Send Orders for Reprints to reprints@benthamscience.ae

The Open Electrical & Electronic Engineering Journal, 2014, *8*, 723-728

Open Access

Research on Power Load Forecasting Model Based on Hybrid Algorithm Optimizing BP Neural Network

Chenhao Niu[1], Xiaomin Xu[*,2], Yan Lu[2] and Mian Xing[3]

[1]*Information Management Office, North China Electric Power University, Beijing 102206, China*

[2]*School of Economics and Management, North China Electric Power University, Beijing 102206, China*

[3]*Mathematics and Physics Department, North China Electric Power University, Beijing 102206, China*

Abstract: Short time load forecasting is essential for daily planning and operation of electric power system. It is the important basis for economic dispatching, scheduling and safe operation. Neural network, which has strong nonlinear fitting capability, is widely used in the load forecasting and obtains good prediction effect in nonlinear chaotic time series forecasting. However, the neural network is easy to fall in local optimum, unable to find the global optimal solution. This paper will integrate the traditional optimization algorithm and propose the hybrid intelligent optimization algorithm based on particle swarm optimization algorithm and ant colony optimization algorithm (ACO-PSO) to improve the generalization of the neural network. In the empirical analysis, we select electricity consumption in a certain area for validation. Compared with the traditional BP neutral network and statistical methods, the experimental results demonstrate that the performance of the improved model with more precise results and stronger generalization ability is much better than the traditional methods.

Keywords: Neural network, particle swarm optimization algorithm, ant colony optimization algorithm, hybrid optimization.

1. INTRODUCTION

Power load forecasting is an important part of the rapid development of electric power and relate to the long-term planning of power industry. At the same time, it is also an important part of power system planning and scheduling and is an important foundation for the economic operation of electric power system. According to the time classification, forecasting load can be classified as long-term, medium-term, short-term and ultra-short term load forecasting [1].

At present, there are many kinds of methods to study power load forecasting. Each method has its advantages and disadvantages and there hasn't occurred a universal method for various situations yet. Considering different economic background and influence factors, we can choose different forecasting methods [2]. According to advantages and shortcomings of each kind of electric power load forecasting method, the selection of reasonable forecasting model can improve the accuracy of power load forecasting and provide better decision-making basis for power grid.

Forecasting for the future development of less than 1 year is called short-term prediction. Many scholars and experts have carried out a theoretical study and practical simulation of a large number of short term load forecasting. The methods used in the prediction include regression models, time series model, and later to the intelligent models, such as neural network [3-5], support vector machine [6] and so on. Neural network has strong nonlinear fitting capability, it can be mapped to arbitrary complex nonlinear relationship through training samples, and can also intelligently adapts to arbitrary nonlinear variation in the short term. Its learning rule is simple, and easy to operate and implement. Therefore, this model can effectively meet the demand forecasting including arbitrary power economic indicators affected by multiple factors in the short term.

However, the neural network is easy to fall into local optimum and appears the disadvantages of overfitting, low efficiency of calculation and poor generalization, which is difficult to guarantee the prediction accuracy of models [7-8]. The methods commonly used for the parameter optimization of the neural network include: particle swarm algorithm, ant colony algorithm, genetic algorithm and so on. As the most commonly used swarm intelligence optimization algorithms, ant colony optimization and particle swarm optimization have greater optimization features. Ant colony optimization, which is the simulation of ant colony foraging process, has been successfully applied to many discrete optimization problems. Particle swarm optimization, which is the simulation of birds foraging process, is an efficient parallel search algorithm in continuous optimization field. Ant colony optimization uses pheromone to transmit information, while particle swarm optimization uses three information of the information of its own, individual extreme information, and global extreme information to guide the particle to the next iteration [9]. Using the organic combination of the positive feedback principle and some heuristic algorithms, ant colony

