International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization on TPE (IOTPE)

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

September 2010 | Issue 4 | Volume 2 | Number 3 | Pages 12-22

A HYBRID PARTICLE SWARM OPTIMIZATION BACK PROPAGATION ALGORITHM FOR SHORT TERM LOAD FORECASTING

H. Shayeghi [1] H.A. Shayanfar [2] G. Azimi [2]

1 Technical Engineering Department, University of Mohaghegh Ardabili, Ardabil, Iran, hshayeghi@gmail.com
2 Center of Excellence for Power System Automation and Operation, Electrical Engineering Department, Iran University of Science and Technology, Tehran, Iran, hashayanfar@yahoo.com

Abstract- As accurate Short Term Load Forecasting (STLF) is very important for improvement of the management performance of the electric industry, various short term loads forecasting methods have been developed. This paper addresses an issue of the optimal design of a neural network based short term load forecaster. A new hybrid evolutionary algorithm combining the Particle Swarm Optimization (PSO) algorithm and Back Propagation (BP) algorithm, referred to as HPSOBP algorithm, is proposed to evolve the optimum large neural network structure, connecting weights and bias values for one-day ahead electric load forecasting problem. The hybrid algorithm can make use of not only strong global searching ability of the PSO algorithm, but also strong local searching ability of the BP algorithm. In addition, the input layer of the proposed ANN model receives all relevant information that contributes extensively to the prediction process. The proposed method is applied to STLF of the local utility. Data are clustered due to the differences in their characteristics. Special days are extracted from the normal training sets and handled, separately. In this way, a solution is provided for all load types, including working days, weekends and special days. The experimental results confirm that the proposed method optimized by HPSOBP can quicken the learning speed of the network and improve the forecasting precision compared to the BP and PSO methods and gives lower percent errors all the time. Thus, the proposed method is practical and effective for STLF problem and can be applied to automatically design an optimal load forecaster based on historical data.

Keywords: ANN, Short Term Load Forecasting, Hybrid Particle Swarm Optimization Back Propagation Algorithm.

I. INTRODUCTION

Short term Load Forecasting (STLF) is aimed at predicting electric loads for a period of minutes, hours, days or weeks. The STLF plays an important role for the economic and secure operation of power systems. The accuracy of the operation system, which is derived from the accuracy of the forecasting approach used, will determine the economics of the operation of the power system. Therefore, many forecasting models have been proposed and implemented in this field [1-2].

For the past several years, Artificial Neural Networks (ANNs) methods have received a great deal of attention and were proposed as powerful computational tools to solve the load forecasting problem. ANNs are able to give better performance in dealing with the nonlinear relationships among their input variables [1-3]. ANNs could extract implicit nonlinear relationships among input variables by learning from training data. Using trained supervised networks requires a measure of the discrepancy between the network output value and the desired value [4].

With the development of artificial intelligence in recent years, some approaches have been presented to load forecasting using ANNs with a back propagation (BP) algorithm [1], Genetic Algorithm (GA) [1, 6-8] and Particle Swarm Optimization (PSO) [4, 8-10] methods. Back propagation is a gradient-based method. Although the BP algorithm has solved a number of practical problems, but firstly it easily gets trapped in local minima especially for complex function approximation problem, so that back propagation may lead to failure in finding a global optimal solution. Second, the convergent speed of the BP algorithm is too slow even if the learning goal, a given termination error, can be achieved. The important problem to be stressed is that the convergent behavior of the BP algorithm depends very much on the choices of initial values of the network connection weights as well as the parameters in the algorithm such as the learning rate and momentum. To improve the performance of the original BP algorithm, researchers have concentrated on the following two factors: (1) selection of better energy function; (2) selection of dynamic learning rate and momentum. However, these improvements have not removed the disadvantages of the BP algorithm getting trapped into local optima in essence. In particular, with feed-forward neural networks structure becoming more complex; its convergent speed will be even slower [10].

Genetic algorithm seems to be good methods to solve optimization problems, when applied to problems consisting of more number of local optima, the solution from GA are just near global optimum areas. Also, it takes long simulation time to obtain the solution. Moreover, when the number of parameter is more, optimization problem is complex and coding chromosomes with more genes for increasing algorithm accuracy is caused the GA convergent speed will become very slow, so that convergent accuracy may be influenced by the slow convergent speed [11].

It should be noted that optimization of neural network architecture design, including selecting the number of input variables, input nodes and the number of hidden neurons, to improve forecasting performance is becoming more and more important and desirable. Recently, a new evolutionary computation technique, the Particle Swarm Optimization (PSO) is proposed in the learning and structure improvement of the neural network for the short term electric load forecasting problem [9-10]. PSO technique finds the optimal solution using a population of particles. Each particle represents a candidate solution to the problem. Some of the attractive features of the PSO include ease of implementation and the fact that no gradient information is required [13].

The PSO algorithm has a strong ability to find the most optimistic result, but it has a disadvantage of easily getting into a local optimum [14-15]. The BP algorithm, on the contrary, has a strong ability to find local optimistic result. In this paper, to create a superior forecasting method, the combined the particle swarm optimization and BP algorithms is proposed to exploit the advantages of the two methods and, furthermore, to eliminate the known drawbacks of the ANN trained by the BP and PSO methods. This hybrid uses the PSO algorithm to do global search in the beginning of stage, and then uses the BP algorithm to do local search around the global optimum. In particular, this hybrid algorithm (HPSOBP) will be used to train the ANN weights for short term load forecasting model. Since optimization of neural network architecture design, including selecting the number of input variables, input nodes and the number of hidden neurons, to improve forecasting performance is becoming more and more important and desirable. For this reason, HPSOBP algorithm is employed to obtain the optimum large neural networks structure for one-day ahead electric load forecasting problem.

For the solution of the STLF problem, a large artificial neural network intelligence approach based on day-type cluster is chosen in this study. As the three-layer perceptron is the most common architecture. Thus, the ANN architecture for STLF is feed-forward three-layer perceptron (an input layer, a hidden layer and an output layer). Neural network forecasts are sufficiently good for weekdays and weekends; but, they have to be revised and modified for holidays. Thus, a new approach based on the shape of the daily load curves and correlation analysis on the available data is proposed for such cases. In general, the load curves of special days are dissimilar to those of normal weekdays. Moreover, they are also dissimilar each year. This paper presents a new short term load forecasting method for special days in anomalous load conditions, such as national holidays and religious holidays. In this study, the entire load patterns of special days are classified into number of holidays. Then, a separate ANN model is used for each holiday. Unlike traditional neural network short term load forecasting modeling approach, the interrelationship among the input variables and outputs of the neural network is considered. The input variables of the proposed neural network based model are historical load data and temperature. The outputs produced by this model are peak and hourly load values. The simulation results show that the proposed HPSO-BP based method provides a greater degree of accuracy in many cases for STLF problem, compared to the PSO and BP methods. Moreover, it gives lower Mean Absolute Percentage Error (MAPE). Thus, it can be applied to automatically design an optimal load forecaster based on historical data.

