17th International Conference on the Applications of Computer
Science and Mathematics in Architecture and Civil Engineering
K. Gürlebeck and C. Könke (eds.)
Weimar, Germany, 12–14 July 2006

COMPARISON OF ANN AND CBR MODELS FOR EARLY COST PREDICTION OF STRUCTURAL SYSTEMS

S. Z. Dogan*, D. Arditi, and H. M. Gunaydin

* *Izmir Institute of Technology*
Gülbahçe Urla 35430 İzmir, Turkey
E-mail: dogan@iit.edu

Keywords: Cost Prediction, Case Based Reasoning, Artificial Neural Networks

Abstract. *Reasonably accurate cost estimation of the structural system is quite desirable at the early stages of the design process of a construction project. However, the numerous interactions among the many cost-variables make the prediction difficult. Artificial neural networks (ANN) and case-based reasoning (CBR) are reported to overcome this difficulty. This paper presents a comparison of CBR and ANN augmented by genetic algorithms (GA) conducted by using spreadsheet simulations. GA was used to determine the optimum weights for the ANN and CBR models. The cost data of twenty-nine actual cases of residential building projects were used as an example application. Two different sets of cases were randomly selected from the data set for training and testing purposes. Prediction rates of 84% in the GA/CBR study and 89% in the GA/ANN study were obtained. The advantages and disadvantages of the two approaches are discussed in the light of the experiments and the findings. It appears that GA/ANN is a more suitable model for this example of cost estimation where the prediction of numerical values is required and only a limited number of cases exist. The integration of GA into CBR and ANN in a spreadsheet format is likely to improve the prediction rates.*

1 INTRODUCTION

Cost-estimating models are very useful for making decisions at the early stages of a building design process. Designers use a number of cost estimating techniques and intuitive judgment by utilizing both their experience and data from previous projects. When little is known about the project's scope, cost estimating depends on the relationship between the basic design variables and the final cost of the project. However, the assessment of the impact of different combinations of various design variables on construction cost is very difficult. Using conventional statistical methods such as regression analysis to identify the cost-governing factors is mostly unsuccessful. Another major disadvantage of the conventional techniques is the requirement that a specific mathematical form be defined for the cost function. Two AI techniques are able to overcome the drawbacks of conventional methods. Current research demonstrated that ANN and CBR can be successfully used in cost estimation [1,2,3,4].

This paper attempts to present transparent models of ANN and CBR in spreadsheet format. A three-layer ANN simulated in a spreadsheet format by Hegazy and Ayed [2] is the first model used for cost-estimation in this study. The second one is the spreadsheet simulation of a CBR model developed by utilizing an Excel-based CBR software [5]. The two Excel templates of these AI techniques are populated with cost data at hand to establish cost estimation models. Then to improve the prediction accuracy of these models, a spreadsheet add-in tool based on GA is used to optimize attribute weights [6]. The GA augmented CBR (GA/CBR) and the GA augmented ANN (GA/ANN) models are compared in light of the cost prediction models developed. Finally, conclusions are drawn.

2 CASE STUDY

The sample data employed for cost estimation in this paper comes from a research report of cost analyses of residential building construction undertaken in Turkey [7]. As a developing country, Turkey experiences rapid population growth, and parallel to this an increasing demand for housing. Residential building construction constitutes 72.8% of the construction market [8]. Eighty-two percent of these buildings are 4–8 storey apartment blocks with reinforced concrete structural systems [8]. The cost of the structural system is an important issue in these circumstances.

The cost of a building consists of several items including the structural system, the walls, the doors and windows, the mechanical system, finishings, etc. The relative weights of these items differ for different projects according to the type and usage of the buildings [9], and as a result, cost estimating requires extensive multi-disciplinary collaboration [10]. It has been observed that the cost of building materials constitutes about 60% of the cost of residential buildings [11]. For multistory reinforced concrete residential apartment buildings however, the structural frame system including the foundations covers about 25% of the total construction cost [12]. The overall cost of a multistory residential building may come down considerably if the structural system is designed efficiently. It is understandable that both architects and structural engineers should exercise maximum care in making design decisions for the structural system. Eight design factors were identified as affecting the unit structural cost of a typical building. These factors include: 1) the total area of the building, 2) the ratio of the typical floor area to the total area of the building, 3) the ratio of the ground floor area to the total area of the building, 4) the number of floors, 5) the type of overhang, 6) the location

of the core, 7) the type of floor structure, 8) the foundation system of the building. These design variables were then used as input parameters to determine the output parameter (i.e., the unit structural cost). First, the basic principles involved in the two models and the simulation procedure are described. Then, with the input and output values defined, relevant data are entered into the Excel-based spreadsheet model. CBR and ANN are compared in the following section. Then conclusions are drawn.

3 SPREADSHEET SIMULATION OF ANN

ANN is among the most popular AI techniques. The fundamental idea behind ANN was originally developed by researchers who were inspired by the functioning mechanism of the human brain. It resembles the human brain in two aspects; (1) the knowledge is acquired by the network through a learning process, and (2) inter-neuron connection strengths known as synaptic weights are used to store the knowledge [13]. ANN consists of an input layer, one or more hidden layers and an output layer. The input layer receives data from outside the network; hidden layers, whose input and output signals remain within the network, extract and remember features and subfeatures in order to generate predictions; and the output layer sends data out of the network. The synaptic weights are trained to contain meaningful information, whereas before training, they are random and have no meaning. There is an extensive literature on ANN and corresponding information can easily be found elsewhere [13, 14, 15, and 16].

