A Neural Fuzzy System for Soft Computing

Ö. Ciftcioglu
Department of Building Technology
Delft University of Technology
Delft, 2628 CR, The Netherlands
o.ciftcioglu@tudelft.nl

M. S. Bittermann
Department of Building Technology
Delft University of Technology
Delft, 2628 CR, The Netherlands
m.s.bittermann@tudelft.nl

I. S. Sariyildiz
Department of Building Technology
Delft University of Technology
Delft, 2628 CR, The Netherlands
i.s.sariyildiz@tudelft.nl

***Abstract* - An innovative neural fuzzy system is considered for soft computing in design. A neural tree structure is considered with nodes of neuronal type, where Gaussian function plays the role of membership function. The total tree structure effectively works as a fuzzy logic system having system inputs and outputs. In the system, as result of special provisions, the locations of the Gaussian membership functions of non-terminal nodes happen to be unity, so that the system has several desirable features; it represents a fuzzy model maintaining the transparency and effectiveness while dealing with complexity. The research is described in detail and its outstanding merits are pointed out in a framework having transparent fuzzy modeling properties and addressing complexity issues at the same time. A demonstrative application of the model is presented from a demonstrative simple architectural design exercise and the favorable performance for similar applications is highlighted.**

I. INTRODUCTION

Although introduction of fuzzy logic into science more than four decades, due to its inherent limitations, it had to be supported by other paradigms to increase its merits and effectiveness. In this respect, artificial neural networks which are developed essentially afterwards, made an important impact on the application potential of fuzzy logic. The relationship between fuzzy logic and neural networks can be seen as a symbiotic partnership which is beneficial to both sides by jointly increasing their application potential. Such systems are known to be neuro-fuzzy systems and these systems were central to computational intelligence research in the 90s. The essential limitations of a fuzzy logic system are due to the imprecision of (a) the membership function type (b) the number of membership functions (c) the location of a membership function (d) the curse of dimensionality. Introduction of a neural network strategy into a fuzzy system substantially reduces the effects of the source of limitations at the cost of transparency, which is the essential feature of a fuzzy logic system that it is praised for. Because of this, the hype of neuro-fuzzy systems of the 90s has diminished in the new millennium, and the exploration of new avenues in the realm of fuzzy logic became very desirable.

In this respect, neural tree structures introduced at the beginning of the 90s [1-5] together with evolutionary computation can be another important paradigm boosting fuzzy logic concept in order to deal with the complex problems of soft computing.

Based on the views put forward above, in this work, the potentials of neural tree for structuring information is combined with the reasoning process of fuzzy logic to obtain a special type of neural tree which is transparent as well as able to deal with complexity. In other words, the limitations of a fuzzy logic system in a complex environment are substantially circumvented by integrating the domain knowledge into the tree structure and determining the fuzzy membership functions accordingly. In this model meta-knowledge is obtained by evolutionary search in a multidimensional search space.

The paper is organized as follows. In section 2 we describe the structure of a neural tree. In section 3 we present the integration of the complex domain knowledge into a tree structure. This is accomplished by means of a matrix computation known as Analytical Hierarchy Process (AHP) or eigenvector method. Section 4 describes neural tree as an underlying structure of domain knowledge. Section 5 describes the results obtained from the implementation of the model. Section 6 discusses neural tree in a fuzzy perspective and the conclusions

II. NEURAL TREE MODELS

A neural tree is composed of terminal nodes, non-terminal nodes, and weights of connection links between two nodes. The non-terminal nodes represent neural units and the neuron type is an element introducing a non-linearity simulating a neuronal activity. In the present case, this element is a Gaussian function which has several desirable features for the goals of the present study; namely, it is a radial basis function ensuring a solution and the smoothness. At the same time it plays the role of membership function in the tree structure which is considered to be a fuzzy logic system as its outcome is based on fuzzy logic operations and thereby associated reasoning. An instance of a neural tree is shown in Fig. 1.

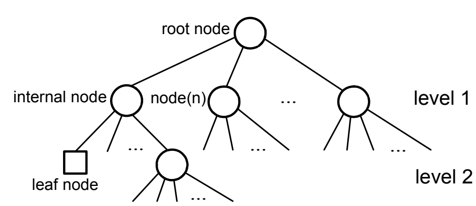

Fig. 1. The structure of a neural tree

Each terminal node, also called *leaf*, is labelled with an element from the terminal set $T=\{x_1, x_2, \ldots, x_n\}$, where x_i is the i-th component of the external input $\boldsymbol{x}$ which is a vector**.** Each link (j,i) represents a directed connection from node j to node i. A value w_{ij} is associated with each link. In a neural tree, the root node is an output unit and the terminal nodes are input units. The node outputs are computed in the same way as computed in a feed-forward neural network. In this way, neural trees can represent a broad class of feed-forward networks that have irregular connectivity and non-strictly layered structures. In particular, in the present work the nodes are similar to those used in a radial basis functions network with the Gaussian basis functions.

