Multi-objective design for space layout topology

Ö. Ciftcioglu, S. Durmisevic and S. Sariyildiz

Department of Computer Science
Delft University of Technology, Faculty of Architecture
Berlageweg 1, 2628 CR Delft, The Netherlands

ABSTRACT

A novel method to produce space layout topologies for architectural design is described. From the uniformly distributed design solutions in the solution space the corresponding design requirements are computed according to a given norm and metric function. The system is based on graph representation of the layout so that the desired relations between the pairs of nodes are considered to be independent variables of appropriate series of multivariable functions mapping the requirements into the solution space. The system so established is used as a knowledge-base for robust layout design where knowledge base having been established, the layout design requirements are introduced to the system as design constraints and the output is identified in the multidimensional solution space by means of interpolation. Since the smoothness of the interpolation is guaranteed, robust design layout, in the form of node locations, is obtained.

1 INTRODUCTION

One of the important architectural design problems is the related space layout and the problem is addressed by many researchers (Steadman1983, Zandi 1992, Damski 1997, Chitchian 1997) among others. These are a variety of approaches dealing with the problem. These extend from dimensionless form of rectangular units on a plan to area, width and length of each space and the optimal dimensions according to some criteria. However in most cases, the criteria used are not satisfactory enough for a design since the optimization (unconstrained or constraint) is based on a scalar quantity, generally referred to as "cost function" and architectural designs concerns in most cases multi-objective considerations. In this case, such designs are mostly considered to be design decision support systems, therefore. In essence, the problem stems from its ill-conditioned nature. That is, architectural problems are rather soft relative to engineering problems and therefore engineering approach to such soft problems may not be conclusive. Particularly, in the second half of the last decade, evolutionary algorithms are developed to tackle the ill-posed problem, like architectural design problems. Several researchers used Genetic Algorithms approach and reported their results in literature (Damski 1997, Jagielski

1997). Genetic algorithms (GAs) have been applied almost exclusively to single-attribute problems. Typically, an ad-hoc function of multiple attributes is found to yield a scalar fitness function. However, architectural design problems are multi-attribute. The extension of GAs for multi-attribute case is the Pareto algorithm (Horn 1994) where the GA is used to find all possible tradeoffs among the multiple, conflicting objectives. In attribute space, the set of non-dominated solutions lies on a surface known as Pareto Optimal Frontier. The goal is to find and maintain a representative sampling of solutions on the Pareto front. However, still it is not clear what is the best way to handle multiple objectives with genetic algorithms and its utilization for architectural design needs further refinements in the method for robust solutions.

In addition to the multi-objective optimization problem where even the objectives might be conflicting among themselves, there is another important issue worth to mention. In architectural design, the actual locations of the nodes represented in the corresponding graph are most relevant to the layout rather than the boundaries determined by the method being used. In other words, for an architect it is most desirable to identify the central locations of the units on the layout, so that he can further elaborate on the architectural design determining the unit boundaries in most convenient way directed by the requirements of the actual utilization of the space. Such flexibility provides the architect with an additional dimension in his professional domain. The optimization algorithms, so far endeavor to establish the unit boundaries in the solution space so that from the design viewpoint, the case can be seen as a 'computer enhanced design', rather than an 'architectural design'.

The present approach intends to introduce robust multi-objective design solutions for architectural design problems providing architect with the additional architectural dimension described above. For this purpose, briefly, from a set of uniformly distributed design solutions in the multidimensional solution space the corresponding design requirements are computed according to given norms and metric functions. The system is based on graph representation of the layout so that the desired relations (attributes) on the graph are considered to be independent variables of a series of appropriate multivariable functions mapping the requirements into the solution space by in advance computation or consideration. The system so established is used as a knowledge base for layout design where the desired layout design requirements are introduced to the knowledge base as design and constraints at the input. The output is identified in the multidimensional solution space by means of interpolation accomplished by the knowledge base. Since the smoothness of the interpolation is guaranteed, robust design layout is obtained. The organization of the paper is as follows. Section 2 describes multivariable functional interpolation, Section 3 describes the knowledge base formation and Section 4 gives the test results which are followed by conclusions. Since the interpretation of multidimensional functional interpolation for the knowledge-base outcomes is essential, it is briefly described below.

