1. Introduction of Algorithm Principle

This part mainly introduces the algorithms involved in the experiment: the details and principles of PSO and CLPSO. The CLPSO algorithm is obtained by optimizing the PSO algorithm. The important parameters of these two algorithms will have an important impact on the calculation results. By adjusting the parameters, the final result can be obtained. CLPSO can retain the best history of other particles, thereby ensuring the diversity of the population and preventing premature convergence of the population. This experiment selects questions 5-14 from the CEC2017 question set to test and adjust the parameters, and use CLPSO to solve questions 5-9 Several experiments have been carried out to verify the performance of CLPSO and PSO algorithms.

A. PSO algorithm:

In the PSO algorithm, the solution of each optimization problem is approximated as a particle in space, a random solution is generated by initializing the particle, and then the optimal solution of the particle is found through multiple iterations. In each iteration, the particle updates the particle's speed and position data by updating the particle's historical optimal solution and the population's historical optimal solution. The specific formula is as follows, equation performs consistently on unrotated and rotated problems[1]:

$$V_i = V_i + C_1 * rand() * (pbest_i - X_i) + C_2 * rand() * (gbest_i - X_i) \quad (1)$$

$$X_i = X_i + V_i \quad (2)$$

1) Argument definition description:

Parameter	Description
i	The value range is [1,N], N represents the total number of particles in the population
V_i	The velocity of the i particle in the population
rand()	Generate a random number, the size is between (0,1)
X_i	The current position of the i particle in the population

C_1, C_2	The learning factor of the population determines the learning ability of the particle by adjusting C1 and C2

The maximum value of V_i is V_{max}, and the value of V_{max} is greater than 0. When $V_i > V_{max}$, take $V_i = V_{max}$.

This formula is the basic form of the PSO algorithm, V_i is the memory item, C_1 *rand()*($pbest_i$ -X_i) is the self-cognition item, and C_2 *rand()*($gbest_i$ -X_i) is the group recognition item.

2) Optimization of PSO algorithm

With the evolution of the PSO algorithm[2], the non-negative inertia factor ω is introduced, and the formula (1) is improved:

$$V_i = \omega * V_i + C_1 * rand() * (pbest_i - X_i) + C_2 * rand() * (gbest_i - X_i) \qquad (3)$$

When the value of ω is larger, the global search ability of the population is stronger and the local search ability is lower. When the value of ω is small[3], the global search ability is weaker but the local search ability is stronger. In practical applications, linearly decreasing weights are often used, so that in the iterative process, ω adjusts the global and local search capabilities for different problems, and changes the performance of the PSO algorithm. The current formulas (2) and (3) are called the standard PSO algorithm.

3) Learning factor analysis

a. When C_1=0, $V_i = \omega * V_i + C_2 * rand() * (gbest_i - X_i)$, which is called the global PSO algorithm. At this time, each particle in the population has no local search capability, and has a faster expansion space speed and Convergence speed, compared with standard PSO, the population is more likely to fall into a local optimal solution.

b. When C_2=0, $V_i = \omega * V_i + C_1 * rand() * (pbest_i - X_i)$, which is called the local PSO algorithm. At this time, each particle in the population has no global search ability, and the convergence speed is slow. It is possible that the optimal solution may not be obtained under the order.

4) Standard PSO algorithm flow

a. Initialize the population size, initialize the position and velocity of the particles, and calculate the fitness of each particle.

b. Compare the fitness of each particle with the historical best fitness of the particle. If the current fitness of the particle is better, change It is updated to the best fitness in the history of the particle.

c. Compare the fitness of each particle with the historical best fitness of the population. If the current fitness of the particle is better, then update it to the historical best fitness of the population.

d. Update the velocity and position of the particle.

e. Determine whether the termination condition is reached.

