

# Application of Chaotic Particle Swarm Optimization in Wavelet Neural Network

Cuijie Zhao<sup>\*1</sup>, Guozhen Wang<sup>2</sup>

<sup>1</sup> Pearl River College, Tianjin University of Finance and Economics, Tianjin, 301811, China

<sup>2</sup> Bohai professional and technical College, Tianjin, 300402, China

\*Corresponding author, e-mail: 36872763@qq.com

## Abstract

Currently, the method of optimizing the wavelet neural network with particle swarm plays a certain role in improving the convergence speed and accuracy; however, it is not a good solution for problems of turning into local extrema and poor global search ability. To solve these problems, this paper, based on the particle swarm optimization, puts forward an improved method, which is introducing the chaos mechanism into the algorithm of chaotic particle swarm optimization. Through a series of comparative simulation experiments, it proves that applying this algorithm to optimize the wavelet neural network can successfully solve the problems of turning into local extrema, and improve the convergence speed of the network, in the meantime, reduce the output error and improve the search ability of the algorithm. In general, it helps a lot to improve the overall performance of the wavelet neural network.

**Keywords:** chaotic particle swarm optimization, convergence speed, wavelet neural network

## 1. Introduction

The optimism theory and method have existed since ancient time, among which, the relatively representative one is the golden section method. Optimism mainly solves the problem of finding the best solution from many solutions. We can defined optimism as: under certain restrictions, to make the problem reach a best measurement, or to find out a set of parameters, and make certain indicators reach the maximum or minimum. As an important branch of science, the optimism method is gaining more and more attention, and plays important roles in many fields, such as engineering technology, electrical engineering, image processing etc. However, in real life application, since the complexity and nonlinearity of many problems, the target functions of these problems are often discrete and of multi-point value, furthermore, the modeling the problem itself is also very difficult. When applying traditional optimizations like Newton method, dynamic programming, branch and bound method, etc. to solve these complex optimism problems, one usually need to traverse the entire search space, which will waste a lot of time, and can not meet the actual requirement in the aspects of the convergence of the problems and the optimization calculation speed. Therefore, in the current field of optimism, the key job is to seek the efficient optimization.

Particle swarm optimization gain the attention of many international scholars in related fields rapidly since its advent. First, Kennedy J and Eberhart R. C. put forward the binary particle swarm optimization in 1997. Then, in 1998, in order to improve the convergence of the algorithm, Shi Y and Eberhart R C introduced the inertia weight parameter into the speed item of the PSO and proposed to dynamically adjust the inertia weight to balance the convergence speed during the process of evolution. This algorithm is called the standard PSO. Then, they put forward the linear decreasing inertia weight LDW-PSO, however, if it deviated from the overall optimum solution in the initial state, then the linear decreasing will continuously enhance the local search ability, which may end up with local optimism. Clerc, et al in 1999, put forward the CF-PSO with shrinkage factors by introducing shrinkage factors into the evolution equation to ensure the convergence of the algorithm. And in order to overcome the problem of the premature convergence of LDW-PSO, they put forward the random inertia weight RandW-PSO, so that under a certain range of accuracy, the multimodal function can quickly converge. At present, the improvement of particle swarm optimization mainly includes: first, introducing a variety of mechanism into the particle swarm optimization to study various of improved PSO;

second, combining the PSO with other intelligent optimization and studying a variety of mixed algorithms to complement each other and improve the performance of the algorithm [1].

This paper introduces the definitions and theories related to the wavelet neural network, as well as some frequently-used training methods of wavelet neural network. It also elaborates the principle, definition and basic working process of the particle swarm optimization, and makes a detailed explanation on the improving method applied by the paper. Then, it introduces the basic idea and design approach of the method of optimizing the wavelet neural network with particle swarm. It shows the feasibility and superiority of the chaotic particle swarm optimization through comparative experiments, and proves the feasibility and superiority of the improved algorithm it proposed by applying the chaotic particle swarm optimized wavelet neural network to simple target tracing.

## 2. Basic Particle Swarm Optimization

### 2.1. Basic Idea of the Basic Particle Swarm Optimization

The basic idea of the basic particle swarm optimization is: the potential solution of every optimism problem is the search space particle. Every particle has fitness value determined by the optimized function, and has a speed vector determining its flying direction and distance. Then these particles will follow the search in the solution space of the current optimized particle. The initialization of the particle swarm optimization is a swarm of random particles, then it finds out the optimized solution through iteration. In every iteration, the particles update themselves by tracing two extrema. One is the optimized solution found by the particle itself so far, which is the individual optimized solution. The other one is the optimized solution found by the whole particle swarm so far, which is the overall optimized solution. Obviously, the particle swarm optimization also bases on individual cooperation and competition to complete the search of the optimized solution in a complex space. It is an evolutionary computation technique based on swarm intelligence method. The particle swarm optimization conducts search by each particle following the optimized particle. Therefore, it is simple and easy, and does not need to adjust many parameters [2].