2 MULTIVARIABLE FUNCTIONAL INTERPOLATION

We consider a set of N data vectors $\{x_i ,\ i=1,...,N\}$ dimension of p in R^p and N real numbers $\{d_i,\ i=1,2,...,N\}$. We seek a function $f(x)$: $R^p \rightarrow R^1$ that satisfies the interpolation conditions $f(x_i)=d_i$, i=1,2,...N. There are several methods as solutions for this interpolation problem, like Lagrange interpolation functions. Here we consider radial base functions (RBF) due to their suitability for use in the present research. The characteristic feature of radial functions considered here is that their response decreases monotonically with distance from a central point. The RBF approach constructs a linear space using a set of radial basis functions $\phi(||x-c_j||)$ defined with a norm which is generally Euclidean. The center described with a vector c_j, a distance scale and the shape of the radial function are parameters of the model. By means of these base functions, we can model the function as

$$f(x) = \sum_{j=1}^{N} w_j \phi(\| x - c_j \|) \qquad c_j,\ x \in R^p \ ,$$

where w_j are weights or coefficients. The interpolation conditions $f(x_k) = d_k$, k=1,2,...N can be generalized as

$$f_k(x) = \sum_{j=1}^{N} w_{jk} \phi(\| x_k - c_j \|) \quad x \in R^p,\ \ k = 1,...,s$$

where the mapping from input to output is $R^p \rightarrow R^s$ and $f_k(x_i) = d_{ik}$, i=1,2,......,p ; k=1,2,.....,s. Once the appropriate basis functions (ϕ) and the distance measure are selected the interpolation function can be established. Among several radial-basis functions, the Gaussian basis function is of particular interest and used in this research:

$$\phi(r) = \exp(-\frac{r^2}{2\sigma^2})$$

While radial basis functions have been used for many years for multivariable interpolation with firm mathematical base, it was only recently that they are introduced for use with neural (Broomhead 1988). Therefore, such a structure is named as radial basis functions network. The use with neural network architecture has a number of advantages over the other type of feed-forward neural network architectures with respect to training and locality of approximation. From the architectural design viewpoint however, more interestingly, they are closely related to and under mild conditions equivalent to fuzzy logic so that radial basis functions network acts as a knowledge base providing inductive outcomes with possible interpretations in terms of fuzzy logic.

3 KNOWLEDGE BASE FORMATION

The structure of the RBF network for is given in figure 1 where only one output node is indicated for simplicity and more nodes can be added in the same way. For a set of given design information the RBF network is "trained" which means, the information given is stored in the network where the functional relationships between the input and output data are structured. Although this can be considered as a common case for feed-forward networks, since RBF network is closely connected to multi-valued fuzzy logic operations and the outcomes are uniquely inductive based on knowledge it bears. In this form the radial base functions (here, they are family of gaussian functions) act as membership functions of fuzzy logic. Therefore the neural RBF network serves as an effective support to designer for decision-making. Exclusively, this fuzzy logic interpretation of the RBF network makes it interesting for architectural implementations when dealing with the imprecise/fuzzy architectural data like adjacency, privacy, closeness, similarity etc. Referring to this interpretation of RBF networks, then, also the training method is of importance. Therefore, the network weights are determined effectively by the method of orthogonal least squares (OLS) (Chen 1991). Due to OLS method of training, the underlying knowledge elicited from the information introduced to the network is structured in most desirable way. Namely, the input information is subjected to clustering while the cluster centers are selected from the input patterns according to their degree of suitability to represent the common features of the total input information. This is in contrast with the conventional neural network training methods where the problem is taken purely as a mathematical one in the form of a cost function minimization subject to advanced mathematical methods for solution.

