Discovering Design Principles for Soft Multi-objective Decision Making

Sunith Bandaru[a], Michael Bittermann[b] and Kalyanmoy Deb[c]
[a,c]Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, India
[b]Faculty of Architecture, Delft University of Technology, The Netherlands.
[a]sunithb@iitk.ac.in, [b]m.s.bittermann@tudelft.nl, [c]deb@iitk.ac.in

KanGAL Report Number 2011015

ABSTRACT
Decision-making tasks sometimes involve soft objectives. They are soft in the sense that they contain uncertainty, imprecision or vagueness. For example, decisions on built environment aim to maximize comfort or other experiential qualities. Pareto-optimal solutions to such problems can be found using multi-objective evolutionary search together with other soft computing methods. Beyond optimality, professionals are interested in knowing how different aspects of the problem influence each other in optimal solutions. Such knowledge is referred to as the design principles. Through them decisions can be taken with great confidence and knowledge for similar future design cases is gained. Using the Pareto-optimal solutions for this purpose is known as innovization and it has been exercised for various crisp engineering problems. In the present paper automated innovization is used to discover the principles for a soft decision making problem. The process involves the use of a grid-based clustering technique integrated with a genetic algorithm for unsupervised learning of the principles. Multiple design principles are discovered simultaneously through a niching strategy. The large number of variables originating from the softness of the problem poses an additional challenge of parsimonious knowledge representation for ensuring interpretability. The problem investigated is a real-world decision making task concerning the optimal placement of a number of residential units in an urban design, involving two soft objectives: the recuperative quality of the neighborhood, as well as its living comfort should both be maximized. The underlying design principles are obtained and interpreted from the point of view of the decision-maker. This demonstrates the relevance of evolutionary knowledge discovery in decision-making, a matter which should provide decision-makers adequate and informed knowledge for choosing a single preferred solution among the Pareto-optimal ones, and also to understand intricate trade-offs among decision variables.

KEY WORDS: innovization, architectural design, design principles, decision making, multi-objective evolutionary algorithm, fuzzy neural tree

1 Introduction

Many engineering problems can be characterized as crisp. This means that the problem does not require means to address imprecision, uncertainty or vagueness. Examples of such problems are the well known engineering design problems, such as design of truss structure, ammonia reactor design, car suspension design etc. (Deb, 2001). In such problems the objectives refer physical object features, such as minimizing stress or material volume that are unambiguously defined in computational or mathematical terms. However, in other problems, particularly when the object of concern facilitates complex human experiences regarding products, services, buildings or cities, more often than not, objectives are described in linguistic terms that are not readily amenable for computation. Examples of such objectives are when an object is required to be functional, sustainable, or posses aesthetic qualities. Such concepts contain some imprecision or vagueness, so that they are referred to as soft. The softness is due to the abstract nature of the linguistic labels, which stem from human perception and cognition (Zadeh, 1997). Soft problems are an important class of decision making problems. This is seen, for instance from the fact that a substantial portion of the energy and materials consumed on the globe is through products and buildings, while decision making on such objects traditionally involves perception-based objectives. Thus decision making in industrial and architectural design is generally a soft issue. Due to their ill-defined nature, soft problems require special methodologies to deal with them, in particular the methodologies from the domain of soft computing, including fuzzy, neural, and evolutionary computation. These methodologies are able to absorb the imprecision inherent to soft decision making problems, since the underlying mathematical structures are able to handle a high degree of non-linearity, and generally some from of machine learning is used to establish the models (Broomhead and Lowe, 1988; Hunt et al., 1996). Due to their mathematical complexity, and as many real-world decision making tasks involve softness, it is an important matter to identify the invariant principles characterizing a soft problem. In particular in this work we investigate this issue using a post optimality analysis known as automated innovization (Bandaru and Deb, 2011b). This is studied by means of an application, where Pareto-optimal solutions of a soft problem are analyzed using the clustering based innovization approach.

The paper is structured as follows. In the rest of this section the concept of a design principle and the approach used for deciphering them are explained. In Sec. 2 the clustering based automated innovization method is described, followed by an illustration of its relevance to and importance in decision making. Thereafter in Sec. 3 we consider a soft multi-objective architectural design problem as a case study and develop a detailed neuro-fuzzy approach for modeling the inherent softness. In Sec. 4 we decipher several principles using the automated innovization approach and assess them against expert opinion. This is followed by conclusions.

1.1 Deciphering design principles

The invariant principles mentioned above are so called because they apply to all or most Pareto-optimal solutions of a multi-objective optimization problem, soft or otherwise. Also, they are unique only to these solutions and are not satisfied by other feasible or infeasible solutions. Thus they are characteristics of the Pareto-optimal front and their knowledge is crucial not only for the designer but also for the decision maker, as will be illustrated in Sec. 2.4. In this context these principles have been conventionally referred to as *design principles* (Deb and Srinivasan, 2006). These principles provide the recipe for creating more Pareto-optimal solutions. Moreover, they can be used to identify the most significant and least significant parameters of the problem.

Identifying commonalities or invariant principles among a given set of solutions is a challenging task. Firstly, it is difficult to predict the mathematical or logical form of the principles. Most notable of past studies have used decision trees and data-mining techniques to overcome this problem. Instead of assuming a definite form for the principles, decision trees work by successively dividing the dataset into segments based on each variable one at a time. Data-mining methods like SOMs provide a visual means of deciphering the commonalities. However, in both these cases the obtained knowledge is in a form that cannot be used by designers and decision makers effectively. Secondly, a given principle may not be applicable to the whole dataset. However, it is desirable that the principle encompasses a significant portion of the dataset. Decision trees fail when applied to such datasets, often resulting in a large number of terminal nodes. Lastly, the computational nature of obtaining the trade-off solutions in most if not all cases, introduces approximations and hence noise in the dataset. Any algorithm that aims to automatically extract the design principles must be able to filter out points which do not comply with the principle in hand.

1.2 Innovization

Unlike other methods, *innovization* (Deb, 2003) focuses on the mathematical structure of the principles. The goal of any innovization study is to identify different combinations of variables, constraints and/or objectives such that the resulting mathematical expression, when evaluated for all solutions in the dataset, remains constant or invariant across most of them. By definition, these combinations then become the design principles of the multi-objective problem. In this context, a design principle denotes a mathematical relationship between two or more entities of a problem, which is valid for a significant portion of the Pareto optimal front. Problem entities include the decision variables, the objectives as well as the constraint functions. The easiest way to obtain these principles is to manually choose various pairwise combinations of variables, constraints and objectives and plot them against each other in two dimensions to see if there is any correlation. This is known as manual innovization. For example, if a variable x_1 when plotted against an objective f_1 results in the data points falling approximately on a straight line with positive slope, then $x_1 \propto f_1$ and therefore $\frac{x_1}{f_1} = constant$ becomes the design principle. There are however, some glaring limitations to this approach. Firstly, it is difficult to manually plot and analyze all combinations of problem entities especially when a large number of variables are involved. The method of identification of design principles is visual and hence is not reliable if the problem entities under consideration have different magnitudes or if there is a subset of points on which the principle is not applicable. Thirdly, linear correlations, though easy to identify, are rare and hence there is a need for preprocessing the data using transformations (eg. logarithmic) and post-processing with regression techniques. Lastly, the human element in the manual innovization procedure makes the process prone to errors. In the following section we discuss the clustering based automated innovization approach developed to circumvent these problems.

2 Clustering based automated innovization

Automated innovization (Bandaru and Deb, 2010), proposes the use of the following mathematical structure for the design principles,

$$\Pi_{j=1}^{N} \phi_j(\mathbf{x})^{a_j b_j} = c. \tag{1}$$

This mathematical form is supported by previous manual innvoization studies like Deb and Srinivasan (2006). Additionally, it is easy to see that such a form easily lends itself for interpretation and future reference. Here ϕ_j's are the N chosen problem entities. They can also be any other user defined functions. Since they form the basic units which form a design principle, they are also called the basis functions (not to be confused with the basis functions used later for neuro-fuzzy modeling). The constant c, called the parametric constant, quantifies the degree of invariance of the principle defined by the expression on the left hand side of Eq. (1).

a_j's are Boolean variables which represent the presence (1) or absence (0) of the j-th basis function. Given a set of a_j's the automated innovization algorithm tries to find the corresponding exponents b_j's such that Eq. (1) represents a *significant* design principle. Since the c-values quantify the invariance, a valid design principle should have the same value of c for a relatively large number of Pareto-optimal solutions. As discussed earlier, this design principle may not apply to the whole Pareto-optimal front The percentage of points which have very close c-values can be used to calculate the significance of a design principle.

2.1 Grid-based clustering

Before calculating the significance described above, the trade-off points have to be classified into two groups: one which has the same (or very close) c-values and the other which have c-values very different from rest of the dataset. Automated innovization uses the grid-based clustering technique in the one dimensional space of c-values. It involves the following steps:

1. Sort $\{c^{(1)}, c^{(2)}, \ldots, c^{(m)}\}$ evaluated for the m trade-off solutions.
2. Divide the range $[c_{min}, c_{max}]$ into d divisions.
3. Count the number of c-values n_d falling within each division.
4. Label the divisions with $n_d \geq \lfloor \frac{m}{d} \rfloor$ (the average number of c-values per division) as sub-clusters.
5. Label the trade-off points corresponding to the c-values in the remaining divisions as unclustered.
6. Merge adjacent sub-clusters to form $\mathcal{C}$ clusters.
7. Count the number of unclustered trade-off points $\mathcal{U}$.

