Building Performance Analysis Supported by GA

Özer Ciftcioglu, *Senior Member, IEEE*, I. Sevil Sariyildiz and Michael S. Bittermann

***Abstract*— A neural tree structure is considered with nodes of neuronal type which is a Gaussian function and it plays the role of membership function. The total tree structure effectively works as a fuzzy logic system having system inputs and outputs. In this system the locations of the Gaussian membership functions of non-terminal nodes are unity so that the system has several desirable features and 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 modelling properties and addressing complexity issues at the same time. A demonstrative real-life application of this model is presented and the favourable performance for similar applications is highlighted.**

***Index Terms*— Neural tree, fuzzy logic, analytical hierarchy process, knowledge model,**

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. Added to these there is another limitation known as *the curse of dimensionality* which occurs because the number of rules increases exponentially as the number of input variables increases [1]. 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 [2-6] together with the evolutionary computation can be another important paradigm boosting the fuzzy logic concept in order to deal with the complex problems of soft computing.

Manuscript received March 31st 2007. Authors are with Delft University of Technology, Faculty of Architecture, Dept. of Building Technology, Technical Design and Informatics (e-mails: o.ciftcioglu@tudelft.nl; m.s.bitterman@tudelft.nl; i.s.sariyildiz@tudelft.nl).

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 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 reports the results obtained from the implementations of the model. Section 6 discusses neural tree in a fuzzy perspective. This is followed by 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. Each terminal node, also called *leaf*, is labelled with an element from the terminal set $T=\{x_1, x_2, \dots, x_n\}$, 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-

1-4244-1340-0/07$25.00

forward neural network. In particular, in the present work the nodes are similar to those used in a radial basis functions network with the Gaussian basis functions. In this way, neural trees can represent a broad class of feed-forward networks that have irregular connectivity and non-strictly layered structures.

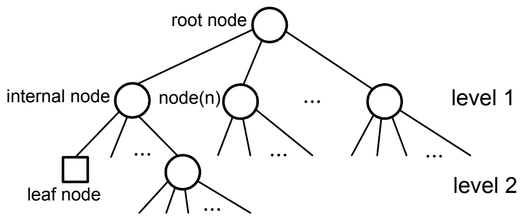

Fig. 1. The structure of a neural tree

III. Analytical hierarchy process and the formation of a neural tree structure

A. Analytical hierarchy process in brief

The AHP method is a technique developed by Saaty [7] 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 [8].

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.

$$f(X) = w\phi(\| X - c \|^2) \quad (1)$$

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

$$X_i = w_i w_{ij} \quad (2)$$

where w_{ij} is the weight connecting terminal node i to non-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.

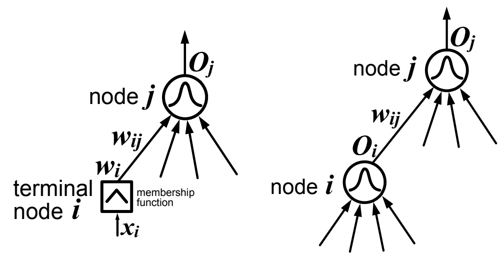

Fig. 2. The detailed structure of a neural tree with respect to different type of node connections

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

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

which becomes due to (2)

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

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

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

which becomes

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

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

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

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

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 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 figure 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 $\mathbf{c} =\{1, 1, 1,\ldots,1\}$, 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 figure 4. 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. By means of this, the imprecision about the locations of the fuzzy membership functions is circumvented.

- 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 $O_j=1$. 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.

- 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.

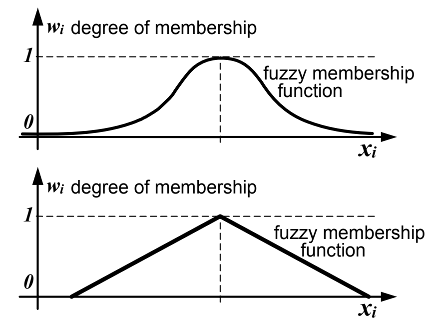

Fig. 3. Two possible fuzzy membership function type among many others, at the terminal node.

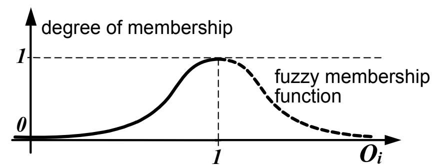

Fig. 4. Fuzzy membership function at non-terminal node.

