TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.7, July 2014, pp. 4973 ~ 4980
DOI: 10.11591/telkomnika.v12i7.4477

Short-Term Prediction of Wind Power Based on an Improved PSO Neural Network

Hong Zhang*, Guo Zhao, Lixing Chen, Bailiang Liu
School of Electrical Engineering, Southeast University, Nanjing, Jiangsu, 210096, China
*Corresponding author, e-mail: hazh0216@163.com

Abstract

Connecting wind power to the power grid has recently become more common. To better manage and use wind power, its strength must be predicted precisely, which is of great safety and economic significance. In this paper, the short-term power prediction of wind power is based on self-adaptive niche particle swarm optimization (NPSO) in a neural net. Improved PSO adopts the rules of classification and elimination of a niche using a self-adaptive nonlinear mutation operator. Compared with the traditional method of maximum gradient, NPSO can skip a local optimal solution and approach the global optimal solution more easily in practice. Compared with the basic PSO, the number of iterations is reduced when the global optimal solution is obtained. The method proposed in this paper is experimentally shown to be capable of efficient prediction and useful for short-term power prediction.

***Keywords**: PSO, niche, mutation operator, short-term power prediction, neural net*

1. Introduction

Wind power is a renewable energy source that is becoming increasingly popular for application in the grid because of its environmentally friendly and low-cost properties. However, because the power fluctuates with the wind strength, connecting wind power to the grid is challenging. To make the use of wind power reasonable and reduce its negative effects on the power grid, scientists in many countries have been working to develop methods to predict the power of the wind generators, which is of great importance to the economical distribution and operation of the power grid. Denmark was among the first countries to develop a system of power prediction for wind power [1]. Prediktor is the wind power work prediction system developed by Ris National Laboratory of Denmark, which mainly applies physical models [2]. ANEMOS, a research project sponsored by the European Union, combines physical and statistical methods [3]. The eWind is a system developed by AWS Truewind in America [4]. The highly precise mathematical models of atmospheric physics and adaptive statistical models are combined; the velocity of the wind and the power of the wind power plants have been investigated in studies based on time serials and neural networks [5-7]. The back propagation (BP) neural network is the mostly widely used neural network. The classic BP learning law is typically used in BP neural networks to determine network connection weights. However, this technique is slow in practice and may lead to a local optimal solution. In this paper, the short-term power prediction of the wind power is based on self-adaptive niche particle swarm optimization (NPSO) in a neural network. Improved PSO adopts the rules of classification and elimination of a niche and uses a self-adaptive nonlinear mutation operator. Compared with the traditional method of maximum gradient, NPSO can skip a local optimal solution and approach the global optimal solution more easily in practice. Compared with the basic PSO, the number of iterations is reduced when the global optimal solution is obtained. The method proposed in this paper is experimentally shown to be capable of efficient prediction and useful for short-term power prediction.

Received September 25, 2013; Revised January 26, 2014; Accepted February 10, 2014

2. Theoretical Basis for Improved Self-Adaptive PSO

2.1. Theoretical Basis for Basic Particle Swarm Optimization

In 1995, J. Kennedy and R. C. Eberhart developed PSO [8, 9], which aims to simulate a simple social system, such as a bird flock searching for foods, to study and explain complex social behavior. In basic PSO, every candidate solution is compared to a bird searching the space and is called a particle. The position and velocity of a particle is denoted as $X_i = (x_{i1}, x_{i2}, ggg, x_{iD})$ and $V_i = (v_{i1}, v_{i2}, ggg, v_{iD})$, respectively. At the initial stage, a swarm of particles is randomly selected. Then, the swarm is updated according to the best known positions of individual particles and the entire swarm. The equations defining the position and velocity of the particles are shown below:

$$\begin{aligned} v_{id}(k+1) &= wv_{id}(k) + c_1 r_1 (p_{id}(k) - x_{id}(k)) \\ &\quad + c_2 r_2 (g_{id}(k) - x_{id}(k)) \\ x_{id}(k+1) &= x_{id}(k) + v_{id}(k) \end{aligned} \tag{1}$$

$$w = (w_{ini} - w_{end})(k_{max} - k) / k_{max} + w_{end} \tag{2}$$

In Equations (1) and (2), p is the best known position of a particle and g is the best known position of the entire swarm; i = 1,2⋯n; D is the dimension of a particle; k is the k-th iteration; d is the d-th dimension; kmax is the maximum number of iterations; w is the inertia weight; wini is the initial inertia weight; wend is the final inertia weight; c_1 and c_2 are learning factors; and r1 and r_2 are uniform random numbers in the range [0, 1].

