



NOVEL MLP NEURAL NETWORK WITH HYBRID TABU SEARCH ALGORITHM

*M. Reza Peyghami**, *R. Khanduzi†*

Abstract: In this paper, we propose a new global and fast Multilayer Perceptron Neural Network (MLP-NN) which can be used to forecast the automotive price. Nowadays, the gradient-based techniques, such as back propagation, are widely used for training neural networks. These techniques have local convergence results and, therefore, can perform poorly even on simple problems when forecasting is out of sample. On the other hand, the global search algorithms, like Tabu Search (TS), suffer from low rate convergence. Motivated by these facts, a new global and fast hybrid algorithm for training MLP-NN is provided. In our new framework, a hybridization of an extended version of TS with some local techniques is constructed in order to train the connected weights of the network. The extended version of TS in the proposed scheme consists of a simple TS together with the intensification and diversification search methods, and the local search methods are based on a direct strategy of Nelder-Mead (NM) or Levenberg-Marquardt (LM) techniques. This hybridization leads us to have a global and fast trained network in order to use in some forecasting problems. To show the efficiency and effectiveness of our new proposed network, we apply our new scheme for forecasting the automotive price in Iran Khodro Company which is the biggest car manufacturer in Iran. The results are promising compared to the cases when we apply the TS and some other forecasting techniques individually. We also compare the results with the case when we employ the gradient-based optimization techniques such as LM, and global search methods such as Genetic Algorithm (GA) and hybrid of MLP-NN with GA.

Key words: *Forecasting, multilayer perceptron neural network, heuristic and meta-heuristics algorithms, hybrid tabu search, Nelder-Mead method, Levenberg-Marquardt algorithm*

Received: February 8, 2010

Revised and accepted: May 26, 2013

*M. Reza Peyghami – Corresponding Author

Department of Mathematics, K. N. Toosi Univ. of Tech., P. O. Box 16315-1618; Scientific Computations in OPTimization and Systems Engineering (SCOPE), K. N. Toosi Univ. of Tech., Tehran, Iran, E-mail: peyghami@kntu.ac.ir, Tel: +98-21-23064316

†R. Khanduzi

Department of Mathematics, Shiraz University of Technology, Shiraz, Iran, E-mail: r.khanduzi@sutech.ac.ir, raheleh.khanduzi@gmail.com

1. Introduction

Artificial Neural Networks (ANNs) have been deployed in a variety of real world problems [10, 15]. Recently, a wide class of interested application of Multilayer Perceptron Neural Network (MLP-NN) in the financial forecasting has been considered. One can refer to the most famous books in the literature for more information, for example see [5, 15]. The success of ANNs for a particular problem depends on the adequacy of the training algorithm according to the necessities of the problem. The Back-Propagation (BP) technique is a gradient-based optimization technique [17, 19], which has been widely used in the training of ANNs. The so-called Levenberg-Marquardt (LM) method [9, 26], another gradient-based method for training of the networks, is an extension of the Gauss-Newton technique. It is worth mentioning that the networks using the LM for training enjoy fast convergence rate compared to the BP-based training networks. Due to the local convergence property of the gradient techniques, they often become trapped at suboptimal solutions depending upon the initial random starting point. Therefore, using a global search technique seems more suitable for getting a global solution in the training of ANNs [22]. Due to this fact, using heuristic or meta-heuristic methods such as Genetic Algorithm (GA) or Tabu Search (TS) methods [21, 23] are of interest in the recent researches. However, meta-heuristic methods may suffer from costly computations due to their slow convergence. Thus, combining these methods with local search methods is a practical remedy to overcome their slow convergence rate. For this purpose, several attempts have been made in the literature for developing such combinations especially for GA, see e.g. [18, 20].

The popularity of TS has grown significantly in the past few years as a global search technique. However, most of the applications which are using this technique are the combinatorial optimization problems, such as the traveling salesman problem, design optimization, and the quadratic assignment problem [3, 13, 24].

As we are interested in forecasting the automotive price in the biggest car manufacturer company in Iran, i.e. Iran Khodro Company, it is of our great interest in this paper to use the same line of research in hybridizing local and global heuristic and meta-heuristic methods for accomplishing this job. The first work in this subject was done by the authors in [20] in which a hybridization of GA and LS is introduced for training connected weights in the network. This hybridization enjoys some advantages such as getting rid of falling in the local optima and having a fast convergence rate with low computational results. Motivated by this work, in this paper, our attempt is to introduce another hybridization technique for forecasting the automotive price. The proposed hybrid scheme enjoys a global optima and good performance compared with that presented in [20]. To be exact, our scheme leads us to have a significant reduction in computational time as well as the high accuracy of the obtained solution in comparison with that of [20].

