Article

Hybrid Particle Swarm Optimization Algorithm Based on the Theory of Reinforcement Learning in Psychology

Wenya Huang [1,2], Youjin Liu [1] and Xizheng Zhang [2,*]

[1] School of Business, Hunan University of Science and Technology, Xiangtan 411201, China
[2] School of Artificial Intelligence, Hunan Institute of Engineering, Xiangtan 411104, China
[*] Correspondence: zxz@hnie.edu.cn

Abstract: To more effectively solve the complex optimization problems that exist in nonlinear, high-dimensional, large-sample and complex systems, many intelligent optimization methods have been proposed. Among these algorithms, the particle swarm optimization (PSO) algorithm has attracted scholars' attention. However, the traditional PSO can easily become an individual optimal solution, leading to the transition of the optimization process from global exploration to local development. To solve this problem, in this paper, we propose a Hybrid Reinforcement Learning Particle Swarm Algorithm (HRLPSO) based on the theory of reinforcement learning in psychology. First, we used the reinforcement learning strategy to optimize the initial population in the population initialization stage; then, chaotic adaptive weights and adaptive learning factors were used to balance the global exploration and local development process, and the individual optimal solution and the global optimal solution were obtained using dimension learning. Finally, the improved reinforcement learning strategy and mutation strategy were applied to the traditional PSO to improve the quality of the individual optimal solution and the global optimal solution. The HRLPSO algorithm was tested by optimizing the solution of 12 benchmarks as well as the CEC2013 test suite, and the results show it can balance the individual learning ability and social learning ability, verifying its effectiveness.

Keywords: particle swarm algorithm; psychological enhancement theory; adaptive; mutation

Citation: Huang, W.; Liu, Y.; Zhang, X. Hybrid Particle Swarm Optimization Algorithm Based on the Theory of Reinforcement Learning in Psychology. *Systems* **2023**, *11*, 83. https://doi.org/10.3390/systems11020083

Academic Editor: Vladimír Bureš

Received: 8 January 2023
Revised: 25 January 2023
Accepted: 2 February 2023
Published: 6 February 2023

1. Introduction

In order to more effectively solve problems in many fields in real life, scholars mathematically model them; that is, they establish an optimization model [1]. In the process of this mathematical modeling, it is found that some problems are difficult to accurately model or solve. To facilitate the solution of traditional methods, the target usually needs to be processed, which increases the complexity of the problem [2]. The intelligent optimization method does not have this limitation and can solve the target model more conveniently [3,4]. Li et al. [5], Dokeroglu et al. [6] and Xue et al. [7] presented some comprehensive surveys of the state-of-the-art schemes on intelligent optimization for feature selection, which is helpful for optimization performance. Therefore, intelligent optimization methods have developed rapidly. Intelligent optimization methods include the genetic algorithm (GA) [8], the artificial bee colony algorithm (ABC) [9], the simulated annealing algorithm (SA) [10], the particle swarm optimization (PSO) algorithm [11], etc. Among them, PSO has attracted scholars' attention because of its simple structure and easy implementation [12].

PSO was first proposed by Kennedy and Eberhart [12]. The initially proposed optimization effect of PSO was unexceptional. Later, scholars usually attempted to improve the inertia weight ω with a nonfixed value in the PSO, and the particle renewal formula was first proposed by Yuhui and Eberhart [13]. Subsequent scholars have carried out a lot of research regarding how the optimization ability of PSO can be improved. PSO usually randomly generates various potential solutions in the range of the solution of the optimization problem, which are called "particles". In reference [14], in order to improve the quality of initial particles, Tian et al. replaced the method of generating initial particles via random mapping in PSO with logical mapping. Chen et al. [15] first used random mapping to generate initial particles and then combined this method with a reinforcement learning strategy [16] to generate another batch of reinforcement particles. In this method, after comparing the fitness values of the particles generated using the two methods, the particles with good fitness values are left as the initial particles. Gao et al. [17] first initialized particles via sinusoidal mapping and then used a reinforcement learning strategy to generate a batch of reinforcement particles. Then, they compared the advantages and disadvantages of the two batches of particles to leave particles closer to the optimal solution. In this method, new particles are generated through the two cores of PSO's updated velocity and displacement formula. In the velocity formula, the degree to which the velocity of the new particle is affected by the previous velocity is determined by the inertia weight ω. The degree of influence of the global optimal solution and the individual optimal solution is controlled by the acceleration coefficients c1 and c2. Therefore, ω and c1/c2 have great influence on the final optimization results. To this end, the strategies used to improve ω include the linear strategy [13], the nonlinear strategy [18], the fuzzy rule [19], the chaotic strategy [15], etc. With regard to the acceleration coefficient, sometimes variable acceleration coefficients [20], fixed value acceleration coefficients [21], etc., are used. However, some scholars have improved other values of the updated formula. For example, Xu et al. proposed a dimension learning strategy to improve the individual optimal solution. In this method, the value of each dimension of each individual optimal solution is replaced by the value of each dimension of the global optimal solution one by one. If the effect is positive, the value of the corresponding dimension of the global optimal solution will be retained, and if not, the original state will be maintained [3]. Liang et al. proposed a comprehensive learning strategy to remove the social learning aspect from the speed update formula of classical PSO so that all the remaining individual optimal solutions have the opportunity to learn from the historical individual optimal solutions of other particles, which creates the opportunity for particles to learn from all of the individual optimal solutions [22]. Li et al. combined the improvement of the comprehensive learning strategy and mutation strategy to improve the optimization ability of PSO [23]. Mendes et al. established a speed update strategy in which the particle speed update depends not only on the historical optimal solution of the particle, but also on the historical optimal solution of all other particles [24]. Some scholars applied a mutation strategy to the position of particles to make particles jump out of the local optimal solution. After updating the historical individual optimal particles and historical global optimal particles in PSO, Wang et al. used a mutation strategy to mutate them [25]. This mutation strategy includes Cauchy, Levy, and Gaussian mutations; then, a roulette selection mechanism is used to select mutation factors [26]. Li et al. performed a mutation operation on the global optimal solution in the algorithm when improving PSO. The mutation factor was generated from the difference between two random particles in the population [23]. This research represents the main improvements made by scholars regarding the ontology of the PSO algorithm, while some scholars have combined other algorithms with PSO to form a better algorithm. For example, in reference [27], PSO and GSA were combined to form a hybrid algorithm. The aim was to combine the local development performance of the GSA and the global exploration performance of PSO to form a complementary algorithm. PSO can also be mixed with the sine cosine algorithm [28], the genetic algorithm [29], etc. However, these modified PSOs are still likely be categorized as individual optimal solutions, leading to the transition of the optimization process from global exploration to local development.

In summary, the main challenges of the PSO algorithm are to improve the optimization ability of both the local exploitation and the global exploration by combining all kinds of other algorithms. This leads to the transition of the optimization process from global exploration to local development.

To improve the optimization performance, in this paper, we propose a Hybrid Reinforcement Learning Particle Swarm Algorithm (HRLPSO) based on the theory of reinforcement learning in psychology, which is based on teamwork and runs parallelly.

(1) A Hybrid Reinforcement Learning Particle Swarm Algorithm was proposed. To enhance the optimization capability of HRLPSO, five strategies were applied to improve the traditional PSO in this work. (i) An opposition-based learning strategy was combined with random mapping to generate the initial population; (ii) cubic mapping and adaptive strategies were combined and applied to the weights; (iii) the c_i parameter was controlled to vary nonlinearly within a certain range; (iv) a dimensional learning strategy was applied to the optimal solution; (v) Cauchy and Gaussian mutation strategies were used in the optimal solution to increase the diversity of the solutions.

(2) The results regarding standard functions show that the proposed HRLPSO strategy works well in both stand-alone and ensemble applications, and the results regarding the CEC2013 test suite further demonstrate the good optimization capability of HRLPSO.

