World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:4, 2009

Intelligent Neural Network Based STLF

H. Shayeghi, H. A. Shayanfar, G. Azimi

Abstract**—**Short-Term Load Forecasting (STLF) plays an important role for the economic and secure operation of power systems. In this paper, Continuous Genetic Algorithm (CGA) is employed to evolve the optimum large neural networks structure and connecting weights for one-day ahead electric load forecasting problem. This study describes the process of developing three layer feed-forward large neural networks for load forecasting and then presents a heuristic search algorithm for performing an important task of this process, i.e. optimal networks structure design. 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 and weekends and special days. We find good performance for the large neural networks. The proposed methodology gives lower percent errors all the time. Thus, it can be applied to automatically design an optimal load forecaster based on historical data.

Keywords—Feed-forward Large Neural Network, Short-Term Load Forecasting, Continuous Genetic Algorithm.

I. INTRODUCTION

SORT Term Load Forecasting (STLF) can be defined as the forecasting of electric demand one-to-seven days in advance [1]. It plays an important role in power system planning and operation. Basic operation functions such as unit commitment, economic dispatch, fuel scheduling, hydro-thermal coordination, control of spinning reserve, and unit maintenance can be performed efficiently with an accurate forecast [2]. They are also required for power system security studies, including contingency analysis and load management. Another application of short-time load forecasting is optimizing the operational state of a power system in terms of load flow and reactive power management [3].

STLF is not an easy task. Load series are generally complex and the load at a certain hour depends on the loads from undetermined number of past hours. Moreover, weather variables such as temperature, daylight time, winds, humidity, etc. affect the consumption considerably [1].

Owing to the importance of STLF problem, different forecasting models have been employed in power systems for achieving forecasting accuracy. Among the models are time series, regression, state-space and Kalman filtering [4-8] methods. In addition, artificial intelligence-based algorithms have been introduced based on expert system [9-10], evolutionary programming [11], fuzzy system [2,13], Artificial Neural Network (ANN) [14-16], and a combination of these algorithms. Among these algorithms, ANN has received more attention because of its clear model, easy implementation and good performance. The main reason of ANN becoming so popular lies in its ability to learn complex and nonlinear relationships that are difficult to model with conventional techniques [17]. It should be noted that one neural network will not be capable of handling all load types, several data clusters are formed. As resemblance measure, correlation analysis is selected. Since the past loads, temperature and time (hour, day, season, etc.) play the greatest roles in next day's load; they are used as the input variables to the proposed model.

For the solution of the STLF problem, a large artificial neural network intelligence approach based on day-type cluster is chosen in this work. 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. 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. There is still no best method in determining the number of neurons for hidden layer. This number is usually found by using heuristics. One has often to resort to is the trial and error approach.

For this reason, Continuous Genetic Algorithm (CGA) is employed to evolve the optimum large neural networks structure and connecting weights for one-day ahead electric load forecasting problem in this paper. The design of neural networks using genetic algorithm (GA) can be very helpful in terms of two main issues [18]:

- It automates the design of the network which will otherwise have to be done by hand using trial and error.
- The process of design can be analogous to a biological process in which the neural network blueprints encoded in chromosomes develop through an evolutionary process.

In this study, the optimized large neural networks modeling approach using the continuous genetic algorithm for short-term load forecasting based on only multiple delayed historical power load data and temperature data is proposed. 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 performance of the proposed method is also compared with the forecasting results of the back propagation based neural network model.

H. Shayeghi is with the Department of Technical Eng., University of Mohaghegh Ardabili, Ardabil, Iran (corresponding author to provide phone: 98-551-2910; fax: 98-551-2904; e-mail: hshayeghi@ gmail.com).

H. A. Shayanfar and G. Azimi is with the Electrical Engineering Department, Iran University of Science and Technology, Tehran, Iran.

In recent years, significant advancement has been made in the field of genetic algorithms. Genetic algorithm-based load forecasting methods [12,14] have been reported to yield results that have been more than encouraging. The majority of models that used artificial neural networks (e.g. the most frequently used back propagation model [15-16]) have a set problems:

a) Dependence on initial parameters.
b) Long training time.
c) Lack of problem-independent way to choose appropriate network topology.
d) Incomprehensive (black box) nature of ANNs.

In order to overcome some of these drawbacks continuous genetic based design of neural networks has been proposed. Here CGA is employed to evolve the optimum neural network structure and weights for one-day ahead electric load forecasting problem. GAs are powerful search techniques to find the global optimal area. During evolution, GAs require only information of the quality of the fitness value produced by each parameter set. This differs from many optimization methods requiring derivative information or complete knowledge of the problem structure and parameter [19].