2.2. Adaptive Niching Particle Swarm Optimization

Basic PSO may lead to premature convergence to a local optimum, thus affecting the quality of the solution. The probability of prematurity can be reduced by mixing basic PSO with other algorithms or by adopting a comprehensive strategy. Niche technology simulates ecological balance, i.e., a species evolves to establish a surviving niche in a larger environment, which reflects the evolutionary rule of survival of the fittest. Goldberg and Richardson described niche technology based on a sharing mechanism in [10] and Brits et al., described NPSO in [11]. The following formulae are based on adaptive NPSO:

$$\begin{aligned} v_{id}(k+1) &= wv_{id}(k) + c_1 r_1 (p_{id}(k) - x_{id}(k)) + \\ &\quad c_2 r_2 (g_{id}(k) - x_{id}(k)) + c_3 r_3 (\overline{p}_{id}(k) - x_{id}(k)) \\ x_{id}(k+1) &= x_{id}(k) + v_{id}(k) \end{aligned} \tag{3}$$

$$w = (w_{ini} - w_{end}) \exp(-1/[1 + (1 + \frac{k}{k_{max}})] + w_{end} \tag{4}$$

In Equation (3) and (4), $\overline{p}$id is the best known position of a sub-swarm; c_3 is the learning factor; and r_3 is a uniform random sequence in the range [0, 1]. The diversity selection of the swarm regulates the adaptability of individual particles by reflecting the sharing functions among them, upon which the later evolutionary process is selected, to create an evolved environment and to realize swarm diversity.The adaptive mutation operator adopts an adaptive non-linear decreasing inertia weight function[12]. The decreasing velocity of the inertia weight is accelerated in the first iteration of the algorithm to achieve a more efficient solution.

2.3. The Main Steps of the Improved PSO Algorithm

The main steps of the improved PSO algorithm are as follows:

Step 1: Start.
Step 2: Generate the initial population by chaotic iteration.
Step 3: Initialize parameters.

Step 4: Select a particle randomly and divide all of the particles evenly into m small niche subpopulation based on adaptive functions.

Step 5: Establish the initial velocity of the particles randomly.

Step 6: Set the initial position of the present particle as the individual historical optimal value, pbx; set the historical optimal value of the optimal individual in each subpopulation as the population historical optimal value, $\overline{p}$ bx; and set the historical optimal value of all of the particles as the overall historical optimal value, gbx.

Step 7: When k is less than the maximum number of iterations, the following cycle of operations is performed for each subpopulation:

a) Calculate the inertia weight, threshold value, and calibration coefficient.

b) Update the velocity and position of every particle within each subpopulation.

Step 8: Adopt a niche elimination strategy.

Step 9: Determine whether the convergence conditions are met; if so, stop the calculation and output the results; if not, go to Step 6.

Step 10: End.

The flow chart in Figure 1 illustrates the main steps of the improved PSO algorithm.

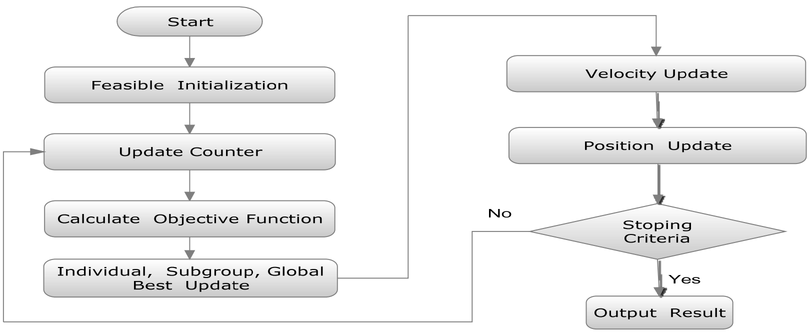


Figure 1. Flowchart of the Improved PSO Algorithm

2.4. Testing the Improved PSO Algorithm Using Standard Test Functions

To test the performance of the improved PSO algorithm, two standard testing functions are selected: the 2-D Rosenbrock function and 2-D Rastrigin function. Standard testing functions are commonly employed in the optimization literature to evaluate the efficiency of new algorithms [13, 14]. The two standard testing functions have numerous local optima and a global minimum that is very difficult to locate.