(3) Compared with the existing schemes, the main contributions of the proposed HRLPSO are as follows: (i) The theory of reinforcement learning in psychology is firstly applied and the opposition-based learning strategy is proposed to generate the initial population of the PSO. (ii) Unlike the traditional PSO algorithm, which only uses a few hybrid methods, the proposed HRLPSO fully considers the improvement measures at each stage and the five hybrid methods stated above in (1) are applied to improve the optimization performance.

2. Particle Swarm Optimization Algorithm

The particle swarm optimization algorithm is an evolutionary algorithm. The algorithm first generates a set of "solutions" within the approximate range of the solution of the optimization problem, that is, "particles" $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. The value of i is an integer from 1 to N, N is the number of particles, and D is the dimension of particles. Then, by comparing the corresponding objective function values of these particles in the optimization problem, the historical individual optimal solution $Pbest_i = (pbest_{i1}, pbest_{i2}, \ldots, pbest_{iD})$ and the historical global optimal solution $Gbest = (gbest_1, gbest_2, \ldots, gbest_D)$ are obtained. The new particles are updated using the following formula:

$$v_{(i+1)d} = \omega * v_{id} + c_1 rand()(pbest_{id} - x_{id}) + c_2 rand()(gbest_d - x_{id}), \tag{1}$$

$$x_{(i+1)d} = x_{id} + v_{id}, \tag{2}$$

In Equation (1), v represents the velocity of particles, and all the velocity vectors are represented by $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. The values of c_1 and c_2 are weight factors that control particles' individual learning and social learning, and ω is the inertia weight that controls the influence of the previous particle velocity on the updated particle velocity.

3. Hybrid Reinforcement Learning Particle Swarm Optimization Algorithm

3.1. Initial Population Based on Positive Reinforcement Learning

The initial population reinforcement theory based on positive reinforcement learning is a theory proposed by Skinner, an American psychologist and behavioral scientist. Skinner was one of the founders of new behaviorist psychology. He believed that people or animals will display certain behaviors to act on the environment in order to achieve a certain purpose. When the consequences of such behavior are beneficial to the individual, such behavior will repeat in the future; when unfavorable, this behavior weakens or disappears. People can use this method of positive reinforcement or negative reinforcement to change the consequences of behavior and modify their behavior. This is reinforcement theory, also known as behavior modification theory [5]. The convergence speed and accuracy of the particle swarm optimization algorithm are easily affected by the quality of the initial population. In order to improve the quality of the initial population, reinforcement learning is applied to the process of initializing the population.

In the optimization process of various algorithms, some random individuals are randomly generated in the range of solutions as potential solutions and then continuously approach the optimal solution through various iterative mechanisms to produce the optimal solution. However, these algorithms can be improved by later scholars so as to make the algorithm approach better and faster and produce the optimal solution. In this study, reinforcement learning was applied to the algorithm. Reinforcement learning [12] is defined as follows:

Suppose a real number $x^{rn} \in [A, B]$, and the opposite number of x^{on} is defined as follows:

$$x^{on} = A + B - x^{rn} \tag{3}$$

The remaining two definitions are based on the definitions above. Apply the definitions above to the position of an algorithm, such as the PSO algorithm, in which particle $X_i^{rn} = (x_{i1}{}^{rn}, x_{i2}{}^{rn}, \ldots, x_{iD}{}^{rn})$, and enhanced particle $X_i^{on} = (x_{i1}{}^{on}, x_{i2}{}^{on}, \ldots, x_{iD}{}^{on})$, where $x^{rn} \in [A, B]$, and

$$x_{ij}^{on} = A_i + B_i - x_{ij}^{rn} \tag{4}$$

Then, by comparing the fitness values of X_i^{rn} and X_i^{on} in the optimized objective function $f(x)$, the particles with excellent fitness values are left.

3.2. Chaos Adaptive Inertia Weight

The optimization ability of PSO can be effectively improved by reasonably setting the change in the inertia weight coefficient. It has been proved in [9,26] that a linear decline in inertia weight within a certain range can effectively enhance the performance of PSO. The linear decline formula is

$$\omega = \omega_{\max} - \frac{(\omega_{\max} - \omega_{\min})}{\max gen} i, \tag{5}$$

where ω is the value of the inertia weight coefficient under the current number of iterations, $\omega_{max}/\omega_{min}$ is the maximum/minimum value of the inertia weight coefficient, i is the current number of iterations, and *maxgen* is the maximum number of iterations. At present, the most commonly used $\omega_{max}/\omega_{min}$ values in this formula are 0.9/0.4, respectively. In this study, cubic mapping was applied to linearly decreasing weight coefficients as follows [27]:

$$x_{n+1} = ax_n^3 + (1-a)x_n, \tag{6}$$

where x_n denotes the n-th chaotic state in the range of $[-1, 1]$; the initial value x_0 of x_n cannot take 0; and a is the bifurcation coefficient in the semi-open interval (0, 4]. When the value of a increases from zero, the fixed points in Figure 1, the bifurcation graph generated by Equation (6), vary from 1 to 2 and, after that, from 4 to 2n. This variation presents as unlimited and stable, but when a increases close to 3.598076211, the duration proves to be infinite, even aperiodic. When a lies in the range of [3.598076211, 4], the chaotic state occurs and the system presents as unstable when a is bigger than 4, as depicted in Figure 1, where the different random numbers are displayed in different colors.

After setting the absolute value range of the mapped fluctuation (the range is obtained through continuous experimental parameter adjustment), the absolute value range of the fluctuation is as follows:

$$V(i) = Max - \frac{i}{\max gen} Max, \tag{7}$$

where $V(i)$ represents the absolute value of the fluctuation of the mapping under the current number of iterations, and *Max* is the absolute value of the fluctuation at the first iteration.

Combined with cubic mapping, a linearly decreasing mixed disturbance is formed. The combined formula is as follows:

$$C(i) = x(i) * V(i), \tag{8}$$

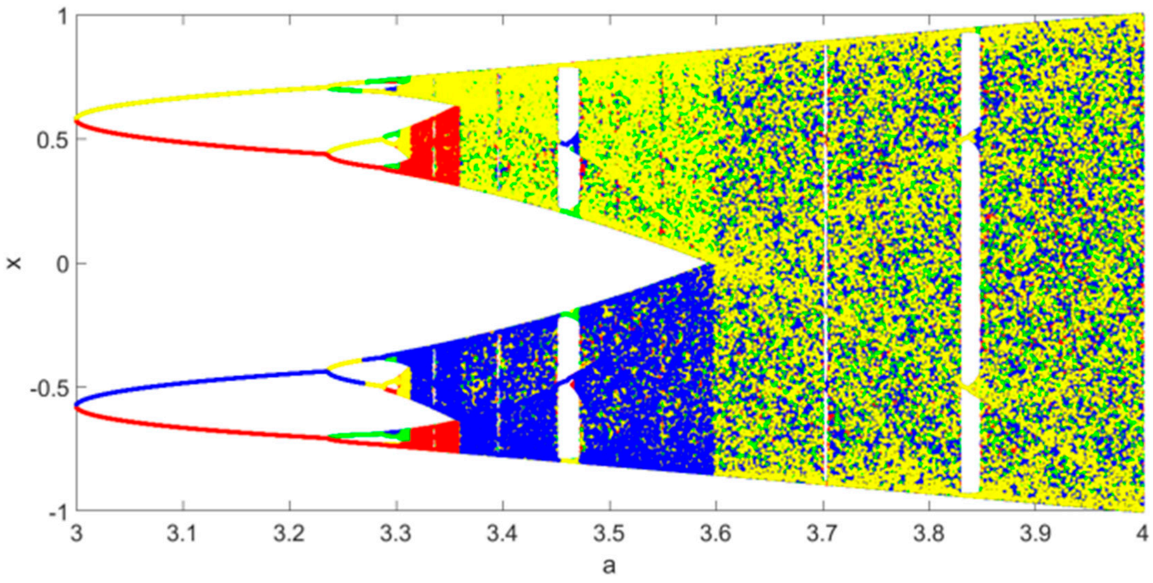


Figure 1. The bifurcation graph with $3 \le a \le 4$.

