Supporting Exploration of Design Alternatives using Multivariate Analysis Algorithms

Šileryte, Rusne; D'Aquilio, Antonio; Di Stefano, D.; Yang, Ding; Turrin, Michela

Publication date
2016

Document Version
Final published version

Published in
Proceedings of the Symposium on Simulation for Architecture and Urban Design (simAUD 2016)

Citation (APA)
Šileryte, R., D'Aquilio, A., Di Stefano, D., Yang, D., & Turrin, M. (2016). Supporting Exploration of Design Alternatives using Multivariate Analysis Algorithms. In R. Attar, A. Chronis, S. Hanna, & M. Turrin (Eds.), *Proceedings of the Symposium on Simulation for Architecture and Urban Design (simAUD 2016)* (pp. 215-222). simAUD.

Supporting Exploration of Design Alternatives using Multivariate Analysis Algorithms

Rusne Sileryte[1], Antonio D'Aquilio[1], Danilo Di Stefano[2], Ding Yang[3, 1] and Michela Turrin[3, 1]

[1]Delft University of Technology
Delft, The Netherlands
{a.daquilio, r.sileryte, d.yang-2,
m.turrin}@tudelft.nl

[2]ESTECO SpA
AREA Science Park, Padriciano
99, Trieste 34012, Italy
distefano@esteco.com

[3] State Key Lab of Subtropical
Building Science, South China
University of Technology
Guangzhou, China

ABSTRACT

Parametric modelling allows quick generation of a large number of design alternatives. Ultimately, it can be combined with optimization algorithms for obtaining optimal performance-driven design. However, setup of design space for optimization is a very complex task requiring designer's a priori knowledge and experience. Therefore, this paper focuses on the process that happens before the optimization. It proposes to use multivariate analysis algorithms for exploring and understanding the relations between various design parameters, after sampling the design space. Additionally, portrayal of geometry is introduced as an extension of conventional visualization methods, which accounts for evaluation of ill-defined design criteria by using designer's expertise. The proposed method is computationally efficient and integrated into an environment familiar to architects. It relies on multivariate analysis algorithms together with database querying capabilities and an interactive dashboard developed for geometry portrayal.

Author Keywords

Design & Optimization, Design Computation, Multi-objective Optimization, Visual Analytics, Multivariate analysis, Design Exploration, Parametric Modelling

ACM Classification Keywords

H.2.8 Database Applications; J.6 Computer Aided Design

1 INTRODUCTION

The traditional design process relies on designer's knowledge, experience and intuition, which help to determine an optimal solution to a design problem. However, in recent years project complexity has increased while challenges of performance evaluation have decreased due to emergence of computational tools and methods. Therefore, design process is no more expected to find safe solutions by performing the correct calculations in later stages of a design, but rather explore a wide range of potential options, to come to informed decisions at every design stage. Fortunately, the advance in big data handling and analysis technologies has made design studies easier, providing more information to a designer.

Parametric modelling allows quick generation of a large number of design alternatives. Ultimately, it can be combined with optimization algorithms for obtaining optimal performance-driven design. This however, accounts mostly for numerical performance values, disregarding ill-defined criteria like design aesthetics or contextual appropriateness. In contrast to conventional optimization processes, interactive evolutionary optimization has been proposed by various researches. An overview of human input in the optimization process is available in [14]. The more relevant to architectural optimization ParaGen tool [15], has shown the benefits of combining parametric modelling and interactive evolutionary optimization on a wide variety of design examples. Even more control and interaction in performance-focused design space exploration is given in [11].

However, as noted in [14], automated optimization procedures fail to take advantage of designer's expertise, while in architectural design an important role should be given to the learning process of a designer, providing him with knowledge on the trade-offs between various disciplines (e.g. climate, structural design, etc.) and performance objectives. This problem has been partly tackled in [16] by utilizing statistical assessment of complex data for post-processing results. Furthermore, [8] has used k-means clustering and Archetypal Analysis to derive general knowledge linking architectural features to design performance. In [10] a phi-array visualization system is introduced to analyse sub-optimal solution.