In this section, a spreadsheet simulation of ANN was implemented on Microsoft Excel. Many practitioners are familiar with spreadsheet applications. Excel-based simulation of ANN is a simple and transparent approach to ANN modeling. It was adapted from the work of Hegazy and Ayed [2]. The spreadsheet represents a template for an ANN with one hidden-layer that is suitable for most applications [17]. The processing of the template incorporates seven steps, following the widely known back-propagation formulation [14]. The general structure and forward computations of this type of ANN are presented in the following steps:

Step 1. Data organization: As a preliminary stage to ANN modeling, the problem at hand needs to be thoroughly analyzed. Through this process, the independent factors affecting the problem are identified and considered as (N) input parameters represented by nodes at the input layer of the ANN. Similarly, the number of intermediate outputs *(L)* and associated conclusions (O) are represented by nodes at the hidden and output layers, respectively. The relationships between the input layer and hidden layers are denoted by W while the relationships between the hidden layer and the output layer are denoted by W'. Once input and output parameters are identified, their corresponding data are collected from the (P) cases. A schematic illustration of ANN Excel simulation notations of N, L, O, P, W, and W' are shown in Figure 1.

To implement this step, the data are first transformed into numerical values and stored in a data-set that is a matrix of (N+O) columns and (P) rows (Figure 2). The numerical transformation of textual data is done in a continuous or binary manner. The minimum and maximum values of each variable are also identified in this step for use in Step 2.

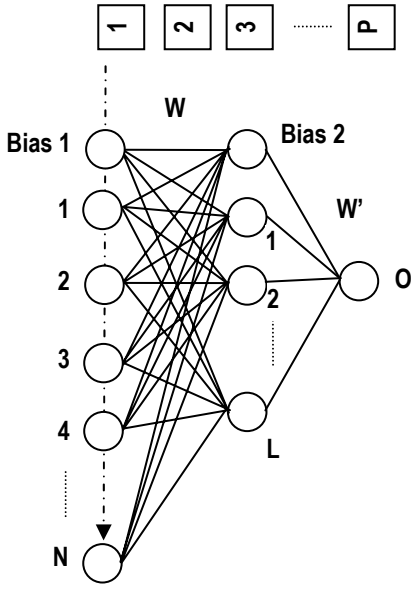

Figure 1. Schematic Illustration of ANN Excel Simulation Notations of N, L, O, P, W, W'

	A	B	C	D	E	F
1	Project No.	Inputs				Outputs
2		1	2	…	N	O
3	1					
4	2					
5	3					
6	⋮					
7	P					
8	Min.:	MIN(B3:B7)				MIN(F3:F7)
9	Max.:	MAX(B3:B7)				MAX(F3:F7)

Figure 2. Schematic Illustration of ANN Excel Simulation Notations of N, L, O, P, W, W'

Step 2. Data scaling: In this step, the values in the input-data matrix (*N* columns by *P* rows) are scaled to fit a range of [-1 to 1] to suit ANN processing. This is done by using the following formula:

$$\text{Scaled Value} = \frac{2\times\left(\text{Unscaled Value} - \text{Column Min}\right)}{\left(\text{Column Max} - \text{Column Min}\right)} - 1 \qquad (1)$$

This scaling formula is written in one cell (B15 for example, in Figure 3), and then copied to all cells in the scaled matrix. To the right of this matrix, a column was added with unit values associated with the bias node, as illustrated in Figure 3.

	A	B	C	D	E	F
⋮						
13	**Project No.**	**Scaled Inputs**				
14		**1**	**2**	…	**N**	**Bias 1**
15	1					1
16	2					1
17	⋮					1
18	P					1

=2*(B3-B$8)/(B$9-B$8)-1
Made once and copied to all cells

Figure 3. Scaling of input values to a range (-1,1)

Step 3. Weight matrix (W): The third step is to construct and initialize the weight matrix (Figure 4). All inputs (1 to *N*) and a bias node were fully connected to the hidden nodes. The number of hidden nodes (*L*) was set as one-half of the total input and output nodes, as heuristically suggested in the literature [17]. All of the values in the weight matrix are considered variables to be determined in the ANN modeling. Hegazy and Ayed [2] suggested that setting the initial weight values to 1 is appropriate for inputs scaled to a range of (-1 to 1).

	A	B	C	D	E	F
⋮						
25	**To Hidden**	**Weights from Inputs & Bias 1**				
26		**I'_1**	**I'_2**	…	**I'_N**	**Bias 1**
27	Node 1					
28	Node 2					
29	Node 3					
30	⋮					
31	Node L					

Cells contain weights values put initially as 1.0s. The matrix elements are set as variables in the optimization.

Figure 4. Weight matrix (W) from (N) inputs to (L) hidden nodes

Step 4. Output of hidden nodes: This step is to allow the hidden nodes to process the input data and produce values to be forwarded to the next layer. According to ANN processing, each hidden node *j* receives activation X_j, which is the sum product of scaled inputs by their associated connection weights. Accordingly, each hidden node produces an output X'_j that is a function of its activation, as follows:

$$X_j = \sum_{i=1}^{N} \left(I_i \times W_{ij}\right) + B_{1j} \times 1.0 \tag{2}$$

$$X'_j = \tanh\left(X_j\right) \tag{3}$$

Experimenting with different activation functions such as linear, logistic, and tanh has shown that the tanh function produces the best results. As shown in Figure 5, a formula was written for the first row of all hidden nodes and then copied to the cells below.

