A New Improved Simplified Particle Swarm Optimization Algorithm

LIU Haikuan, YUE Dachao, ZHANG Lei, LI Zhiyuan, JIANG Dawei
Jiangsu Normal University of Electrical Engineering and Automation, Xuzhou 221116, China
Email: jsnu_paper@163.com

Abstract. A new simplified particle swarm optimization algorithm is proposed for the question that the basic particle swarm optimization algorithm is easy to fall into the local optima. The algorithm introduces new parameters on the basis of simplifying particle swarm, adjusts parameters by adaptive method, and coordinates the relationship between various parameters, which increases the use of particle information and ensures the difference between particles. Through the test function of eight, the improved algorithm can effectively avoid the premature convergence and greatly improve the convergence speed and convergence precision.

1. Introduction
Particle swarm optimization (PSO) is an optimization algorithm proposed by Kenndy and Eberhart et al in 1995. The algorithm simulates the bird's foraging behavior and considers the solution of the optimization problem as a bird in the search space called particles. Particles have three properties: position, velocity and fitness of objective function. Firstly, the algorithm initializes a group of particles randomly within the search range, and then iterates continuously. In the process of iteration, two parameters are updated, one is the optimal position of the particle itself and another is the optimal position of the population, until the terminating condition is reached.

The particle swarm algorithm method still has many problems, such as convergence speed and local optimality. In order to correct the shortcomings, many scholars have proposed improved methods and achieved certain results. At present, the refined method can be summarized as five types [1]:

- The deformation of standard PSO algorithm [2], such as adjustment of inertia weight and learning factor, add convergence factor, constraint factor, etc.
- Mixture of particle swarm algorithm and other algorithms [3-4], such as quantum particle swarm optimization, immune particle swarm optimization, chaotic particle swarm optimization, etc.;
- Binary particle swarm optimization also known as discrete particle swarm optimization.
- Cooperative particle swarm optimization [5]. For example, mixed three groups of collaborative particle swarm optimization algorithm.
- Simplified particle swarm algorithm [6-8].

The traditional particle swarm algorithm(PSO) usually only uses the particle’s information of individual history optimality and global optimality, but tends to ignore the contemporary optimal particle generated in each iteration process. In this paper, a new simplified particle swarm algorithm is proposed. On the basis of simplified particle swarm algorithm, the new algorithm introduces a new parameter that is contemporary social guidance, namely contemporary optimal particle. In addition,
the new algorithm works as much as possible to coordinate the relationship between the various influencing factors, improves the use of particle information, ensures the differences between particles, improves the premature phenomenon, improves global convergence, and improves the performance of the algorithm.

2. Particle swarm optimization

Particle swarm optimization is an intelligent optimization algorithm. The algorithm initializes some population particles first, and then iterates the optimal solution. Supposing that in the D dimensional search space, S particles are set and iterate T times. The i-th particle is represented by \( X_{i\text{d}} \) at the d-dimensional position, the velocity is expressed in \( V_{i\text{d}} \), the best position of the particle is recorded with \( p\text{Best}_{i\text{d}} \), and the best location of the population is recorded by \( g\text{Best}_{\text{d}} \). After obtaining the above two optimal values, the position and velocity are updated according to

\[
V_{i\text{d}}^{t+1} = \omega V_{i\text{d}}^{t} + c_1 r_1 (p\text{Best}_{i\text{d}} - X_{i\text{d}}^{t}) + c_2 r_2 (g\text{Best}_{\text{d}} - X_{i\text{d}}^{t})
\]

\[
X_{i\text{d}}^{t+1} = X_{i\text{d}}^{t} + V_{i\text{d}}^{t+1}
\]

In (1),(2), \( t = 1, 2, ..., T \), \( i = 1, 2, ..., S \), \( d = 1, 2, ..., D \). \( r_1 \) and \( r_2 \) are random numbers that follow the U(0,1) distribution; \( c_1 \) and \( c_2 \) are the learning factor. \( W \) is a weighting coefficient (between 0.1 and 0.9), and the algorithm terminates when it reaches the maximum number of iterations or target accuracy.