II. DATA ANALYSIS AND PREPROCESSING

The available data for this research are total hourly actual loads of a local utility in Iran for the years 2000 to 2003. In order to use these data in a meaningful and logical manner, first of all they should be closely analyzed and their dynamics should be clearly understood. Then, they can be clustered into smaller sets according to some common characteristics and separate models can be built for each cluster. This is necessary because it has always been emphasized in the literature that it is impossible to reflect every different type of load behavior with a single model. The load profile is a dynamic process. Temporal variations, abrupt increases in demand, outages or other random disturbances all affect the load level. It is noteworthy that load shapes for the same weekday for different weeks are quite similar; and that load shapes for different weekdays for the same week are roughly similar. Figure 1 shows hourly load curves for a sample week. This graph gives an idea about how the electric load varies from hour to hour and day to day. It is seen that four working days (Sunday-Wednesday) have very similar patterns and Saturday (first working day of the week in Iranian calendar) is slightly different from the other workings days. Also, Weekend days, i.e. Thursday and Friday are different from the other days [1].

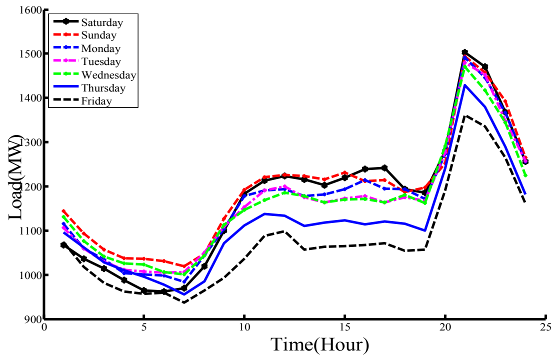


Figure 1. Daily load curves for a typical week, Saturday, Sunday, to Wednesday, Thursday and Friday

A. Correlation Analysis

If the training set of a neural network contains patterns that have characteristics close to each other and the output carries the same kind of information as the inputs then this model gives successful result. In order to evaluate the validity of this hypothesis, a measure of the resemblance between daily load sequences is thought to be established. For this reason, the correlation function is taken into consideration. Cross correlation coefficients are computed for each data pair as follows:

$$C_{xy} = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}} \tag{1}$$

where $S_{xy} = \sum_{i=1}^{n} |x_i - \bar{x}||y_i - \bar{y}|$, $S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2$, $S_{yy} = \sum_{i=1}^{n} (y_i - \bar{y})^2$ and x and y represent the data pairs, $\bar{x}$ and $\bar{y}$ are the mean values calculated over the samples and n is number of samples.

Table 1 summarizes the correlations of the daily electric load consumptions in year 2003. Column ($D+i$) represents the ith day after the day D, given row-wise. It can be seen that, weekdays highly correlated with each other; but Thursday and Friday have lower correlations with each other and with weekdays. Saturday is the day which has the lowest correlations with the other weekdays.

Table 1. Daily load correlations in year 2003

Day	D+1	D+2	D+3	D+4	D+5	D+6	D+7	D+14
Saturday	0.9879	0.9904	0.9821	0.9750	0.9636	0.9411	0.9905	0.9908
Sunday	0.9937	0.9895	0.9858	0.9802	0.9633	0.9879	0.9937	-
Monday	0.9953	0.9930	0.9845	0.9612	0.9904	0.9937	0.9954	-
Tuesday	0.9950	0.9876	0.9620	0.9821	0.9895	0.9953	0.9902	-
Wednesday	0.9852	0.9652	0.9750	0.9858	0.9930	0.9950	0.9900	-
Thursday	0.9793	0.9636	0.9802	0.9845	0.9876	0.9852	0.9893	0.9872
Friday	0.9411	0.9633	0.9612	0.9620	0.9652	0.9793	0.9773	0.9575

B. Data Clustering

Based on the shape of the daily load curves and correlation analysis on the available data, an efficient clustering can be done. First of all, religious and national holidays should be excluded from the regular day data and handled separately, since their characteristics are completely different. Thus, four weekdays (Sunday-Wednesday) can be examined in the same group. It does not seem necessary to create a distinct group for each of these weekdays as they are highly correlated. Moreover, a separate group should be formed for the first working day (Saturday), because they come just after the weekend and do not resemble the other weekdays. For weekends, two groups should be formed as Thursdays and Fridays, since they have unique characteristics [1].

III. ARCHITECTURE OF NEURAL NETWORK BASED STLF

In a supervised leaning ANN, a feed-forward multi-layered perceptron neural networks is widely used and many enhancements have been explored. The partitioning method is one of the enhancements. It was developed because of differences in the load shape for every season (see Figure 2) and every day (vide Figure 1). The partitioning method divides the network into several sub-networks. In this study, the network is divided into the following groups: Sunday through Wednesday, Thursday, Friday and Saturday. Moreover, for partitioning a year, is divided into four seasons (spring, summer, fall and winter), and every season is divided into three different kinds of day (Saturday, weekday and weekend). Figure 2 depicts the hourly loads for each season. The highest load would occur in summer. The time of peek load in each season is difference [1].

It should be noted that would be better to distinguish between the seasons by using different ANN modules. Accordingly, the training would be easier and there is a chance to have better results. Thus, four ANN modules for summer, winter, spring and fall is used in STLF problem. In the training process, every network is only supplied by data on that particular season.

Four seasonal networks have the same architecture, which is a three-layer feed-forward neural network. For every season, the number of neurons in the input or the output layer is already fixed, based on the input and output data chosen. But the number of neurons in the hidden layer is different and here, is obtained using the HPSO-BP based method.

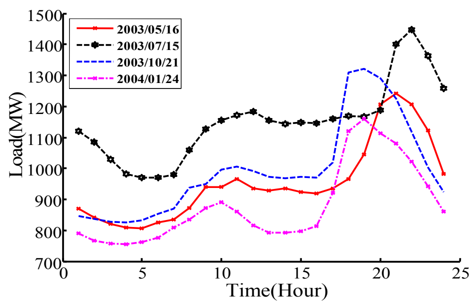


Figure 2. Daily load curves for a typical day for each season

A. Proposed ANN Architecture

There are several nontrivial tasks associated with the design of a neural network based load forecaster. One such task is the selection of input variables and the network structure that would secure an acceptable forecast accuracy and network training time. For example, a network with too many hidden neurons will memorize the training data instead of learning general relationships and will perform poorly while applied to new data. Training data extraction and design of an efficient and reliable learning algorithm are also of great importance. At present, there is no systematic methodology for optimal design and training of an artificial neural network. One has often to resort to the trial and error approach. This paper summarizes the optimal design of a neural network based load forecaster. A systematic approach to solving these problems is proposed. It can be applied to the automatic design of an optimal forecaster based on the available historical data.

The process of developing an artificial neural network based on load forecasting can be divided into 5 steps [1]:
1. Selection of input variables.
2. Design of neural network structure.
3. Extraction of training, test and validation data.
4. Training of the designed neural networks.
5. Validation of the trained neural networks.

