Proceedings of TMCE 2008 Symposium, April 21–25, 2008, Izmir, Turkey, edited by I. Horváth and Z. Rusák

PERFORMANCE-BASED PARETO OPTIMAL DESIGN

I. Sevil Sariyildiz
Department of Building Technology
Delft University of Technology
the Netherlands
i.s.sariyildiz@tudelft.nl

Michael S. Bittermann

Özer Ciftcioglu
Department of Building Technology
Delft University of Technology
the Netherlands
{m.s.bittermann, o.ciftcioglu}@tudelft.nl

ABSTRACT

A novel approach for performance-based design is presented, where Pareto optimality is pursued. Design requirements may contain linguistic information, which is difficult to bring into computation or make consistent their impartial estimations from case to case. Fuzzy logic and soft computing are the essential means to deal with this matter. In this work an innovative neural fuzzy system is considered for soft computing in design. The system has a neural network structure with the properties of neural tree. The nonlinear processing units at the nodes are selected as Gaussians, so that the system can be interpreted in fuzzy terms. Such a knowledge model can be subject to employment in many diverse areas. In this work it is used for a soft computing application in architectural design, where a number of linguistic information is used in the specification of requirements. The quantifications of qualitative descriptions in design are integrated into the system and fuzzy computations are carried out in a neural network framework. The application concerns a layout of multiple housing units, involving multiple, conflicting requirements, so that Pareto optimality is aimed for. This is a much desirable aid in a design process as it provides guidance for design enhancement, where the design quality underlies the guaranteed design performance as to the specifications.

KEYWORDS

Neural fuzzy system, Pareto optimal design, soft computing, knowledge model, intelligent computing, evolutionary computation

1. INTRODUCTION

Design requirements may contain linguistic information, which is difficult to bring into computation. For example one may require a very open space or a design with high functionality. This difficulty is usually not addressed for design tasks that are concerned with a limited aspects of a design, where requirements may be crisply defined for the sub-domain of design performance, such as aspects of HVAC design (Huang, Lam, 1997; Wright, 1996), structural design (Soh, Yang, 1996; Camp, et al., 1998; Ishida, Sugiyama, 1995; Wang, Chen, 1996), and layout design (Damsky, Gero, 1997; Gero, Kazakov, 1998; Jo, Gero, 1998). However, generally design requirements have a linguistic character, which entails complexity and imprecision forming a fundamental bottleneck for computational design. In order to take these issues into account, fuzzy logic and soft computing are the essential means to be employed.

Fuzzy logic was introduced into science more than four decades ago. 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 were 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 as neuro-fuzzy systems. 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 diminished in the new millennium, and the exploration of new avenues in the realm of fuzzy logic became desirable. In this respect, neural tree structures introduced at the beginning of the 90s (Foresti, Micheloni, 2002; Sankar, Mammone, 1991; Sirat, Nadal, 1990; d'Alché-Buc, et al., 1994) together with evolutionary computation can be another important paradigm boosting the fuzzy logic concept in order to deal with the complex problems of design.

The goal of this paper is to present a novel method for modelling design requirements and demonstrate its merits for performance assessment in computational design. Based on the views put forward above, in this work, the potentials of neural trees for structuring information are combined with the reasoning process of fuzzy logic. This yields a special type of knowledge model, which is both, transparent and 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 a tree structure and determining the fuzzy membership functions accordingly. In this way a neural-fuzzy model is established that handles the common linguistic nature of the design performance concept.

The capability of the model for performance-based design is demonstrated by means of an implementation, where the model is used during multi-objective-optimization-based positioning of houses in a residential neighbourhood. Optimal positioning satisfying multiple objectives is accomplished using a genetic algorithm. These methods are extensively discussed by Deb (Deb, 2001). In the present work the neural-fuzzy knowledge model plays the role of fitness function, and the search aims to identify Pareto-optimal solutions.

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 neural 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. This is followed by conclusions.

2. 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 Figure 1.

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 ***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.

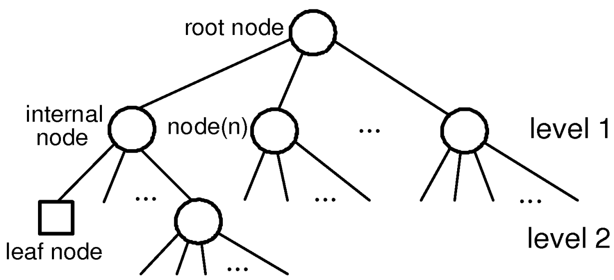

Figure 1 The structure of a neural tree

3. ANALYTICAL HIERARCHY PROCESS AND THE FORMATION OF A NEURAL TREE STRUCTURE

The AHP method is a technique developed by Saaty (Saaty, 1980) 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, taking 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 matrix A 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 (Saaty, 1980; Saaty, 2000).

4. NEURAL TREE AS UNDERLYING DOMAIN KNOWLEDGE STRUCTURE

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 nodew_i is related to X by the relation

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

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

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

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.

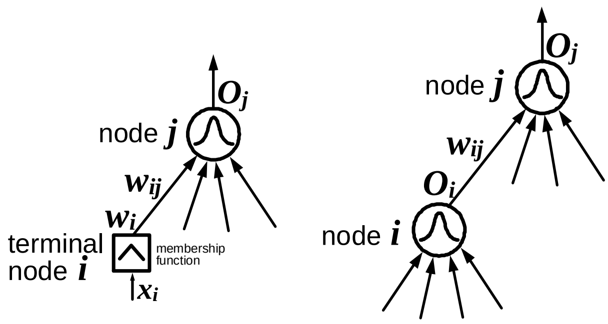

Figure 2 The detailed structure of a neural tree with respect to different type of node connections

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} of a node. They are dependent on the neural tree structure and determined by the domain knowledge obtained using the method of AHP, for instance. Note that this is a novel type of computation at each node which is slightly different than conventional radial basis function (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 $c = [1, 1, 1, \ldots, 1]$, 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, which corresponds to a logical OR operation.

Using 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 in Figure 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 Figure 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 $c = [1, 1, 1, \ldots, 1]$, 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.

