using mailed standard questionnaires. Then multiple regression exercises were conducted to assess the project data and build a prediction model. The results showed that the overall construction duration of such projects could be determined on the basis of the set of scope criteria, construction method and housing scheme selected.

Similarly, Li et al. (2006) devised a method for predicting potential cost overruns and schedule delays in construction projects. The output of the method was helpful in analyzing the project status at different time horizons and measuring the impact of the performance indicators on the profitability of the job. Jeong et al. (2009) developed an automated contract time determination system for highway projects. They provided a framework for determining the contract time of the projects in Oklahoma, and proposed a stand-alone computer software package that automated the entire procedure. Chao and Chien (2009) presented a neural network model for generating an alternative S-curve estimate to facilitate planning before construction. Chao and Chien (2010) also concentrated on the concept of case-based reasoning and presented a progress-matching method entirely on the basis of matching actual progress so far against S-curves of historical cases and by using similar historical cases for estimation to generate a subsequent S-curve estimate during construction. In short, a model combining the preliminary estimate from the neural network model and the subsequent estimate from the progress-matching method was developed.

Following the studies mentioned above and similar ones, the present study aims to formulate a precise tool of construction industry norms for the overall construction period of projects in the developing countries, particularly Iran. Focusing on the results of the overall construction durations, this paper recommends an effective tool for measuring the construction time of projects and estimating project time in the construction industry. In other words, an intelligent model is proposed based on a locally linear neuro-fuzzy (LLNF) with a tree learning algorithm for construction projects in order to improve decision-making in feasibility studies and performing projects' life cycles. The model which addresses the effects of a number of criteria on project time and schedule by taking the advantages of both fuzzy sets theory (FST) and ANNs can be successfully used for the long term prediction of the time data in the construction industry.

The rest of the paper is organized as follows: in section 2, our intelligent model is introduced. Then the applicability of the proposed neuro-fuzzy model is examined by using the dataset provided in the construction projects in section 3. In addition, the prediction accuracy and the required effort of the proposed model is compared with one of the well-known intelligent models namely BPNN. Finally in section 4, concluding remarks and future research are presented.

2. The Proposed Neuro-Fuzzy Model

Predicting project time data using linear models cannot often be conducted precisely. Indeed, real applications are not amenable to linear prediction techniques as time data applications in the construction industry are complex and nonlinear in nature. Thus, rather than conventional linear prediction models, advanced models such as AI approaches for estimating time series data should be used. In this regard, to overcome the shortcoming of the commonly-used models, this paper concentrates on the time prediction of projects by introducing an intelligent model based on an LLNF with a tree learning algorithm as described below.

2.1. Locally linear Neuro-Fuzzy Model

In this section, the mathematical formulation of a locally linear model (LLM) is presented. The fundamental approach to the LLNF model divides the input space into small subspaces with fuzzy validity functions (Gholipour et al., 2006; Vahdani et al., 2012; Tavakkoli-Moghaddam et al., 2011). Any constructed linear part with its validity function can be regarded as a fuzzy neuron. The overall model is a neuro-fuzzy network with one hidden layer and a linear neuron in the output layer that simply calculates the weighted sum of the outputs of locally linear models (LLMs).

$$\hat{y}_i = \omega_{i0} + \omega_{i1}u_1 + \omega_{i2}u_2 + \ldots + \omega_{ip}u_p,$$
$$\hat{y} = \sum_{i=1}^{M} \hat{y}_i \phi_i(u), \tag{1}$$

This configuration is depicted in Figure 1 where $u = [u_1, u_2, \ldots, u_p]^T$ is the model input and M is the number of neurons.