B. Selection of Input Variables

Neural network input variables are selected from load affecting factors. One of the keys for designing a good architecture in ANN is choosing appropriate input variables from load affecting factors. Those factors may vary from one utility to another based on the load characteristics. However, there are several factors that are commonly used. In the case of short term load forecasting problem, these inputs can be divided into time, electrical load and weather information. The time information may include the type of season, days of a week, and hours of a day. The load information may include previous loads. The weather information may include previous and future temperatures, cloud cover, thunderstorm, humidity, and rain. As shown in Figure 1, load changes during the day from one hour to another and from one day to another during the week. On the other hand, the load at a given hour is dependent not only on the load at the previous hour but also on the load at the same hour on the previous day and on the load at the same hour on the day with the same denomination in the previous week.

Until now, there have been no general regulations on input types in designing the ANN for STLF problem. However, as a matter of principle, historical load and temperature represent the most important inputs. For a normal climate area, these two inputs and other related inputs (e.g., time) would be sufficient to make a good short-time load forecasting model. However, for extreme weather conditions in humid areas or in areas with many thunderstorms, additional weather factors should be included for forecasting.

In the proposed architecture, ANN is designed based on previous loads, type of season, type of day, hours of a day, previous day's temperature and temperature forecast. Only two weather factors are used in this architecture, since the area of the forecasted load is a normal climate area.

A block diagram for the proposed ANN architecture is shown in Figure 3. The numbers of neurons in the input and output layers are determined by the number of input and output variables respectively. The nodes in the input layer are used for the distribution of the inputs to the hidden layer neurons and are not actual neurons.

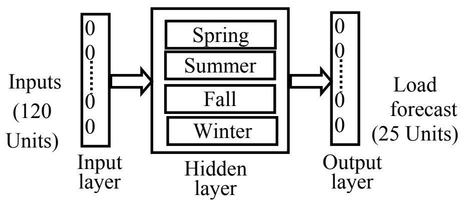


Figure 3. Block diagram of the proposed ANN Architecture

There are a total of 120 neurons in the input layer. The details of ANN input variables for groups Sunday through Wednesday, Thursday, Friday and Saturday are given as follows, and the variables are selected according to the discussions as mentioned in Sec. 2 and by trial and error.

A. The ANN input variables for group Sunday through Wednesday:
1. The first 24 input neurons represent hourly scaled load values of the previous day.
2. The next 48 neurons are used to capture the effect of temperature: hourly temperature data from previous day and hourly temperature data for the day of forecast.
3. The next 24 input neurons represent hourly scaled load values of the two previous days.
4. The next 24 input neurons represent hourly scaled load values of the same day in the previous week.

B. The ANN input variables for groups Thursday, Friday and Saturday:
1. The first 24 input neurons represent hourly scaled load values of the same day in the previous week.
2. The next 48 neurons are used to define the effect of temperature: The first 24 neurons represent hourly scaled temperature data of the same day in the previous week and another 24 neurons represent hourly load temperature data of the next day (the day of forecast).
3. The next 48 neurons represent hourly scaled load values of the same day in the two and three previous weeks.

C. Design of Neural Network Structure

To design a multi-layer feed-forward network, one needs to select the number of hidden layers, type of connection between the layers, number of neurons in each layer and a neuron's activation function. In practice, a fully connected network with one hidden layer is a reasonable choice. Thus, three-layered percepteron, that has proved its good performance, is used in this application. According to the discussions as mentioned in Sec. 2, a separate ANN model is designed for each of the four-day classes. Each network has 120 neurons in input layer and its output layer consists of 25 neurons; the first 24 neurons, each represent the predicted hourly load covering 24 hours of day and 25th neuron represent the predicted maximum load of day. To design a three-layered percepteron network, one needs to select the number of neurons in hidden layer and neuron's activation function. Good candidates for an activation function are sigmoid (S-shaped) functions. The exact shape of the sigmoid function has little effect on the network performance. It may have a significant impact on the training speed. In this work, the output and hidden layers have sigmoid activation function in order to eliminate additional errors for extreme forecasts due to the saturation of the activation function [10].

The number of neurons in the hidden layer determines the network's learning capabilities and its selection is the key issue in optimal network structure design. If the number is too small, the network cannot find the complex relationship between input and output and may have difficulty in convergence during training. If the number is too large, the training process would take longer and could harm the capability of ANN. It would vary for different applications and could usually depend on the size of the training set and the number of input variables. The hidden layer size and its neurons number are selected either arbitrarily or based on the trial and error approach. In this study, a hybrid particle swarm optimization-back propagation based method is proposed for find the optimal number of the hidden layer neurons. The HPSOBP algorithm includes two stages. At the first stage, PSO is employed to search for the optimum. At the second stage, BP algorithm is used to search around the global optimum according to this heuristic knowledge. In this way, this hybrid algorithm may find an optimum more quickly.

The flowchart of the proposed method is shown in Figure 4. The simulation results for the optimal number of neurons in the hidden layer based on the Mean Absolute Percentage Error (MAPE) performance index for Sunday-Wednesday class in the winter is shown in Figure 5 for example, and for Saturday, Thursday and Friday classes the optimal number of neurons in the hidden layer is 10, 10 and 8 neurons, respectively.

D. Extraction of Training, Test and Validation Data

Collecting training data is very important to achieve the desired level of ANN performance to STLF problem. It should be noted that for the network updating a few patterns is required if the numbers of training data are much. Moreover, to assure a good network performance, the training data should be representative and it is normalized. A normalization step helps in preventing the simulated neurons from being driven too far into saturation. Through transformation the data of the input and the output of neural network are limited to the interval $[\alpha,\beta]$, $\alpha\neq\beta$, where $0\leq\alpha,\beta\leq1$. In this study the values of α and β are 0.1 and 0.8, respectively. In this way, the training convergence speed can be increased and the overflow of calculation can be avoided. Data normalization can be calculated by the following equations:

$$a_i=(\beta-\alpha)/(\max(x_i)-\min(x_i)) \\ b_i=(\beta-a_i)\times\max(x_i) \quad (2)$$

$$X_{i,j}^N=a_i\times x_{i.j}+b_i \quad (3)$$

Where, $x_{i.j}$ and $X_{i,j}^N$ refer to the actual hourly temperature/load and the normalized value of the ith day at the jth hour respectively; also $X_{i,j}^N$ is the input data of input nodes of neural network; $\max(x_i)$, $\min(x_i)$ refer to the maximum temperature/load and minimum temperature/load of the ith day, respectively. The training output values are also normalized in the same manner.

The data that were used for training, testing and validating of the ANN was total hourly actual loads and weather data of a local utility in Iran for the years 2000 to 2003. All data are divided into four parts based on the type of season. The data for each season are divided again into four parts based on day cluster. The data for each cluster are divided into three parts as training, test and validate sets. Test and validate sets will not be used for training; their purpose is only to examine errors produced by ANN after training [10].