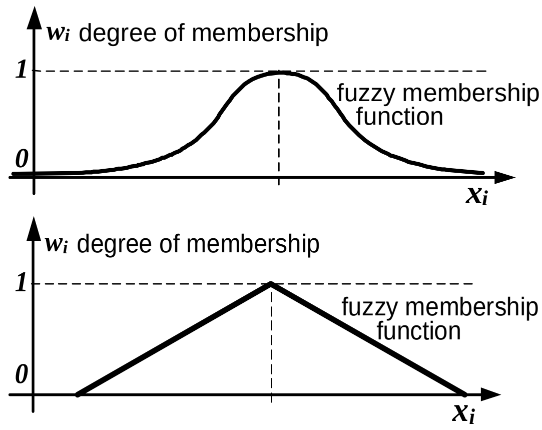

Figure 3 Two possible fuzzy membership function type among many others, at the terminal node

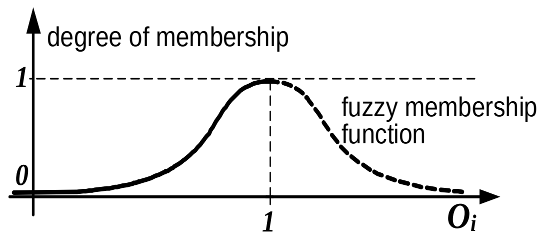

Figure 4 Fuzzy membership function at non-terminal node

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 a logic AND operation. The centre location of the i-th Gaussian membership function is selected as w_{ij} due to the 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 exemplified in the following architectural design application.

5. IMPLEMENTATION OF THE MODEL

The important feature of this concept put forward 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 an architectural design application. The design task is the identification of optimal locations of a number of housing units on their respective lots. The streets and lots are provided in advance in this design case. Figure 5 shows 20 houses. 17 of them are subject to optimal positioning.

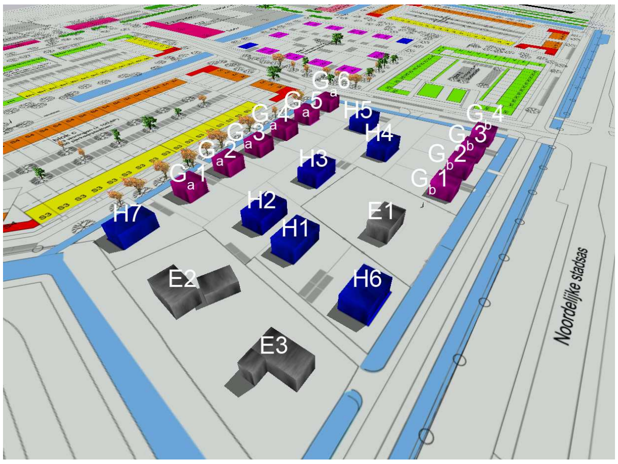

Figure 5 The buildings subject to optimal positioning, except buildings E1, E2, and E3, which are existing buildings

The houses that are not subject to positioning are $E1$, $E2$ and $E3$,since they are existing buildings. All buildings are two storeys high. Houses $E1$, $E2$ and $E3$ have varying floor plan dimensions and orientations; houses $H1 - H7$ are 12m long, 8m wide and their longer axis is oriented in east-west direction; the houses $G_a1 - G_a6$ and $G_b1 - G_b4$ form two groups of houses, which are situated along a line parallel to the perimeter of the neighbourhood. It was an initial basic choice of the architect to align these houses with respect to each other, and this is respected as an architectural premise throughout the implementation, so that any computational solution identified later on has this property. These houses have a square shaped floor plan of 8m by 8m and they are located along a line at equal distance from each other. The south direction in the situation is towards the street indicated as *Noordelijke stadsas* in Figure 5. The configuration shown in the figure is a design proposed by an urban design office. In the design task for optimal positioning two partially conflicting aspects are considered. The first one is the visual privacy of the buildings, and the second one is the size of the gardens.

5.1. Assessment of visual privacy aspects

Figure 6 shows the same situation as Figure 5 from the viewpoint of a virtual observer labelled *avatar*, which is standing nearby house $H4$ and is viewing the scene. The figure illustrates the principle model behind the computation of the perception-based *visual privacy*. The perception is obtained using a probabilistic perception theory (Ciftcioglu, et al., 2006).

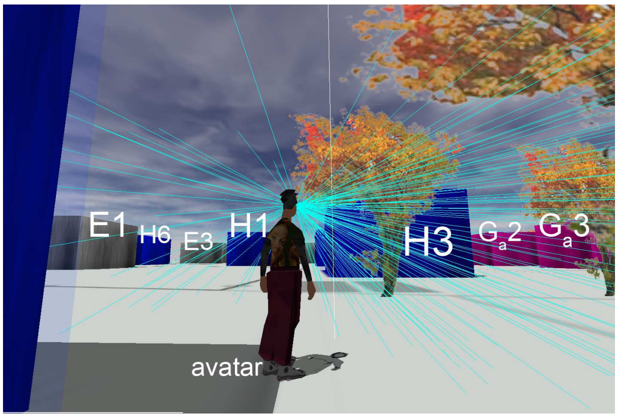

Figure 6 Implementation of the probabilistic perception model by means of an avatar: The amount of sightlines interacting with the objects in view quantifies the degree of awareness for the objects

In the perception theory the visual attention an observer pays to a scene is modelled as a probability density function (pdf). This is illustrated in Figure 6 by means of a number of vision rays that are leaving the eyes of the avatar in random directions. The randomness of the directions is shaped in accordance with the probability density involved in the probabilistic perception theory mentioned above. Integration of the pdf over a certain domain yields perception that becomes a probability. This probability quantifies the degree an observer is mentally aware of the objects in his/her environment. This method is implemented into the computational design process, so that the perception of one building from another one is quantified. Figure 7 illustrates the computation of visual perception of the buildings $H1$, $H2$, $H3$, and $H4$ from building $E1$. Here the viewpoint of the observer is taken as the geometric centre point of the north facade of building $E1$.

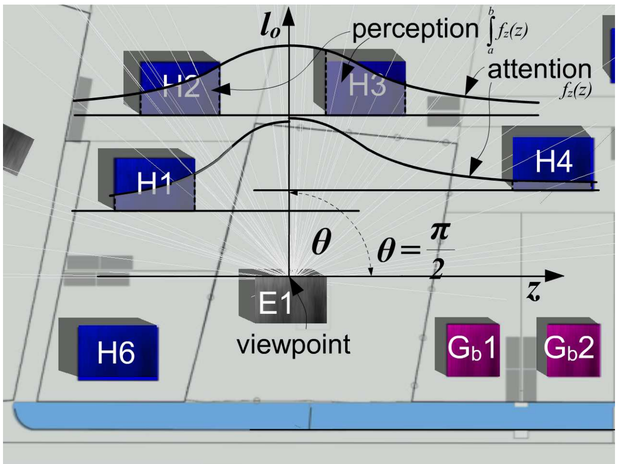

Figure 7 Sketch indicating the computation of the degree of perceptions of the houses $H1$, $H2$, and $H3$ from the viewpoint $E1$

The curves plotted along the z axis are the probability density functions belonging to the perceptions of the houses $H1 - H4$, which gives the degree of visual attention along a building. The integral of the pdf over the length of the south facade of each house is indicated as a shaded area and it quantifies the perception of the respective facade. Based on the probabilistic perception in this implementation the visual privacy belonging to an area is quantified as the reciprocal of the summed up perception of the area obtained from the relevant observation points in an environment. Explicitly we calculate the visual privacy of an object O as

$$Y_{privacy}(O) = \frac{1}{\sum P(O,V_1)+P(O,V_2)+...+P(O,V_n)} \quad (9)$$

where $P(O, V_n)$ is the degree of perception of object O from the n-th viewpoint. In this implementation we consider the visual privacy of the south facade of the building, because in this design it is expected that living rooms and openings to the garden are oriented to the south side of the buildings, and these areas are considered the most important ones with respect to privacy perception in this design. Figure 8 illustrates the implementation of the visual privacy computation for the houses of the housing complex.