III. Analytical Hierarchy Process and the Formation of a Neural Tree Structure

The AHP method is a technique developed by Saaty [6] to compute the priority vector, ranking the relative importance of factors being compared. The only inputs to be supplied by an expert in these procedures are the pair-wise comparisons of relative importance of factors, taken two at a time. This means, in an environment of complex relationships among the variables, one follows the principle of "divide and rule". If we denote the expert input comparing the i-th variable with respect to the j-th variable by $a_{ij} = p_i/p_j$, then the relative importance of the j-th variable with respect to the i-th variable is represented as $1/a_{ij} = p_j/p_i$. Obviously, in an environment with high number of complex relations to make a judicious relational assertion is not easy. However, to make a simple comparison between any two attributes and to make a judgment is much easier for an expert. The $[n \times n]$ matrix obtained by arranging these pair-wise comparison ratios is termed the reciprocal judgment matrix and designated as A. where n is the number of factors subjected to pair-wise comparison. The diagonal elements of A matrix are all unity. Since we take the reciprocals, we have to fill the upper diagonal elements which are altogether $n(n-1)/2$. The details of this technique are given by Saaty [7].

IV. Neural Tree as Underlying Structure Domain Knowledge

In the neural tree considered in this work the output of i-th terminal node is denoted w_i and it is introduced to a non-terminal node. A non-terminal node consists of a Gaussian radial basis function.

$$f(X) = w\phi(\| X - c \|^2) \tag{1}$$

where $\phi(.)$ is the Gaussian basis function, $\boldsymbol{c}$ is the centre of the basis function. The Gaussian is of particular interest and used in this research due to its relevance to fuzzy-logic. The width of the basis function σ is used to measure the uncertainty associated with the node inputs designated as external input X. The output of i-th terminal node w_i is related to X by the relation

$$X_i = w_i w_{ij} \tag{2}$$

where w_{ij} is the weight connecting terminal node i to terminal node j. It connects the output of a basis function to a node in the form of an external input. This is shown in Fig. 2.

The centres of the basis functions are the same as the input weights of that node. Therefore, for a *terminal node connected to a non-terminal node*, we can express the non-terminal node output denoted by O_j, as

$$O_j = \exp(-\frac{1}{2}\sum_i^n \left[\frac{X - w_{ij}}{\sigma_j}\right]^2) \tag{3}$$

which becomes due to (2)

$$O_j = \exp(-\frac{1}{2}\sum_i^n \left[\frac{w_{ij}(w_i - 1)}{\sigma_j}\right]^2) \tag{4}$$

where j is the layer number; i denotes the i-th input to the node; w_i is the degree of membership at the output of the terminal node; w_{ij} is the weight associated with the i-th terminal node and the non-terminal node j.

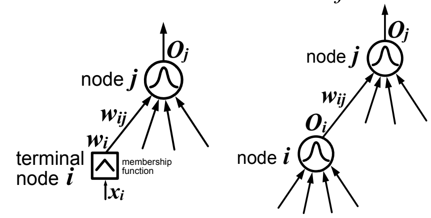

Fig. 2. The detailed structure of a neural tree with respect to different type of node connections

For a *non-terminal node connected to a non-terminal node*, (3) becomes

$$O_j = \exp(-\frac{1}{2}\sum_i^n \left[\frac{w_{ij}O_i - w_{ij}}{\sigma_j}\right]^2) \tag{5}$$

which becomes

$$O_j = \exp(-\frac{1}{2}\sum_i^n \left[\frac{w_{ij}(O_i - 1)}{\sigma_j}\right]^2) \tag{6}$$

We can express (4) and (6) in the following form

$$O_j = \exp(-\frac{1}{2}\sum_i^n \left[\frac{(w_i - 1)}{\sigma_j / w_{ij}}\right]^2) \tag{7}$$

$$O_j = \exp(-\frac{1}{2}\sum_i^n \left[\frac{(O_i - 1)}{\sigma_j / w_{ij}}\right]^2) \tag{8}$$

