Comprehensive Learning Particle Swarm Optimization Based on Optimal Particle Recombination

Xiaobin Chen, Kezong Tang, and Lihua Yang

Abstract—**Particle swarm optimization (PSO) algorithm has been widely used in large-scale complex problems such as resource allocation in recent years because of its simple implementation and easy operation. However, the slow convergence speed and low solution accuracy of the algorithm also restrict its further applications. To solve the above problems, this paper introduces the chromosome crossing characteristics of genetic algorithm (GA), and proposes a comprehensive learning particle swarm optimization based on optimal particle recombination. With the help of the ability of comprehensive learning strategy to efficiently explore the solution space, this method organically combines the excellent information explored by each particle through the optimal particle recombination, so as to obtain a better individual, speed up the convergence of the algorithm, and improve the solution accuracy of the problem. The experimental results of benchmark function show that the proposed algorithm has faster convergence speed and optimization accuracy than the original algorithm, and the results of Friedman test and Wilcoxon signed-rank test prove the feasibility of the optimal particle recombination operation in particle swarm optimization.**

Index Terms—**Particle swarm optimization, genetic algorithm, comprehensive learning strategy, optimal particle recombination, Friedman test, Wilcoxon signed-rank test**

I. INTRODUCTION

Particle swarm optimization (PSO) is a heuristic optimization algorithm based on artificial life, bird foraging, and other swarm intelligence behaviors of biological activities in nature [1−3]. Compared with other heuristic search algorithms, PSO has been successfully applied to many optimization problems in the real world, such as machine learning, photovoltaic parameter identification, resource scheduling, allocation, etc., because of its easy implementation and simple operation [4−10].

However, the heuristic search characteristics of PSO make the algorithm have some problems such as premature convergence and low solution accuracy [11]. In order to solve these problems, some scholars have made a lot of improvements in particle behavior, particle performance evaluation, and adaptive adjustment of inertia weight [12]. In the basic PSO algorithm, the inertia weight is proposed to adjust the exploration and exploitation progress of the algorithm, which has become a classic improved algorithm template. Eberhart and Shi [13] unified the exploration and exploitation tasks by analyzing the trajectory of particles moving in discrete time and developing it into continuous time, and used fixed parameter contraction factor instead of inertia weight to improve the optimization ability of the algorithm. In Refs. [14−17], researchers discuss the particle dynamic tracking and fuzzy system of PSO, and study the influence of different population topological structures on the optimization ability of PSO combined with the connection relationship between individual and collective particles in particle population, respectively.

In addition, communication learning among particle populations has become a popular direction of improvement in recent years. Liang et al. [18] tried to update the particle position from the excellent information of the historical optimal position of other particles in the population, and proposed a comprehensive learning particle swarm optimization (CLPSO). The experiment shows that this strategy can keep the diversity of the population for a long time, explore the search space more fully, and avoid the premature convergence of the population. Yu and Zhang [19] added a dynamic disturbance term when updating the particle velocity on the basis of CLPSO, so as to make the best use of particle swarm exploitation and exploration, thus overcoming the problem of CLPSO's solution accuracy. Lynn and Suganthan [20] used a comprehensive learning strategy to generate exploration and exploitation sub-populations, which not only kept the performance of algorithm development fully, but also maintained the diversity of particle swarms. Li et al. [21] used an information sharing mechanism (ISM) to share the information among particles to enhance the communication among particles and improve their search ability. On the other hand, it is a good improvement method to analyze the movement behavior of PSO. In order to improve the convergence speed and global search ability of PSO, Ref. [22] adaptively adjusts the inertia weight and speed update mode of particle swarm by dynamically evaluating the distribution and fitness of particle swarm. Tang et al. [23] proposed a double center particle swarm optimization (DCPSO) by analyzing the motion state of particles, which

Manuscript received: 30 March 2023; revised: 10 August 2023; accepted: 21 December 2023. *(Corresponding author: Kezong Tang.)*

Citation: X. Chen, K. Tang, and L. Yang, Comprehensive learning particle swarm optimization based on optimal particle recombination, *IJICS*, 2024, 29(1), 21–29.

Xiaobin Chen, Kezong Tang, and Lihua Yang are with School of Information Engineering, Jingdezhen Ceramic University, Jingdezhen 333403, China (e-mail: chenxuebin_mail@foxmail.com; tangkezong@126.com; yanglhjci@foxmail.com).

Xiaobin Chen is also with School of Computer Science and Engineering, Guangzhou Institute of Science and Technology, Guangzhou 510540, China.

Digital Object Identifier 10.62678/IJICS202403.10105

can improve the convergence speed and accuracy of the algorithm without increasing the complexity of the algorithm.

Besides analyzing the heuristic characteristics of PSO to improve it, many scholars also get inspiration from other algorithms. Jakubik et al. [24] used Gaussian pre-process fitting method to predict the optimal solution direction of the current problem optimization space, and guided the particle swarm search, speeding up the search speed of algorithm and enhancing the diversity of particle swarm. Liu et al. [25] adjusted the acceleration coefficient adaptively through a weighting strategy of sigmoid function to improve the convergence speed of the algorithm.

Genetic algorithm (GA) [26] is an optimization algorithm developed by referring to the phenomena in evolutionary biology, such as heredity, mutation, natural selection, and hybridization. Compared with PSO, this algorithm can maintain chromosome diversity through mutation, and at the same time carry out gene recombination through crossover operation to obtain better results. Therefore, many scholars will use the related concepts of genetic algorithm for reference to improve particle swarm optimization. Another dynamic multi-population particle swarm optimization with recombination learning and hybrid mutation is proposed by reference to GA in Ref. [27], and the optimization ability of the algorithm is improved by recombination learning and hybrid mutation. Gong et al. [28] put forward a genetic learning particle swarm optimization algorithm based on the concept of GA individual reproduction and the social learning operation of PSO particles.