Notice that the grid-based clustering described above introduces a parameter d. It can take any value in the range $[1, m]$. However, instead of asking the user to set a value for it, the automated innovization algorithm is designed to choose an optimal value for it based on the accuracy of the deciphered design principle. Once the trade-off points are classified into the above defined groups the significance can now be obtained as,

$$S = \frac{m - \mathcal{U}}{m} \times 100\%. \tag{2}$$

2.2 Deciphering a single design principle

It is clear from the discussion above that the c-values in the $\mathcal{C}$ clusters found above should all have approximately the same value. The algorithm is expected to find suitable a_j's and b_j's such that the invariance property of the design principle is satisfied. The coefficient of variance c_v of the c-values is used for measuring this invariance. For converging iteratively to a design principle and hence improving its significance, an optimization problem is formulated with the minimization of the sum of c_v's in all clusters as the objective. The exponents a_j's and b_j's are the variables. The parameter d of the clustering algorithm is also included in the variable set, and its value is optimized for minimum number of clusters. Since within each cluster, the c_v is minimized, the accuracy of the design principle can be indirectly controlled through the variable d. The percentage coefficient of variation is used instead of c_v so that the following weighted objective function can be used,

$$\text{Minimize} \quad \mathcal{C} + \sum_{k=1}^{\mathcal{C}} c_v^{(k)} \times 100\%. \tag{3}$$

Notice that the c-values on the RHS are only used for clustering and their numerical values are irrelevant. Therefore, there can be infinite sets of optimal exponents b_j's thats will result in the same number of clusters. The objective function in Eq. (3) is hence multi-modal. We overcome this, by restricting the variables b_j to the range $[-1, 1]$ by always ensuring that the exponent with largest absolute value is always 1 and that the other exponents are modified proportionately if needed. For example, a design principle of the form $\phi_1^{2.5}\phi_2^{1.0}\phi_3^{-4.0} = c$, will be represented in the automated innovization algorithm as, $\phi_1^{-0.625}\phi_2^{-0.25}\phi_3^{1.0} = c'$.

2.3 Niching for preserving multiple design principles

The approach discussed above is only suitable if the user wants to decipher only the most significant design principle. Often, the user is interested in finding all design principles which are above a threshold significance value. Previously, manual and automated innovization (Bandaru and Deb, 2011b) studies achieved this through multiple runs with different sets of basis functions. A recent study (Bandaru and Deb, 2011a) exploits the population based nature of the GA to evolve multiple design principles simultaneously using a less known niching strategy. Since each population member represents a possible design principle, for the i-th individual, Eq. (1) can be written as,

$$\Pi_{j=1}^{N} \phi_j(\mathbf{x})^{a_{ij} b_{ij}} = c_i. \tag{4}$$

The significance S_i can now be calculated by first clustering all c_i-values to determine $\mathcal{U}_i$ and then using Eq. (2). Population members with different sets of a_{ij}'s represent design principles which use different basis functions and hence they are not compared during the selection operation of GA. Thus different 'species' of design principles can be made to coexist in the population. Boolean a_{ij}'s are encoded as a binary string of length N and a GA which can handle both real and binary variables is adopted.

In order to prevent trivial solutions (when $a_{ij} = 0 \ \forall \ j$) and complex design principles (when many basis function are involved), the constraint $1 \leq \sum_j a_{ij} \leq \mathcal{N}$ is introduced in the problem formulation, whose complete form (for the i-th design principle) is:

$$\begin{array}{lll} \text{Minimize} & \mathcal{C}_i + \sum_{k=1}^{\mathcal{C}_i} c_v^{(k)} \times 100\%, & \\ \text{Subject to} & S_i \geq S_{reqd}, & \\ & 1 \leq \sum_j a_{ij} \leq \mathcal{N}, & \text{1 binary variable} \\ & -1.0 \leq b_{ij} \leq 1.0, & N \text{ real vaiables} \\ & 1 \leq d_i \leq m, & \text{1 integer variable.} \end{array} \quad (5)$$

where S_{reqd} and $\mathcal{N}$ are user-supplied values indicating respectively, the minimum threshold significance for the variables and the maximum number of basis functions that can participate in forming a design principle. The mixed variable nature of the problem also justifies our use of a GA to solve Eq. (5). The constraints are handled using Deb's penalty-parameter-less approach (Deb, 2000).

2.4 Innovization for decision making

In this section, we shall illustrate the role of innovization in decision making through two well known engineering design problems. Though the problem formulations in both the cases are crisp, the procedure is applicable as is to soft problems.

2.4.1 Brushless DC motor design

A detailed manual innovization study for the design of a brushless DC permanent magnet motor design can be found in Deb and Sindhya (2008). The design requires minimization of manufacturing cost of the motor along with maximization of the peak output torque. The five design variables are: the number of laminations to be used in the motor (n_l), the number of turns in each coil (N), lamination type (L_{type}), type of electrical connection (M_{ph}) and the wire gauge (A_{gauge}) to be used in the windings. The optimization problem is posed as:

$$\begin{array}{ll} \text{Minimize} & C_{total}(n_l, N, L_{type}, M_{ph}, A_{gauge}), \\ \text{Maximize} & T_p = 87300.00\, C_{tor} N R_{si} A_{wire} n_l, \\ \text{Subject to} & g_1(\mathbf{x}) = T_p \geq 0.83, \\ & g_2(\mathbf{x}) = T_p \leq 5.27, \\ & g_3(\mathbf{x}) = A_{wire} N \leq \{150, 240, 280\} \times 10^{-7} \\ & \qquad \text{when } L_{type} = \{A, B, C\} \text{ respectively,} \\ & 20 \leq n_l \leq 200, \\ & 10 \leq N \leq 80, \\ & L_{type} \in \{A, B, C\}, \\ & M_{ph} \in \{Y, \Delta\}, \\ & A_{gauge} \in \{16.0, 16.5, \ldots, 23.5\}, \\ & n_l \text{ and } N \text{ are integers, } A_{gauge} \text{ is discrete.} \end{array} \quad (6)$$

The cost function C_{total} can be found in Deb and Sindhya (2008). A, B and C are different kinds of laminations with specific R_{si} values. The electrical connection can be of Y type or Δ type, each having a different C_{tor} value. Finally, the cross-section are of the wire A_{wire} is a function of the gauge number A_{gauge} and hence can also take 16 discrete values.

Fig. 1 shows the Pareto-optimal front obtained using local search based NSGA-II (Deb et al., 2002). Interestingly, the Pareto-optimal solutions all differ only with respect to the number of laminations n_l which takes all integer values in the range $[28, 172]$. All other variables are fixed for the entire set of Pareto-optimal solutions. $N = 18$, $L_{type} = B$, $M_{ph} = Y$ and $A_{gauge} = 16.0$ are simple but very useful design principles deciphered for this problem. A decision maker with knowledge of these principles can enhance the productivity of a manufacturing unit producing these motors by simply setting the wire turning machine to a fixed value of 18. The armatures will thus never have over-turns or under-turns. The inventory and the associated costs can also be reduced by only ordering laminations of type B and wire of gauge 16.0.

2.4.2 Spring design

In this design task, a helical compression spring needs to be designed such that both its volume and developed stress are minimized. The wire diameter d, the mean

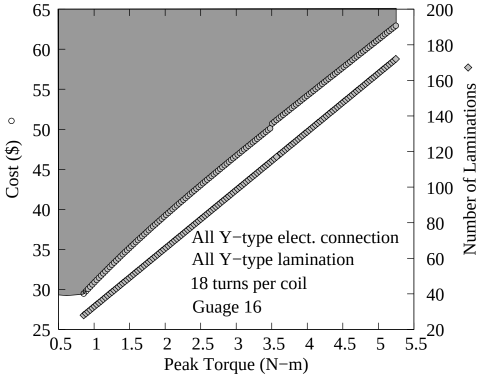

Figure 1: Pareto-optimal front and innovized principles for brushless DC motor design.

coil diameter D and the number of turns N are the design variables. The optimization problem formulation is as follows. A detailed description of all the symbols can be found in Bandaru and Deb (2011b).

$$
\begin{array}{ll}
\text{Minimize} & f_1(\mathbf{x}) = V = 0.25\pi^2 d^2 D(N+2), \\
\text{Minimize} & f_2(\mathbf{x}) = S = \frac{8KP_{max}D}{\pi d^3}, \\
\text{Subject to} & g_1(\mathbf{x}) = l_{max} - \frac{P_{max}}{k} - 1.05(N+2)d \geq 0, \\
& g_2(\mathbf{x}) = d - d_{min} \geq 0, \\
& g_3(\mathbf{x}) = D_{max} - (d+D) \geq 0, \\
& g_4(\mathbf{x}) = C - 3 \geq 0, \\
& g_5(\mathbf{x}) = \delta_{pm} - \delta_p \geq 0, \\
& g_6(\mathbf{x}) = \frac{P_{max}-P}{k} - \delta_w \geq 0, \\
& g_7(\mathbf{x}) = S - \frac{8KP_{max}D}{\pi d^3} \geq 0, \\
& g_8(\mathbf{x}) = V_{max} - 0.25\pi^2 d^2 D(N+2) \geq 0, \\
& 1 \leq N \leq 32, \\
& 1 \leq D \leq 30 \text{ in.}, \\
& N \text{ is integer, } d \text{ is discrete, } D \text{ is continuous.}
\end{array}
\tag{7}
$$

The variable d is allowed to take any of the following 42 discrete values:

$$
\begin{pmatrix}
0.009, & 0.0095, & 0.0104, & 0.0118, & 0.0128, & 0.0132, \\
0.014, & 0.015, & 0.0162, & 0.0173, & 0.018, & 0.020, \\
0.023, & 0.025, & 0.028, & 0.032, & 0.035, & 0.041, \\
0.047, & 0.054, & 0.063, & 0.072, & 0.080, & 0.092, \\
0.105, & 0.120, & 0.135, & 0.148, & 0.162, & 0.177, \\
0.192, & 0.207, & 0.225, & 0.244, & 0.263, & 0.283, \\
0.307, & 0.331, & 0.362, & 0.394, & 0.4375, & 0.5.
\end{pmatrix} \text{in.}
$$

Automated innovization yields various design principles for the Pareto-optimal set of this problem (Bandaru and Deb, 2011b). However, the most important for the decision maker is the principle $d = constant$ which results in seven clusters as shown in Fig. 2. Out of the 42 allowed values for the spring wire diameter, only the last seven are optimal. This information is very crucial to the designer and the decision maker who can now propose to reduce the inventory size of the wires from 42 different groups to just 7. Using such a cluster plot and its mapping on the Pareto-optimal front, the decision maker can also deduce that choosing one of the seven optimal values of d is the highest level of decision making when an optimal design has to be selected.