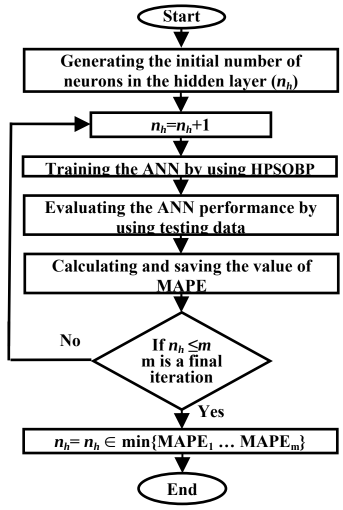


Figure 4. The flowchart of the proposed method for selection of the number of hidden layer neurons

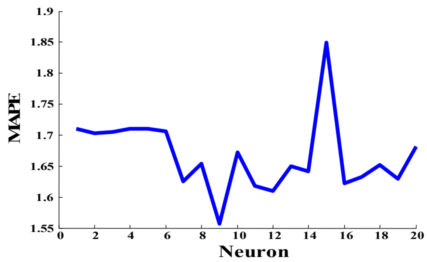


Figure 5. The optimal number of neurons based on the MAPE performance index for Sunday-Wednesday class

E. Training of the Designed Neural Networks

Training of a neural network is a process of determining the network parameters (weights) in order to achieve the desired objective based on the set of examples called the training set. The use of ANN for solution of the STLF problem can be broken down into two groups based on learning strategies: supervised and unsupervised learning. Supervised learning is based on direct comparison between the inputs and outputs. This is usually formulated as the minimization of an error function such as the total mean square error between the actual output and the desired output summed over all available data. The unsupervised learning is solely based on the correlations among input data. No information on "correct output" is available for learning. Most applications use a supervised learning ANN.

Most of the researches focus on layer weights and network topology. Regarding the forecast problem, the learning algorithm is an essential part. There are several methods of training ANN such as back propagation, genetic algorithm, particle swarm optimization, and so on. These algorithms have some drawbacks. The BP algorithm has the slowness of learning speed, possibility of falling into local minimum [4] and the necessity of adjusting a learning constant in every application. When the GA applied to problems consisting of more number of local optima, the solution from GA are just near global optimum areas. Also, it takes long simulation time to obtain the solution. The PSO algorithm has a disadvantage that the search around global optimum is very slow. In this paper, we combined the particle swarm optimization algorithm with the BP algorithm to exploit the advantages of the two methods and, furthermore, to overcome the known drawbacks of the ANN trained by the BP, GA and PSO methods. Combine HPSO-BP with the neural network model to forecast the short term electric load in order to improve the forecast precision and the over-generalized capability of the model.

F. Validation of the Trained Neural Network

Since acceptable training errors do not always guarantee similar network performance for a different set of data, for example due to the lack of representativeness of the training set or the improperly selected network size, it is necessary to validate the network performance after it is trained. This is usually done by randomly selecting 10-20% of the total training data and setting it aside for testing. Based on the above discussions, test and validation data is randomly extracted by selecting 20% and 10% from the entire training data, respectively and the rest of entire data (about 70%) is used for the networks training.

If the testing errors are unacceptable, possible causes should be identified and corrected, and the network should be retrained. The old test set should be included into the training set. If new relevant data is available, a new test set should be collected. Otherwise, a new test set is again randomly extracted from the entire training data.

IV. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a kind of algorithm to search for the best solution by simulating the movement and flocking of birds. The algorithm works by initializing a flock of birds randomly over the searching space, where every bird is called as a "particle". These "particles" fly with a certain velocity and find the global best position after some iteration. At each iteration, each particle can adjust its velocity vector, based on its momentum and the influence of its best position as well as the best position of its neighbors, and then compute a new position that the "particle" is to fly to [14].

The PSO definition is presented as follow [10]:

1) Each individual particle i has the following properties: a current position in search space x_i, a current velocity v_i, and a personal best position in search space p_i.

2) The personal best position p_i corresponds to the position in search space, where particle i presents the smallest error as determined by the objective function f, assuming a minimization task.

3) The global best position denoted by $\tilde{p}$ represents the position yielding the lowest error among all the p_i's.

It should be noted that, when the PSO algorithm is used in evolving weights of feed-forward neural network, every particle represent a set of weights.

Equations (4) and (5) define how the personal and global best values are updated at time, respectively. It is assumed below that the swarm consists of particles.

Thus, $i \in 1 \cdots s$

$$p_i(t+1) = \begin{cases} p_i(t), & \text{if} \quad f(p_i(t)) \le f(x_i(t+1)) \\ x_i(t+1), & \text{if} \quad f(p_i(t)) > f(x_i(t+1)) \end{cases} \quad (4)$$

$\tilde{p}(t) \in \{p_0(t), p_1(t), \cdots, p_s(t)\}$ and

$$\tilde{p}(t) = \min\{f(p_0(t)), f(p_1(t)), \cdots f(p_s(t))\} \quad (5)$$

During each iteration, every particle in the swarm is updated using (6) and (7). Tow pseudorandom sequences $r_1 \sim U(0,1)$ and $r_2 \sim U(0,1)$ are used to affect the stochastic nature of the algorithm. For all dimensions $j \in 1 \cdots n$, let $x_{i,j}$, $p_{i,j}$ and $v_{i,j}$ be the current position, current personal best position, and velocity of the j^{th} dimension of the i^{th} particle. The velocity update step is

$$\begin{aligned} v_{i,j}(t+1) &= \omega v_{i,j}(t) + c_1 r_{1,i}(t)(p_{i,j}(t) - w_{i,j}(t)) \\ &\quad + c_2 r_{2,i}(t)(\tilde{p}_j(t) - w_{i,j}(t)) \end{aligned} \quad (6)$$

The new velocity is then added to the current position of the particle to obtain its next position

$$x_i(t+1) = x_i(t+1) + v_i(t+1) \quad (7)$$

The value of each dimension of every velocity vector v_i is clamped to the range [$-v_{max}$, v_{max}] to reduce the likelihood of the particle leaving the search space. The value of v_{max} is usually chosen to be $v_{max}= kx_{max}$, where $0.1 \le k \le 1$ and x_{max} denotes the domain of the search space. Note that this does not restrict the value x_i of to the range [$-v_{max}$,v_{max}]. Rather than that, it merely limits the maximum distance that a particle will move.

The acceleration coefficients c_1 and c_2 control how far a particle will move in a single iteration. Typically, these are both set to a value of 2, although it has been shown that setting $c_1 \ne c_2$ can lead to a good performance. The inertia weight ω in (6) is used to control the convergence behavior of the PSO, where ω is a new inertial weight. This algorithm by adjusting the parameter ω can make ω reduce gradually as the generation increases. Small values of ω result in more rapid convergence usually on a suboptimal position, while a too large value may prevent divergence. In the searching process of the PSO algorithm, the searching space will reduce step by step nonlinearly as the generation increases, so the searching step length for the parameter ω here also reduces correspondingly. Similar to GA, after each generation, the best particle of particles in last generation will replace the worst particle of particles in current generation, thus better result can be achieved.