In 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 are established by means of the domain knowledge. This is accomplished by means of the AHP method. The AHP method delivers it results which are context dependent. In the utilisation with a neural tree it should have the context which is determined by the context of the root node and this applies all the levels throughout the tree structure, including the terminal level. This is exemplified in the following real-life industrial application, where the results spun off from this research is currently in use.

V. Implementation of the Model

For the implementation of the novel neural tree structure presented by this research, an industrial application concerns the building performance analysis of dwellings disposed by the governmental organisation known as *Building Management* acting on behalf of the municipality of Utrecht, in the Netherlands. The organisation is located in Utrecht whose mission is supervision of some infirmaries for elderly people and gives advice to constructors for their renovation/transformation and/or reconstruction. The project is carried out on behalf of Technical University Delft in the Netherlands and it is accomplished and tendered in January 2007. The application is related to a building technological parameter which is known as *transformation capacity (TC)* [9]. In basic terms, it is defined as the potentiality of a building for transformation for its sustained utilization rather than its demolition. This is directly related to the sustainability of the building with extended life cycle. The important feature of this concept is the possibility of effective decision-making in a construction process while decision-making on a complex building technological issue is boiled down to a single parameter as TC expressed in fuzzy logic terms. In the model, the total knowledge about the sustainability of a given pilot building is represented as follows. This pilot building is a typical representation of a group of about 400 houses, which form a certain category. It may be of value to emphasize that the question is extremely important since it is a decision about a building concerning its further existence. To make such a tough decision for a number of houses is another challenge which concerns a number of experts working on the question for some time measured in units of years. Therefore, neural tree approach presented here is very appealing as it provides the user with effective and efficient decision making about a categorical group of houses virtually in real-time, provided once the neural tree is established for this very group. Another point worth to emphasize is that, the outcomes within the neural tree is a result of logical feed-forward rule chaining operations based on the domain expert knowledge embedded in the neural tree. Here, the properties of AHP are important for this operation. Consequently, the neural tree operates as an inference engine which underlies the final outcome.

After carefully studying this pilot building, the neural tree structure for this case is established as shown in Fig. 5. In the context of sustainability, the transformation capacity is determined by two sub-domains, namely its *spatial transformation* and *technical transformation* aspects at one level below from the root node. At one level further below we identify four sub-domains, namely *dimensioning*, *positioning*, *disassembly potential*, *capacity potential*. At the terminal level, the determinants of the transformation capacity at the output take place which are given in Table I. For the structure established above, at each level the connection weights assessed by domain experts are given in Table II. This structure can be considered as a model of constitutional domain knowledge where the connecting weights between the nodes are determined by the AHP. Each aspect is considered in the context of sustainability and based on the AHP process, after normalisation, eventually assessed between zero and unity. These assessments are used as connection weights w_{ij} in the neural tree. The model should comply with the condition stated as greater the membership value, i.e., the membership value w_i of an aspect greater the sustainability. Since only the quantities between zero and unity are involved, only the left half side of the Gaussians beyond the terminal nodes are used during the computations. This situation makes the structure a multivariable increasing function for the whole region in use for computation. This ensures that greater the membership value w_i of an aspect at the input to a radial basis function greater the sustainability.

TABLE I
DETERMINANTS OF THE BUILDING PERFORMANCE

level 1	***Dimensioning***	***Positioning***	***Disassembly***	***Capacity***
First leaf node of the nodes at level 1	minimum dimensions at building level	corridor width/ load-bearing structure's position	main installation network	installation duct capacity
second leaf node of the nodes at level 1	load-bearing structure: axel separation	corridor width/ Installations' position	distribution network	load-bearing structure's capacity for perforation
third leaf node of the nodes at level 1	load-bearing structure: floor-hight, floor-thickness	vertical installations' position	non-structural walls' separation	load-bearing structure's capacity for widening corridor
fourth leaf node of the nodes at level 1	load-bearing structure: HVAC	vertical installations' orientation	---	building expansion
fifth leaf node of the nodes at level 1	corridorwidth	vertical parts' clustering		lift capacity
sixth leaf node of the nodes at level 1	facadeopenings/depth of rooms	---		---

TABLE II
WEIGHTS OF THE NEURAL TREE FOR BUILDING PERFORMANCE.

weight number	1	2	3	4	5	6	7	8	9	10
level 2	.60	.40								
level 1	.60	.40	.50	.50						
level 0	.28	.33	.08	.08	.18	.05	.45	.16	.14	.07

weight number	11	12	13	14	15	16	17	18	19	
level 0	.18	.16	.38	.46	.10	.34	.24	.07	.25	

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. As it was pointed out in figure 3, the membership functions at the terminal nodes are application dependent and therefore their shapes and locations are determined accordingly. However, 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.