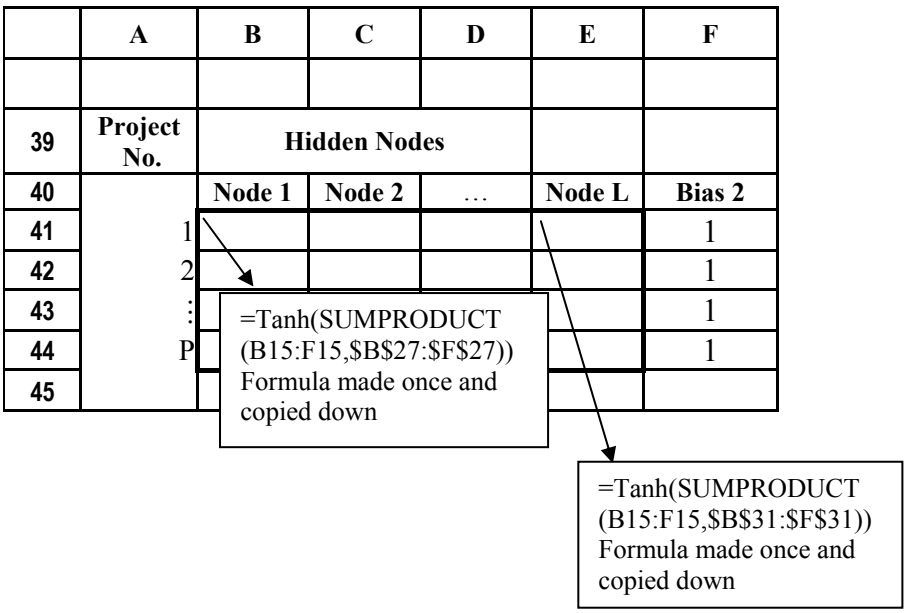

Figure 5. Outputs of hidden nodes

Step 5. Weight matrix (W'): Similar to the weight matrix constructed in Step 3, a second matrix was constructed to connect the (L) hidden and bias nodes to the single output node (Figure 6). These weights are additional variables in the model and were initialized as previously described in Step 3.

Step 6. Final ANN output: Similar to Step 4, the output of the ANN (O) is computed by calculating the sum product (Y) of each hidden node by its connection weight and then processing this value through the tanh function as follows (see Figure 7 for Excel calculations):

$$Y = \sum_{j=1}^{L} \left(X'_j \times w'_{j1} \right) + B_2 \times 1.0 \tag{4}$$

$$O = \tanh(Y) \tag{5}$$

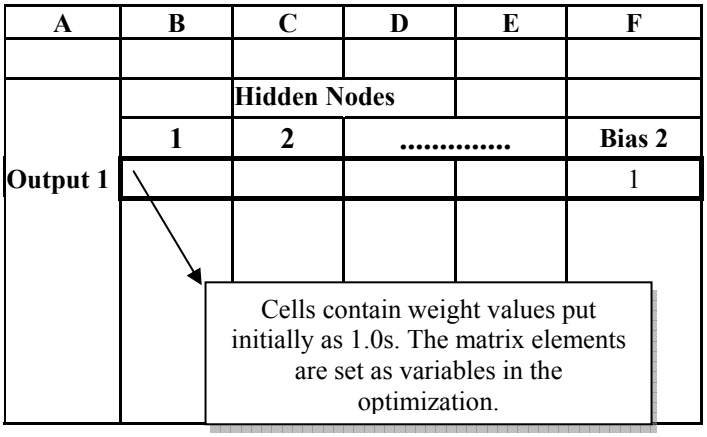

Figure 6. Weights W' from hidden nodes to output nodes

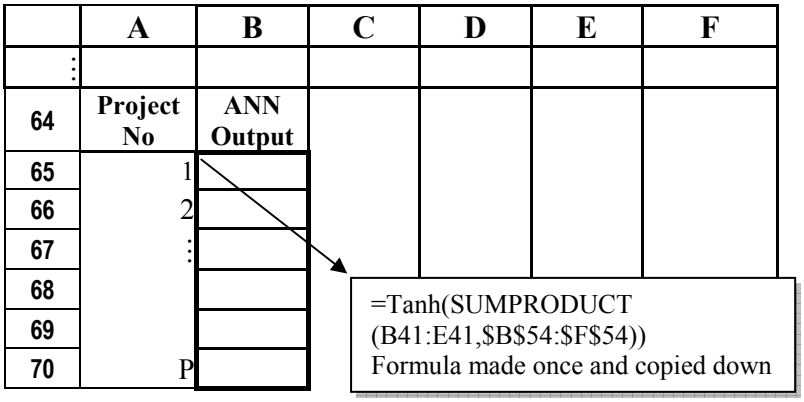

Figure 7. Final ANN Outputs

Step 7. Scaling back the ANN output and calculating the error: In this step, the ANN output (O) is scaled back to the original range of values using the reverse of formula (1) as follows:

$$\text{Output Scaled Back} = \frac{(\text{Output Value}+1)(\text{Max Output}-\text{Min Output})}{2} + \text{Min Output} \qquad (6)$$

To calculate a measure of the ANN performance, a column is constructed in the spreadsheet (see Figure 8) for determining the error between the actual output and ANN output as follows:

$$\text{Estimating Error (\%)} = \frac{(\text{Neural Network Output} - \text{Actual Output})}{\text{Actual Output}} \times 100 \qquad (7)$$

It is customary in ANN simulation to use some cases for training and others for testing. The average error of each batch can be calculated and placed in a different cell and then combined in a cell that determines the overall performance of the ANN. For example:

Weighted Error (%) = 0.5 (Test Set Average Error) + 0.5 (Training Set Average Error) (8)

where weights of 0.5 and 0.5 are assumed for illustration.