1874-1290/14 2014 Bentham Open

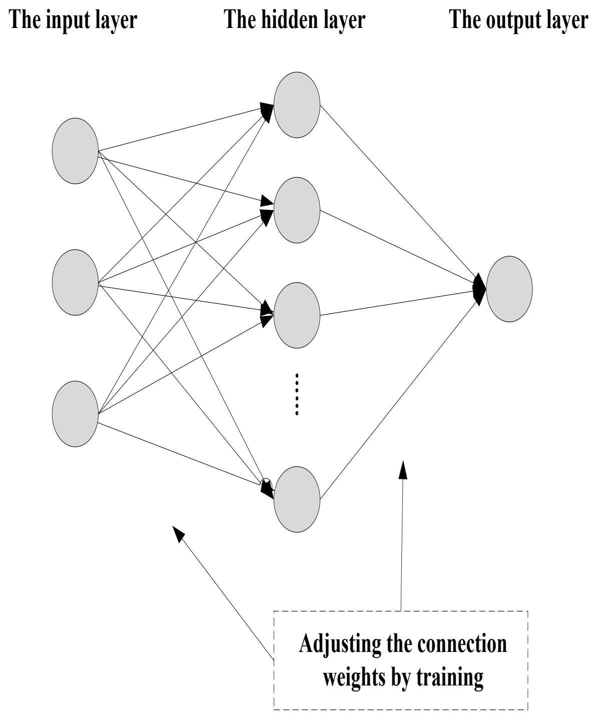

Fig. (1). Neural network structure diagram.

optimization is easy to run into prematurity and fall into local optimum. The organic combination of those two algorithms can overcome the shortcomings of them effectively and improve the computational efficiency significantly [10].

Given this, according to complementary characteristics of each method, this paper presents a hybrid optimization algorithm to optimize the parameters of neural network. The paper is organized as follows: The second chapter introduces the basic theory of neural network and optimization algorithm; the third chapter carries on the short-term load forecasting of a certain area using the proposed new method., at the same time, using the traditional BP algorithm and multiple linear regression method to predict the same target for the verification of accuracy and generalization ability the proposed model; the fourth chapter carries on the summary, and puts forward the further research work.

2. BASIC THEORIES

2.1. Neural network prediction model and BP learning algorithm

Neural network has strong nonlinear fitting capability, it can be mapped to arbitrary complex nonlinear relationship through training samples, and can also intelligently adapts to arbitrary nonlinear variation in the short term. Its learning rule is simple, and easy to operate and implement. Therefore, this model can effectively meet the demand forecasting including arbitrary power economic indicators affected by multiple factors in the short term.

A simple neural network is usually divided into three layers, including the input layer, hidden layer and output layer. Suppose the number of input layer neurons is N, number of hidden layer neurons is L, twice the number of input layer neurons according to the experience, that is $L > 2N$; the number of output layer neurons is M. The topological structure of BP neural network is shown in Fig. (**1**).

We put the BP neural network training samples in the input layer and output layer, then get the mapping from input to output. The mapping relationships don't need a specific calculation formula, which is adjusted by back propagation error signal neuron threshold and link weights, making the error minimum [11].

The error back propagation algorithm of the neural network is to define an error E_r (typically a certain norm between the output and the expected results), and then calculate the weight vector meeting the minimal error. If the error is seen as a continuous function (functional), it will meet that the partial derivative of each component of the weight vector is zero, but in fact, it is discrete, we need to use iteration to find the minimum gradient.

In practice, given a maximum number of iterations n and an error limit E_l, using the error back propagation algorithm, every step of weight correction can make E_r reducing, namely component of the weight vector go along the direction of the gradient decreases in advance. Although, when sample is large enough and n tends to infinite, the error will be convergence theoretically. Actually when the number of iterations reach n, the error E_r may be still greater than the error limit E_l. That means the weight vector does not meet the requirements, so network training fails. Of course, we also can use gradient limit as the terminating condition. In this case, non-convergence means gradient is no less than a certain value in iteration after n times, and thus the weight vector does not meet the requirements. Therefore, convergence is the weight vector of the meet gradient limit.

Therefore, after setting the convergence conditions in the BP neural network, such as the relative error of no more than 3% stipulated by the power system, it is likely to be non-convergence in the training, namely when the neural network training and learning samples at the same time, it is difficult to guarantee the global optimum, and it is easy to fall into local optimum loop, which may affects the precision of prediction. In order to continue to obtain the predict results of given data, we need to use other prediction methods. The solution given in this paper is to add judgment conditions in the neural network model, like training 100000 times. When the training sample data is not convergence, the data automatically fill to polynomial regression model, terminating the neural network prediction and jumping out of local minimum dead circulation. The polynomial regression model is established and prediction results are obtained. The combination method is trained and learned automatically intelligently. It has high accuracy and generalization ability, and can be general in the short-term economic indicators forecasting work.

2.2. Particle Swarm Optimization Algorithm

Particle swarm optimization (PSO) is a kind of evolutionary algorithm, derived from the observation of the birds' behavior of searching for food. The conversion process of the motion of whole flock from disorderly to orderly comes from information shared by each individual birds in the flock, so as to find food [9].