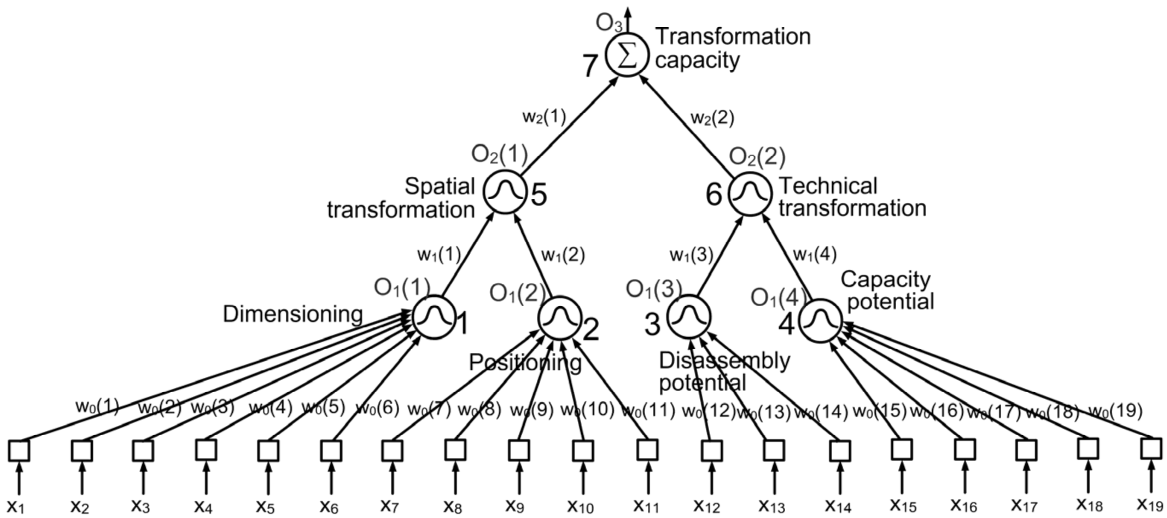

Fig. 5. Neural tree structure for assessment of building performance.

It was pointed out in the preceding section that 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, this means 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. An example from architectural domain may be the following.

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.

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 out in such a way that the consistency is inherently present in this formulation. Peculiar to this application at hand, the consistency condition is a set of multi-input single-output data as given in Table III and Table IV. The imposition of the consistency/boundary conditions can be carried out by genetic search or by adaptive learning and it is carried out by adaptive learning. Note that, 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 advanced learning methods. The network training results are given in Table V, where the first column is the desired output value. The second column is the output provided by the model. The third column is the difference. The mean squared error is found to be 4.9×10^{-4}. As seen from the table, the training results as to the boundary conditions are quite satisfactory for the estimation of the transformation capacity computation and ensuing decision making. For instance for the input pattern given in Table VI we obtain the transformation capacity as .37. This is given in the first row of Table VII, where the outputs of the inner nodes are also indicated.

TABLE III
DATASET AT THE NEURAL TREE INPUT FOR CONSISTENCY CONDITION..

leaf node	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	...	*19*
data sample 1	*.1*	*.1*	*.1*	*.1*	*.1*	*.1*	*.1*	*.1*	*.1*	*.1*	...	*.1*