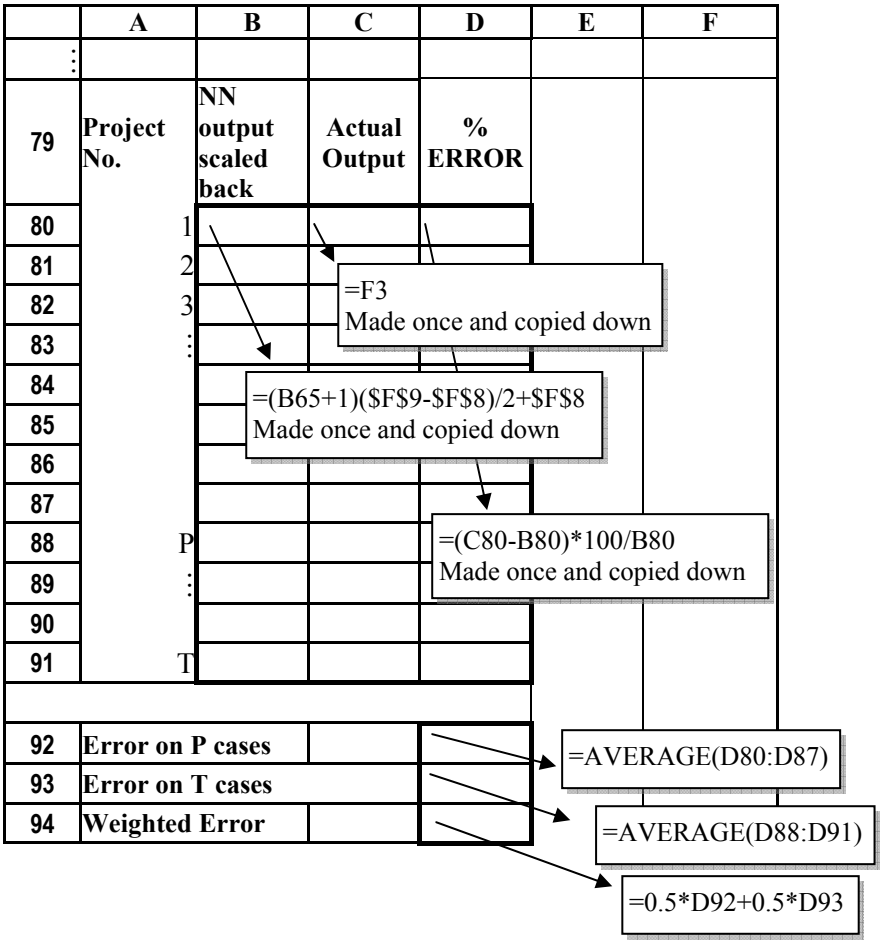

Figure 8. Scaling output back and calculating the error

4 SPREADSHEET SIMULATION OF CBR

Case based reasoning (CBR) involves applying past experiences, in the form of prior cases, to guide current decision making. Therefore, the basic element of a CBR system is the 'case base' [18]. In essence, the cased based reasoner assigns an outcome to a problem based on the outcomes of recent similar prior cases [19]. A case is situation-specific, unlike a rule, which is a unit of generalized knowledge [20]. A case is considered as a set of features, attributes, and relations of a given situation and its associated outcome(s). For a bibliographic categorization and review of CBR research see Aamodt and Plaza [21]; Watson and Marir [22] and Stottler [23].

As a simple and transparent approach to CBR modeling, a spreadsheet simulation of a case-based system was implemented on Excel. This spreadsheet model may present an estimation template for many prediction problems. The processing of the template involves seven steps.

Step 1. Data organization and formatting to a case spreadsheet: The problem at hand is analyzed and the factors affecting the problem are determined. These factors are represented in (F) columns in the spreadsheet (Figure 9). Similarly, the associated output (O) is also represented in a column. Once information about (P) cases is available, this information is entered row by row. The data matrix is therefore composed of (F+O) columns and (P) rows

(Figure 9) (A blank row is reserved above the matrix for weight values, which will be used in Step 5).

A matrix of (R) reference cases is located below the data matrix. This matrix contains all (F) input and (O) outputs associated with these cases. The performance of the CBR model is tested by using these reference cases.

After formatting, it is necessary to add semantics to the data in the form of meanings about the fields (F) and field similarities. To implement this in an Excel spreadsheet, the data are arranged into numerical (N) and textual (C) field values.

1	A	B	C	D	E	F	G	H
2	Weights							
3	**Case No.**	CASE BASE **Factors (Inputs)**						**Output**
4		1	…	N	1	…	C	O
5	1							
6	2							
7	3							
8	:							
9	P							
10								
11	1							
12	:							
13	R							
14								

Figure 9. Data Organization and Formatting

Step 2. Field similarity function for textual symbols: Field similarity functions are used to define how similar the field values are to each other. Field similarities are computed with respect to each reference case in the reference matrix. This computation is done for every case in the casebase.

A specific similarity function for nominal values (textual symbols) (*C*) is defined as follows:

If text f_1 appears exactly in text f_2 or if text f_2 appears exactly in text f_1,

then similarity $(f_1,f_2)= 1$, or else similarity $(f_1,f_2)= 0$. (9)

This similarity formula is written in one cell, and then copied to all celss in the Similarity Matrix in Figure 10 (C columns by P rows).

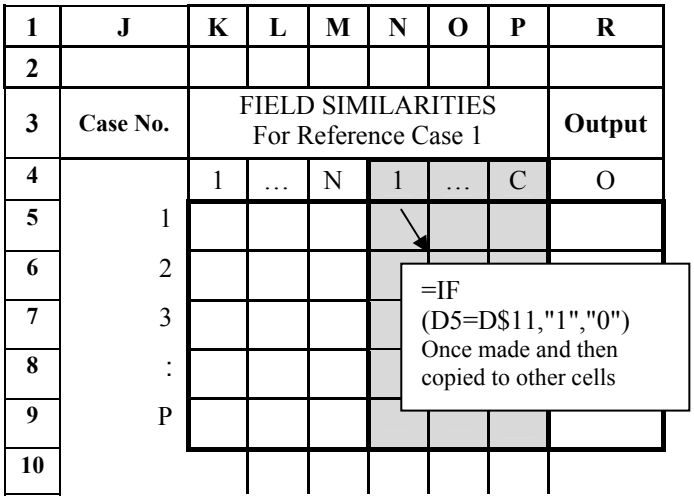

Figure 10. Field Similarity Function for Textual Symbols

Step 3. **Field similarity function for numerical values**: These values denote magnitudes. Accordingly, the similarity measure is as follows:

$$\text{If } f_1 \neq f_2$$

$$\text{then similarity } (f_1,f_2) = \min(|f_1|,|f_2|)/\max(|f_1|,|f_2|), \text{ or else similarity } (f_1,f_2) = 1. \quad (10)$$

This similarity formula is written in one cell, and then copied to all cells in the Similarity Matrix in Figure 11 (*N* columns by *P* rows).