Binary Genetic Algorithm (BGA) has been also used in training FNNs for solution of the STLF problem [14], but in training process, algorithm needs encoding operator and decoding operator. The number of bits to be used to represent each weight can have a significant effect. If too few bits are used the effect of weight quantization will be very significant resulting in poor convergence. On the other hand, a large number of bits per weight will again lead to slow convergence because of a long chromosomes string [18]. In order to overcome this drawback, the CGA method is proposed to improve the speed of the algorithms convergence for solution of the STLF problem. According to the performance criteria, continuous genetic algorithm is used to globally optimization the large neural networks architecture. The optimization process determines the number of neurons in the hidden layer. The proposed method is applied to STLF of the local utility. The simulation results show that the proposed CGA based method provides a greater degree of accuracy in many cases for STLF problem, compared to the BP method. 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. LOAD DATA ANALYSIS

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. Fig 1 shows hourly load curves for a sample week. 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.

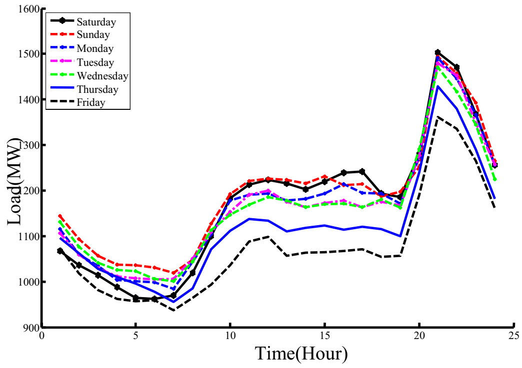


Fig. 1. Daily load curves for a typical week, Saturday, Sunday, to Wednesday, Thursday and Friday

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:

$$C_{xy} = \frac{S_{xy}}{\sqrt{S_{xx} S_{yy}}} \qquad (1)$$

Where,

$$S_{xy} = \sum_{i=1}^{n} |x_i - \bar{x}||y_i - \bar{y}|, \qquad S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2$$

$$S_{yy} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

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.

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

From the Table 1 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 [1].

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. 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.

III. ANN ARCHITECTURE FOR 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 Fig. 2) and every day (vide Fig. 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) [15,20]. Fig. 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. The temperature would also differ in every season, that is, winter would have the lowest temperature and summer the highest temperature.

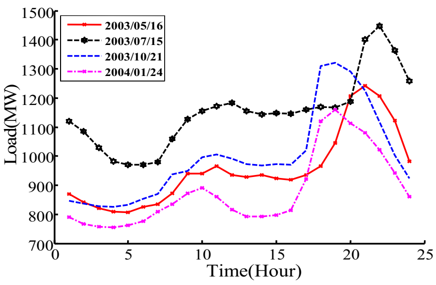


Fig. 2. Daily load curves for a typical day for each season

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 GA based method.

A. Proposed ANN Architecture

The process of developing an artificial neural network based on load forecasting can be divided into 5 steps [16]:

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.

A.1. Selection of input variables

One of the keys for designing a good architecture in ANN is choosing appropriate input variables. 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 [15, 20]. As shown in Fig. 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 [21].

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 Fig. 3.

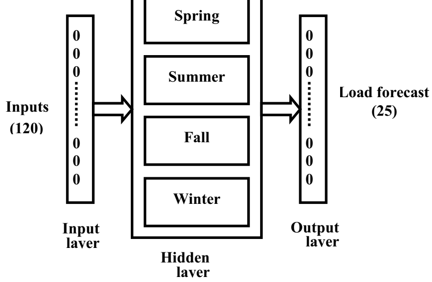


Fig. 3. Block diagram of the proposed ANN Architecture

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 day.
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 week.

A. 2 Design of neural network structure

The design of neural network architecture involves making several decisions regarding the type, size and number of neural network used. The well-known 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 (see previous subsection for more details) 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 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. Two commonly used activation functions are logistic and hyperbolic tangent. 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.

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. It has to be determined by heuristics method, since there is no general approach in available to determine the exact number of neurons in the hidden layer. 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. But in this research, a heuristic method is proposed for find the optimal number of neurons in the hidden layer. By using continue genetic algorithm, the number of neurons in the hidden layer is determined based on the flowchart as shown in Fig 4, that is Fig. 5 is shown the optimal number of neurons in the hidden layer based on the mean absolute percentage error (MAPE) performance index for each of the four-day classes in the summer.

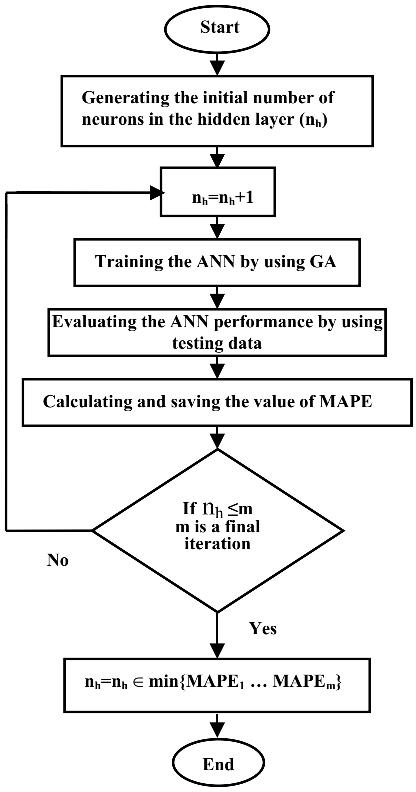


Fig. 4. The flowchart of the proposed method

A.3 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 [10]. Data normalization can be calculated by the following equations:

$$a_i=(\beta-\alpha)/(\max(x_i)-\min(x_i)) \text{ , } b_i=\beta-a_i\times \max(x_i) \quad (2)$$
$$X^N_{i,j}= a_i\times x_{i,j} +b_i \quad (3)$$