In our new hybridized scheme, the connected weights in the ANN are trained by a Hybrid Tabu Search (HTS) which consists of a modified TS with some direct search methods such as Nelder-Mead (NM) or LM techniques. In the modified TS algorithm, three search procedures are employed; Exploration, Diversification and Intensification. In the exploration search, a reduction procedure is introduced to generate trial moves. In this stage, the concepts of TS memory elements called

Tabu Regions (TRs) and Tabu List (TL) are introduced to provide an anti-cycling rule. Moreover, another memory element called Visited Regions List (VRL) is also introduced as a tool for the diversification search to diversify the search of unvisited areas of the solution space. Assume that one of the best points obtained by the exploration and diversification searches is close enough to a the global minimum. Then, the intensification search is applied again at the final stage to refine the elite solutions that have been visited so far. Actually, the proposed diversification and intensification searches try to follow some best known strategies from the high level TS with a long term memory. From the best solution obtained at this stage, the NM or LM techniques are applied to speed up the rate of convergence. In order to show the efficiency and effectiveness of the proposed hybrid scheme, we apply the scheme for forecasting the automotive price in the Iran Khodro Company. As we are interested in comparing the results with the hybrid technique proposed in [20], we restrict ourselves to the price data obtained in the period of March 2007 to March 2009. Moreover, it has to be noted that, due to the policy of the Iran government in removing subsidies in the market since December 2010, in our opinion, the price data are not reliable enough for citation and, therefore, we ignore those here. The results of forecasting by the new scheme are compared with the simple TS and some gradient based techniques such as LM, individually. Moreover, the results are compared with the results that are reported in [20]. The comparisons show that our new scheme for forecasting is rather efficient in producing high accuracy and quality solutions than the other mentioned approaches.

The rest of the paper is organized as follows: In Section 2, the structure of an MLP-NN is provided in details. Section 3 is devoted to constructing the new hybrid scheme by providing its elements in more details. Simulation Results on forecasting the automotive price in the Iran Khodro Company as well as the comparisons are given in Section 4. We end up the paper by some concluding remarks in Section 5.

2. Structure of a Simple MLP-NN

A simple two layers MLP-NN structure, as described in [5, 15], can be outlined as in Fig. 1.

In this structure, the parameters are:

- p_j : the input node
- R : the number of input nodes
- S : the number of hidden nodes
- $w_{i,j}$: connected weights between input and hidden layers
- v_i : connected weights between hidden and output layers
- f : the activation function of hidden layer
- Σ : the activation function of output layer
- a : the output of the network

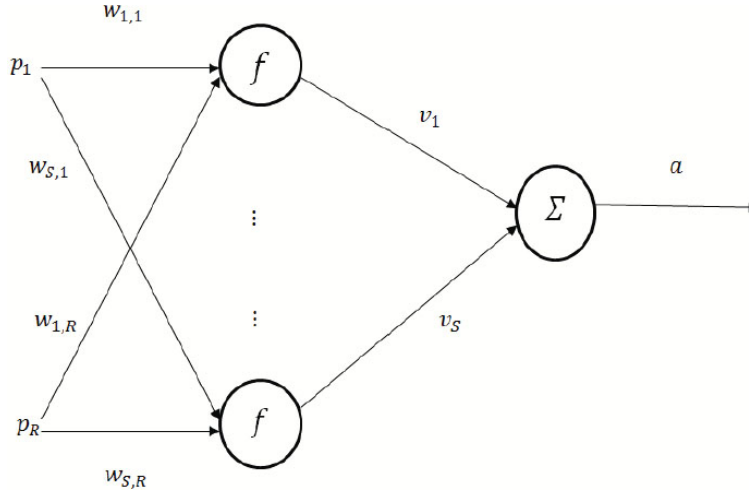


Fig. 1 MLP-NN structure.

It is worth mentioning that, according to the algorithms that are employed in training the connected weights and the activation functions, several MLP-NN have been studied and developed in the literature, see e.g. [14].

In our new proposed scheme for the network, $w_{i,j}$'s and v_i 's are trained by HTS method. For ease of reference, we denote these weights by $w_{i,j}^{HTS}$ and v_i^{HTS} , respectively. The *sigmoid* and *linear* functions are applied as activation functions in hidden and output layers, respectively. Finally, the output a is computed as follows:

$$a = \sum_{i=1}^s \left(\text{logsig} \left(\sum_{j=1}^s w_{i,j}^{HTS} p_j \right) + v_i^{HTS} \right),$$

where the *logsig* function is defined as

$$\text{logsig}(x) = \frac{1}{1 + e^{-x}}$$

McNelis in [15] introduced some factors that adequate the performance of an MLP-NN. Some of these factors are: the network architecture, number of input and hidden nodes, layer activation functions, number of layers, the convergence criteria and learning algorithms for training the network. These factors in our new proposed scheme are: the architecture of the network consists of one hidden layer, the number of input nodes is obtained using trial and error process, the number of hidden nodes is determined using Zhang's relation [27], the *sigmoid* and *linear* functions are used as activation functions in hidden and output layers, respectively, and the convergence criteria for training are MSE or maximum iterations number.