The comprehensive learning strategy of CLPSO can better explore the whole problem space in the search process, but the slow convergence rate has become the deficiency of the further development of the algorithm. In order to improve the solution accuracy and convergence speed of CLPSO, this paper proposes a CLPSO based on optimal particle recombination (OPR) by referring to the concept of chromosome crossing in GA. The algorithm makes use of CLPSO's outstanding global exploration ability to effectively explore the excellent spatial information and save it in each particle. By applying the proposed optimal particle recombination method, the excellent information found by each particle in the search space is organically combined, so as to produce better results, thus improving the convergence speed and solution accuracy of the algorithm, and at the same time, improving the efficient utilization of the excellent information in the particle swarm.

II. Basic Theories

A. PSO

PSO is a heuristic optimization method that combines particle social learning behavior and group cooperation to realize problem solving. In this algorithm, each particle in the population is abstracted as a particle with the characteristics of velocity and position. With the optimization process, each particle updates its velocity and position according to Eqs. (1) and (2) under the guidance of the global optimal extremum and individual extremum.

$$V_i = V_i + c_1 r_1 \times (\text{pbest}_i - X_i) + c_2 r_2 \times (\text{gbest} - X_i) \tag{1}$$

In Eq. (1), several key elements constitute the operational framework of the particle swarm optimization algorithm. Here, V_i symbolizes the velocity of particle i, determining its traversal pace and direction through the solution landscape. Concurrently, X_i denotes the instantaneous location of particle i at any given step within the algorithm's progression

Embedded within each particle's strategy is its personal best position, pbest_i, a critical piece of information storing the most favorable location encountered by particle i throughout its search endeavor. On a higher level, the global best position, gbest, acts as a guiding beacon for the entire swarm, representing the peak of collective achievement and steering the particles collectively towards the globally optimal solution by embodying the best position found across the population.

To facilitate this dynamic navigation, two acceleration constants c_1 and c_2 serve as learning rates, with c_1 encouraging movement towards the individual pbest_i and c_2 promoting alignment towards the swarm's gbest. This dual mechanism fosters both individual exploitation and social exploration. Additionally, randomness is injected into the system through two uniformly distributed random numbers, r_1 and r_2, each ranging within the interval [0, 1], ensuring a balance between deterministic movements and stochastic explorations, thereby enhancing the algorithm's adaptability and search efficiency.

$$X_i = X_i + V_i \tag{2}$$

The guidance of pbest and gbest to particles enables them to explore extensively in the search space, and the acceleration coefficients of pbest and gbest respectively represent the influence degree of particle velocity by them. In order to further balance the global search ability and local search ability of particle swarm optimization, Shi and Eberhart [12] introduced the inertia weight ω on the basis of Eq. (1), and used Eq. (3) to update the speed. This improvement greatly improved the optimization ability of the algorithm, and at the same time it became the improved standard algorithm today.

$$V_i = \omega V_i + c_1 r_1 \times (\text{pbest}_i - X_i) + c_2 r_2 \times (\text{gbest} - X_i) \tag{3}$$

B. GA

GA is an optimization model inspired by evolution. This algorithm encodes the potential solutions of specific problems on simple chromosome-like data structures, and applies the recombination algorithm to these structures to retain key information. GA initializes a certain number of chromosomes randomly, evaluates these structures in a specific way, and allocates reproductive opportunities in such a way that those chromosomes that represent better solutions to target problems have more "reproductive" opportunities than those with poor solutions. The "advantages" of the solution are usually defined according to the current chromosome. In GA, there are mainly the following three operations: selection, crossover, and mutation. At the same time, there are a lot of optional methods in the above three operations. In the selection operation, you can choose roulette, Boltzmann selection [29] and other methods. There are also a lot of crossover technologies in crossover operation, but the mainstream can be divided into two types: single-point crossover and double-point crossover, as shown in Fig. 1.

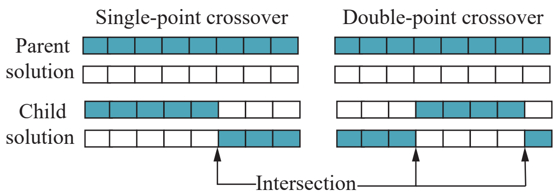


Figure 1 Schematic diagram of interlaced operation.

After the crossover operation is completed, GA will mutate with low probability for every bit in the population. In real-valued functions optimization, the length of GA chromosome is closely related to the accuracy of solution. In addition, in the process of optimization, the chromosome coding and decoding operation is required when evaluating the optimization scheme.

III. Comprehensive Learning Particle Swarm Optimization Based on Optimal Particle Recombination

A. Comprehensive Learning Strategy

In classical PSO, the trajectory of each particle is affected by both global extremum and individual extremum. Because the movement of the whole particle population will be affected by the global extreme value, the particle swarm in the classical PSO will converge to the vicinity of gbest quickly, which will lead to premature convergence of the particle swarm, and it is impossible to explore a wider area of the search space more widely. Therefore, the classical PSO can get a high-precision solution in the single-mode function relatively quickly. However, in the multi-mode function optimization, too fast convergence may cause the algorithm to fall into the local optimum, so that it cannot get a more accurate solution. In order to give PSO a better optimization ability in multi-modal functions, the particle velocity is updated by Eq. (4).

$$v_i^d = \omega v_i^d + c r_i^d \times \left(\text{pbest}_{f_i(d)}^d - x_i^d\right) \tag{4}$$

where c is the comprehensive learning factor, v_i^d, x_i^d, and r_i^d represent the flight velocity, position, and random number of the i-th particle in the d-th dimension, and $f_i(d)$ indicates that pbest represents the optimal value of the evaluation function f for the i-th particle, respectively.

At this time, the update of particles is no longer affected by the global optimal gbest, but guided by $\text{pbest}_{f_i(d)}^d$, where the calculation rule of $\text{pbest}_{f_i(d)}^d$ is shown in Eq. (5). By using the new speed update method to control the particle swarm to search for the optimized space, each particle can explore the surrounding area more fully.

$$\text{pbest}_{f_i(d)}^d = \begin{cases} \text{pbest}_i^d, & r_i^d < \text{Pc}_i; \\ \text{pbest}_j^d, & \text{otherwise} \end{cases} \tag{5}$$

where pbest_i^d represents the personal best position of particle i in dimension d, while pbest_j^d represents the personal best position of another particle j in dimension d.