Advantages of the PSO:

- (a) No crossing and mutation operations, depending on particle speed to complete search, high convergence speed;
- (b) Applying the method of simultaneously addressing more than one particle in the particle swarm to simultaneously search certain area of the design space, having the nature of parallelism;
- (c) Adopting real number coding, solving the problem directly on the problem domain, less parameters to be setted, easy to adjust, so the algorithm is simple, and easy for implementation;

Disadvantages of the PSO:

- (a) Easy to turn into local extrema;
- (b) Low search accuracy;
- (c) The high efficiency information sharing mechanism might lead to the overconcentration of particles when they are seeking for the optimized solution, which makes all the particles move to certain overall optimized point, and cannot be applied to multimodal function optimization;
- (d) When solving problems of optimization with discrete variables, the rounding of the discrete variables may appear great errors;
- (e) The algorithm theory is not perfect, especially lacking practical guidelines for specific practice.

The metathetic description is: each particle is considered as a point in the D dimensional space, the location of the No.i particle is marked as  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ , the particle's individual extrema I is marked as  $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ , the overall extrema's subscript is reprinted by "g", particle i's speed is marked as  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ , particles will adjust their speed and location according to the following equations:

$$v_{id}^{t+1} = v_{id}^t + c_1 r_1 (P_{id} - x_{id}^t) + c_2 r_2 (P_{gd} - v_{id}^t) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (2)$$

Among which, “ $d=1,2\dots D$ ,  $i=1,2\dots m$ ” is the swarm scale. “ $t$ ” is the current evolution algebra. “ $c_1$ ” and “ $c_2$ ” are acceleration constants, which are positive constants. “ $r_1$ ” and “ $r_2$ ” are two random number within the range of  $[0, 1]$ . Moreover, in order to control the particle speed, one can set a speed limit  $V_{\max}$ , that is, in equation (1), when  $V_{id} > V_{\max}$ , consider  $V_{id} = V_{\max}$ , when  $V_{id} < -V_{\max}$ , consider  $V_{id} = -V_{\max}$ , the first part of equation (1) is the previous speed item, the second part is the cognition part of the particle itself, that is, the impact on the current location of the particle’s historical best location, which represents the information sharing and cooperation among particles[3],[4].

### 2.1. Basic PSO Procedure

- (a) Randomly initialize the location and speed of the particle swarm, it is usually generated randomly within the allowed range. The pbest coordinate of each particle is its current location. Calculate its corresponding individual extremum (i.e. the individual fitness value),  $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ . The overall extremum (i.e. the overall fitness value)  $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$  is the best of the individual extrema. Mark the number of the particle with the best value as “ $g$ ”, and set the gbest with the current location of the best particle.
- (b) Calculate each particle’s fitness value.
- (c) Compare each particle’s fitness value with its individual extremum, if better, update the current individual extremum.
- (d) Compare each particle’s fitness value with the overall extremum, if better, update the current overall extremum.
- (e) Update each particle’s location and speed according to equation (1) and (2).
- (f) If not reach the previous steted termination standard (usually set as the largest number of iteration  $T_{\max}$ ), then return to step (2); if reached, then stop calculating [5].

## 3. Improvement of the Particle Swarm Optimization Based on Chaotic Mechanism

### 3.1. Idea of the Chaotic Particle Swarm Procedure

Strictly speaking, the chaos phenomenon refers to the internal random action acted by a system of complete certainty and without any random factors. The chaotic optimization conducts search mainly by making use of the ergodicity of the chaotic motion, so as to avoid turning into local minimum. The chaotic optimization possesses features like randomness, ergodicity, regularity, nonlinearity and long-term behavior unpredictability, etc. Track ergodicity means that the chaos sequence can go through all the states within certain range without repetition. It is the fundamental starting point of the function optimization through chaos. Usually, the search process based on chaos dynamic is divided into two stages. First, the ergodicity track generated by iteration with certainty inspects the entire solution space. When certain termination condition is met, and the discovered best state during the search is considered as close to the optimal solution, and it is regarded as the search starting point of the second stage. Second, taking the result gained by the first stage as the center, conducting further in-depth local search by adding slight perturbations, until the termination standard is met, among which, the added slight perturbations can be the chaos variables, or random variables based on Gaussian distribution, Cauchy distribution or uniform distribution, etc. and also can be the offset value generated by the calculation based on the gradient descent mechanism. Based on the above mentioned idea, adopting methods similar to carrier wave to introduce the chaos variables generated by Logistic mapping into the optimized variables, in the meantime, transferring the ergodic range of the chaos motion into the optimized variable domain, then searching with chaos variables[6],[7].