It is worth mentioning that the *sigmoid* activation function needs data to be scaled into $[0,1]$. To do so, an efficient preprocessing of the data is necessary to input into the network [25]. We utilize the following normalization scheme for the

data, which is given as time series $X = (X_t)$:

$$Z_t = \frac{X_t - \min(X)}{\max(X) - \min(X)}$$

In this formula, $\min(X)$ and $\max(X)$ indicate the minimum and maximum values of the series X , respectively.

In what follows, we are going to describe the components of our hybridization scheme in more details.

3. The New Algorithm

As our hybridize scheme consists of several optimization techniques for training the connected weights in the network, such as LM technique, NM method, TS algorithm and their hybridizations, it is required to describe these methods in more details. In the sequel, we first provide the requirements of the new scheme.

3.1 The Levenberg-Marquardt technique

Many attempts have been made to speed up the BP algorithm. A significant improvement in realization performance can be observed by using various second order approaches such as Newton's method, conjugate gradients, and LM technique [2, 4, 9]. Among these methods, the LM algorithm is widely accepted as the most efficient algorithm in the sense of realization accuracy [26]. It gives a good compromise between the speed of the Newton method and the stability of the steepest descent method, and consequently it constitutes a good transition between these methods. In order to compare our simulation results with LM training approach, we briefly describe the LM training technique in this section. The performance index in the LM algorithm that should be optimized is the MSE function which is defined by:

$$F(\mathbf{w}) = \frac{\sum_{i=1}^R (d_i - a_i)^2}{N}, \tag{1}$$

where N is the number of the total weights, $\mathbf{w} = (w_1, w_2, \dots, w_N)^T$ is the weights of the network and d_i 's and a_i 's are the desired and actual values of the i^{th} pattern, respectively. Equation (1) can be rewritten as:

$$F(\mathbf{w}) = \mathbf{E}^T \mathbf{E},$$

where \mathbf{E} is the error vector defined by:

$$\mathbf{E} = [e_1, e_2, \dots, e_R]^T \tag{2}$$

and

$$e_i = \frac{d_i - a_i}{\sqrt{N}}, \quad i = 1, 2, \dots, R.$$

The Jacobian of \mathbf{E} is defined by:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \cdots & \frac{\partial e_1}{\partial w_N} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \cdots & \frac{\partial e_2}{\partial w_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_R}{\partial w_1} & \frac{\partial e_R}{\partial w_2} & \cdots & \frac{\partial e_R}{\partial w_N} \end{bmatrix}. \quad (3)$$

In the LM algorithm, the connected weights are computed by the following iterative scheme:

$$\mathbf{w}_{j+1} = \mathbf{w}_j - (\mathbf{J}_j^T \mathbf{J}_j + \mu \mathbf{I})^{-1} \mathbf{J}_j^T \mathbf{E}_j, \quad (4)$$

where \mathbf{I} is identity matrix, μ is the learning parameter and \mathbf{J}_j is the Jacobian matrix computed at the j -th iteration. Note that, for $\mu = \mathbf{0}$, this procedure produces the Gauss-Newton iterations, and for large enough μ , it generates the steepest descent iterations. In each iteration, the parameter μ is automatically adjusted in order to warrant convergence of the iterations.

3.2 The Nelder-Mead method

The Nelder-Mead method [16] is a local search technique and one of the most popular derivative-free optimization methods. In each iteration, the NM method starts by a none degenerate simplex, a N dimensional geometric shape of nonzero volume generated by the convex hull of the vertices w_1, w_2, \dots, w_{N+1} , and their respective function values to produce the new points and their function values leading to a new simplex. For this purpose, the following parameters should be specified:

- The reflection parameter $\rho > 0$
- The expansion parameter $\chi > 1$
- The contraction parameter $0 < \gamma < 1$
- The shrinking parameter $0 < \sigma < 1$

According to these parameters, the Nelder-Mead method can be outlined as follows [11]:

Algorithm 1. The Nelder-Mead method

Data: Given $\varepsilon > 0$, $\rho > 0$, $\chi > 1$, $0 < \gamma < 1$, and $0 < \sigma < 1$, repeat the following steps:

Order: Order and re-label the $N+1$ vertices as w_1, w_2, \dots, w_{N+1} so that $F(w_1) \leq F(w_2) \leq \dots \leq F(w_{N+1})$. We refer to w_1 as the best point, and w_{N+1} as the worst point.