This implies that the width of the Gaussian is scaled by the input weight w_{ij}. In other words, as to width, the shape of Gaussian fuzzy membership function is dependent on the input weights w_{ij} at the terminals. They are dependent on the neural tree structure and determined by the domain knowledge possibly using the method of AHP. Note that this is a novel type of computation at each node which is slightly different than conventional RBF type computation where the centres are determined by other means, clustering for instance. At the terminal nodes membership functions are not necessarily Gaussian; they can be triangular, among many other types depending on the application. Some membership function types at the terminal node are illustrated in Fig. 3. Note that degree of membership is denoted by w_i for this case. For the input $w_1=1, w_2=1, \ldots, w_n=1$, the radial basis function output at the non-terminal node is also 1; namely, in (7), the centres of the basis functions are given by a vector $\boldsymbol{c} = \{1, 1, 1, \ldots, 1\}$, that is $c_i=1$. This implies that the Gaussian fuzzy membership functions have their maximum value at the point where all w_i

inputs are unity. For a non-terminal node, the same situation is illustrated in Fig. 4.

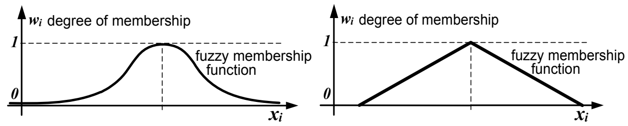

Fig. 3. Two possible fuzzy membership function type among many others, at the terminal node.

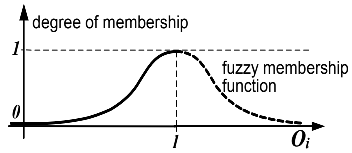

Fig. 4. Fuzzy membership function at non-terminal node.

In this neural tree structure, only the root node performs a simple weighted summation of the inputs coming from the immediate layer below. Terminologically, this is the de-fuzzification process for the final outcome. By means of the above described approach, the locations of the Gaussian membership functions at the non-terminal nodes are well-defined. Furthermore, the following observations are essential.

-Referring to (7), the centre location of the membership functions at the terminal node is always located at the point c_i=*1*. Since w_i is never greater than unity, the right hand side of the Gaussian is represented with broken line. This is indicated in Fig. 4.

- Referring to (8), the centre location of the membership functions at the non-terminal node connected to a non terminal node is always located at the point O_i=*1*. This is indicated in Fig. 4. Since O_j is never greater than unity, the right hand side of the Gaussian is represented with broken line.

- Although at the non-terminal nodes, the type of the fuzzy membership functions are determined as Gaussians, their shape, i.e., the widths, remains to be determined. However, at the terminal nodes, membership functions may be taken other than Gaussian as well as Gaussian.

- The number of Gaussian fuzzy membership functions relevant to a non-terminal node is the same as the number of inputs w_i or O_i to that node. We can consider this differently referring to a multidimensional Gaussian fuzzy membership function. A multidimensional Gaussian membership function is a radial basis function and it can be decomposed into single-dimensional membership functions the number of which is equal to the number of inputs to that node.

- The curse of dimensionality is circumvented since the radial basis function centre of each node is determined as $\mathbf{c} = \{1, 1, 1, \ldots, 1\}$, which is independent of other nodes.

- With the increasing membership function values w_i at the terminal nodes, the output at the root node increases as well. In the fuzzy logic terminology, approaching to the maximum of the fuzzy membership function at the input is reflected to the output of the model; that is with respect to degree of membership w_i, the output of the neural tree follows the same trend at the input.

In the above discussion the shape of the fuzzy membership functions at the non-terminal nodes are Gaussians due to logic operations. Namely, each input to a node has contribution to the output of that node based on logic AND operation. The centre location of the *i*-th Gaussian membership function is selected as w_{ij} due to particular neural tree structure put forward in this research, where the system structure, namely the connection weights connecting the nodes, is established by means of the domain knowledge. This is exemplified in the following architectural design application.

V. Implementation of the Model

The important feature of this concept is the possibility of effective decision-making in a design process, while decision-making on a complex design issue is boiled down a single parameter as *design performance* expressed in fuzzy logic terms. The model is implemented in a design application. The design task is the identification of optimal locations of three high-rise buildings in an architectural design. Fig. 5 shows three towers *T1*, *T2*, and *T3* subject to optimal positioning and the locations of two perceivers *P1* and *P2* viewing the scene. Fig. 6 is the same as Fig. 5 from the viewpoint of observer *P1*. Fig. 7 is the same as Fig. 5 but from the viewpoint of observer *P2*. In the design the distances among the towers and perception locations are considered. To obtain the distances the centres of the building volumes are taken. Next to the distances the visual perception of the buildings from *P1* and *P2* is considered. The perception is obtained using a probabilistic perception theory [8] where the visual attention is modelled as a probability density function (pdf). Integration of the pdf over a certain domain yields perception that becomes a probability. In the computation of the perceptions in this implementation, occlusion is omitted.