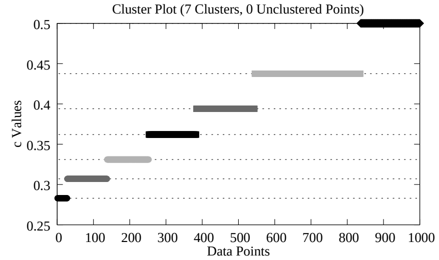

Figure 2: Cluster plot for the design principle $d = constant$.

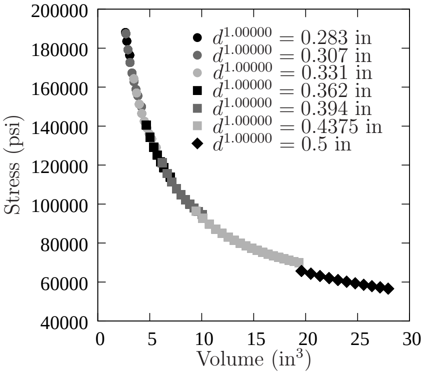

Figure 3: Pareto-optimal front for the spring design problem. The clustered points in Fig. 2 fall in different segments on the Pareto-optimal front.

3 Architectural design problem

3.1 Softness and relevance

The present problem is from the domain of architectural design and concerns a typical real-world decision making case in this domain. A decision is to be made regarding optimal positions of an ensemble of residential housing units on their respective lots, where two objectives are subject to maximization. The first objective is to provide a *comfortable living experience* to the residents by maximizing the visual privacy they experience in their homes. The second objective is to maximize the *recuperative quality* of the neighbourhood as determined by the suitability of the gardens for this purpose. These criteria are major aspects in architectural and urban decision making tasks. That is, living comfort and recuperative performance are significant features determining the quality of life for the residents and the value of an area, so that these issues frequently occur as objectives to be maximized in these tasks. It is noted that in the decision making task considered in this work, aspects such as social control for safety of residents are omitted, as these issues do not play a significant role in the problem at hand. The building site is one of the largest areas in the Netherlands subject to development, named Leidsche Rijn. The site has a size of $3600m^2$. The streets and lots are provided in advance in this case.

In Fig. 4 the building site is seen from plan view. On the site twenty buildings are situated. Three of them already exist on the site. These buildings are shaded with black colour in the figure. Thus, seventeen houses are subject to optimal positioning. The houses G_a1-G_a6 and G_b1-G_b4 form two groups of houses, respectively termed G_a and G_b, which are situated along a line parallel to the north and south perimeter of the neighbourhood, as seen from Fig. 4.

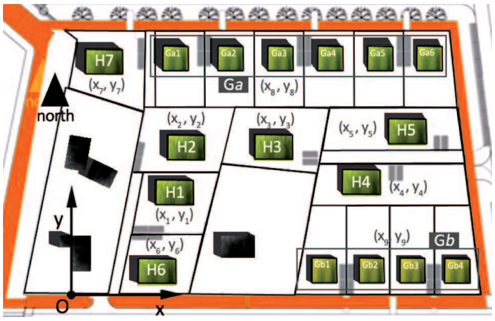

Figure 4: Decision variables of the problem.

It is an initial basic choice of the decision makers to align the houses of a group G_a or G_b with respect to each other, so that their y coordinates are the same, and also the distances among the houses in x direction are kept constant. Therefore, a single x-y coordinate pair suffices per group - respectively (x_8, y_8) and (x_9, y_9) - to describe the location of the houses. The houses of G_a and G_b have a square shaped floor plan of $8m \times 8m$, while houses $H1$-$H7$ are $12m$ long, $8m$ wide and their longer axis is oriented in east-west direction as seen from Fig. 4. All buildings are two storeys high. It is noted that the existing buildings do play a role in the assessment of the performance of the designs, although they are not subject to optimal placement. This is because they influence the visual privacy of a number of buildings to be positioned, thereby influencing the living comfort of the neighborhood and the performance of the neighbourhood. The objective for high living comfort of the neighborhood entails that all residential units should have a high visual privacy, meaning that every unit should be minimally exposed to visual perception from the other buildings around it. Next to this, the recuperative quality of neighbourhood is maximized by placing the houses in such a way that the gardens are as large as possible in the desirable cardinal direction for all houses of the neighborhood. One aspect of the softness of this problem is that the living comfort is determined by visual perception aspects, the human visual perception process is complex, leading to an inherent uncertainty in the assessment of visual perception features of objects.

3.2 Computing visual perception information

Visual perception is deemed a soft issue, because brain processes play an essential role in it. Therefore, up till now visual perception based requirements have generally not been subject to computational treatment, although they play an important role for the qual-

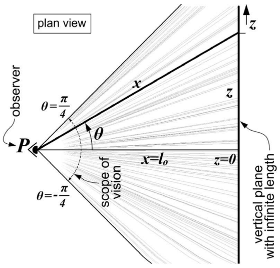

Figure 5: Probabilistic perception model for a basic geometric situation. Figure taken from Bittermann et al. (2007).

ity of a design. In the decision making problem addressed in this work treatment of this issue is exemplified by computing visual perception information based on a probabilistic perception theory (Bittermann et al., 2007). In this theory the faculty of consciousness known as *attention* in psychology, is modeled by means of probability density function (pdf), while perception is modeled by means of a probability being a scalar quantity. This probability is obtained by integrating the pdf that characterizes visual attention over some physical domain of an object. This is exemplified for the basic geometry in Fig. 5, where an observer at P is viewing a plane with infinite length.

Taking the scope of vision in this situation to be defined by the angle $-\pi/4 \leq \theta \leq \pi/4$, the probability density function f_θ that characterizes unbiased attention with respect to the angle θ in Fig. 5 is given by

$$f_\theta = \frac{1}{\pi/2} \tag{8}$$

as a uniform pdf. This implies that the observer has no a-priori bias for any direction in his view, which reflects the lack of information on such preferences in a general viewing situation. Based on Eq. (8) and the geometric relation $\tan(\theta) = z/l_0$, probability theoretic computations yield the function of the random variable z for the interval $-l_0 \leq z \leq l_0$ as (Bittermann et al., 2007),

$$f_z(z) = \frac{2}{\pi}\frac{l_0}{(l_0^2 + z^2)} \tag{9}$$

which is a Cauchy function shown in Fig. 6 for $l_0 = 4$, $l_0 = 6$ and $l_0 = 8$.

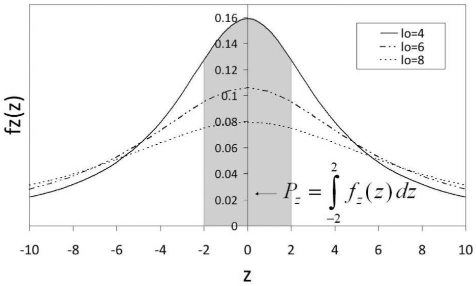

Figure 6: Plot of $f_z(z)$ for $l_0 = 4$, $l_0 = 6$ and $l_0 = 8$.

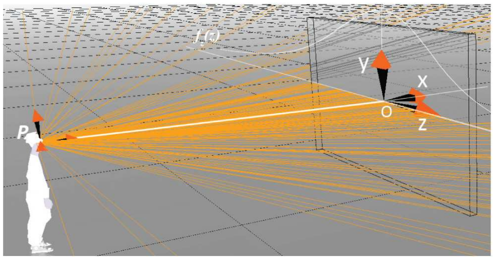

Figure 7: Illustration of the result from the perception model.

From the plot it is seen that the maximum of attention is located directly in front of the observer, i.e. at $z = 0$, where the distance x to the object is minimal, namely $x = l_0$. This result is confirmed from common vision experience: For instance when we visually experience a wall without an *a priori* bias for any part of the wall, we naturally pay more attention to the details of the wall region that is directly in front of us compared to adjacent regions. From the plot we also note that with increasing distance from the plane the attention $f_z(z)$ becomes less peaked. This means for a far distance, instead of paying attention primarily to the middle part of the wall, an observer gives attention to the whole object with almost equal intensity, i.e. attention is less accentuated for the frontal region of the object compared to to the case of a nearer distance.