Reflect: Compute the reflection point w_r by

$$w_r = \bar{w} + \rho(\bar{w} - w_{N+1}),$$

where \bar{w} is the centroid of the N best points, i.e., $\bar{w} = \frac{\sum_{i=1}^N w_i}{N}$. Evaluate $F(w_r)$. If $F(w_1) \leq F(w_r) \leq F(w_N)$, replace w_{N+1} by the reflected point w_r and go to **Stopping criteria**.

Expand: If $F(w_r) \leq F(w_1)$, compute the expansion point w_e by

$$w_e = \bar{w} + \chi(w_r - \bar{w}).$$

Evaluate $F(w_e)$. If $F(w_e) \leq F(w_r)$, replace w_{N+1} by w_e and go to **Stopping criteria**. Otherwise, replace w_{N+1} by w_r and go to **Stopping criteria**.

Contract: If $F(w_r) \geq F(w_N)$, perform a contraction between \bar{w} and the best point between w_{N+1} and w_r .

Outside: If $F(w_N) \leq F(w_r) < F(w_{N+1})$ (i.e., w_r is strictly better than w_{N+1}), perform an outside contraction by computing

$$w_{oc} = \bar{w} + \gamma(w_r - \bar{w}).$$

Evaluate $F(w_{oc})$. If $F(w_{oc}) \leq F(w_r)$, replace w_{N+1} by w_{oc} and go to **Stopping criteria**. Otherwise, go to **Shrink**.

Inside: If $F(w_r) \geq F(w_{N+1})$, perform an inside contraction by computing

$$w_{ic} = \bar{w} + \gamma(w_r - \bar{w}).$$

Evaluate $F(w_{ic})$. If $F(w_{ic}) \leq F(w_{N+1})$, replace w_{N+1} by w_{ic} and go to **Stopping criteria**. Otherwise, go to **Shrink**.

Shrink: Evaluate F at the new N points

$$w'_i = w_1 + \sigma(w_i - w_1), \quad i = 2, \dots, N + 1.$$

Replace the vertices w_2, \dots, w_{N+1} by the new vertices w'_2, \dots, w'_{N+1} .

Stopping criteria: Order and re-label the vertices of the new simplex as w_1, w_2, \dots, w_{N+1} so that $F(w_1) \leq F(w_2) \leq \dots \leq F(w_{N+1})$. If $F(w_{N+1}) - F(w_1) < \varepsilon$, then stop. Otherwise, go to **Reflect**.

—

3.3 The tabu search algorithm

The simple tabu search algorithm is a meta-heuristic method, originally proposed by Glover [6] in 1986. It was essentially proposed and developed for combinatorial optimization problems [7, 8]. The main feature of this algorithm is its use of an adaptive memory and responsive exploration. In this algorithm, the combination of a reduction procedure with anti-cycling memory-based rules prevent the search from getting trapped in the local minima. Specifically, TS constructs a list of more recently visited solutions, the so-called Tabu List (TL), in order to allow itself for getting back to them. In each iteration, at first, the algorithm generates many trial solutions in the neighborhood of the current solution in order to avoid from generating any trial solutions that are recently visited. Then, the best trial solution

among the generated solutions becomes the next solution. Therefore, TS can accept uphill movements in order to avoid getting trapped in the local minima. The method terminates if the number of iterations without any improvement exceeds a predetermined maximum number.

The concept of memory plays an important role in TS, especially when it is used as the learning processes. An efficient memory conception in intensification and diversification schemes makes TS to behave as an intelligent search technique.

The search algorithms in the optimization area can be classified in two categories; point-to-point methods and population-based methods. The TS method is one of the point-to-point methods. However, the long-term memory in TS makes it competitive with the population-based methods in keeping the diversity. In TS, with long-term memory, the search can be restarted from new diverse solutions whenever the diversification is needed, or can be intensified to improve the elite solutions whenever the intensification is needed. The concepts of diversification and intensification have turned out to be effective in many combinatorial optimization problems, see [2].

Let us briefly describe the concepts and implementations of the TS memory elements. First, we let TL to be a set of some visited solutions. The points in the TL are ranked and saved according to their recency and their objective function values. Therefore, some positions in the TL are kept for the best visited solutions which help an intensification scheme to refine the search from these best solutions at the final stage. Around each solution saved in the TL, two types of regions are specified in the search space. The first one is a Tabu Region (TR) in which no new trial point is allowed to be generated, and the other one is a Semi-Tabu Region (Semi-TR) that comprises a surrounding region around TR. The main role of the Semi-TRs is to generate neighboring around trial points in a special way so that returning to a visited TR is avoided when the trial solution lies inside a Semi-TR. Another memory element is the Visited Region List (VRL). The centers of the visited regions and the frequency of visiting these regions are saved in the VRL in order to direct a diversification scheme to explore the space outside the visited regions.