3. Improvement of simplified particle swarm optimization

3.1. Simplified Particle Swarm Optimization

Hu Wang et al simplified the basic particle swarm algorithm to:

\[
X_{i\text{d}}^{t+1} = \omega X_{i\text{d}}^{t} + c_1 r_1 (p\text{Best}_{i\text{d}} - X_{i\text{d}}^{t}) + c_2 r_2 (g\text{Best}_{\text{d}} - X_{i\text{d}}^{t})
\]

In (3), the first item on the right is particle inertia, which regulates the influence of individual history to the present, uses the \( \omega \) to adjust. The second is self-cognition, which is the particle’s thinking on itself. \( c_1 \) and \( c_2 \) are the learning factor. The third is the social guidance, which indicates the particle's imitation of the best particle in history. \( r_1 \) and \( r_2 \) are random numbers that submit the U(0,1) distribution. The algorithm is updated through the position of particles, and the equation is reduced from second order to first order. The validity and superiority of the method is available by theory and experiment. This improvement makes the whole process more simple.

3.2. New Simplified Particle Swarm Optimization

In the simplified particle swarm algorithm(SPSO), each round of particle update is to make the particle individuals greatly approach the extremum of the individual and the entire population, and each particle is evolved with the same iterative formula. The difference between the particles is feeble. It is easy to fall into a local optimal solution, so that individuals of the entire population are surrounded by a certain point, and a phenomenon of premature occurs. In order to improve the phenomenon of premature and maintain the diversity of the population, in the improved algorithm (N-SPSO), the concept of contemporary social guidance is introduced. That is to select the best particles of each round and use them as impact factors to join. In the new update strategy, compared to the simplified particle swarm algorithm, the particle information can be used more fully to ensure the diversity of the particle population, and the performance of the algorithm can be better. The new update strategy is:
In each round of evolution, with certain probability $\eta$, the best contemporary particle is used to update the particle position. In (4), $nBest_{id}$ is the optimal particle of the present and $c_3$ is a learning factor.

In the SPSO, the three parameters of $\omega$, $c_1$ and $c_2$ are completely independent, but this will have an adverse effect on some extent. If the individual’s self-recognition and social guidance are overused, it will cause the global search to fail. For example, if $c_1$ is large, the particles will linger in their own range, and the search speed will be slow. If $c_2$ is large, the particles will converge to the local optimum at the initial stage and fall into the local optimum. Conversely, if the parameter values are small, the particle information is not fully utilized. It is also possible that the algorithm can not obtain the global optimal value, and the particles may be too slow to move out of the local optimal value [9-11].

Therefore, the better parametric adjustment strategy is that in the early stages of evolution, there should be greater individual inertia influence factors, and smaller social guidance capabilities, which will preserve the diversity of particles, and the ability of search throughout the space. In the late stages of evolution, it should have small inertia and large social guidance capability, which will help the algorithm converge to a global optimal solution, thereby improving the algorithm convergence speed and accuracy.

In summary, this paper develops a dynamic evolutionary strategy of “large-scale small search, small-scale large search”. That is, the search space is large, in the early stage of the algorithm, and the particle needs to be searched in a small area around itself so that the particles have a strong self-Cognition. This is used to improve the ability of the algorithm to search globally and prevent it from falling into a local optimum. At the later stage of the algorithm, the search space becomes smaller and the particles need to have stronger social cognition to perfect the local fine search ability of the algorithm. A certain potential correlation has been established between particle inertia, individual cognition, social guidance, and contemporary social guidance, as follows:

$$
\omega = \begin{cases} 
\omega_{\text{max}} & X'_{id} > X_{\text{mean}} \\
\omega_{\text{min}} - (\omega_{\text{max}} - \omega_{\text{min}}) \cdot \frac{(X'_{id} - X_{\text{min}})}{(X_{\text{mean}} - X_{\text{min}})} & X'_{id} \leq X_{\text{mean}}
\end{cases}
$$

(5)

$$
c_1 = 0.5 + 2 \cdot \cos\left(\frac{\pi \cdot (t-1)}{2 \cdot (T-1)}\right)
$$

(6)

$$
c_2 = 0.5 + 2 \cdot \sin\left(\frac{\pi \cdot (t-1)}{2 \cdot (T-1)}\right)
$$

(7)

$$
c_3 = c_1
$$

(8)

The adaptive inertia weight tactic is used. The range is $\omega \in [\omega_{\text{min}}, \omega_{\text{max}}]$. The introduction of trigonometric functions in learning factors is to avoid large mutations. It can be seen that the new algorithm pays more attention to the individual in the early stage, and then highlights the population intelligence of the whole population at a later stage. The newly introduced contemporary social guidance parameter takes the same position as individuals.