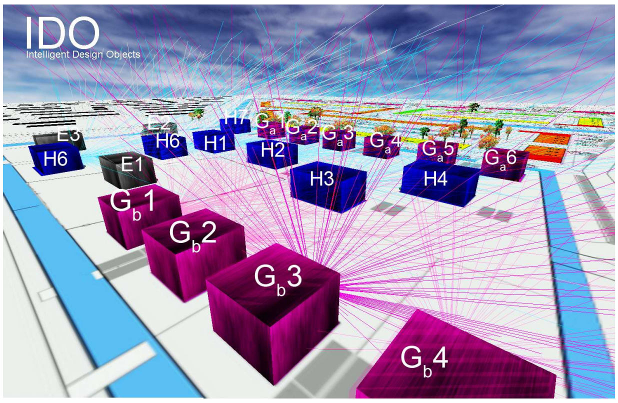

Figure 8 Illustration of the visual privacy computation based on the probabilistic perception model indicated in Figure 7

Every south facade is perceived from several viewpoints and the visual privacy for each house is computed. In the computation of the perceptions in this implementation, occlusion is considered. This is done by a simplified test of the visibility of a building viewed from another one. The mechanism is sending a ray from the centre location of the first building to the viewpoint identifying if the ray is intercepted by another building located in between them. If this is this case the perception of the building from the second one is considered to be zero.

5.2. Assessment of garden aspects

A second aspect considered in the design of the housing complex is the size of the gardens. We consider that in general a garden located south of the building it belongs to is most desirable due to exposure to direct sunlight. Therefore the garden performance is calculated regarding the south garden. In particular the size of the south garden is considered to be relevant. The buildings $H4$ and $H5$ form an exception. The lots of these houses are oriented in east-west direction. Therefore, next to the garden in south direction, the gardens west of the buildings are considered. In this case the west direction is used and not the east direction, assuming that for this design task the residents of the houses $H4$ and $H5$ appreciate more to have direct sunlight in their garden during the evening rather than in the morning.

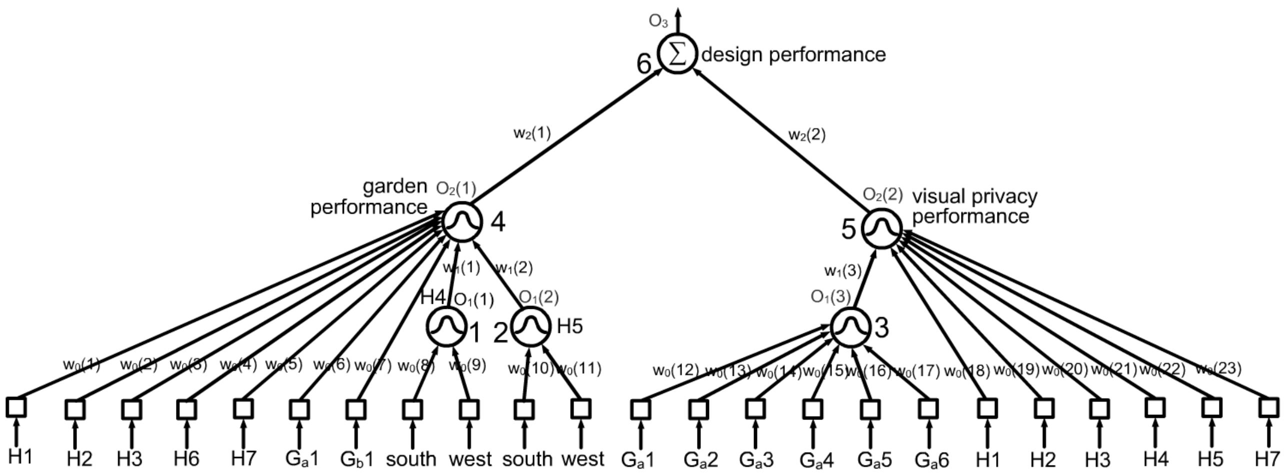

Figure 10 Neural tree structure for assessment of design performance

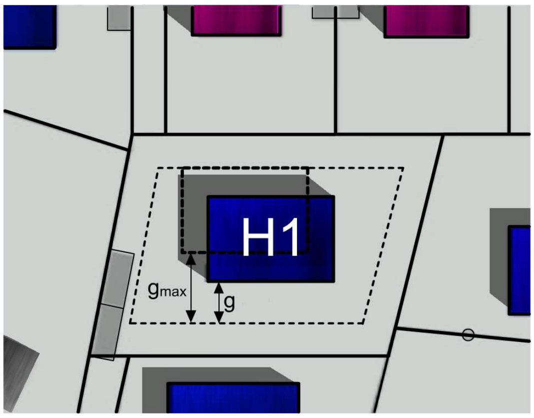

Figure 9 Calculation of the garden performance

In order to determine the garden performance the size of the garden in south direction is normalized with respect to the maximally possible size of the garden in this direction. The maximum size of the garden in south direction is restricted by the minimum distance between the boundaries for placement in north and south direction and the width of the house. This is illustrated in Figure 9 using house $H1$ as an example.

In the figure the boundary of the lot is shown as a solid line while the placement boundary is shown as a dashed line. Explicitly the garden performance G is given by g/g_{max}.

5.3. Establishing the knowledge model

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 Figure 10. In the context of the design application the design performance is determined by two sub-domains, namely the performance of the garden and the performance in terms of the visual privacy at one level below from the root node, designated as level 2. At one level further below is the terminal level except with respect to the garden performance of houses $H4$ and $H5$, where the garden performance has additional two sub-aspects. These aspects are the performance of the garden to the west and the south side of the house respectively. Another exception is the privacy performance of the houses $G_a1 - G_a6$, which together form an additional sub-aspect of the privacy performance. The determinants of the design performance on the terminal level are given in Table 1.

