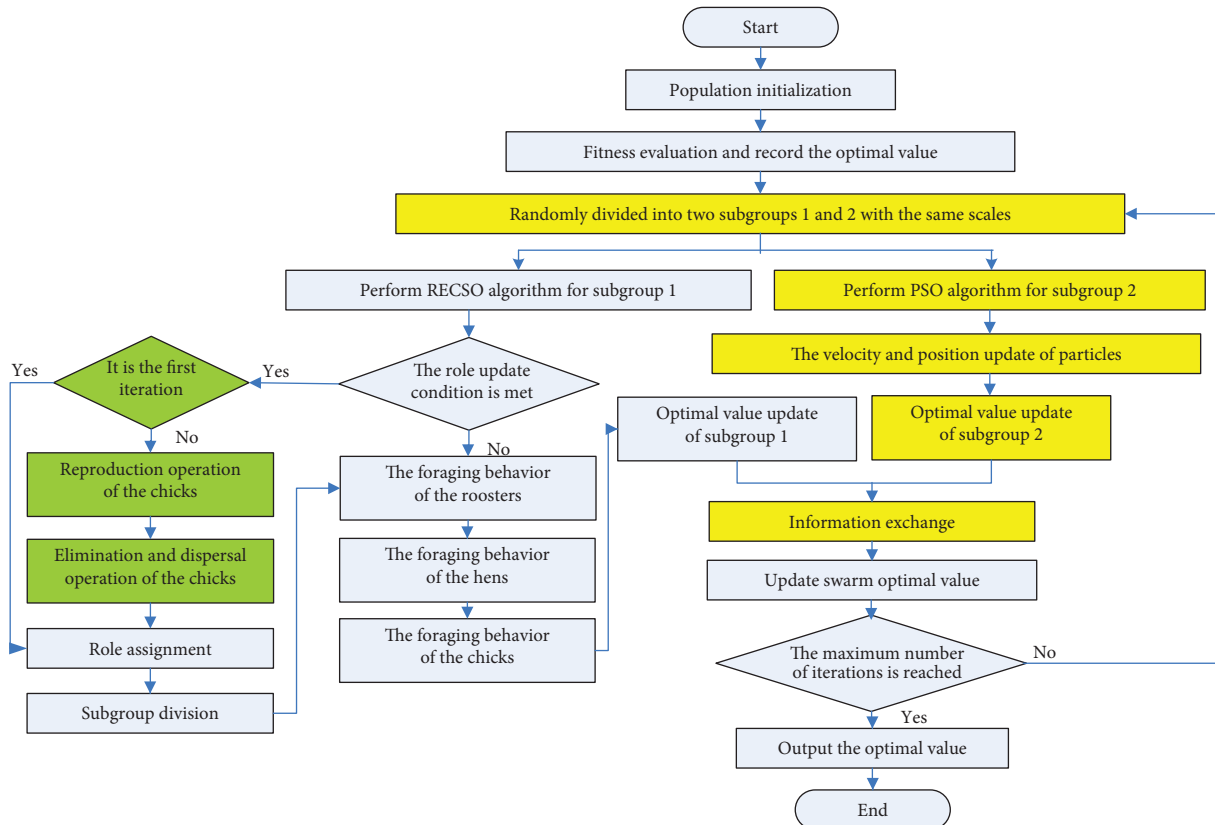


FIGURE 3: Flowchart of ICSO algorithm.

composition functions: $f_{23} \sim f_{30}$. The search range of all test functions is $[-100, 100]$. In order to understand CEC2014 function test suite more intuitively, the function types, numbers, names, and theoretical global optimal values are given in Table 1.

4.1.2. The Parameter Settings. To make a more reasonable comparison, for all algorithms involved in this paper, the population size is set to 100, the maximum number of iterations is 10000, and the dimension of solution space is 10. All algorithms are independently run 30 times on each test function, and then the mean values are calculated. Other parameter settings involved in this experiment are as follows:

- (1) *Parameter settings of ICSO and CSO algorithms.* The limited number of role updates in the chicken swarm $G = 10$. The ratios of roosters and hens in the whole chicken swarm $rPercent$ and $hPercent$ are 0.15 and 0.7, respectively. The proportion of mother hens in the hens $mPercent = 0.5$. In addition, the elimination-dispersal probability P_{ed} of the ICSO algorithm is 0.25.
- (2) *Parameter settings of BFA.* The chemotactic operation $N_c = 100$. The reproduction operation $N_{re} = 10$. The elimination-dispersal operation $N_{ed} = 10$, and the length of a swim $N_s = 4$.
- (3) *Parameter settings of the PSO algorithm.* Two learning factors c_1 and c_2 are both 2.
- (4) *Parameter settings of AFSA.* The visual field of artificial fish $visual = 2.5$. The step length $step = 0.3$. The maximum tentative number $try_number = 5$.
- (5) *Parameter settings of GA.* The binary digits $PRE_CI = 20$. The generation gap $GGAP = 0.9$. The crossover probability $P_c = 0.7$, and the mutation probability $P_m = 0.01$.

In aforementioned parameter settings, the parameters of CSO, BFA, and AFSA are set according to the literature [13, 25] and [14], where CSO, BFA, and AFSA are proposed. It is worth noting that the reason why N_{re} and N_{ed} are set to 10 in BFA is to ensure that the maximum number of iterations is 10000. The parameter settings of PSO and GA are derived from the comparison algorithms mentioned in the literature [28, 32], respectively.

In the ICSO algorithm, considering that if the value of P_{ed} is large, chicks are easy to fall into random exhaustive search. If the value of P_{ed} is small, it is not conducive to maintaining population diversity, which will reduce the local search ability of the algorithm. Therefore, in this section, to give an appropriate value of P_{ed} , we select four typical functions f_3, f_6, f_{20} , and f_{27} from the CEC2014 function test suite for experiments, where f_3 is a unimodal function, f_6 is a simple multimodal function, f_{20} is a hybrid function, and f_{27} is a composition function. At the same time, we choose the ICSO algorithm to run these four functions 30 times independently to calculate their mean values. The experimental results are shown in Table 2.

In Table 2, F_i^* represents the theoretical global optimal values corresponding to different functions, where $i = 3, 6, 20, 27$. The value of P_{ed} is taken from 0.05 to 0.85 with an interval of 0.2. $|D_i|$ represents the absolute value of the difference between the theoretical global optimal value and the actual mean one obtained on each function, where $i = 0.05, 0.25, \dots, 0.85$. It is easy to see from Table 2 that when $P_{ed} = 0.25$, the value of $|D_i|$ is the smallest, that is, the actual mean optimal values obtained by the ICSO algorithm on four functions are the closest to the theoretical ones. Furthermore, when the value of P_{ed} changes from 0.25 to 0.85 with an interval of 0.2, the value of $|D_i|$ is larger than that of $|D_{0.25}|$. Therefore, in the experiment, we choose $P_{ed} = 0.25$.

4.2. The Effectiveness Test of ICSO, RECSO, and CSO-PSO. To verify the effectiveness of the three improved algorithms proposed in this paper, namely ICSO, RECSO, and CSO-PSO, compared with the standard CSO algorithm, in this section, these four algorithms are tested on the CEC2014 function test suite, and the experimental comparison is made in terms of optimization accuracy and convergence performance. The experimental parameter settings are shown in Section 4.1.

4.2.1. The Effectiveness Test for Optimization Accuracy. To verify the effectiveness of the three improved algorithms (ICSO, RECSO, and CSO-PSO) in terms of optimization accuracy, in this section, we use ICSO, RECSO, CSO-PSO, and CSO to run all 30 functions of the CEC2014 test suite independently 30 times to obtain their mean values. The results are shown in Table 3, where the symbols “>,” “=,” and “<” indicate that the experimental results of the comparison algorithms are superior, equal, and inferior to the CSO algorithm, respectively. The optimal results are shown in bold.

It can be clearly seen from Table 3 that the ICSO algorithm has the largest number of optimal values, followed by CSO-PSO, and the CSO algorithm is the worst. Most specifically, the operation results of the ICSO algorithm on 28 functions are better than those of the CSO algorithm, and the operation results on 2 functions are the same as those of the CSO algorithm. The operation results of the RECSO algorithm on 19 functions are better than those of the CSO algorithm. The operation results of it on 6 functions are worse than those of the CSO algorithm, and its operation results are the same as those of the CSO algorithm on the other 5 functions. The results of the CSO-PSO algorithm are similar to those of the ICSO algorithm. There are 28 functions whose operation results are better than those of the CSO algorithm, and the operation results on 2 functions are the same as those of the CSO algorithm. However, from the number of optimal values obtained, the ICSO algorithm is far superior to the CSO-PSO algorithm. This shows the effectiveness of the three improved strategies in terms of optimization accuracy compared with the CSO algorithm, and the performance of the ICSO algorithm is the best among the four algorithms.

TABLE 1: The description of the CEC2014 test functions.