Now, the procedure of a simple extended Tabu Search algorithm can be outlined as follows [11]:

Algorithm 2. The tabu search algorithm

- Step 1.** Given an initial weight w_0 , set the Tabu List (TL) as empty and $k = 0$.
- Step 2.** According to the tabu restrictions, generate the neighborhood movements list $M(w_k) = \{w' : w' \in N(w_k)\}$, where $N(w_k)$ is a neighborhood of w_k .
- Step 3.** Set w_{k+1} to be equal to the best trial weight in $M(w_k)$, and update TL.
- Step 4.** If stopping criteria are not satisfied, then go to Step 2.

—

Algorithm 2 is called a short-term memory TS. Updating of the memory-based TL can be modified and controlled by the following concepts:

- **Tabu tenure:** number of iterations in which a tabu movement is considered to remain tabu or forbidden;
- **Aspiration criteria:** accepting an improving solution even if generated by a tabu movement.

The short-term memory is built to keep the recency only. In order to achieve better performance, long-term memory has been proposed to keep more important search features besides the recency, such as the quality and the frequency. Specifically, long-term memory in high-level TS records attributes of special characters like elite and frequently visited solutions. Then, the search process of TS can adapt itself by using these special types of solutions in:

- **Intensification:** giving priority to elite solutions in order to obtain much better solutions in their vicinity.
- **Diversification:** discouraging attributes of frequently visited solutions in new movement selection functions in order to diversify the search to other areas of solution space.

3.4 The new hybridized schemes

In the literature, two kinds of hybridization can be distinguished: strong and weak hybridizations. In the first one, the knowledge is included using specific operators or representations. In the latter case, several algorithms are combined somehow. In fact, in this case, an algorithm can be used to improve the results of another one separately or it can be used as an operator of the other one [1]. Our new hybridized schemes are the combination of two algorithms (weak hybridization) in which one of them acts as an operator in the other. Indeed, we hybridize TS with the LM algorithm (TS-LM) and TS with NM (TS-NM) in order to update the connected weights of the network. In both cases, the problem-specific algorithm (LM and NM) is used as a mutation-like operation of the general search template.

We stress that the success of an ANN strongly depends on its training algorithms. Gradient-based local search techniques, such as LM technique, have been widely used in the ANNs training softwares. However, in many cases, these algorithms are very slow and susceptible to fall in the local minima problem. Recently, there has been a growing interest in solving this problem using meta-heuristic methods such as TS method. However, meta-heuristic methods may suffer from costly computations due to their slow convergence rate. In this case, combining these methods with some local search methods such as NM or LM techniques seems to be a practical remedy to overcome their slow convergence rate. Therefore, the TS-NM and TS-LM combinations are made to perform as global and faster convergence rate schemes.

Two procedures of our hybrid schemes are outlined as follows:

Procedure 1. Hybridization of TS and LM

Step 1. Scale data to the interval $[0,1]$.

Step 2. Choose the number of layers, related nodes, activation functions and stopping criteria.

Step 3. Initialize trial weights and set $k = 1$.

Step 4. Update weights by TS-LM algorithm.

Step 5. Compute the Error function ($F(w)$).

Step 6. If the stopping criteria are not satisfied, then set $k = k + 1$ and go to Step 4. Otherwise, re-scale output and return k and $F(w)$.

—

Procedure 2. Hybridization of TS and NM

Step 1. Scale data to the interval $[0,1]$.

Step 2. Choose the number of layers, related nodes, activation functions and stopping criteria.

Step 3. Initialize trial weights and set $k = 1$.

Step 4. Update weights by TS-NM algorithm.

Step 5. Compute the Error function ($F(w)$).

Step 6. If the stopping criteria are not satisfied, then set $k = k + 1$ and go to Step 4. Otherwise, re-scale output and return k and $F(w)$.

—

4. Case Study

We have simulated and implemented our new hybrid schemes for forecasting the automotive price in the Iran Khodro Company. The data are the price data of some models of cars in this company. As we would like to compare the new schemes with those presented in [20], we restrict ourselves to the price data as considered in [20], i.e. in the period of March 2007 till March 2009 which consists of only working days price data and has been taken from one of the most famous economic daily journal in Iran called “Abrar Eghtesadi”. Moreover, since December 2010, the Iran government has started to remove subsidies in the market. This causes an ungovernable inflation in the products of the market especially in the products of the Iran Khodro Company. In addition, in order to control inflation, the daily journals are sometimes prohibited by the government to report the prices. This may cause an increase in the number of missing data in this period. Therefore, in our opinion, the price data from December 2010 are not reliable enough to be included in our study.