The integration of visual attention paid to an object yields perception, which is quantified by a probability (Ciftcioglu et al., 2006). This probability can be interpreted as the degree of awareness the observer has for the object attended to. For simplicity of the explanation, preliminarily perception is investigated in the z-x plane seen in Fig. 7. In case of the infinite plane in Fig. 5 being the single object in the scene, the integration yields

$$\int_{-l_0}^{+l_0} f_z(z)dz = \frac{2l_0}{\pi}\int_{-l_0}^{+l_0} \frac{dz}{l_0^2+z^2} = 1 \qquad (10)$$

so that $f_z(z)$ is verified as a pdf. This means that in case there is a single object fully spanning the visual scope, then the probability that the object is seen is unity, i.e. the observer is aware of the object's presence with certainty. For an object that is small enough, so that it does not span the entire visual scope, perception of the object is computed by

$$P_z = \int_{b_1}^{b_2} f_z(z)dz, \qquad (11)$$

where b_1 and b_2 are the boundaries of the object in z-direction. For illustrative purposes the perception of the wall object shown in Fig. 7 is calculated, where $l_0 = 4m$ and the object size is also $4m$ in z direction, so that the perception becomes

$$\begin{aligned} P_z &= \int_{-2}^{+2} f_z(z)dz = \frac{4l_0}{\pi}\int_{0}^{+2} \frac{dz}{l_0^2+z^2} \\ &= \frac{4}{\pi}\tan^{-1}\frac{2}{l_0} \approx 0.591. \end{aligned} \qquad (12)$$

The integration for this case is illustrated in Fig. 6 by means of the shaded area. For the perception in three dimensional space the perception in y direction should be considered as well. Due to the probabilistic nature of the approach the perceptions in either direction are independent events. Therefore the perception of the object in space is obtained by using the multiplication rule $P(P_z \cap P_y) = P_z P_y$, i.e. multiplying the probability quantifying perception in z direction and perception in y direction (Bittermann and Ciftcioglu, 2008). For the wall object in Fig. 7, having a height of $3.0m$ the perception in y direction becomes

$$\begin{aligned} P_y &= \int_{-1.5}^{+1.5} f_y(y)dy = \frac{4l_0}{\pi}\int_{0}^{+1.5} \frac{dy}{l_0^2+y^2} \\ &= \frac{4}{\pi}\tan^{-1}\frac{1.5}{l_0} \approx 0.457. \end{aligned} \qquad (13)$$

Therefore the visual perception of the object is

$$P_{y,z} = P_y P_z \approx 0.270. \qquad (14)$$

This means the object 'occupies' a significant portion of the visual awareness of an unbiased observer, namely more than a quarter of it. From the figure it is clearly noticed that an object located directly in front of the observer and nearer to him yields a higher degree of perception, due to the peaked shape of the Cauchy function. Since the sharpness of the peak depends on the distance l_0, as seen from Fig. 6, the integral, i.e. perception also diminishes as the object moves away form the observer.

Clearly due to the geometry of the Cauchy function, if an object appears not directly in front of the observer but lateral to the observer's central viewing line, the object will also have reduced perception compared to the former case. This is illustrated in Fig. 8, where several perception computations are shown. For the sake of simplicity the vertical dimension is omitted in the following consideration. The perception of the buildings $H1$, $H2$, $H3$, and $H4$ is investigated. Here the viewpoint of the observer is taken as the geometric centre point of the north facade of a building $E1$. The curves plotted along the z axis illustrate the probability density functions belonging to the perception of houses $H1$-$H4$ as viewed from $E1$. In particular they show the degree of visual attention paid to these buildings' south facades for the z dimension. The integral of these pdfs over the length of the respective south facade of a house is indicated as a shaded area. It quantifies the perception of the respective facade from the viewpoint considered. Comparing the perception of house $H3$ and $H2$ it is clear that $H3$ is perceived more strongly compared to $H2$, as $H3$ is located more close to the frontal direction marked by l_0 in the figure.

It is noted that every house is perceived from several viewpoints at the same time. From the decision making viewpoint it is desirable in the present case that the total perception of a building's facade from these viewpoints should be small. This is to increase the visual privacy experienced in the house. In this case this is relevant with respect to the south facade of the house, since the living areas of a house are generally

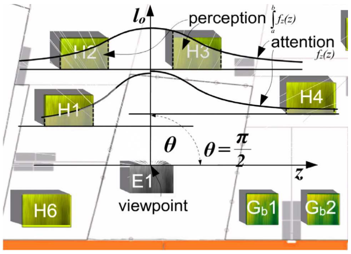

Figure 8: Sketch indicating the computation of the degree of perceptions of the houses $H1$, $H2$, and $H3$ from the viewpoint $E1$.

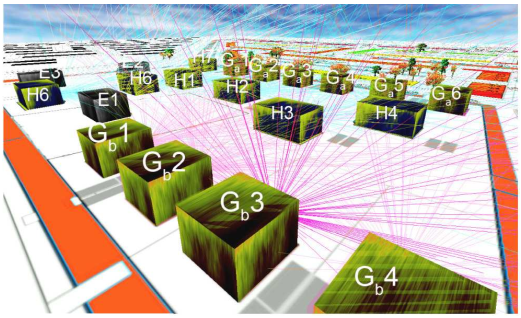

Figure 9: Illustration of the visual privacy computation based on the probabilistic perception model.

located behind the south facade in north western Europe. Therefore, the perceptions 'impinging' on the south facades of the buildings are considered. This is illustrated in Fig. 9. In the figure an implementation of the perception model described before is seen, where a random process is used to generate vision rays based on the pdf in Eq. (8). These are emitted in northern direction from the different houses according to the uniform pdf of attention with respect to the vision angle. Based on these considerations visual privacy of a facade is defined by

$$P_{priv}(O) = \left(\sum_{1}^{n} P(O, V_n) \right)^{-1} \tag{15}$$

where $P(O, V_n)$ is the degree of perception of object O from the n-th viewpoint V. That is, the visual privacy of a facade is considered the reciprocal of the sum of attention paid to the facade from the other houses.

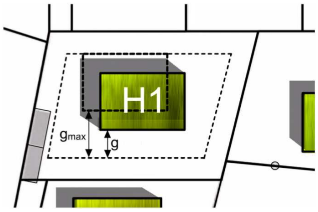

Figure 10: Calculation of the garden performance.

For the computation of the perceptions, in this implementation occlusion is considered by means of a simple test of the visibility of a building from another one. This is detected by sending a test-ray from the centre location of the first building to the original viewpoint. In case this ray is intercepted by a building located in between the two points, then the component of perception of this building from the viewpoint considered is taken to be zero.

3.3 Assessment of recuperative performance

A second aspect considered in the design of the housing complex is the performance of the neighborhood for recuperative purposes. In general a large garden located south of the building is considered most desirable for this purpose, due to exposure to direct sunlight in moderate climates on the northern hemisphere. Therefore the performance of a garden is obtained based on the size of the south garden in general. 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 for recuperation as well. In this case the west direction is used and not the east direction. This is done because the houses $H4$ and $H5$ should have direct sunlight in their gardens during the evening, as the occupants are expected to make use of their gardens mainly in the evening in this case. In order to determine the garden performance G for an individual house, the size of its garden in south direction is normalized with respect to the maximum size of the garden in this direction, i.e. $G = g/g_{max}$. This expresses to what degree a house is located so that it maximizes the size of the garden. 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 in north-south direction. This is illustrated in Fig. 10 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.

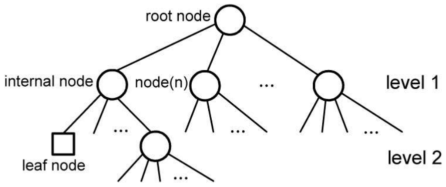

Figure 11: The neural tree structure.

3.4 Fuzzy modeling for treating soft objectives

The computations for determining the privacy of a facade and the performance of a garden described in the previous subsection are addressing the elemental aspects that characterize the performance of the design. Both demands, that the neighbourhood should have a high recuperative quality and that the living comfort should be high as well, are soft in nature. This is for two reasons. First, it is vague to some extend what is considered to be a 'high recuperative quality' although it is clear that it is determined by the largeness of the gardens, where it is desirable that all gardens are large in the desirable cardinal direction. Explicitly stated the demand all gardens should be large in the suitable directions implies that $G1$ should be large, AND $G2$ should be large, AND ..., AND GJ should be large for J number of gardens. The softness of this demand is clear considering for instance that in case the demand for largeness is not completely fulfilled concerning one of the gardens, this does not mean that the objective for large gardens for all houses is totally not fulfilled. It means that the objective is merely partly fulfilled. On the other hand, for instance summing up the actual sizes of the gardens and using this as an objective does not reflect the actual demand at hand. Namely in case one of the gardens would be undesirably small, summing up its size with the size of others that are large will obscure the information that one of the gardens is small, as it does not model the demand for simultaneous presence of largeness, i.e. it does not model a logical 'AND' condition, which is the base for the objective at hand.

To deal with the softness of the objectives of the problem, in this work a soft computing technique is used, which is neuro-fuzzy modeling employed in the framework of a neural tree structure. Using this method the suitability of a solution is estimated based on human-like reasoning. That is, the simultaneous presence of the desired features is measured by means fuzzy set and chained fuzzy logic operations embedded in a neural tree structure, as follows. A neural tree is composed of terminal nodes, also termed leaf-node and non-terminal nodes, also termed internal nodes. Nodes of a neural tree are connected by means of links, where the structure is built up in such a way that at least two leaf nodes are linked to one inner node, and at least two inner nodes are linked to another inner node one level above the former inner nodes. This way the amount of nodes per level reduces for levels that are more and more remote from the leaf node level. Ultimately the nodes on the uppermost level of the tree are linked to one or several root nodes of the tree, which generally act as model output. This structure is seen from Fig. 11.