2.4.1. The 2-D Rosenbrock Function

The 2-D Rosenbrock function is given by Equation (5):

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \tag{5}$$

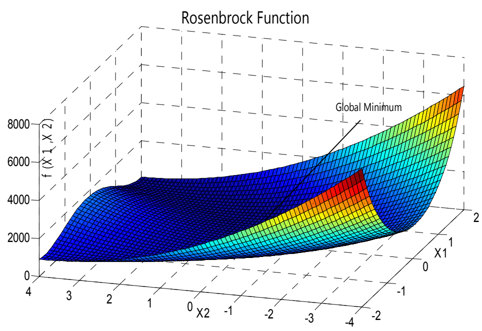


Figure 2. Graph of the Rosenbrock Function

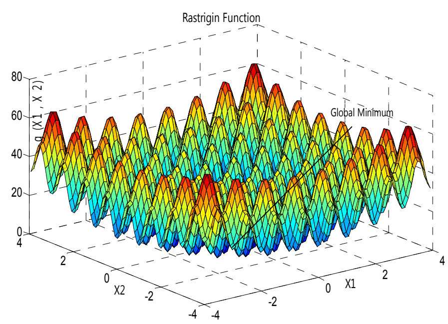


Figure 3. Graph of the Rastrigin Function

For the 2-D Rosenbrock function in this paper, the global minimum is fglobal = 0 as x = (1,1), but the valley in which the minimum lies has steep edges and a narrow ridge. The tip of ridge is also steep. Figure 2 illustrates the main characteristics of the 2-D Rosenbrock function.

2.4.2 The 2-D Rastrigin function

The 2-D Rastrigin function is given by Equation (6):

$$g(x_1,x_2)=x_1^2+x_2^2-10[\cos(2\pi x_1)+\cos(2\pi x_2)]+20 \tag{6}$$

For the 2-D Rastrigin function employed in this paper, the global minimum is fglobal = 0 when x = (0,0). There are many local minima arranged in a lattice configuration, as shown in Figure 3. Figure 3 illustrates the main characteristics of the 2-D Rosenbrock function. The global minima of the 2-D Rosenbrock function and 2-D Rastrigin function can be located by simulation computation based on the improved PSO algorithm. Thus, the model based on the improved PSO can be used in practice.

3. Neural Network Model Based on Self-Adaptive Niche PSO

3.1. Theoretical Basis for the Basic Neural Network

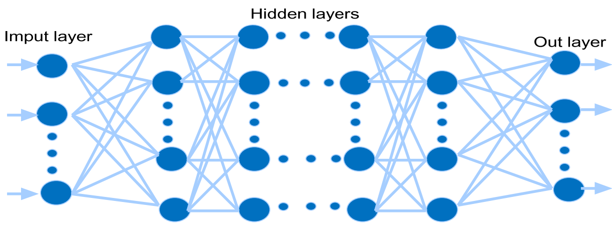


Figure 4. Artificial intelligence neural network.

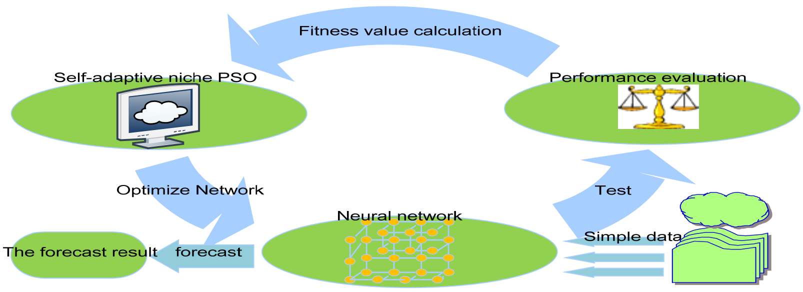


Figure 5. Learning process of the NPSO neural system

Since the insightful study of the neural network in the 1980s [15-16], neural networks have been widely applied to the industrial field. The artificial intelligence neural network is a complex nonlinear system. The artificial neural network is also a nonlinear mapping system with good self-adaptability and can be used to identify any complicated state or process. Figure 4 describes a simple artificial intelligence neural network. The basic principle of the neural network model to process information is that the input signal X(i) acts on the intermediate node (the hidden layer), leading to a result from the output node, which utilizes a non-linear transformation and generates an output signal Y(k) by adjusting W(ij), the value relating to the input nodes and hidden layer nodes.T(jk), the value relating to the hidden layer nodes, the output node, and their respective values, is reduced by repetitive learning training; the network parameters (weights and threshold values) relating to the minimum error are determined. The training continues until the error reaches the threshold value. The BP neural network model is expressed in Equation (7):

$$O_j = f(\sum W_{ij} \times X_i - q_j)$$
$$Y_k = f(\sum T_{jk} \times O_j - q_k) \qquad (7)$$

Where f is the activating function and q is the neural cell threshold. Figure 5 illustrates the performance of prediction based on the improved PSO neural network.