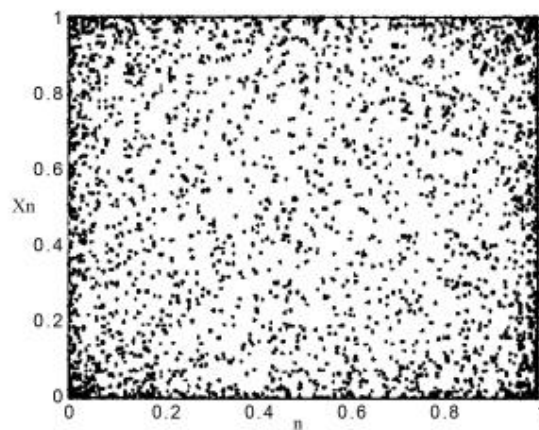


Figure 1. The periodicity of chaotic variables

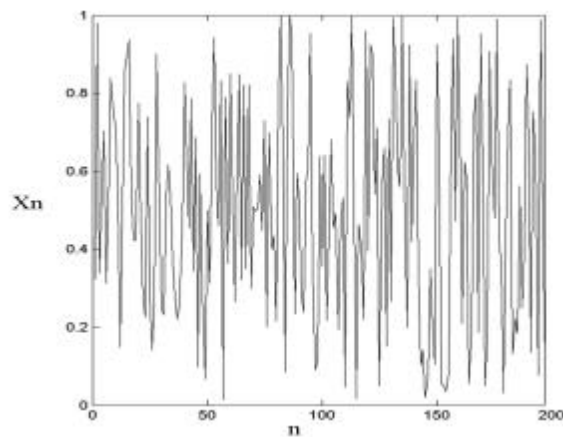


Figure 2. The randomness of chaotic motion

### 3.1. Basic Procedures of The Improved Particle Swarm Optimization

The search process of the two stages combined with chaotic particle swarm optimization, the overall search steps are as follows[8]:

- (a) Set the particle swarm size as "N" and the maximum number of the iteration, and randomly initialize the location and speed of the particles within the range of feasible adoption.
- (b) evaluate the fitness of each particle; set the particle fitness ranging first as the overall optimum; the initial location of the particle is the particle's individual extremum.

$$V_i^{(k+1)} = V_i^{(k)} + c_1 \times rand() \times (P_{ibest} - X_i) + c_2 \times Rand() \times (Gbest - X_i) \quad (3)$$

$$X_i^{(k+1)} = X_i^{(k)} + V_i^{(k+1)} \quad (4)$$

- (c) update the speed and location of the particles according to equation(3)and(4).
- (d) evaluate the fitness of each particle; compare it with its previous fitness, update the individual extremum with the better fitness; compare the fitness of the current optimized particle with its pervious fitness, update the overall optimal value with the better fitness.
- (e) reserve the first N/5 particles of the swarm.
- (f) update the locations of these particle through chaos local search and CLS result. If the termination standard is met, output the current optimized solution.

- (g) narrow the search space, and randomly generate  $4N/5$  new particles in the narrowed search space.
- (h) constitute a new swarm with the updated particles through CLS and these  $4N/5$  new particles. (9) make  $k=k+1$ , return to step(3).

#### 4. Experiment Simulation And Related Applications

##### 4.1. Experimental Model Construction And Data Analysis

In order to verify the effectiveness of the chaotic particle swarm optimization proposed by this paper regarding the application in optimizing wavelet neural network, this paper adopts simulation software to conduct simulation experiments, thus to verify the method proposed thereby. It provides a function group, as follows:

$$f(x) = \begin{cases} -2.186x - 1.286 & -1 \leq x \leq -0.2 \\ 4.246x & -0.2 \leq x \leq 0 \\ e^{-0.5x-0.5} \sin[(3x+7)x] & 0 \leq x \leq 1 \end{cases} \quad (5)$$

In interval  $[-1,1]$ , randomly generate 50 points with same intervals, and mark as  $x_k$ , in which,  $k=1,2,\dots,50$ . In this model, respectively apply CPSO optimized wavelet network and Basic PSO optimized wavelet network to train the functions. On the hidden layer, we choose the wavelet function of Morlet wavelet, since the Morlet function possesses the features of continuous conductivity and good video localization, moreover, its function expression is simpler. Then apply the chaotic particle swarm optimized wavelet network to train the functions, in which, respectively select the number of the particle as “ $N=50$ ”, learning factor as “ $c1=c2=2$ ”, the maximum and minimum inertia factors as “ $\omega_{\max} = 0.9$ ” and “ $\omega_{\min} = 0.6$ ”, the maximum number of iteration as “5000”.