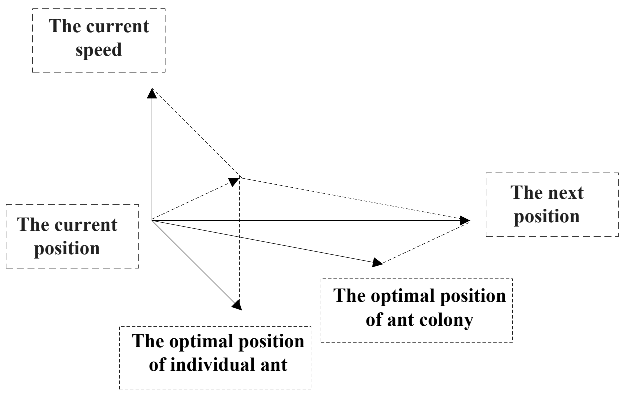

Fig. (2). Movement of particles.

The method used by PSO to solve optimization problems is to initialize a group of random particles and find the optimal solution through several iterations. In the process of iterations, each particle updates its direction and position constantly according to two extreme values. The first one is the optimal solution found by the particle itself, called individual extreme $pbest$, and the other one is the current optimal solution found by the entire particle swarm, named global extreme $gbest$. At the beginning of the iterations, the position of each initialized particle is the individual extreme, while the best position of the particle swarm is the global extreme. After all of the particles in the swarm complete the first iteration, we should compare the position front and rear of each particle, and update the individual extreme with the optimal solution in this iteration if the new position is better than the previous one. Then, we need to get the optimal solution throughout individual extremes of all particles in the swarm as global extreme by comparison, and update the global extreme if the new one is better than the old one. The final global extreme obtained through these cycle iteration operations determines the optimal solution [12].

After obtaining individual extreme and global extreme in the process, each particle needs to update its velocity and position according to the following formulas:

$$\begin{aligned} v_{i,j+1} &= wv_{i,j} + c_1 * random() * (pbest_{i,j} - P_{i,j}) + \\ & c_2 * random() * (gbest_{i,j} - P_{i,j}) \end{aligned} \tag{1}$$

$$P_{i,j+1} = P_{i,j} + v_{i,j+1} \tag{2}$$

Where, $v_{i,j}$ denotes the velocity of the i particle after j iterations, $P_{i,j}$ means the position of the i particle after j iterations, $pbest_{i,j}$ and $gbest_{i,j}$ are on behalf of the individual extreme and global extreme of the i particle after j iterations, w is the inertia weight of the updated speed to the speed of pre update, $random()$ is a random number within $(0,1)$, c_1 and c_2 are learning factors within $(0,2]$. The velocity of particles in the swarm is limited in $(0, v_{\max})$, and the updated value should be replaced with $v_{\max}$ if it exceeds maximum $v_{\max}$ in the process of iterations.

In the process of particle swarm optimization algorithm, particle shares global extreme value to other particles within the group. This one-way flow of shared information and data makes the whole search process follow the group within the current optimal solution. Therefore, the initial particle swarm optimization algorithm has fast global convergence capability.

2.3. Ant colony optimization algorithm

ACO is inspired by Italy scholar Dorigo M from the foraging behavior of real ant colony in nature. He found that an individual ant doesn't have much wisdom or master the nearby geographic information. But the colony can find an optimal path from nest to food sources. Dorigo. M and other researchers proposed ACO theory in 1991, attracting research enthusiasm of many scholars. The basic ACO model consists of the following three equations:

$$\begin{aligned} P_{ij}^{k} &= \tau_{ij}^{\alpha}\eta_{ij}^{\beta} / \sum_{j\in\Lambda} \tau_{ij}^{\alpha}\eta_{ij}^{\beta} \\ \tau_{ij}(n+1) &= \rho\tau_{ij}(n) + \sum_{k=1}^{m} \triangle\tau_{ij}^{k} \end{aligned} \tag{3}$$

$$\tau_{ij}^{k} = \frac{Q}{\sum L_k} \tag{4}$$

Where, m is the number of ants, n is the number of iterations, i is the position of ants, j is the position where ants can reach, Λ is the set of the position where ants can reach, η_{ij} is the heuristic information, which means the visibility of the path from i to j, named $\eta_{ij} = \frac{1}{d_{ij}}$, L_k is the objective function, τ_{ij} is the pheromone intensity of the path from i to j, τ_{ij}^{k} is the number of pheromone left by ants on the path from i to j, α is the weight of the path, β is the weight of the heuristic information, ρ is the evaporation factor of the number of pheromone on the path, Q is coefficient of the pheromone quality, and P_{ij}^{k} denotes the transition probabilities of the NO k ant moving from i to j.