Type	No	Functions	F_1^*
Unimodal functions	f_1	Rotated high conditioned elliptic function	100
	f_2	Rotated bent cigar function	200
	f_3	Rotated discus function	300
Simple multimodal functions	f_4	Shifted and rotated Rosenbrock's function	400
	f_5	Shifted and rotated Ackley's function	500
	f_6	Shifted and rotated Weierstrass's function	600
	f_7	Shifted and rotated Griewank's function	700
	f_8	Shifted Rastrigin's function	800
	f_9	Shifted and rotated Rastrigin's function	900
	f_{10}	Shifted Schwefel's function	1000
	f_{11}	Shifted and rotated Schwefel's function	1100
	f_{12}	Shifted and rotated katsuura function	1200
	f_{13}	Shifted and rotated HappyCat function	1300
	f_{14}	Shifted and rotated HGBat function	1400
	f_{15}	Shifted and rotated expanded Griewank's plus Rosenbrock's function	1500
	f_{16}	Shifted and rotated expanded Scaffer's F6 function	1600
Hybrid functions	f_{17}	Hybrid function 1 ($N=3$)	1700
	f_{18}	Hybrid function 2 ($N=3$)	1800
	f_{19}	Hybrid function 3 ($N=4$)	1900
	f_{20}	Hybrid function 4 ($N=4$)	2000
	f_{21}	Hybrid function 5 ($N=5$)	2100
	f_{22}	Hybrid function 6 ($N=5$)	2200
Composition functions	f_{23}	Composition function 1 ($N=5$)	2300
	f_{24}	Composition function 2 ($N=3$)	2400
	f_{25}	Composition function 3 ($N=3$)	2500
	f_{26}	Composition function 4 ($N=5$)	2600
	f_{27}	Composition function 5 ($N=5$)	2700
	f_{28}	Composition function 6 ($N=5$)	2800
	f_{29}	Composition function 7 ($N=3$)	2900
	f_{30}	Composition function 8 ($N=3$)	3000

TABLE 2: The experimental results via different P_{ed} .

Functions	f_3	f_6	f_{20}	f_{27}
F_1^*	300	600	2000	2700
$P_{ed}=0.05$	300.1282	600.7110	$2.0077e+03$	$2.8786e+03$
$ D_{0.05} $	0.1282	0.7110	7.7	178.6
$P_{ed}=0.25$	300.0813	600.7149	2.0053e+03	2.8314e+03
$ D_{0.25} $	0.0813	0.7149	5.3	131.4
$P_{ed}=0.45$	300.1122	600.8051	$2.0068e+03$	$2.8911e+03$
$ D_{0.45} $	0.1122	0.8051	6.8	191.1
$P_{ed}=0.65$	300.1803	600.8868	$2.0075e+03$	$2.8463e+03$
$ D_{0.65} $	0.1803	0.8868	7.5	146.3
$P_{ed}=0.85$	300.1330	601.0191	$2.0074e+03$	$2.8795e+03$
$ D_{0.85} $	0.1330	1.0191	7.4	179.5

TABLE 3: The experimental results of ICSO, RECSO, CSO-PSO, and CSO algorithms.

Functions	ICSO	CSO	RECSO	CSO-PSO
f_1	2.5963e+03 >	$7.4764e+05$	$3.2289e+05$ >	$6.1321e+03$ >
f_2	728.1158 >	$1.5107e+07$	$5.8310e+06$ >	$1.0100e+03$ >
f_3	300.0813 >	957.7185	446.6751>	300.2133>
f_4	411.1957 >	431.2580	426.6130>	414.4211>
f_5	517.9530 >	519.9147	519.2104>	519.1367>
f_6	600.7149 >	602.1075	601.8348>	600.8712>
f_7	700.0947 >	701.4794	701.4910<	700.1054>
f_8	800.1327 >	808.6386	809.2422<	800.3648>

TABLE 3: Continued.

Functions	ICSO	CSO	RECSO	CSO-PSO
f_9	904.7067>	911.8049	911.5464>	905.7708>
f_{10}	1.0499e+03>	1.1771e+03	1.1783e+03<	1.1091e+03>
f_{11}	1.2233e+03>	1.4489e+03	1.4199e+03>	1.2716e+03>
f_{12}	1.2001e+03>	1.2005e+03	1.2005e+03=	1.2001e+03>
f_{13}	1.3001e+03=	1.3001e+03	1.3001e+03=	1.3001e+03=
f_{14}	1.4001e+03>	1.4003e+03	1.4003e+03=	1.4001e+03>
f_{15}	1.5007e+03>	1.5011e+03	1.5009e+03>	1.5009e+03>
f_{16}	1.6015e+03>	1.6020e+03	1.6020e+03=	1.6015e+03>
f_{17}	2.6873e+03>	3.6491e+03	5.8702e+03<	3.0513e+03>
f_{18}	2.6243e+03>	5.1907e+03	3.5783e+03>	4.7507e+03>
f_{19}	1.9008e+03>	1.9023e+03	1.9018e+03>	1.9013e+03>
f_{20}	2.0053e+03>	2.0848e+03	2.0673e+03>	2.0068e+03>
f_{21}	2.1390e+03>	2.3364e+03	2.3150e+03>	2.1486e+03>
f_{22}	2.2045e+03>	2.2244e+03	2.2264e+03<	2.2084e+03>
f_{23}	2.6122e+03>	2.6138e+03	2.5955e+03>	2.6122e+03>
f_{24}	2.5148e+03>	2.5258e+03	2.5211e+03>	2.5158e+03>
f_{25}	2.6629e+03>	2.6923e+03	2.6713e+03>	2.6694e+03>
f_{26}	2.7001e+03=	2.7001e+03	2.7001e+03=	2.7001e+03=
f_{27}	2.8314e+03>	2.9131e+03	2.8878e+03>	2.8759e+03>
f_{28}	3.1761e+03>	3.1878e+03	3.1792e+03>	3.1765e+03>
f_{29}	3.2961e+03>	3.5482e+03	3.5628e+03<	3.3786e+03>
f_{30}	3.5856e+03>	3.8147e+03	3.6982e+03>	3.7041e+03>
>	28	—	19	28
<	0	—	6	0
=	2	—	5	2
The number of optimal values	29	2	3	5

4.2.2. The Effectiveness Test for Convergence Performance.

To verify the effectiveness of the abovementioned three improved algorithms in terms of convergence performance compared to the CSO algorithm, Figure 4 shows the average convergence curves of the four algorithms independently running 30 times on 30 functions. In order to show the convergence effect of each algorithm more clearly, the average fitness function values are logarithmically processed in Figure 4. In addition, the convergence curves of some functions contain subgraphs, such as those of f_4 , f_8 , and f_{13} , which are locally magnified renderings.

It can be seen from Figure 4 that the convergence performance of the ICSO algorithm on functions f_1 , f_5 , f_9 , f_{10} , f_{11} , f_{17} , f_{18} , f_{25} , and f_{27} is significantly better than that of the other three algorithms. On functions f_2 , f_3 , f_6 , f_8 , f_{12} , f_{16} , f_{22} , and f_{24} , ICSO and CSO-PSO algorithms have similar convergence performance. The former is slightly better than the latter, and both of them are better than CSO and RECSO algorithms (only on function f_{16} , CSO-PSO algorithm is slightly better than the ICSO algorithm). On functions f_4 , f_7 , f_{13} , f_{14} , f_{15} , f_{19} , f_{20} , f_{21} , f_{28} , f_{29} , and f_{30} , the four algorithms have comparable convergence performance, and ICSO is slightly superior. On functions f_{23} and f_{26} , the RECSO algorithm has the best convergence performance.

To sum up, the convergence performance of the ICSO algorithm on 27 functions is the best, and it is better than that of the CSO algorithm on all 30 functions. The CSO-PSO algorithm has the best convergence performance on one function, and its convergence performance on 28 functions

is better than that of the CSO algorithm. The RECSO algorithm has the best convergence performance on two functions, and only the convergence performance on functions f_8 , f_{13} , and f_{17} is slightly inferior to that of the CSO algorithm. This fully shows the effectiveness of the three improved algorithms in terms of convergence performance compared with the CSO algorithm, and the ICSO algorithm has the best convergence performance.

It is concluded that the reason why the ICSO algorithm is superior to the other three algorithms may be that the algorithm integrates the ideas of BFA and PSO, which not only improves the depth search ability of the algorithm but also improves its breadth searchability.

4.3. The Superiority Comparison among Several Swarm Intelligent Algorithms.

To verify the superiority of the ICSO algorithm proposed in this paper, in this section, we compare the performance of seven algorithms, namely, ICSO, CSO, ICSOII proposed in the literature [30] (named ICSOII to distinguish it from the ICSO algorithm), BFA, PSO, AFSA, and GA from the aspects of optimization accuracy and convergence performance.

