Improved Particle Swarm Optimization Algorithm Based on Dynamic Change Speed Attenuation Factor and Inertia Weight Factor

Chao Wang¹, Yang Ou² and Zhiyong Shan²,*

¹School of Electronic Information Science and Technology, Donghua University, Shanghai, China
²School of Electronic Information Science and Technology, Donghua University, Shanghai, China

*Corresponding author e-mail: 1913891129@qq.com

Abstract. Compared with other optimization algorithms, the particle swarm optimization algorithm (PSO) has the advantages of fast convergence, simple calculation, and fewer parameters to be adjusted. Therefore, it has been greatly developed in the past period of time. In order to further accelerate convergence, in recent years, more efficient and convenient particle swarm optimization algorithms such as LDWPSO, NLPWSO, and DPSO have been proposed. However, these algorithms have many problems such as large amount of calculation, lack of precision, and easy to fall into local optimum. Therefore, the algorithm still needs to be improved. In order to obtain an efficient global search ability early in the algorithm iteration and better local optimization performance in the later stage of the algorithm iteration, this paper proposes a special expression form of dynamic change speed attenuation factor and inertia weight factor, and in the algorithm some random parameters are added to the iterative formula and applied to the algorithm iteration formula to improve the optimization speed and optimization effect of the algorithm. Finally, the effectiveness and correctness of the algorithm are also demonstrated by computer simulation experiments.

1. Introduction

The particle swarm optimization algorithm originated from the 1987 reyhold simulation study of the bird population social system. In the flock of birds flying in the air, each bird will follow the following three rules: the first one, avoid conflicts with adjacent birds; the second, try to keep pace with the birds around you; and the third, try to be close to the group that you think. Based on the above characteristics, this feature was applied to the algorithm by Kennedy and Eberhart in 1995 [1]. In order to find the best solution in the algorithm, the evolutionary principle constructed is to form all the members (particles) into a group society. In this group, these particles adjust their position and speed in the next iteration by sharing their flight experience. At the end of the iteration, the best location and the best convergence time for the entire process of the algorithm can be found in the population.

In recent years, the improved PSO algorithm generally adopts the method of modifying the relevant parameters in the iterative formula of the standard particle swarm optimization algorithm [2]. Or change the particle learning model by designing different types of topologies to improve population...
diversity\[3,4,5\]. Or adopt a combination of PSO and other optimization algorithms to form a hybrid PSO algorithm\[6,7,8\]; There is also the application of niche technology to PSO, which improves the PSO algorithm\[9,10\]; Most of these improved algorithms replace the fixed parameter values with more efficient expressions for adjusting parameters, and thus reduce the cost and complexity of the algorithm.

Among the various strategies described above, the adjustment parameters are fixed with more efficient expressions to have fixed parameter values, thereby reducing the complexity of the cost and algorithm. Shi and eberhart proposed in 1998 to introduce a linearly decreasing inertia weighting factor \( w \) into the velocity update equation\[11\]. The velocity attenuation factor \( c_1,c_2 \) proposed by Du and Cui dynamically uses the cosine function to change, and the inertia weighting factor \( w \) dynamically uses the sine function to change\[12\]. The above two improved algorithms are named LDWPSO and NLPSON in this paper. By adjusting the inertia weight, the search range of the particle can be well controlled, and the decisiveness of \( V_{\text{max}} \) is greatly weakened. Larger inertia weights are suitable for large-scale exploration of the solution space, while smaller inertia weights are suitable for small-scale mining\[13,14\]. Therefore, in order to ensure good convergence of particles throughout the convergence process. In this paper, the velocity decay factor \( c_1,c_2 \) is transformed into a form of partial cosine function Taylor expansion of dynamic attenuation, and the inertia weighting factor \( w \) is expressed as a partially sinusoidal Taylor expansion of dynamic attenuation. And on this basis, a certain amount of orbiting parameters are added to make the particles more precise local optimization in the optimization and iterative process, and reduce the probability of falling into local optimum. In the second half of the article, we sample four (sphere, ackley, griewank, sum squares) test functions to verify the correctness and efficiency of our algorithm for both convergence speed and convergence effects compared with LDWPSO and NLPSON.