This paper proposes a hybrid algorithm based on particle swarm and ant colony. In the algorithm process, this paper uses the particle swarm optimization algorithm for fast global search to determine the parameters of ant colony and transform the better value into initial information pheromone. Then, using ant colony algorithm for path searching, we put the length of optimal solution, the running time and the numbers of iterations calculated by this set of parameters into the PSO algorithm and update the velocity and position of each particle according to formula until we get the optimal solution for ant colony algorithm [13, 14].

3. EMPIRICAL ANALYSIS

Draw total output value of industry and agriculture and consumption statistics of each year in a period time in a given area. Gross output value of industry and agriculture are in constant prices in a year.

The paper chooses the training and prediction samples to establish the BP neural network, including 4 consecutive years of consumption data and fifth years of gross value of industrial output data as input and fifth years of consumption data as output. The neural network model contains only one hidden layer and the number of neurons in the hidden layer is selected by trial and error method. We first set fewer neurons, and gradually increase the number of neurons, until the network training error reaches the expected range. In order to avoid falling into local minimum for neural network, particle swarm and ant colony optimization are introduced to train the weights and threshold for the global optimum [13, 14].

We sequentially select 4 continuous data as the first 4 input variables from the 37 consumption data. The fifth input variables are the gross value of industrial output data of fifth years. As shown in Table **1**. The input data table is established as shown in Table **2**.

Table 1. Input values of the model.

Category	Input description
Electricity consumption	the annual electricity consumption of 4 years before forecasting year
	the annual electricity consumption of 3 years before forecasting year
	the annual electricity consumption of 2 years before forecasting year
	the annual electricity consumption of 1 years before forecasting year
Gross industrial production	Gross industrial production of the forecasting year

Table 2. Input and output of training data.

Input 1	Input 2	Input 3	Input 4	Input 5	Output
1.5711	2.0016	2.8261	3.5058	84.35	4.1166
2.0016	2.8261	3.5058	4.1166	86.51	4.8186
2.8261	3.5058	4.1166	4.8186	94.12	4.5670
3.5058	4.1166	4.8186	4.5670	95.96	5.7369
4.1166	4.8186	4.5670	5.7369	98.87	6.1161
4.8186	4.5670	5.7369	6.1161	119.26	9.2105
4.5670	5.7369	6.1161	9.2105	135.17	13.8258
5.7369	6.1161	9.2105	13.8258	137.16	19.5787
6.1161	9.2105	13.8258	19.5787	105.52	17.0507
9.2105	13.8258	19.5787	17.0507	97.66	16.9980
13.8258	19.5787	17.0507	16.9980	110.07	17.7732
19.5787	17.0507	16.9980	17.7732	133.74	20.7335
17.0507	16.9980	17.7732	20.7335	156.51	26.2680
16.9980	17.7732	20.7335	26.2680	186.38	33.1576
17.7732	20.7335	26.2680	33.1576	163.60	29.4300
20.7335	26.2680	33.1576	29.4300	168.79	28.0715
26.2680	33.1576	29.4300	28.0715	186.39	34.0092
33.1576	29.4300	28.0715	34.0092	228.89	45.8980
29.4300	28.0715	34.0092	45.8980	270.49	59.0935
28.0715	34.0092	45.8980	59.0935	289.92	68.1366
34.0092	45.8980	59.0935	68.1366	322.31	78.1489
45.8980	59.0935	68.1366	78.1489	321.21	72.6880
59.0935	68.1366	78.1489	72.6880	353.04	84.7979
68.1366	78.1489	72.6880	84.7979	381.45	96.0597
78.1489	72.6880	84.7979	96.0597	419.15	107.3389
72.6880	84.7979	96.0597	107.3389	485.90	121.8486
84.7979	96.0597	107.3389	121.8486	549.30	139.9338
96.0597	107.3389	121.8486	139.9338	622.72	163.5153
107.3389	121.8486	139.9338	163.5153	673.80	173.0294
121.8486	139.9338	163.5153	173.0294	736.93	184.4195
139.9338	163.5153	173.0294	184.4195	822.81	198.8965
163.5153	173.0294	184.4195	198.8965	979.81	215.5823
173.0294	184.4195	198.8965	215.5823	1240	297

Before the neural network training, the input and output data should be normalized with the formula below.

$$\bar{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \qquad (5)$$

After normalization, the value of each variable is between $\left[0,1\right]$, which helps to eliminate the effects of dimensionless.