Finally, the chaotic adaptive inertia weight is obtained by adding it to Equation (5).

$$\omega(i) = \omega(i) + C(i), \tag{9}$$

The whole process is shown in Figure 2, where the variables are depicted as the blue curves

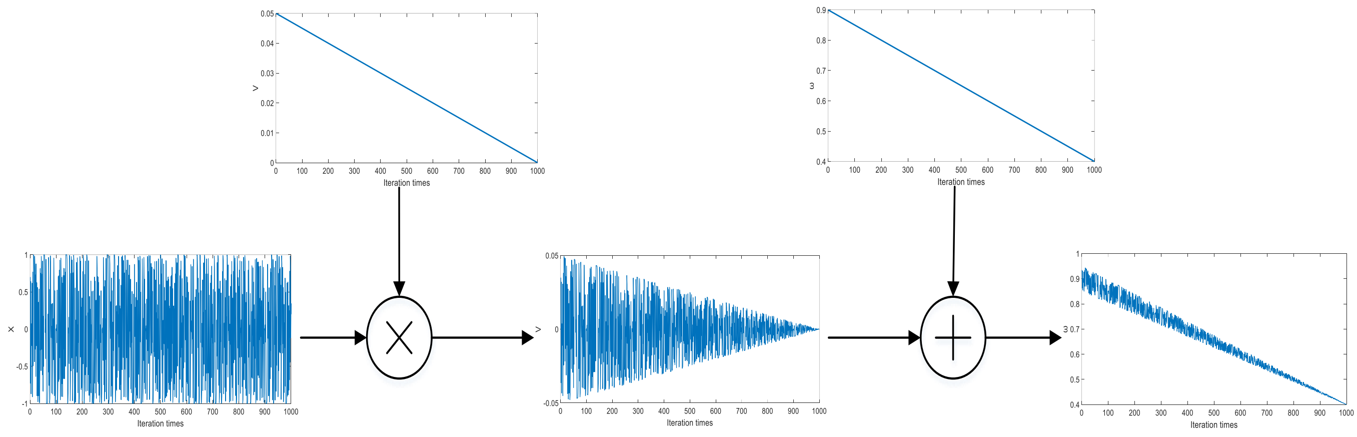


Figure 2. The chaotic adaptive inertia weight coefficient process.

3.3. *Adaptive Learning Factor*

The research regarding learning factors usually focuses on two aspects. On the one hand, the learning factor can be set to a fixed constant. The most typical example of this is the original PSO algorithm. The value of both learning factors is set to 2 in the literature [1]. On the other hand, the learning factor can be set as an adaptive learning factor. Usually, the value of the learning factor is fixed in a certain range and changes with the number of iterations. In the typical research literature [11,16,28,29], this value increases or decreases linearly or nonlinearly between 0.5 and 2.5 as the number of iterations changes. This study is based on adaptive learning factors. The formula of the variable learning factor is as follows:

$$c_1(i) = \alpha \times \sqrt{1 - (1 - (i/\max gen))^2} + \beta, \quad (10)$$

$$c_2(i) = \alpha \times \left(1 - \sqrt{1 - (1 - (i/\max gen))^2}\right) + \beta, \quad (11)$$

where α = 2, β = 0.5. The iterative curve of the learning factors is shown in Figure 3.

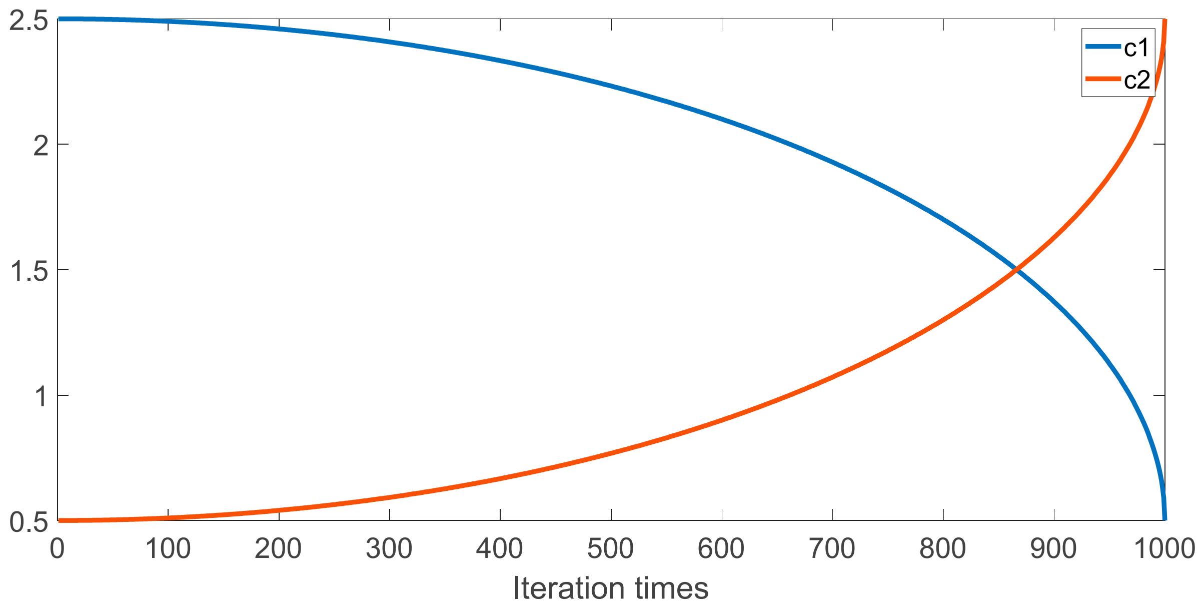


Figure 3. Adaptive learning factors.

3.4. Update Strategy

3.4.1. Dimension Learning

Xu et al. proposed a dimension learning strategy. The principle of this strategy is to replace the value of each dimension of the historical individual optimal solution with the value of the corresponding dimension of the historical global optimal solution. If the objective function value of the optimization problem corresponding to the replaced historical individual optimal solution is better, the value of the replaced dimension will be retained [4]. The advantage of this work is that the best solution is selected from the historical individual optimal solution with the reinforcement learning strategy, and it is compared with the historical global optimal solution, improving the historical global optimal solution. The updated formula is as follows:

$$v_{(i+1)d} = \omega * v_{id} + c_1 rand()(pbest_{id}^{dl} - x_{id}) + c_2 rand()(gbest_d^{dl} - x_{id}), \quad (12)$$

$Pbest_i^{dl}$ and $Gbest^{dl}$ represent the historical individual optimal solution and the historical global optimal solution of the reinforcement learning strategy, respectively.

3.4.2. Mutation

The PSO algorithm has inherent defects and can be easily categorized as local optimization. Particle mutation is an effective strategy to alleviate this situation. The random number of Gaussian distribution functions is mainly concentrated near 0, meaning Gaussian mutation is suitable for particle exploration. Compared with the Gaussian distribution function, the number randomly generated by the Cauchy distribution function is far from 0, meaning that the Cauchy mutation is suitable for particle development. In our work, we aimed to mutate the particle when the particle of the PSO algorithm was categorized as local optimization and simultaneously carried out Gaussian mutation and Cauchy mutation on the particle. The mutation method that produced better results was adopted. The mutation formula is as follows:

$$p_{id}^{dlm} = p_{id}^{dl} + mutation_d() \tag{13}$$

$$p_{gd}^{dlm} = p_{gd}^{dl} + mutation_d() \tag{14}$$

where $mutation_d$ () is the mutation factor.