Generally, in the beginning stages of algorithm, the inertial weight ω should be reduced rapidly, when around optimum, the inertial weight ω should be reduced slowly. So in this paper, we adopted the following selection strategy [14]:

$$\omega = \begin{cases} \omega_0 - (\omega_1 / \max gen_1) \times generation \\ \qquad\qquad 1 \le generation \le \max gen_1 \\ (\omega_0 - \omega_1) \times e^{(\max gen_1 - generation)/k} \\ \qquad\qquad \max gen_1 \le generation \le \max gen_2 \end{cases} \tag{8}$$

where, ω_0 is the initial inertial weight, ω_1 is the inertial weight of linear section ending, maxgen_2 is the total searching generations, maxgen_l is the used generations that inertial weight is reduced linearly, *generation*, is a variable whose range is [1, maxgen_2]. Through adjusting k, we can achieve different ending values of inertial weight. Figure 6 illustrates the reduction scheme for the inertial weight. In particular, the value of maxgen_2 is selected according to empirical knowledge.

The PSO system combines two models: a social-only model and the cognition-only model. These models are represented by the velocity update, shown in (6). The second term in the velocity update equation is associated with cognition since it only takes into account the particle's own experiences. The third term in the velocity update equation represents the social interaction between the particles. It suggests that individuals ignore their own experience and adjust their behavior according to the successful beliefs of individuals in the neighborhood [10].

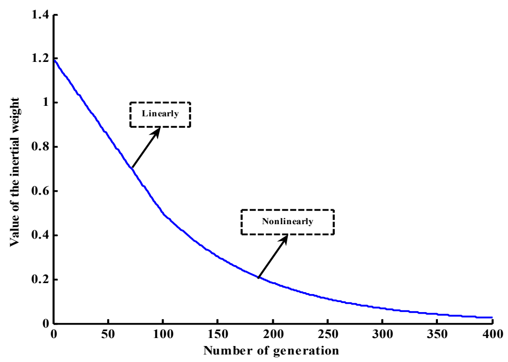


Figure 6. The reduction scheme for value of the inertia weight ω

V. THE EVOLUTION NEURAL NETWORK MODEL BASED ON HPSO-BP OPTIMIZE ALGORITHM

The HPSO-BP is an optimization algorithm combining the PSO with the BP. The PSO algorithm is a global algorithm, which has a strong ability to find global optimistic results, however, it has a disadvantage that the search around global optimum is very slow. The BP algorithm, on the contrary, has a strong ability to find local optimistic result, but its ability to find the global optimistic result is weak. By combining the PSO whit the BP, a new algorithm referred to as PSO-BP hybrid algorithm is formulated in this paper. The fundamental idea for this hybrid algorithm is that at the beginning stage of searching for the optimum, the PSO is employed to accelerate the training speed. When the fitness function value has not changed for some generations, or value changed is smaller than a predefined number, the searching process is switched to gradient descending searching according to this heuristic knowledge.

Similar to the PSO algorithm, the HPSO-BP algorithm's searching process is also started from initializing a group of random particles. First, all the particles are updated according to the (6) and (7), until a new generation set of particles are generated, and then those new particles are used to search the global best position in the solution space. Finally, the BP algorithm is used to search around the global optimum. In this way, this hybrid algorithm may find an optimum more quickly [16].

The procedure for this HPSO-BP algorithm can be summarized as follows:

Step 1: Initialized the position and velocities of a group of particles randomly in the range of [0, 1].

Step 2: Evaluate each initialized particle's fitness value, and p_i is set as the personal best positions of the current particles, while $\tilde{p}$ is set the global best position of the initialized particles.

Step 3: If the maximal iterative generations is arrived, go to Step 8, else, go to Step 4.

Step 4: The best particle of the current particles is stored. The positions and velocities of all the particles are updated according to (6) and (7), then a group of new particles are generated.

Step 5: Evaluate each new particle's fitness value, and the worst particle is replaced by the stored best particle. If the ith particle's new position is better than p_{ib}. p_{ib} is set as the new position of the particle (vide Equation 4). If the global best position of all new particles is better than $\tilde{p}$, then $\tilde{p}$ is updated (vide Equation 5).

Step 6: Reduce the inertia weights ω according to the selection strategy described in Section IV.

Step 7: If the current $\tilde{p}$ is unchanged for ten generations, then go to Step 8; else, go to Step 3.

Step 8: Use the BP algorithm to search around $\tilde{p}$ for some epochs, if the search result is better than $\tilde{p}$, output the current search result; or else, output $\tilde{p}$.

The parameter ω, in the above HPSO-BP algorithm also reduces gradually as the iterative generation increases, just like the PSO algorithm. The selection strategy for the inertial weight ω is the same as the one described in Section IV, i.e., firstly reduce ω linearly then reduce it nonlinearly. But the parameter maxgen_1 generally is adjusted to an appropriate value by many repeated experiments, and then an adaptive gradient descending method is used to search around the global optimum $\tilde{p}$.

In this study, the value of the mean squared error (MSE), shown as (9), serves as the objective function for identifying suitable parameters for use in the HPSO-BP model.P

$$f_{fitness} = \frac{1}{N}\sum_{k=1}^{N}[\frac{1}{O}\sum_{l=1}^{O}(T_{kl} - P_{kl})^2]x_i \tag{9}$$

Where $f_{fitness}$ is the fitness value, T_{kl} is the target output; P_{kl} is the predicted output based on x_i; N is the number of training set samples; and, O is the number of output neurons.

VI. HANDLING THE SPECIAL DAYS

In Iran, there are two kinds of holidays, national and religious. National holidays are fixed in time, but religious holidays are moving each year. Load curve of holidays differ from a typical weekday, also number of these days in historical information in comparison with typical weekdays is less. It is important to forecast the loads of such days as well, in order to have a complete model. It is known that electric consumption decreases on holidays, as shown in Figure 7. If the neural networks, designed for regular load forecasting, are directly used for special day load forecasting, large errors are observed. Thus, they should be analyzed separately.

One exception can be done to single day holidays that coincide to Fridays. They are not so much different than the regular Friday data, therefore, there is no need to form a cluster for this kind of data; instead, they can be put into the Friday training set.