In Eq. (5), the choice of $\text{pbest}_{f_i(d)}^d$ is determined by the probability Pc_i. When the random probability r_i^d is greater than Pc_i, it is guided by the historical optimal position of the current particle itself, otherwise it is guided by the historical optimal position of other particles. When the particle is extensively explored, but the individual optimal position is not updated, it means that the particle updated by Eq. (4) may fall into local optimum. Therefore, when the number of time of stopping update reaches the stagnation coefficient m, the particle velocity update becomes Eq. (3), and the current particle is guided to other areas for extensive search through the global optimal position.

B. Optimal Particle Recombination Operation

In PSO population, individual particles are independent and autonomous, and their behaviors will be influenced by the global optimal particles, but the possible optimal information cannot be shared among particles, so the optimization efficiency and accuracy will be affected to some extent. In GA, the corresponding positions of two outstanding chromosomes can be exchanged by chromosome crossing, so that outstanding individuals can exchange information directly. Therefore, in order to further improve the convergence speed and optimization accuracy of particle swarm optimization, this paper proposes an optimal particle recombination operation based on the social learning mode of PSO, which occurs between optimal particles. As this seed contains the best information of each particle and the population, it is possible to recombine the best information with a high probability through the recombination operation, thus obtaining a better optimal position. The reorganization execution equation is shown in Eq. (6), where p_{cross1} and p_{cross2} are the offspring results of reorganization of the current optimal position p_i and the random individual historical optimal position p_r, respectively.

$$p_i = \arg\min\{f(p_i), f(p_r), f(p_{\text{cross1}}), f(p_{\text{cross2}})\} \tag{6}$$

where f represents the corresponding evaluation function.

In the recombination operation, in order to enrich the diversity of recombination operations and better recombine the excellent information in the population, we select the type of recombination operation through a random probability r. When the random probability r is greater than 0.5, we perform single-point cross recombination, and when the random probability r is less than or equal to 0.5, we perform double-point cross recombination. The pseudo-code of OPR is shown in Algorithm 1.

Algorithm 1 OPR pseudo-code

if $r > 0.5$ **then**
| $[p_{\text{cross1}}, p_{\text{cross2}}]$ = SinglePointCross (p_i, p_r);
else
| $[p_{\text{cross1}}, p_{\text{cross2}}]$ = DoublePointCross (p_i, p_r);
end
$p_i = \arg\min\{f(p_i), f(p_r), f(p_{\text{cross1}}), f(p_{\text{cross2}})\}$;
Finish

C. Particle Optimization Stagnation and Optimal Particle Recombination Adaptation

In the process of comprehensive learning, the particle still falls into the local optimal area when exploring its surrounding area under the comprehensive learning strategy. At this time, the particle cannot jump out of the current area only through $\text{pbest}^d_{f_i(d)}$ guidance, so that it cannot continue to play its role of extensive exploration in space. Therefore, in order to solve this problem, the comprehensive learning strategy keeps the speed update mode of classical particles, and sets the stagnation coefficient m. When a particle stops updating more than this value, the particle will update according to Eq. (3), and the current particle state will be activated by the guidance of the global optimal particle, so that the particle can continue to search in space.

When adapting the optimal particle recombination operation, we first need to clarify the purpose of the operation. Because the optimal operation is to make the excellent information in the particle swarm exchange with each other during the recombination process, we should try our best to make the particle swarm complete full exploration, and then combine the excellent information of each particle through the recombination operation to obtain better results. In this paper, after a lot of tests, we recommend that when the number of iterations is between $[0.6T, 0.8T]$ (where T represents the maximum number of iterations for the algorithm to run), the optimal particle recombination should be started.

In addition, in order to further optimize the performance of the algorithm and improve the optimization efficiency, in CLPSO-OPR, we set up two optimal particle recombination operations, one is the individual historical optimal position recombination and the other is the global historical optimal position recombination. The recombination of individual historical optimal particles occurs in the process of particle stagnation of comprehensive learning strategy. When a particle stagnates, it means that the area around the particle has been fully exploited, and the particle will retain the optimal information of the current area. Therefore, using gbest to recombine with the current particle will obtain the excellent information of the particle with a high probability, and at the same time greatly improve the global optimal position. The pseudo-code of the final comprehensive learning strategy is shown in Algorithm 2, in which N represents the number of particles in the particle swarm within the algorithm. The reorganization of the global optimal position takes place in the whole algorithm cycle. When the number of iterations of the algorithm is able to trigger the reorganization operation, the global optimal historical position will randomly select a particle optimal position for reorganization. At the same time, in this reorganization, the four positions after reorganization will be evaluated in turn, and the two optimal positions will be updated to pbest and gbest, which will speed up the population convergence and improve the optimization accuracy of the algorithm.

Algorithm 2 Comprehensive learning strategy pseudo-code

for $i = 1{:}N$ **do**
  if $m \leqslant \text{StopTimes}_i$ **then**
    $V_i = \omega V_i + c_1 r_1 (\text{pbest}_i - X_i) + c_2 r_2 (\text{gbest} - X_i)$;
    Execute OPR;
    $\text{StopTimes}_i = 0$;
  else
    for $d = 1$:dimension **do**
      if $r_i^d > \text{Pc}_i$ **then**
        $\text{pbest}^d_{fi(d)} = \text{pbest}^d_i$;
      else
        X = randperm (N, 1);
        $\text{pbest}^d_{fi(d)} = \text{pbest}^d_j$;
      end
      $v_i^d = \omega v_i^d + c r_i^d (\text{pbest}^d_{fi(d)} - x_i^d)$;
      $x_i^d = x_i^d + v_i^d$
    end
    if $f(x_i) < f(\text{pbest}_i)$ **then**
      $\text{pbest}_i = x_i$;
      $\text{StopTimes}_i = 0$;
      if $f(x_i) < f(\text{gbest})$ **then**
        $\text{gbest} = x_i$;
      end
    else
      $\text{StopTimes}_i = \text{StopTimes}_i + 1$;
    end
  end
end
Finish