Table 1 Determinants of the design performance

Garden performance	Visual privacy performance
Garden of house $H1$	Privacy of house $H1$
Garden of house $H2$	Privacy of house $H2$
Garden of house $H3$	Privacy of house $H3$
Garden of house $H4$	Privacy of house $H4$
Garden of house $H5$	Privacy of house $H5$
Garden of house $H6$	Privacy of house $H6$
Garden of house $H7$	Privacy of house $H7$
Garden of house G_a1	Privacy of group G_a
Garden of house G_b1	

These determinants form a multidimensional search space, which is complex with respect to its dimensionality. In this space, Pareto optimality is most desirable for multi criteria based search. This will be elaborated later on. For the tree structure established, the connection weights at each level assessed by do-

Table 2 Weights of the neural tree for the design performance

weight nr.	1	2	3	4	5	6	7	8	9
level 2	.60	.40							
level 1	.11	.09	.15						
level 0	.28	.33	.08	.08	.18	.05	.45	.16	.14

weight nr.	10	11	12	13	14	15	16	17	18
level 0	.28	.33	.08	.08	.18	.05	.45	.16	.14

weight nr.	19	20	21	22	23
level 0	.28	.33	.08	.08	.18

main experts are given in Table 2. These weights indicate the relative importance of a sub-aspect compared to other sub-aspects. The structure can be considered as constitution of domain knowledge, where the connecting weights between the nodes are determined by expert judgment.

Each aspect is considered in the context of design performance of the housing complex and eventually assessed between zero and unity. This assessment may be accomplished by using the method of AHP, in a complex design task. The assessments having been made duly, they are used as connection weights w_{ij} in the neural tree. Determining the parameter values in this structure, namely the weights and the individual width of the Gaussians at the non-terminal nodes, 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. Therefore the structure represents a multi-variable 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.

TThe 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 Figure 11.

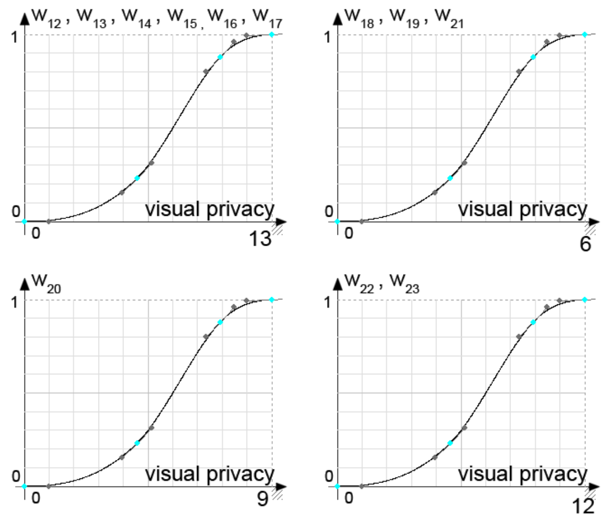

Figure 11 Membership functions at the terminal nodes

he shapes are selected by domain experts. Explicitly, the fuzzy functions are the representations of the requirement specifications of the design. Please note that the functions selected for the privacy performance measurement all have the same basic shape, however the output maxima are at different locations to express the different requirements that are due to the different housing types and lot conditions involved. Concerning the garden performance the fuzzy membership function used is simply $w_i = x$ because the garden performance $G = g/g_{max}$ in Figure 9 is already normalized between zero and one, so that it directly serves as the node output of the respective terminal node.

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.

5.4. Training of the neural tree

The neural tree output follows the trend of the terminal node outputs w_i representing the associated degree of membership. 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. Since the research is carried out in a department of architecture, an example from the architectural domain is more relevant. If all the design determinants belong to a design that is by all means modern, where the attribute modern is reflected by a high output at the terminal level, then the final design output belongs also to 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 a design that is by no means modern, then, the final design output does not belong 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 still give some value as an output even when 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 out 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 3 and Table 4, 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 3 and 4 is seemingly simple, imposition of this simple data set on the highly non-linear fuzzy knowledge model requires adaptive or genetic learning. In the present implementation adaptive learning is used for high accuracy. The approximation error for this data set is relatively higher for the lower input/output pairs. This is seen from Table 5.

Table 3 Dataset at the input of the neural tree to establish the consistency condition

leaf node	1	2	3	4	5	...	23
data sample 1	.1	.1	.1	.1	.1	...	.1
data sample 2	.2	.2	.2	.2	.2	...	.2
data sample 3	.3	.3	.3	.3	.3	...	.3
data sample 4	.4	.4	.4	.4	.4	...	.4
data sample 5	.5	.5	.5	.5	.5	...	.5
data sample 6	.6	.6	.6	.6	.6	...	.6
data sample 7	.7	.7	.7	.7	.7	...	.7
data sample 8	.8	.8	.8	.8	.8	...	.8
data sample 9	.9	.9	.9	.9	.9	...	.9

Table 4 Neural tree output to establish the consistency condition

data sample 1	*data sample 2*	*data sample 3*	*data sample 4*	*data sample 5*
.1	.2	.3	.4	.5

data sample 6	*data sample 7*	*data sample 8*	*data sample 9*
.6	.7	.8	.9

Table 5 Adaptive learning results from the datasets given in Table 3 and Table 4

Given for all inputs and the root output	Approximation	Error
1.00 10^{-1}	1.68 10^{-1}	-6.82 10^{-2}
2.00 10^{-1}	2.30 10^{-1}	-3.00 10^{-2}
3.00 10^{-1}	3.06 10^{-1}	-5.96 10^{-3}
4.00 10^{-1}	3.95 10^{-1}	5.47 10^{-3}
5.00 10^{-1}	4.91 10^{-1}	8.62 10^{-3}
6.00 10^{-1}	5.89 10^{-1}	1.12 10^{-2}
7.00 10^{-1}	6.79 10^{-1}	2.11 10^{-2}
8.00 10^{-1}	7.80 10^{-1}	1.98 10^{-2}
9.00 10^{-1}	9.43 10^{-1}	-4.34 10^{-2}

5.5. Identification of Pareto optimal designs

Having established the fuzzy neural tree the design task is to maximize the output at the root node by

Table 6 Resulting widths of the Gaussians at the non-terminal nodes

Node nr.	1	2	3
σ	7.22 10^{-2}	3.36 10^{-1}	1.95 10^{-1}

Node nr.	4	5
σ	1.77 10^{-1}	1.55 10^{-1}

identifying optimal location of the buildings. This is accomplished by genetic search. The output at the root node, which expresses the design performance by a scalar number, can be used as the representation of the fitness of the respective chromosome. In this way the genetic algorithm (GA) uses the knowledge embedded in the neural tree during its search for obtaining maximal performance, while the search is essentially treated as a single-objective problem.