4.3.1. The Superiority Comparison in Optimization Accuracy.

To verify the superiority of the ICSO algorithm in terms of optimization accuracy, this section presents the experimental results of the abovementioned seven algorithms to optimize the CEC2014 function test suite. The data in Table 4 are the mean values of 30 independent runs of each

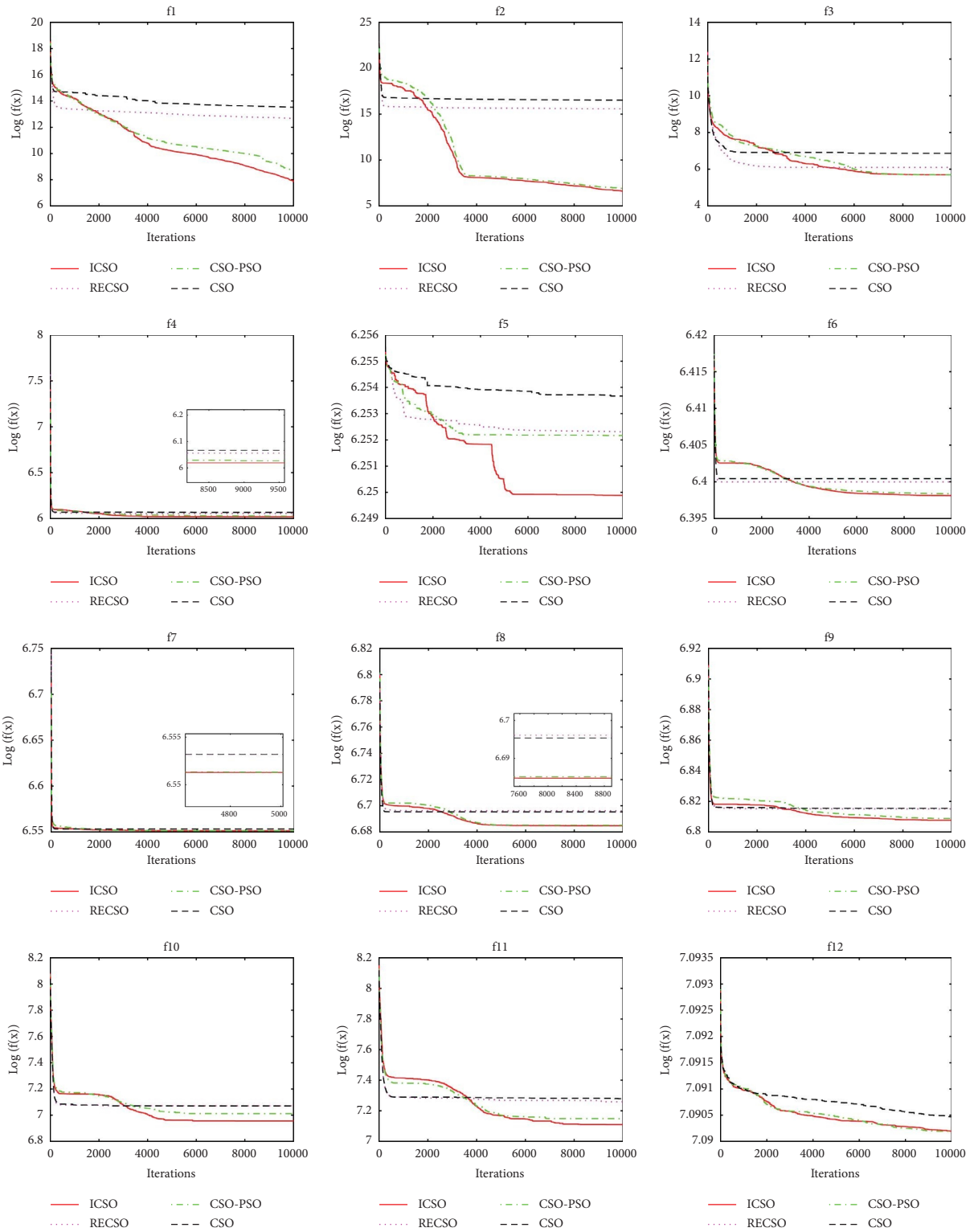


FIGURE 4: Continued.

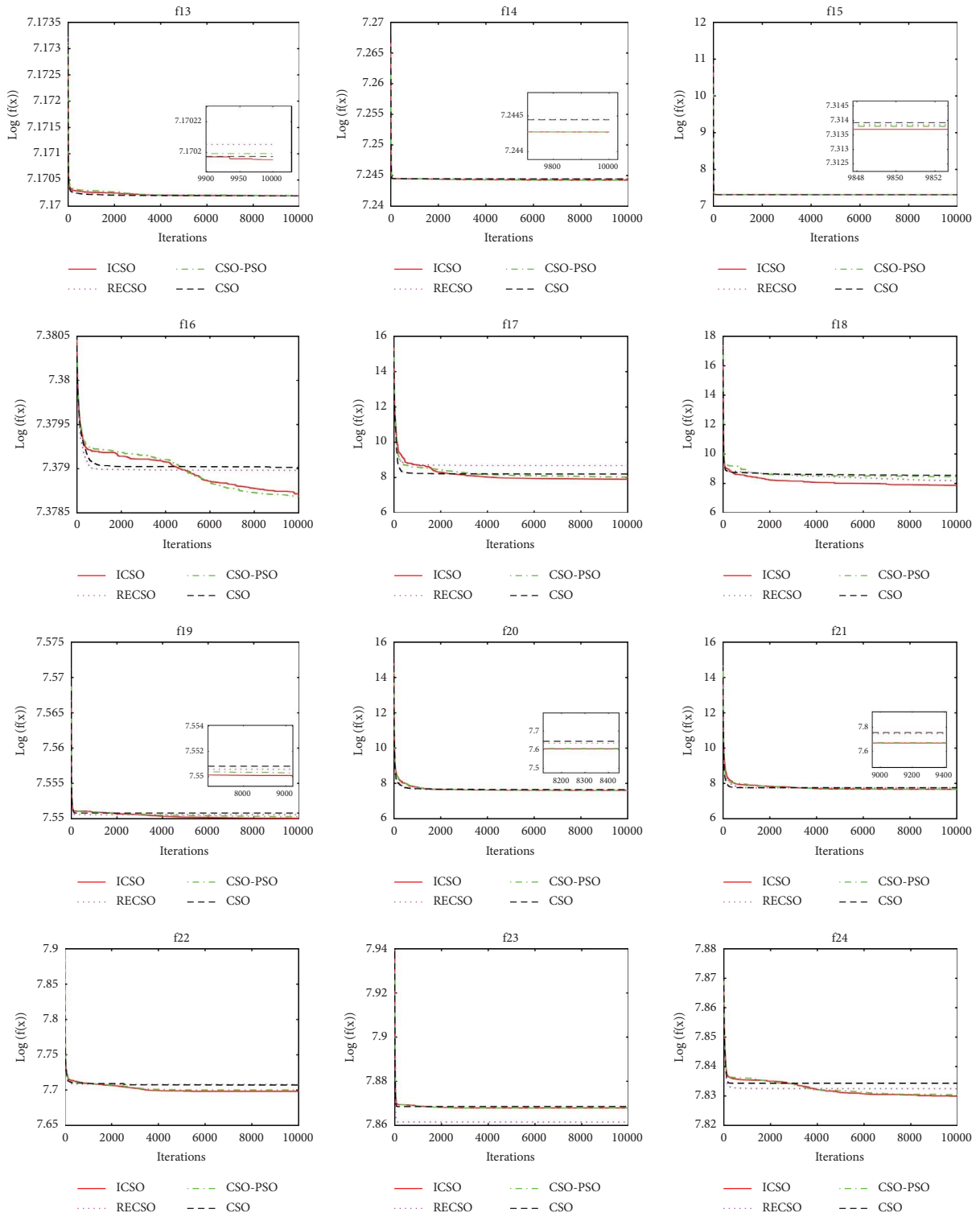


FIGURE 4: Continued.