B. CLPSO algorithm:

CLPSO is essentially an improvement of the PSO algorithm, by deleting the global optimal item of the PSO algorithm, and replacing the historical optimal item of the particle with a comprehensive learning factor. The specific formula is as (4), where each parameter has the same meaning as PSO, and $Pbest_{fr(i)}$ is a comprehensive learning factor, which is introduced in CLPSO.

$$V_i = \omega * V_i + C * rand_i^d * (pbest^d_{fi(d)} - X^d_i) \qquad (4)$$

1) Comprehensive learning factor analysis:

Randomly select two particles in the population and compare their fitness, select the particle with better fitness as the candidate, and use the cross probability Pc to cross the candidate particle with the historical optimal[4] term of the specified particle to generate a comprehensive learning factor $Pbest_{fr(i)}$ The Pc of each particle in the population increases linearly. Removal of the global optimal term can solve the oscillation problem in PSO. When the Pbest and Gbest terms share the opposite direction of the particles, it will cause the problem of too fast convergence.

2) CLPSO algorithm process:

a. Randomly select two particles in the population and calculate the fitness, select the better one as the pending particle.

b. Randomly select a certain dimension of the specified particle, and learn from the corresponding dimension of the pending particle.

c. Determine whether it has reached Termination condition.

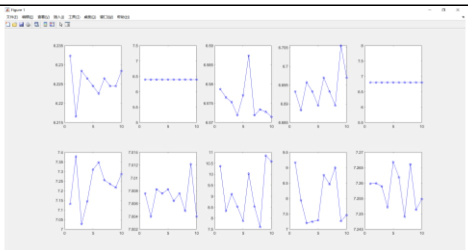
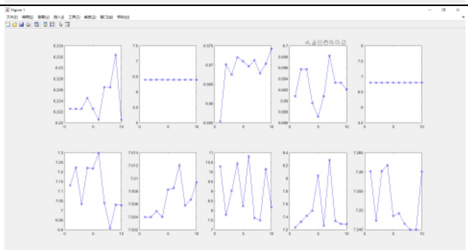
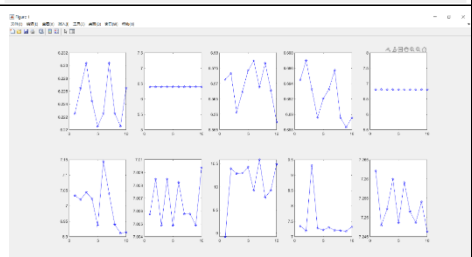
2. Experimental results and analysis

Select the functions numbered 5-14, and compare the stability of the convergence results by setting different parameters. In the following experiment, select the function numbered 5-14 for tuning, set the dimension to 10D problem, the value range of the function variable is between [-100,100], the number of particles is 100, and the maximum number of iterations is 50000.

A. **PSO experimental results and analysis**:

Fixing the learning factor CC in the PSO, adjust the inertia factor ω, and compare the influence of ω on the particle swarm results. When setting ω=0.5, 0.7, 0.9, the results are shown in table 2.1 below Show:

Table 2.1

<table>
<tr><th>parameter</th><th>results when Runs=10</th><th>mean, median and standard deviation when Runs=10</th></tr>
<tr><td>CC=2
ω=0.5</td><td></td><td>
<table>
<tr><th>number</th><th>the middle value</th><th>the mean value</th><th>the std value</th></tr>
<tr><td>5</td><td>504.4773158</td><td>505.3064468</td><td>2.878</td></tr>
<tr><td>6</td><td>600</td><td>600</td><td>0.000</td></tr>
<tr><td>7</td><td>716.0873277</td><td>715.4516664</td><td>4.516</td></tr>
<tr><td>8</td><td>805.9697543</td><td>806.964709</td><td>3.695</td></tr>
<tr><td>9</td><td>900</td><td>900</td><td>0.000</td></tr>
<tr><td>10</td><td>1252.674828</td><td>1269.395925</td><td>175.929</td></tr>
<tr><td>11</td><td>1102.984877</td><td>1103.320857</td><td>2.174</td></tr>
<tr><td>12</td><td>6643.741953</td><td>16990.59402</td><td>17157.092</td></tr>
<tr><td>13</td><td>3355.723123</td><td>5334.547631</td><td>4217.405</td></tr>
<tr><td>14</td><td>1421.048476</td><td>1416.40682</td><td>10.777</td></tr>
</table>
</td></tr>
<tr><td>CC=2
ω=0.7</td><td></td><td>
<table>
<tr><th>number</th><th>the middle value</th><th>the mean value</th><th>the std value</th></tr>
<tr><td>5</td><td>505.9697543</td><td>505.9034217</td><td>2.464</td></tr>
<tr><td>6</td><td>600</td><td>600</td><td>0.000</td></tr>
<tr><td>7</td><td>714.8266406</td><td>714.914524</td><td>3.231</td></tr>
<tr><td>8</td><td>807.9596674</td><td>807.860171</td><td>3.128</td></tr>
<tr><td>9</td><td>900</td><td>900</td><td>0.000</td></tr>
<tr><td>10</td><td>1259.555345</td><td>1295.8968</td><td>179.614</td></tr>
<tr><td>11</td><td>1102.984999</td><td>1103.516523</td><td>1.898</td></tr>
<tr><td>12</td><td>20827.92508</td><td>21343.10918</td><td>16720.009</td></tr>
<tr><td>13</td><td>2377.611033</td><td>4265.108485</td><td>3571.505</td></tr>
<tr><td>14</td><td>1411.600576</td><td>1414.408717</td><td>10.877</td></tr>
</table>
</td></tr>
<tr><td>CC=2
ω=0.9</td><td></td><td>
<table>
<tr><th>number</th><th>the middle value</th><th>the mean value</th><th>the std value</th></tr>
<tr><td>5</td><td>528.0972469</td><td>522.087867</td><td>14.15888899</td></tr>
<tr><td>6</td><td>616.4638811</td><td>612.4971839</td><td>9.218200876</td></tr>
<tr><td>7</td><td>818.3250059</td><td>785.0972054</td><td>52.34232641</td></tr>
<tr><td>8</td><td>841.3779988</td><td>831.8347451</td><td>18.78923066</td></tr>
<tr><td>9</td><td>1211.886958</td><td>1144.453369</td><td>188.1923336</td></tr>
<tr><td>10</td><td>1964.204092</td><td>1781.80607</td><td>450.6833537</td></tr>
<tr><td>11</td><td>1204.147418</td><td>1199.239641</td><td>86.55397607</td></tr>
<tr><td>12</td><td>4738327.166</td><td>5544282.049</td><td>5432715.237</td></tr>
<tr><td>13</td><td>9657.64346</td><td>10502.85355</td><td>9456.002677</td></tr>
<tr><td>14</td><td>1482.714859</td><td>1469.519654</td><td>45.81267569</td></tr>
</table>
</td></tr>
</table>

In order to increase the credibility of the structure and increase the number of runs, set runs=30 to draw the standard deviation line chart after the experiment is shown in fig2.1

and fig2.2. After removing the maximum value and the corresponding function, the experimental results can be clearer Is analyzed.

1） When the initial value of the inertia factor is set to ω=0.7, the overall convergence result is the best. As the number of iterations increases, the value of ω decreases linearly. At the beginning, the value of ω is large, and the global search ability is strong, preventing the particle swarm from converging at the local optimum. At the end of the iteration, ω is reduced to search for the optimum value locally, and the final result is obtained.

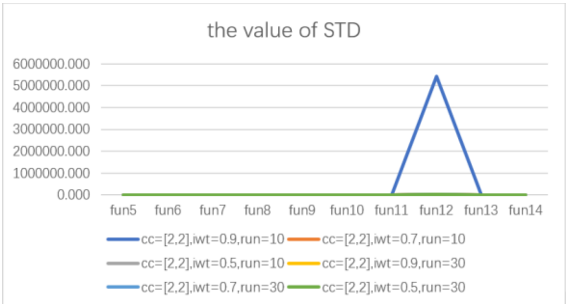


Fig 2.1

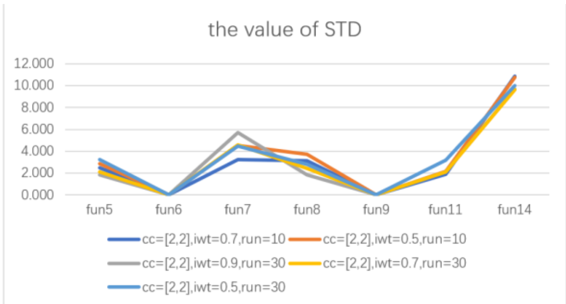


Fig 2.2

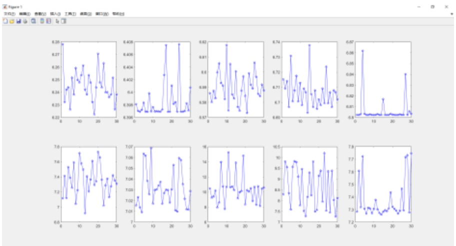
2） After setting ω=0.7, adjust the learning factors C1, C2, and set C1,C2=[1,1], C1,C2=[1.5,1.6], C1,C2=[1.5,1.8], the results are shown in the following table 2.2