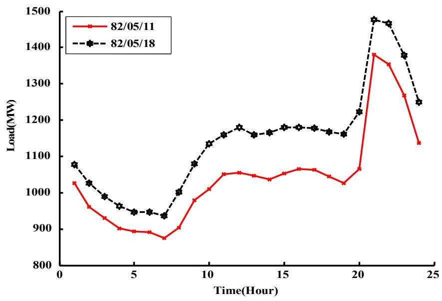


Figure 7. Comparison of daily load curves for a special day (religious holiday-Saturday 2003.08.02) and the same day in the next week (regular day - Saturday 2003.08.09)

In this study, the entire load patterns (from 1998 to 2003 years) of special days are classified into number of holidays. Then, a separate ANN is used for each holiday. A three layer feed-forward neural network is used for each of holidays and by using hybrid particle swarm optimization-back propagation algorithm, the number of hidden layer is fined. ANN architectures are as follow:

- **Input variables:** Selection of input variables based on the shape of the daily load curves and correlation analysis on the available data. As seen from Table 2 and Figure 8 special days have highly correlations and highly similarity daily load curves with last Friday and the same special day in the previous year. But, correlation and similarity daily load curves special day relative the previous day depends to the special day-type. By the above analysis, the input variables are as follow:
- 24 hourly scaled load values and their temperatures data of the previous day (48 units).
- 24 hourly scaled load values of the previous Friday (24 units).
- 24 hourly scaled load values of the same special day in the previous year (24 units).
- 24 hourly scaled temperatures data of the special day in the current year (forecast day) (24 units).
- **Output variables**: 24 hourly scaled load values and maximum scaled load values of the special day in the current year (25 units).

Table 2. Special daily load correlations in year 2003

Special days (Day of forecast)	Previous day	Previous Friday	The same special day in the previous year
Arbein (Religious holiday)	0.9420	0.9926	0.9789
14 Khordad (National holiday)	0.9727	0.9902	0.9916

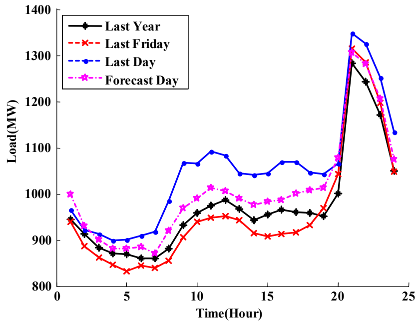


Figure 8. Daily load curves for a special day, previous day previous Friday and the same special day in the previous year

VII. SIMULATION RESULT AND EVALUATION

To evaluate, the performance of the proposed neural networks is tested on real data of a local utility in Iran power system. Four and six years of historical data is used to train the regular days and special day neural network based STLF model, respectively. Actual weather data is used. The evaluation of the forecasting accuracy is accomplished by a MAPE and Absolute Percentage Error (APE), which are given by:

$$MAPE = \frac{1}{N}\sum_{1}^{N}\left|\frac{Actual_i - Forecast_i}{Actual_i}\right| \times 100 \tag{10}$$

$$APE = \left|\frac{Actual_i - Forecast_i}{Actual_i}\right| \times 100 \tag{11}$$

where, N is the total number of hours, $Actual_i$ is the actual load at hour i and $forecast_i$ is the forecast value of the load at that hour. The mean squared error is used as the fitness function that is optimized by the HPSO, PSO and BP algorithms.

To demonstrate the effectiveness of the proposed ANN based model for solution of the STLF problem, some simulations are carried out. ANN is designed for each of four-day classes and special days, we ran the three training algorithms (HPSO, PSO and BP), respectively. Some simulation results for evaluation are shown in Figures 9 and 10 for four-day classes. Figure 9 shows the actual and forecasted daily load. Figure 10 depicts the APE. Also, Table 3 summaries the MAPE for different day classes and Table 4 summaries the APE of maximum daily load for different day classes. The actual and forecasted daily load of two typical of special day (Arbein and 14 Khordad) is shown in Figure 11. Figure 12 depicts the absolute percentage error of two typical of special day. Also Table 5 summaries the mean absolute percentage errors of two aforesaid special days with HPSO, PSO and BP training methods. From the above figures, it can be seen that the HPSO-BP based STLF is very close to the actual load compared with the PSO and BP methods and achieves higher accuracy than the PSO and BP algorithms. Thus, the HPSO approach is more effective and economical, and can effectively improve the forecasting precision, and has very less the forecasted load errors.

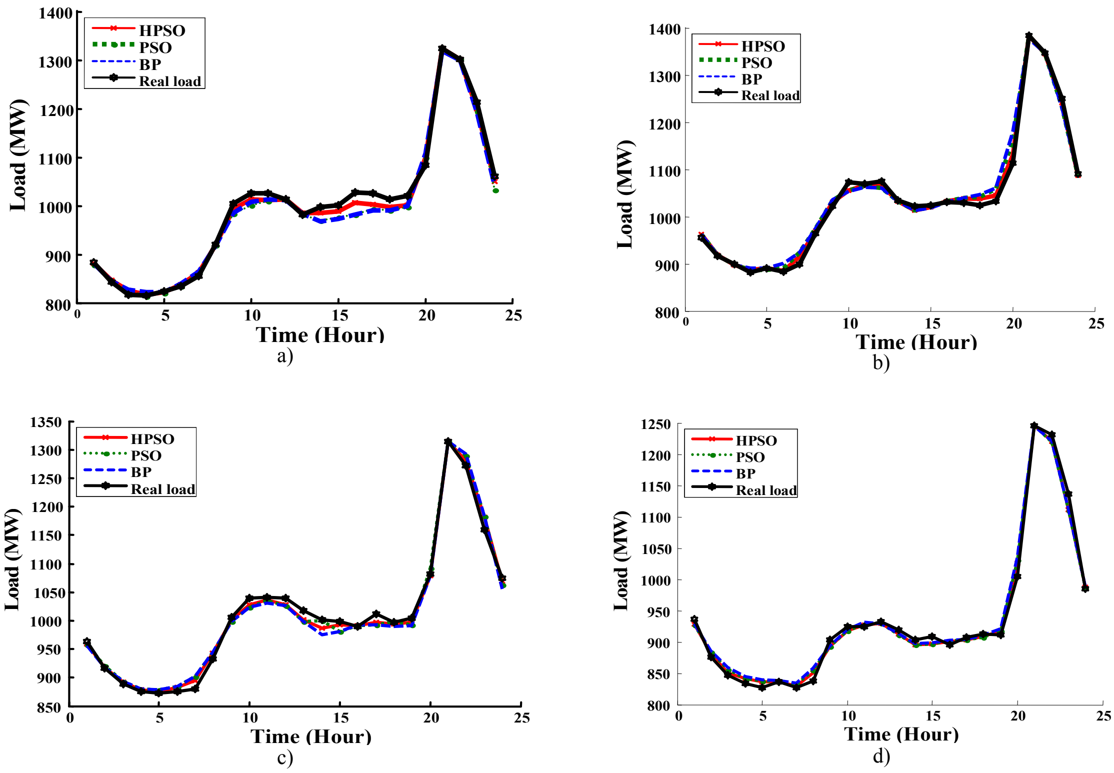


Figure 9. ANN forecasted loads for sample four-day classes: (a) Saturday (b) Remaining working days (c) Thursday (d) Friday