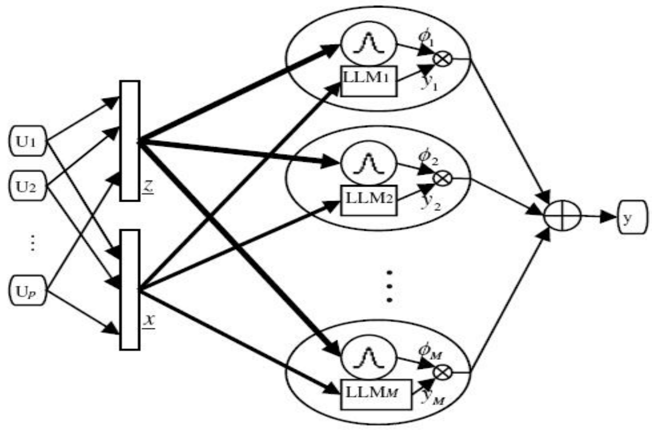

Fig. 1. Topology of a locally linear neuro-fuzzy model (Nelles, 1999)

With regard to the utilized validity function, the interpolation quality is altered. Since normalization is essential for a suitable interpretation of validity functions, the validity function of the normalized Gaussian function is used as given (Abdollahzade et al., 2010; Nelles, 1999).

$$\mu(x) = e^{\left(-\frac{(x-c)^2}{2\sigma^2}\right)} \tag{2}$$

where, c is the center and σ is the standard deviation of the Gaussian function. Thus, we have

$$\phi_i(u) = \frac{\mu_i(u)}{\sum_{j=1}^{M} \mu_j(u)} \tag{3}$$

s.t.

$$\mu_i(u) = e^{\left(-\frac{1}{2}\left(\frac{(u_1-c_{i1})^2}{\sigma_{i1}^2} + \frac{(u_2-c_{i2})^2}{\sigma_{i2}^2} + \ldots + \frac{(u_p-c_{ip})^2}{\sigma_{ip}^2}\right)\right)} = e^{\left(-\frac{(u_1-c_{i1})^2}{2\sigma_{i1}^2}\right)} \cdot e^{\left(-\frac{(u_2-c_{i2})^2}{2\sigma_{i2}^2}\right)} \cdot \ldots \cdot e^{\left(-\frac{(u_p-c_{ip})^2}{2\sigma_{ip}^2}\right)} \tag{4}$$

As demonstrated in Eqs. (3) and (4), the nonlinear hidden layer of the LLNF with parameters (c_{ij}, σ_{ij}) as the parameters of validity functions and the rule-consequent parameters of the LLMs (ω_{ij}) must be adjusted with a learning and an optimization method, respectively. From this point of view, the least squares optimization method is considered for fine-tuning the ω_{ij}, and the c_{ij} and σ_{ij} are reconciled through the locally linear model tree (LOLIMOT) training method as follows. Finally, we have a complete parameter vector:

$$\omega = [\omega_{10}, \omega_{11}, \ldots, \omega_{1p}, \omega_{20}, \omega_{21}, \ldots, \omega_{2p}, \ldots, \omega_{M0}, \omega_{M1}, \ldots, \omega_{Mp}] \tag{5}$$

and an associated regression matrix X for N measured data samples:

$$X = [X_1, X_2, \ldots, X_M] \tag{6}$$

where

$$X_i = \begin{bmatrix} \phi_i(u(1)) & u_1(1)\phi_i(u(1)) & u_2(1)\phi_i(u(1)) & \ldots & u_p(1)\phi_i(u(1)) \\ \phi_i(u(2)) & u_1(2)\phi_i(u(2)) & u_2(2)\phi_i(u(2)) & \ldots & u_p(2)\phi_i(u(2)) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_i(u(N)) & u_1(N)\phi_i(u(N)) & u_2(N)\phi_i(u(N)) & \ldots & u_p(N)\phi_i(u(N)) \end{bmatrix}. \tag{7}$$

Therefore,

$$\hat{y} = X \cdot \hat{\omega} \ ; \hat{\omega} = \left(X^T X + \alpha I\right)^{-1} X^T y, \alpha << 1 \tag{8}$$

where α is a regularization parameter for avoiding any near singularity of matrix $X^T X$ (Sharifie et al., 2006). Utilizing the training method considered in the next section depends on the data sets that must be normalized linearly to the interval [-1, 1].