D. CLPSO-OPR

CLPSO-OPR improves the optimization ability of the algorithm in the multi-modal problem model by virtue of the excellent wide-area exploration ability of the comprehensive learning strategy, and reduces the risk of the algorithm falling into the local optimum. Combining the concept of chromosome crossing in genetic algorithm, we propose an optimal particle recombination method, and apply the optimal particle recombination in the process of comprehensive learning strategy and global optimal update. The optimal position information of each particle in its respective area is collected by the individual historical optimal particle recombination operation, which effectively improves the optimization efficiency. At the same time, in the late stage of global optimization, the global optimal position reorganization is used to quickly collect the optimal information in the population and summarize it to the global optimal position, so as to further improve the optimization accuracy and enable gbest to better guide other particles to search. Compared with the original algorithm, CLPSO-OPR does not increase the complexity of algorithm space and the difficulty of implementation, and its algorithm flow chart is shown in Fig. 2.

IV. Experimental Results and Analysis

A. Platform Construction and Preparation before Testing

The software environment and hardware environment used in this test are configured as follows: Windows 11 operating system, Matlab2020b running environment, Intel (R) Core (TM) i7-9700 CPU with the main frequency of 3.00 GHz,

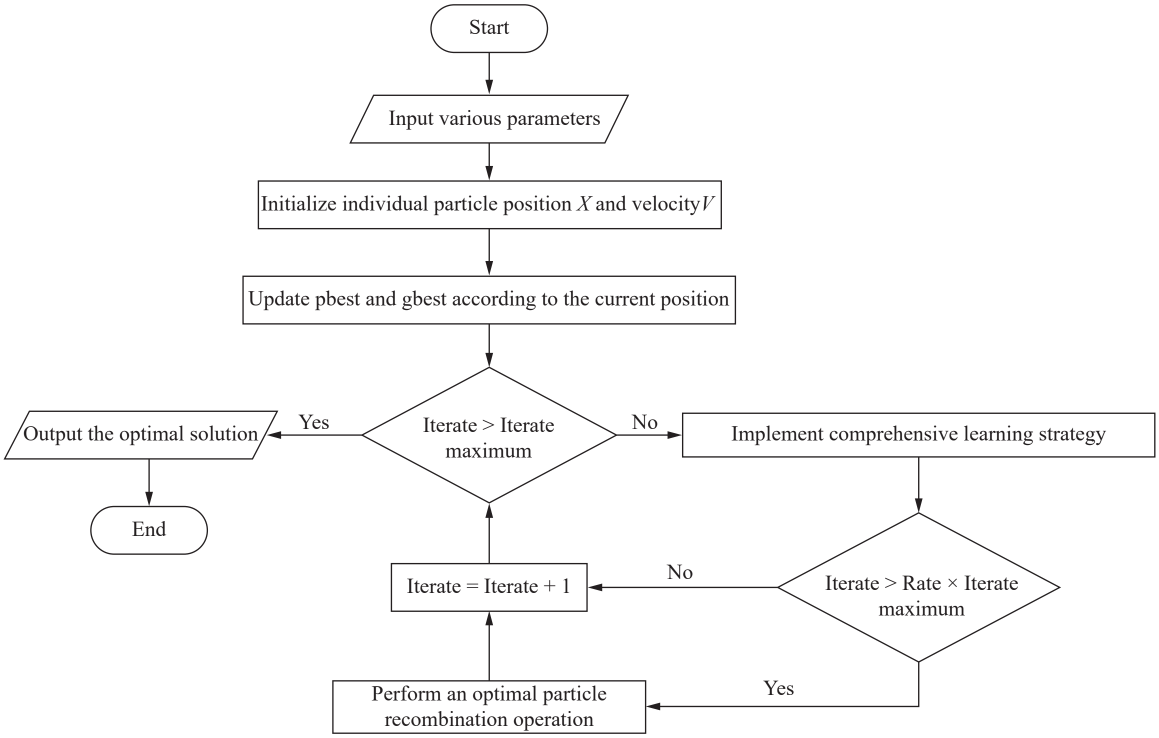


Figure 2 CLPSO-OPR execution flow chart.

8 GB of memory. The selected comparison algorithms are OPSO [12], VPSO [13], CLPSO [18], adaptive particle swarm optimization (APSO) [22], heterogeneous comprehensive learning particle swarm optimization (HCLPSO) [20], competitive and cooperative particle swarm optimization (CCPSO)-ISM [21], DCPSO [23], and adaptive weighting particle swarm optimization (AWPSO) [25], respectively. The algorithm parameters, particle population size, and the maximum number of iterations are shown in Table 1. Parts such as Sphere, Rosenbrock, and other classical optimization problems are selected as benchmark functions, and the details are shown in Table 2, in which rows 1−6 and rows 7−12 are unimodal and multi-modal functions, respectively.

In Table 2, D represents the solution dimension of the problem, and f^* is the theoretical optimal value of the corresponding benchmark function in the corresponding dimension.

In this paper, minimum (min), mean, and standard deviation (std) are selected as the basic evaluation indices, which can represent the best optimization ability, average optimization effect, and optimization stability of the algorithm, and can intuitively reflect the performance of each algorithm in the corresponding benchmark function. In addition, statistical test method is one of the key methods to test the performance of optimization algorithm, and more reliable conclusions can be drawn through tests [30]. In this paper, Friedman test is used to verify the differences among the calculation methods, and Wilcoxon signed-rank test is further used to verify the significance level of the differences between CLPSO-OPR and other comparison algorithms.

B. Analysis of Algorithm Test Results

According to the settings of the above experimental environment, each algorithm completed independent and repeated optimization experiments in all test functions, and the detailed results are shown in Table 3. It can be seen from the results in Table 3 that CLPSO-OPR is superior to the other seven improved PSO algorithms in three indices in the test

Table 1 Parameter setting of PSOs.