However, a GA used for this type of problem is usually sensitive to small changes in the objective function coefficients, which correspond in the present case to the weight factors and widths of the Gaussians in the neural tree. Another drawback is that a GA applied in this way converges to a single solution, and does not provide information about alternative solutions that are equally valid in Pareto sense. Therefore we apply GA using a different approach, which is based on the concept of Pareto optimality. The two objectives, to maximize the garden and the privacy performance of the design simultaneously, are conflicting. The conflict is that satisfaction of one objective diminishes satisfaction of the other one.

In such multi-objective optimization problems exists a set of solutions which are *non-dominated*. This means for each solution of this set there is no other solution in the population that performs better with respect to all objectives. In the objective space the set of non-dominated solutions lie on a surface know as the Pareto-optimal frontier (Fonseca, 1995). We use GA to identify this frontier. This is accomplished by assigning the fitness to a chromosome in a population depending on how many other chromosomes are dominating it. Chromosomes that are not dominated are assigned fitness f_{max}, while the fitness in general is calculated as the reciprocal of the amount of chromosomes that dominate the chromosome in question. In the present implementation f_{max} is set to 10. The fitness is thereafter converted to a probability for reproduction applying the well-known roulette wheel selection principle (Goldberg, 1989).

The boundary of the space for the locations of the houses is given in Table 7, namely the minimal and maximal x and z coordinates for the positions of the houses $H1 - H7$, $G_a1 - G_a6$ and $G_b1 - G_b4$. The boundaries are selected, so that the facades of the buildings are at a distance greater than 3m from the boundary of the lot, as this is required by legal regulations in this design case. For the sake of simplicity of the implementation the boundaries of the placement are taken parallel to the x and z axis. The z axis is in north direction, and the x axis is in east direction.

Table 7 Solution space

House	$H1$		$H2$		$H3$	
	x	z	x	z	x	z
min	25.0	26.0	26.0	46.0	56.0	47.0
max	31.0	34.0	36.0	56.0	69.0	56.0

House	$H4$		$H5$		$H6$	
	x	z	x	z	x	z
min	81.0	34.0	86.0	52.0	21.0	6.0
max	117.0	38.0	114.0	57.0	28.0	16.0

House	$H7$		G_a1		G_b1	
	x	z	x	z	x	z
min	3.0	70.0	27.0	67.0	76.0	7.0
max	10.0	80.0	32.0	81.0	79.0	22.0

Figure 12 shows a design at the beginning of the genetic search process, which is a random configuration. It has a design performance of 0.41, which is the output value at the root node of the tree. The marked areas on the lot are the locations originally proposed by the urban design office. The result of the search process is shown in Figure 13, where the best fitness that occurred during the search is plotted together with the average fitness of the chromosomes for each generation.

Please note that the fitness shown in Figure 13 is the output at the root node. Figures 14-16 show the performance of each chromosome in the population plotted with respect to its privacy perception performance and garden performance values. In the beginning of the search process the population is distributed evenly in terms of garden and privacy performance. This is shown in Figure 14. After five generations the GA found a convex Pareto optimal frontier. This is shown in Figure 15. After 20 generations the population of the GA clustered at four locations on the Pareto frontier. This is shown in Figure 16. This behaviour of the evolutionary algorithm is due to its inherent “pressure” towards the Pareto optimal frontier, which is achieved by the dominance based

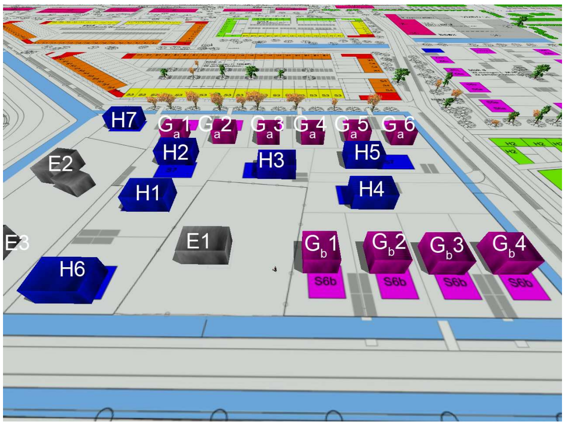

Figure 12 Illustration of a design with a design performance of 0.41 at the beginning of the Genetic search process

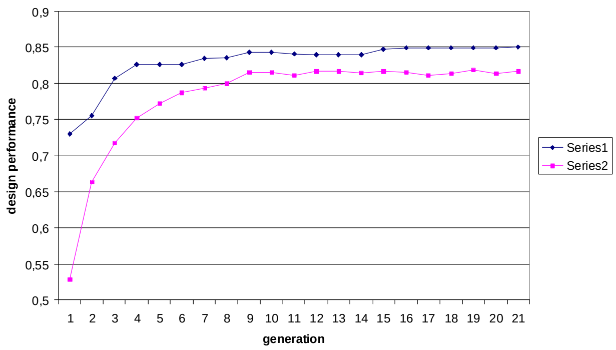

Figure 13 Genetic search process results

selection procedure of the GA described above. The resulting designs along the Pareto optimal front are equally valid solutions, while each solution has a different tradeoff with respect to the design criteria.

Four Pareto optimal designs are shown in Figures 17-20. The designs shown in the figures belong to the solutions indicated as nr. 1, 2, 3, and 4 in Figure 16, respectively. The design shown in Figure 17 has the greatest garden performance of the four designs shown. This is because all houses have large south gardens, respectively west gardens in the case of houses $H4$ and $H5$. In Figure 17 the visual privacy is relatively low compared to the other designs, because many houses are located quite close to and are directly facing the south façade of neighbouring buildings.

The design shown in Figure 18 provides a higher visual privacy compared to the design from Figure 17. This can be explained from the fact that the houses $G_b1 - G_b4$ are located at greater distance from house $H4$ thereby increasing privacy of $H4$.