Although, the automotive price forecasting in the first step does not seem to be a suitable case for forecasting, there are some strong reasons for considering this case in the Iranian market in the period of 2007 to 2009, among them:

1. The price is not optionally determined by the producers and it is directly affected by the policies taken from the Ministry of Mine and Industries due to some parameters such as market demand, international sanctions, and some other natural phenomena.
2. The delivering of the produced cars in the company is not on time, i.e. the customers have to wait for some time (weeks and months) to get their car. In this case, forecasting of the automotive price for the dealers sounds to be very important.

We have implemented our new schemes in MATLAB 7.6.0 (R2008a) on a PC with 2.2 GHz, 2 MB of RAM and 160 GB of memory. As we are interested to compare the results with that of [20], we have adjusted the parameters the same as in [20], i.e.,

- 10 input nodes
- 21 hidden nodes
- One output node
- Hidden layer activation function is *logsig* function from MATLAB library
- Output layer activation function is *purelin* function from MATLAB library
- The stopping rule is $MSE \leq 10^{-4}$ or max iteration number =1000

The simulation results of applying new schemes and some other methods on the Iran Khodro Company price data are given in Tabs. I, II, III. For ease of reference, we use the following notations in the tables:

- MLP-LM: Training MLP-NN with LM method
- MLP-GA: Training MLP-NN with GA [20]
- MLP-TS: Training MLP-NN with TS
- MLP-LM-GA: Hybridizing MLP-LM with GA [12]
- MLP-TS-LM: Training MLP-NN with hybridization of TS and LM methods
- MLP-TS-NM: Training MLP-NN with hybridization of TS and NM methods

Three kinds of experiments are performed to compare the new hybridized schemes with some other hybrid schemes in the literature. These experiments are:

Experiment 1. In the first experiment, we apply the LM technique for training the connected weights in the network. The results are given in Tab. I. Although the aim in this experiment is to introduce a fast MLP-NN, the quality of the predicted solutions is not good enough.

Experiment 2. In this experiment, the global algorithms GA and TS are used to train the connected weights in the network. The results are provided in Table II. This experiment confirms the good performance for MLP-NN with slow rate. Also, the results show the priority of the TS than GA in the convergence rate.

Experiment 3. Experiment 3 is aimed at showing the quality of prediction in the new proposed hybrid schemes. The results of this experiment are given in Tab. III. This experiment shows the efficiency and effectiveness of the hybridized schemes in improving the quality of prediction in the network in comparison with the results obtained in Experiments 1 and 2.

	MLP-LM
MSE of training	0.1830×10^{-4}
Iteration	145
MSE of testing	0.3259×10^{-4}

Tab. I *The identification of TRAINING and TESTING errors for LM method.*

	MLP-GA	MLP-TS
MSE of training	0.0943×10^{-4}	0.0356×10^{-4}
Iteration	834	233
MSE of testing	0.1064×10^{-4}	0.2940×10^{-4}

Tab. II *The identification of TRAINING and TESTING errors for TS and GA algorithms.*

	MLP-LM+GA	MLP-TS-LM	MLP-TS-NM
MSE of training	0.0858×10^{-4}	0.0335×10^{-4}	0.0324×10^{-4}
Iteration	583	126	112
MSE of testing	0.1589×10^{-4}	0.1058×10^{-4}	0.1017×10^{-4}

Tab. III *The identification of TRAINING and TESTING errors for the new proposed hybrid schemes.*

Tabs. I-III provide the MSE of testing and training as well as number of iterations for the above-mentioned hybridized schemes. More precisely, Tab. I gives the MSE of testing and training as well as the iteration numbers for the MLP-LM algorithm. In Tab. II, the MSE of testing and training together with the number of iterations are provided for the MLP-GA and MLP-TS algorithms. Tab. III is devoted to reporting the MSE of testing and training as well as the iteration numbers for the new proposed schemes, i.e. plus iteration number of our new schemes