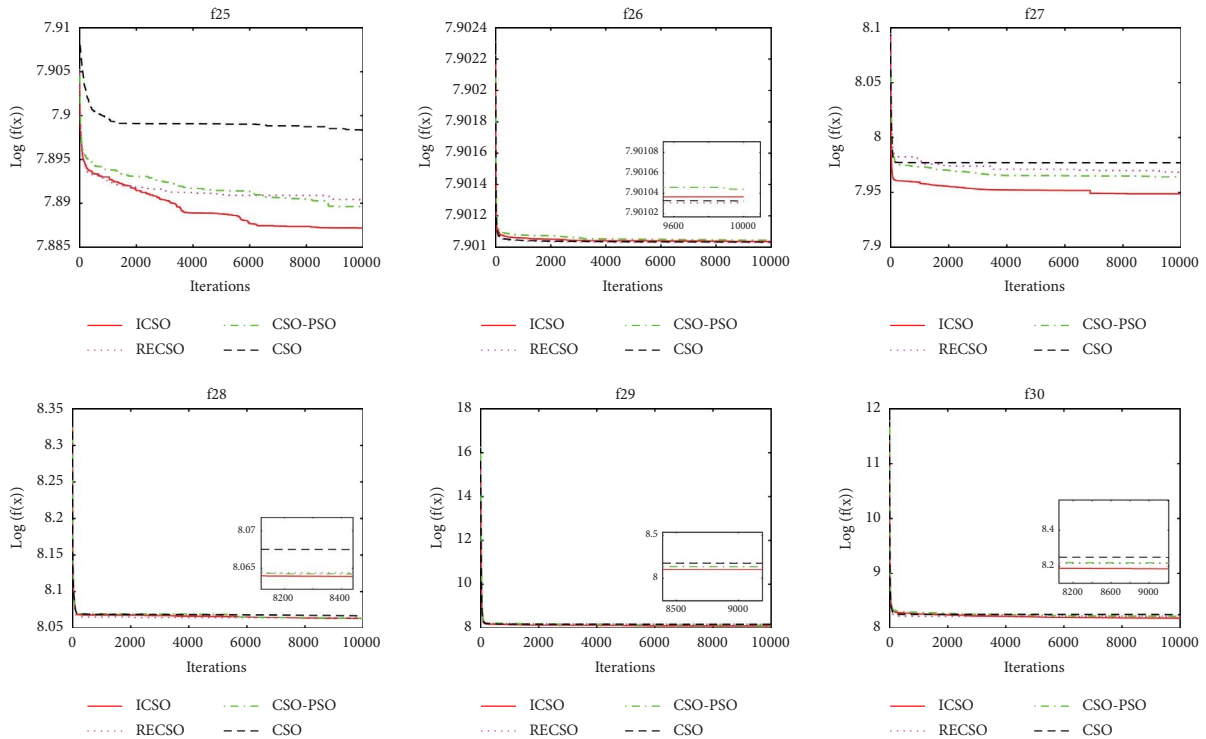


FIGURE 4: Convergence curves of 4 algorithms on 30 test functions.

algorithm on each function. The bold data in Table 4 are the optimal values.

It can be seen from Table 4 that for functions f_1 and f_2 , the ICSO algorithm directly reduces the order of magnitude of optimization accuracy from 5 and 7 to 3 and 2, respectively. In addition, from the number of optimal values that can be found, the number of optimal values that can be found by CSO and GA is 2. AFSA and PSO can find 3 and 9 optimal values, respectively. The optimal value of ICSOII and BFA is both 6. The number of optimal values that can be found by the ICSO algorithm is 18, which shows the superiority of the ICSO algorithm in optimization accuracy.

4.3.2. The Superiority Comparison in Convergence Performance. To verify the superiority of the ICSO algorithm in convergence performance, in this section, we give the average convergence curves of 30 test functions on the CEC2014 function test suite optimized by the above-mentioned seven algorithms (each algorithm is independently run 30 times on each test function). The parameter settings in this section are shown in Section 4.1.2. As mentioned in Section 4.2.2, the ordinates in Figure 5 are the logarithms of the average fitness function values, and subgraphs in some convergence curves are locally magnified renderings, so as to show the convergence effect of each algorithm more clearly.

It can be seen from Figure 5 that the convergence performance of the ICSO algorithm is the best on functions $f_1, f_5 \sim f_7, f_{11}, f_{14}, f_{15}, f_{17}, f_{19}, f_{21}, f_{22}, f_{24}, f_{25},$ and f_{27} , among which, on the functions $f_5, f_6, f_{11}, f_{25},$ and f_{27} , the convergence advantage of the ICSO algorithm is particularly obvious. On

functions $f_2, f_3, f_8, f_9,$ and f_{20} , the convergence performance between ICSO and PSO algorithms is very close, the performance of the PSO algorithm is slightly better, and they all outperform the performance of the other algorithms. On functions f_{13} and f_{26} , the convergence performance of ICSO and CSO algorithms is the best, and the performance of the ICSO algorithm is slightly inferior to that of the CSO algorithm, ranking second. On the function f_{16} , ICSOII has the best convergence performance and the ICSO algorithm ranks second. Only on functions $f_4, f_{12}, f_{18}, f_{23},$ and $f_{28} \sim f_{30}$, the convergence performance of the ICSO algorithm is not as good as that of BFA, AFSA, and GA, ranking 3rd or 4th. From this, we can see the superiority of the ICSO algorithm proposed in this paper in terms of convergence performance.

4.3.3. Friedman Test of Algorithms. To compare the performance of various algorithms more reasonably, this section uses the Friedman test to test the performance of the above-mentioned 7 algorithms (ICSO, ICSOII, PSO, CSO, GA, BFA, and AFSA) from a statistical point of view. The Friedman test is a nonparametric test method, which is often used to test the performance of algorithms due to its simple operation and lax requirements on the test data [32, 33, 36]. For the minimum optimization problem, the smaller the average ranking of the algorithm, the better its performance. Table 5 is the Friedman test results of 7 algorithms on 30 functions.

It can be seen from Table 5 that the average ranking of the ICSO algorithm is 1.90, ranking the highest, 0.87 lower than that of ICSOII, 2.70 lower than that of the CSO

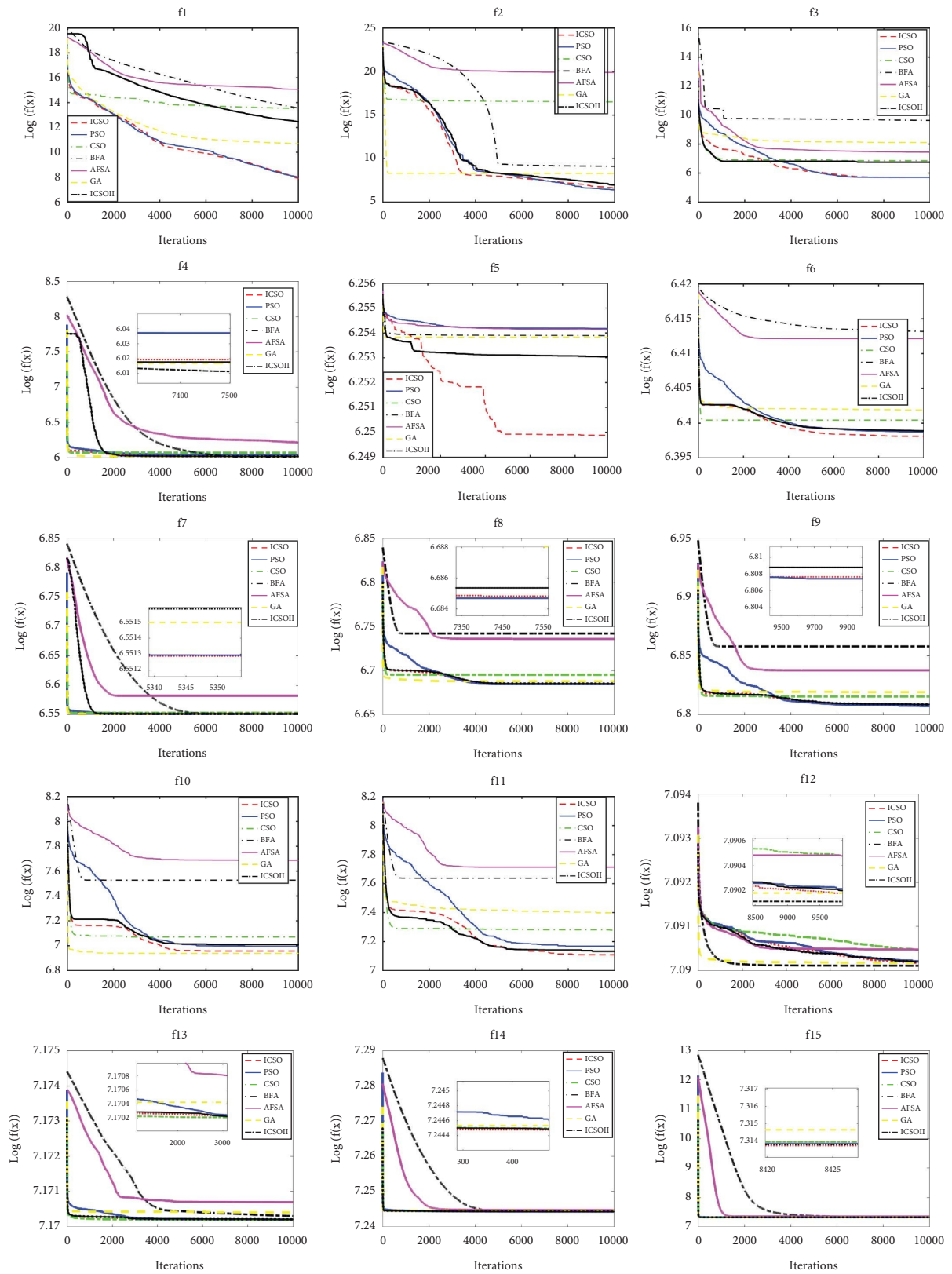


FIGURE 5: Continued.