In the fuzzy neural model, the knowledge about the performance of the design is represented as follows. The neural tree structure for this case is established as shown in Fig. 8. In the context of the design application the design performance is determined by two sub-domains, namely its *functional* and *perception* aspects at one level below from the root node, that is designated as level 2. At one level further below designated as level 1, there are five sub-domains, namely:

1- *Requirement satisfaction of distances from the perceiver P1*. This accounts for the appropriateness of the distances as to the requirements. The calculation is carried out as logic AND operation through the Gaussian function at the respective node where the inputs stem from the outputs of the preceding lower level, that is, level 0.

2- *Requirement satisfaction of distances from the perceiver P2*. This accounts for the appropriateness of the distances as to the requirements. The calculation is carried out as logic AND operation through the Gaussian function at the respective node where the inputs stem from the outputs of the preceding lower level, that is, level 0.

3- *Requirement satisfaction of distances among the towers*.

This accounts for the appropriateness of the distances as to the requirements. The calculation is carried out in the same way as is the case for the other nodes at this level.

4- *Requirement satisfaction of perceptions by the perceiver P1*. This accounts for the appropriateness of perceptions as to the requirements. The calculation is carried out in the same way as is the case for the other nodes at this level.

5- *Requirement satisfaction of perceptions by the perceiver P2*. This accounts for the appropriateness of perceptions as to the requirements. The calculation is carried out in the same way as is the case for the other nodes at this level.

At the terminal level, the selected determinants of the design performance are given in Table I. These determinants form a multidimensional search space which is complex as to the dimensionality. In this space, Pareto optimality is most desirable for multi criteria based search, which is not the case here, for the sake of simplicity of the demonstrative exercise being described. For the tree structure established, the connection weights at each level assessed by domain experts are given in Table II. The structure can be considered as constitution of domain knowledge, where the connecting weights between the nodes are determined by expert judgement.

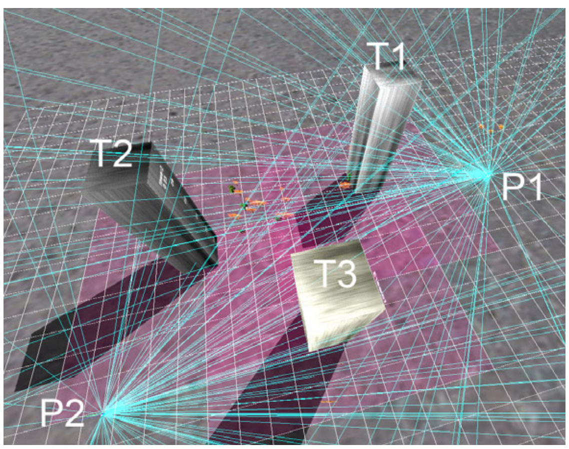

Fig. 5. Three towers *T1*, *T2*, and *T3* subject to optimal positioning and the locations of two perceivers *P1* and *P2* viewing the scene.

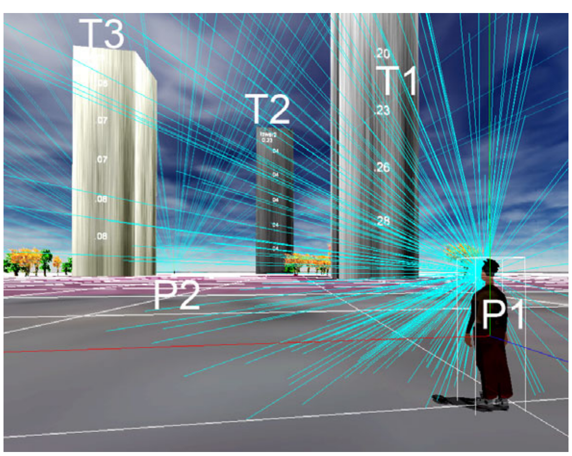

Fig. 6. With reference to figure five, the perception of the scene perceived according to observer *P1*.

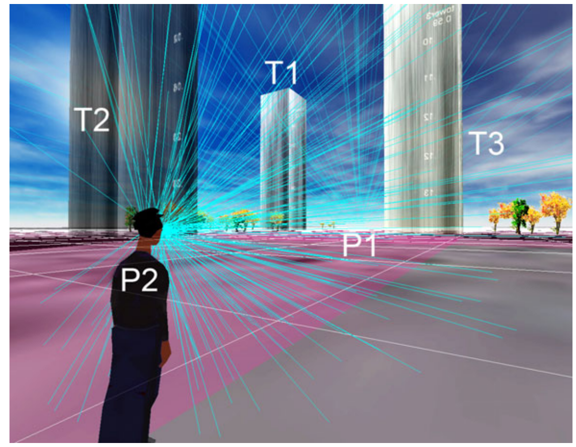

Fig. 7. With reference to figure five, the perception of the scene perceived according to observer *P2*.