3.2. The Steps of the Prediction Algorithm Based on Self-Adaptive NPSO Neural Network

The main steps of prediction algorithm based on the self-adaptive NPSO neural network are as follows:

Step 1: Start.
Step 2: Input the initial values and target values of the samples.
Step 3: Initialize the coupling weight values and thresholds.
Step 4: Convert connection weights and thresholds to particles.
Step 5: Divide the initial population into several small niche subpopulations.
Step 6: Calculate the adaptive values of the particle swarm.
Step 7: Determine the best known positions of the individuals, sub-populations, and overall population.
Step 8: Adjust the adaptability and inertia weight and update the velocity and position of the particles.
Step 9: Judge whether the niche update conditions are met. If not, go to Step 6.
Step 10: Run the niche optimization rules.
Step 11: Judge whether the maximum time is reached. If not, go to Step 6.
Step 12: Determine the coupling value and threshold.
Step 13: End.

The flow chart in Figure 6 illustrates the main steps of the prediction algorithm based on the self-adaptive NPSO neural network.

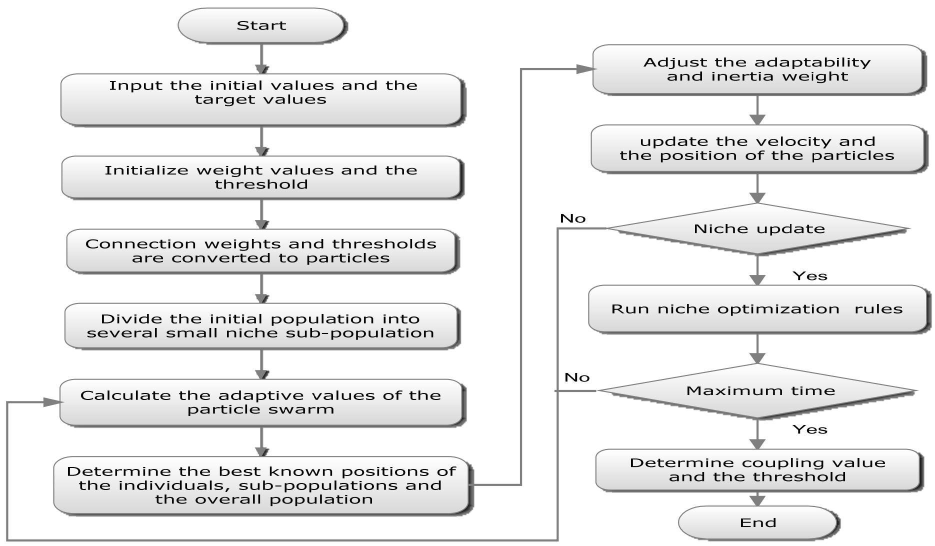
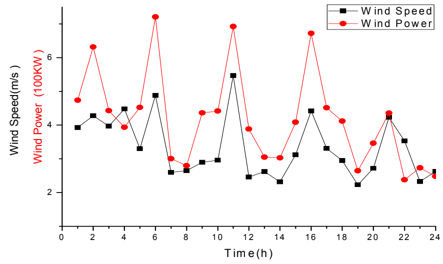


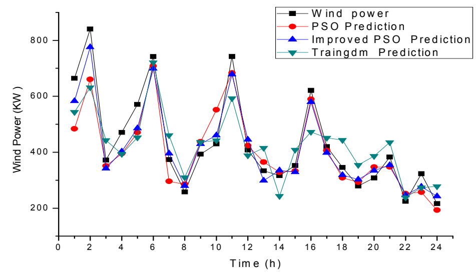
Figure 6. Flow Chart of the Prediction Algorithm Based on the Self-adaptive NPSO Neural Network