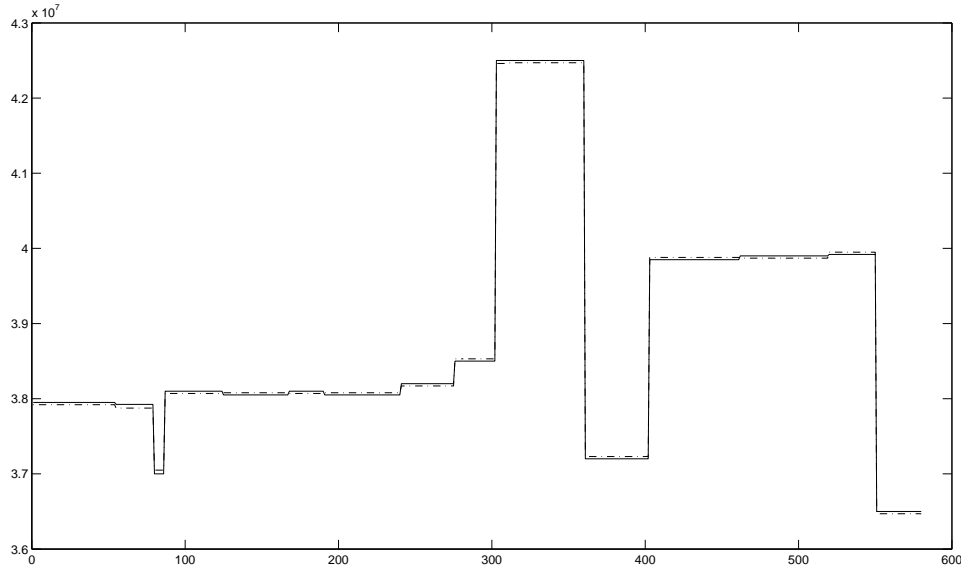


Fig. 2 Training of MLP-NN with TS-NM.

together with the MLP-LM-GA, MLP-TS-LM and MLP-TS-NM algorithms. As can be observed from the tables, the first stopping rule, i.e. $MSE \leq 10^{-4}$, is always satisfied in all experiences. Therefore, the less number of iterations can be considered as a good criterion for showing the efficiency of the algorithms. Due to this fact, one can realize that the new schemes are more efficient and effective than those obtained in Experiments 1 and 2. Moreover, among the new hybridized schemes, the MLP-TS-NM seems to be more efficient than the MLP-TS-LM and MLP-LM-GA schemes.

As the figures of the actual outputs and those obtained from the training of the network by the Experiments 1-3 are roughly close to each other, we just provide here the figure of the actual output and the network output obtained from the training of the network by TS-NM in Fig. 2. The black and dash lines in this figure indicate the actual output and the network output, respectively. Also, the horizontal axis in the figure denotes the working days in the period March 2007 to March 2009 (almost 570 days) and the vertical axis indicates the median price in each day.

Our observation on the performance of the training schemes of Experiments 1-3 are as follows:

In the periods of [1, 54], [55,79], and [551, 570] working days, the actual prices are 37950000, 37925000 and 36500000, respectively. These prices are predicted by MLP-TS, MLP-TS-LM, MLP-TS-NM and MLP-LM+GA algorithms well while they are not predicted properly by MLP-LM and MLP-GA algorithms. In the working days intervals [80,86], [276,302] and [520,550], the real prices are 37000000, 38500000 and 39920000, respectively. All the methods in these intervals have almost the same performance and are able to predict these prices as close as possible. For the period of [87,124] working days, the real price 38100000 is forecasted by all

the methods properly except for the MLP-LM algorithm. This scenario of prediction is repeated in the intervals [168,190], [191,240] and [241,275] of working days for the actual prices 38100000, 38050000 and 38200000, respectively. Moreover, all the algorithms except the MLP-GA and MLP-TS algorithms could be able to predict the real price 38050000 on the working days [125,167]. The actual price 42500000 in the period [303,360] of working days is predicted by all the methods but not as good as the prediction provided by the MLP-TS-NM method. In the interval [361,402] of working days, the real price 37200000 is predicted poorly by MLP-GA method while it is predicted by the other methods well. The prediction for the real price 39850000 in the interval [403,461] of working days is done poorly by the MLP-LM, MLP-GA and MLP-TS algorithms while the others have good performance in this period. In the interval [462,519] of working days, a weak prediction is done for the real price 39900000 by the MLP-LM, MLP-GA, MLP-TS and MLP-LM+GA algorithms while the MLP-TS-LM and MLP-TS-NM methods have a good performance.

It is our conviction that the proposed scheme can perform well in forecasting data in the other case studies and models in the real world phenomena, especially in economics and engineering. The authors would like to invite the interested readers to apply our approach to possible real data in their field of research to verify its performance on that data prediction.

5. Conclusion and Future Researches

In this paper, we present a hybrid MLP-NN learning algorithm in which the Hybrid Tabu Search (HTS) is used to train the connected weights between the layers of the network. Our new hybrid algorithm integrates TS and NM methods in the HTS in order to:

- (i) obtain a faster convergence rate than TS and GA, individually
- (ii) achieve more global results than LM

We have implemented our new scheme on the daily automotive price data of the Iran Khodro Company. Simulation results show good performance results in terms of quality of forecasting and excellent results in terms of convergence rate.