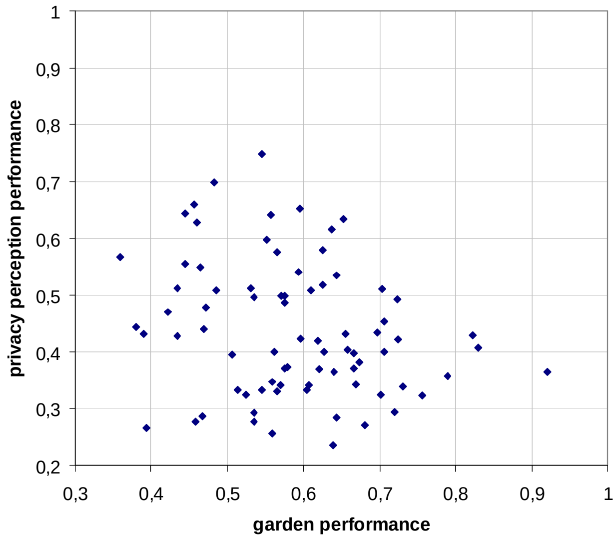

Figure 14 Privacy performance and garden performance belonging to each chromosome of the population in the first generation of the GA

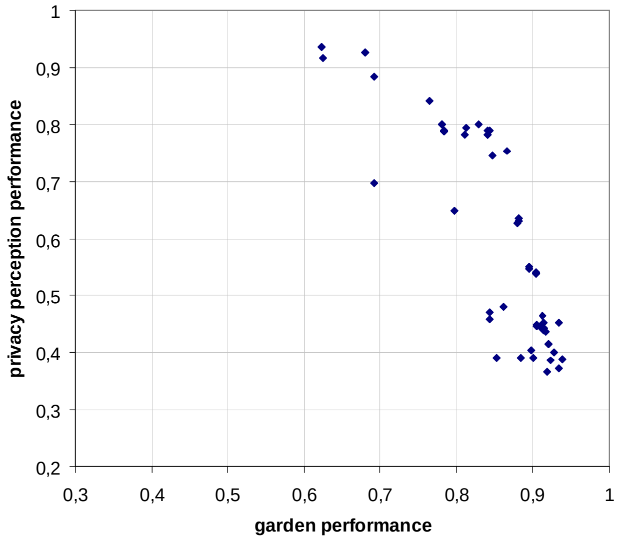

Figure 15 Forming the Pareto optimal front: privacy performance and garden performance belonging to each chromosome of the population in the 5th generation of the GA

Additionally $H4$ is located not directly below $H5$, so that the privacy of $H5$ is increased compared to Figure 17. Figure 19 is similar to Figure 18 with the difference that house $H4$ is moved directly south of $H5$. Therefore the privacy performance of Figure 19 is reduced compared to Figure 18. Figure 20 is similar to Figure 18 with the difference that the houses $G_b1 - G_b4$ are at a greater distance from $H4$, so that the privacy is increased and garden performance is

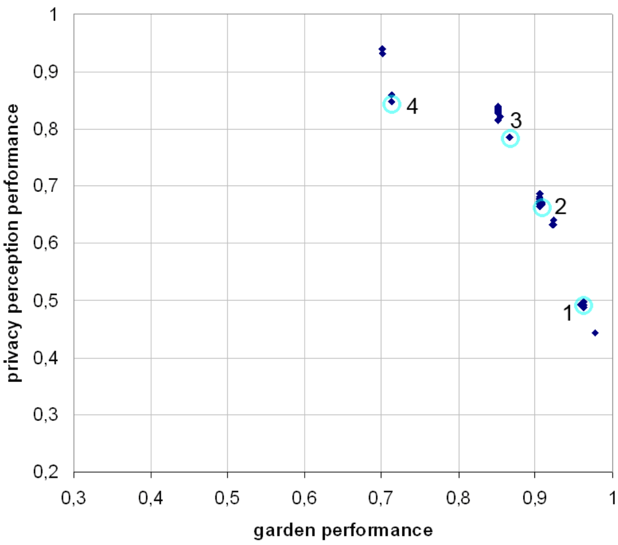

Figure 16 Pareto optimal front formed by privacy performance and garden performance belonging to each chromosome of the population in the 20^{th} generation of the GA

reduced in Figure 20 compared to Figure 18.

A designer may select any of the designs on the Pareto-optimal front for further elaboration; having certainty that each of the solutions he/she is choosing from is Pareto-optimal with respect to the design criteria put forward. In order to make a decision about which design to pick, higher-level design criteria can be brought into play. In the present case it is natural to consider the relative importance of privacy and garden performance, as this is already integrated into the defuzzification process in the neural model formed earlier. Explicitly in this case the garden performance is considered 60/40 times more relevant than the privacy performance. Based on this higher-level criterion we select a certain design located on the Pareto front, which is both, non-dominated and has the highest output value at the root-node of the neural tree. This means the design selected has maximal design performance at the same time.

The selected design is shown in Figure 21. It has a design performance of 0.85. For this design the resulting design parameters as location of the buildings are given in Table 8. The outputs of the tree nodes for the designs shown in Figure 5 and 21 are given in Table 9 for comparison. The results indicate that the combination of fuzzy neural tree and genetic algorithm is able to identify Pareto optimal designs with maximal design performance, while insight into the contributions of the model constituents is provided

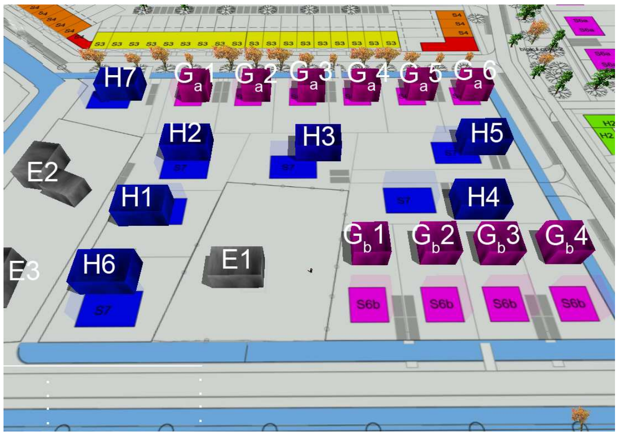

Figure 17 Resulting Pareto-optimal design indicated as solution nr. 1 in Figure 16

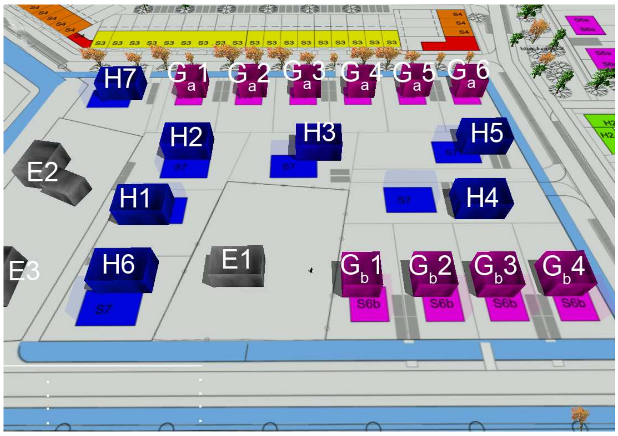

Figure 18 Resulting Pareto-optimal design indicated as solution nr. 2 in Figure 16

due to the transparency of the approach. This is seen in Table 9. In the figures both design alternatives, namely the one proposed by the computational design system, and the design proposed by the human design professionals based on conventional methods are shown. The latter one is indicated by rectangles on the respective lots on the ground plane, which are the projections of the professional design onto the plane. This is done, so that the computational designs can be easily visually compared with the conventional one.