Algorithm	Related parameters	Population size (N)	Iterative upper limit
OPSO	$\omega = [0.9, 0.4]$, $c_1 = 2$, $c_2 = 2$	30	5000
VPSO	$\chi = 0.729844$, $c_1 = 2.05$, $c_2 = 2.05$	30	5000
CLPSO	$\omega = [0.9, 0.4]$, $c = 1.49445$, $c_1 = 2$, $c_2 = 2$, $m = 7$	30	5000
DCPSO	$\omega = [0.95, 0.30]$, $c_1 = 2$, $c_2 = 2$	30	5000
HCLPSO	$\omega = [0.99, 0.20]$, $c_1 = [2.5, 0.5]$, $c_2 = [0.5, 2.5]$, $c_t = [3.0, 1.5]$	30	5000
CCPSO-ISM	$\omega = 0.6$, $c = 2$, $p = 0.05$	30	5000
AWPSO	$\omega = [0.9, 0.4]$, $a = 0.000035s$, $b = 0.5$, $c_{hc} = 0$, $d = 1.5$	30	5000
CLPSO-OPR	$\omega = [0.9, 0.4]$, $c = 1.49445$, $c_1 = 2$, $c_2 = 2$, $m = 7$	30	5000

Note: In VPSO, χ represents the contraction factor. In HCLPSO, c_t stands for the time varying acceleration coefficient. In CCPSO-ISM, p is the cooperative behavior control parameter. In AWPSO, a signifies the steepness of the curve, s represents the search range of the optimization problem, b indicates the peak value of the curve, c_{hc} denotes the horizontal coordinate of the center point of the curve, and d is a normal numerical value, respectively.

Table 2 Information of benchmark function.

Type of function	Function name	Name of benchmark function	D	f^*	Search space
Unimodal	$f_1(x)$	Quadric	30	0	[−10, 10]
	$f_2(x)$	Sphere	30	0	[−5.12, 5.12]
	$f_3(x)$	Schwefel's P22	30	0	[−10, 10]
	$f_4(x)$	Rosenbrock	30	0	[−5, 10]
	$f_5(x)$	Sum squares	30	0	[−10, 10]
	$f_6(x)$	Step	30	0	[−100, 100]
Multi-modal	$f_7(x)$	Ackley	30	0	[−32, 32]
	$f_8(x)$	Michalewicz	10	−9.66	[0, π]
	$f_9(x)$	Schwefel	30	0	[−500, 500]
	$f_{10}(x)$	Dixon-price	30	0	[−10, 10]
	$f_{11}(x)$	Griewank	30	0	[−600, 600]
	$f_{12}(x)$	Rastrigin	30	0	[−5.12, 5.12]

Table 3 Comparison on basic test suite. Bold indicates the optimal value solved by all current algorithms.

Function	Criterion	Experiment result							
		OPSO	VPSO	CLPSO	DCPSO	HCLPSO	CCPSO-ISM	AWPSO	CLPSO-OPR
f_1	min	1.17E−03	1.32E−10	9.61E−04	1.62E−04	**3.65E−14**	3.79E+00	6.32E−06	3.75E−07
	mean	2.51E+00	2.56E+00	5.05E−03	2.53E−03	**2.63E−07**	8.39E+00	3.45E−01	4.82E−06
	std	7.35E+00	1.13E+01	3.90E−03	2.37E−03	**4.65E−06**	2.05E+00	5.96E+00	4.79E−06
f_2	min	2.63E−36	6.03E−92	9.25E−57	1.31E−61	**1.89E−100**	4.40E−25	1.47E−33	4.86E−71
	mean	4.36E−32	9.44E−77	4.66E−55	2.50E−55	**3.78E−80**	2.33E−21	7.41E−21	2.25E−67
	std	1.67E−31	8.54E−76	5.52E−55	2.08E−54	**3.74E−79**	1.48E−20	7.06E−20	3.20E−67
f_3	min	1.85E−22	1.85E−27	1.49E−32	1.30E−36	9.03E−18	9.20E−10	3.41E−13	**2.11E−40**
	mean	2.17E−19	1.29E−12	1.57E−31	7.31E−31	3.88E−09	2.66E−06	5.08E−07	**1.84E−39**
	std	1.02E−18	8.08E−12	1.07E−31	2.48E−30	2.08E−08	5.95E−06	2.18E−06	**1.75E−39**
f_4	min	4.83E+00	**1.24E−06**	3.12E−01	2.17E+01	1.19E−05	7.39E−01	2.86E−02	1.20E−04
	mean	1.37E+02	8.37E+01	4.76E+01	2.66E+01	3.60E+01	3.15E+01	3.66E+02	**1.21E−01**
	std	2.47E+02	3.67E+02	3.54E+01	7.23E−01	2.98E+01	7.07E−01	2.51E+03	**4.29E−01**
f_5	min	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	mean	**0.00E+00**	2.47E+00	**0.00E+00**	**0.00E+00**	6.00E−02	**0.00E+00**	4.00E−02	**0.00E+00**
	std	**0.00E+00**	4.55E+00	**0.00E+00**	**0.00E+00**	2.39E−01	**0.00E+00**	1.97E−01	**0.00E+00**
f_6	min	1.86E−34	4.68E−89	1.28E−54	1.32E−101	**7.77E−102**	5.59E−24	2.53E−33	1.41E−67
	mean	6.60E−31	6.37E−75	1.82E−53	2.56E−54	**7.85E−87**	9.76E−20	5.00E−20	1.27E−65
	std	1.92E−30	4.29E−74	1.94E−53	1.45E−53	**7.12E−86**	6.99E−19	4.49E−19	2.11E−65
f_7	min	**8.88E−16**	4.44E−15	4.44E−15	4.44E−15	2.22E−14	1.71E−10	**8.88E−16**	7.99E−15
	mean	4.44E−15	1.16E−02	4.44E−15	**4.34E−15**	1.22E−01	2.92E−07	4.48E−15	1.11E−14
	std	5.05E−16	1.16E−01	**0.00E+00**	7.00E−16	3.78E−01	1.71E−06	7.98E−16	3.55E−15
f_8	min	−5.35E+00	−4.88E+00	−4.47E+00	−7.53E+00	−4.72E+00	−8.07E+00	−5.63E+00	**−8.56E+00**
	mean	−3.46E+00	−3.51E+00	−3.18E+00	−6.15E+00	−3.36E+00	**−7.14E+00**	−3.46E+00	−6.62E+00
	std	5.47E−01	4.01E−01	4.42E−01	4.74E−01	4.54E−01	**3.36E−01**	5.31E−01	8.90E−01
f_9	min	3.30E+03	4.34E+03	3.00E+03	1.76E+03	3.59E+03	2.80E+03	3.44E+03	**3.82E−04**
	mean	5.28E+03	5.98E+03	4.45E+03	2.86E+03	4.82E+03	4.12E+03	5.25E+03	**3.62E+02**
	std	6.51E+02	8.15E+02	4.51E+02	4.31E+02	6.61E+02	3.98E+02	7.08E+02	**1.95E+02**
f_{10}	min	**6.67E−01**	**6.67E−01**	**6.67E−01**	**6.67E−01**	2.59E+03	**6.67E−01**	6.67E−01	**6.67E−01**
	mean	**6.67E−01**	**6.67E−01**	**6.67E−01**	**6.67E−01**	6.60E−01	**6.67E−01**	8.87E+00	**6.67E−01**
	std	1.67E−14	1.20E−15	**1.77E−16**	1.92E−16	6.67E−02	5.93E−03	4.66E+01	2.56E−16
f_{11}	min	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	mean	1.80E−02	3.46E−02	2.61E−03	1.17E−02	1.78E−02	1.45E−03	1.71E−02	**8.91E−04**
	std	1.94E−02	6.53E−02	5.23E−03	1.52E−02	2.33E−02	6.32E−03	1.80E−02	**6.36E−04**
f_{12}	min	2.49E+01	2.09E+01	9.95E−01	4.98E+00	1.39E+01	1.62E+01	2.19E+01	**0.00E+00**
	mean	4.40E+01	4.54E+01	5.99E+00	2.28E+01	3.35E+01	3.27E+01	4.54E+01	**1.95E+00**
	std	1.33E+01	1.12E+01	3.50E+00	6.65E+00	9.44E+00	1.12E+01	1.60E+01	**1.35E+00**