TABLE I
DETERMINANTS OF THE DESIGN PERFORMANCE

Distance from P1	*Distance from P2*	*Distance among towers*	*Perception from P1*	*Perception from P2*
Distance between *P1* and *T1*	Distance between *P2* and *T1*	Distance between *T1* and *T2*	Perception of *T1* from *P1*	Perception of *T1* from *P2*
Distance between *P1* and *T2*	Distance between *P2* and *T2*	Distance between *T1* and *T3*	Perception of *T2* from *P1*	Perception of *T2* from *P2*
Distance between *P1* and *T3*	Distance between *P2* and *T3*	Distance between *T2* and *T3*	Perception of *T3* from *P1*	Perception of *T3* from *P2*

TABLE II
WEIGHTS OF THE NEURAL TREE FOR THE DESIGN PERFORMANCE

Level 2	.40	.60													
Level 1	.30	.20	.50	.60	.40										
Level 0	.25	.30	.45	.40	.30	.30	.70	.20	.10	.50	.25	.25	.20	.30	.50

Each aspect is considered in the context of design performance and eventually assessed between zero and unity. This assessment may be the outcome of the method of AHP, in a complex design task. In the demonstrative exercise presented here, AHP becomes a trivial endeavour. The assessments duly made are used as connection weights w_{ij} in the neural tree. After determination of the parameter values in this structure, namely the weights and the individual width of Gaussian at each non-terminal node, a knowledge model is formed. The model should comply with the condition stated as the greater the membership value w_i of an aspect, the greater the design performance. Due to the peculiarity of this structure described in the preceding section, only the left half side of the Gaussians beyond the terminal nodes are used during the computations. This situation makes the structure representing a multivariable increasing function for the whole region beyond the terminal nodes. This ensures that greater membership value w_i of an aspect at the input to a radial basis function yields greater node output. Note that the model is completely knowledge-driven and highly non-linear due to the Gaussians at least at the non-terminal nodes and fuzzy membership functions at the terminals.

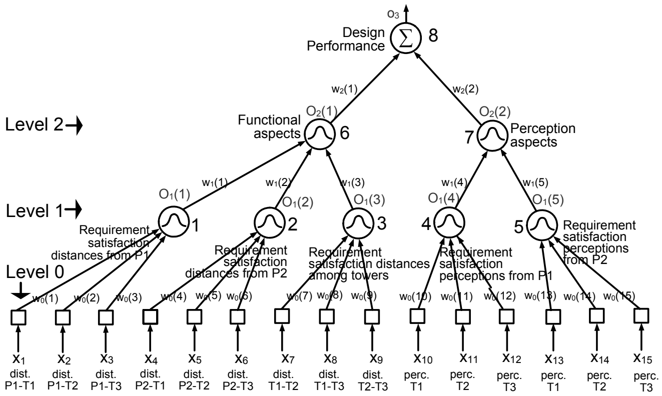

Fig. 8. Neural tree structure for assessment of design performance.

The membership functions at the terminal nodes are application dependent, and therefore their shapes and locations are determined accordingly. The membership functions used in the present case are shown in Fig. 9. The shapes are selected by domain experts. Explicitly, the fuzzy functions are the representations of the requirement specifications of the design.

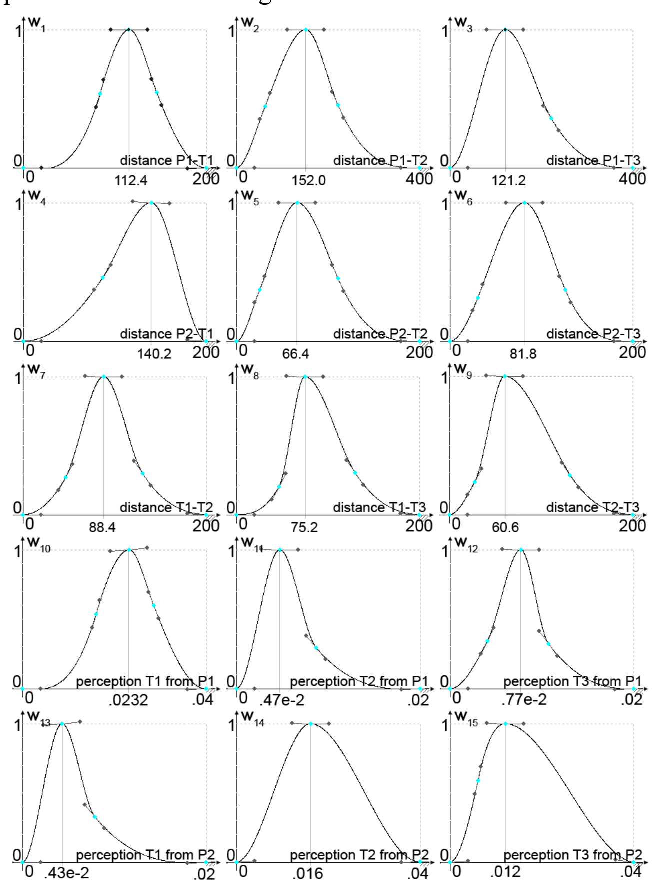

Fig. 9. Membership functions at the terminal nodes.