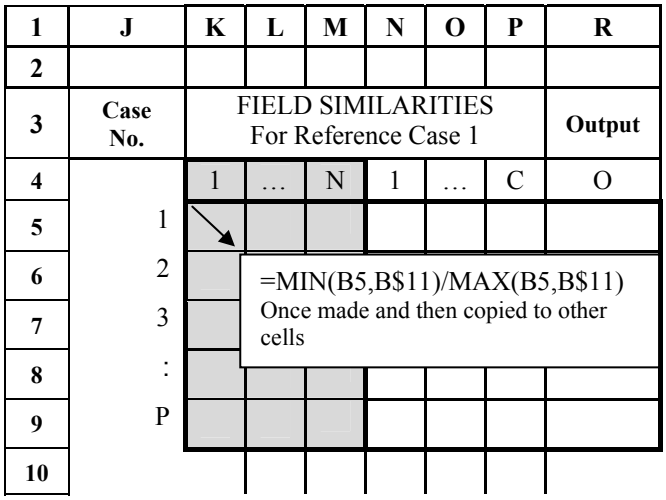

Figure 11. Field Similarity Function for Numerical Values

Step 4. Weight matrix: After all the field similarity values are calculated in an ($F \times P$) matrix, the weight matrix is constructed (see Figure 12) for the computation of matching scores. All of the values in the weight matrix are considered variables to be determined in CBR modeling. A weighted score is calculated from the field weights and field similarity values. For positive weights and for normalized results of field similarities, linear weight scores are always between 0 and 1, with score 1 indicating the case most similar to the reference case and 0 the least.

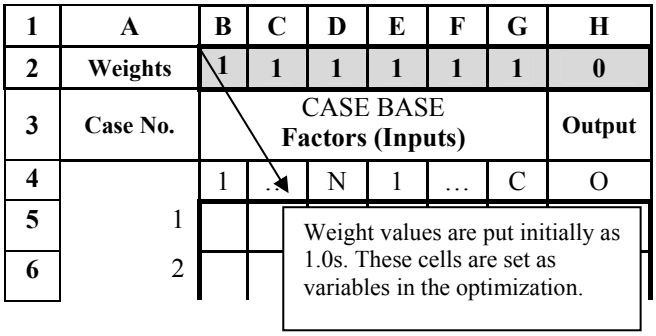

Figure 12. Weight Matrix

Step 5. Case similarity function: Similar to Steps 2 and 3, case similarities are computed with respect to a reference case and this is done for every case in the casebase. Each field value is compared with the corresponding field value in the reference case. The similarity between reference case and a case in the casebase is computed by using a case similarity function that defines linear weighted scores based on the following formula:

$$\text{Similarity (case, reference)} = (\sum w_i \times v_i) / (\sum |w_i|) \qquad (11)$$

where v_i = similarity (r_i, c_i), i.e., the similarity between the reference and the case for field i, and w_i: weight of field i. The formula is made once for cell S5 in Figure 13 and then copied down to the other cells.

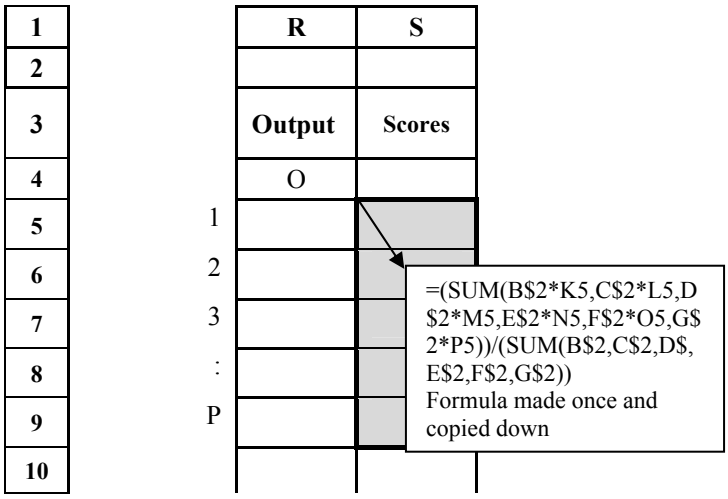

Figure 13. Case Similarity Function

Step 6: **Sorting scores and corresponding outputs**: The highest case similarity score indicates the closest matching case (in the casebase) to the reference case. In order to find the output value of the highest scored case, the scores are sorted by output values (Figure 14).

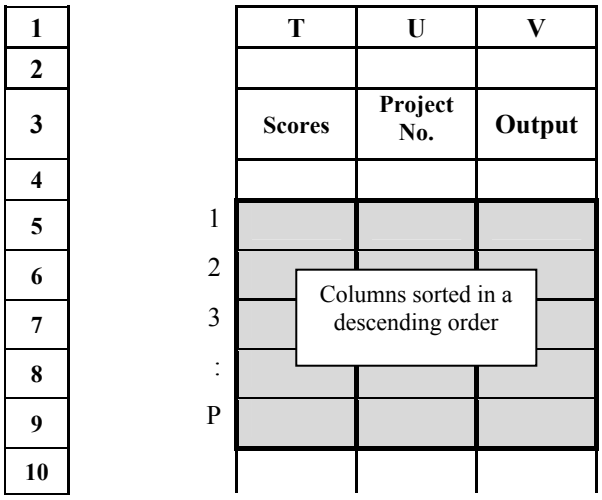

Figure 14. *Sorting Scores and Corresponding Outputs*

Step 7: Highest score and calculating the error: In this step, the CBR output is compared to the actual output of the reference case by the percentage formula (see Figure 15). This step is repeated for all the cases in the reference matrix. Then the average error is calculated for all reference cases.