In the fuzzy neural tree implementation the terminal nodes convert crisp input data into fuzzy information. This information is introduced to the inner nodes and then further propagated 'upwards' through the model. The inner nodes of a fuzzy neural tree model neural activity in a human brain. A neuron in the tree performs a non-linear mapping on the input information coming to it, simulating a neuronal activity. In the present case, the non-linearity of an inner node is established by means of Gaussian functions. The Gaussian function is of particular interest for the intended modeling purpose due to its relevance to fuzzy logic, namely using a Gaussian allows considering the model both as a neural model and as a fuzzy system (Hunt et al., 1996). Namely the Gaussian function plays the role of activation function when we consider the model from the viewpoint of the neural network paradigm, and at the

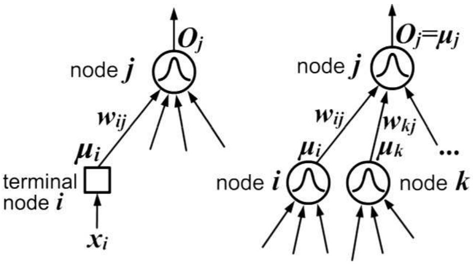

Figure 12: Different type of node connections for the fuzzy information processing executed at the neural tree nodes.

same time it plays the role of membership function in the terminology of fuzzy information processing. Due to the latter, a fuzzy neural tree using Gaussian functions at the inner node represents a fuzzy logic system, i.e. it performs chained logic operations mimicking human-like reasoning.

Using a neural tree for evaluating the suitability of a decision, the root node of a neural tree represents the ultimate goal subject to maximization, namely in our case the design performance of the neighbourhood. The tree branches form the objectives constituting this ultimate goal - in our case maximizing recuperative and living comfort performance. The connections among the inner nodes and terminal nodes have a weight w_{ij} associated with them, as seen from Fig. 12. The weights are given by a decision maker. They specify the relative significance a node i has for the node j, which is situated one level closer to the root node in the neural structure compared to i.

The weight of a link connecting node i to node j denotes to what extent, relative to other inputs of j, the linguistic label associated to i is to be considered as a constituent of the concept associated with node j. The fuzzy neural tree is used to model knowledge of a decision maker. In the present case the knowledge includes that high recuperative quality of the neighbourhood means simultaneous presence of several object features, such as high privacy for every house in the neighbourhood. Therefore the operation performed at an inner node j is an AND operation that is similar to the computations in radial basis function (RBF) neural networks. In RBF networks the Gaussian function is used as basis function, given by

$$f(X) = w\Phi(||X - c||^2) \tag{16}$$

where $\phi(.)$ denotes the Gaussian basis function, and c denotes the centre of the basis function. The width of the basis function μ_j at node j is used to measure the uncertainty associated with the input vector X_{jj} to the node j. For the AND operation at node j the multiplication rule is used, so that for n number of inputs to node j, where every component of the input vector X_{ij} is denoted by x_{ij}, the nodal operation is given by

$$O_j(X_j) = e^{-\frac{1}{2}\sum_{i=0}^{n}\left(\frac{x_{ij}-c_{ij}}{\sigma_j}\right)^2}. \tag{17}$$

It is noted that in Eq. (17) the multiplication rule is expressed by means of the summation at the exponent of Euler's number for every input and centre. The input to the basis function at inner node j denoted by X_{ij} is related to the output μ_i from node i by the relation

$$X_{ij} = \mu_i w_{ij} \tag{18}$$

where w_{ij} is the weight connecting μ_i to the node j. The weight w_{ij} expresses the relative importance of the i-th input μ in the AND operation. It is noted that the weights w_{ij} for all inputs to node j sum up to unity, as the weights represent the relative importance among the inputs to the node. Using the radial basis function neuron for the AND operation, the centres c_{ij} of the Gaussian basis functions are set to take the same value as the input weights w_{ij} to the node. This way it is ensured that for an input $\mu_i = 1.0$, i.e. when a elemental requirement is fully satisfied, the membership degree used in the multiplication representing the AND is also maximal, namely 1.0. This inherently ensures that when all elemental requirements are totally satisfied, i.e. $\mu_1 = \mu_2 = \ldots = \mu_n = 1.0$, then the output O_j of the inner node is also 1.0, meaning the requirement modeled by node j is completely fulfilled. This is seen when we use Eq. (18) in Eq. (17), as well as $c_{ij} = w_{ij}$, so that we have

$$O_j = e^{-\frac{1}{2}\sum_{i=0}^{n}\left(\frac{w_{ij}\mu_i-w_{ij}}{\sigma_j}\right)^2}, \tag{19}$$

which can be written as

$$O_j = e^{-\frac{1}{2}\sum_{i=0}^{n}\left(\frac{w_{ij}(\mu_i-1)}{\sigma_j}\right)^2}, \tag{20}$$

Eq. (20) can be expressed in the following form

$$O_j = e^{-\frac{1}{2}\sum_{i=0}^{n}\left(\frac{\mu_i - 1}{\sigma_j / w_{ij}}\right)^2}, \qquad (21)$$

From Eq. (21) it is clearly seen that when all inputs are $\mu_i = 1.0$ then O_j becomes unity inherently. Also it is noted that the width parameter of the Gaussian σ is scaled by the weight w_{ij} associated with the i-th input.

It is noted that due to the particularity of the neural tree structure, only the left half side of the Gaussians at the inner nodes are used during the logic operations. Therefore the inner nodes represent a multivariate increasing function. This ensures that the greater a membership value μ_i is belonging to an aspect at the input to a radial basis function, the greater the node output will be. This implies that improvements with respect to any elemental requirement in the decision making yields an improvement of the general performance. This property is seen from Fig. 13, where the logic operations occurring in every inner node are illustrated. The operation at an inner node is a two-stage process. First the information μ_i coming into an inner node is subjected to a non-linear mapping using the Gaussian membership functions $g_i = f(\mu_i)$ in Fig. 13. Explicitly we can write these basis functions for every input as

$$g_{ij}(\mu_i) = e^{-\frac{1}{2}\left(\frac{\mu_i - 1}{\sigma_j / w_{ij}}\right)^2}, \qquad (22)$$

so that the output O_j of the j-th inner node is obtained by $O_j = g_{1j} g_{2j} \ldots g_{nj}$. The basis functions g_{ij} can be considered as *activation functions* in the terminology of artificial neural networks. Before the multiplication, the membership function g_{ij} signifies the closeness to the full satisfaction of the input at unity. That is, g_{ij} models the fuzzy set of decisions satisfying the i-th elemental decision requirement. Clearly a wider Gaussian implies greater tolerance for deviation from full satisfaction at $\mu_i = 1.0$. This width is in proportion to the significance w_{ij} of an input coming to the node as seen from Eq. (21). This means, an input having a high significance implies a Gaussian with a small width, so that small deviation from the maximum input value, i.e. $\mu_i < 1.0$, will already yield a significantly smaller value for g. Conversely, when the significance of an input is relatively small, then the associated Gaussian basis function is relatively wide, so that a small deviation from $\mu_i = 1.0$ yields still a value close to unity for g_{ij}. The functionality of this becomes clear when as we consider the ensuing AND operation being executed by means of multiplication. This is illustrated in Fig. 13 for a condition with equal weights. A small value for $g(\mu)$, i.e. a value nearby zero will have a significant impact in the multiplication, drastically reducing the output $O_j = g_1 g_2 \ldots g_n$. This means for an input with great relative importance, the node will react relatively sensitive to the fulfilment of the requirement modeled by this input, compared to the other inputs.

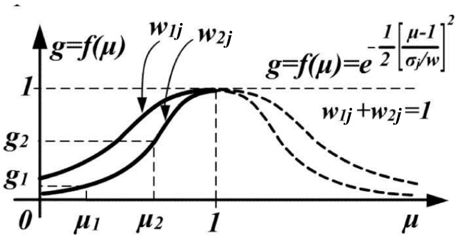

Figure 13: Fuzzification of an input at an inner node.

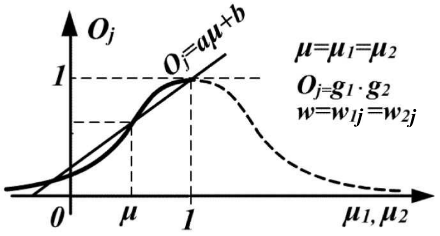

Figure 14: Linear approximation to Gaussian function at AND operation.

The parameter σ is the only parameter left to be determined in the model. It should be selected in such a way that the following condition holds: If all requirements are satisfied by the same degree, for instance all requirements are 50% satisfied - i.e. $\mu_1 = \mu_2 = \ldots = \mu_n = 0.5$, then it is consistent for our modeling purpose that the output for this solution should also be $O_j = 0.5$. It is noted that this condition is irrespective of the pattern of weights associated to the inputs of the node j. In the same way, in general when $\mu_1 = \mu_2 = \ldots = \mu_n = a$, then it follows that $O = a$.

Table 1: Input-output datasets used for consistency establishment for 6 inputs per node.