table 2.2

Parameter	results when Runs=30	mean, median and standard deviation when Runs=30

ω=0.7

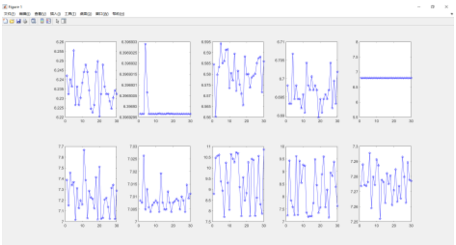
CC=[1,1]



number	the middle value	the mean value	the std value
5	513.4319372	514.8248657	6.949
6	600.0868485	600.8831063	1.357
7	723.5528119	725.3556776	7.581
8	817.9092277	817.8097389	8.169
9	900.454324	911.2978534	53.729
10	1767.624493	1743.553904	271.715
11	1131.365725	1140.867666	29.617
12	14685.26947	86415.19621	263027.253
13	6499.207559	8110.300297	5898.188
14	1495.397899	1687.558132	336.201

ω=0.7

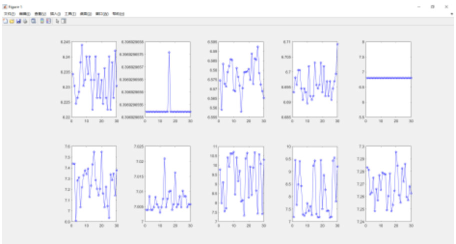
CC=[1.5,1.6]



number	the middle value	the mean value	the std value
5	508.9546265	509.2199475	3.813
6	600	600.000009	0.000
7	720.1132957	720.2648215	6.302
8	808.954629	810.3807314	3.756
9	900	900.0151441	0.083
10	1431.629527	1428.388743	187.950
11	1104.97479	1105.99512	6.207
12	7752.287524	16790.41158	16155.573
13	5723.847474	7061.414807	5391.511
14	1438.826497	1439.763192	13.862

ω=0.7

CC=[1.5,1.8]



number	the middle value	the mean value	the std value
5	507.9596699	508.6229738	3.753
6	600	600.0526578	0.288
7	718.2940028	717.9048006	5.028
8	810.4470625	811.1435319	4.574
9	900	900.0211127	0.100
10	1430.863703	1419.759498	208.869
11	1103.979834	1104.94695	3.848
12	14908.11166	21368.83039	17514.421
13	4426.883023	5824.59795	4695.957
14	1431.561171	1433.96169	16.655

C1, C2 represent the local and global search capabilities of the particles in the population. When C1 is small, the local search capabilities of the particles are weaker, the convergence speed of the space is faster, and the probability of the population falling into the local optimal solution is greater, which reduces the stability of the result. When C2 is small, the particle's global search ability is small, the convergence speed is slow, and the optimal solution may not be reached within a limited number of iterations, which affects the stability of the particle swarm to find the best value. The corresponding results are obtained by adjusting the C1 and C2 parameters. And perform analysis of variance. As shown in figure2.3 below, when C1, C2=[2,2], the experimental result is the most stable, and the result after removing fun12 is shown in figure 2.4.