2.2. Locally linear model tree algorithm

Nelles (1999, 2001) introduced the LOLIMOT algorithm to approximate a nonlinear function with a piecewise linear model. Substituting the weight of outputs layer with a linear function of the network input puts out the LOLIMOT, where each neuron represents the LLM with its related validity function. Dividing the input space by axis-orthogonal splits has made the LOLIMOT learning an additional tree-construction. In each iteration, a new rule or LLM is added to the model. Moreover, utilizing the local weighted least square method for optimizing the corresponding rule consequents and comparing the validity functions associated with the actual dividing of the input space is done. The procedure of the LOLIMOT algorithm is as follows:

1) *Start with an initial model*: A single LLM is selected so that it will be a global linear model over the input space with $\phi_1(u) = 1$. Then, let *M*=1 and consider the initial structure if there is a priori input space dividing.
2) *Find the worst LLM:* Find the worst performing LLM with the mean square error measure.
3) *Check all divisions:* Select the worst LLM as the additional refinement. The hyper rectangle of this LLM is dividing into two halves with an axis orthogonal split. Divisions for every dimension are tried, and for each of the *p* divisions the following steps are taken:
   - Creating the multiple dimensional membership functions for both of hyper rectangles generated.
   - Creating all validity functions: in the parts, only the membership function of the LLM, which is divided, will alter and that of other neurons will not change. But all validity functions must be updated for all LLMs via relation (3).
   - Estimating the rule consequent parameters for recently produced LLMs.
   - Calculating the loss function for the current overall model.
4) *Find the best division:* The best division which is verified in Step 3 is considered for building its related validity functions and LLMs.
5) *Test the termination condition:* If the stopping criteria are reached, stop; otherwise go to Step 2.

The stopping criteria are met with a pre-defined error between the output (y) and the LLNF output with M neuron ($\hat{y}$) (i.e., when the condition: $\|y-\hat{y}\| \le \varepsilon$ is fulfilled). In practice, a pre-defined number of neurons in the LOLIMOT is applied, the error as a function of this number is plotted, and the increase in the number of neurons is reserved until a reasonable performance is attained. Figure 2 shows the operation of the LOLIMOT algorithm in the iterations for a two-dimensional input space.

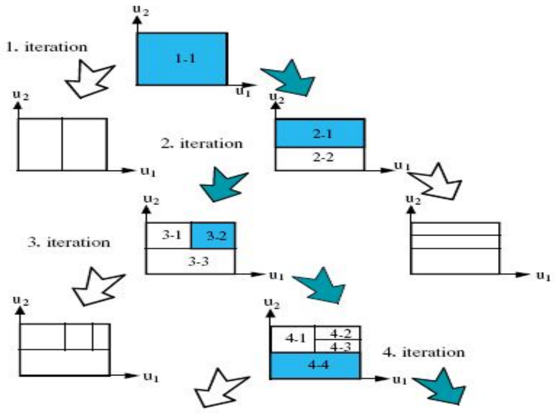

Fig. 2. Iterations of the LOLIMOT algorithm for a two-dimensional input space

3. Model Validation and Comparative Results

In this section, in order to test the effectiveness of the proposed neuro-fuzzy model based on the LOLIMOT algorithm, a data set is employed on the basis of real time data of a construction project in Iran which is presented in (Omrani, 2010).

The experimental data must be divided into two subsets, namely the training data set and the test data set. The data set size must be large enough to provide suitable training and test sets. Thus, the data of the construction project are divided into 120 sections, each of them representing interval periods of the total project completion (i.e., 460 days). In fact, the main activities are separated into 120 tasks in this project. To develop historical data, three sequential periods of the time data set are used as input patterns, with the next used as the output as shown in Table 1. A total of 84 training data points and 36 test data points are provided in the project. Hence, the real data set is divided into training and test data set with the ratio of 70%: 30%.

The basic idea for the data set is based on the concept of project' S-curve in the construction industry (See Table 1). The S-curve graphically illustrates the cumulative progress of the project over time. The shape of the S-curve may indicate that the project progress is typically slow in the beginning and ending periods but faster in the middle when the work intensifies. Thus, first the primary three time data of fed-in subset are fed into the proposed model, the structural risk minimization principle is employed to minimize the training error. Then the one-step ahead estimating time, namely the 4th estimating time is obtained. Afterwards, the next four time data, including three of the fed-in subset data (from 1st to 3th) plus the 4th data in the fed-out subset are similarly fed into the model while the structural risk minimization principle is again employed to minimize the training error. At this point, the one-step ahead estimating time, namely the 5th estimating time is obtained. This prediction procedure is repeated until the 120th estimating time is obtained. Meanwhile, the training error in this training period is also computed.