4. Predictive Analysis of the Neural Network Based on Self-Adaptive NPSO

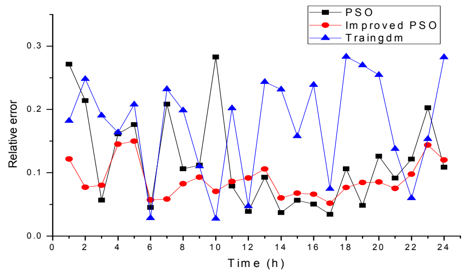
The power prediction model is established by the neural network based on self-adaptive NPSO (improved PSO). The power of a wind generator in Dongtai (Jiangsu, China) was predicted in 2008 based on the meteorological data and data for the power generated by the wind generator in the previous months. The predictive models for the neural network are based on PSO, NPSO, and Traingdm. First, the original data related to wind speed and wind power must be processed and normalized by advanced mathematical methods [17]. For example, the model will observably decrease systematic error when the origin data have been processed by the Kalman filter described in the literature [18]. All predictive models are trained beforehand. Figure 7 illustrates the main characteristics obtained from different prediction models 3 h ahead. Figure 7(a) illustrates that higher wind powers generally correspond to higher wind speeds. Figure 7(b) presents the measured power and forecasted power based on PSO, improved PSO, and Traingdm. Comparing the results of the three methods, the forecasted wind power curve based on the improved PSO is the closest to the measured power in Figure 7(b). Figure 7(c) presents the relative error from different predictions. The minimum relative error of the forecast wind power is obtained by the improved PSO method.



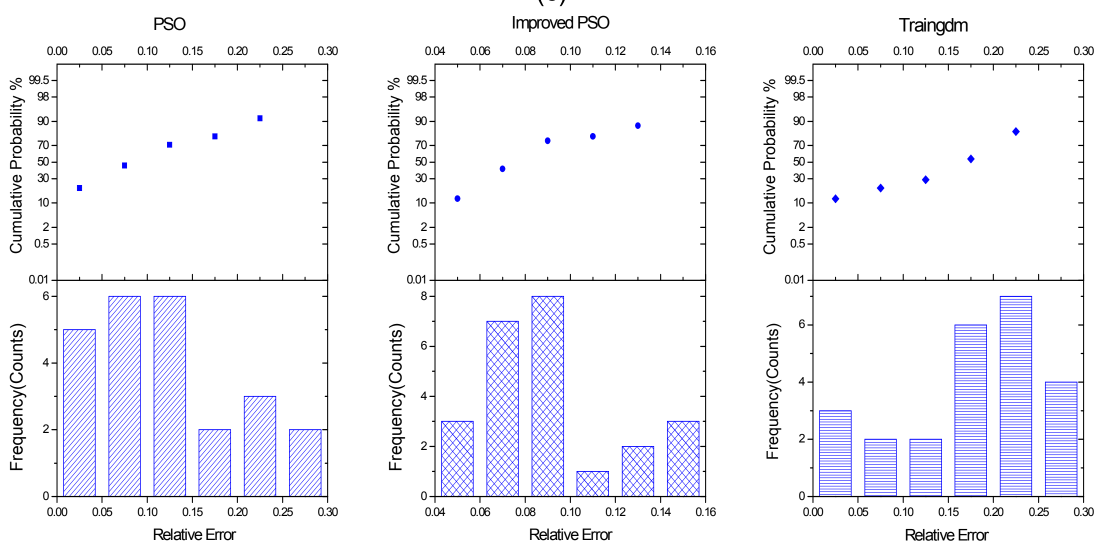
(a)



(b)



(c)



(d)

Figure 7. Main Characteristics Obtained from the Three Different Predictions. (a) Wind speed and wind power. (b) The measured power and forecasted power based on PSO, improved PSO, and Traingdm. (c) Relative error from different prediction models based on PSO, improved PSO, and Traingdm. (d) Frequency and probability from different prediction models based on PSO, improved PSO, and Traingdm

Figure 7(d) illustrates the frequency and probability from different prediction models based on PSO, improved PSO, and Traingdm. The probability of a relative error of less than 0.1 for the improved PSO method is greater than those of PSO and Traingdm. Thus, the prediction accuracy of the improved PSO method is better than those of PSO and Traingdm. The absolute error, relative error, mean absolute error, mean relative error, standard deviation, relative standard deviation, and interval probability in this paper are illustrated by Equation (8)[19]. From Figure 7, the statistical data indicate that the prediction power based on improved PSO is better than those based on PSO and Traingdm.