data sample 2	*.2*	*.2*	*.2*	*.2*	*.2*	*.2*	*.2*	*.2*	*.2*	*.2*	...	*.2*
data sample 3	*.3*	*.3*	*.3*	*.3*	*.3*	*.3*	*.3*	*.3*	*.3*	*.3*	...	*.3*
data sample 4	*.4*	*.4*	*.4*	*.4*	*.4*	*.4*	*.4*	*.4*	*.4*	*.4*	...	*.4*
data sample 5	*.5*	*.5*	*.5*	*.5*	*.5*	*.5*	*.5*	*.5*	*.5*	*.5*	...	*.5*
data sample 6	*.6*	*.6*	*.6*	*.6*	*.6*	*.6*	*.6*	*.6*	*.6*	*.6*	...	*.6*
data sample 7	*.7*	*.7*	*.7*	*.7*	*.7*	*.7*	*.7*	*.7*	*.7*	*.7*	...	*.7*
data sample 8	*.8*	*.8*	*.8*	*.8*	*.8*	*.8*	*.8*	*.8*	*.8*	*.8*	...	*.8*
data sample 9	*.9*	*.9*	*.9*	*.9*	*.9*	*.9*	*.9*	*.9*	*.9*	*.9*	...	*.9*

TABLE IV
DATASET AT THE NEURAL TREE OUTPUT FOR CONSISTENCY CONDITION..

data sample 1	*data sample 2*	*data sample 3*	*data sample 4*	*data sample 5*	*data sample 6*	*data sample 7*	*data sample 8*	*data sample 9*
.1	.2	.3	.4	.5	.6	.7	.8	.9

TABLE V
ADAPTIVE LEARNING RESULTS FROM THE DATASETS GIVEN IN TABLE I AND TABLE II

Given for all inputs and output	Approximation	Error
1.000000E-01	1.542416E-01	5.424161E-02
2.000000E-01	2.049175E-01	-4.917488E-03
3.000000E-01	2.789017E-01	2.109829E-02
4.000000E-01	3.768269E-01	2.317312E-02
5.000000E-01	4.908685E-01	9.131491E-03
6.000000E-01	6.069597E-01	-6.959677E-03
7.000000E-01	7.102828E-01	-1.028281E-02
8.000000E-01	7.851830E-01	1.481700E-02
9.000000E-01	9.021183E-01	-2.118289E-03

TABLE VI
INPUT COMPOSITION FOR BUILDING PERFORMANCE.

leaf node	1	2	3	4	5	6	7	8	9	10	11	12
input value	.1	.5	.7	.1	.1	.5	.5	.7	.3	.9	.9	.5

leaf node	13	14	15	16	17	18	19					
input value	.9	.8	.1	.7	.1	.5	.9					

TABLE VII
BUILDING PERFORMANCE RESULTS FROM NEURAL TREE.

Transformation capacity	.37				
Spatial transform; Technical transform;	.42	.29			
Dimensioning; Positioning; Disassembly potential; Capacity potential	.45	.64	.10	.35	
Graded influential inputs	7	14	17	1	2

In Table VII also the graded five most influential inputs are seen. The influence of each input on the transformation capacity at the output of the neural tree is computed by means of *sensitivity analysis*. The same experiment is carried out with the input composition given by Table VIII. The results for this are presented in Table IX.

TABLE VIII
INPUT COMPOSITION FOR BUILDING PERFORMANCE.

leaf node	1	2	3	4	5	6	7	8	9	10	11	12
input value	.3	.5	.7	.1	.1	.5	.5	.7	.3	.9	.9	.9

leaf node	13	14	15	16	17	18	19					
input value	.5	.9	.1	.7	.1	.5	.9					

TABLE IX
BUILDING PERFORMANCE RESULTS FROM NEURAL TREE.

Transformation capacity	.46				
Spatial transform; Technical transform	.51	.37			
Dimensioning; Positioning; Disassembly potential; Capacity potential	.53	.64	.24	.35	
Graded influential inputs	7	17	14	1	2

Comparison of Table VII and Table IX, it is seen that, with respect to the transformation suitability of the building, the improvement at the input space is reflected at the output as well as at the nodes. At the same time the sequence of the influential inputs changed as well. The results reported above are substantiated by the domain experts who structured the domain knowledge as neural tree. The final step of this application for a desired transformation capacity value at the output, the necessary input composition for this demand is sought. This is accomplished by means of genetic algorithm (GA). This is virtually the unique choice, due to the dimensionality of the search space which is twenty, i.e., 19+1, in this case. As a particular case, for a transformation value of 0.60, the input composition obtained as result of the evolutionary search is given in Table X. For this composition, the system properties namely *dimensioning*, *positioning*, *disassembly potential*, *capacity potential* of the knowledge model are shown in Table XI. Although evolutionary search provides the transformation capacity as 0.6 for the inputs given in Table X, this figure is obtained to be 0.59 in place of 0.60, during the verification of the result. This is due to round off errors.