As far as non-terminal nodes are concerned the widths of the Gaussians are still to be determined and they are obtained by means of the *consistency condition,* which serves as boundary condition for the neural tree model. This is explained below.

The neural tree output follows the trend of input w_i representing the degree of membership associated. Considering this property, the *consistency* refers to the fact that, in the knowledge domain if all the inputs w_i are unity, all system determinants have the value where the associated fuzzy membership functions at the terminal node take the value of 1; as result of this, all the non-terminal node outputs are accordingly 1 and therefore system output at the root node is also 1. This condition is inherently satisfied in the present neural tree structure and this is easily seen by (7) and (8); namely if all w_i are 1, then all non-terminal node outputs O_i are 1 and then the neural tree output is 1. This is more explicitly explained by the following example. While the research is carried out in a department of architecture, an example from architectural domain is more relevant. If all the design determinants belong to by all means modern type of architectural design then, the final design output belongs to also a modern type of architecture and the neural tree output is high. The reverse of this situation state that, if all the design determinants belong to by no means modern type of architecture then, the final design output does not belongs to a modern type of architecture meaning that output vanishes. This latter condition cannot be strictly satisfied since the Gaussians extend to ± infinity and therefore gave still some value even the inputs at the terminal node w_i vanish. Because of this very reason any non-terminal node output O_i theoretically never vanishes but may take sufficiently small values.

Following the above example, the case which can be described by taking all the input determinants as, say 0.5 would yield the neural tree output also as 0.5. Note that, this does not mean the system is linear. On the contrary, the system is highly non-linear. However, the consistency condition as given above is stipulated on it. This imposition is accomplished as described below. In the formation of the modelling the domain knowledge, the system determinants selected should be carefully verified in advance that they observe this stipulation designed as *consistency condition*. In general, the consistency condition is a kind of *boundary condition* which should be satisfied by the fuzzy knowledge model represented by the neural tree structure. The consistency condition as boundary condition is application dependent and the condition or possibly a set of conditions should be imposed on the knowledge model. Therefore, care has to be exercised that the problem formulation is carried appropriately so that the consistency is inherently present in this formulation. Peculiar to the application being presented, the consistency condition is a set of multi-input single-output data as given in Table III and Table IV, respectively. The imposition of the consistency or boundary conditions can be carried out by adaptive or genetic learning. As result of the learning process, the width of each individual Gaussian at each non-terminal node is established. In this way, the cascade feed-forward fuzzy logic operations are clearly defined exhibiting features of transparency in the model. Although the input/output data set given in Tables III and IV is seemingly simple, imposition of this simple data set on the highly non-linear fuzzy knowledge model requires adaptive or genetic learning. The approximation error for this data set is relatively higher for the lower input/output pairs.

TABLE III
DATASET AT THE NEURAL TREE INPUT FOR CONSISTENCY CONDITION

.1	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1
.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2	.2
.3	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3
.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4
.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5
.6	.6	.6	.6	.6	.6	.6	.6	.6	.6	.6	.6	.6	.6	.6
.7	.7	.7	.7	.7	.7	.7	.7	.7	.7	.7	.7	.7	.7	.7
.8	.8	.8	.8	.8	.8	.8	.8	.8	.8	.8	.8	.8	.8	.8
.9	.9	.9	.9	.9	.9	.9	.9	.9	.9	.9	.9	.9	.9	.9

TABLE IV
DATASET AT THE NEURAL TREE OUTPUT FOR CONSISTENCY CONDITION

.1	.2	.3	.4	.5	.6	.7	.8	.9

The fuzzy neural tree having been established the design task is to maximize the output at the root node by identifying optimal location of the buildings. This is accomplished by genetic search. The output at the root node, which quantifies the design performance, is used as the representation of the fitness of the respective chromosome. In this way the genetic algorithm makes use of the knowledge embedded in the neural tree during its search for obtaining maximal performance. The boundary of the space for the locations of the towers is given in Table V, namely the minimal and maximal x and z coordinates for the positions of *T1*, *T2*, and *T3* as illustrated in Figs. 10 and 11.