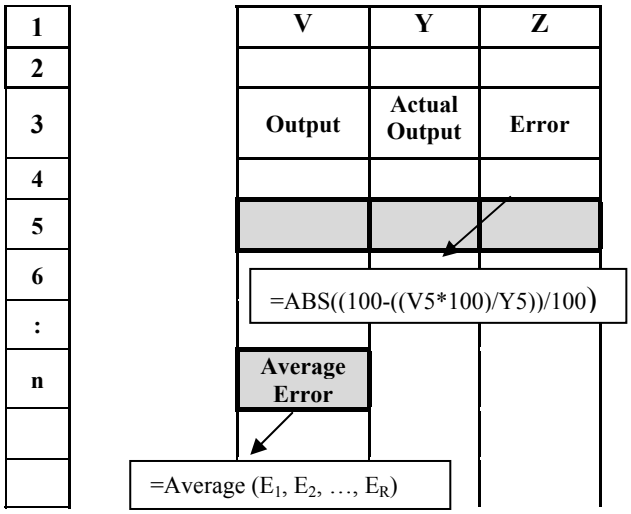

Figure 15. Highest Score and Calculating the Error

5 GENETIC ALGORITHMS

GA uses the method of evolution, specifically survival of the fittest. The theory behind GA is that a population of certain species will adapt to live better in its environment after many generations of random evolution. Thus, GA first creates a population of possible solutions to the problem. Individuals in the population are then allowed to randomly breed, which is called crossover, until the fittest offspring (the one that solves the problem best) is generated. After a large number of generations, a population eventually emerges where the individuals

will provide an optimum solution. The basic principles of GA are well described in many texts [e.g., 24, 25].

For this study, the commercial GA software Evolver was used to find the optimum weights of the ANN and CBR models [6]. Evolver works as an add-in to Microsoft Excel. Weight matrices of ANN and CBR simulations are constructed as variables to be optimized in Excel by Evolver. The ANN weighted error is set to be minimized by GA optimization while CBR score weights are set to be maximized. Determining weight values by GA increased the prediction accuracy in both models compared to previously used other weight generation methods such as gradient descent, feature counting, etc. [26, 4]

6 COST DATA AS AN EXAMPLE APPLICATION

ANN spreadsheet simulation: The ANN Excel template was modified to suit the development of a cost model for residential building projects. First eight input variables and one output variable (i.e., the cost of the structural system per square meter) were defined. The values of the selected variables were extracted from Saner's [7] study. Using the described procedure for simulating ANN on Excel, the data associated with the 29 projects studied by Saner [7] were then entered into Excel. All ranges and matrix dimensions were set according to the number of inputs, the number of outputs, historical examples, test examples, and hidden nodes (i.e., N=8, O=1, P=22, T=7 and L=4, respectively). Evolver was able to come up with an overall weighted error of 11%, with 0.5 weight on the training set and 0.5 weight on the testing set.

CBR spreadsheet simulation: The CBR-Excel template was populated by the same cost data collected from residential building construction projects [7]. Relevant data were also entered into the CBR-Excel model using the procedure described. The dataset of 29 projects was randomly split into an input set containing 24 projects (P=24), and a test set containing 5 projects (R=5). In other words, the CBR-Excel simulation described previously was set as follows: N=4, C=5, O=1, P=24, R=5. The GA-augmented CBR model yielded an average error of 16%.

When GA is used to generate weights in the CBR study, one of the cases in the input casebase is removed and called an evaluation case. The similarities between the attributes of the evaluation case and the corresponding attributes of the remaining cases are calculated by using Equations (8 and 9). Then the case similarities are derived between the evaluation case versus the remaining input cases by taking the average of all attribute similarities. The relationship that governs the similarity of the input case that has an output that is closest to the output of the evaluation case is plugged into the GA algorithm (Evolver) for maximization. In this study, the range of attribute weights was set between 1 and 10, the default population size of 50 was used, and Evolver was run 15,000 times to find the optimum attribute weights that generated the maximum case similarity closest to 1 (100%). Weights generated by Evolver were then plugged into the CBR Excel model manually.

In ANN modeling, the optimization variables are the weights generated in the transition from inputs nodes to hidden nodes, and from hidden nodes to output nodes (Figures 4 and 6). ANN weights are optimized by GA in order to reach the minimum average error value.

7 COMPARISON OF GA/CBR AND GA/ANN

This study has evaluated the performance of GA-augmented ANN (GA/ANN) and GA-augmented CBR (GA/CBR) in predicting the unit cost of a building's structural system. So far, these two techniques were compared based on their prediction accuracy. However, there are other characteristics of these techniques that will have an equal, if not greater impact upon their adoption. The relative merits and shortcomings of these techniques are discussed below. Five characteristics are considered to assess these techniques' utility: preprocessing effort, configurability, explanatory value, accuracy, and improvement potential.

7.1 Preprocessing Effort for Conversion of Data

Data consist of cases and their related features. The content could be both in numerical and textual form. The techniques of handling data for ANN and CBR systems are different. ANN can handle only numerical values, which also need to be scaled to a certain range. Conversions of numerical and textual input data are essential to suit ANN processing. The numerical values are often scaled to a range of [-1, 1] for tanh activation function to avoid fluctuations in the mathematical calculations of the ANN system. Although both CBR and ANN systems require the organization of data into a matrix form to suit the Excel format, ANN needs three additional steps in order to process input data and produce a meaningful output. This certainly adds to the preprocessing effort. Spreadsheet simulations have the advantage of transparency, but they cannot avoid the considerable time required to build them up, when compared with commercial software. Therefore, ANN is at a disadvantage when dealing with a large data set. It is easier to use CBR which handles cases in their original representations, without converting the data from one type to another. This may also be important in order to prevent loss of information since modifications to data may deteriorate learning performance from the level that might have been attained by learning from the original data [27]. In this study, data were in the form of both numerical and textual values. Features expressed as text were used in the CBR study. Textual data were subjected to numerical transformation in a continuous manner in the ANN study; the numerical data were reduced to a range [-1, 1] with a linear scaling formula.