However, the challenge of exploring design alternatives still lies in facilitating interpretation of numerous optimization results both in means of comprehensive geometry visualization and understanding separate design parameter influence on the performance values. The practicability of tools from architect's point of view has also been addressed as an important issue [16].

This paper presents a design environment that integrates guidance-based support for exploration of the design space, combing efficiency, user-friendliness and flexibility. It uses multivariate analysis algorithms of ModeFRONTIER [9] optimization software in tandem with Grasshopper-based [3] geometry visualization dashboard. The proposed method can additionally be used to explore solutions after running an

SimAUD 2016 May 16-18 London, UK

optimisation; however, it primarily aims at supporting initial knowledge generation before defining the desired design space.

The research is conducted in TU Delft, Faculty of Architecture with the help of ESTECO on using multivariate analysis algorithms. A case study (Jiangmen Sports Center, Jiangmen, China) is provided by South China University of Technology.

2 EXPLORATION OF DESIGN ALTERNATIVES

The interactive evolutionary optimization algorithms focus on including a designer's preferences mostly within the optimization cycle in the phases of selection and breeding [15, 11, 1] or result in visualization after running an optimization process for investigating sub-optimal results [16, 10]. In contrast, this paper focuses on supporting the design space definition before running the optimization (Figure 1).

The design space definition typically describes the following elements: set-up of a model, which usually contains the solver or the simulation for the evaluation of the chosen performance values (both defined numerically and ill defined); establishment of criteria, which must be minimized, maximized or constrained; selection of the parameters and their domains. It is a complex procedure requiring a priori knowledge and experience; however, it also has a great influence on the optimization results. On the one hand, a too limited design space can miss a large number of optimal or sub-optimal solutions; while on the other hand, a too broad design space may result into a number of redundant computations and extended optimization times. In addition, the relevance of the model set-up to the defined problem needs to be tested explicitly to ensure that the subsequent evaluation is able to lead the optimization towards a desired solution.

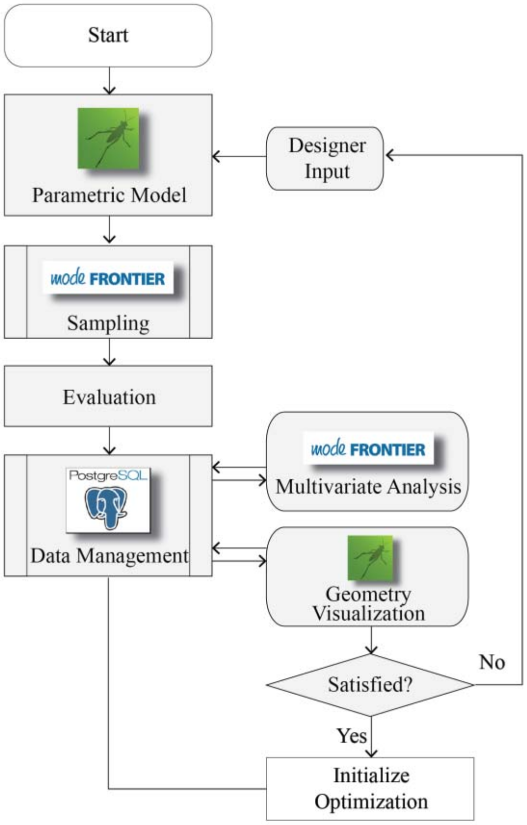


Figure 1. A framework for design alternative exploration performed before defining preferred design space for the following optimization.

The proposed framework for exploration of design alternatives can be seen in Figure 1. A parametric model is developed using Grasshopper plugin for Rhino [3]. Sampling of the design space is consigned to ModeFRONTIER optimization software [9]. The software allows sampling of the design space using such algorithms as Random Sequence, Sobol, Latin Hypercube, Monte Carlo, Cross-Validation, etc. [9]. These methods eliminate subjective bias and allow a good sampling of the design space. The evaluation (or simulation) of the chosen samples can be performed in any desired software both through Grasshopper and ModeFRONTIER environments.