Where, $x_{i,j}$ and $X^N_{i,j}$ refer to the actual hourly temperature/load and the normalized value of the *i*th day at the jth hour respectively; also $X^N_{i,j}$ 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 *i*th day, respectively. The training output values are also normalized in the same manner. The normalized output values are then renormalized again to obtain the load values in MW.

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.

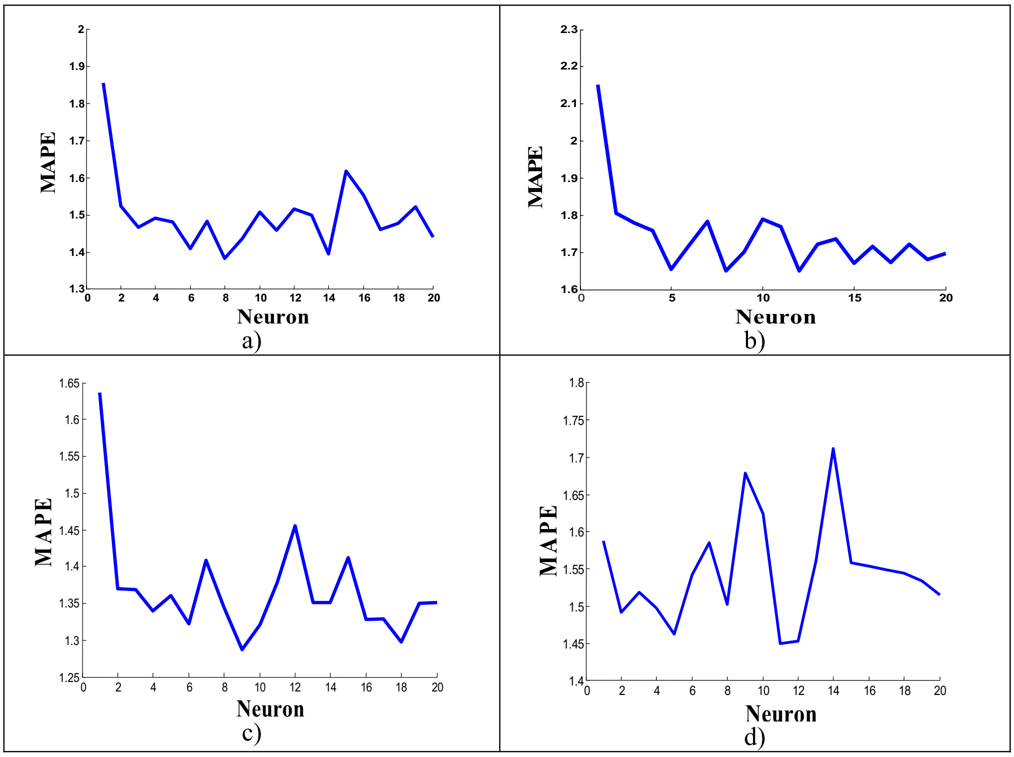


Fig.5. The optimal number of neurons based on the MAPE performance index for each of the four-day classes
a) Saturday b) Sunday-Wednesday c) Thursday d) Friday.

A.4 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 [18, 20]. Most applications use a supervised learning ANN.

The most commonly used training method is known as the back propagation algorithm [15-16, 22]. This algorithm is an iterative procedure. It has some drawbacks. Some of them are the slowness of learning speed, possibility of falling into local minimum and the necessity of adjusting a learning constant in every application [18]. In order overcome these drawbacks genetic based design of neural networks has been proposed.

A.5 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 [3].

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. Neural network training algorithms

The ANN use the training algorithms that are pre-determined, because the ANN which is created, initialing, is purposeless and after some time the training algorithms change behavioral condition of all the neurons of ANN or, in other words, its connection weights and corrects them in order to have a suitable reaction suitable to achieve desirable objective. In this section, the proposed neural network is employed to learn the input-output relationship of an application using the BP and CGA.

A. Back propagation algorithm

Regarding the ANN training, the mostly used training algorithm is the gradient descending method based on BP algorithm. Hence, some inherent problems existing in BP algorithm are also frequently encountered in the use of this algorithm. Firstly, the BP algorithm will easily get trapped in local minima especially for those nonlinearly separable pattern classification problems or 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 the 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 FNN's structure becoming more complex, its convergent speed will be even slower [23].