The flows of the new algorithm are as follows:

1) Initialize N particle positions, $pBest_{id}$, $gBest_{id}$, and $nBest_{id}$ within the search scope.
2) Perform the following operations on all particles in the population:

2.1) According to (4), (5), (6), (7), calculating \( c_1, c_2, c_3, \omega \).

2.2) Calculating particle fitness and reset \( n_{Best} \).

2.3) If the current particle's fitness is better than its own history \( p_{Best} \), update \( p_{Best} \) to the current position.

2.4) If the current particle's fitness is better than the global particle \( g_{Best} \), then the \( g_{Best} \) is updated to the current position.

2.5) If the current particle's fitness is better than all contemporary particle records \( n_{Best} \), update \( n_{Best} \) to the current position.

3) To determine whether the algorithm satisfies the accuracy or reaches the maximum number of iterations. If yes, go to step 4, otherwise go to step 2.

4) Outputting \( g_{Best} \), get the corresponding value, the algorithm ends.

4. Experiment analysis

In order to test the effectiveness of the new algorithm proposed in this paper, five algorithms are selected for comparative analysis. The five algorithms are

- Adjust particle swarm optimization (APSO) \([12]\).
- Extreme simplified particle swarm optimization (ESPSO).
- Adjust weight particle swarm optimization (AWPSO).
- Proposed simplified particle swarm optimization (PSPSO) \([13]\).
- Simplified particle swarm optimization (SPSO) \([14]\).

Eight classical test functions are used to test the performance of the algorithm. These test functions are

| Function name          | Formula                | Function name          | Formula                |
|------------------------|------------------------|------------------------|------------------------|
| Sphere                 | \( f(x) = \sum x_i^2 \) | High conditioned elliptic | \( f(x) = \sum (x_i + 10^{0.001} x_i^2) \) |
| Rastrigin              | \( f(x) = \sum (x_i^2 - 10 \cos(2\pi x_i)) \) | Quartic                | \( f(x) = \sum x_i^4 + \text{rand}() \) |
| Sum of different power | \( f(x) = \sum (x_i + |x_i|^{0.1}) \) | Schwefel 1.2           | \( f(x) = \sum (\sum x_j)^2 \) |
| Alpine                 | \( f(x) = \sum (x_i + |x_i \sin(x_i) + 0.1x_i|) \) | XinSheYang01           | \( f(x) = \sum (x_i + |x_i| \cdot \text{rand}) \) |

Each set of tests used the same population size of 40 particles, \( x \in [-100, 100] \), a particle dimension of 10, a cycle of 30 times, and an average of 100 generations per evolution. The results obtained are as follows
Fig 1. Sphere function curve

Fig 2. Rastrigin function curve

Fig 3. Sum of different power function curve

Fig 4. Alpine function curve

Fig 5. High conditioned elliptic function curve

Fig 6. Quartic function curve

Fig 7. Schwefel function curve

Fig 8. XinSheYang01 function curve
Table 2. Comparison of function results A

| function algorithm | Sphere | Rastrigin | SumOfDifferentPowers | Alpine |
|--------------------|--------|----------|----------------------|--------|
|                    | mean   | optimum  | mean                 | optimum |
| APSO               | 0.0014 | 3.62e-5  | 4.01e1               | 8.9130 |
| ESPSO              | 8.37e3 | 2.30e-18 | 2.16e4               | 7.3e3  |
| AWPSO              | 3.0650 | 0.2172   | 7.948e1              | 3.62e1 |
| PSPSO              | 50.852 | 1.2062   | 1.783e2              | 3.61e1 |
| SPSO               | 5.9e-20| 5.5e-24  | 1.132e2              | 2.21e-8|
| N-SPSO             | 3.5e-76| 1.14e-78 | 2.414e1              | 1.35e-9|

Table 3. Comparison of function results B

| function algorithm | HighConditionedE | Quartic | Schwefel1.2 | XinSheYang01 |
|--------------------|-------------------|---------|-------------|--------------|
|                    | mean              | optimum | mean        | optimum      | mean       | optimum    |
| APSO               | 9.31e5            | 0.0814  | 0.129       | 0.013        | 2.11e3     | 6.19e2     | 4.44e5     | 0.0244     |
| ESPSO              | 3.80e7            | 9.9e-17 | 5.21e8      | 1.24e6       | 1.01e5     | 0.0022     | 2.3e4      | 8.3e6      |
| AWPSO              | 1.87e6            | 3.87e2  | 108.9       | 1.557        | 3.45e3     | 7.36e2     | 4.9e7      | 3.95e1     |
| PSPSO              | 3.31e5            | 1.65e3  | 6.57e4      | 1.13e2       | 1.51e3     | 2.66e2     | 1.95e6     | 2.61e1     |
| SPSO               | 1.7e-17           | 1.7e-21 | 0.675       | 0.0345       | 8.3e-16    | 2.5e-16    | 3.49e-1    | 1.7e-8     |
| N-SPSO             | 2.6e-72           | 1.7e-73 | 0.071       | 0.0053       | 2.8e-75    | 4.71e-76   | 3.12e-5    | 1.51e-30   |