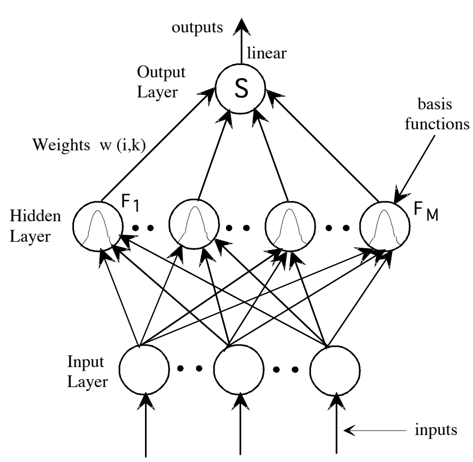

Figure 1: **Basic architecture of a RBF network for multivariable interpolation**

4 SPACE LAYOUT TOPOLOGY THROUGH KNOWLEDGE BASE

To demonstrate the potential of the method for design of space layout topology a knowledge base of adjacencies between the nodes in the layout topology is considered. In general, the adjacencies can be assigned through elaboration of the following aspects:

1. Connectivity pattern defined by type of the nodes
   Depending on the functionality of the spaces (e.g., offices), different adjacency values can be attached to the spaces that are directly connected with each other and to those that are indirectly connected for example via a corridor. This can be shown through adequate adjacency value
2. Optimal distance in relation with the requirements
   This is connected to various factors like functional and structural factors as examples. For example, for a certain level of sound attenuation, less soundproof walls are used, as the distance between two spaces (represented as nodes) can be larger, and vice-versa.
3. Cost function derived from the above-mentioned aspects, etc.

By combining all of the above-mentioned aspects, a final model can be accomplished, which would represent the optimization of all aspects considered together as a whole. The graph representation, would give to the architect enough space for the final design, since in a way, the results provided would serve him as the guidelines for design approach, indicating spatial layout, but still not determining the final shape of the units. The architect can see immediately, what the consequences are for the design by appointing higher importance to certain aspects. He/she can also choose a simple design decision-making which would mean that all aspects are of the same importance. The soft computing approach in fact gives to Architect additional freedom of design flexibility in contrast with the traditional design optimization outcomes. He or she also sees directly the consequences of each design solution with respect to marginal client's needs next to the design requirements. This is especially of importance, since each design and circumstances surrounding it, would require considerations of certain design factors at the cost of some others. By the traditional methods dealing with such different design objectives is not an easy task. For example privacy and spatial closeness between units are two different design qualities which can easily be considered separately by soft computing and the design outcomes can be easily be integrated with final architectural judgement. This is in contrast with the traditional design optimization methods which lead often inconvenient solid design solutions because of a single objective function to be optimized and also difficulties due to dealing with such vague concepts. In the latter case, therefore the results are hardly of practical use if they are not inconclusive at all. Below, two examples are provided, showing the possible layout design as multi-objective design for each case.

For implementation of the method for layout topology determination a simple architectural floor plan design with four vertices is considered as an example for concrete description which is given below. The knowledge base is prepared basically by means of computation from a simulation model, simulating adjacency, similarity, closeness, privacy etc. whatsoever relative to each pair of locations represented as nodes. These relationships among the nodes are considered to be the characteristic values describing the relative position of each pair of nodes, e.g., distance functions in the space layout. A simple computation is employed for the simulated space layout for the sake of easy presentation and verification of various aspects of the design performance. The computation may also be the outcome of elaborate considerations at any depth eventually contributing to the integrity of the knowledge base, the effectiveness of the method being unaffected.

The graph representation of this layout is shown in figure 2. The adjacency matrix for this graph is given by:

where $A = [a_{ij}]$ is a 4 □ 4 matrix and each row (and each column) of A corresponds to a distinct vertex of V. Then, $a_{ij}=1$ if vertex v_I is adjacent to vertex v_j and $a_{ij}=0$ otherwise. Note that $a_{ii}=0$ for each i=1,2,.,4. The adjacency matrix is a symmetric (0,1)-matrix, with zeros down the main diagonal. The adjacency matrix contains all the structural information about the plan. Considering the graph-theoretic definition of adjacency, the graded architectural relationships between the nodes are referred to as ‘proximity’, in place of adjacency.