The BP algorithm is an iterative procedure. There are three steps performed during at each iteration [16]:

1. Forward pass-the outputs are calculated for given inputs.
2. Backward pass-the errors at the output layer are propagated backwards toward the input layer, with the partial derivatives of the total error with respect to the weights calculated in each layer appearing along the way.
3. Weights adjustment-a multivariate optimization algorithm finds the weights minimizing the error based on the gradient computed.

Usually, the mean squared error is employed as the minimized error function which assures faster convergence of the training process.

The back propagation algorithm suffers from two major drawbacks, namely network paralysis and trapping at local minima. These issues are briefly outlined below [24].

- Network paralysis: As the network receives training, the weights are adjusted to large values. This can force all or most of the neurons to operate at large outs, i.e., in a region where F'(Net)→0. Since the error sent back for training is proportional to F'(Net), the training process comes to a virtual standstill. One way to solve the problem is to reduce learning rate (η), which, however, increases the training time.
- Trapping at local minima: Back propagation adjusts the weights to reach the minima (see Fig. 6) of the error function (of weights). However, the network can be trapped in local minima. This problem, however, can be solved by adding momentum to the training rule or by statistical training methods applied over the back propagation algorithm.

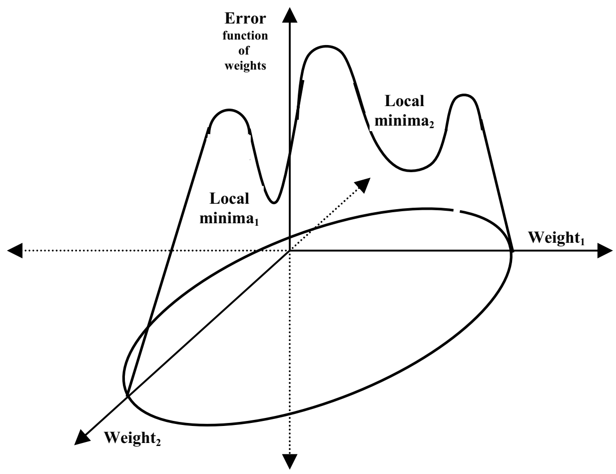


Fig. 6. Valleys in error function cause back propagation algorithm trapped at local minima

B. Continuous genetic algorithm

GAs are search algorithms based on the mechanism of natural selection and natural genetics that operate without knowledge of the task domain and utilize only the fitness of evaluated individuals. In general, reproduction, crossover and mutation are the three basic operators of GAs. They can be considered as a general purpose optimization method and have been successfully applied to search and optimization [25-[27]. During the evolution, GA requires only information of the quality of the fitness value produced by each parameter set. This differs from many optimization methods requiring derivative information or complete knowledge of the problem structure and parameter [19]. GA is applied to neural network in two different ways:

- They either employ a fixed network structure whit connection under evolutionary control.
- They are used in designing the structure of the network.

Thus, the evolution that has been introduced to neural networks can be divided roughly into different levels: (a) connection weights; (b) architecture; (c) learning rules.

In the application of GA for training of neural networks, the parameters of the problem are encoded as a set of chromosomes called the population and the candidate solutions are assigned fitness values based on the constraints of the problem. Based on each individual's fitness, a selection mechanism selects a mate with high fitness value for genetic manipulation. The manipulation process uses standard crossover and mutation genetic operators to produce a new population of individuals (offspring). Genetic algorithms are based on models of genetic change in a population of individuals. These models consist of three basic elements: a fitness measure which governs an individual's ability to influence future generations, a selection and reproduction process which produces offspring for the next generation and genetic operators which determine the genetic makeup of the offspring. The distinguishing feature of a GA with respect to other function optimization techniques is that the search towards an optimum solution proceeds not by incremental changes to a single structure but by maintaining a population of solutions from which new structures are created using genetic operators [18].

Binary Genetic Algorithm (BGA) has been also used in training FNNs for solution of the STLF problem [14]. But in training process, algorithm needs encoding operator and decoding operator. Usually there are three kinds of complicated evolutionary operators with this algorithm, i.e., selection, crossover and mutation, it was found in experiments that when the structure of FNNs is simple, its training results may be better than the ones using the BP algorithm to train, when the structure of FNNs becomes complex and there are large training samples, the GA's convergent speed will become very slow, so that the convergent accuracy may be influenced by the slow convergent speed [23]. The number of bits to be used to represent each weight can have a significant effect. If too few bits are used the effect of weight quantization will be very significant resulting in poor convergence. On the other hand, a large number of bits per weight will again lead to slow convergence because of a long chromosomes string [18]. In order to overcome this drawback, the CGA method is proposed to improve the speed of the algorithms convergence. When the variables are continuous, it is more logical to represent them by floating-point numbers. In addition, since the binary GA has its precision limited by the binary representation of variables, using floating point numbers instead easily allows representation to the machine precision. This continuous GA also has the advantage of requiring less storage than the binary GA because a single floating-point number represents the variable instead of N_{bits} integers. The continuous GA is inherently faster than the binary GA, because the chromosomes do not have to be decoded prior to the evaluation of the cost function. Fig. 7 shows the flowchart of continuous GA.