Table 1
The time data set for the construction project

C	Input patterns			
	Input			Output
	1 1st period	2 2nd period	3 3rd period	4 4th period
		Training data set		
1	7	14	17	15
2	14	17	18	19
3	17	18	20	26
4	18	20	24	31
⋮	⋮	⋮	⋮	⋮
82	321	329	333	336
83	329	333	336	337
84	333	336	337	340
		Test data set		
	336	337	340	341
	337	340	341	343
	340	341	343	346
⋮	⋮	⋮	⋮	⋮
118	445	449	452	454
119	449	452	454	457
120	452	454	457	460

Performance criteria: Prevalent statistical metrics including (1) mean absolute error (MAE), (2) mean squared error (MSE) and (3) mean absolute percentage error (MAPE) are employed to evaluate the estimation performance of the proposed neuro-fuzzy model. These metrics are defined by:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|p_i - \hat{p}_i\right| \tag{9}$$

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(p_i - \hat{p}_i\right)^2 \tag{10}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{p_i - \hat{p}_i}{p_i}\right| \times 100\% \tag{11}$$

where p_i and $\hat{p}_i$ represent the actual and estimated values of the i-th data, respectively. These three metrics are used to measure the deviation between the actual and predicted values.

The p_i and $\hat{p}_i$ representing the actual and estimated values of the time data for the construction project are given in Table 2.

Table 2
Actual and estimated values of the time data for the construction project

	Actual value	Estimated value of the proposed model	Estimated value of BPNN
85	341	342.37	341.07
86	343	346.27	345.15
87	346	347.62	342.00
⋮	⋮	⋮	⋮
118	454	455.29	451.00
119	457	460.92	461.25
120	460	457.00	460.64

The parameters of the well-known intelligent model employed in this case study are given below. The number of neurons in the hidden layer is 2 and the function is $logsig(n) = \frac{1}{(1+\exp(-n))}$. Also, for the proposed neuro-fuzzy model, the number of neurons is 4.

The overall comparative results according to the MAE, MSE and MAPE indices for the proposed model are presented in Table 3.

Table 3
Comparative results according to the prevalent indices

Intelligent models	MAE	MSE	MAPE
BPNN	2.7031	9.8294	0.0066
The proposed neuro-fuzzy model	2.0198	5.2017	0.0050

According to Table 3, the proposed neuro-fuzzy model is placed in the first rank while the BPNN is placed in the second rank. Moreover, Figure 3 compares the results obtained from the prediction results from the proposed neuro-fuzzy model and those from the BPNN by considering the actual time data for the test records (85-120). It is clear that compared to the results of the BPNN model, the results obtained from the proposed model are closer to the actual data.

The results reveal that the proposed model can be implemented in different construction projects to predict their durations in S-curve forms. The shape of the S-curve, which is derived from Figure 3, shows the progress of the construction project. This curve can be used in an earned value analysis and compared with a progress estimate obtained from re-scheduled task times to see if the schedule-based estimate is reasonable.

4. Conclusion and Further Research

Since construction projects often face time delays during their life-cycle, a proactive approach is essential for controlling project time and identifying potential problems. In construction management, time prediction is an indicator that helps project managers recognize potential problems and produce appropriate responses. To meet the needs of project planning in the construction industry, several attempts have been made to provide different time prediction methods and models for estimating projects progress that can supplement the traditional schedule-based prediction methods. Likewise, in this paper a neuro-fuzzy model based on the LLNF with tree learning (LOLIMOT) is introduced to predict the time of construction projects with a higher reliability. The LOLIMOT algorithm used in the model is suitable for the