Figure 2: **Graph representation of a design layout. Node #1→ circle; node #2→;star; node #3:triangle; node #4:→ diamond**

The designer states the design attributes. These attributes are normally introduced to the RBF network during the “training” process to form the knowledge base. In space layout topology, the design is connected to the norms defined between the units in that topology. The distances between any pair of units can be computed from a metric function with a selected norm. Let the distance between any two point x and y be denoted by d(p,q). This is the minimum length of an p-q path in the graph. The following properties hold for the distance function d:

- $d(p,q) \geq 0$ and $d(p,q)=0$ if, and only if $p=q$
- $d(p,q)= d(q,p)$
- $d(p,q)+d(q,z) \geq d(p,z)$

These three properties define what is normally called a metric function on the vertex set of a graph. In general, if the nodes in the graph represent units of interest, the difference between the units can be expressed by the distance function. If we define a proximity function representing the relative status of p and q, as prox(p,q), we write

$$0 \leq \text{prox}(p,q) \leq 1 \quad \text{and} \quad \text{prox}(q,q) = 1$$

that is, the proximity would diminish with the increasing distance. The way of diminishing is dependent on the design problem at hand and might take various forms, like

$$\text{prox}(p,q)= 1/[1+d(p,q)] \quad \text{or} \quad \text{prox}(p,q) = e^{-d(p,q)}$$

as examples. In the space layout problem, the nodes (vertices in the graph) represent the locations and the proximity measure can be used to represent any desired (complex) relationship between the nodes and can be computed by means of a chosen distance function using Euclidean metric for the differences between the locations. After the locations having been properly determined in accordance with the design requirements the architect determines final space layout as appropriate partitions. For instance, the relative difference can be calculated as the attenuation of noise level between any two locations and it is dependent on the Euclidean distance between the same locations used in the relevant distance function. In general, theoretically one can think of that, provided any distance measure and any distance function relevant to the design problem are available, then an appropriate knowledge base could be prepared as accumulation of a series of design cases. Afterwards, the inputs of such a knowledge base is used for the introduction of any related design requirements with any complexity. The solution through the knowledge base outputs is guaranteed to be compatible with the knowledge base prepared. Note that, since the solution is in the form of multivariable functional interpolation, it is an approximation degree of which is dependent on the comprehensiveness of the knowledge base. The same thing in fuzzy logic terms can be stated as follows. Since the solution is outcome of fuzzy logic, the solution is also fuzzy approximation where the degree of fuzziness dependent on the number of fuzzy partitions in the universe of discourse subject to fuzzy computation.

To demonstrate the potential of neuro-fuzzy knowledge base for the space layout problem, a knowledge base of a simulated case is prepared for test. For the simplicity, the design attributes are related to Euclidean distances between each pair of nodes, as mentioned before. The attributes are taken to be proximity values (i.e., adjacency, similarity, closeness, privacy etc. whatsoever relative to each other) that are defined by the help of multivariable radial gaussian functions for the reason of computational convenience and verification of various aspects of the design performance. The proximity value □ for a particular Euclidean distance r between any two locations p and q is computed by the distance function d(p,q)

$$\square = d[r(p,q)] = \exp[-0.5\ (r/\square)^2]$$

where □ is the width parameter of the gaussian. With this definition all proximity values between the pair of nodes are calculated. Since the proximity matrix is symmetrical, for the calculations only the upper triangle excluding the main diagonal is considered. In this particular example, with four vertices only six pairs are to consider.

For defining the locations of four nodes, one needs eight coordinates (x and y values for each node). For the formation of the knowledge base we consider the case as an inverse problem. Therefore eight arrays (each has a length of 250) of uniform random numbers in between 0 and 1 are generated so that a random output matrix dimension of [250 x 8] is obtained. For each set of eight coordinates six proximity attribute values are computed for □ = 0.6 so that for total 250 sets, an input matrix with dimension of [250 x 6] is computed. These two matrices are used as design specification information and a RBF network is formed as a knowledge base from the set of these 250 input-output pairs. In this case, the RBF input is provided with the data from six dimensional input space and the output is provided with the data of coordinates (total eight) from two dimensional layout solution space.