The main control parameters of a GA are: the population size, the selection mechanism, the crossover rate, the mutation rate and the number of generations allowed for the evolution of the required structure. The basic GA cycle developed for this problem consists of the following steps:

1. Construct an initial population of chromosomes.
2. Evaluate the fitness of each chromosome.
3. Perform fitness scaling if necessary.
4. Select the mating pairs of chromosomes.
5. Create new offspring through crossover and mutation operations
6. From a population for the next generation.
7. If process has converged, return the best chromosome as the solution, otherwise go to step2.

Often, a scaling of the performance measure is provided to increase diversity in a population. This process avoids domination by a super individual that would lead to a convergence at a local optimum. The probabilistic nature of the selection process gives a chance of production even to the weakest member of the population. Similarly, there is a chance that the best performing member (elite) might not be present in the next generation, either due to non-selection for mating or due to structural changes following crossover and mutation. In any search it is always desirable to copy the elite structure intact into the next generation [14].

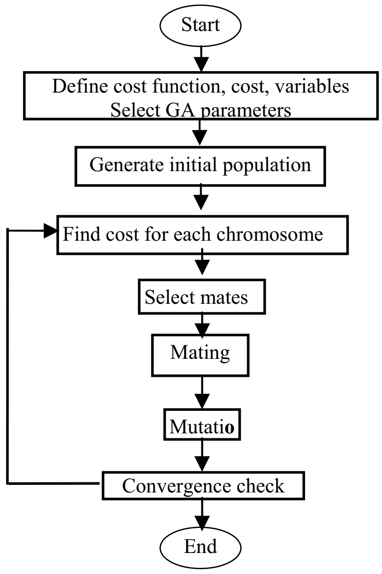


Fig.7. Flowchart of a continuous GA

V. Handling the Special Days

The special holidays of Iran are divided into 2 groups: solar and lunar calendar special days. Thus, n Iran, there are tow kinds of holidays, national and religious. National holidays are fixed in time, but religious lunar 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 Fig. 8. If the neural networks, designed for regular load forecasting, are directly used for special day load forecasting, large errors are observed because of this fact. 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.

In this paper, 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 continuous genetic algorithm, the number of hidden layer are fined and 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 Fig. 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 II
SPECIAL DAILY LOAD CORRELATIONS IN YEAR 2003

Special days (Day of forecast)	Previous day	Previous Friday	The same special day in the previous year
Gorban festival (Religious holiday)	0.9581	0.9895	0.9670
22 Bahman (National holiday)	0.9959	0.9843	0.9938

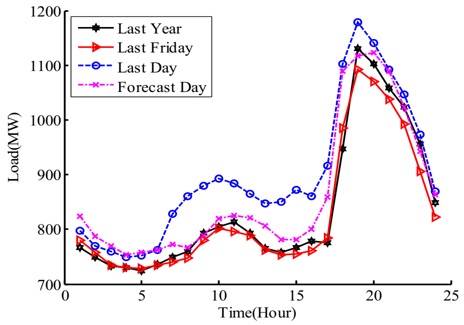


Fig. 8. Daily load curves for a special day, previous day previous Friday and the same special day in the previous year.

V. 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:

$$MAPE = \frac{1}{N}\sum_{1}^{N}\left|\frac{Actual_i - Forecast_i}{Forecast_i}\right| \times 100 \quad (4)$$

$$APE = \left|\frac{Actual_i - Forecast_i}{Forecast_i}\right| \times 100 \quad (5)$$

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 genetic and back propagation algorithms.

To demonstrate the effectiveness of the proposed ANN based model for solution of the STLF problem, some simulations are carried out. Two sets of hourly load patterns (weekdays and weekends) for some days are used to evaluate the performance of ANN load forecasting models based on training algorithm. ANN is designed for each of four-day classes and special days, we ran the two training algorithms (CGA and BP), respectively. Samples from the evaluation results are shown in Figs. 9 and 10 for four-day classes. Fig. 9 shows the actual and forecasted daily load. Fig. 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. Fig. 11 shows the actual and forecasted daily load of two typical of special day (Gorban festival and 22 Bahman). Fig. 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 GA and BP training methods. The results evaluation shows that, the proposed CGA based ANN model is effective and not only has a good performance, but also, achieves higher accuracy than the BP algorithm and has very less the forecasted load errors.