4. Experimental Setup

In this study, classical test functions were used to test the performance of the algorithm, including seven unimodal functions (F1–F7) and five multimodal functions (F8–F12) [30]. When comparing HRLPSO with the other four algorithms, the population size was set to 30, and each algorithm optimized the test function 20 times. When comparing the performance of the improved method separately, the number of iterations was 1000, and when comparing the performance of HRLPSO with the other four algorithms, the number of iterations was 10,000. The maximum speed limit was consistent with the range of the test function. In addition, in this study, the average value of the final result of the algorithm's optimization of the test function after 20 times is displayed in bold for easy observation. CIPSO was directly applied to engineering problems in the original text, and the performance of the algorithm was analyzed based on the results of engineering problems. In CLPSO and DLPSO, standard test functions are mainly used to assess the performance of an algorithm. The parameter settings of all the algorithms in this work are shown in Table 1.

Table 1. Parameter settings.

Algorithm	Parameter		Reference
PSO	the population size is 30, each algorithm is optimized 20 times, the number of iterations is 10,000, the maximal speed is within the range of F1~F12	w: 1, c_1: 2, c_2: 2	[8]
CIPSO		w: 0.9~0.4, c_1: 3.5~0.5, c_2: 0.5~3.5	[31]
CLPSO		w: 0.9~0.4, c: 1.5	[18]
DLPSO		w: 0.7298, c_1: 1.5, c_2: 0.5~2.5	[3]
HRLPSO		w: 0.9~0.6, c_1: 2.5~0.5, c_2: 0.5~2.5, a: 4, Max: 0.05	-

In the original literature, the capabilities of CIPSO were evaluated by optimizing the results for application to engineering problems. In the original literature, two algorithms, CLPSO and DLPSO, were mainly used to test the algorithm performance with standard test functions. The parameters of the PSO variants are shown in Table 1.

Figure 4 displays the flow chart of HDLPSO, in which *Fit* is the fitness value of the solution. As test functions exist in minimum values, the solution with the smaller fitness value was taken as the better solution when comparing the fitness values.

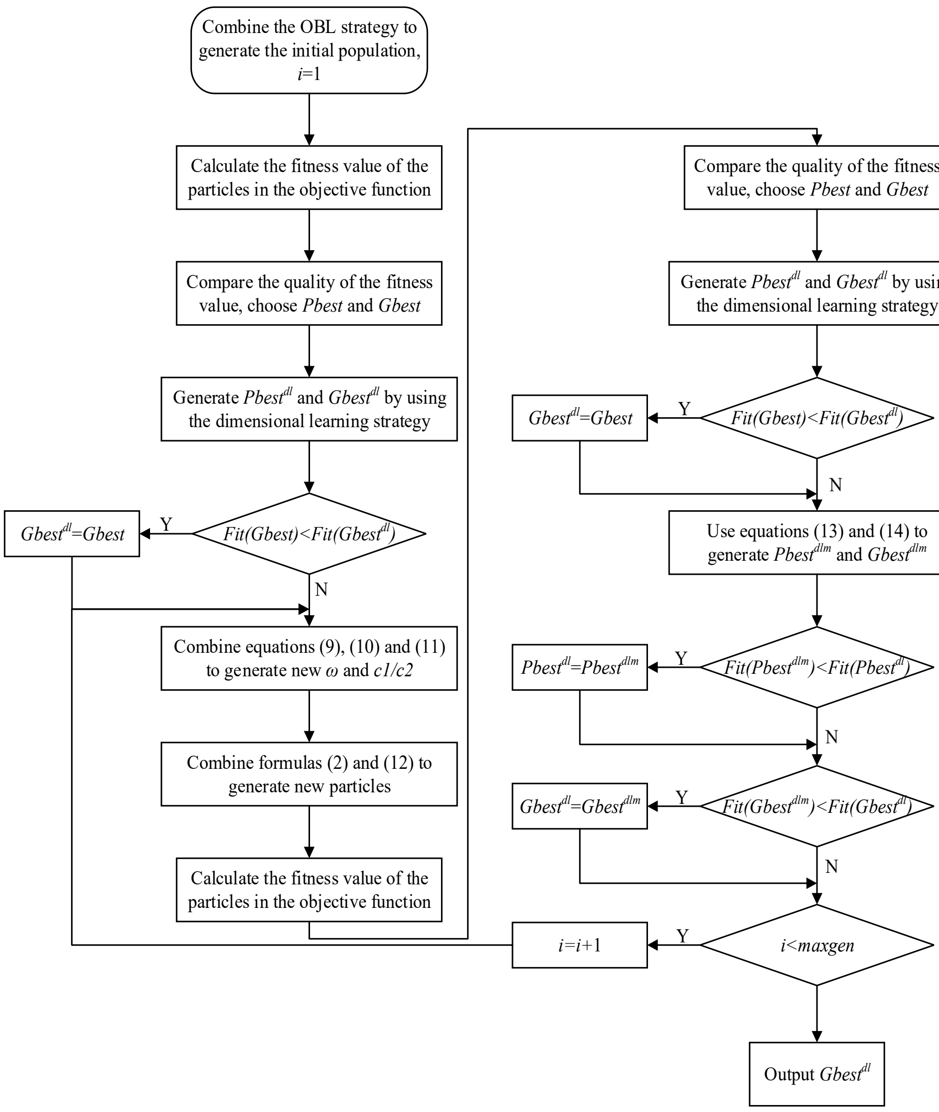

Combine the OBL strategy to generate the initial population, i=1
Calculate the fitness value of the particles in the objective function
Compare the quality of the fitness value, choose Pbest and Gbest
Generate Pbest^dl and Gbest^dl by using the dimensional learning strategy
Fit(Gbest)<Fit(Gbest^dl)
Y
Gbest^dl=Gbest
N
Combine equations (9), (10) and (11) to generate new ω and c1/c2
Combine formulas (2) and (12) to generate new particles
Calculate the fitness value of the particles in the objective function
Compare the quality of the fitness value, choose Pbest and Gbest
Generate Pbest^dl and Gbest^dl by using the dimensional learning strategy
Fit(Gbest)<Fit(Gbest^dl)
Y
Gbest^dl=Gbest
N
Use equations (13) and (14) to generate Pbest^dlm and Gbest^dlm
Fit(Pbest^dlm)<Fit(Pbest^dl)
Y
Pbest^dl=Pbest^dlm
N
Fit(Gbest^dlm)<Fit(Gbest^dl)
Y
Gbest^dl=Gbest^dlm
N
i<maxgen
Y
i=i+1
Output Gbest^dl

Figure 4. Flow chart of HDLPSO.

5. Discussion

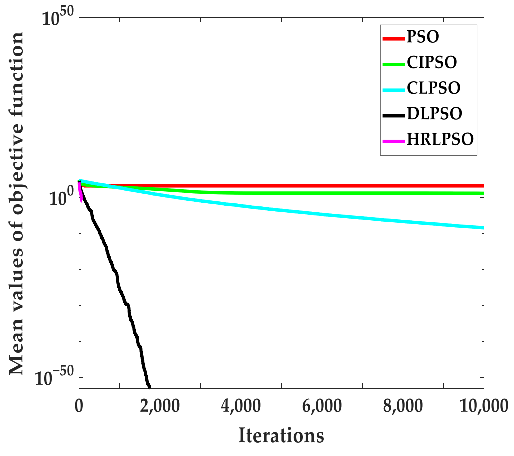
5.1. Test Results of the PSO Variants under Benchmark

Taking 12 benchmark functions as experimental objects, we compared the optimization results of HRLPSO and four other algorithms. There were 10,000 iterations. The results of the comparison of HRLPSO and the four other algorithms are shown in Table 2. For the other four functions aside from the HRLPSO function in the table, the global optimal value of 0 could be obtained. It was shown that even the most original PSO could obtain the global optimal value of 0 on function F4. However, the global optimal results of HRLPSO on functions F1, F2, F3, F4, F6, F8, and F10 were all 0, and the standard deviation was also 0, which shows that HRLPSO can obtain the optimal value, 0, of the function every time it is optimized on these test functions, reflecting its better global optimization ability. Moreover, HRLPSO ranked first in the 12 test functions and the other 6 functions, as well as in the average ranking and final ranking. In the table, F represents the function name, D represents the dimension of the test function, Mean represents the average value of the objective function, and S.D. represents the standard deviation of the objective function value.