results of four benchmark functions f_3, f_9, f_{11}, and f_{12}, and the optimization accuracy, average performance, and stability of the algorithm are significantly improved. In functions f_1, f_2, and f_6, HCLPSO has better optimization performance. CCPSO-ISM shows better stability and average optimization ability in f_{11} than other improved PSOs. In the benchmark function f_5, all algorithms can get the best results, but OPSO, CLPSO, DCPSO, CCPSO-ISM, and CLPSO-OPR are more stable.

In order to analyze the convergence state of each algorithm conveniently, this paper makes the convergence result manifest according to Eq. (7). Through this operation, the convergence state can be analyzed intuitively while keeping the relative convergence speed unchanged.

$$\text{data}_{\text{show}} = \sqrt[k]{\text{data}_{\text{real}}} \tag{7}$$

where $\text{data}_{\text{real}}$ represents the real data, k stands for the square root constant, and $\text{data}_{\text{show}}$ represents the relative values used in plotting the reduced-order image, respectively.

It can be seen from the convergence state of each algorithm in the corresponding function in Fig. 3 that CLPSO-OPR has the best performance in convergence speed and accuracy in functions f_3, f_8, and f_9 within the specified iteration time, and HCLPSO performs better in function f_1, which can reach higher accuracy faster in the later period. In a word, the convergence speed of CLPSO-OPR is faster than that of CLPSO, and CLPSO-OPR has better optimization ability in multi-modal functions compared with other improved PSO algorithms.