After the completion of neural network training, we still use the input value of training data for simulation and get the forecasting data. The historical data and the change trend of the forecasting value are shown in Fig. (**2**). Overall, the neural network after optimization has good fitting degree.

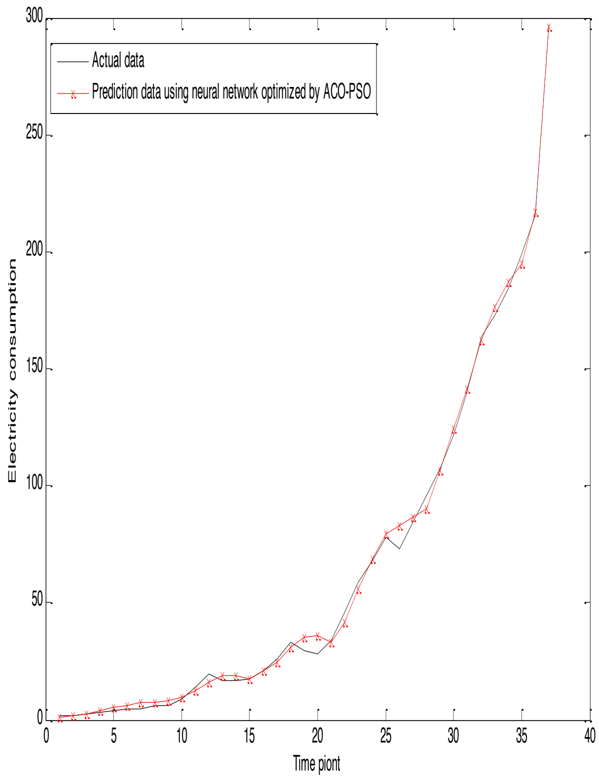

Fig. (3). Prediction value of neural network based on ACO-PSO optimization

This paper uses the following prediction model as the comparison method, including traditional neural network model, unary linear regression model, exponential function prediction model and the inverted index prediction model. The predicted results are listed as Fig. (**3**) and Fig. (**4**).

Unary linear regression equation is $\mathrm{y} = -12.88 + 0.26x$

Exponential function model equation is $\mathrm{y} = 7.93\mathrm{e}^{0.0043\mathrm{x}}$

Inverted index model equation is $\mathrm{y} = 217.99\mathrm{e}^{\frac{-311.48}{\mathrm{x}}}$

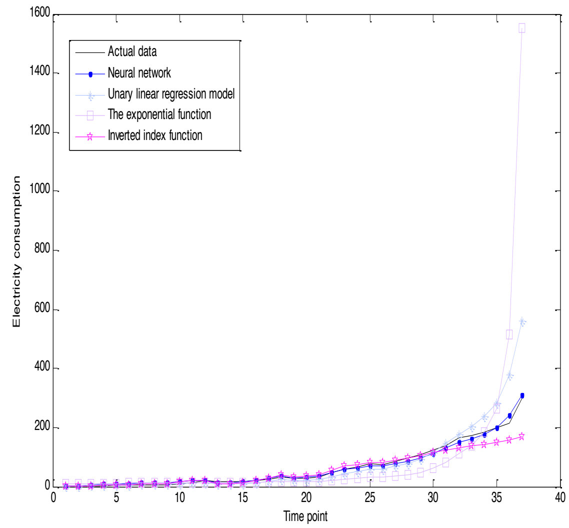

Fig. (4). Different prediction values of comparison methods.

This paper choose Mean Absolute Error (MAE) to quantitative evaluation predict results [15].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \qquad (6)$$

The mean absolute error of each model is as shown in Table **3**.

Table 3. Mean absolute error of each model

Forecasting model	MAE
Neural network based on ACO and PSO	2.50
Traditional neural network	5.73
Unary linear regression model	12.87
Exponential function model	43.65
Inverted index model	13.38

In the error evaluations, the forecasting results of proposed model are the best fit. The error value of this model is 2.5 in MAE evaluation while it separately ups to 5.73, 12.87, 43.65 and 13.38 of the four methods for comparison. As a comparison, the neural network based on ACO and PSO forecasting model has obvious advantages. It has more accurate prediction to get better result and the model has high accuracy and generalization ability, which can be used to power load forecasting.