TABLE III
MAPE BY GA AND BP ALGORITHMS FOR DIFFERENT DAY CLASSES

Day-classes		MAPE	
		BP	GA
Class 1	Saturdays	0.9793	0.9371
Class 2 (Remaining working days)	Sundays	1.2267	0.8618
	Mondays	1.4034	0.9456
	Tuesdays	1.9133	1.6194
	Wednesdays	1.908	1.5472
Class 3	Thursdays	1.4248	1.268
Class 4	Fridays	1.1704	1.0697

TABLE IV
APE OF MAXIMUM DAILY LOAD FOR DIFFERENT DAY CLASSES

Day-classes		APE	
		BP	GA
Class 1	Saturdays	0.4488	0.3323
Class 2 (Remaining Working days)	Sundays	0.3625	0.0545
	Mondays	0.3775	0.0189
	Tuesdays	0.3686	0.0808
	Wednesdays	0.3706	0.1546
Class 3	Thursdays	0.1549	0.0312
Class 4	Fridays	0.4247	0.1534

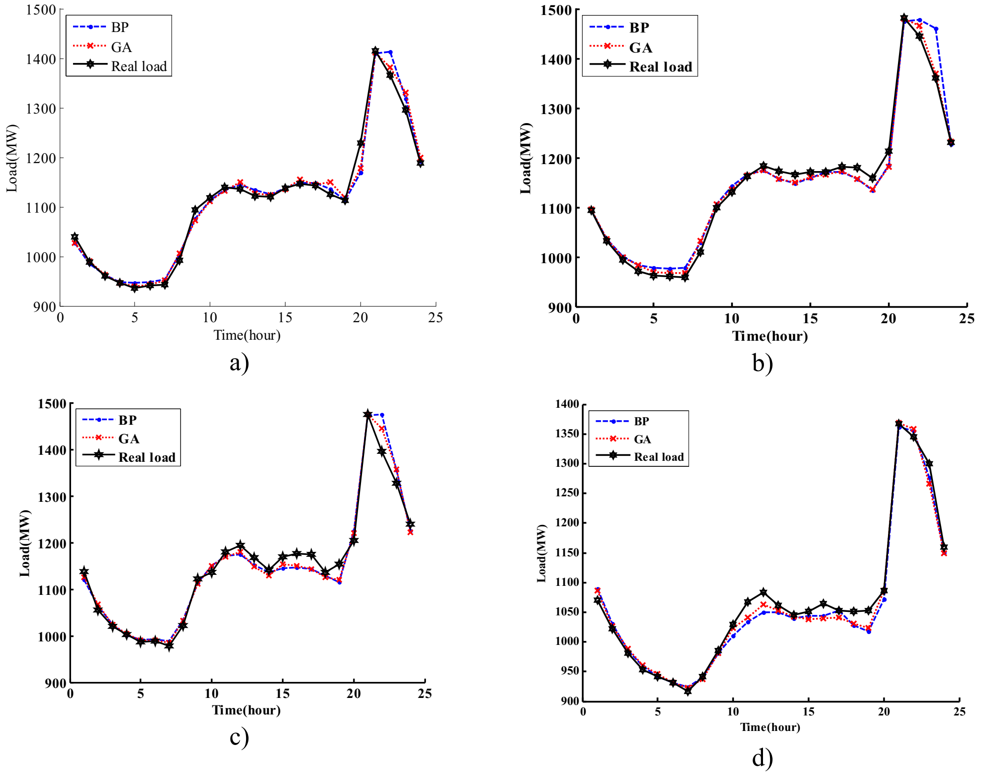


Fig. 9. ANN forecasted loads for sample four-day classes: (a) Saturday (b) Remaining working days (c) Thursday (d) Friday