TABLE X
INPUT COMPOSITION FOR BUILDING PERFORMANCE.

leaf node	1	2	3	4	5	6	7	8	9	10	11	12
input value	.52	.80	.21	.87	.56	.80	.88	.31	.14	.51	.18	.52
leaf node	13	14	15	16	17	18	19					
input value	.80	.21	.87	.56	.80	.88	.31					

TABLE XI
SPATIAL PERFORMANCE RESULTS FROM NEURAL TREE.

Spatial performance	.59				
Function Perception	.85	.20			
Size, Position, Object, Space	.81	.73	.00	.21	
Graded influential inputs	14	1	17	11	16

Another point to mention is that the figure of the transformation capacity as .60 is also a boundary condition as seen in Table III. This is not something to be confused. Since the model is non-linear, the same output, namely 0.6 at the root node can be obtained for both cases. It should be pointed out that, such a search is not a trivial process and the solution by evolutionary search is almost necessary due to multidimensionality of the search space thereby the complexity. The solution conforms to the knowledge embedded in the neural structure. Such studies and ensuing real-life applications exemplified here are important in the building design processes where the eventual transformation possibility of building being designed is in the context at the very beginning. In this way, the life cycle of a building is better determined and improved. Note that, the above input composition given in Table X is not necessarily unique and it is dependent on the initial conditions of the search process, in general. Therefore, multi-objective search can be carried out for optimality satisfying the multi-objectives on the Pareto optimal surface in this multidimensional search space. By means of the GA, the search space can be a sub-set of the total terminal-node inputs, as well as the total input space as is the case in this study. Evolutionary search has unique favourable merits and performance for this search mission; namely the search is in a discrete space and it can be constrained in many different ways to be aligned to what is desired. For instance, by means of the GA the input space can be searched to obtain a suitable input composition satisfying certain conditionality imposed on the inner aspects that belong to the inner nodes which are *positioning*, *disassembly potential*, *capacity potential* in this case, as well as the neural tree output as *transformation capacity* at its root node.

VI. DISCUSSION

Although fuzzy logic has found application areas in all areas of science without exception, its relatively more effective applications are reported from the engineering disciplines where soft computing plays important role from the practical and pragmatic solutions viewpoint. However fuzzy logic, as a machinery of soft computing can provide solutions for the complex problems of soft sciences where complexity is mainly due to the softness of the data. For such problems, classical methods are not convenient due to their crisp computation neglecting the fuzzy concept, yet they prevail in the real world. One of such interesting application areas of soft computing is Architecture and Building Technology where the complexity of data and information prevails due to the amount and softness of both. The application examples presented here are essential computational tools in this domain from the integration of soft computing into building technology viewpoint. It is noteworthy to point out, that there is a bottleneck with regard to the application of fuzzy logic in soft application domains where the *curse of dimensionality* prevails. To circumvent this, in the beginning of 90s adaptive learning systems are introduced into fuzzy logic so that fuzzy modelling has turned to be a function approximation with universal approximation properties [10-17]. However, fuzzy logic is essentially meant for dealing soft issues of science by "computing with words" [18, 19] rather than being another function approximation method akin to neural network systems. Fuzzy models constructed by data analysis are terminologically referred to as *data driven approaches* which apparently deal with the complexity at the cost of compromising the transparency of fuzzy logic systems that is mainly claimed for fuzzy systems. The neural tree approach associated with the evolutionary search presented in this work is one possible alternative to eliminate such paradoxical use of fuzzy logic. The efficiency of data driven fuzzy models in case of multidimensionality of the data space is investigated [20, 21]. From these studies, it is clear, that the efficiency degrades swiftly with the increase of the dimensionality while with low dimensionality and weak non-linearity, fuzzy models clearly score with respect to both accuracy and transparency. The present work is interesting not only due to the application area viewpoint where the methods of fuzzy logic and evolutionary computation are not very familiar to architects and building sector but also due to the novel approach integrating these methods into neural tree. This is beyond the conventional neuro-fuzzy approaches where neural network in some way or other combined with fuzzy concept, the transparency of the model being minimal or the transparency is maintained however the complexity is minimal.

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 [22, 23]. In the neural tree with fuzzy logic presented in this research forms a fuzzy model especially as described by Murray [24], 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.