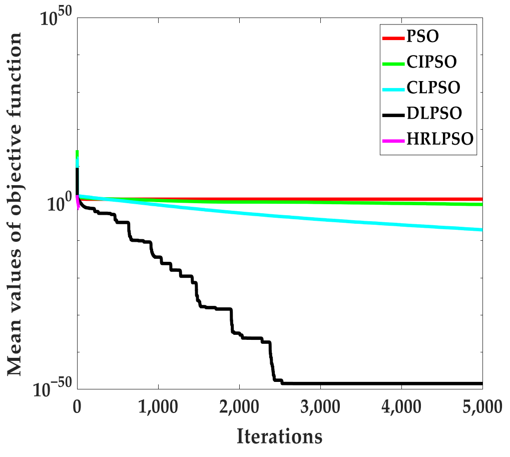
Table 2. Optimization results of HRLPSO and other algorithms under benchmark.

F	D	Item	PSO	CIPSO	CLPSO	DLPSO	HRLPSO
F1	30	Mean	2×10^{3}	1.72×10^{1}	4.43×10^{-9}	0.00	0.00
		S.D.	5.23×10^{3}	8.42	2.53×10^{-9}	0.00	0.00
		Rank	4	3	2	1	1
F2	30	Mean	1.50×10^{1}	5.86×10^{-1}	8.66×10^{-8}	3.49×10^{-43}	0.00
		S.D.	8.89	3.02×10^{-1}	3.54×10^{-8}	1.53×10^{-48}	0.00
		Rank	5	4	3	2	1
F3	30	Mean	1.15×10^{4}	4.96×10^{2}	8.40×10^{3}	1.07×10^{3}	0.00
		S.D.	1.08×10^{4}	2.25×10^{2}	8.92×10^{3}	2.35×10^{3}	0.00
		Rank	5	2	4	3	1
F4	30	Mean	0.00	4.35	1.28	2.01×10^{-13}	0.00
		S.D.	0.00	1.06	5.87×10^{-1}	8.98×10^{-13}	0.00
		Rank	1	4	3	2	1
F5	30	Mean	5.70×10^{1}	6.31×10^{2}	2.31×10^{2}	4.91×10^{1}	1.99×10^{-1}
		S.D.	1.26×10^{2}	4.24×10^{2}	6.75×10^{2}	4.01×10^{1}	8.91×10^{-1}
		Rank	3	5	4	2	1
F6	30	Mean	1.01×10^{3}	1.80×10^{1}	6.38×10^{-9}	0.00	0.00
		S.D.	3.11×10^{3}	6.69	4.53×10^{-9}	0.00	0.00
		Rank	4	3	2	1	1
F7	30	Mean	1.34	1.19×10^{-2}	9.17×10^{-4}	2.16×10^{-2}	3.49×10^{-4}
		S.D.	3.54	5.20×10^{-3}	3.78×10^{-4}	1.46×10^{-2}	3.13×10^{-4}
		Rank	5	3	2	4	1
F8	30	Mean	6.51×10^{1}	6.20×10^{1}	7.06	5.42	0.00
		S.D.	4.26×10^{1}	1.54×10^{1}	2.58	3.88	0.00
		Rank	5	4	3	2	1
F9	30	Mean	7.24	3.11	1.90×10^{1}	1.25×10^{-1}	8.88×10^{-16}
		S.D.	8.44	4.73×10^{-1}	4.70×10^{-1}	3.85×10^{-1}	0.00
		Rank	4	3	5	2	1
F10	30	Mean	9.02×10^{1}	1.11	1.85×10^{-10}	1.41×10^{-2}	0.00
		S.D.	2.78×10^{1}	3.84×10^{-2}	1.54×10^{-1}	2.24×10^{-2}	0.00
		Rank	5	4	2	3	1
F11	30	Mean	1.49×10^{-1}	7.92×10^{-1}	4.23×10^{-11}	3.63×10^{-2}	1.57×10^{-32}
		S.D.	3.87×10^{-2}	3.83×10^{-1}	2.78×10^{-11}	5.07×10^{-2}	2.81×10^{-48}
		Rank	4	5	2	3	1
F12	30	Mean	2.07	1.93e	4.38×10^{-10}	2.69×10^{-2}	1.35×10^{-32}
		S.D.	3.20	1.13	3.05×10^{-10}	9.00×10^{-2}	2.81×10^{-48}
		Rank	5	4	2	3	1
Average Rank			4	3.89	2.78	2.44	1
Final Rank			5	4	3	2	1

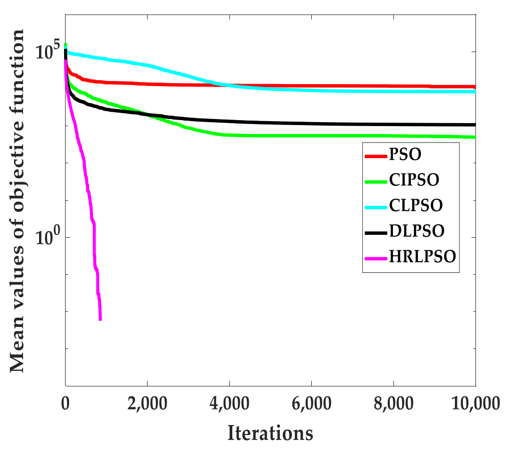
The average evolution curves under the 12 test functions are shown in Figure 5. In the 12 evolution curves, the final convergence accuracy of CLPSO on the test functions F1, F2, F3, F6, F7, F8, and F10 was better than that of PSO, but the convergence accuracy was worse than that of PSO when the number of iterations was 1000. Although HRLPSO did not converge when the number of iterations on test functions F1, F2, F3, F4, and F6 was 1000, it also achieved good convergence accuracy. It converged when the number of iterations on test functions F5, F7, F8, F9, F10, F11, and F12 was 1000. HRLPSO only converged after 1000 iterations of test functions F5, F6, F7, F11, and F12. Therefore, in this test function experiment, the number of iterations was set to 10000. Among the 12 evolution curves, HRLPSO had the highest convergence accuracy. Secondly, as shown in the figures, HRLPSO had the fastest convergence speed on unimodal functions F1, F2, F3, and F4 and multimodal functions F8, F9, and F10. By combining these results with the previous analysis regarding convergence accuracy, it can be concluded that HRLPSO not only has good convergence accuracy but also has good convergence speed.