2. LDCPSO algorithm

Suppose a population of \( n \) particles in a \( D \)-dimensional search space \( X=(X_1,X_2,\ldots,X_n) \). The \( i \)-th particle is represented as a \( D \)-dimensional vector \( X_i=(x_{i1},x_{i2},\ldots,x_{iD}) \), It represents the position of the \( i \)-th particle in the \( D \)-dimensional search space and also represents a potential solution to the problem. The fitness value corresponding to the position of each particle \( X_i \) can be calculated according to the objective function. The speed of the \( i \)-th particle is \( V_i=(v_{i1},v_{i2},\ldots,v_{iD}) \), its individual extremum is \( P_i=(p_{i1},p_{i2},\ldots,p_{iD}) \), the group extremum of the population is \( P_g=(p_{g1},p_{g2},\ldots,p_{gD}) \). During each iteration, the particles update their speed and position through equations (1) and (2), ie

\[
V_{id}^{k+1}=wV_{id}^k+c_1r_2(P_{id}^k-X_{id}^k)+c_1r_1(P_{gd}^k-X_{id}^k) \tag{1}
\]

\[
X_{id}^{k+1}=X_{id}^k+V_{id}^{k+1} \tag{2}
\]

Where \( w \) is the inertia weight factor; \( d=1,2,\ldots,D; k \) is the current number of iterations; \( V_{id} \) is the velocity of the particles; \( c_1 \) and \( c_2 \) are velocity decay factors; \( r_1 \) and \( r_2 \) are random numbers distributed over [0,1]. In order to prevent blind search of particles, it is generally recommended to limit their position and speed within a certain interval. The inertia weight reflects the ability of the particle to inherit the previous velocity. Shi.Y first introduces the inertia weight into the PSO algorithm, and analyzes that a large inertia weight is beneficial to the global search, and a smaller inertia weight is beneficial to the local search. Therefore, in order to better balance the search ability of the algorithm, Shi.Y expresses the inertia weighting factor as follows: (3) is in the form of dynamic linear decrement, and this improved algorithm is referred to as LDWPSO in this paper. The modification method is as follows, ie

\[
w(k)=w_{\text{Start}}+(w_{\text{Start}}-w_{\text{End}})\times(k/T)2 \tag{3}
\]

Where \( w_{\text{Start}} \) is the initial inertia weight, \( w_{\text{End}} \) is the inertia weight after the iteration ends, \( k \) is the number of current iterations, and \( T \) is the maximum number of iterations. In general, the value of inertia weight is the best when \( w_{\text{Start}}=0.9, w_{\text{End}}=0.4 \). Thus, as the iteration progresses, the inertia weight decreases linearly from 0.9 to 0.4. The inertia weight at the beginning of the iteration is to maintain the strong global search ability, while the smaller inertia weight in the later iteration is beneficial to the
algorithm for more accurate local search. Dynamic inertia weight is just an empirical practice. There are many other situations that are commonly used. In this paper, we mainly compare it with the pso algorithm under one of the dynamic weights.