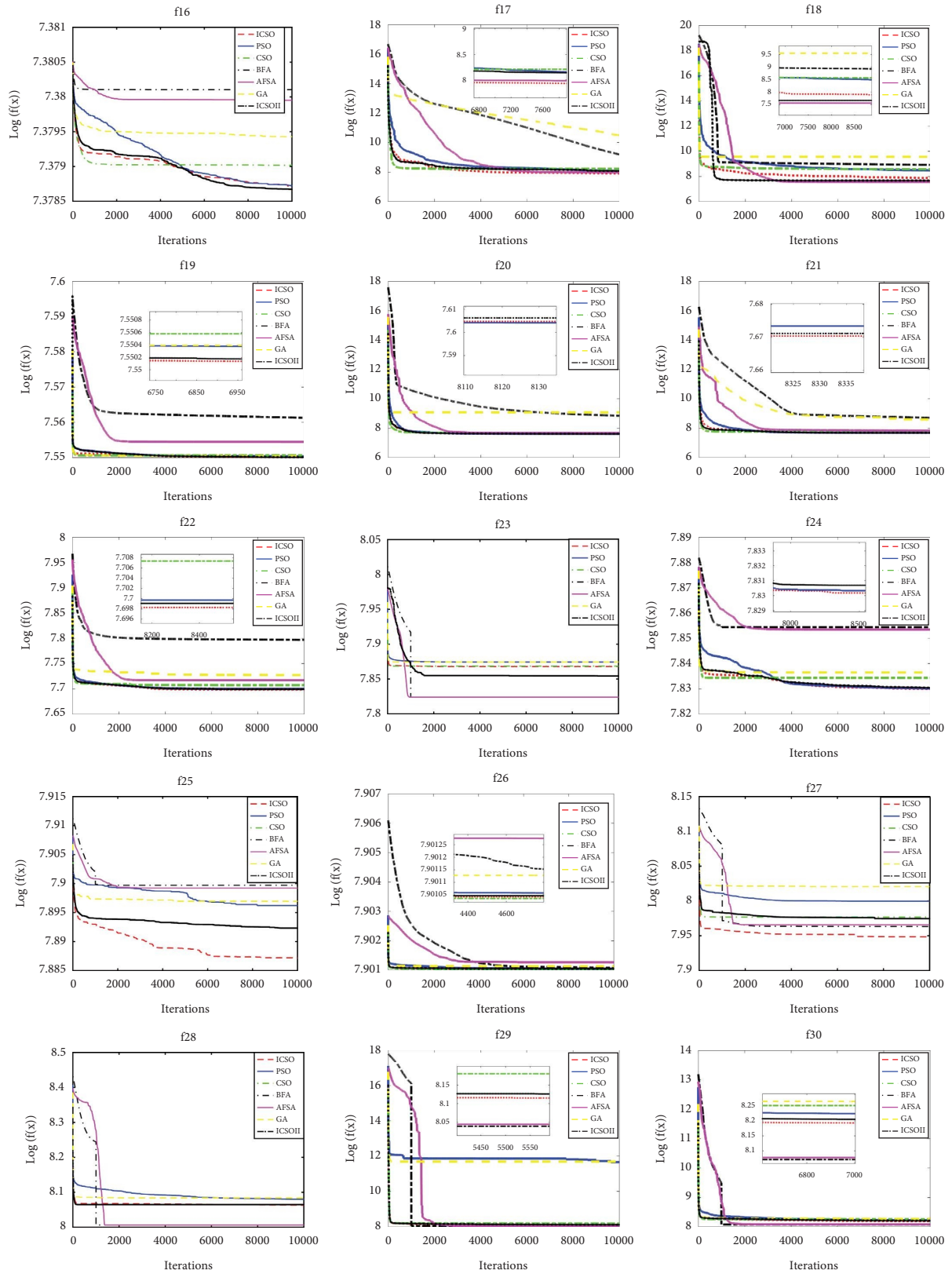


FIGURE 5: Convergence curves of 7 algorithms on 30 test functions.

TABLE 4: The optimization accuracy comparison of 7 intelligent algorithms.

Function	ICSO	CSO	ICSOII	BFA	PSO	AFSA	GA
f_1	2.5963e+03	7.4764e+05	2.6050e+05	7.8186e+05	3.0053e+03	3.4393e+06	4.4127e+04
f_2	728.1158	1.5107e+07	1.0659e+03	8.9827e+03	599.1427	4.3975e+08	3.9775e+03
f_3	300.0813	957.7185	857.3808	1.5267e+04	300.0106	1.7089e+03	3.3396e+03
f_4	411.1957	431.2580	409.3101	404.7945	418.4158	499.9456	410.1793
f_5	517.9530	519.9147	519.5907	520.0342	520.1769	520.1581	519.9999
f_6	600.7149	602.1075	600.9178	609.8307	601.0835	609.2003	602.9651
f_7	700.0947	701.4794	700.3004	700.3424	700.0974	722.0824	700.2899
f_8	800.1327	808.6386	800.5306	847.2295	800.0663	842.1636	802.7196
f_9	904.7067	911.8049	905.7401	951.4788	904.5150	932.2919	915.1409
f_{10}	1.0499e+03	1.1771e+03	1.0717e+03	1.8578e+03	1.0897e+03	2.1799e+03	1.0321e+03
f_{11}	1.2233e+03	1.4489e+03	1.2524e+03	2.0745e+03	1.2969e+03	2.2376e+03	1.6340e+03
f_{12}	1.2001e+03	1.2005e+03	1.2001e+03	1.2001e+03	1.2002e+03	1.2005e+03	1.2001e+03
f_{13}	1.3001e+03	1.3001e+03	1.3001e+03	1.3002e+03	1.3001e+03	1.3007e+03	1.3004e+03
f_{14}	1.4001e+03	1.4003e+03	1.4001e+03	1.4003e+03	1.4001e+03	1.4006e+03	1.4004e+03
f_{15}	1.5007e+03	1.5011e+03	1.5008e+03	1.5083e+03	1.5009e+03	1.5398e+03	1.5021e+03
f_{16}	1.6015e+03	1.6020e+03	1.6015e+03	1.6038e+03	1.6015e+03	1.6035e+03	1.6027e+03
f_{17}	2.6873e+03	3.6491e+03	3.1426e+03	9.7385e+03	3.1797e+03	2.8286e+03	3.5779e+04
f_{18}	2.6243e+03	5.1907e+03	2.1108e+03	7.3224e+03	4.7427e+03	1.9077e+03	1.4040e+04
f_{19}	1.9008e+03	1.9023e+03	1.9008e+03	1.9222e+03	1.9013e+03	1.9092e+03	1.9014e+03
f_{20}	2.0053e+03	2.0848e+03	2.0067e+03	6.8899e+03	2.0040e+03	2.1115e+03	8.6438e+03
f_{21}	2.1390e+03	2.3364e+03	2.1546e+03	5.9637e+03	2.1501e+03	2.5022e+03	5.3393e+03
f_{22}	2.2045e+03	2.2244e+03	2.2063e+03	2.4322e+03	2.2077e+03	2.2462e+03	2.2689e+03
f_{23}	2.6122e+03	2.6138e+03	2.5766e+03	2.5000e+03	2.6295e+03	2.5000e+03	2.6295e+03
f_{24}	2.5148e+03	2.5258e+03	2.5159e+03	2.5769e+03	2.5153e+03	2.5745e+03	2.5312e+03
f_{25}	2.6629e+03	2.6923e+03	2.6766e+03	2.6965e+03	2.6871e+03	2.6952e+03	2.6890e+03
f_{26}	2.7001e+03	2.7001e+03	2.7001e+03	2.7002e+03	2.7001e+03	2.7007e+03	2.7003e+03
f_{27}	2.8314e+03	2.9131e+03	2.9066e+03	2.8737e+03	2.9798e+03	2.8807e+03	3.0434e+03
f_{28}	3.1761e+03	3.1878e+03	3.1799e+03	3.0000e+03	3.2304e+03	3.0000e+03	3.2409e+03
f_{29}	3.2961e+03	3.5482e+03	3.3122e+03	3.1000e+03	1.1433e+05	3.1064e+03	1.1836e+05
f_{30}	3.5856e+03	3.8147e+03	3.6306e+03	3.2000e+03	3.6875e+03	3.2197e+03	3.8617e+03
The number of optimal values	18	2	6	6	9	3	2

algorithm and 1.38 lower than that of the PSO algorithm, which fully demonstrates the effectiveness of the improved strategy in this paper.