VII. Conclusion

Knowledge driven fuzzy modelling is described for performance analysis where the model has neural tree structure. This is exemplified by a building technological application with respect to building cycle. The research has novelty features as it integrates analytical hierarchy process in the modelling process that it provides efficiency and effectiveness in knowledge representation. The model is finally determined by learning where learning refers to the integration of the boundary conditions, if any, of the knowledge at hand. The boundary conditions refer to a set of rules, which are to consider as inherent conditions, and the knowledge in question should naturally contain it. The neural tree approach presented here is very appealing as it provides the user with effective and efficient decision making. In a particular application presented in this work the decision-making is about a categorical group of 400 houses. The decision-making about each house is virtually in real-time, provided once the neural tree is established for this very group. Another point worth to emphasize is that, the outcomes within the neural tree is a result of logical feed-forward rule chaining operations based on the domain expert knowledge embedded in the neural tree. The properties of AHP matching the fuzzy logic inference are important for this operation, so that the neural tree operates as an inference engine which underlies the final outcome. The reported cooperative work with a semi-governmental building management organisation in the Netherlands indicates the suitability of the work for a wide range of similar applications of technological, industrial and practical interest. After the establishment of the knowledge model, there are essential search issues to obtain meta-knowledge about the application. The search issues are carried out by evolutionary algorithms due to the complexity. For this very reason, to proceed with GA is virtually unique and imperative. The research is described in detail and its merits are pointed out in a framework having transparent fuzzy modelling properties and addressing complexity issues at the same time.

References

[1] B. Kosko, Fuzzy Engineering, Prentice Hall, 1997.

[2]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

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

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

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

[6] 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.

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

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

[9] E. Durmisevic, *Transformable Building Structures*, PhD Thesis, Delft University of Technology, The Netherlands, 2006.

[10] L. X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least square learning," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 807-814, 1992

[11] X. J. Zeng and M. G. Singh, "Approximation Theory of Fuzzy Systems – SISO case," *IEEE Trans. Fuzzy Syst.*, vol. 2, no. 2, pp 162-176, 1994.

[12] X. J. Zeng and M. G. Singh, "Approximation theory of fuzzy systems – MIMO case," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 2, pp 219-235, 1995.

[13] Hao Ying, "General SISO Takagi-Sugeno fuzzy systems with linear rule consequent are universal approximators," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 4, pp. 582 – 587, 1998.

[14] B. Kosko, "Fuzzy systems as universal approximators," *IEEE Trans. Computers*, vol. 43, no. 11, pp. 1329 – 1333, 1994.

[15] J. L. Castro and M. Delgado, "Fuzzy systems with defuzzification are universal approximators," *IEEE Trans. Syst. Man and Cybern., Part B*, vol. 26, no. 1, pp. 149 – 152, 1996.

[16] R. Rovatti, "Fuzzy piecewise multilinear and piecewise linear systems as universal approximators in Sobolev norms," *IEEE Trans. Fuzzy Systems*, vol. 6, no. 2, pp. 235 – 249, 1998.

[17] Hao Ying, Yongsheng Ding, Shaokuan Li and Shihuang Shao, "Comparison of necessary conditions for typical Takagi-Sugeno and Mamdani fuzzy systems as universal approximators," *IEEE Trans. Syst., Man and Cybern., Part A*, vol. 29, no. 5, pp. 508 – 514, 1999.

[18] L. A. Zadeh, J. Kacprzyk, *Computing with words in Information/Intelligent systems 1: Foundations,* L.A. Zadeh (Ed.), Springer Verlag, Heidelberg, 1999.

[19] L. A. Zadeh, J. Kacprzyk, *Computing with words in Information/Intelligent systems 2: Applications,* L.A. Zadeh (ed.), Springer Verlag, Heidelberg, 1999.

[20] Ö. Ciftcioglu and I. S. Sariyildiz, "Identification of enhanced fuzzy models with multidimensional fuzzy membership functions," *Proc. Fuzz-IEEE 2005, Int. Conf. on Fuzzy Systems*, Reno, Nevada, USA, 2005.

[21] Ö. Ciftcioglu and I. S. Sariyildiz, "On the efficiency of multivariable TS fuzzy modeling," *Proc. Fuzz-IEEE 2006, Int. Conf. on Fuzzy Systems*, Vancouver, BC, Canada, 2006.

[22] 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.

[23] 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

[24] 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.