In order to solve the problem of global optimization due to premature convergence and local optimal value, multi-mode and high (low) dimensional complex problems, Du, Cui obtains the best value range of parameter variation in iterative process through a large number of experiments, which is that $c_1$ and $c_2$ is in the range of $1.4$ to $1.6$, $w$ is in the range of $0.6$ to $0.8$. Du, Cui proposed a new adaptive strategy for selecting appropriate parameters [10]. The modification method is represented by the following formula (4) (5) (6)

\[
c_1 = 0.1 \times \cos(\pi k/T) + 1.5 \tag{4}
\]
\[
c_2 = -0.1 \times \cos(\pi k/T) + 1.4 \tag{5}
\]
\[
c_1 = 0.1 \times \sin(\pi k/T) + 0.7 \tag{6}
\]

Where $T$ is the maximum number of iterations and $k$ is the number of current iterations. The algorithm improved by this strategy is temporarily named NLPSO. Inspired by the above research, this paper introduces a novel decay mode of each parameter factor, as shown in the following equation (7) (8) (9)

\[
V_{d_k}^{k+1} = w V_{d_k}^{k} + c_1 r_1 (P_{d_k}^{k} - X_{d_k}^{k}) + c_1 r_1 (P_{g_k}^{k} - X_{d_k}^{k})
\]
\[
c_1 = 0.1 \times (1 - ((\pi k + (-1 + 2 \times r_{11})/108)/T)^2/2!) + ((\pi k + (-1 + 2 \times r_{12})/108)/T)^4/4! - \ldots - ((\pi k + (-1 + 2 \times r_{15})/108)/T)^{10}/10! + R_1/12! + 1.5; \tag{7}
\]
\[
c_2 = -0.1 \times (1 - ((\pi k + (-1 + 2 \times r_{21})/108)/T)^2/2!) + ((\pi k + (-1 + 2 \times r_{22})/108)/T)^4/4! - \ldots - ((\pi k + (-1 + 2 \times r_{23})/108)/T)^{10}/10! + R_2/12! + 1.4; \tag{8}
\]
\[
w = 0.1 \times ((\pi k + (-1 + 2 \times r_{31})/108)/T) - ((\pi k + (-1 + 2 \times r_{32})/108)/T)^3/3! + \ldots - ((\pi k + (-1 + 2 \times r_{33})/108)/T)^7/7! + \ldots - ((\pi k + (-1 + 2 \times r_{34})/108)/T)^9/9! - R_3/11! + 0.7; \tag{9}
\]

Where $T$ is the maximum number of iterations, \{R_1, R_2, R_3\} is a minimum value close to 0, $k$ is the number of current iterations, and \{r_{11}, r_{12}, r_{13}, r_{14}, r_{15}, r_{21}, r_{22}, r_{23}, r_{24}, r_{25}, r_{31}, r_{32}, r_{33}, r_{34}, r_{35}\} is a random number in the range [0, 1]. Its purpose is to add certain orbit around the iterative process to make the particles make small spatial orbits in the optimization process, especially in the local search, to improve the accuracy of the final solution. It can be seen from the above velocity iteration formula that in the optimization of the multidimensional problem, if $c_1$ is kept as large as possible, $c_2$ is as small as possible, $c_1$ and $c_2$ are dynamically changed as above, the optimization performance is better. $c_1$, $c_2$, $w$ is expressed as part of the Taylor series expansion of the cosine function and part of the Taylor expansion of the sine function, which can enhance the local search ability of the PSO, and in the later stage of the algorithm, $c$ and $c$ slow down the global search speed. Therefore, it is also beneficial to reduce the probability of falling into a local optimum. Comparing the intrinsic characteristics of the cosine function and the Taylor expansion of the sine function, the early decay step will be larger than the later decay step. And a larger step can bring a larger search area and ensure the algorithm’s global search ability, while in the final stage of the algorithm, the smaller search step will relax the particle update speed, this will help to improve the local search ability as well as avoid the local optimal situation in the particle optimization process. And in addition to the above advantages, a small range of perturbations is added to the iterations, so that the process of spiraling advances in the process of particle optimization, and the resulting values are more accurate.
3. Experimental results and analysis

For a flexible algorithm, it must have a balanced functional allocation in the local and global search regions, and still have good convergence characteristics after multiple iterations. Based on these characteristics, we test the designed algorithm LDCPSO, in which the function of the test includes the convergence speed and the effect of the final global optimal value. In order to achieve the superiority of the highlighting algorithm, we sampled four different test functions (sphere, ackley, griewank, sum squares) to test the effect of the algorithm[15]. The PSO improvement algorithm introduced above, namely LDWPSO and NLPSO, is used in the following to compare with the algorithm designed in this paper.