The solutions are uniformly distributed in a unit square layout area at the output with related design specifications (proximity values in this example) at the input. Note that, input is not uniform in this case although this would be desirable as explained later. The corresponding knowledge-base performance is represented in figure 3 for the sets #1 and #2. The knowledge-base performance for total 250 sets is presented in figure 4. Note that for each set there are 6 inputs, 8 outputs, i.e., 4 x-y coordinates corresponding to four nodes and for each case the data and the estimate coincide indicating that the knowledge base exactly represents the relationships computed by means of the given distance function $d[r(p,q)] = \exp[-0.5\ (r/\square)^2]$.

The knowledge-base having been determined two cases devised arbitrarily for testing the performance of the knowledge base are presented below as typical examples of the performance

Figure 3: **Knowledge-base outputs for two sets of computed input data (upper set and lower set). Note that for each set there are 6 inputs, 8 outputs, i.e., 4 x-y coordinates corresponding to four nodes and for each case the data and the estimate coincide**

Figure 4: **Knowledge base outputs for the total 250 sets. Note that for each set we have 8 outputs, i.e., 4 x-y coordinates corresponding to four nodes in the design layout. Note that design and estimate virtually coincide**

of the approach. For the first case a set of desired proximity design layout, i.e., the proximity assignments, namely □ =d[r(x,y)] values, are in the form of proximity matrix P_r is given by that corresponds to the general form

With this design data in six-dimensional input space the estimated proximity matrix obtained as outcome of the design solution from the knowledge base is

The graph representation of the design layout shown in figure 5 and the outcome is represented in figure 6 and figure 7.

Figure 5: **Graph representation of a design layout for the case studies**

Figure 6: **Design solution estimated by the RBF network knowledge-base corresponding to the proximity values given by the proximity matrix (case 1)**

Figure 7: **Design solution estimated by the RBF network knowledge-base corresponding to the proximity values given by the proximity matrix (case 1)**

Figure 8 provides a possible layout interpretation of the required adjacency values for the 4 units. The points obtained in the figure above were rotated by 25 degrees to provide easier model manipulation. The 4 points are the representatives of each space in the following manner:

Symbols: Connections:

1 _ _
2 _ *
3 _ □
4 _ ◊

The following assumption is made: Unit1 and unit4 needed to be connected, but they also required a certain sound isolation from each other, which in this case is done by placing a corridor in between. In such case the adjacency requirements are met (value 0.2058). If the sound isolation were not required, the whole layout would be more compact. Also a corridor may have been required between units 1 and 3 since the adjacency value for these two units is 0.3324. This example shows that the architect can still determine the shape and the size of the spaces and influence the layout, while he has in mind the optimal position of these spaces in relation to each other. This can help him focus more on the design and final layout rather than having constantly in mind the various requirements and relations between units, which is in many cases rather complex process.

Figure 8: **One possible layout organization based on the graph from figure 7**

For the second test case the design requirements given by the proximity matrix

yielded the estimated counterpart proximity matrix P_e as
The outcome is represented in figure 9 and in Figure 10 where the latter corresponds to the graph representation of the design layout shown in figure 5.

Figure 9: **Design solution estimated by the RBF network knowledge-base corresponding to the proximity values given by the proximity matrix (case 2)**

Figure 10. **Design solution estimated by the RBF network knowledge-base corresponding to the proximity values given by the proximity matrix (case 2)**

The inspection of the estimated proximity matrices indicates that the estimation errors of the proximity values are relatively smaller for the higher proximity values and vice versa. This is simply because of the training procedure and consequently knowledge base formation and therefore this is what one should expect. Without dealing with the mathematical details, this is briefly explained below. For the distance function used in this study, the uniform multivariate density function between 0 and 1 in the solution space results in a probability density function approximately in the same interval (exp(-2) and 1) but higher probabilities for higher values in the input (layout requirements) space. Therefore the knowledge base is formed with more data for higher proximity values relative to the smaller ones and consequently the estimates are more accurate. This is a basic indication in neural network training: with more information with cases in training, the knowledge base gives more accurate estimation around these cases. On the other hand, it is generally desirable that the estimation errors are not biased on one way or other. To provide this, considering the particular distance function employed and/or the available data at hand, input to the knowledge base should be appropriately prepared in advance. That is, it possesses a uniform probability density function in contrast with the present case where in place of input, output has uniform probability density function due to computational convenience for test. The graph from figure 10 served as a starting point for the layout design (figure 11). Again, this is only one of the possible interpretations of the graph. Units 1 and 2 are connected with the unit 3 indirectly through unit 4, but also their direct connection with the unit 3 is possible. In the layout proposed below the reason for not connecting them directly was to have higher privacy value for the units 1 and 2.