TABLE V
SOLUTION SPACE

Object	x_{min}	x_{max}	z_{min}	z_{max}
T1	40	140	20	120
T2	20	120	100	200
T3	0	80	50	150

Fig. 10 shows a design at the beginning of the genetic search process, which is a random configuration. It has a design performance of *0.497*, which is the output value at the root node of the tree. The design obtained after genetic search is shown in Fig. 11. It has a performance of *0.997*.

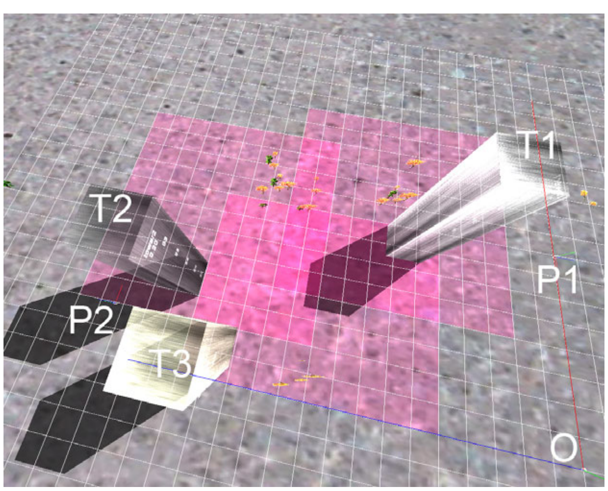

Fig. 10. Illustration of a design with a design performance of .497 at the beginning of the genetic search process; *O* indicates the origin.

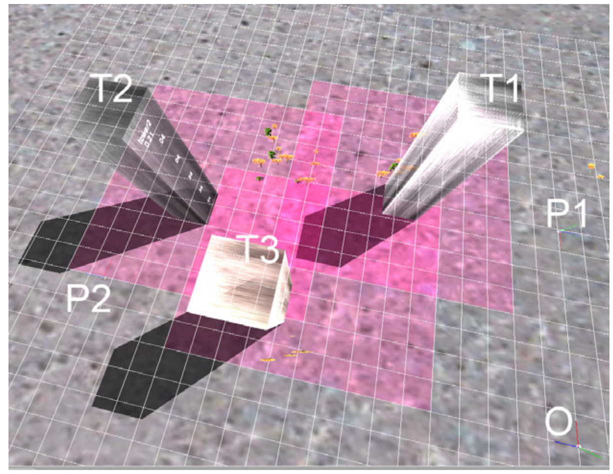

Fig. 11. Illustration of the design result obtained after genetic search with a design performance of 0.997; *O* indicates the origin.

The result of the search process is given in Table VI, where the second column is the best fitness that occurred during the search, and the third column is the average fitness of the respective generation. The resulting design parameters as location of the towers are given in Table VII. The outputs of the tree nodes for the designs shown in Figs. 10 and 11 are given in Table VIII for comparison. The results indicate that the combination of fuzzy neural tree and genetic algorithm is able to identify designs with maximal design performance, while insight into the contributions of the model constituents is provided due to the transparency of the approach. This is seen in Table VIII.

TABLE VI
GENETIC SEARCH PROCESS RESULTS

Generation	Best fitness	Average fitness
1	0.596	0.291
2	0.596	0.323
3	0.596	0.394
4	0.705	0.438
5	0.937	0.465
6	0.937	0.468
7	0.996	0.516
8	0.997	0.529
9	0.997	0.542
10	0.997	0.561

TABLE VII
RESULTING DESIGN FROM GENETIC SEARCH, SHOWN IN FIG. 11

Object	x	z
T1	78.89	68.03
T2	58.02	156.71
T3	25.72	115.04

TABLE VIII
NODE OUTPUTS BELONGING TO THE DESIGNS SHOWN IN FIGS. 10 AND 11.

Node	Out-put	Initial design shown in Fig. 10	Final design shown in Fig. 11
Design Performance	O_3	.497	.997
Functional aspects	$O_2(1)$	.173	.991
Perception aspects	$O_2(2)$	.712	.999
Distance from *P1*	$O_1(1)$	.987	.999
Distance from *P2*	$O_1(2)$	.923	.996