To sum up, the reason why the ICSO algorithm is superior to CSO, PSO, and BFA may be that it greatly enhances the ability of the algorithm to jump out of the local extrema by innovative cooperation between the chicken swarm and particle one to achieve information interaction and improves the depth searchability of the algorithm by integrating the replication and elimination-dispersal operations of BFA. The reason why the ICSO algorithm is superior to ICSOII, GA, and AFSA may be that the ICSO algorithm has a mechanism of subgroup division and multiswarm cooperation based on chicken swarm and particle one, which realizes parallel optimization through group cooperation. Although the ICSOII algorithm has subgroup division, its population cooperation is limited to the cooperation within the chicken swarm, so its optimization ability is weaker than that of the ICSO algorithm.

4.4. Experimental Comparison between ICSO and ICSOII.

To further compare the performance between the ICSO algorithm proposed in this paper and the ICSOII algorithm proposed in the literature [30], we set the parameters of the algorithm according to the literature [30] in this section. The statistical results of the abovementioned two

algorithms running 51 times independently on the CEC2014 function test suite are shown in Table 6. The experimental data of the ICSOII algorithm comes from its corresponding literature.

It can be clearly seen from Table 6 that the number of optimal values obtained by the ICSOII algorithm is 18, and the theoretical optimal values are obtained on 10 functions. While the number of optimal values obtained by the ICSO algorithm is 21, and its theoretical optimal values are obtained on 12 functions.

4.5. Experimental Comparison between ICSO and a State-of-the-Art Algorithm.

To further verify the performance of the ICSO algorithm, DMSDL-QBSA which is a state-of-the-art algorithm proposed in the literature [18] is also used to compare with the ICSO algorithm in this section. In order to make the experimental comparison fairer and more reasonable, the parameter settings of the ICSO algorithm are the same as those of DMSDL-QBSA, that is, the population size is set to 30, and the maximum number of iterations is set to 100000. Other parameter settings can be seen in Section 4.1.2.

The experimental data including the maximum (Max), minimum (Min), mean (Mean), and variance (Var) values are shown in Table 7, where the optimal results are shown in

bold. It is worth noting that the experimental data of DMSDL-QBSA are extracted from its corresponding literature.

By analyzing the data in Table 7, we can see that our ICSO algorithm can get all the best values on functions $f_1 \sim f_3$, $f_5 \sim f_9$, $f_{12} \sim f_{21}$, f_{26} , and f_{29} . On functions f_4 , f_{11} , f_{22} , f_{24} , f_{25} , and f_{30} , the ICSO algorithm has the best results in terms of the maximum, minimum, and mean values but is slightly inferior to DMSDL-QBSA in variance value. DMSDL-QBSA only achieves relatively good results on functions f_{23} , f_{27} , and f_{28} . To sum up, our ICSO algorithm has advantages in most

test functions due to the improvement of global and deep search abilities.

4.6. Welded Beam Design Problem. To verify the performance of the ICSO algorithm to solve practical optimization problems, a welded beam design problem is considered, which has been described in detail in the literature [4, 37] and [38]. The problem is a minimum problem, which can be formulated as follows:

$$\begin{aligned}
 &\text{Minimize } f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2), \\
 &\text{Subject to } g_1(x) = \tau(x) - \tau_{\max} \leq 0, \\
 &\quad g_2(x) = \sigma(x) - \sigma_{\max} \leq 0, \\
 &\quad g_3(x) = x_1 - x_4 \leq 0, \\
 &\quad g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0, \\
 &\quad g_5(x) = 0.125 - x_1 \leq 0, \\
 &\quad g_6(x) = \delta(x) - \delta_{\max} \leq 0, \\
 &\quad g_7(x) = P - P_c(x) \leq 0,
 \end{aligned} \tag{9}$$

where $0.1 \leq x_1, x_4 \leq 2, 0.1 \leq x_2, x_3 \leq 10$,

$$\begin{aligned}
 \tau(x) &= \sqrt{\left((\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2 \right)}, \tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, \\
 M &= P\left(L + \frac{x_2}{2}\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \\
 J &= 2\left\{ \sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right] \right\}, \sigma(x) = \frac{6PL}{x_4x_3^2}, \delta(x) = \frac{4PL^3}{Ex_4x_3^3}, \\
 P_c(x) &= \frac{4.013E\sqrt{x_4^6x_3^2/36}}{L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right).
 \end{aligned} \tag{10}$$

$P = 6000lb$, $L = 14in.$, $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi, $\tau_{\max} = 13600$ psi, $\sigma_{\max} = 30000$, $\delta_{\max} = 0.25$ in.

The comparison results of optimal solutions obtained by different algorithms are shown in Table 8. The statistical results are shown in Table 9, where “Worst,” “Mean,” “Best,” and “SD” stand for the worst, mean, best, and standard deviation values obtained by 30 independent runs, respectively. In addition, the optimal results are shown in bold. It is worth noting that the experimental results of comparison algorithms are extracted from their corresponding literature.

For HFPSO [4] and EPSO [37], because the optimal solutions of the four parameters are not given in the literature [4, 37], they are not listed in Table 8. It can be seen from Tables 8 and 9, ICSO and MBA [38] have obvious advantages over the other two algorithms in terms of the worst, mean values, standard deviation, etc. Although the stability of the ICSO algorithm is slightly inferior to that of MBA [38], it has a higher optimization accuracy, which preliminarily shows that ICSO can be used to solve the welded beam design problem.

TABLE 5: Friedman test results of 7 algorithms.

Algorithm	Average ranking	Ranking
ICSO	1.90	1
ICSOII	2.77	2
PSO	3.28	3
CSO	4.60	4
BFA	5.07	5
GA	5.13	6
AFSA	5.25	7

TABLE 6: The experimental results between ICSO and ICSOII.

Function	ICSO	ICSOII
f_1	3.74e + 03 ± 4.81e ± 03	5.55e + 04 ± 4.75e ± 04
f_2	8.63e + 02 ± 5.85e + 02	1.08e + 03 ± 2.71e + 02
f_3	3.00e + 02 ± 1.19e - 01	7.71e + 02 ± 2.86e + 02
f_4	4.14e + 02 ± 1.45e + 01	4.00e + 02 ± 1.88e - 01
f_5	5.18e + 02 ± 4.75e + 00	5.19e + 02 ± 5.97e + 00
f_6	6.00e + 02 ± 9.13e - 01	6.00e + 02 ± 1.24e + 00
f_7	7.00e + 02 ± 4.93e - 02	7.00e + 02 ± 1.00e - 02
f_8	8.00e + 02 ± 5.16e - 01	8.03e + 02 ± 3.77e + 00
f_9	9.05e + 02 ± 1.94e + 00	9.06e + 02 ± 4.06e + 00
f_{10}	1.05e + 03 ± 6.31e + 01	1.06e + 03 ± 2.36e + 02
f_{11}	1.25e + 03 ± 1.13e + 02	1.46e + 03 ± 2.01e + 02
f_{12}	1.20e + 03 ± 1.46e - 01	1.20e + 03 ± 3.14e - 02
f_{13}	1.30e + 03 ± 2.95e - 02	1.30e + 03 ± 6.97e - 02
f_{14}	1.40e + 03 ± 2.42e - 02	1.40e + 03 ± 7.79e - 02
f_{15}	1.50e + 03 ± 2.56e - 01	1.50e + 03 ± 1.81e - 01
f_{16}	1.60e + 03 ± 4.21e - 01	1.60e + 03 ± 3.18e - 01
f_{17}	2.83e + 03 ± 7.98e + 02	2.16e + 03 ± 2.47e + 02
f_{18}	3.07e + 03 ± 1.78e + 03	1.91e + 03 ± 6.00e + 01
f_{19}	1.90e + 03 ± 7.00e - 01	1.90e + 03 ± 3.48e - 01
f_{20}	2.00e + 03 ± 5.27e + 00	2.10e + 03 ± 6.34e + 01
f_{21}	2.12e + 03 ± 4.53e + 01	2.36e + 03 ± 6.19e + 00
f_{22}	2.20e + 03 ± 9.25e + 00	2.22e + 03 ± 7.12e + 01
f_{23}	2.50e + 03 ± 3.07e + 01	2.40e + 03 ± 7.12e + 01
f_{24}	2.51e + 03 ± 3.71e + 00	2.52e + 03 ± 4.68e + 00
f_{25}	2.67e + 03 ± 3.33e + 01	2.61e + 03 ± 9.46e + 00
f_{26}	2.70e + 03 ± 2.35e - 02	2.70e + 03 ± 4.46e - 02
f_{27}	2.84e + 03 ± 1.67e + 02	2.70e + 03 ± 6.15e + 01
f_{28}	3.17e + 03 ± 4.05e + 01	3.00e + 03 ± 6.35e + 01
f_{29}	3.26e + 03 ± 9.56e + 01	3.14e + 03 ± 9.52e + 01
f_{30}	3.60e + 03 ± 8.94e + 01	3.47e + 03 ± 6.87e + 01
The number of optimal values	21	18

TABLE 7: The experimental results between ICSO and DMSDL-QBSA.