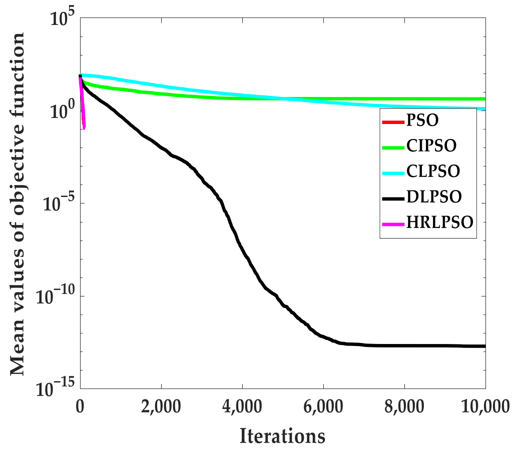
(**a**)



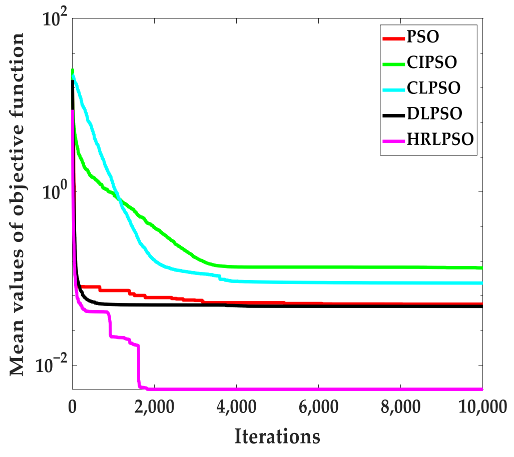
(**b**)



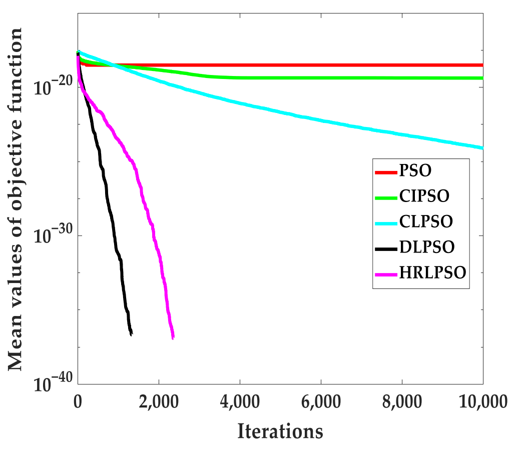
(**c**)



(**d**)



(**e**)



(**f**)

Figure 5. *Cont.*

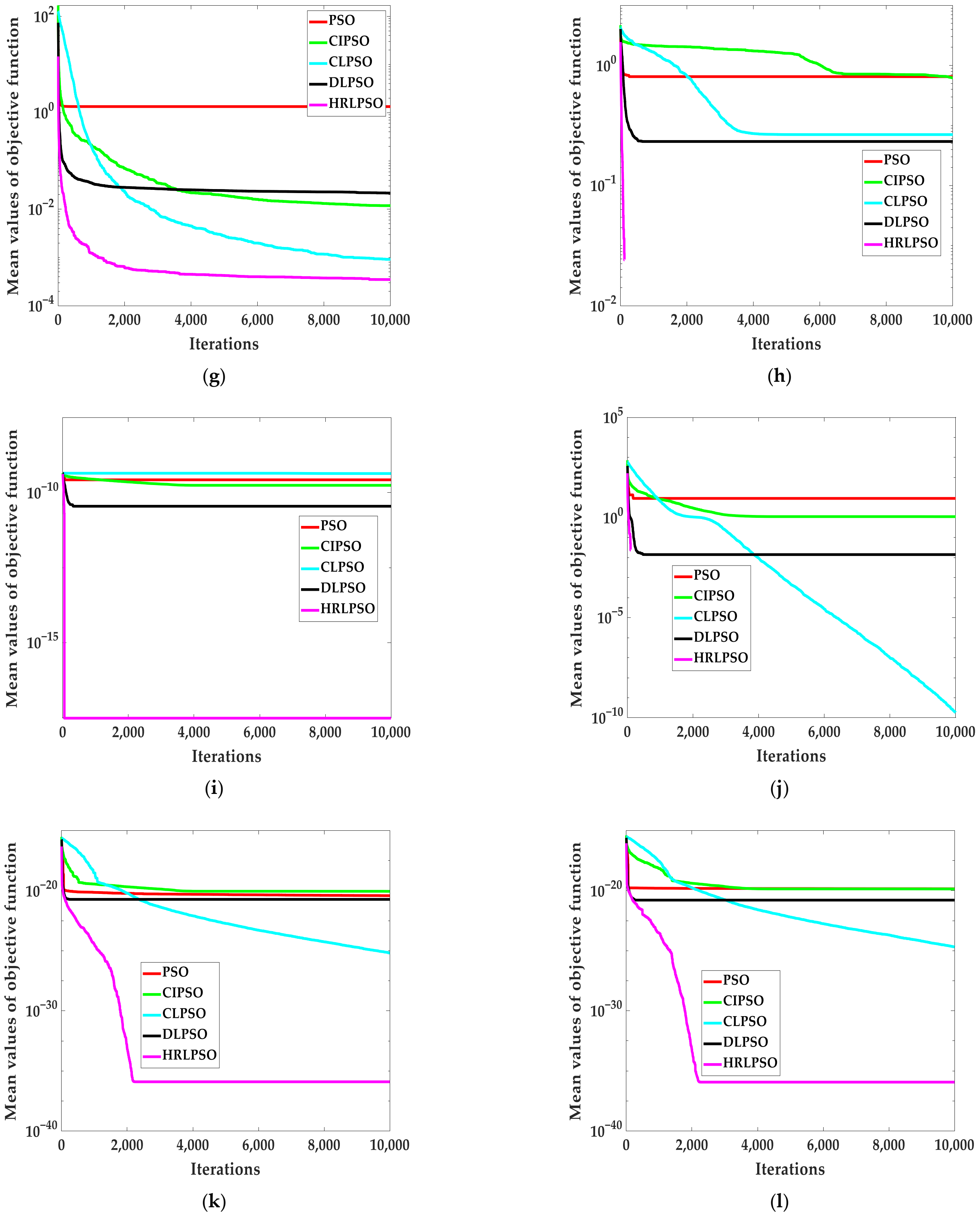


Figure 5. Average evolution curve of 5 algorithms under 12 test functions. (**a**) F1, (**b**) F2, (**c**) F3, (**d**) F4, (**e**) F5, (**f**) F6, (**g**) F7, (**h**) F8, (**i**) F9, (**j**) F10, (**k**) F11 and (**l**) F12.

Table 3 presents a quantitative comparison of the performance indicators of the five algorithms, providing the average computational time and the average rank under 30 running times, achieved under the standard test functions F1~F12. From Table 3, it can be seen that the average rank of HRLPSO is the first and more advanced than the other algorithms. Meanwhile, although the average computational time of HRLPSO is shorter than that of CLPSO, DLPSO, it approximated that of the standard PSO and CIPSO. These indicators show that it performs best.

Table 3. Performance indicators of different PSO algorithms.

Indicators	PSO	CIPSO	CLPSO	DLPSO	HRLPSO
Running times	30	30	30	30	30
Average computational time (s)	13.57	14.11	14.92	14.89	14.88
Average rank	4	3.89	2.78	2.44	1

5.2. Test Results of the PSO Variants under CEC2013 Test Suite

The superiority-seeking performance of HDLPSO was verified in the previous experiments using 12 benchmark test functions. To make the optimization capability of HDLPSO more convincing, in this section, the experimental results of HDLPSO and the other four algorithms under the CEC2013 test suite are provided; see reference [30] for the specific test suite. The experimental parameters were set as the same values used in the previous experiments. In order to distinguish them from the previous 12 benchmark test functions, the 28 functions in the CEC2013 test suite were sequentially sorted by adding 12 to their names.

The optimization results of the five algorithms under the CEC2013 test suite are shown in Table 4, which show that the combined ranking of the five algorithms differed from the previous combined ranking under the 12 benchmark test functions. The ranking of HDLPSO, DLPSO, CLPSO, and PSO remained unchanged; they remained in first, third, fourth, and last place, respectively. Meanwhile, CIPSO ranked second overall; this reflects the fact that an algorithm cannot achieve the best results on every optimization problem. However, when considered together, the optimization results of the five algorithms under the CEC2013 test suite still showed that HDLPSO has excellent optimization capabilities.

Table 4. Optimization results of HRLPSO and other algorithms under the CEC2013 test suite.