Fig. 1 to Fig. 8, Table 2 to Table 3 show the results of testing. It can be seen that N-SPSO has obvious advantages compared to the other five algorithms. Both the evolutionary speed and the convergence speed are very fast, and the convergence accuracy is also high, that is, N-SPSO has a better optimization performance.

5. Conclusion

Aiming at the shortcomings of the original particle swarm optimization algorithm, this paper proposes an improved scheme: a new simplified particle swarm optimization (N-SPSO). The algorithm introduces fresh variables. This can regulate and balance the relationship among the parameters adaptively, thus enhancing the utilization of particle information. The experiments of eight classical test functions show that, compared with the other particle swarm optimization algorithms mentioned above, N-SPSO has the advantages of strong global search ability, fast convergence speed, high precision, and can effectively avoid precocity and so on. The promoted scheme strengthens the performance of the algorithm. However, there are many factors that affect the N-SPSO algorithm, such as how to adjust the relationship between various parameters, how to determine the optimal minimum number of iterations, etc. these are all the future research directions.

References
[1] Huang Taian, Sheng Jia gen, Xu Hongyang, et al. Improved Simplified Particle Swarm Optimization[J]. computer simulation, 2013,30 (02): 327-330+335.
[2] Zhao Yuandong. Particle swarm optimization with weight function’s learning factor [J]. computer application, 2013,33 (08): 2265-2268.
[3] Pan Dazhi, Liu Zhibin. Realization of improved Quantum-behaved Particle Swarm Optimization algorithm [J]. computer engineering and application, 2013,49 (10): 25-27.
[4] Ceng Yanyang, Feng Yunxia, Zhao Wentao. Adaptive Mutative Scale Chaotic Particle Swarm Optimization Based on Logistic Mapping [J]. System Simulation Journal, 2017, 29(10): 2241-2246.

[5] Gao Yunlong, Yan Peng. Unified optimization algorithm based on multi-swarm PSO algorithm and cuckoo search algorithm [J]. Control and Decision, 2016, 31(04): 601-608.

[6] Zhou Dan, Nan Jing Chang, Gao Mingming. Fuzzy neural network for modeling based on improved simplified particle swarm optimization [J]. Computer Application Research, 2015, 32(04): 1000-1003.

[7] Chen Qunlin, Gao Yue Lin, Guo Xiang. Particle swarm optimization algorithm based on equal replacement and random opposition [J]. Computer Application Research, 2017, 34(08): 2364-2367.

[8] Xiong Zhong Wang, Luo Ke. Clustering algorithm based on improved simplified particle swarm optimization [J]. Computer Application Research, 2014, 31(12): 3550-3552.

[9] Sun Zhenlong, Li Xiaoye, Wang Ying. Improved Simple Particle Swarm Optimization Algorithm [J]. Computer Science, 2015, 42(S2): 86-88.

[10] Zhao Jiaxin, Gao Yue Lin. A Modified Adaptive Particle Swarm Optimization [J]. Journal of Ningxia University (Natural Science Edition), 2016, 37(02): 125-130+134.

[11] Ma Guoqing, Li Ruifeng, Liu Li. Particle swarm optimization algorithm of learning factors and time factor adjusting to weights [J]. Computer Application Research, 2014, 31(11): 3291-3294.

[12] Zhou Haotian, Wu Zhiyong, Tian Yubo. Research on improving Simple Particle Swarm Optimization [J]. Computer Engineering and Application, 2012, 48(24): 41-44.

[13] Mustafa Servet Kiran. Particle swarm optimization with a new update mechanism[J]. Applied Soft Computing, 2017, 60.

[14] Hu Wang, Li Zhishu. A simpler and More Effective Particle Swarm Optimization Algorithm [J]. Software Journal, 2007, 18(04): 861-868.