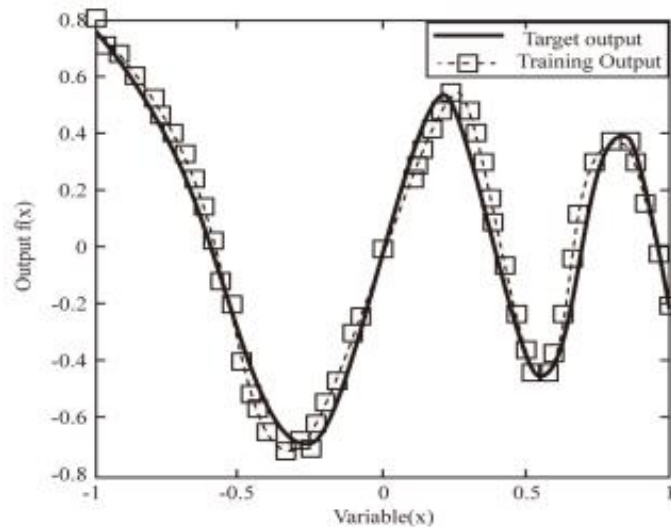


Figure 3. B CPSO optimized wavelet network output

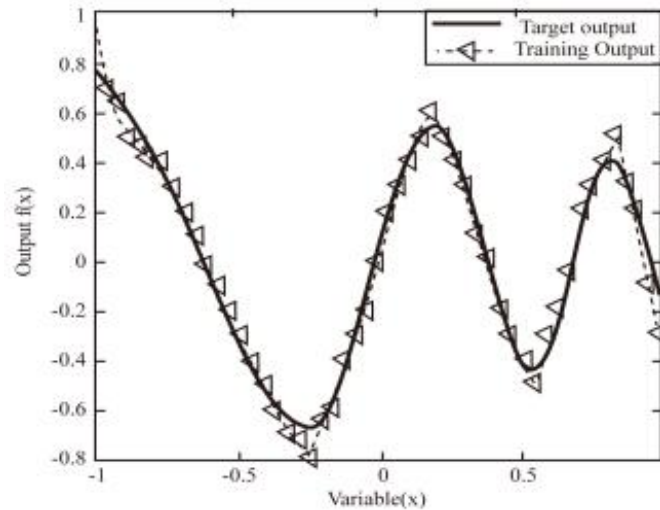


Figure 4. Basic PSO optimized wavelet network output

The above are the output curves generated from the same function trained relatively by the CPSO optimized wavelet network and the BPSO optimized wavelet network. It can be seen that such wavelet neural networks possess fine fitting. Its target value and the training output value are basically consistent, which avoids the risk of turning into local extrema. The following are the output error curves respectively adopting the CPSO and BPSO optimized wavelet neural network. As to the CPSO, before the 2500th training, the error decreases faster, and after the 2500th training, the error is basically stabilized, and the change is relatively small. The error value is approaching zero. It can be seen that at the moment, the network has been gradually began to converge. As to the BPSO, it is not until after the 3500th training, the error become stable, and the network begin to converge. However, the error at that moment is still relatively bigger. Through the comparing experiments, it can be seen that the wavelet neural network based on CPSO is better than the wavelet neural network based on BPSO, for that it accelerates the convergence speed, improves the error accuracy, and avoids turning into the local extrema. Since the wavelet neural network possesses the feature of high-speed convergence, during the training process, the optimal number of convergence is within 3000, and the error accuracy is also the less the better, otherwise it will seriously affect the structure of the network, and lead to a lose structure, weakened generalization ability, and even result in the “Butterfly Effect” of the network output.