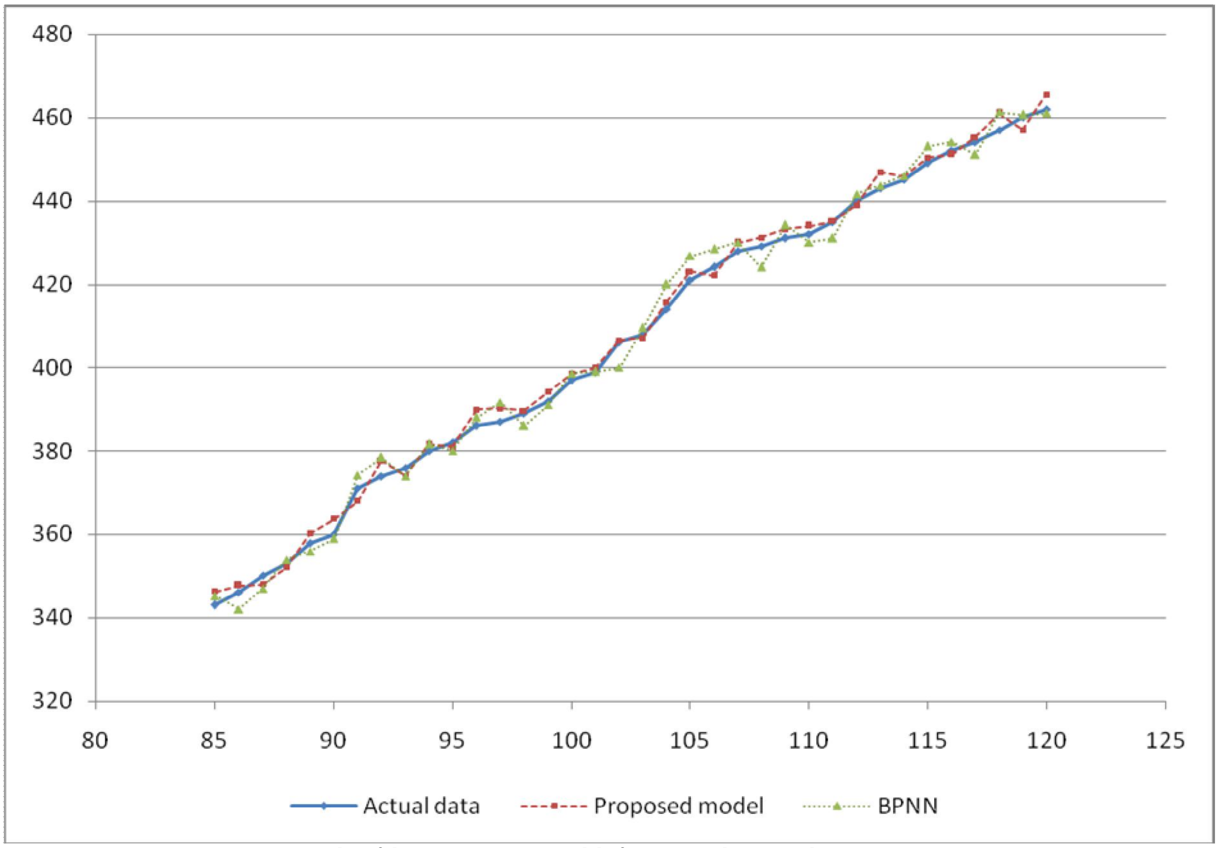

Fig. 3. Results of the two estimation models for time prediction in the construction project

general predictions. Using the real test data set of construction projects in Iran, the proposed model for project time estimation is compared with the BPNN for project time estimation. Through comparison, it is concluded that the proposed neuro-fuzzy model has a better generalization performance and yields a lower estimation error.

It can be suggested that the performance of the proposed neuro-fuzzy model for time prediction may be affected by the value of the input parameter. Thus, further studies may investigate the optimizing input parameter of the proposed model in order to improve computational results of prediction in the construction industry.

Acknowledgement

This paper is partly the result of a research project which was supported by the Industrial Engineering Research Center of Islamic Azad University, Qazvin Branch.

References

Gholipour, A., Araabi, B.N., Lucas, C. (2006) ‘Predicting chaotic time series using neural and neuro fuzzy models: a comparative study’ Neural Process. Lett., Vol. 24, 217–239. 2006.

Griffth, A., Stephenson, P., Watson, P. (2000) ‘Management systems for construction’ New York: Pearson Education Inc.