Function	Term	ICSO	DMSDL-QBSA
f_1	Max	1.2436e + 03	9.6209e + 08
	Min	102.9899	1.7687e + 05
	Mean	447.9202	1.9320e + 06
	Var	7.9873e + 04	1.3093e + 07
f_2	Max	226.3723	1.7326e + 10
	Min	200.0070	1.4074e + 05
	Mean	206.8620	4.8412e + 07
	Var	51.1127	4.2984e + 08
f_3	Max	300	5.3828e + 06
	Min	300	6.3986e + 02
	Mean	300	3.0848e + 03
	Var	3.2679e - 25	8.0572e + 04
f_4	Max	436.1827	4.3338e + 03
	Min	400.0000	4.1267e + 02
	Mean	406.3343	4.3261e + 02
	Var	153.6519	8.5548e + 01
f_5	Max	520.2256	5.2110e + 02
	Min	500	5.2007e + 02
	Mean	518.7619	5.2014e + 02
	Var	26.0152	9.9700e - 02
f_6	Max	603.1944	6.1569e + 02
	Min	600	6.0257e + 02
	Mean	601.0094	6.0358e + 02
	Var	0.9590	1.3117e + 00
f_7	Max	700.3289	9.3907e + 02
	Min	700.0246	7.0069e + 02
	Mean	700.1142	7.0290e + 02
	Var	0.0038	1.3815e + 01
f_8	Max	800.9950	9.3391e + 02
	Min	800	8.0877e + 02
	Mean	800.1327	8.1676e + 02
	Var	0.1183	9.7968e + 00
f_9	Max	908.9546	1.0366e + 03
	Min	900.9950	9.1537e + 02
	Mean	904.9416	9.2335e + 02
	Var	3.3100	9.3860e + 00
f_{10}	Max	1.2722e + 03	3.4795e + 03
	Min	1.0151e + 03	1.4316e + 03
	Mean	1.1078e + 03	1.6111e + 03
	Var	8.5848e + 03	2.2195e + 02
f_{11}	Max	1.6046e + 03	3.7357e + 03
	Min	1.1119e + 03	1.2811e + 03
	Mean	1.2712e + 03	1.4869e + 03
	Var	2.0085e + 04	2.6682e + 02
f_{12}	Max	1.2003e + 03	1.2044e + 03
	Min	1.2000e + 03	1.2017e + 03
	Mean	1.2001e + 03	1.2018e + 03
	Var	0.0062	2.1603e - 01
f_{13}	Max	1.3001e + 03	1.3061e + 03
	Min	1.3000e + 03	1.3004e + 03
	Mean	1.3001e + 03	1.3006e + 03
	Var	7.4041e - 04	2.9913e - 01
f_{14}	Max	1.4001e + 03	1.4493e + 03
	Min	1.4000e + 03	1.4005e + 03
	Mean	1.4000e + 03	1.4009e + 03
	Var	2.7722e - 04	2.8527e + 00

TABLE 7: Continued.

Function	Term	ICSO	DMSDL-QBSA
f_{15}	Max	1.5015e+03	1.9050e+05
	Min	1.5004e+03	1.5026e+03
	Mean	1.5008e+03	1.5816e+03
	Var	0.0731	2.7233e+03
f_{16}	Max	1.6026e+03	1.6046e+03
	Min	1.6001e+03	1.6019e+03
	Mean	1.6014e+03	1.6024e+03
	Var	0.3295	3.6883e-01
f_{17}	Max	3.4698e+03	8.4770e+06
	Min	1.7150e+03	2.0097e+03
	Mean	2.2377e+03	2.1095e+04
	Var	1.9496e+05	2.0329e+05
f_{18}	Max	1.1217e+04	6.6050e+08
	Min	1.8094e+03	1.8288e+03
	Mean	3.2529e+03	1.9475e+05
	Var	5.4030e+06	1.0139e+07
f_{19}	Max	1.9025e+03	1.9555e+03
	Min	1.9000e+03	1.9028e+03
	Mean	1.9011e+03	1.9036e+03
	Var	0.6471	1.4209e+00
f_{20}	Max	2.0095e+03	1.2708e+08
	Min	2.0003e+03	2.0241e+03
	Mean	2.0025e+03	4.5834e+04
	Var	3.7953	2.1988e+06
f_{21}	Max	2.2435e+03	2.6897e+07
	Min	2.1000e+03	2.2314e+03
	Mean	2.1591e+03	7.4587e+03
	Var	3.7630e+03	2.8735e+05
f_{22}	Max	2.2207e+03	3.2211e+03
	Min	2.2000e+03	2.2314e+03
	Mean	2.2064e+03	2.2687e+03
	Var	85.7389	5.1234e+01
f_{23}	Max	2.6295e+03	2.9923e+03
	Min	2500	2.5000e+03
	Mean	2.6208e+03	2.5015e+03
	Var	1.0788e+03	1.7156e+01
f_{24}	Max	2.5213e+03	2.6369e+03
	Min	2.5088e+03	2.5251e+03
	Mean	2.5149e+03	2.5338e+03
	Var	12.8783	1.1715e+01
f_{25}	Max	2.7015e+03	2.7327e+03
	Min	2.6000e+03	2.6635e+03
	Mean	2.6657e+03	2.6894e+03
	Var	1.2703e+03	1.2571e+01
f_{26}	Max	2.7002e+03	2.7116e+03
	Min	2.7000e+03	2.7002e+03
	Mean	2.7001e+03	2.7003e+03
	Var	9.1541e-04	3.5003e-01
f_{27}	Max	3.1009e+03	3.4144e+03
	Min	2.7004e+03	2.7054e+03
	Mean	2.8927e+03	2.7219e+03
	Var	1.9303e+04	6.1325e+01
f_{28}	Max	3.2796e+03	4.8154e+03
	Min	3.1597e+03	3.0000e+03
	Mean	3.1994e+03	3.0065e+03
	Var	1.2209e+03	5.3483e+01

TABLE 7: Continued.

Function	Term	ICSO	DMSDL-QBSA
f_{29}	Max	6.6281e + 03	6.4392e + 07
	Min	3.1309e + 03	3.3287e + 03
	Mean	3.4966e + 03	2.8663e + 04
	Var	5.1852e + 05	8.6740e + 05
f_{30}	Max	4.2010e + 03	1.1393e + 06
	Min	3.4416e + 03	3.6416e + 03
	Mean	3.5937e + 03	4.1746e + 03
	Var	2.8657e + 04	1.8202e + 04

TABLE 8: The comparison results of optimal solutions for the welded beam design.

Algorithms	x_1	x_2	x_3	x_4	$f(x)$
ICSO	0.205730	3.470474	9.036624	0.2057296	1.724852
HFPSO [4]	—	—	—	—	1.724852
EPSO [37]	—	—	—	—	1.724853
MBA [38]	0.205729	3.470493	9.036626	0.205729	1.724853

TABLE 9: The statistical results for the welded beam design.

Algorithms	Worst	Mean	Best	SD
ICSO	1.724852	1.724852	1.724852	6.7089e - 12
HFPSO [4]	1.974449	1.727370	1.724852	2.50e - 02
EPSO [37]	1.747220	1.728219	1.724853	5.62e - 03
MBA [38]	1.724853	1.724853	1.724853	6.94e - 19

5. Conclusion and Future Directions

To overcome the premature convergence problem of the standard CSO algorithm when solving complex optimization problems, an ICSO algorithm is proposed in this paper. For the chicks with the weakest optimization ability, we introduce the reproduction and elimination-dispersal operations of BFA to improve the deep searchability of the CSO algorithm. In addition, in order to improve the global convergence speed of the algorithm, the theory of PSO is integrated to construct a hybrid CSO algorithm. The experimental results show that the ICSO algorithm proposed in this paper can significantly improve the optimization accuracy and convergence performance.

The disadvantage of the proposed algorithm is that with the increase of the dimension of the optimization problem, the optimization ability of the algorithm will decrease, which makes it not suitable for dealing with large-scale optimization problems. Therefore, in the future research work, how to dynamically adjust the limited number of role updates in the chicken swarm according to the number of iterations and how to improve the individual position update formula for hens with relatively weak search ability to further improve the optimization ability still need further research. In addition, we will also consider applying the ICSO algorithm to deal with practical problems, such as path planning in logistics distribution, workshop scheduling, and land use forecast [39].

Data Availability

All data generated or analyzed during this study are included in this published article. Color versions of one or more of the figures in this paper are available from the corresponding authors upon reasonable request.

Conflicts of Interest

The authors declare that they do not have conflicts of interest.

Acknowledgments

This work is partially supported by Hainan Provincial Natural Science Foundation of China (620QN230), The Scientific Research Project of Colleges And Universities in Hainan Province of China (Hnky2020-3), and the National Natural Science Foundation of China (61877038).