7.2 Configurability in the Spreadsheet Format

Configurability is the measure of how much effort is required to build a prediction system that generates useful results. Considering the preprocessing effort mentioned in Section 7.1, CBR needs relatively little effort to construct. However, model building is a more complex issue than simply entering and converting data. An ANN model requires that the number of hidden layers, the number of hidden neurons, the number of bias nodes, the learning algorithm, and the transfer function be specified by the user, whereas CBR only needs the setting of the feature and case similarity functions. These variables are the tools of modeling, but analysts have to experiment with different combinations of these variables in order to find the optimum combination and result. Most of the books published on ANN modeling agree that the process is largely one of trial and error. Therefore, it takes considerable effort to configure the neural network architecture and doing so certainly requires a fair degree of expertise. For this reason, it is difficult to see how an ANN model could easily be built up within the spreadsheet format by analysts, where the analyst has to manually enter all the values, build up the model, evaluate the performance and then experiment with the model again and again until an optimum solution is reached. The burden of the training process in ANN could be intolerable in a spreadsheet format. CBR, on the other hand, does not require

experimentation with combinations of parameters to establish a prediction model. Since it does not predict from scratch, but retrieves cases from a casebase, it uses simple feature similarity and case similarity formulas, which can be made once in Excel and easily copied to all cells thereafter.

7.3 Accuracy of Cost Prediction

Not generating data from scratch but adjusting from a casebase enhances the configurability of CBR in Excel format, but it appears to be a disadvantage in this particular study because there were only few examples stored in the casebase. Consequently, the ANN model was able to produce cost values that were closer to actual costs than the CBR model. Even though CBR worked with full efficiency and selected the closest cost value, it definitely would never be able to predict better than what exists as the closest cost in its casebase. Although several methods utilizing highest score ranks were applied in order to get closer predictions, none produced better results. If the ANN paradigm is suitable for the data available, a key aspect of many ANN models is that they are able to learn, and their behavior may improve with training and experience [28]. In this case this advantage of ANN provided superior prediction results over CBR.

7.4 Explanatory Value

Although ANN models are great learners, almost like humans, the rules behind their judgment are not explainable. One attraction of the transparent spreadsheet simulations carried out in this study is that the analyst is able to see and control all the formulas and connections being used by the prediction model. However, in ANN, if a particular prediction is in some sense surprising to the analyst, it is harder to establish any rationale for the value generated. It is difficult to evaluate the outcome of an ANN study merely by studying the network architecture and neuron weights. By comparison, CBR appears to offer an advantage in this respect. Unlike reducing the error in ANN by generating weights through back-propagation, CBR estimates by analogy. Cases are ordered in degree of similarity to the target case by utilizing similarity assessment methods calculated by assigning weights to the related features. Indeed, in addition to its explanatory value, this technique encourages the participation of the analyst in getting more accurate predictions.

7.5 Improvement Potential via the Application of GA

Weights are the important adjustable variables that can be freely manipulated on an Excel spreadsheet for accurate predictions. Both in ANN and CBR, the weights of the variables are adjusted in order to build up the optimum prediction model. Therefore, the improvement potential of these models is strongly tied to how realistic the weights of the variables are. In the GA/ANN and GA/CBR studies, the optimization of the weights is done by using genetic algorithms.

When comparing the model building effort of the two systems, it was mentioned that the primary advantage of CBR over ANN was that a CBR application did not need to be trained [29]. Yet in the GA/CBR study, the selection of the weights for similarity assessment turned out to be an important operation, which consumed as much time as the training procedure in ANN. By comparison, the integration of GA into ANN is a simpler procedure, which is carried out only once for the whole training cycle. On the other hand, weight generation in CBR is a critical issue on which the success of CBR heavily relies. The GA optimization for

feature weights in this CBR study was carried out once for each case in the casebase in order to get the most benefit out of the integrated system.

For the study carried out with GA/ANN, the GA optimization for weights was not more successful than the simplex optimization method or back-propagation training [26]. However, GA produced several improvements in the GA/CBR study [4]. GA was able to reduce the effect of less important features; and it was able to eliminate the unimportant features when constraints were scored on a scale starting with 0. This means that if a feature is of no importance, it was assigned a 0 weight by GA. In the study carried out by Dogan et al. [4] it was found out that every feature could somewhat improve the accuracy of the prediction; so the constraints were set to begin from 1. Irrelevant features are also an important problem in ANN models and are investigated lately by Shi [30].

Whatever mechanism is being utilized, it is clear that although accuracy is the most important concern, it is not sufficient to consider the accuracy of prediction systems in isolation. The consistency (explanatory value), continuity (configurability and preprocessing effort) and potential improvement of the systems are also of great importance.

8 CONCLUSIONS

A GA/CBR and GA/ANN model were used by Dogan et al. [4] and Dogan et al.[26], respectively, to predict the structural unit cost of residential building projects. Both models were developed by using the same 29 building project cases. A prediction rate of 84% was obtained from the GA/CBR model, whereas the prediction rate obtained from the GA/ANN model was 89%. A comparison of the experiences with the development of CBR and ANN models shows the following:

GA augmented ANN and CBR models may make better predictions than standard methods provided by commercial software of ANN and CBR [4, 26]. However, in both cases, the model building process is quite cumbersome for Excel simulations. It is even more cumbersome when these systems need to be updated with new cases for long-term use since all the model building process should be repeated and tested with each update. This is the reason why the spreadsheet system needs to be automated. Currently, there is no commercial software that can perform GA/CBR. However Jarmulak et al. [31] reported working GA integration into the CBR software called ReCall [32]. As far as ANN system is concerned, ANN software called NeuroShell [33] supports genetic training.

Even after the release of integrated software, more research should be carried out for different data sets because specific recommendations are needed as to which approach would be more appropriate in what type of domain [for what type of output (numerical, textual, binary, etc...)] or for what type of input data (i.e., ratio of inputs/attributes and training and retrieving case numbers). Such guidelines would be of great help to developers of prediction models.

Predicting the unit cost of the structural system of residential buildings in this study has a number of distinct characteristics compared to other prediction problems. First, the training set was comparatively small. Second, the predictions generally have a higher degree of significance to the analyst. This has the consequence that interaction, or collaboration, between the prediction system and the analyst is of great importance. Allowing the analyst to participate in the prediction process by utilizing spreadsheet simulations may lead to two beneficial effects. First, it may enhance accuracy. Analysts may provide some kind of sanity

check on the systems, while the system allows them to control far more characteristics manually than would be possible by commercial software. Second, it may increase confidence in the prediction. This consideration is also important in order to avoid the situation where end-users reject a prediction system.