3.1. Convergence speed.

The rate of convergence, also known as the iterative criterion, represents the number of iterations that arrive at the best region. At the end of a given time, if the convergence region is not reached, it means that the algorithm has no convergence effect, and then the convergence speed is optimized during the iteration. If the algorithm does not end up with an optimal value, then the algorithm is considered not to converge. The convergence speed can be reflected by the curve gradient. The larger the gradient, the faster the convergence speed. Figure 1 depicts the convergence of the algorithm written by different test functions. In order to highlight the advantages and disadvantages of the algorithm, we also show the convergence characteristics of the two improved PSO algorithms, LDWPSO and NLPSO, and compare the effects of the algorithms designed in this paper to highlight our algorithm. Other improvements are even better in terms of convergence speed.

![Figure 1: Comparison of algorithm convergence effects of different test functions](image)

3.2. Convergence stability.

The convergence stability reflects the distribution of the global optimal value of the particle population after a certain iteration. The more concentrated the optimal value, the better the algorithm optimization effect. On the contrary, the more dispersed the optimal value distribution, the weaker the convergence stability of the algorithm. For an intuitive observation, we plot 100 iterations of the global optimal value distribution as shown in Figure 2. The abscissa indicates the number of iterations, the ordinate indicates the size of the optimal value, and the precision is expanded in the form of a logarithm.
Figure 2: Comparison of algorithm convergence effects of different test functions

For a simple analysis, we calculate the variance and mean of the optimal values obtained after 100 optimizations, and list them in Table 1. The variance reflects the distribution of the optimal values, and the mean is the basis for measuring the variance. The smaller the variance, the smaller the divergence in the convergence process, which means that the algorithm has a better convergence stability.

Table 1: Variance and mean of the optimal values obtained after 100 optimization processes

| Function | Criteria  | NLPSO       | LDCPSO      | LDWPSO     |
|----------|-----------|-------------|-------------|------------|
| sphere   | mean      | 2.76E-07    | 1.65E-07    | 0.0107     |
|          | Var       | 2.11E-12    | 8.10E-13    | 0.00021    |
| ackley   | mean      | 0.310668036 | 0.241747666 | 0.488832521|
|          | Var       | 0.322939265 | 0.319681695 | 0.464588714|
| Sum Squares | mean   | 7.45E-05    | 1.76E-05    | 3.60602E-4 |
|          | Var       | 1.78E-07    | 1.04E-08    | 6.16629E-08|
| griewank | mean      | 1.90E-07    | 6.51E-08    | 0.0402     |
|          | std       | 1.67E-12    | 5.81E-14    | 0.00246    |

As can be seen from Figure 2. Compared to NLPSO, LDWPSO's improved LDCPSO algorithm yields an optimal value closer to the ideal optimal value. It can also be seen from Table 1 that the optimal value distribution obtained by 100 iterations of LDCPSO is more concentrated than that of NLPSO, that is, it has smaller variance, so the algorithm designed in this paper is compared with other improved PSO. Have better optimization capabilities.

4. Conclusion
Particle swarm optimization is a relatively efficient algorithm in modern evolutionary algorithms. Most of the PSO improvement algorithms are dedicated to further improving the optimization algorithm by adjusting the parameters; and a large number of experiments have also proved that the improved algorithm by adjusting the parameters is feasible. Therefore, this paper also adjusts the inertia weighting factor and velocity attenuation factor in the particle swarm iteration formula to improve the convergence ability of the algorithm. By replacing the velocity decay factor in the particle-overlapping formula with a partial cosine-chord Taylor expansion with dynamic decay characteristics, and replacing the inertia weight factor with a fixed parameter value into a
Taylor-expanded form with a partial sinusoidal function with dynamic attenuation characteristics. And the tiny random parameters are added to the replacement formula, so that the particles do the spiral motion of the space in the optimization process to get more accurate values. Finally, the final optimal value of the algorithm is generated by using four test functions and compared with the results of LDWPSO and NLPSO based on convergence stability and convergence speed. It is concluded that our improved algorithm has better optimization ability.