CONCLUSION

This paper presents an intelligent model of neural network based on the hybrid optimization of ACO and PSO. It is applied to short-term load forecasting and has extremely important practical significance for the power sector planning and forecasting. Considering the characteristics of local minimum of neural network, the intelligent model introduces PSO-ACO hybrid optimization algorithm. The optimization algorithm can overcome the shortcomings of local optimum of particle swarm algorithm and lack of pheromone of ant colony algorithm, improves the generalization ability of the model and promote the actual application scope of the prediction model. This paper selects short-term electricity in a certain area for validation, and the empirical analysis results show that the improved model has improved accuracy and generalization ability.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

This work was supported by Natural Science Foundation of China (Project no.71471059) and the "Industry-Academics-Research" Scientific Project of "Industry-Academics-Research" Co-culture Post-Graduate Base in Beijing.

REFERENCE

[1] D. X. Niu, S. H. Cao, and L. Zhao, "The technology of power load forecasting and its application," *Chinese electric power press*, Beijing, 2006.

[2] S. H. You, H. Z. Cheng, and H. Xie, "Mid- and long-term load forecast based on fuzzy linear regression model", *Electric Power Automation Equipment*, pp.1493-1498, 2008.

[3] P. Li, N.Li, and M. M. Cao, "Micro-meteorology features extraction and status assessment for transmission line icing based on intelligent algorithms", *Journal of Information & Computational Science*, vol. 7, no.10, pp. 2043-2052, 2010.

[4] J. Karásek, R. Burget, and M. K. Dutta, "Java evolutionary framework based on genetic programming", *Signal Processing and Integrated Networks (SPIN)*, *2014 International Conference on. IEEE*, pp. 606-612, 2014.

[5] W. K. Tang, M. H. Wong, and Y. K. Wong, "Load forecasting by fuzzy neural network in Box-Jenkins models" *Proceedings of the IEEE International Conference on Systems*, pp. 1738-1743,1998.

[6] D. Liu, D. X. Niu, and H. Wang, "Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm", *Renewable Energy*, vol. 62, pp. 592-597, 2014.

[7] Y. M. Sun, and Z. S. Zhang, "A new model of STLF based on the fusion of PSRT and chaotic neural networks", *Proceedings of the Csee*, vol. 25, no. 8, pp. 18-23, 2005.

[8] Q. Zhou, H. J. Ren, and J. Li, "Variable weight combination method for mid-long term power load forecasting based on hierarchical structure", *Proceedings of the CSEE*, vol. 30, no. 16, pp. 47-52, 2011.

[9] D. X. Niu, L. Zhao, B. Zhang, and H. F. Wang, "Application of particle swarm grey model in power load forecasting", *Chinese Journal of Management Science*, vol. 15, no. 1, pp. 69-73, 2007.

[10] Q. Wen, X. H Zhang, and X. Yang, "The application of combination forecasting model in medium-long term load forecasting based on load error and economic development trend", *Power System Protection and Control*, vol. 39, no. 3, pp. 57-61, 2011.

[11] B. H. M. Sadeghi, "A BP-neural network predictor model for plastic injection molding process", *Journal of Materials Processing Technology*. vol. 103, no. 3, pp. 411-416, 2000.

[12] D. X. Niu, J. C. Li, and J. Y. Li, "Middle-long power load forecasting based on particle swarm optimization", *Computers and Mathematics with Applications*, vol. 57,no. 11,pp. 1883-1889, 2009.

[13] W. C. Zhang, "Application of ant colony and particle swarm hybrid optimization algorithm", Tianjin University, 2007.

[14] X. M. Xu, D. X. Niu, M. Meng, and H. F. Shi, "Yearly electricity forecasting using a nonhomogeneous exponential model optimized by PSO algorithm", *Applied Mathematics & Information Sciences*, vol. 8, no. 3, pp. 1063-1069, 2014.

[15] J. J. Wang, L. Li, and D. X. Niu, "An annual load forecasting model based on support vector regression with differential evolution algorithm", *Applied Energy*, vol. 42, pp. 468-475, 2012

Received: February 18, 2014 Revised: August 22, 2014 Accepted: December 29, 2014