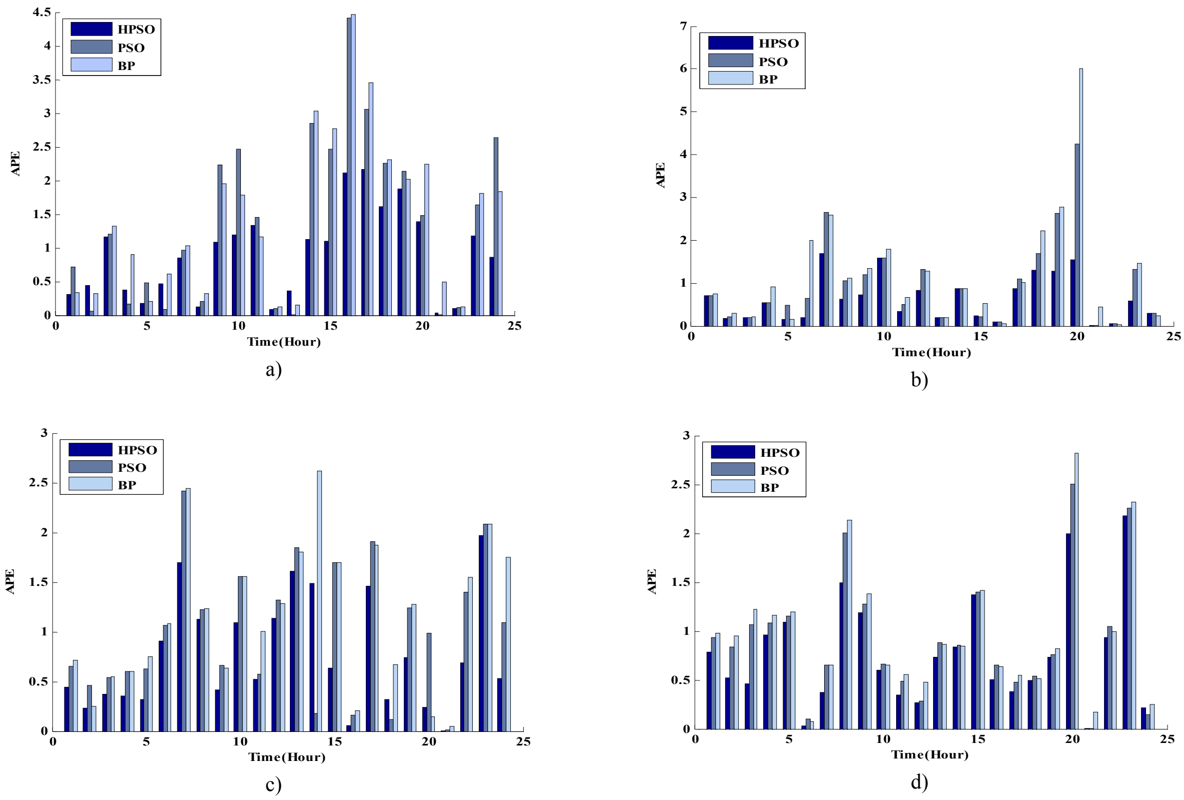


Figure 10. Forecasted absolute percentage errors for sample four-day classes: (a) Saturday (b) Remaining working days (c) Thursday (d) Friday

VIII. CONCLUSIONS

This paper addresses a hybrid HPSO-BP algorithm for training neural network to forecast the short term electric load in order to improve the forecast precision and the over-generalized capability of the model. The proposed algorithm combines the particle swarm optimization algorithm's strong ability in global search and the back-propagation algorithm's strong ability in local search. We can get better search result using this hybrid algorithm.

The propose method is based on clustering data analysis and correlation measures. Clustering is performed after a detailed data analysis, based on correlation measures, daily and seasonal variations, holiday behaviors, etc. Then separate large neural network models are constructed for each cluster. In this paper, HPSO-BP algorithm is employed to find the optimum large neural networks structure, connecting weights and bias values for one-day ahead electric load forecasting problem. This study also presented the research work conducted to improve the short term load forecasting for special days in anomalous load conditions, which was a difficult task using conventional methods.

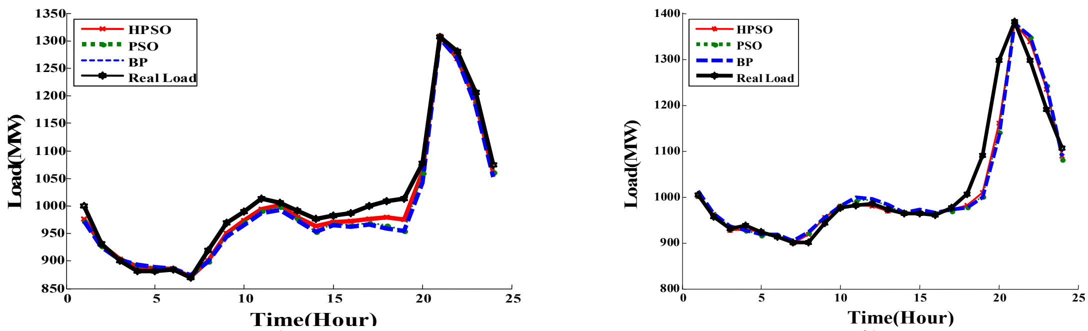
The results show that the proposed ANN-based model not only is effective in reaching proper load forecast but also it can be applied to the automatic design of an optimal forecaster based on the available historical data. Also, it has easy implementation and good performance. The model performance evaluation in terms of 'MAPE' and APE indices reveals that the proposed HPSO-BP based ANN model produces lower prediction error and is superior to the BP and PSO based ANN methods. Thus, the proposed forecasting methods could be provided a considerable improvement of the forecasting accuracy for the regular and the special days and it is recommended as a promising approach for the solution of the STLF problem.

Table 3. MAPE for different day classes

Day-classes		MAPE		
		BP	**PSO**	**HPSO-BP**
Class 1	Saturdays	1.45	1.383	0.896
Class 2 (Remaining working days)	Sundays	1.203	0.991	0.627
	Mondays	1.730	1.535	1.246
	Tuesdays	1.673	1.571	1.024
	Wednesdays	1.124	1.108	0.912
Class 3	Thursdays	1.159	0.950	0.758
Class 4	Fridays	0.985	0.919	0.771

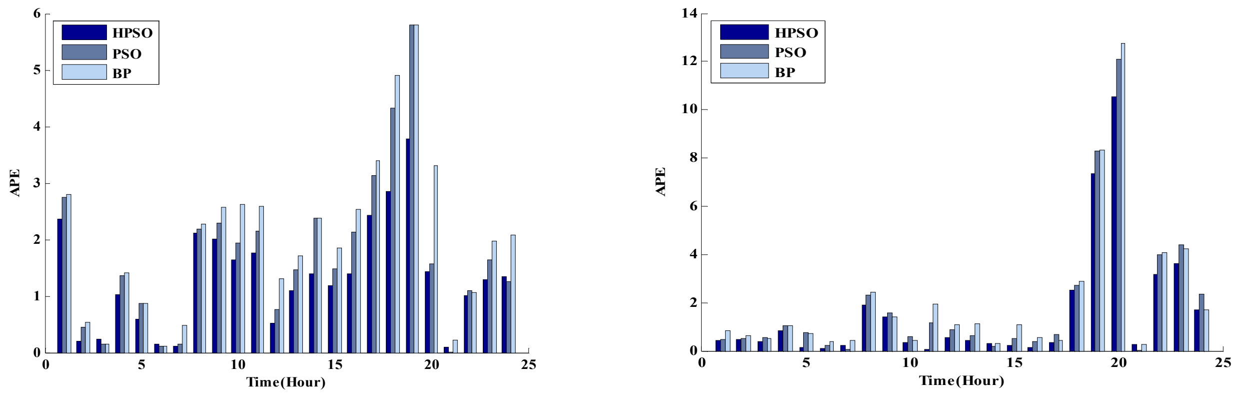
Table 4. APE of maximum daily load for different day classes