Dataset	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	O_j
1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
9	0.9	0.9	0.9	0.9	0.9	0.9	0.9

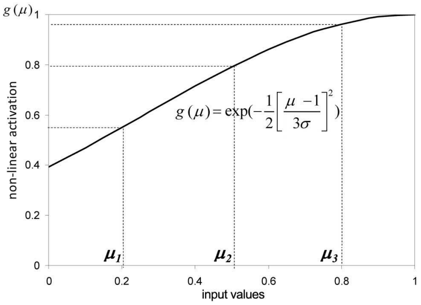

Figure 15: Three identical Gaussian basis functions for a node with three inputs that are equally significant.

This condition is termed as *consistency condition* in this work, and it is used to establish the width parameter σ of the Gaussian basis function (Ciftcioglu et al., 2007). That is, the width is selected in such a way that the deviation from the consistency condition for a number of input data sets is minimal. This is accomplished by means of classical optimization. The input-output datasets used for the consistency establishment in the case of six inputs to a node is given in Table 1.

It should be noted that the present neural modeling approach is in contrast to artificial neural networks as follows. Establishing an artificial neural network is based on data samples and training to minimize the model error in the representation of the dataset. The neural trees used in the present approach are established from expert knowledge and applying a consistency condition. That is, the structure of the tree and the weights are given based on domain knowledge and the model parameters are established by means of a classical optimization algorithm. It is noteworthy that the neural model used in this work is completely knowledge-driven and involves non-linearity due to the Gaussians involved, at least for the non-terminal nodes.

To illustrate the operation occurring at an inner node with three inputs, three identical optimal basis functions $g_{ij}(\mu_i)$ are shown in Fig. 15, belonging to the weights $w_{1j} = 1/3$, $w_{2j} = 1/3$ and $w_{3j} = 1/3$, that is the three inputs are equally important constituents of a node j.

From Fig. 15 it is noted that in case all weights are equal then clearly all basis functions are identical, in particular they have the same width σ/w. In Fig. 15 an input condition is exemplified, where the following fuzzy membership degrees for three inputs are taken: $\mu_1 = 0.2$, $\mu_2 = 0.5$ and $\mu_3 = 0.8$. In this case, from the figure we note the three different activation degrees are $g_1 = 0.55$, $g_2 = 0.79$ and $g_3 = 0.96$ respectively, so that $O_j = g_1 g_2 g_3 = 0.42$. That is the node output of the solution is below 0.5. In a linear computation, had we for instance taken the average among the three input values, the result would have been exactly 0.5, this result is 20% lower compared to the output from the neuron. This means the AND operation 'punishes' more severely for the low satisfaction of input 1, compared to computing the average of the inputs. The fuzzy neural tree established in this way is a model of a knowledge reflecting the inherent consistency of the latter.

3.5 Fuzzy neural tree model for the neighbourhood design

From Fig. 16 the two objectives of the neighbourhood design are seen, namely maximizing the recuperative quality and living comfort performance, located at one level below from the root node. The demand for recuperative quality of the neighborhood becomes satisfied, when all houses have large gardens oriented in the desirable cardinal direction. The demand for living comfort of the neighborhood is satisfied when perception and garden requirements for every house are

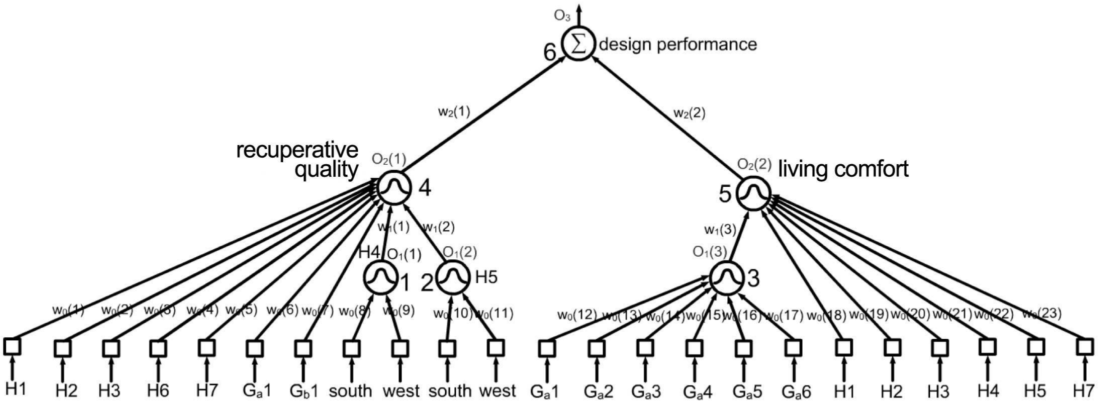
O_3
6
design performance
$w_2(1)$
$w_2(2)$
recuperative quality
$O_2(1)$
4
$O_2(2)$
living comfort
5
$w_1(1)$
$w_1(2)$
$w_1(3)$
H4
$O_1(1)$
$O_1(2)$
$O_1(3)$
1
2
H5
3
$w_0(1)$ $w_0(2)$ $w_0(3)$ $w_0(4)$ $w_0(5)$ $w_0(6)$ $w_0(7)$ $w_0(8)$ $w_0(9)$ $w_0(10)$ $w_0(11)$
$w_0(12)$ $w_0(13)$ $w_0(14)$ $w_0(15)$ $w_0(16)$ $w_0(17)$ $w_0(18)$ $w_0(19)$ $w_0(20)$ $w_0(21)$ $w_0(22)$ $w_0(23)$
H1 H2 H3 H6 H7 G_a1 G_b1 south west south west
G_a1 G_a2 G_a3 G_a4 G_a5 G_a6 H1 H2 H3 H4 H5 H7

Figure 16: Structure of the neural model for measuring the performance of members during the evolutionary search.

also highly satisfied. The recuperative quality is determined by the performances of the individual gardens, such as garden of $H1$, $H2$, $H3$, etc. This is except with respect to the garden performance of houses $H4$ and $H5$, where the recuperative performance has two additional sub-aspects. These aspects are the performance of the garden to the west and the south side of the house respectively. In the same way the living comfort performance is determined by the privacy of the individual houses' south facades. An exception is the privacy performance of the houses G_a1-G_a6, which together form an additional sub-aspect of the privacy performance.

The connection weights in the neural are assessed by a domain expert. They are given in Table 2. It is emphasized that the weights are determined without computation. They are elemental constituents of an expert knowledge. The houses are considered to differ with respect to their importance for recuperative and comfort performance of the neighborhood, which is a common consideration in decision making in architecture, due to different expected user demands for a house. However, concerning the visual privacy of the group of houses G_a1-G_a6 it is noted that in this design every house is considered equally significant with respect to privacy of the design, so that the weights $w_0(12)$-$w_0(17)$ are equal.

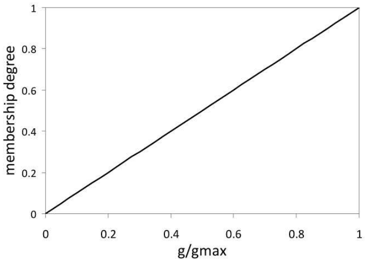

Figure 17: Fuzzy membership function characterizing the inputs that determine the garden performance of the neighborhood.

3.6 Modeling the requirements at the terminal nodes

In order to provide the neural model with input values, fuzzification processes are carried out at the terminal nodes shown by means of square shaped boxes in Fig. 16. From the recuperative quality viewpoint generally the south gardens should be as large as possible, i.e. the percentage g/g_{max} should be close to unity. This requirement is expressed by the fuzzy membership function given in Fig. 17, where the membership degree is proportional to the value g/g_{max}.

The requirements for living comfort entails that the visual privacy P_{priv} experienced for every house

Table 2: Weights of the neural tree for the design performance

Weight	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$	$n=9$	$n=10$	$n=11$	$n=12$
$w_1(n)$	0.10	0.10	0.22	-	-	-	-	-	-	-	-	-
$w_0(n)$	0.10	0.10	0.10	0.10	0.10	0.15	0.15	0.70	0.30	0.70	0.30	0.17

Weight	$n=13$	$n=14$	$n=15$	$n=16$	$n=17$	$n=18$	$n=19$	$n=20$	$n=21$	$n=22$	$n=23$
$w_1(n)$	-	-	-	-	-	-	-	-	-	-	-
$w_0(n)$	0.17	0.17	0.17	0.17	0.17	0.13	0.13	0.13	0.13	0.13	0.13

should be as high as possible, and satisfaction becomes less as the privacy approaches to zero. This requirement is expressed by means of the following Gaussian shaped fuzzy membership functions at the terminal nodes, in order to determine the degree of privacy performance of a house.

$$\mu(P_h) = \begin{cases} e^{-\frac{1}{2}\left(\frac{P_h - P_{sh}}{\sigma_h}\right)^2} & \text{if } P_h \leq P_{sh}; \\ 1.0 & \text{otherwise} \end{cases} \tag{23}$$

where P_h is the degree of privacy of house h; P_{sh} denotes the privacy degree from on which the privacy requirement is considered to be fully satisfied; σ_h denotes the uncertainty associated with the demand for the satisfaction of the privacy demand P_{sh}. The membership functions for the different houses are selected by a domain expert and are shown in Fig. 18-21.

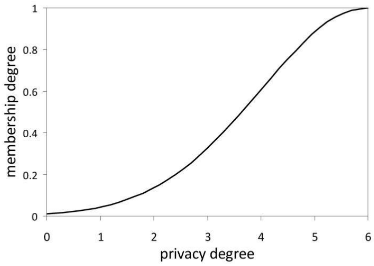

Figure 18: Membership function for privacy performance of $H1$, $H2$ and $H4$, where $P_{sh} = 6.0$, $\sigma_h = 2.0$ for $P \leq 6.0$.