In the following we compare the selected Pareto optimal design shown in Figure 21 with the conventional design. We note that the computational design is similar to the conventional one with respect to the positions of houses $H7$ and the group of houses $G_a1 - G_a6$. There are also differences: In the computational result the group $G_b1 - G_b4$ and houses

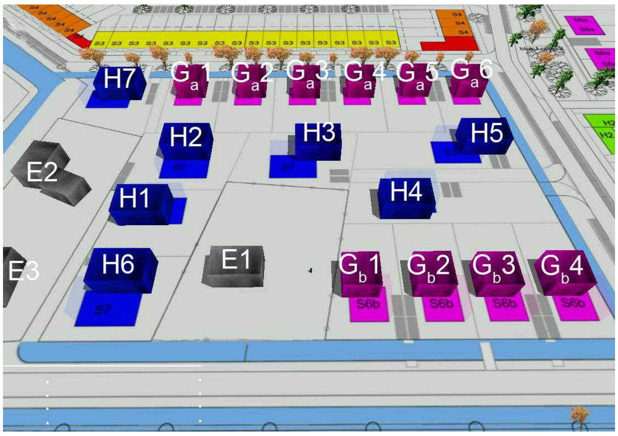

Figure 19 Resulting Pareto-optimal design indicated as solution nr. 3 in Figure 16

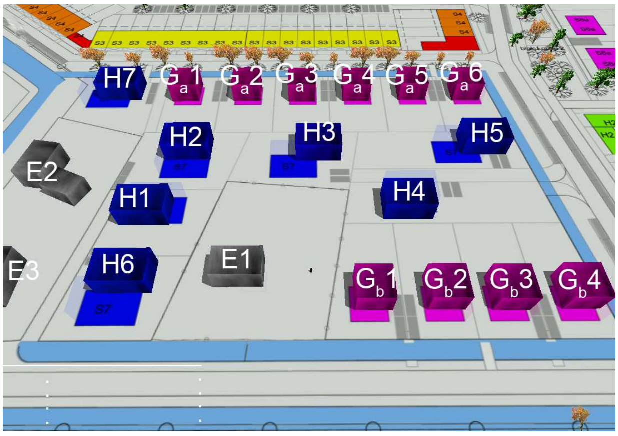

Figure 20 Resulting Pareto-optimal design indicated as solution nr. 4 in Figure 16

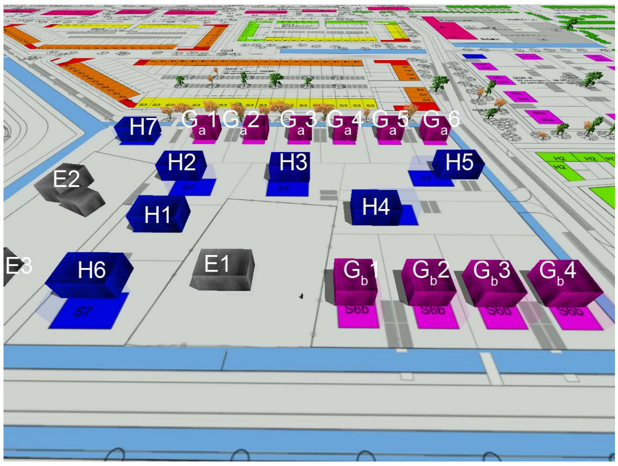

Figure 21 Selected Pareto-optimal design with design performance .85, where garden performance is .87 and visual privacy performance is .82

Table 8 Selected Pareto-optimal design shown in Figure 21

House	$H1$		$H2$		$H3$	
	x	z	x	z	x	z
value	29.0	33.6	28.1	55.5	61.5	55.1

House	$H4$		$H5$		$H6$	
	x	z	x	z	x	z
value	85.0	36.5	113.5	56.9	21.5	14.3

House	$H7$		G_a1		G_b1	
	x	z	x	z	x	z
value	5.0	78.8	29.64	80.2	77.2	13.0

$H3$, $H4$, $H5$ and $H6$ are located further north than in the conventional design, so that the south gardens of these houses are larger in the computational case. For the same reason the privacy of the group of houses $G_a1 - G_a6$ is reduced in the computational case. Houses $H5$ and $H4$ are moved further apart in the computational design, so that the visual privacy of $H5$ is increased compared to the conventional case. House $H2$ is moved to the north west of its lot, so that both its garden is larger and its privacy is increased in the computational design compared to the conventional one.

6. DISCUSSION

The knowledge model presented in this work has a neural tree structure with fuzzy logic processors embedded as the inner nodes of the structure. Depending on the complexity of the domain knowledge, the method of analytical hierarchy process is one alternative, which can be made use of during the constitution of the structure. In this feed-forward structure the output of a node is obtained with fuzzy logic operations using the inputs of the node. This is accomplished by Gaussian membership functions. The model is finally determined by learning where learning refers to the integration of the 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 de-fuzzification at the root node.

The equivalence between neural networks and fuzzy logic for Gaussian fuzzy membership functions is known in the literature (Jang, Sun, 1993; Li, Chen, 2000). The neural tree with fuzzy logic presented in this research forms a fuzzy model especially as described by Hunt, Haas and Murray (Hunt, et al.,

Table 9 Node outputs belonging to the designs shown in Figure 10 and 11

Node	Output	Initial design shown in Figure 10	Selected Pareto-optimal design shown in Figure 20
Design performance	O_3	.411	.850
Garden performance	$O_2(1)$	.463	.869
Visual privacy performance	$O_2(2)$	.333	.821
Garden $H1$	w_1	.917	.951
Garden $H2$	w_2	.895	.996
Garden $H3$	w_3	.098	.897
Garden $H6$	w_4	.048	.825
Garden $H7$	w_5	.802	.883
Garden G_a1	w_6	.052	.944
Garden G_b1	w_7	.0479	.397
South garden $H4$	w_8	.338	.110
West garden $H4$	w_9	.194	.621
South garden $H5$	w_{10}	.297	.981
West garden $H5$	w_{11}	.270	.981
Visual privacy G_a1	w_{12}	.007	.796
Visual privacy G_a2	w_{13}	.204	.957
Visual privacy G_a3	w_{14}	.120	.640
Visual privacy G_a4	w_{15}	.181	.999
Visual privacy G_a5	w_{16}	.013	.988
Visual privacyG_a6	w_7	.694	.746
Visual privacy $H1$	w_{18}	.999	.763
Visual privacy $H2$	w_{19}	.780	.705
Visual privacy $H3$	w_{20}	.703	.914
Visual privacy $H4$	w_{21}	.277	.598
Visual privacy $H5$	w_{22}	.062	.708
Visual privacy $H7$	w_{23}	.608	.892

1996), 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. With respect to the computational power required using the neural tree we note that once the structure is established fulfilling the consistency condition, the execution of the logical operations in the tree can be considered real-time.