$$\begin{aligned}
&\text{absolute error} = |\,\text{forecast}\,(i) - \text{measure}\,(i)| \\
&\text{mean absolute error} = \frac{|\,\text{forecast}\,(i) - \text{measure}\,(i)|}{n} \\
&\text{relative error} = \frac{|\,\text{forecast}\,(i) - \text{measure}\,(i)|}{\text{measure}\,(i)} \\
&\text{mean relative error} = \frac{\text{relative error}}{n} \\
&\text{standard deviation} = \sqrt{\frac{\sum_{i=1}^{n}\left(\text{forecast}\,(i) - \frac{1}{n}\sum_{i=1}^{n}\text{forecast}\,(i)\right)^2}{n-1}} \\
&\text{relative standard deviation} = \frac{\text{standard deviation}}{\frac{1}{n}\sum_{i=1}^{n}\text{forecast}\,(i)} \\
&\text{interval probability} = \frac{\text{frequency(counts)}}{n}
\end{aligned} \tag{8}$$

4. Conclusion

In this paper, a predictive model for neural networks based on self-adaptive NPSO is established. Using model analysis, experiments, and comparison with predictive models based on other algorithms, the model is shown to be more precise than the other two models considered; furthermore, it has the lowest absolute variance, demonstrating its effectiveness. The reliability of the model is significantly related with the precision of the weather forecast, With computers becoming increasingly powerful, the predictive method of the neural network based on hybrid multi-algorithms will be most useful in the future.

Acknowledgements

This paper is supported by the National High Technology Research and Development Program (863 Program) (2011AA05A107).

References

[1] Landberg L, Watson SJ. Short-term prediction of local wind conditions. *Boundary-Layer Meteorology*. 1994; 70(1-2): 171-195.
[2] Landberg L, Giebel G, Nielsen HA, et al. Short-term Prediction: An Overview. Wind Energy. 2003; 6(3): 273-280.
[3] Marti I, Kariniotakis G, Pinson P, et al. Evaluation of advanced wind power forecasting models–results of the Anemos project. 2006.
[4] Porter K, Rogers J. *Status of Centralized Wind Power Forecasting in North America.* NREL/SR-550-47853. Golden, CO: National Renewable Energy Laboratory, 2010.
[5] Yang X, Xiao Y, Chen S. *Wind speed and generated power forecasting in wind farm.* Proceedings of the CSEE 2005, 25: 1-5.
[6] Ding M, Zhang L, Wu Y. Wind speed forecast model for wind farms based on time series analysis. *Electric Power Automation Equipment.* 2005; 8: 32-34.
[7] Xiao Y, Wang W, Huo X. Study on the time-series wind speed forecasting of the wind farm based on neural networks. *Energy Conservation Technology.* 2007: 2: 2.
[8] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science.* Nagoya; 1995: 39-43

[9] Kennedy J, Eberhart R. Particle swarm optimization. *IEEE International Conference on Neural Networks*. 1995: 1942-1948.
[10] Goldberg DE, Richardson J. Genetic algorithms with sharing for multimodal function optimization. Cambridge. 1987: 41-49.
[11] Brits R, Engelbrecht AP, Van den Bergh F. A niching particle swarm optimizer. Singapore. 2002: 692-696.
[12] Gao Y, Ren Z. *Adaptive Particle Swarm Optimization Algorithm with Genetic MutationOperation.* In Third Internation Conference on Natural Computation, Haikou, Japan. 2007: 211-215.
[13] Faerman M, Birnbaum A, Berman F, et al. Resource allocation strategies for guided parameter space searches. *International Journal of High Performance Computing Applications.* 2003; 17(4): 383-402.
[14] Potter MA. The design and analysis of a computational model of cooperative coevolution. *Citeseer.* 1997.
[15] Mcclelland JL, Rumelhart DE, Group PR. Parallel distributed processing. *Explorations in the microstructure of cognition*. 1986; 2.
[16] Hopfield JJ, Tank DW. Computing with neural circuits- A model. *Science*. 1986; 233(4764): 625-633.
[17] Ogasawara E, Martinez LC, de Oliveira D, et al. Adaptive normalization: A novel data normalization approach for non-stationary time series. *The 2010 International Joint Conference on Neural Networks (IJCNN)*. 2010: 1-8.
[18] Louka P, Galanis G, Siebert N, et al. Improvements in wind speed forecasts for wind power prediction purposes using Kalman filtering. *Journal of Wind Engineering and Industrial Aerodynamics*. 2008, 96(12): 2348-2362.
[19] Spiegel M, Schiller J, Srinivasan A. *Schaum's Easy Outline of Probability and Statistics.* McGraw-Hill, 2002.