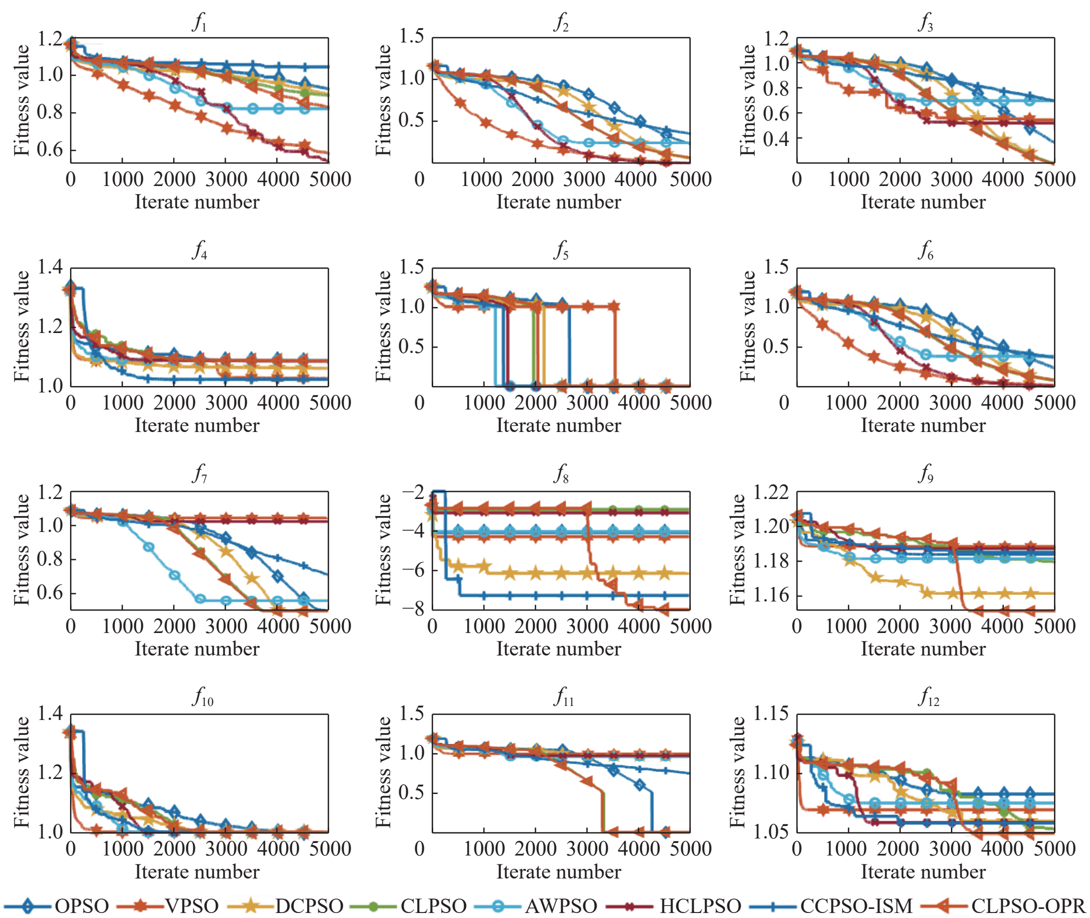


Figure 3 Convergence state diagram of each improved algorithm under a fixed number of iterations.

C. Comparative Analysis of Reliability Based on Friedman Test and Wilcoxon Signed-Rank Test

The Friedman statistical test results of each algorithm are shown in Table 4, in which CLPSO-OPR ranks first among the eight algorithms in benchmark optimization performance. The probability value calculated from the statistical data of Friedman test strongly indicates that there are significant differences among the eight algorithms. In addition, Wilcoxon signed-rank test results of CLPSO-OPR and other reference algorithms are shown in Table 5. The test results show that CLPSO-OPR shows a significant improvement of significance level $\alpha = 0.05$ compared with OPSO, VPSO, CLPSO, and AWPSO. Compared with DCPSO, HCLPSO, and CCPSO-ISM, the difference between CLPSO-OPR and DCPSO, HCLPSO, and CCPSO-ISM is not statistically significant, but it can be seen from the benchmark function test results that CLPSO-OPR, DCPSO, HCLPSO, and CCPSO-ISM have their own advantages, but compared with CLPSO, CLPSO-OPR has obvious improvement. The statistics of Friedman test and Wilcoxon signed-rank test verify the general validity of CLPSO-OPR on benchmark problems.

Table 4 Rank achieved by the Friedman test for basic test suite.

Algorithm	Friedman rank	Final rank
OPSO	5.9583	6
VPSO	6.4583	7
CLPSO	4.7083	3
DCPSO	3.2917	2
HCLPSO	4.9167	4
CCPSO-ISM	5.4167	5
AWPSO	7.2500	8
CLPSO-OPR	3.1250	1

Note: Probability value is the probability of observing a test statistic as extreme as, or more extreme than, the observed value under the null hypothesis. Small values of probability cast doubt on the validity of the null hypothesis. The probability value of the Friedman test is 0.0008.

Table 5 Wilcoxon signed-rank test results of basic test suite.

ECPSO vs.	*H* value	Probability value
OPSO	1	0.0136
VPSO	1	0.0048
CLPSO	1	0.0488
DCPSO	0	0.4921
HCLPSO	0	0.2036
CCPSO-ISM	0	0.3652
AWPSO	1	0.0024

V. Conclusion

According to the problems of slow convergence speed and low optimization accuracy of CLPSO algorithm, this paper proposes CLPSO-OPR. Compared with the original algorithm, CLPSO-OPR improves the optimization efficiency, solution accuracy, and convergence speed of the algorithm. Inspired by chromosome crossing in GA, this algorithm proposes an optimal particle recombination operation to improve the slow convergence speed of particle swarm, and at the same time gives a feasible method of excellent information transmission among particles, which keeps CLPSO's excellent multi-modal development ability and algorithm optimization stability, while further enhancing the optimization ability of the algorithm.

Experimental results show that CLPSO-OPR proposed in this paper has better optimization performance in benchmark functions compared with CLPSO, higher search accuracy in benchmark functions, and better average optimization effect and stability. Compared with other improved PSO algorithms, CLPSO-OPR has a better overall performance. The optimal particle recombination method provides a new idea for the improvement of current PSO, and also provides a new solution for practical optimization problem models such as neural network training and large-scale complex networks. In the future, we will further study the improved operation of the optimal particle recombination, so as to improve the optimization efficiency of the algorithm more efficiently.

Acknowledgment

This work was supported in part by the Scientific Research Program of the Jiangxi Provincial Department of Education (Nos. GJJ210731 and GJJ211331).

References

[1] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proc. International Conference on Neural Networks*, Perth, Australia, 1995, 1942–1948.

[2] R. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, in *Proc. 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, 39–43.

[3] J. Kennedy, Swarm intelligence, in *Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies*. New York, NY, USA: Springer, 2006, 187–219.

[4] J. Robinson and Y. Rahmat-Samii, Particle swarm optimization in electromagnetics, *IEEE Trans. Antennas Propag*., 2004, 52(2), 397–407.

[5] C. L. Huang and J. F. Dun, A distributed PSO–SVM hybrid system with feature selection and parameter optimization, *Appl. Soft Comput*., 2008, 8(4), 1381–1391.

[6] P. Chauhan, M. Pant, and K. Deep, Parameter optimization of multi-pass turning using chaotic PSO, *Int. J. Mach. Learn. Cybern*., 2015, 6(2), 319–337.

[7] A. Anurag, R. Priyadarshi, A. Goel, and B. Gupta, 2-D coverage optimization in WSN using a novel variant of particle swarm optimisation, in *Proc. 7th International Conference on Signal Processing and Integrated Networks* (*SPIN*), Noida, India, 2020, 663–668.

[8] H. G. G. Nunes, P. N. C. Silva, J. A. N. Pombo, S. J. P. S. Mariano, and M. R. A. Calado, Multiswarm spiral leader particle swarm optimisation algorithm for PV parameter identification, *Energy Convers. Manag*., 2020, 225, 113388.