It is important to note that the design solutions in this example are rotation and translation invariant since the distances are measured relative to the associated nodes in accordance with a selected distance function. In space layout problem, the distance function plays an important role since it represents the required design considerations, which can be totally quantitative, as sound levels for instance, or they can be qualitative or linguistic. In the latter case, in addition to the inherent fuzzy logic interpretation of the computations in the knowledge base, the fuzzy logic techniques can be invoked for providing inputs to the distance function, the rest of the design approach presented basically remaining the same. More interestingly, for the requirements, which fall into different categories, sound attenuation and degree of neighborhood between the locations subject to space layout partitions for example, two different layouts can be obtained using appropriate knowledge base separately. Then, having in mind the rotational and translational invariance of the outcomes, they can be desirably fused somehow in most consistent way as a part of the architectural creativity.

6 CONCLUSION

Various optimization procedures as architectural problem solutions are generally based on uni-objective optimization. However such problems are generally complex and need multi-objective considerations. On the other hand, multi-objective heuristic search algorithms are not robust enough for space layout problems. Therefore in both cases design considerations turns out to be decision-makings for given design specifications.

Figure 11: **Spatial layout of four units based on graph from figure 10**

The present approach provides robust decision-makings for architectural design fulfilling the design specifications in a consistent way and effectively, based on a knowledge base with any degree of complexity. Such a knowledge base can be prepared in various ways providing Architect with ample flexibility for design enhanced by his or her professional knowledge. One important aspect of this is that the creativity is not limited in design because of ample design solutions. For demonstration of the effectiveness of the present approach and therefore for the verification of the design outcomes exactly, the database is prepared by computation as a demonstrative examples. Even for the basic examples presented in this work the space layout problem is by no means trivial and the knowledge base provided satisfactory results. Additionally, the design layout established by the present approach indicates the central locations of the architectural design units represented by nodes in the relevant graph rather than the imperative partitions as this is the case for traditional design optimization based outcomes. Therefore desirably, architect can exercise his/her professional considerations and creativity to obtain optimal layout design with appropriate partitions by final human-intelligence based decisions.

7 REFERENCES

Broomhead, D.S. and Lowe, D.(1988) Multivariable Function Interpolation and Adaptive Network. *Complex Systems* 21, p. 321-355

Chen C, F.N. Cowan and P.M. Grant. (1991) *Orthogonal Least Squares Algorithm for Radial Basis Function Networks*. IEEE Trans. on Neural Networks, Vol.2, No.2, March

Chitchian D. (1997) Artificial Intelligence for Automated Floor Plan Generation. Ph.D. Thesis, Delft University of Technology, The Netherlands

Damski Jose C. and John S. Gero (1997) An Evalutionary Approach to Generating Constrained–Based Space Layout Topologies. Proceedings of the 7th International Conference on Computer Aided Architectural Design Futures, Munich, Germany, 4-6 August 1997

Horn J. and N. Nafpliotis (1994) *Multiobjective Optimization Using the Niched Pareto Genetic Algorithm.* Proceedings of the first IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Vol.1 (ICEC'94)

Jagielski R. and J.S. Gero (1997). *A Genetic Programming Approach to the Space Layout Planning Problem.* R. Junge (Ed.). Proc. 7th International Conference on Computer Aided Architectural Design Futures, Munich, Germany, 4- 6 August

Steadman, J.P.(1983) *Architectural Morphology - An Introduction to the Geometry of Building Plans*, Pion, London

Zandi-Nia (1992) *TOPGENE:* An Artificial Intelligence Approach to a Design Process. Ph.D. Thesis, Delft University of Technology, The Netherlands