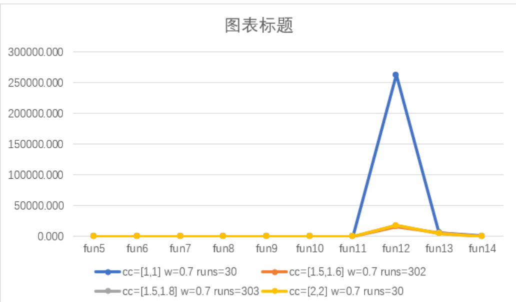


Fig 2.3

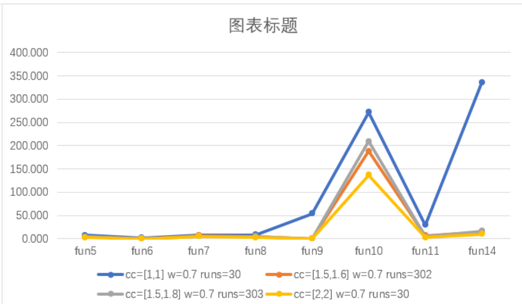
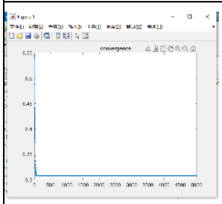
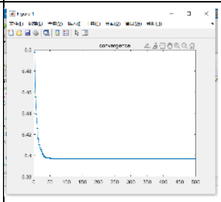
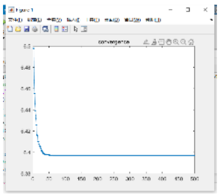
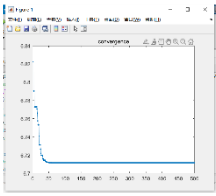
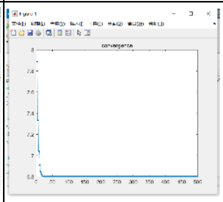
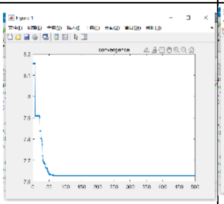
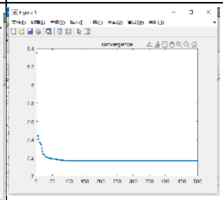
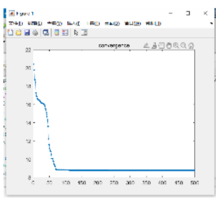
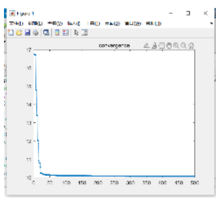
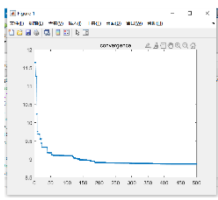


Fig 2.4

The convergence results of the function after determining the parameters are shown in the following table 2.3, and the gbest value obtained after running 30 times is shown in fig2.5.

Table 2.3

Fun5	Fun6	Fun7	Fun8	Fun9
				
Fun10	Fun11	Fun12	Fun13	Fun14
				

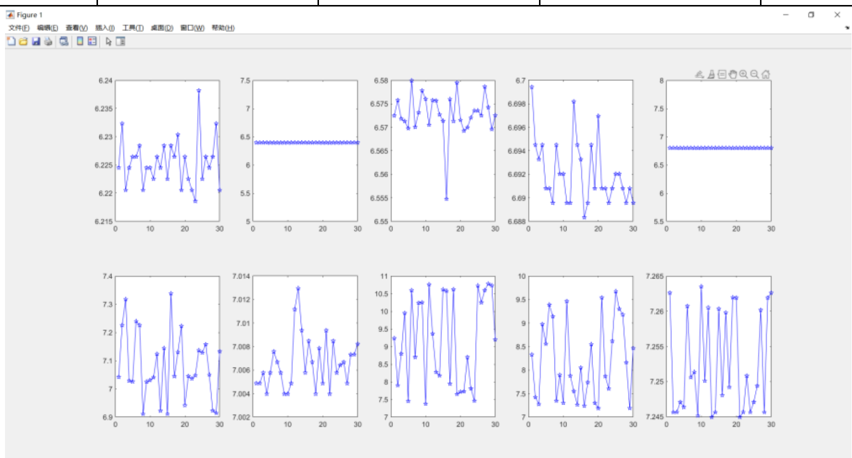
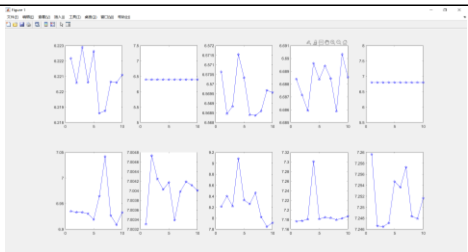
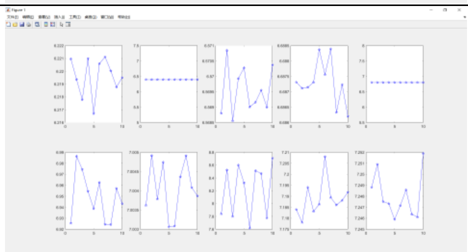
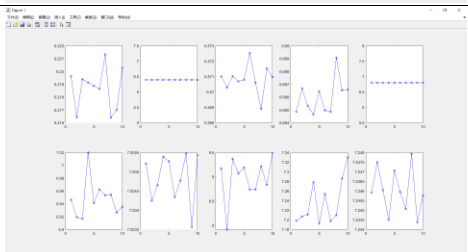
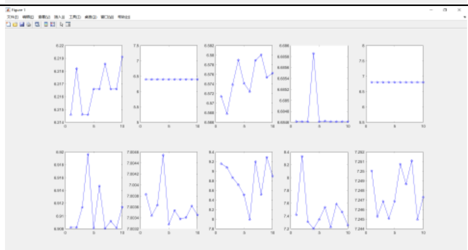