Functions	Dimensions	Indicators	PSO	CIPSO	CLPSO	DLPSO	HDLPSO
F13	30	M	1.21×10^{4}	-1.38×10^{3}	-1.01×10^{3}	-1.40×10^{3}	-1.40×10^{3}
		S	5.90×10^{3}	7.15	4.79×10^{2}	3.54×10^{-13}	1.88×10^{-13}
		R	6	2	3	1	1
F14	30	M	1.31×10^{8}	6.17×10^{6}	3.96×10^{7}	6.61×10^{6}	2.31×10^{4}
		S	8.72×10^{7}	2.37×10^{6}	2.76×10^{7}	3.74×10^{6}	2.25×10^{4}
		R	7	2	5	3	1
F15	30	M	6.53×10^{13}	2.75×10^{8}	3.31×10^{10}	2.36×10^{9}	3.22×10^{8}
		S	1.91×10^{14}	1.44×10^{8}	1.83×10^{10}	2.20×10^{9}	5.37×10^{8}
		R	7	1	4	3	2
F16	30	M	7.83×10^{4}	4.19×10^{3}	1.65×10^{4}	9.77×10^{3}	-6.81×10^{2}
		S	5.77×10^{4}	1.77×10^{3}	8.83×10^{3}	3.85×10^{3}	2.98×10^{2}
		R	7	2	4	3	1
F17	30	M	7.80×10^{3}	-9.80×10^{2}	-6.36×10^{2}	-1.00×10^{3}	-1.00×10^{3}
		S	5.16×10^{3}	9.81	5.56×10^{2}	1.37×10^{-9}	1.14×10^{-13}
		R	6	2	3	1	1
F18	30	M	1.02×10^{3}	-8.34×10^{2}	-8.30×10^{2}	-8.62×10^{2}	-8.81×10^{2}
		S	1.73×10^{3}	1.53×10^{1}	3.18×10^{1}	1.99×10^{1}	1.68×10^{1}
		R	7	3	4	2	1

Table 4. *Cont.*

Functions	Dimensions	Indicators	PSO	CIPSO	CLPSO	DLPSO	HDLPSO
F19	30	M	9.21×10^{2}	7.73×10^{2}	-6.88×10^{2}	-7.06×10^{2}	-7.27×10^{2}
		S	4.22×10^{3}	8.66	3.79×10^{1}	1.66×10^{1}	1.99×10^{1}
		R	7	1	4	3	2
F20	30	M	-6.79×10^{2}	6.79×10^{2}	-6.79×10^{2}	-6.79×10^{2}	-6.79×10^{2}
		S	5.67×10^{-2}	5.05×10^{-2}	6.81×10^{-2}	4.32×10^{-2}	6.91×10^{-2}
		R	1	1	1	1	1
F21	30	M	-5.67×10^{2}	5.80×10^{2}	-5.62×10^{2}	-5.70×10^{2}	-5.78×10^{2}
		S	2.40	2.15	1.24	3.16	3.33
		R	5	1	6	3	2
F22	30	M	1.19×10^{3}	-4.76×10^{2}	-1.91×10^{2}	-4.88×10^{2}	-5.00×10^{2}
		S	9.47×10^{2}	1.32×10^{1}	1.79×10^{2}	1.74×10^{1}	4.31×10^{-2}
		R	7	3	5	2	1
F23	30	M	1.74×10^{1}	-2.94×10^{2}	-3.54×10^{2}	-3.82×10^{2}	-3.64×10^{2}
		S	8.17×10^{1}	2.13×10^{1}	2.26×10^{1}	6.48	9.35
		R	7	4	3	1	2
F24	30	M	8.52×10^{1}	-1.91×10^{2}	-1.14×10^{2}	-1.95×10^{2}	-2.19×10^{2}
		S	9.44×10^{1}	2.08×10^{1}	2.32×10^{1}	3.29×10^{1}	1.85×10^{1}
		R	7	3	4	2	1
F25	30	M	1.71×10^{2}	7.48×10^{1}	-2.15×10^{1}	-4.37×10^{1}	-5.41×10^{1}
		S	7.00×10^{1}	2.08×10^{1}	1.62×10^{1}	3.04×10^{1}	3.26×10^{1}
		R	7	1	4	3	2
F26	30	M	6.50×10^{3}	4.33×10^{3}	2.26×10^{3}	1.47×10^{2}	1.16×10^{3}
		S	4.74×10^{2}	5.89×10^{2}	4.72×10^{2}	1.88×10^{2}	3.45×10^{2}
		R	7	6	3	1	2
F27	30	M	7.42×10^{3}	4.60×10^{3}	7.20×10^{3}	5.03×10^{3}	4.08×10^{3}
		S	3.63×10^{2}	5.32×10^{2}	3.28×10^{2}	6.75×10^{2}	5.63×10^{2}
		R	7	2	6	3	1
F28	30	M	2.02×10^{2}	2.02×10^{2}	2.02×10^{2}	2.02×10^{2}	2.01×10^{2}
		S	3.12×10^{-1}	2.47×10^{-1}	2.74×10^{-1}	3.40×10^{-1}	2.06×10^{-1}
		R	2	2	2	2	1
F29	30	M	8.38×10^{2}	4.61×10^{2}	3.42×10^{2}	3.44×10^{2}	3.42×10^{2}
		S	1.41×10^{2}	3.09×10^{1}	2.51	4.72	7.69
		R	6	3	1	2	1
F30	30	M	8.66×10^{2}	5.72×10^{2}	5.87×10^{2}	5.52×10^{2}	4.87×10^{2}
		S	1.27×10^{2}	1.90×10^{1}	1.01×10^{1}	2.72×10^{1}	1.70×10^{1}
		R	7	3	4	2	1
F31	30	M	1.35×10^{5}	5.12×10^{2}	2.23×10^{3}	5.03×10^{2}	5.03×10^{2}
		S	2.64×10^{5}	2.26	2.50×10^{3}	1.04	1.25
		R	6	2	5	1	1
F32	30	M	6.13×10^{2}	6.11×10^{2}	6.12×10^{2}	6.14×10^{2}	6.12×10^{2}
		S	3.90×10^{-1}	1.37	4.38×10^{-1}	8.16×10^{-1}	9.72×10^{-1}
		R	3	1	2	4	2
F33	30	M	2.24×10^{3}	1.10×10^{3}	1.10×10^{3}	1.03×10^{3}	1.02×10^{3}
		S	5.14×10^{2}	5.13×10^{1}	1.65×10^{2}	1.53×10^{2}	5.91×10^{1}
		R	6	3	3	2	1
F34	30	M	7.96×10^{3}	5.09×10^{3}	2.89×10^{3}	1.38×10^{3}	2.02×10^{3}
		S	5.85×10^{2}	5.47×10^{2}	5.56×10^{2}	3.83×10^{2}	3.87×10^{2}
		R	7	4	3	1	2

Table 4. *Cont.*

Functions	Dimensions	Indicators	PSO	CIPSO	CLPSO	DLPSO	HDLPSO
F35	30	M	8.13×10^3	5.55×10^3	8.14×10^3	6.53×10^3	5.11×10^3
		S	4.65×10^2	7.54×10^2	2.75×10^2	6.16×10^2	8.07×10^2
		R	6	2	7	4	1
F36	30	M	1.30×10^3	1.26×10^3	1.28×10^3	1.28×10^3	1.27×10^3
		S	7.28	6.62	5.07	1.03×10^1	7.17
		R	5	1	3	3	2
F37	30	M	1.42×10^3	1.38×10^3	1.39×10^3	1.39×10^3	1.38×10^3
		S	1.45×10^1	1.07×10^1	9.56	7.53	8.22
		R	5	1	2	2	1
F38	30	M	1.56×10^3	1.47×10^3	1.50×10^3	1.40×10^3	1.40×10^3
		S	7.22×10^1	7.39×10^1	9.29×10^1	4.01×10^{-1}	1.30×10^{-3}
		R	5	2	3	1	1
F39	30	M	2.57×10^3	2.08×10^3	2.51×10^3	2.39×10^3	2.25×10^3
		S	1.16×10^2	7.64×10^1	9.04×10^1	8.60×10^1	9.18×10^1
		R	7	1	6	3	2
F40	30	M	4.69×10^3	1.87×10^3	3.18×10^3	2.08×10^3	1.76×10^3
		S	6.74×10^2	6.70×10^1	3.92×10^2	5.11×10^2	2.59×10^2
		R	7	2	4	3	1
	Average R		5.96	2.18	3.71	2.21	1.36
	Final R		7	2	4	3	1