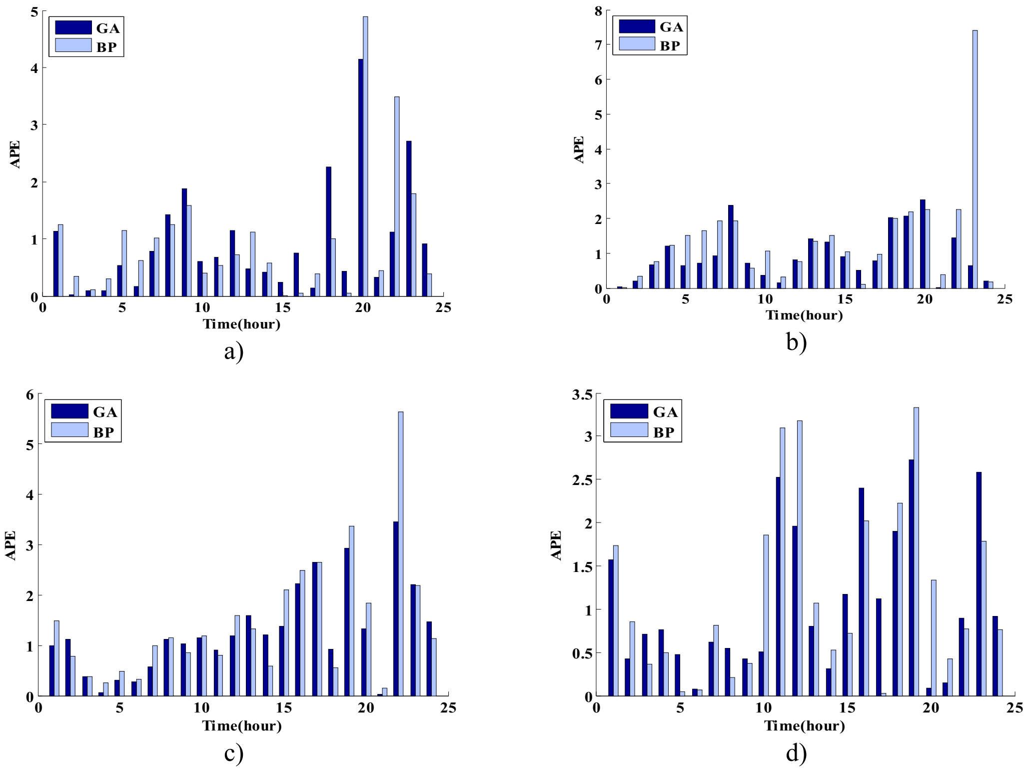
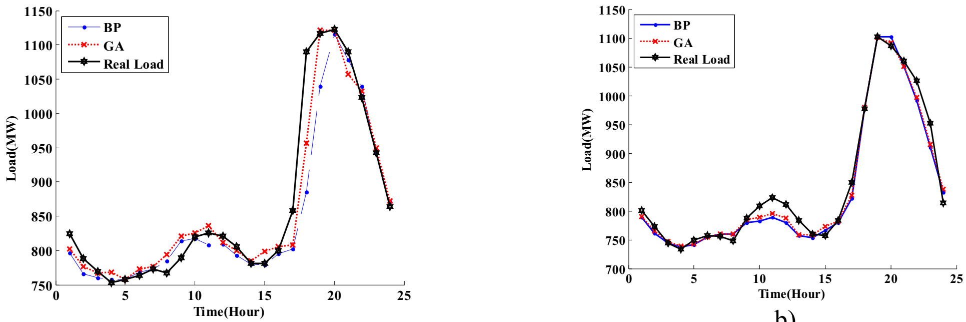
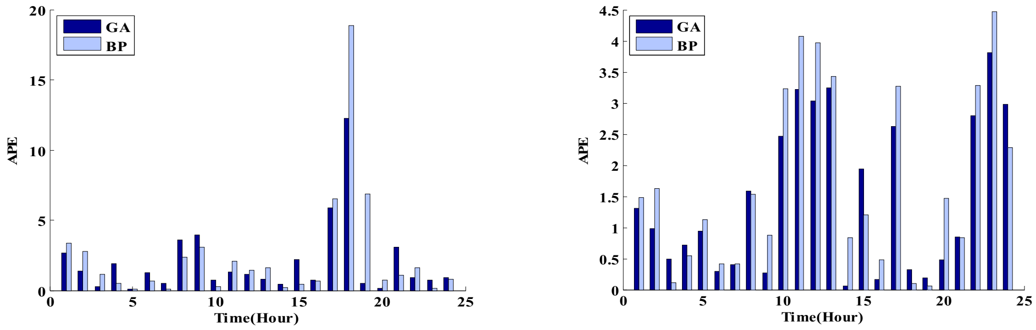


Fig. 10. Forecasted absolute percentage errors for sample four-day classes: (a) Saturday (b) Remaining working days (c) Thursday (d) Friday



a) b)

Fig. 11. ANN forecasted loads for sample of two special days: (a) Gorban festival (b) 22 Bahman.



a) b)

Fig.12. APE of two typical of special days: (a) Gorban festival (b) 22 Bahman.

TABLE V
MAPEs OF TWO SPECIAL DAYS WITH GA AND BP ALGORITHMS

Special days (Day of forecast)	MAPE	
	BP	GA
Gorban festival (Religious holiday)	2.3949	1.9709
22 Bahman (National holiday)	1.713	1.4656

X. CONCLUSIONS

This paper describes the process of developing three layer feed-forward large neural networks for short-term load forecasting based on CGA technique. It is based on clustering data analysis and correlation measures. Data clustering is considered load consumption profile of a local utility in Iran; and the proposed solution for all day types, including special days. 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. At present there is no systematic methodology for optimal design and training of an artificial neural network for short-term load forecasting. For this reason, continuous genetic algorithm is employed to find the optimum large neural networks structure and connecting weights for one-day ahead electric load forecasting problem in this paper.

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 test results show that the proposed forecasting methods could provide a considerable improvement of the forecasting accuracy for the regular and the special days. 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 ANN-CGA based model produces lower prediction error and is superior to the ANN-BP method. 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.

REFERENCES

[1] A. K. Topalli, I. Erkmen, I. Topalli, Intelligent short-term load forecasting in Turkey, *Electrical Power and Energy Systems*, Vol. 28, 2006, pp. 437-447.

[2] K.-H. Kim, H.-S. Youn and Y.-C. Kang, Short-term load forecasting for special days in Anomalous load conditions using neural networks and fuzzy inference method, *IEEE Trans. on Power Systems*, Vol. 15, No. 2, 2000, pp. 559-565.

[3] H. shayeghi, H. A. Shayanfar, A. Jalili, M. Porabasi, A Neural network based short-term load forecasting, *Proc. of the 2007 Internatinal Conf. on Artificial Intelligence (ICAI 07),* Las Vegas, U.S.A., June 2007.