[9] J. P. Dou, J. Li, D. Xia, and X. Zhao, A multi-objective particle swarm optimisation for integrated configuration design and scheduling in reconfigurable manufacturing system, *Int. J. Prod. Res*., 2021, 59(13), 3975–3995.

[10] F. Yan, T. T. Peng, and C. R. Shen, Time-space cluster based location-routing problem with capacitate constraints, *Control Decis*., 2021, 36(10), 2504–2510, (in Chinese).

[11] Y. Shi and R. C. Eberhart, Empirical study of particle swarm optimization, in *Proc. 1999 Congress on Evolutionary Computation-CEC99* (*Cat. No. 99TH8406*), Washington, DC, USA, 1999, 1945–1950.

[12] Y. Shi and R. Eberhart, A modified particle swarm optimizer, in *Proc. 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence* (*Cat. No. 98TH8360*), Anchorage, AK, USA, 1998, 69–73.

[13] R. C. Eberhart and Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in *Proc. 2000 Congress on Evolutionary Computation. CEC00* (*Cat. No. 00TH8512*), La Jolla, CA, USA, 2000, 84–88.

[14] Eberhart and Y. H. Shi, Particle swarm optimization: Developments, applications and resources, in *Proc. 2001 Congress on Evolutionary Computation* (*IEEE Cat. No. 01TH8546*), Seoul, Korea, 2001, 81–86.

[15] Y. H. Shi and R. C. Eberhart, Fuzzy adaptive particle swarm optimization, in *Proc. 2001 Congress on Evolutionary Computation* (*IEEE Cat. No. 01TH8546*), Seoul, Korea, 2001, 101–106.

[16] J. Kennedy and R. Mendes, Population structure and particle swarm performance, in *Proc. 2002 Congress on Evolutionary Computation. CEC'02* (*Cat. No. 02TH8600*), Honolulu, HI, USA, 2002, 1671–1676.

[17] M. Clerc and J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE Trans. Evol. Comput*., 2002, 6(1), 58–73.

[18] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE Trans. Evol. Comput*., 2006, 10(3), 281–295.

[19] X. Yu and X. Q. Zhang, Enhanced comprehensive learning particle swarm optimization, *Appl. Math. Comput*., 2014, 242, 265–276.

[20] N. Lynn and P. N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, *Swarm Evol. Comput*., 2015, 24, 11–24.

[21] Y. H. Li, Z. H. Zhan, S. J. Lin, J. Zhang, and X. N. Luo, Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems, *Inf. Sci.*, 2015, 293, 370–382.

[22] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, Adaptive particle swarm optimization, *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, 2009, 39(6), 1362–1381.

[23] K. Z. Tang, B. X. Liu, J. Y. Yang, and T. K. Sun, Double center particle swarm optimization algorithm, *J. Comput. Res. Dev.*, 2012, 49(5), 1086–1094, (in Chinese).

[24] J. Jakubik, A. Binding, and S. Feuerriegel, Directed particle swarm optimization with Gaussian-process-based function forecasting, *Eur. J. Oper. Res.*, 2021, 295(1), 157–169.

[25] W. B. Liu, Z. D. Wang, Y. Yuan, N. Y. Zeng, K. Hone, and X. H. Liu, A novel sigmoid-function-based adaptive weighted particle swarm optimizer, *IEEE Trans. Cybern.*, 2021, 51(2), 1085–1093.

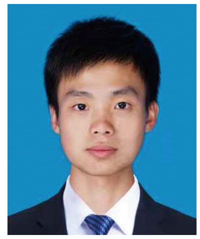
[26] S. Mirjalili, Genetic algorithm, in *Evolutionary Algorithms and Neural Networks: Theory and Applications*. Cham, Switzerland: Springer, 2019, 43–55.

[27] K. X. Tang, X. L. Liang, W. F. Zhou, Q. H. Ma, and Y. Zhang, Dynamic multi-population particle swarm optimization algorithm with recombined learning and hybrid mutation, *Control Decis.*, 2021, 36(12), 2871–2880, (in Chinese).

[28] Y. J. Gong, J. J. Li, Y. C. Zhou, Y. Li, H. S. H. Chung, Y. H. Shi, and J. Zhang, Genetic learning particle swarm optimization, *IEEE Trans. Cybern.*, 2016, 46(10), 2277–2290.

[29] D. E. Goldberg, A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing, *Complex Syst.*, 1990, 4(4), 445–460.

[30] J. Carrasco, S. García, M. M. Rueda, S. Das, and F. Herrera, Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review, *Swarm Evol. Comput.*, 2020, 54, 100665.



Xiaobin Chen received the BS degree in computer science and technology from Jingdezhen University, China, in 2020, and the ME degree from School of Information Engineering, Jingdezhen Ceramic University, China, in 2023. He is currently a full-time faculty member at School of Computer Science and Engineering, Guangzhou Institute of Science and Technology, China. His research interests include intelligent computing, deep learning, and image processing.

Kezong Tang received the BS degree in computer application from Jiangxi Normal University, China, in 2002, the MS degree in pattern recognition and intelligent systems from Jiangsu University of Science and Technology, China, in 2008, the PhD degree in control science from Nanjing University of Science and Technology, China, in 2012, and the postdoctoral degree from Southeast University, China, in 2016. He was a visiting scholar at Department of Electronic Engineering, University of Houston, USA, in 2010 and 2011. He is currently a professor and deputy director at Jingdezhen Ceramic University, China. His research interests include intelligent computing, pattern recognition, and intelligent systems.

Lihua Yang received the BS degree in information and computing science and the MS degree in probability theory and mathematical statistics from Jingdezhen Ceramic University, China, in 2003 and 2011, respectively. He was a visiting scholar at School of Computer Science, Wuhan University, China, in 2013 and 2014. He is currently an associate professor in information and computational science at Jingdezhen Ceramic University, China. His research interests include computer vision, natural language processing, and intelligent computing.