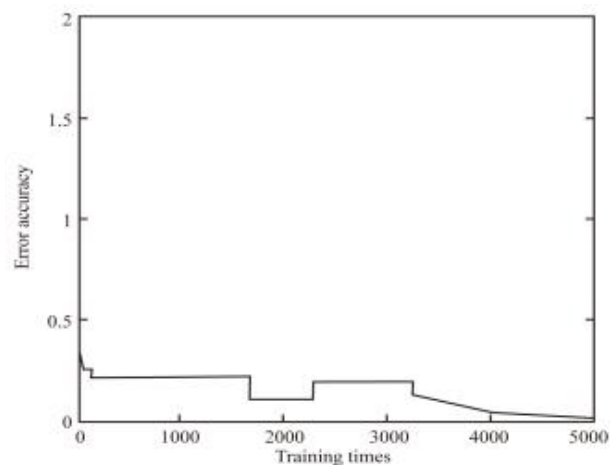


Figure 5. CPSO optimized wavelet network error curve

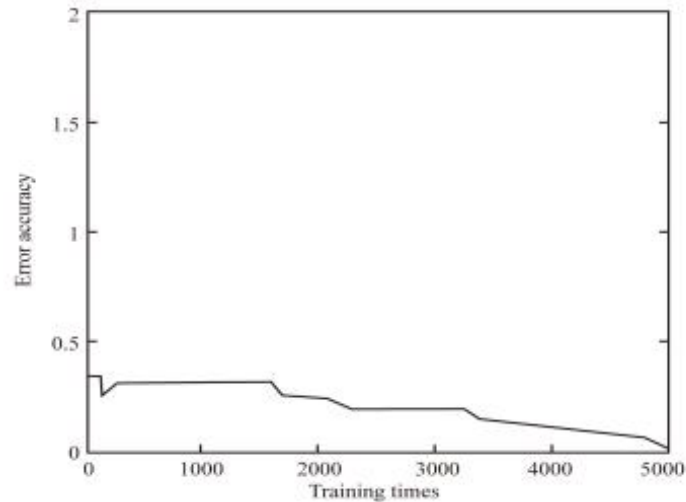


Figure 6. Basic PSO optimized wavelet network error curve

In order to assure the reliability of the experiment data, this paper conducts several repeated experiments to this model, and the experiments data are as follows.

Table 1. The data comparison of the two optimization

Experiment	Training error of BPSO	Training error of CPSO
First	0.0835	0.0475
Second	0.0087	0.0073
Third	0.05354	0.01267

## 5. Conclusion

This paper mainly introduces the particle swarm optimization. It starts from the introduction of basic theories, and gradually explores into the number selection methods, and then elaborates the whole algorithm by analyzing the pros and cons of the algorithm and introducing the algorithm procedure. In order to verify the superiority of the algorithm proposed here, this paper adopts the chaotic particle swarm optimization and the basic particle swarm optimization to respectively calculate the minimum of two testing functions. It can be seen from the experiment data analysis that the chaotic particle swarm optimization can not only improve the error accuracy, but also accelerate the convergence speed, and enhance the ability of avoiding local extrema. By conducting the comparative simulation experiments, and respectively using the basic particle swarm optimized wavelet neural network and the chaotic particle swarm optimized wavelet neural network to train the functions, it shows that the chaotic particle swarm optimized wavelet neural network possessed not only higher convergence speed, but also smaller error accuracy, and is a feasible training method.

## References

- [1] Lin Wang, Bo Yang, et al. Improving particle swarm optimization using multi-layer searching strategy. *Information Sciences*. 2014; 274(1): 70-94.
- [2] Guohua Wu, Dishan Qiu, et al. Superior solution guided particle swarm optimization combined with local search techniques. *Expert Systems with Applications*. 2014; 41(16): 7536-7548.
- [3] Nabila Nouaouria, Mounir Boukadoum. Improved global-best particle swarm optimization algorithm with mixed-attribute data classification capability. *Applied Soft Computing*. 2013; 21: 554-567.
- [4] Sarthak Chatterjee, Debdipta Goswami, et al. Behavioral analysis of the leader particle during stagnation in a particle swarm optimization algorithm. *Information Sciences*. 2014; 279(20): 18-36.
- [5] Xinchao Zhao, Ziyang Liu, et al. A multi-swarm cooperative multistage perturbation guiding particle swarm optimizer. *Applied Soft Computing*. 2012; 22(9): 77-93.

- [6] Xiang Yu, Xueqing Zhang. Enhanced comprehensive learning particle swarm optimization. *Applied Mathematics and Computation*. 2014; 242(1): 265-276.
- [7] Jianli Ding, Jin Liu, et al. A particle swarm optimization using local stochastic search and enhancing diversity for continuous optimization. *Neuro computing*. 2014; 137(5): 261-267.
- [8] Amer Fahmy, Tarek M. Hassan, et al. Improving RCPSP solutions quality with Stacking Justification–Application with particle swarm optimization. *Expert Systems with Applications*. 2013; 41(13): 5870-5881.