We note that in the present application the performance aspects considered are the visual privacies and the sizes of the gardens of the residential units, exclusively. Other aspects, which a designer may consider relevant constituents of design performance, such as other perceptual aspects, may be easily integrated into the neural tree model presented. In this case it is required that also for these aspects fuzzification at the leaf node be defined, i.e. there has to be some mapping from properties of the design to the degree of satisfaction of the perceptual requirement concerned. These features are the manifestations of the transparent nature of the structure, where the meaning of each node is known.

Knowledge driven fuzzy modelling is described for identification of performance-based Pareto optimal architectural designs. The novel knowledge modelling method 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. The potential of the novel method for design is demonstrated by means of an implementation, where the model is used for knowledge-based performance assessment during a computational design process. Particularly the model plays the role of fitness-function during a genetic search. The search aims to find optimal solutions in Pareto-sense, while the search procedure is equipped with the detailed knowledge of the designer on how to evaluate the alternatives. Due to the multi-objective nature of the design task, application of the Pareto concept is most appropriate for effective and efficient solution identification. The results indicate the suitability of the work for a wide range of similar applications of technological, industrial and practical interest.

Ranking by Pareto dominance on problems with an increased number of objectives might not longer be effective (Hughes, 2005; Purshoe, Fleming, 2003). One of the important issues to address in this respect is the diversity of the Pareto solutions with minimal aggregation at the Pareto front (Aguirre, Kiyoshi, 2007). The aggregation of the solutions in this work is seen in Figure 16. This state-of-the-art issue is addressed in the literature and novel methods are proposed (Sato, et al., 2007). The adaptation of such methods is anticipated as an improvement of similar researches including the present one and therefore remains as a future work.

7. CONCLUSIONS

The marked significance of this work is that designer's knowledge on the design requirements can be put in the play effectively and efficiently in architectural design. In particular the uncertainty and imprecision issues that naturally occur when a designer evaluates design alternatives using conventional means are alleviated. This is accomplished by consistently synthesizing designer's knowledge with a higher level of granulation, making the meta-knowledge known that has been previously unknown.

REFERENCES

Aguirre, H. E., Kiyoshi, T., 2007, "European Journal of Operational Research", 181(3), pp. 1670–1690

Camp, C., Pezeshk, S., Cao, G., 1998, "Optimized design of two-dimensional structures using a genetic algorithm", Journal of Structural Engineering 124(5), pp. 551–559

Ciftcioglu, Ö., Bittermann, M. S., Sariyildiz, I. S., 2006, "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

d'Alché-Buc, F., Zwierski, D., Nadal, J. P., 1994, "Trio learning: a new strategy for building hybrid neural trees", Int. J Syst. 5, pp. 259–274

Damsky, J., Gero, J., 1997, "An evolutionary approach to generating constraint-based space layout topologies", Proc. CAAD Futures, pp. 855–874

Deb, K., 2001, "Multiobjective Optimization using Evolutionary Algorithms", John Wiley & Sons

Fonseca, C. M., 1995, "An overview of evolutionary algorithms in multiobjective optimization", Evolutionary Computation 3, pp. 1–16

Foresti, G. L., Micheloni, C., 2002, "Generalized neural trees for pattern classification", Neural Networks, IEEE Transactions on 13, pp. 1540–1547

Gero, J., Kazakov, V., 1998, "Evolving design genes in space layout problems", Artificial Intelligence in Engineering 12, pp. 163–176

Goldberg, D. E., 1989, "Genetic Algorithms", Addison Wesley

Huang, W., Lam, H., 1997, "Using genetic algorithms to optimize controller parameters for HVAC systems", Energy and Buildings 26, pp. 277–282

Hughes, E. J., 2005, "Evolutionary many-objective optimisation: many once or one many?" Proc. IEEE Congress on Evolutionary Computation CEC'2005, Edinburgh, Scotland, pp. 222–227

Hunt, K. J., Haas, R., Murray-Smith, R., 1996, "Extending the functional equivalence of radial basis function networks and fuzzy inference systems", IEEE Trans. Neural Networks 7

Ishida, R., Sugiyama, Y., 1995, "Proposal of constructive algorithm and discrete shape design of the strongest column", AIAA Journal 33, pp. 401–406

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

Jo, J., Gero, J., 1998, "Space layout planning using an evolutionary approach", Artificial Intelligence in Engineering 12, pp. 163–176

Li, H. X., Chen, C. L. P., 2000, "The equivalence between fuzzy logic systems and feedforward neural networks", IEEE Trans. Neural Networks 11

Purshoe, R. C., Fleming, P. J., 2003, "Conflict, harmony and interdependence: relationships in evolutionary multi-criterion optimisation", In: Lecture Notes in Computer Science, Second int. Conf. on Evolutionary Multi-Criterion Optimisation EMO 2003, vol. 2632, P. J. F. C.M. Fonseca, E. Zitzler, K. Deb and L. Thielee, Ed.: Springer Verlag, pp. 16–30

Saaty, T. L., 1980, "The Analytic Hierarchy Process", McGraw-Hill New York

Saaty, T. L., 2000, "The Brain: Unraveling the mystery of how it works", RWS Publications Pittsburgh

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

Sato, H., Aguirre, H. E., Tanaka, K., 2007, “Controlling dominance area of solutions and its impact on the performance of MOEAs”, Springer Heidelberg

Sirat, J. A., Nadal, J. P., 1990, “Neural tree networks”, Network 1, pp. 423–438

Soh, C. K., Yang, J., 1996, “Fuzzy controled genetic algorithm search for shape optimization”, J. of Computing in Civil Engineering 10, pp. 143–150

Wang, B., Chen, J., 1996, “Application of genetic algorithm for the support location optimization of beams”, Computers and Structures 58, pp. 797–800

Wright, J., 1996, “HVAC optimization studies: Sizing by genetic algorithm”, Building Services Engineering Research and Technology 17, pp. 1–14