Fig 2.5

B. CLPSO experimental results and analysis:

Select the functions numbered 5-14, and use the parameters set in PSO to further optimize. In the CLPSO algorithm, ω is still the inertia factor but the learning factor is $Pbest_{fr(i)}$, and the comprehensive learning factor is generated It is determined by Pc, so under the condition of keeping the PSO parameters unchanged, set Pc=0.0, 0.1, 0.3, 0.5 to get the experimental results as shown in Table 2.4:

Table 2.4

<table>
<tr><th>Parameter</th><th>results when Runs=10</th><th>mean, median and standard deviation when Runs=10</th></tr>
<tr><td>ω=0.7
Pc=0</td><td></td><td><table>
<tr><th></th><th>std</th><th>middle</th><th>mean</th></tr>
<tr><td>fun5</td><td>0.72703</td><td>503.024</td><td>503.139</td></tr>
<tr><td>fun6</td><td>0</td><td>600</td><td>600</td></tr>
<tr><td>fun7</td><td>0.78801</td><td>712.804</td><td>712.975</td></tr>
<tr><td>fun8</td><td>1.19356</td><td>803.052</td><td>802.885</td></tr>
<tr><td>fun9</td><td>0</td><td>900</td><td>900</td></tr>
<tr><td>fun10</td><td>40.5653</td><td>1025.62</td><td>1036.15</td></tr>
<tr><td>fun11</td><td>0.45475</td><td>1101.11</td><td>1101.05</td></tr>
<tr><td>fun12</td><td>1780.09</td><td>3792.1</td><td>4180.88</td></tr>
<tr><td>fun13</td><td>52.8949</td><td>1315.6</td><td>1331.61</td></tr>
<tr><td>fun14</td><td>6.74259</td><td>1408.12</td><td>1410.02</td></tr>
</table></td></tr>
<tr><td>ω=0.7
Pc=0.1</td><td></td><td><table>
<tr><th></th><th>std</th><th>middle</th><th>mean</th></tr>
<tr><td>fun5</td><td>0.7445</td><td>502.588</td><td>502.494</td></tr>
<tr><td>fun6</td><td>0</td><td>600</td><td>600</td></tr>
<tr><td>fun7</td><td>0.54931</td><td>712.911</td><td>713.051</td></tr>
<tr><td>fun8</td><td>0.57626</td><td>802.13</td><td>802.152</td></tr>
<tr><td>fun9</td><td>0</td><td>900</td><td>900</td></tr>
<tr><td>fun10</td><td>22.6938</td><td>1042.04</td><td>1042.49</td></tr>
<tr><td>fun11</td><td>0.7581</td><td>1101</td><td>1101.08</td></tr>
<tr><td>fun12</td><td>1492.85</td><td>4444.52</td><td>3982.02</td></tr>
<tr><td>fun13</td><td>10.7791</td><td>1322.5</td><td>1324.72</td></tr>
<tr><td>fun14</td><td>2.84253</td><td>1404.49</td><td>1405.38</td></tr>
</table></td></tr>
<tr><td>ω=0.7
Pc=0.3</td><td></td><td><table>
<tr><th></th><th>std</th><th>middle</th><th>mean</th></tr>
<tr><td>fun5</td><td>0.82626</td><td>502.206</td><td>502.061</td></tr>
<tr><td>fun6</td><td>0</td><td>600</td><td>600</td></tr>
<tr><td>fun7</td><td>0.65381</td><td>714.003</td><td>713.97</td></tr>
<tr><td>fun8</td><td>1.096</td><td>801.023</td><td>801.115</td></tr>
<tr><td>fun9</td><td>0</td><td>900</td><td>900</td></tr>
<tr><td>fun10</td><td>31.5362</td><td>1036.8</td><td>1041.03</td></tr>
<tr><td>fun11</td><td>0.7041</td><td>1102.12</td><td>1101.95</td></tr>
<tr><td>fun12</td><td>3067.28</td><td>9131.47</td><td>8523.49</td></tr>
<tr><td>fun13</td><td>67.5672</td><td>1353.58</td><td>1390.66</td></tr>
<tr><td>fun14</td><td>1.68322</td><td>1402.39</td><td>1402.5</td></tr>
</table></td></tr>
<tr><td>ω=0.7
Pc=0.5</td><td></td><td><table>
<tr><th></th><th>std</th><th>middle</th><th>mean</th></tr>
<tr><td>fun5</td><td>0.830995</td><td>500.995</td><td>501.0036</td></tr>
<tr><td>fun6</td><td>0</td><td>600</td><td>600</td></tr>
<tr><td>fun7</td><td>2.74185</td><td>716.7656</td><td>716.8964</td></tr>
<tr><td>fun8</td><td>0.314152</td><td>800</td><td>800.1009</td></tr>
<tr><td>fun9</td><td>0</td><td>900</td><td>900</td></tr>
<tr><td>fun10</td><td>3.846944</td><td>1000.94</td><td>1002.956</td></tr>
<tr><td>fun11</td><td>0.399332</td><td>1100.469</td><td>1100.602</td></tr>
<tr><td>fun12</td><td>2521.489</td><td>7198.492</td><td>7240.787</td></tr>
<tr><td>fun13</td><td>832.6169</td><td>1611.376</td><td>1859.75</td></tr>
<tr><td>fun14</td><td>3.268235</td><td>1404.01</td><td>1404.861</td></tr>
</table></td></tr>
</table>