6. Discussion

In order to improve the optimization ability of PSO, five improvement strategies were applied to the PSO algorithm. The reinforcement learning strategy in psychology was applied to the random generation of the initial population to leave better particles. The combination of cubic mapping and an adaptive strategy was applied to ω. This has the advantages of chaotic mapping and being adaptive at the same time. The adaptive strategy was used to adjust c1 and c2 to balance the individual learning ability and social learning ability of the algorithm. The dimension learning strategy was applied to improve the convergence speed and accuracy of the algorithm. Finally, Cauchy mutation and Gaussian mutation strategies were applied to the historical individual optimal solution and the historical global optimal solution, leaving better solutions to jump out of the local optimal solution. Using 12 benchmark functions, the algorithm and existing strategies were verified. The experimental results show that the proposed strategy has a good effect and prove the effectiveness and good optimization ability of the proposed strategy.

The future work is to further refine the HRLPSO algorithm as well as its parameters so that it can be applied to complex economic models.

Author Contributions: Conceptualization, W.H. and Y.L.; methodology, W.H.; software, X.Z.; validation, W.H. and X.Z.; formal analysis, W.H.; investigation, W.H.; resources, Y.L.; writing—original draft preparation, W.H.; writing—review and editing, X.Z.; supervision, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Social Science Foundation of China, grant number 17ZDA046; the National Natural Science Foundation of China (NSFC), grant number 62173134; and the key scientific research project of Hunan Province, grant numbers 21A0452 and HNJG-2021-0168.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sheng, X.; Lan, K.; Jiang, X.; Yang, J. Adaptive Curriculum Sequencing and Education Management System via Group-Theoretic Particle Swarm Optimization. *Systems* **2023**, *11*, 34. [CrossRef]
2. Wang, R.; Hao, K.; Chen, L.; Wang, T.; Jiang, C. A novel hybrid particle swarm optimization using adaptive strategy. *Inf. Sci.* **2021**, *579*, 231–250. [CrossRef]
3. Li, T.; Liu, Y.; Chen, Z. Application of Sine Cosine Egret Swarm Optimization Algorithm in Gas Turbine Cooling System. *Systems* **2022**, *10*, 201. [CrossRef]
4. Shi, L.; Cheng, Y.; Shao, J.; Sheng, H.; Liu, Q. Cucker-Smale flocking over cooperation-competition networks. *Automatica* **2022**, *135*, 109988. [CrossRef]
5. Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R.P.; Tang, J.; Liu, H. Feature Selection: A Data Perspective. *ACM Comput. Surv.* **2016**, *50*, 94. [CrossRef]
6. Xue, B.; Zhang, M.; Browne, W.N.; Yao, X. A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Trans. Evol. Comput.* **2016**, *20*, 606–626. [CrossRef]
7. Dokeroglu, T.; Deniz, A.; Kiziloz, H.E. A comprehensive survey on recent metaheuristics for feature selection. *Neurocomputing* **2022**, *494*, 269–296. [CrossRef]
8. Schockenhoff, F.; Zähringer, M.; Brönner, M.; Lienkamp, M. Combining a Genetic Algorithm and a Fuzzy System to Optimize User Centricity in Autonomous Vehicle Concept Development. *Systems* **2021**, *9*, 25. [CrossRef]
9. Ganguli, C.; Shandilya, S.K.; Nehrey, M.; Havryliuk, M. Adaptive Artificial Bee Colony Algorithm for Nature-Inspired Cyber Defense. *Systems* **2023**, *11*, 27. [CrossRef]
10. Abdelbari, H.; Shafi, K. A System Dynamics Modeling Support System Based on Computational Intelligence. *Systems* **2019**, *7*, 47. [CrossRef]
11. Li, Y.; Wei, K.; Yang, W.; Wang, Q. Improving wind turbine blade based on multi-objective particle swarm optimization. *Renew. Energy* **2020**, *161*, 525–542. [CrossRef]
12. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995.
13. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings, Anchorage, AK, USA, 4–9 May 1998.
14. Tian, D.; Shi, Z. MPSO: Modified particle swarm optimization and its applications. *Swarm Evol. Comput.* **2018**, *41*, 49–68. [CrossRef]
15. Chen, K.; Zhou, F.; Yin, L.; Wang, S.; Wang, Y.; Wan, F. A hybrid particle swarm optimizer with sine cosine acceleration coefficients. *Inf. Sci.* **2018**, *422*, 218–241. [CrossRef]
16. Ahandani, M.A. Opposition-based learning in the shuffled bidirectional differential evolution algorithm. *Swarm Evol. Comput.* **2016**, *26*, 64–85. [CrossRef]
17. Gao, W.F.; Liu, S.Y.; Huang, L.L. Particle swarm optimization with chaotic opposition-based population initialization and stochastic search technique. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 4316–4327. [CrossRef]
18. Malik, R.F.; Rahman, T.A.; Hashim, S.Z.M.; Ngah, R. New particle swarm optimizer with sigmoid increasing inertia weight. *Int. J. Comput. Sci. Secur.* **2007**, *1*, 35–44.
19. Robati, A.; Barani, G.A.; Pour, H.N.A.; Fadaee, M.J.; Anaraki, J.R.P. Balanced fuzzy particle swarm optimization. *Appl. Math. Model.* **2012**, *36*, 2169–2177. [CrossRef]
20. Ratnaweera, A.; Halgamuge, S.K.; Watson, H.C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans. Evol. Comput.* **2004**, *8*, 240–255. [CrossRef]
21. Tanweer, M.R.; Suresh, S.; Sundararajan, N. Self regulating particle swarm optimization algorithm. *Inf. Sci.* **2015**, *294*, 182–202. [CrossRef]
22. Liang, J.J.; Qin, A.K.; Suganthan, P.N.; Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.* **2006**, *10*, 281–295. [CrossRef]
23. Li, W.; Meng, X.; Huang, Y.; Fu, Z.H. Multipopulation cooperative particle swarm optimization with a mixed mutation strategy. *Inf. Sci.* **2020**, *529*, 179–196. [CrossRef]
24. Mendes, R.; Kennedy, J.; Neves, J. The fully informed particle swarm: Simpler, maybe better. *IEEE Trans. Evol. Comput.* **2004**, *8*, 204–210. [CrossRef]
25. Wang, L.; Yang, B.; Orchard, J. Particle swarm optimization using dynamic tournament topology. *Appl. Soft Comput.* **2016**, *48*, 584–596. [CrossRef]
26. Wang, H.; Wang, W.; Wu, Z. Particle swarm optimization with adaptive mutation for multimodal optimization. *Appl. Math. Comput.* **2013**, *221*, 296–305. [CrossRef]
27. Mirjalili, S.; Hashim, S.Z.M. A new hybrid PSOGSA algorithm for function optimization. In Proceedings of the 2010 International Conference on Computer and Information Application, Tianjin, China, 2–4 November 2010.
28. Fakhouri, H.N.; Hudaib, A.; Sleit, A. Hybrid particle swarm optimization with sine cosine algorithm and nelder–mead simplex for solving engineering design problems. *Arab. J. Sci. Eng.* **2020**, *45*, 3091–3109. [CrossRef]
29. Sedki, A.; Ouazar, D. Hybrid particle swarm optimization and differential evolution for optimal design of water distribution systems. *Adv. Eng. Inform.* **2012**, *26*, 582–591. [CrossRef]

30. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [CrossRef]
31. Rogers, T.D.; Whitley, D.C. Chaos in the cubic mapping. *Math. Model.* **1983**, *4*, 9–25. [CrossRef]