Day-classes		APE		
		BP	**PSO**	**HPSO-BP**
Class 1	Saturdays	0.497	0.0096	0.03
Class 2 (Remaining Working days)	Sundays	0.4308	0.0	0.0
	Mondays	0.412	0.0	0.0
	Tuesdays	0.246	0.0	0.0
	Wednesdays	0.3	0.0	0.0
Class 3	Thursdays	0.0466	0.0114	0.0137
Class 4	Fridays	0.174	0.0	0.0011



a) b)

Figure 11. ANN forecasted loads for sample of two special days: (a) 14 Khordad (b) Arbein



a) b)

Figure 12. APE of two typical of special days: (a) 14 Khordad (b) Arbein

Table 5. MAPEs of two special days with HPSO, PSO and BP algorithms

Special days (Day of forecast)	MAPE		
	BP	PSO	HPSO-BP
Arbein (Religious holiday)	2.04	1.727	1.332
14 Khordad (National holiday)	2.067	1.938	1.562

REFERENCES

[1] H. Shayeghi, H.A. Shayanfar and G. Azimi, "Intelligent Neural Network Based STLF", Int. Journal of Intelligent Systems and Technologies, Vol. 4, No. 1, pp. 17-27, 2009.
[2] H. Shayeghi, H.A. Shayanfar, A. Jalili and M. Porabasi, "A Neural Network Based Short term Load Forecasting", Proc. of the 2007 International Conf. on Artificial Intelligence (ICAI 07), Las Vegas, U.S.A., June 2007.
[3] T.W.S. Chow and C.T. Leung, "Neural Networks Based Short term Load Forecasting Using Weather Compensation", IEEE Trans. Power Systems, Vol. 11, No. 4, pp. 1736-1742, 1996.
[4] Z.A. Bashir and M.E. EL-Hawary, "Applying Wavelets to Short term Load Forecasting Using PSO-Based Neural Networks", IEEE Trans. on Power Systems, Vol. 24, No. 1, pp. 20-27, 2009.
[5] R. Afkhami and F. Mosalman Yazdi, "Application of Neural Networks for Short term Load Forecasting", in Proc. of IEEE Power India Conf., pp. 5, 2006.
[6] D. Srinivasan, "Evolving Artificial Neural Networks for Short Term Load Forecasting", Neurocomputing, Vol. 23, pp. 265-276, 1998.
[7] G.C. Liao and T.P. Tsao, "Application of a Fuzzy Neural Network Combined with a Chaos Genetic Algorithm and Simulated Annealing to Short term Load Forecasting", IEEE Trans. Evolutionary Computation, Vol. 10, No. 3, pp. 330-340, 2006.
[8] C.F. Juang, "A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design, IEEE Trans. on systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 34, No. 2, pp. 997-1006, 2004.
[9] A.U. Asar, S.R.U. Hassnain and A. Khan, "Short term Load Forecasting Using Particle Swarm Optimization Based ANN Approach", Proc. of IEEE Int. Joint Conf. on Neural Networks, Orlando, Florida, USA, pp. 6, 2007.
[10] H. Shayeghi, H.A. Shayanfar and G. Azimi, "STLF Based on Optimized Neural Network Using PSO", Int. J. of Electronics, Circuits and Systems, Vol. 3, No. 2, pp. 113-123, 2009.
[11] H. Shayeghi, A. Jalili and H.A. Shayanfar, "Multi-Stage Fuzzy Load Frequency Control Using PSO", Energy Conversion and Management, Vol. 49, pp. 2570-2580, 2008.
[12] J. Kennedy and R. Eberhart, "Particle Swarm Optimization, Proc. of IEEE Int. Conf. Neural Network, Vol. 4, pp. 1942-1948, 1995.
[13] M. Clerc and J. Kennedy, "The Particle Swarm Explosion, Stability, and Convergence in a Multidimentional Complex Space", IEEE Trans. on Evolutionary Computation, Vol. 6, No. 1, pp. 58-73, 2002.
[14] J.R. Zhang, J. Zhang, T.M. Lok and M.R. Lyu, "A Hybrid Particle Swarm Optimization Back Propagation Algorithm for Feed-forward Neural Network Training", Applied Mathematics and Computation, Vol. 185, pp. 1026-1037, 2007.
[15] Y. Shi, "Particle Swarm Optimization", Electronic Data Systems, Inc. IEEE Neural Networks Society, pp. 8-13, 2004.
[16] X. Wang, T. Wang and B. Wang, "Hybrid PSO-BP Based Probabilistic Neural Network for Power Transformer Fault Diagnosis", Second International Symposium on Intelligent Information Technology Application, IEEE, pp. 545-549, 2008.

BIOGRAPHIES

Hossein Shayeghi received the B.S. and M.S.E. degrees in Electrical and Control Engineering in 1996 and 1998, respectively. He received his Ph.D. degree in Electrical Engineering from Iran University of Science and Technology, Tehran, Iran in 2006. Currently, he is an Assistance Professor in Technical Eng. Department of University of Mohaghegh Ardabili, Ardabil, Iran. His research interests are in the Application of Robust Control, Artificial Intelligence to Power System Control design, Operation and Planning and Power System Restructuring. He is a member of Iranian Association of Electrical and Electronic Engineers and IEEE.

Heidar Ali Shayanfar received the B.S. and M.S.E. degrees in Electrical Engineering in 1973 and 1979, respectively. He received his Ph. D. degree in Electrical Engineering from Michigan State University, U.S.A., in 1981. Currently, he is a Full Professor in Electrical Engineering Department of Iran University of Science and Technology, Tehran, Iran. His research interests are in the Application of Artificial Intelligence to Power System Control Design, Dynamic Load Modeling, Power System Observability Studies, Voltage Collapse, Congestion Management in a Restructured Power System, Reliability Improvement in Distribution Systems and Reactive Pricing in Deregulated Power Systems. He is a member of Iranian Association of Electrical and Electronic Engineers and IEEE.

Ghaffar Azimi received the B.S. and M.S.E. degrees in Electrical Engineering from Islamic Azad University, Science and Research Branch, and Iran University of University of Science and Technology, Tehran, Iran in 2001 and 2008, respectively. His areas of interest in research are application of artificial Intelligence to load forecasting.