[4] G. Box, G.M. Jenkins, Time series analysis, forecasting and control, San Francisco: Holden-Day; 1970.

[5] J.H. Park, Y.M. Park and K.Y. Lee, Composite modeling for adaptive short-term load forecasting, *IEEE Trans. on Power Systems*, Vol. 6, 1991, pp. 450-457.

[6] A. D. Papalexopoulos, T. C. Hesterberg, A regression-based approach to short-term system load forecasting, IEEE Trans on Power Systems, Vol. 4, No. 4, 1990, pp. 1535-1547.

[7] J.W. Taylor, R. Buizza, Using weather ensemble predictions in electricity demand forecasting, *International Journal of Forecastingt*, Vol. 19, 2003., pp. 57-70.

[8] A. D. Trudnowski et al., Real-Time very short-term load prediction for power system automatic control, IEEE Trans. Control Systems Technology, Vol. 9, No. 2, 2001, pp. 254-260.

[9] M. S. S. Rao, S. A. Soman, B. L. Menezes, P. Chawande, P. Dipti, T. Ghanshyam, An expert system approach to short-term load forecasting for reliance energy limited, *IEEE PES Meeting*, Mumbai, 2006.

[10] S. Rahman and R. Bhatnagar, An expert system based algorithm for short-term load forecasting, *IEEE Trans. on Power Systems*, Vol. 3, No. 2, pp. 392-399. 1988.

[11] T. P. Tsao, G. C. L. Iao, S. H. Chen, Short-term load forecasting using neural networks and evolutionary programming, *Proc. of Fifth International Power Engineering Conference*, Singapore, 2001, pp. 743-748.

[12] S. H. Ling, F. H. F. Leung, H. K. Lam, Y.-S. Lee, P. K. S. Tam, "A novel genetic-algorithm-based neural network for short-term load forecasting, *IEEE Trans. on Industrial Electronics*, Vol. 50, No. 4, 2003, pp. 793-799.

[13] S. Sheng, C. Wang, Integrating radial basis function neural network with fuzzy control for load forecasting in power system, *IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific*, Dalian, China, 2005, pp. 1-5.

[14] D. Srinivasan, Evolving artificial neural networks for short term load forecasting, *Neurocomputing*, Vol. 23, 1998, pp. 265-276.

[15] J. A. Momoh Y. Wang, M. Elfayoumy, Artificial neural network based load forecasting, *IEEE International Conference on Systems, Man and Cybernetics, Computational Cybernetics and Simulation*, 1997, pp. 3443-3451.

[16] W. Charytoniuk, M.S. Chen, Neural network design short-term load forecasting, *Proc. of IEEE International Conference on Electric Utility Deregulation and Restructuring and Power Technologies*, City Univercity, London, 2000, pp. 554-561.

[17] C. Sun, D. Gong, Support vector machines with PSO algorithm for short-term load forecasting, *Proc. of IEEE International Conference on Networking, Sensing and Control*, 2006, pp. 676-680.

[18] N. O. Attoh-Okine, Application of genetic-based neural network to lateritic soil strength modeling, Construction and Building Materials, 2004, Vol. 18, pp. 619- 623.

[19] H. Shayeghi, A. Jalili and H. A. Shayanfar, Robust modified GA based multi-stage fuzzy LFC, *Energy Conversion and Management*, Vol. 48, pp. 1656-1670. 2007.

[20] M. Shahidehpour, H. Yamin, Z. Li, Market operations in electric power systems: forecasting, scheduling, and risk management, Wiley-Interscience Publication, 2002.

[21] H. S. Hippert, C. E. Pedreira, R. C. Souza, Neural networks for short-term load forecasting: A review and evaluation, *IEEE Trans. on Power Systems*, Vol. 16, No. 1, 2001, pp. 44-55.

[22] H.S. Hippert, D.W. Bunn, R.C. Souza, Large neural networks for electricity load forecasting: are they overfitted, *International Journal of Forecasting*, 2005, Vol. 21, pp. 425-434.

[23] J.-R. Zhang, J. Zhang, T.-M. Lok, M. R. Lyu, A hybrid particle swarm optimization-back propagation algorithm for feed-forward neural network training, *Applied Mathematics and Computation*, Vol. 185, 2007, pp. 1026-1037,.

[24] A. Konar, Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain, CRC Press, 1999.

[25] E. D. Goldberg, Genetic algorithms in search and machine learning, reading, MA: Addison-Wesley; 1989.

[26] D. T. Pham and D. Karaboga, Intelligent optimization techniques, genetic algorithms, Tabu search, simulated annealing and neural networks, Berlin, Germany: Springer, 2000.

[27] S. Amin, J. L. Fernandez-Villacanas, Dynamic local search, *Proc. of the 2nd International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, 1997, pp.129-132.

[28] R. L. Haupt, S. E. Haupt, Practical genetic algorithms, Wiley-Interscience publication, 2004.