References

- [1] Y. Li, Y. Chen, J. Zhong, and Z. Huang, "Niching particle swarm optimization with equilibrium factor for multi-modal optimization," *Information Sciences*, vol. 494, pp. 233–246, 2019.
- [2] A. Smirnova, B. Pidgeon, G. Chowell, and Y. Zhao, "The doubling time analysis for modified infectious disease Richards model with applications to COVID-19 pandemic," *Mathematical Biosciences and Engineering*, vol. 19, no. 3, pp. 3242–3268, 2022.
- [3] Y. H. Xu, X. Chen, H. G. Zhang et al., "Online identification of battery model parameters and joint state of charge and state of health estimation using dual particle filter algorithms," *International Journal of Energy Research*, vol. 46, no. 14, pp. 19615–19652, 2022.
- [4] I. B. Aydilek, "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems," *Applied Soft Computing*, vol. 66, pp. 232–249, 2018.

- [5] M. W. Zhang, Y. Chen, and J. Q. Lin, "A privacy-preserving optimization of neighborhood-based recommendation for medical-aided diagnosis and treatment," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10830–10842, 2021.
- [6] Y. P. Villa Acuna and Y. Sun, "An efficiency-improved genetic algorithm and its application on multimodal functions and a 2D common reflection surface stacking problem," *Geophysical Prospecting*, vol. 68, no. 4, pp. 1189–1210, 2020.
- [7] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, and W. A. Chaovalitwongse, "Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 718–731, 2019.
- [8] R. Y. Dong, *Research and Application of Meta-Heuristic Optimization Algorithms [D]*, Jilin University, Changchun, China, 2019.
- [9] D. P. F. Cruz, R. D. Maia, and L. N. de Castro, "A framework for the analysis and synthesis of Swarm Intelligence algorithms," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 33, no. 4, pp. 659–681, 2021.
- [10] Q. Li and S. Y. Li, "Optimization of artificial CNN based on swarm intelligence algorithm," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 4, pp. 6163–6173, 2021.
- [11] S. M. Ermakov and D. N. Semenchikov, "Genetic global optimization algorithms," *Communications in Statistics - Simulation and Computation*, vol. 51, no. 4, pp. 1503–1512, 2022.
- [12] Z. Liu and T. Nishi, "Strategy dynamics particle swarm optimizer," *Information Sciences*, vol. 582, pp. 665–703, 2022.
- [13] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control [R]," *IEEE Control Systems Magazine*, vol. 6, pp. 52–67, 2002.
- [14] X. L. Li, *A New Intelligent Optimization Method-Artificial Fish School algorithm*, Zhejiang University, Hangzhou, China, 2003.
- [15] W. Fei, C. Liu, and S. Hu, "Research on swarm intelligence optimization algorithm[J]," *The Journal of China Universities of Posts and Telecommunications*, vol. 27, no. 03, pp. 5–24, 2020.
- [16] H. Ren, X. C. Shen, and X. J. Jia, "A novel dual-biological-community swarm intelligence algorithm with a commensal evolution strategy for multimodal problems," *The Journal of Supercomputing*, vol. 77, no. 10, pp. 10850–10895, 2021.
- [17] S. N. Chegini, A. Bagheri, and F. Najafi, "PSOSCALF: a new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems," *Applied Soft Computing*, vol. 73, pp. 697–726, 2018.
- [18] J. N. Zhang, K. W. Xia, Z. P. He, and S. Fan, "Dynamic multi-swarm differential learning quantum bird swarm algorithm and its application in random forest classification model," *Computational Intelligence and Neuroscience*, vol. 2020, no. 4, Article ID 6858541, 24 pages, 2020.
- [19] Y. Y. Su, L. J. Han, H. M. Wang, and J. Wang, "The workshop scheduling problems based on data mining and particle swarm optimisation algorithm in machine learning areas," *Enterprise Information Systems*, vol. 16, no. 2, pp. 363–378, 2022.
- [20] J. H. Mou, P. Y. Duan, L. Gao, X. Liu, and J. Li, "An effective hybrid collaborative algorithm for energy-efficient distributed permutation flow-shop inverse scheduling," *Future Generation Computer Systems*, vol. 128, pp. 521–537, 2022.
- [21] Y. Yongbin, Y. Chenyu, D. Quanxin, N. Tashi, L. Shouyi, and Z. Chen, "Memristive network-based genetic algorithm and its application to image edge detection," *Journal of Systems Engineering and Electronics*, vol. 32, no. 5, pp. 1062–1070, 2021.
- [22] N. Ai, B. Wu, B. Y. Li, and Z. Zhao, "5G heterogeneous network selection and resource allocation optimization based on cuckoo search algorithm," *Computer Communications*, vol. 168, pp. 170–177, 2021.
- [23] A. M. Altabeeb, A. M. Mohsen, and A. Ghallab, "An improved hybrid firefly algorithm for capacitated vehicle routing problem[J]," *Applied Soft Computing*, vol. 84, pp. 1568–4946, 2019.
- [24] J. W. Wang, J. W. Tian, and X. Zhang, "Control of time delay force feedback teleoperation system with finite time convergence[J]," *Frontiers in Neurorobotics*, vol. 16, pp. 1–9, 2022.
- [25] X. B. Meng, Y. Liu, X. Z. Gao, and H. Zhang, "A new bio-inspired algorithm: chicken swarm optimization[C]," in *Proceedings of the 5th International Conference on Swarm Intelligence*, pp. 86–94, Ostrava, Czech Republic, September 2014.
- [26] S. Deb, X. Z. Gao, K. Tammi, K. Kalita, and P. Mahanta, "A novel chicken swarm and teaching learning based algorithm for electric vehicle charging station placement problem," *Energy*, vol. 220, Article ID 119645, 2021.
- [27] S. Deb, X. Z. Gao, K. Tammi, K. Kalita, and P. Mahanta, "Recent studies on chicken swarm optimization algorithm: a review (2014–2018)," *Artificial Intelligence Review*, vol. 53, no. 3, pp. 1737–1765, 2020.
- [28] J. H. Liang, L. F. Wang, and M. Ma, "A new image segmentation method based on the ICSO-ISPCNN model," *Multimedia Tools and Applications*, vol. 79, no. 37–38, pp. 28131–28154, 2020.
- [29] D. R. Cristin, K. Suresh, and D. P. Anbhazhagan, "Severity level classification of brain tumor based on MRI images using fractional-chicken swarm optimization algorithm[J]," *The Computer Journal*, no. 10, p. 10, 2021.
- [30] B. Li, G. J. Shen, and G. Sun, "Improved chicken swarm optimization algorithm[J]," *Journal of Jilin University (Engineering and Technology Edition)*, vol. 49, no. 4, pp. 1339–1344, 2019.
- [31] K. D. Sampath, "Enhanced energy steady clustering using convergence node based path optimization with hybrid chicken swarm algorithm in MANET," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 20, pp. 767–788, 2018.
- [32] Y. C. Gu, H. Y. Lu, X. Lei, and S. Wanqiang, "Adaptive simplified chicken swarm optimization based on inverted S-shaped inertia weight[J]," *Chinese Journal of Electronics*, vol. 31, no. 2, pp. 343–362, 2022.
- [33] M. Lin, Y. W. Zhong, J. Lin, and X. Lin, "Enhanced chicken swarm optimisation for function optimisation problem," *International Journal of Wireless and Mobile Computing*, vol. 15, no. 3, pp. 258–269, 2018.
- [34] S. Deb, K. Tammi, X. Z. Gao, and K. Kalita, "A hybrid multi-objective chicken swarm optimization and teaching learning based algorithm for charging station placement problem[J]," *IEEE Access*, vol. 8, pp. 92573–92590, 2020.
- [35] J. J. Liang, B. Y. Qu, and P. N. Suganthan, *Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical optimization[C]*, IEEE Congress on Evolutionary Computation, Beijing, China, 2014.
- [36] X. Y. Song, M. Zhao, Q. F. Yan, and S. Xing, "A high-efficiency adaptive artificial bee colony algorithm using two strategies for continuous optimization," *Swarm and Evolutionary Computation*, vol. 50, Article ID 100549, 2019.
- [37] T. T. Ngo, A. Sadollah, and J. H. Kim, "A cooperative particle swarm optimizer with stochastic movements for

- computationally expensive numerical optimization problems,” *Journal of Computational Science*, vol. 13, pp. 68–82, 2016.
- [38] A. Sadollah, A. Bahreininejad, H. Eskandar, and M Hamdi, “Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems,” *Applied Soft Computing*, vol. 13, no. 5, pp. 2592–2612, 2013.
- [39] L. F. Xu, X. Liu, D. Tong, Z. Liu, L. Yin, and W. Zheng, “Forecasting urban land use change based on cellular automata and the PLUS model,” *Land*, vol. 11, no. 652, pp. 1–16, 2022.