Distance among towers	$O_1(3)$	.777	.984
Perception from *P1*	$O_1(4)$	.907	.996
Perception from *P2*	$O_1(5)$	.485	.973
Distance *P1-T1*	w_1	.999	.999
Distance *P1-T2*	w_2	.962	.984
Distance *P1-T3*	w_3	.870	.990
Distance *P2-T1*	w_4	.972	.980
Distance *P2-T2*	w_5	.707	.923
Distance *P2-T3*	w_6	.731	.976
Distance *T1-T2*	w_7	.933	.994
Distance *T1-T3*	w_8	.857	.947
Distance *T2-T3*	w_9	.351	.817
Perception of *T1* from *P1*	w_{10}	.998	.999
Perception of *T2* from *P1*	w_{11}	.948	.983
Perception of *T3* from *P1*	w_{12}	.424	.889
Perception of *T1* from *P2*	w_{13}	.895	.931
Perception of *T2* from *P2*	w_{14}	.005	.742
Perception of *T3* from *P2*	w_{15}	.441	.977

VI. DISCUSSION AND CONCLUSIONS

The knowledge driven fuzzy modelling is described for performance analysis of an architectural design where the model has neural tree structure. Depending on the complexity of the structure, the method of analytical hierarchy process can be used during the constitution of the structure. In this feed-forward structure output of a node is obtained with fuzzy logic operations using the inputs of this node. This is accomplished by Gaussian membership functions. The model is finally determined by learning where learning refers to the integration of the intrinsic conditions stipulated by the knowledge being modelled. It is noteworthy to mention, that the nodes of the neural tree correspond to fuzzy logic rules so that the outcome of the model is result of a number of logic operations and finally defuzzification at the root node. The equivalence between neural networks and fuzzy logic for Gaussian fuzzy membership functions is known in the literature [9, 10]. In the neural tree with fuzzy logic presented in this research forms a fuzzy model especially as described by Murray [11], where some strict conditions stipulated on the equivalency earlier are relaxed. This implies that, neural tree structures provide additional possibilities to fuzzy logic systems enhancing their transparency and soft computing possibilities for dealing with soft issues, as they are meant to. Integration of evolutionary algorithms into such studies opens new avenues for the effectiveness of the neuro-fuzzy applications. It is emphasized that, the consistency condition introduced in this research is application dependent in general. The peculiarities of a particular application beyond the knowledge-base associated with the application can be embedded in the knowledge model in a natural way in the form of boundary condition.

The application in architectural design is reported indicating the suitability of the work for a wide range of similar applications of technological, industrial and practical interest. In such applications, after the initial establishment of the constitutional knowledge model there are essential search issues to obtain the final form of the model where all the model parameters are determined. The search issues are best carried out by evolutionary algorithms due to the complexity of such a model, in general. The novel research is described in detail and its significant merits are pointed out in a fuzzy framework having transparent fuzzy modelling properties and addressing complexity issues at the same time.

REFERENCES

[1] A. Sankar and R. J. Mammone, "Neural tree networks," in *Neural Networks:Theory and Applications*, R.J. Mammone and Y. Zeevi, Eds, New York: Academic, 1991, pp.281 –302

[2] G. L. Foresti and C. Micheloni, "Generaliuzed neural trees for pattern classification," *IEEE Trans. Neural Networks*, vol. 13, pp. 923-933, May 1992

[3] J. A. Sirat and J. P. Nadal, "Neural tree networks," *Network*, Vol.1, pp.423-438, 1990

[4] G. L. Foresti and C. Micheloni, "Generalized neural trees for pattern classification," *IEEE Trans. Neural Networks*, vol.13, pp.1540-1547, 2002.

[5] F. D. 'Alche-Buc, D. Zwierski, and J. P. Nadal, "Trio learning: a new strategy for building hybrid neural trees," *Int. J Syst.,* vol. 5, no. 4, pp. 259-274, 1994.

[6] T. L. Saaty, *The Analytic Hierarchy Process*, McGraw-Hill, New York, 1980.

[7] T. L. Saaty, *The Brain: Unraveling the mystery of how it works*, RWS Publications, Pittsburgh, PA 15213, 2000.

[8] Ö. Ciftcioglu, M. S. Bittermann, and I. S. Sariyildiz, "Towards computer-based perception by modeling visual perception: a probabilistic theory," *Proc. 2006 IEEE Int. Conf. on Systems, Man, and Cybernetics*, Taipei, Taiwan, pp. 5152-5159, 2006.

[9] J. S. R. Jang and C. T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems", *IEEE Trans. Neural Networks*, vol. 4, pp. 156-158, 1993.

[10] H. X. Li and C. L. P. Chen, "The equivalence between fuzzy logic systems and feedforward neural networks", *IEEE Trans. Neural Networks*, vol.11, no.2, March 2000

[11] K. J. Hunt, R. Haas and R. Murray-Smith, "Extending the functional equivalence of radial basis function networks and fuzzy inference systems", *IEEE Trans. Neural Networks*, vol. 7, no. 3, May 1996.