It is noted that the membership functions selected for the privacy performance evaluation all have the same basic shape. However, the output maxima are located at different locations. This is to account for different requirements that are due to the different housing types and lot conditions involved. For example the membership function for the Houses $H1$, $H2$, and $H4$ expresses that visual privacy is required to be value 6.0 or greater as an ideal situation. Explicitly when privacy $P \geq 6.0$ the leaf node output is unity, representing maximum satisfaction of the corresponding requirement. In case $P < 6.0$ the output of the respective leaf node becomes less than unity, as specified by the membership function. For the other houses more privacy is deemed desirable as can be seen from the location of the respective maxima P_{sh}.

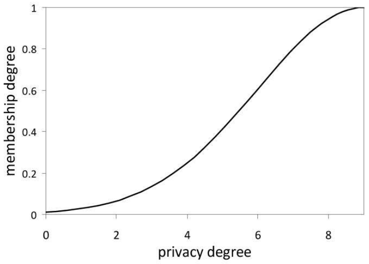

Figure 19: Membership function for privacy performance of $H3$, where $P_{sh} = 9.0$, $\sigma_h = 3.0$ for $P \leq 9.0$.

4 Results

4.1 Obtaining Pareto-optimal solutions

Before the neural tree can be used for fitness evaluation during the evolutionary search, the consistency condition has to be imposed on the neural tree. This can be carried out by classical or evolutionary learning. In the present implementation classical learning is

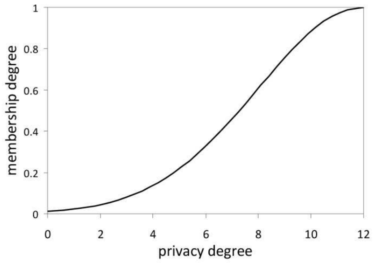

Figure 20: Membership function for privacy performance of $H5$ and $H7$, where $P_{sh} = 12.0$, $\sigma_h = 4.0$ for $P \leq 12.0$.

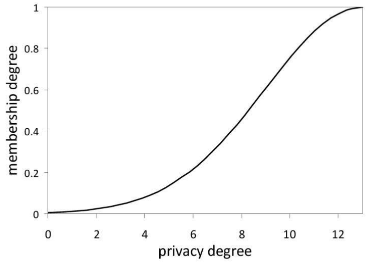

Figure 21: Membership function for privacy performance of G_a1-G_a6, where $P_{sh} = 13.0$, $\sigma_h = 4.0$ for $P \leq 13.0$.

used for high accuracy. As result of the learning process, the width of every Gaussian at the non-terminal nodes is established as seen from Table 3.

The approximation error for the consistency training data set is relatively higher for the area around $\mu_1 j = \mu_2 j = ... = \mu_i j = 0.3$ and $\mu_1 j = \mu_2 j = ... = \mu_i j = 0.8$. This is illustrated in Fig. 14, where it is seen that the deviation of the Gaussian from the line $O_j = a\mu + b$ is maximal around these points. To exemplify this approximation let us consider imposing the consistency condition on node 5 which has an optimal $\sigma_5 = 0.164$. The input dataset for imposing consistency is shown in Fig. 22, as well as the approximation for node 5 in the model. The horizontal axis in the figure shows the input values for the seven nodes coming to node 5, and the vertical axis denotes the output $O_2(2)$of node 5. It is to be noted that for the seven inputs, all of them have the same value, i.e. for a value x on the horizontal axis $\mu_1 j = \mu_2 j = ... = \mu_i j = x$. From the figure it is seen that due to the Gaussian shape used as basis function at the inner node the approximation error is maximal at two locations that are approximately at the inflexion points of the resulting Gaussian.

Table 3: Resulting widths of the Gaussians at the non-terminal nodes.

Node no.	1	2	3	4	5
σ	0.0322	0.0322	0.173	0.144	0.164

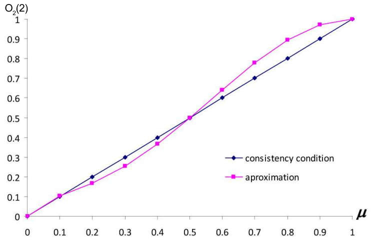

Figure 22: Verification of the consistency condition applied to node 5.

4.2 Design principles through automated innovization

Having established the fuzzy neural tree, the design task in this implementation is to maximize the outputs of the nodes 4 and 5 of the neural tree as the two objectives. For this purpose an elitist multi-objective evolutionary algorithm (MOEA), NSGA-II is employed. The variable boundaries for the locations of the houses are given in Table 4.

The boundaries are given by the minimal and maximal x and y coordinates for the positions of the houses $H1$-$H7$, G_a1-G_a6 and G_b1-G_b4. These boundaries are selected, so that the facades of the buildings are at a distance greater than $3m$ from the boundary of the lot. This is according to legal safety regulations applying to this design case. The boundaries of the placement are taken parallel to the x and y-axis. The y-axis is in north direction, and the x-axis is in east direction.

Table 4: Variable bounds for the problem.

House	$H1$		$H2$		$H3$		$H4$		$H5$	
	x	y	x	y	x	y	x	y	x	y
Min.	25.0	26.0	26.0	46.0	56.0	47.0	81.0	34.0	86.0	52.0
Max.	31.0	34.0	36.0	56.0	69.0	56.0	117.0	38.0	114.0	57.0

House	$H6$		$H7$		G_a		G_b	
	x	y	x	y	x	y	x	y
Min.	21.0	6.0	3.0	70.0	27.0	67.0	76.0	7.0
Max.	28.0	16.0	10.0	80.0	32.0	81.0	79.0	22.0

As the problem is treated as bi-objective problem, the evolutionary algorithm is not converging to a single solution, but it yields a Pareto-optimal front in the two-dimensional objective space. This is relevant, so that decision making remains flexible, and a final decision is taken with great awareness. It is interesting to note that generally the problem could have been treated as a single-objective optimization problem, under the condition that the weights $w_2(1)$ and $w_2(2)$ from figure Fig. 16 are fixed. However, as the living comfort and the recuperative quality are rather abstract concepts, such a commitment is rather problematic to make for a decision maker, so that the multi-objective approach is more appealing.

The Pareto-optimal front obtained after 50 generations is seen from Fig. 23. Automated innovization is carried out on these solutions by solving Eq. 5. S_{reqd} is chosen to be 90% and $\mathcal{N}$ is taken as 4 to avoid complex relationships. It is noted that only one solution among all the Pareto-optimal solutions remains unclustered in the process, and this solution is marked by '$\times$' sign in the figure. A total of 303 significant relationships are found. Here we discuss and interpret the following six:

$$x_4^{1.0000} y_4^{0.9801} = constant \quad (24)$$
$$x_4^{0.9001} y_5^{1.0000} = constant \quad (25)$$
$$x_3^{0.3571} y_3^{0.7110} x_4^{1.0000} = constant \quad (26)$$
$$y_2^{1.0000} x_3^{0.3468} x_4^{0.8250} = constant \quad (27)$$
$$y_1^{0.7647} x_2^{-0.1746} x_4^{1.0000} y_4^{0.7479} = constant \quad (28)$$
$$y_2^{-0.6557} y_3^{0.3970} x_4^{0.6304} x_8^{1.0000} = constant \quad (29)$$

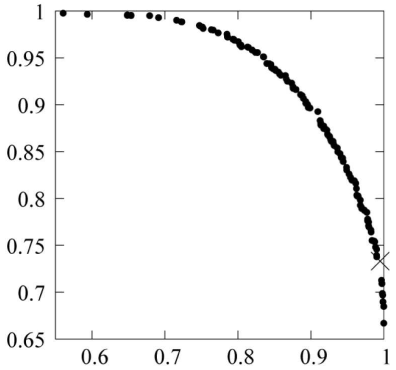

Figure 23: Pareto-optimal solutions obtained for the problem. The horizontal axis represents the recuperative quality; vertical axis represents the living comfort

4.3 Interpretation of design principles

In order to illustrate interpretation of the principles obtained by means of automated innovization, a number of them are interpreted by a domain expert as follows. The first principle given by Eq. (24) involves two basis functions. This principle implies that as house $H4$ is moved east, its west garden size increases and the privacy of $H5$ is reduced. This means the recuperative quality of the neighbourhood increases somewhat, at the cost of some reduction in living comfort. In order to still have a solution that satisfies the Pareto optimality condition, in case no other variable is modified, $H4$ should be moved south to increase the privacy of $H5$. This is seen in Fig. 24.

The second principle involving two basis functions is given by Eq. (25). This rule implies that, as house

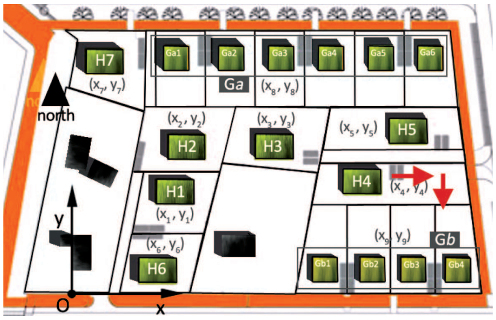

Figure 24: Illustration of implications of the principle in Eq. (24).