Pc affects the generation of the comprehensive learning factor. Pc determines the probability that the selected particle will cross the specified particle. When the generated random number is less than the Pc value, two random particles are selected for comparison, and the specified particle is crossed. When the number is greater than Pc, crossover cannot be performed. Therefore, the smaller the Pc value, the less likely the particles will cross, and the more likely it is to learn from itself. The results of the analysis of variance for the experimental results corresponding to each function are shown in fig2.6 and fig2.7. When Pc=0.1, the particle swarm produces the best results.

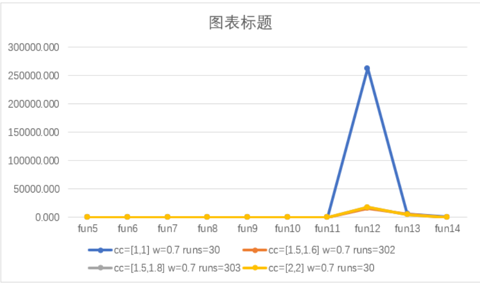


Fig 2.6

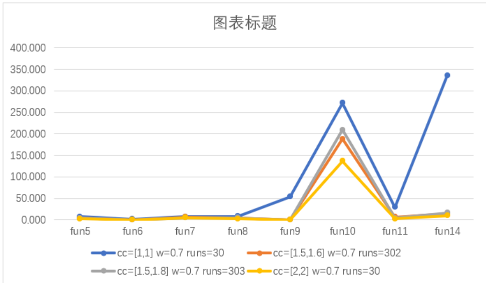
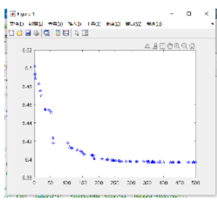
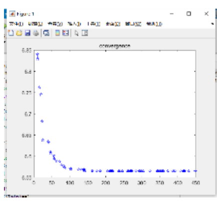
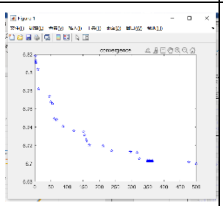
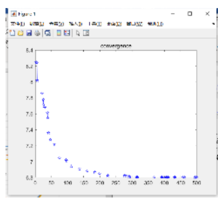
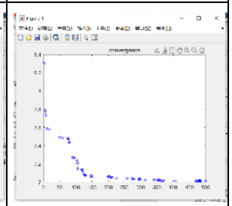