References

[1] J. Kennedy, R. C. Eberhart. Particle swarm optimization, In Proceedings of IEEE International Conference on Neural Networks, IV, IEEE Service Center, Piscataway, NJ, pp. 1942-1948, 1995.
[2] M. Clerc, J. Kennedy. The particle swarm-explosion, stability and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation, vol.6, no.1, pp.58-73, 2002.
[3] Kennedy J. Small Worlds and Mega-minds:Effects of Neighborhood Topology on Particle Swarm Performance. In: The 1999 Congress on Evolutionary Computation. Piscataway, NJ: IEEE Press, 1999. 1931-1938.
[4] 4. X.H. Shi Y.C. Liang H.P. Lee C. Lu Q.X. Wang "Particle swarm optimization-based algorithms for TSP and generalized TSP" Information Processing Letters vol. 103 pp. 169-176 2007.
[5] S.M. Chen C.Y. Chien "Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques" Expert Systems with Applications vol. 38 pp. 14439-14450 2011
[6] R. Dong and W. Pedrycz, "Sparse optimization using a mixed GA-PSO optimization framework," 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, Yantai, 2010, pp. 1862-1866.
[7] C.J. Lin C.H. Chen C.T. Lin "A hybrid of cooperative Particle Swarm Optimization and cultural algorithm for neural fuzzy networks and its prediction applications" IEEE Transaction Systems Man and Cybernetics Part C vol. 39 pp. 55-68 2009.
[8] M. Mahia O. K. Baykanb H. Kodazb "A new hybrid method based on Particle Swarm Optimization Ant Colony Optimization and 3-Opt algorithms for Traveling SalesmanProblem" Applied Soft Computing vol. 30 pp. 484-490 2015.
[9] Y.Hui,Y Zhong and L. Weihua, "Research on Planning Multiple Routes for Air Vehicles Based on Niche Particle Swarm Optimization Technology." 2008 International Conference on Computer Science and Software Engineering, Hubei, 2008, pp. 1211-1214.
[10] Zhifeng Liu, "Technology niche: Concept,characteristic and dimension," 2011 International Conference on Computer Science and Service System (CSSS), Nanjing, 2011, pp. 3636-3639.
[11] Y. Shi, R.C. Eberhart. A modified particle swarm Optimizer, In Proceedings of the IEEE Congress on Evolutionary Computation, Piscataway, NJ: IEEE Press, pp. 69-73, 1998.
[12] L. Yi, “Study on an improved PSO algorithm and its application for solving function problem,” Int. J. Smart Home, vol. 10, no. 3, pp. 51–62, 2016
[13] M.Campos,R.A.Krohling and I. Enriquez, "Bare Bones Particle Swarm Optimization With Scale Matrix Adaptation," in IEEE Transactions on Cybernetics, vol. 44, no. 9, pp. 1567-1578, Sept. 2014.
[14] D. J. Ramsdale and R. A. Howerton, “An improved strategy of PSO for solving multimodal and higher dimensional complicated optimization problems,” in Proc. Int. Conf. Comput. Intell. Commun. Netw. (CICN),2015, pp. 1229–1234.
[15] YAN Chun-man, GUO Bao-long , WU Xian-xiang. Empirical Study of the Inertia Weight Particle Swarm Optimization with Constraint Factor, International Journal of Soft Computing And Software Engineering . vol. 2, no.2, pp.1-8, 2012