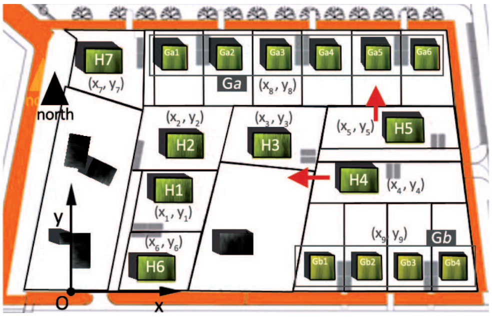

Figure 25: Illustration of implications of the principle in Eq. (25).

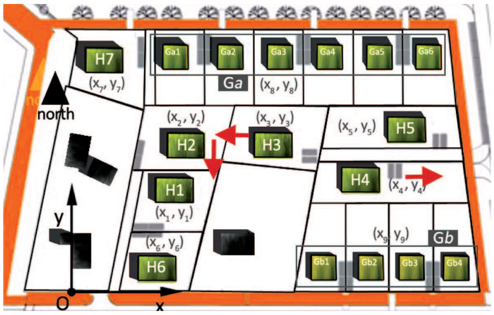

Figure 26: Illustration of implications of the principle in Eq. (26).

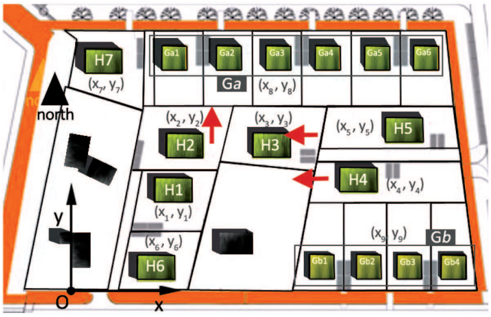

Figure 27: Illustration of implications of the principle in Eq. (27).

$H5$ is moved north its south garden size is increasing, yet the house is moving closer to the group G_a. This means the recuperative quality of the neighborhood is slightly increasing at the cost of reduced living comfort. To remain Pareto-optimal, $H4$ should move west, reducing the garden performance of $H4$ somewhat and further increasing the privacy of $H5$. This is seen in Fig. 25.

A third principle involving, which involves three basis functions is given by Eq. (26). Interpretation of the principle is that as house $H4$ is moved east its garden size increases while privacy of $H5$ decreases. This means the recuperative quality of the neighborhood increases somewhat at the cost of some reduction in living comfort. To reach a solution that is Pareto-optimal, $H3$ should move south, trading in some recuperative quality to increase the living comfort of the neighbourhood. Namely, as $H3$ moves south its perception of the houses G_a becomes desirably reduced. This is seen in Fig. 26.

A fourth principle, which also involves three basis functions, is given by Eq. (27). As house $H2$ is moved north, its garden size increases, thereby increasing the recuperative quality of the neighbourhood to some extent, while compromising the living comfort, as it moves closer to G_a. To obtain a solution that satisfies the Pareto optimality criterion, $H4$ should move west, increasing the living comfort, while losing some of the previously gained recuperative quality $H5$. This is seen in Fig. 27.

A fifth principle, which contains four basis func-

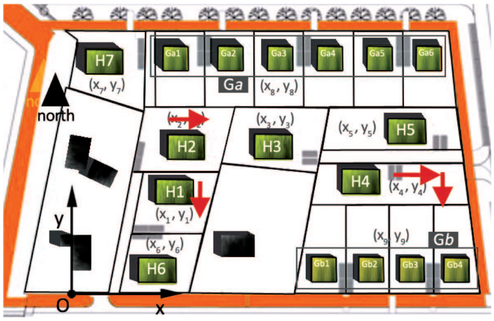

Figure 28: Illustration of implications of the principle in Eq. (28).

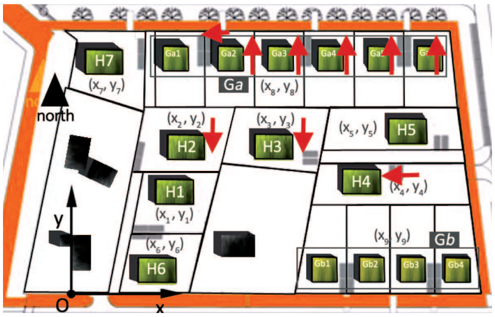

Figure 29: Illustration of implications of the principle in Eq. 29.

tions, is given by Eq. (28). The principle is interpreted as follows. As house $H4$ is moved east, its garden performance increases, i.e. the recuperative quality of the neighborhood slightly increases, while the living comfort slightly diminishes, as the privacy of $H5$ decreases. To still remain Pareto-optimal, the loss of living comfort is compensated by moving $H4$ and $H1$ in south direction, while moving $H2$ east. This way the privacy of $H5$ and $H2$ is increased, compensating the loss in living comfort. This is seen in figure Fig. 28.

A sixth principle that also consists of four basis functions is given by Eq. (29) This principle entails that, as houses G_a are moved north, their privacy increases as well as their garden performance. This means that the living comfort of the neighborhood and also the recuperative quality is increased somewhat. For Pareto optimality, the gain in recuperative performance is reduced and living comfort is increased by moving $H2$ and $H3$ south and moving $H4$ west, as seen from Fig. 29.

From these examples it is seen that the interpretation of the principles is rather intuitive. This is also due to the fact that in the present case fuzzy modeling is used to treat the objectives in the decision making. As the two objectives, maximize recuperative quality and living quality are linguistic terms that are well understood by a decision maker, it is easy to understand the relations among a number of decision variables while keeping in mind the Pareto optimality condition. However, it is noted that the interpretation of the principles for rules with more than three basis functions becomes less intuitive, as relationships among seemingly remotely related objects appear in the principles. However, the latter implies that a decision maker is able to gain insight into subtle influences present in the decision making problem at hand.

It is further noted that the innovized principles can be used to fill gaps on the Pareto-optimal front. In order to identify a Pareto-optimal solution on the front that has not been found by the evolutionary algorithm, merely one of the principles has to be used, as each of them applies to ensure Pareto optimality. This means, through the innovized principles the set of optimal solutions evolved by the evolutionary algorithm can be extended to a larger set.

5 Conclusions

Automated innovization is used to extract design principles in a soft multi-objective decision making problem. Despite the rather simple looking Pareto-optimal front of the bi-objective problem being smooth and convex, rich diverse design knowledge in the form of invariant principles among the decision variables are obtained from the corresponding information. This exemplifies the suitability of the clustering based automated innovization method for complex real-world problems that involve vagueness and imprecision, and a high degree of non-linearity in the fitness evaluation. The empirical design rules obtained from this process are interpreted by a domain expert, and the interpretation is intuitive. For rules involving a larger number of basis functions, interpretation becomes less intuitive, so that the decision maker can understand subtle relations among the decision variables. The extracted design principles can help decision makers to reach an informed decision as they understand with precision the implications of the trade-offs under consideration. Another interesting benefit from the innovization is that gaps in the non-dominated front can be filled using the obtained design principles. Some relevant future research directions are lower level automated innovization, higher level automated innovization, studying the performance of clustering with higher number of objectives and identifying design principles in problems with simulation based objective evaluation.

References

Bandaru, S. and Deb, K. (2010). Automated discovery of vital knowledge from Pareto-optimal solutions: First results from engineering design. In *2010 IEEE world congress on computational intelligence*, pages 1224–1231. IEEE Press.

Bandaru, S. and Deb, K. (2011a). Automated innovization for simultaneous discovery of multiple rules in bi-objective problems. In *Evolutionary Muticriterion Optimization EMO 2011*. Springer.

Bandaru, S. and Deb, K. (2011b). Towards automating the discovery of certain innovative design principles through a clustering based optimization technique. *Engineering optimization*.

Bittermann, M. and Ciftcioglu, O. (2008). Visual perception model for architectural design. *Journal of Design Research*, 7(1):35–60.

Bittermann, M., Sariyildiz, I., and Ciftcioglu, Ö. (2007). Visual perception in design and robotics. *Integrated Computer-Aided Engineering*, 14(1):73–91.

Broomhead, D. S. and Lowe, D. (1988). Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems 2*, pages 321–355.

Ciftcioglu, O., Bittermann, M., and Sariyildiz, I. (2006). Towards computer-based perception by modeling visual perception: A probabilistic theory. In *IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC'06*, volume 6, pages 5152–5159. IEEE.

Ciftcioglu, O., Sariyildiz, I., and Bittermann, M. (2007). Building performance analysis supported by ga. In *IEEE Congress on Evolutionary Computation, 2007. CEC 2007.*, pages 859–866. IEEE.

Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2–4):311–338.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. New York: Wiley.

Deb, K. (2003). Unveiling innovative design principles by means of multiple conflicting objectives. *Engineering Optimization*, 35(5):445–470.

Deb, K., Agarwal, S., Pratap, A., and Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb, K. and Sindhya, K. (2008). Deciphering innovative principles for optimal electric brushless D.C. permanent magnet motor design. In *2008 IEEE world congress on computational intelligence*, pages 2283–2290. IEEE Press.

Deb, K. and Srinivasan, A. (2006). Innovization: Innovating design principles through optimization. In *GECCO '06 - Proceedings of the 8th annual conference on genetic and evolutionary computation*, pages 1629–1636. New York: ACM.

Hunt, K., Haas, R., and Murray-Smith, R. (1996). Extending the functional equivalence of radial basis function networks and fuzzy inference systems. *Neural Networks, IEEE Transactions on*, 7(3):776–781.

Zadeh, L. (1997). Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy sets and systems*, 90(2):111–127.