In this paper two artificial intelligence techniques augmented by an optimization technique were compared when used to predict the cost of the structural system of residential buildings. These techniques were compared in terms of preprocessing effort, accuracy, explanatory value, configurability, and improvement potential. Despite finding that there are differences in prediction accuracy levels, it is argued that the other characteristics of these techniques may also have an impact upon their adoption. It was found that the explanatory value of predicting by analogy gives CBR an advantage when considering its interaction with the analyst and end-users. It was also found that problems of configuring neural networks tend to counteract their superior performance in terms of accuracy. This preliminary research has shown that the AI techniques used in this study are locally significant but are not generalizable. It is believed that it is important to further investigate these AI methods, particularly to explore under which conditions they are most likely to be effective.

REFERENCES

[1] H. M. Günaydın and S. Z. Doğan, A Neural Network Approach for Early Cost Estimation. *International Journal of Project Management,* **22**(7), 595-602, 2004.

[2] T. Hegazy and A. Ayed, Neural Network Model for Parametric Cost Estimation of Highway Projects. *Journal of Construction Engineering and Management,* **124**(3), 210-218, 1998.

[3] N. J. Yau and J. B. Yang. Case Based Reasoning in Construction Management." *Computer-Aided Civil and Infrastructure Engineering,* **13**, 143-150, 1998.

[4] S. Z. Doğan, D. Arditi and H. M. Günaydın, Determining Attribute Weights in a CBR Model for Early Cost Prediction of Structural Systems. *Journal of Construction Engineering and Management,* Accepted for publication, 2006.

[5] *Induce-It user manual.* Inductive Solutions, Inc., New York, NY, 2000.

[6] *Evolver 4.0 for Excel reference manual*. Palisade Corporation, Newfield, NY, 1998.

[7] C. Saner, A Proposal for Cost Estimation for Structural Systems 4-8 Storey Residential Buildings. M.Sc. Thesis, Istanbul Technical University, 1993.

[8] State Institute of Statistics , Construction Permits for the Year 2003. Available from: www.die.gov.tr/english/SONIST/INSAAT/050903g.htm; last accessed October 2003.

[9] Pulver, H. E, *Construction Estimates and Costs.* McGraw-Hill, NY, 1989.

[10] Y. E. Kalay, Enhancing Multi-disciplinary Collaboration through Semantically Rich Presentation. *Automation in Construction*, 10, 741–755, 2001.

[11] A. O. Olotuah, Recourse to Earth for Low-Cost Housing in Nigeria. *Building and Environment*, **37**, 123–129, 2002.

[12] F. E. Gould and N. E. Joyce, *Construction Project Management.* Prentice Hall, Englewood Cliffs, NJ, 2000.

[13] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan, NY, 1994.

[14] D. E. Rumelhart, G. E. Hinton and J. R. Williams. Learning Representations by Backpropagation Errors. *Nature*, **323**, 533-536, 1986.

[15] L. Fausett, *Fundamentals of Neural Networks*. Prentice Hall, Englewood Cliffs, NJ, 1994.

[16] S. Z. Doğan and H. M. Günaydın, Applications of Artificial Neural Networks and Their Potential Uses for Building Construction Industry: A Review B. Tuncer, S.S. Ozsariyildiz and S. Sariyildiz eds. *9th EuropIA International Conference on E-Activities and Intelligent Support in Design and the Built Environment*, Istanbul, Turkey, 79-89, 2003.

[17] T. Hegazy, O. Moselhi and P. Fazio, Developing Practical Neural Network Application Using Backpropagation. *Journal of Microcomputers in Civil Engineering,* Blackwell Publisher, **9**(2), 145-159, 1994.

[18] J. L. Kolodner, *Case based reasoning.* Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1993.

[19] C. K. Riesbeck and R. C. Schank, *Inside case-based reasoning.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1989.

[20] U. G. Gupta, How Case-Based Reasoning Solves New Problems. *Interfaces*, **24**(6), 110-119, 1994.

[21] A. Aamodt and E. Plaza, Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AICOM,* **7**(1), 39-59, 1994.

[22] I. Watson and F. Marir, Case Based Reasoning: A Categorized Bibliography. *The Knowledge Engineering Review*, **9**(4), 1994, web access: http://www.salford.ac.uk/survey/staff/IWatson/cbrefs.htm.

[23] R. H. Stottler, CBR for Cost and Sales Prediction. *AI Expert*, August, 25-33, 1994.

[24] L. Davis, *Handbook of genetic algorithms*. Van Nostrand Reinhold, New York, NY, 1991.

[25] D. Beasley, D. R. Bull and R.R. Martin, An Overview of Genetic Algorithms: Part 1, Fundamentals. *University Computing*, **15**(2), 58-69, 1993.

[26] S. Z. Doğan, D. Arditi and H. M. Günaydın, GA Augmented ANN for Early Cost Prediction. Unpublished article, 2005.

[27] Y. Reich, Machine Learning Techniques for Civil Engineering Problems. *Microcomputers in Civil Engineering,* **12**(4), 295-310, 1997.

[28] H. Borrow, Connectionism and Neural Networks. *Artificial Intelligence,* M. A. Boden, ed., Academic Press, San Diego, Calif., 135-155, 1996.

[29] K. Kasravi, Understanding Knowledge-Based CAD/CAM. *Computer-Aided Engineering,* **13**(10), 72-78, 1994.

[30] J. J. Shi, Reducing Prediction Error by Transforming Input Data for Neural Networks. *Journal of Computing in Civil Engineering,* **14**(2), 109-116, 2000.

[31] J. Jarmulak, S. Craw and R. Rowe, Genetic Algorithms to Optimize CBR Retrieval. E. Blanzeri and L. Portinale eds. *EWCBR 2000, LNAI 1898*, Springer-Verlag Berlin Heidelberg, 136-147, 2000.

[32] *ReCall CBR Toolkit*. Alice d'ISoft, Gif-sur-Yvette, France, 1993.

[33] *NeuroShell reference manual*. Ward Systems Group Inc., Frederick, MD, 1997.