Boussabaine, A.H. (1996) ‘The use of artificial neural networks in construction management: a review’ Construction Management and Economics, Vol. 14, 427-436.

Boussabaine, A.H. (2001) ‘Neurofuzzy modelling of construction projects’ duration I: principles’ Engineering Construction and Architectural Management, Vol. 8, 104-113. 2001.

Chan, A.P.C., Chan, D.W.M. (2003) ‘A benchmark model for construction duration in public housing developments’ International Journal of Construction Management, Vol. 3, No. 1, 1–14.

Chan, A.P.C., Chan, D.W.M. (2004) ‘Developing a benchmark model for project construction time performance in Hong Kong’ Building and Environment, Vol. 39, 339 – 349.

Chan, A.P.C. (1996) ‘Determinants of project success in the construction industry of Hong Kong. PhD Thesis’ University of South Australia, Australia.

Vahdani, B., Iranmanesh, H., Mousavi, S.M., Abdollahzade, M. (2012) ‘A locally linear neuro-fuzzy model for supplier selection in cosmetics industry, Applied Mathematical Modelling, Vol. 36, 4714-4727.

Chan, A.P.C. (2001) ‘Time cost relationship of public sector projects in Malaysia’ International Journal of Project Management, Vol. 19, No. 4, 223–229.

Walker, D.H.T. (1994) ‘An investigation into factors that determine building construction time performance’ PhD Thesis, Royal Melbourne Institute of Technology, Australia.

Chan, D.W.M., Kumaraswamy, M.M. (1999) ‘Modelling and predicting construction durations in Hong Kong public housing’ Construction Management and Economics, Vol. 17, No. 3, 351–362.

Chan, D.W.M. (1998) 'Modelling construction durations for public housing projects in Hong Kong' PhD Thesis, The University of Hong Kong, Hong Kong.

Deputy of Construction and Development in Tehran Municipality. (2010) Construction project reports. < http://omrani.tehran.ir/ >.

Jeong, H.S., Atreya, S., Oberlender, G.D., Chung, B.Y. (2009) 'Automated contract time determination system for highway projects' Automation in Construction, Vol. 18, 957–965.

Li, J., Moselhi, O., Alkass, S. (2006) 'Forecasting project status by using fuzzy logic, Journal of Construction Engineering and Management' Vol. 132, No. 11, 1193-1202.

Sharifie, J. Lucas, C. Araabi, B.N. (2006) 'Locally linear neurofuzzy modeling and prediction of geomagnetic disturbances based on solar wind conditions' Space Weather, Vol. 4, 1–12.

Chao, L.-C., Chien, C.-F. (2010) 'A model for updating project s-curve by using neural networks and matching progress' Automation in Construction, Vol. 19, 84–91.

Chao, L.C., Chien, C.F. (2009) 'Estimating project S-curves using polynomial function and neural networks' Journal of Construction Engineering and Management-ASCE, Vol. 135, No. 3, 169–177.

Abdollahzade, M., Mahjoob, M.J., Zarringhalam, R., Miranian, A. (2010) 'Locally linear neuro-fuzzy (llnf) electricity price forecasting in deregulated power markets' Int. J. Innov. Comp. Inf. Control, Vol. 6, 4203–4218.

Nelles, O. (1999) 'Nonlinear System Identification with Local Linear Neuro-Fuzzy Models' PhD Thesis, TU Darmstadt, Shaker Verlag, Aachen, Germany.

Nelles, O. (2001) 'Nonlinear system identification' Springer Verlag, Berlin.

Tavakkoli-Moghaddam R., Mousavi, S.M., Hashemi, H., Ghodratnama, A. (2011) 'Predicting the conceptual cost of construction projects: A locally linear neuro-fuzzy model' International Conference on Data Engineering and Internet Technology (DEIT 2011), 15-17 March, Indonesia.

Nkado, R.N. (1992) 'Construction time information system for the building industry' Construction Management and Economics, Vol. 10, No. 6, 489–509.

Ng, S.T., Mak, M.M.Y., Skitmore, R.M., Lam, K.C., Varnam, M. (2001) 'The predictive ability of Bromilow's time-cost model' Construction Management and Economics, Vol. 19, No. 2, 165–73.