Fig 2.7

The parameters are set to: CC=[2,2], Pc=0.1, ω=0.7, and the convergence diagram of the objective function is obtained as shown in Table 2.5 below and the gbest value obtained after running 30 times is shown in fig2.8:

Table 2.5

Fun6	Fun7	Fun8	Fun9	Fun11
				

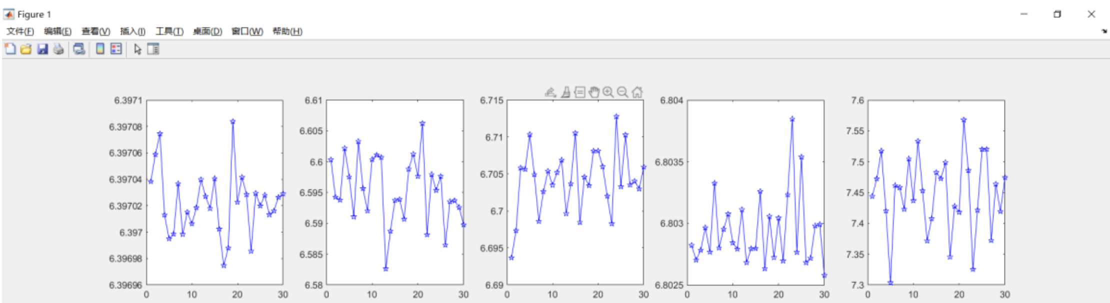


fig2.8

C. Experimental mean, median and variance

1) The mean, median and standard deviation of the PSO algorithm

	Mean value	Median value	Standard deviation
Fun5	504.445	503.980	2.086
Fun6	600	600	0
Fun7	713.979	715.254	4.604
Fun8	805.671	804.975	2.439
Fun9	900	900	0

2) The mean, median and standard deviation of the CLPSO algorithm

	Mean value	Median value	Standard deviation
Fun6	600	600	0
Fun7	712.759	712.781	0.69524
Fun8	802.206	802.266	0.80278
Fun9	900	900	0
Fun10	1718.758	1729.578	105.450891754101

3. Experimental conclusions:

This experiment analyzes the principles of the PSO and CLPSO algorithms, and shows the optimization results of PSO and CLPSO when the number of particles, the problem dimension and the maximum number of iterations are given. For PSO, it learns from the best historical dimension of the population as a whole, and the best historical dimension of the particle itself; for CLPSO, it learns from any example in the population, and learning more samples will also converge. The speed is reduced. Experiments show that for the PSO algorithm, when the learning factor is [2, 2] and the inertia factor is 0.7, the performance of the PSO algorithm is better, and it has more general applicability. CLPSO improves the performance of PSO to a large extent, reduces the oscillation of the PSO algorithm and the instability of the results, and has better performance on different problems. Under the condition of keeping the other parameters unchanged, the performance of CLPSO reaches the best when Pc=0.1.

Reference

[1] D. N. Wilke, “Analysis of the particle swarm optimization algorithm,” master’s thesis, Dept. Mechanical and Aeronautical Eng., Univ. of Pretoria, Pretoria, South Africa, 2005.

[2] M. P. Wachowiak, R. Smolikova, Y. F. Zheng, J. M. Zurada, and A. S. Elmaghraby, “An approach to multimodal biomedical image registration utilizing particle swarm optimization,” IEEE Trans. Evol. Comput., vol. 8, pp. 289–301, Jun. 2004.

[3] H. Y. Fan and Y. Shi, “Study on Vmax of particle swarm optimization,” in Proc. Workshop Particle Swarm Optimization, Indianapolis, IN, 2001.

[4] P. J. Angeline, “Using selection to improve particle swarm optimization,” in Proc. IEEE Congr. Evol. Comput., Anchorage, AK, 1998, pp. 84–89.