Acknowledgment

The authors would like to thank the Research Councils of K. N. Toosi University of Technology and Shiraz University of Technology for supporting this research. The work also has been supported by the Research and Development Office of the Iran Khodro Company for the second author.

References

- [1] Alba E., Chicano J. F.: Training Neural Networks with GA Hybrid Algorithms. K. Deb (ed.), Proceedings of GECCO'04, Seattle, Washington, LNCS 3102, 2004, pp. 852-863.

- [2] Battiti R., Tecchioli G.: The continuous reactive tabu search: Blending combinatorial optimization and stochastic search for global optimization. *Annals of Operations Research*, **63**, 1996, pp. 153-188.
- [3] Bland J. A., Dawson G. P.: Tabu search and design optimization. *Computer Aided Design*, **23**, 1991, pp. 195-201.
- [4] Charalambous C.: Conjugate gradient algorithm for efficient training of artificial neural networks. *IEEE PROCEEDINGS-G*, **139**, 3, 1992, pp. 301-310.
- [5] Chow T. W. S., Cho S. Y.: *Neural Networks and Computing*. Imperial College Press, 2007.
- [6] Glover F.: Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, **13**, 5, 1986, pp. 533-549.
- [7] Glover F.: Tabu search – Part I. *ORSA Journal on Computing*, **1**, 1989, pp. 190-206.
- [8] Glover F.: Tabu search – Part II. *ORSA Journal on Computing*, **2**, 1990, pp. 4-32.
- [9] Hagan M. T., Menhaj M.: Training feed forward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, **5**, 6, 1994, pp. 989-993.
- [10] Haykin S.: *Neural Networks: A Comprehensive Foundation*. Macmillan, New York, NY, 2007.
- [11] Hedar A. A.: *Studies on Metaheuristics for Continuous Global Optimization Problems*. Ph.D. Thesis, University of Kyoto. Japan, 2004.
- [12] Khanduzi R.: *Automotive Sales Price Forecasting at Iran Khodro Company Using Data Mining and Neural Networks*. M.Sc. Thesis (in Persian), K. N. Toosi University of Technology, Iran, 2009.
- [13] Knox J. E.: *The Application of Tabu Search to the Symmetric Traveling Salesman Problems*. Ph.D. Thesis, University of Colorado. USA, 1989.
- [14] Krose B., Smagt P. V. D.: *An Introduction to Neural Networks*. The University of Amsterdam, 1996.
- [15] McNelis P. D.: *Neural Networks in Finance: Gaining Predictive Edge in the Market*. Elsevier Academic Press, 2005.
- [16] Nelder J. A., Mead R.: A simplex method for function minimization. *The Computer Journal*, **7**, 1965, pp. 308-313.
- [17] Otair M. A., Salameh W. A.: Speeding up back propagation neural networks. *Informing Science and Information Technology Education Conference*, Arizona, USA, 2005, pp. 167-173.
- [18] Pan S., Lai C.: Identification of chaotic systems by neural network with hybrid learning algorithm. *Chaos, Solitons & Fractals*, **37**, 1, 2006, pp. 233-244.
- [19] Pasti R., Castro L. N.: An Immune and gradient-based method to train multi-layer perceptron neural networks. *International Joint Conference on Neural Networks*, Vancouver, Canada, 2006, pp. 2075-2082.
- [20] Peyghami M. R., Khanduzi R.: Predictability and forecasting automotive price based on a hybrid train algorithm of MLP neural network. *Neural Computing and Applications*, **21**, 2012, pp. 125-132.
- [21] Sexton R. S., Alidaee B., Dorsey R. E., Johnson J. D.: Global optimization for artificial neural networks: A tabu search application. *European Journal of Operational Research*, **106**, 1998, pp. 570-584.
- [22] Shang Y., Wah B. W.: Global optimization for neural network training. *IEEE Computer Society Press*, **29**, 3, 1996, pp. 45-54.
- [23] Siddique M. N. H., Tokhi M. O.: Training neural network: back propagation vs genetic algorithms. *International Joint Conference Neural Networks*, **4**, 2001, pp. 2673-2678.
- [24] Skorin-Kapov J.: Tabu search applied to the quadratic assignment problem. *ORSA Journal on Computing*, **2**, 1990, pp. 33-45.
- [25] Thiesing F. M., Vornberger O.: Sales forecasting using neural networks. *International Conference on Neural Network*, **4**, 1997, pp. 2125-2128.

- [26] Wilamowski B. M., Iplikci S., Kaynak O., Efe M. O.: An algorithm for fast convergence in training neural networks. International Joint Conference on Neural Networks, **3**, 2001, pp. 1778-1782.
- [27] Zhang G. P.: Neural Networks in Business Forecasting. IRM Press (an important of Idea Group Inc), 2004.