An important requirement for setting up the exploration environment is to have a comprehensive and versatile representation of the results. A large number of design alternatives may be formed by numerous inputs, all yielding multiple performance values. They can form a result space, which is impossible to perceive through conventional data visualization methods like scatter plots or parallelograms. Similarly, such graphs as correlation or scatter matrices are able to summarize the interrelationships, without, however, providing a visualization, which would be intuitive enough to analyse the data. Distinct from the aforementioned methods, multivariate analysis algorithms such as Self Organizing Maps (SOM) and Hierarchical Clustering are more suitable for capturing interrelated effects within a broad design space.

However, the above algorithms can only account for the numerically defined performance values, while the exploration of the design problem requires a portrayal of corresponding geometry. For instance, some certain design variables can have no influence on performance; however, they cannot be discarded from the optimization process due to their influence on aesthetics or similar design criteria.

In order to address this problem, the chosen clusters or parts of SOM are visualized in a Grasshopper-based dashboard through a series of pictures, which correspond to the generated solutions. A PostgreSQL database [13] is used as an intermediary between the two software for efficient organization and querying functionality, as well as preserving a backup of collected data.

Use of the database also facilitates a selection of preferred design direction, which can be chosen using a Grasshopper-based dashboard and fed as a new initialized population for an optimization algorithm. Alternatively, the design space can be redefined and re-explored repeating the previous steps until a commonly acceptable design direction is chosen. The possibility to re-explore previous definitions, as well as combinations of multiple design spaces or their parts, is also enabled by employing the database.

3 MULTIVARIATE ANALYSIS ALGORITHMS IN MODEFRONTIER

ModeFRONTIER, developed by ESTECO, includes a set of multivariate analysis algorithms, some of which are described hereafter.

3.1 Self-Organizing Maps

SOM is an unsupervised neural network algorithm that projects high dimensional data onto a two-dimensional map [5]. The projection preserves the topology of the data so that similar data items will be mapped to nearby locations on the map. This is particularly useful when analysing un-correlated multidimensional data – such as dependencies between numerous design parameters and multiple objectives. Moreover, the representation of SOM is generally summarizing all the dependencies in a 2D space, which is an intuitive visualization technique.

SOM is a sheet-like neural network, with nodes arranged as a regular, usually two-dimensional grid (for an example, see Figure 7). Each node is directly associated with a weight vector. The items in the input dataset are assumed to be in vector format. If *n* is the dimension of the input space, then every node on the map grid holds an *n*-dimensional vector of weights. The basic principle is to adjust these weight vectors until the map represents "a picture" of the input data set. The objective is to achieve a configuration in which the distribution of the data is reflected and the most important metric relationships are preserved. Interest is in obtaining a correlation between the similarity of items in the dataset and the distance of their most alike representatives on the map. In other words, items that are similar in the input space should map to nearby nodes on the grid.

The algorithm proceeds iteratively. In each training step, a data sample **x** from the input space is selected. The learning process is competitive, meaning that a winning unit *c* on the map is determined when its weight vector **m** is most similar to the input sample **x**.

$$\| x - m_c \| = \min_i \| x - m_i \| \qquad (1)$$

The weight vector $\mathbf{m}_c$ of the Best Matching Unit (BMU) is modified to match the sample **x** even closer. As an extension to standard competitive learning, the nodes surrounding the BMU are adapted as well. Their weight vectors $\mathbf{m}_i$ are also "moved towards" the sample **x**. Nodes closer to the BMU will be more strongly adjusted than nodes further away. At the beginning of the learning process, the BMU will be modified very strongly and the neighbourhood is large. Towards the end, only very slight modifications take place and the neighbourhood includes little more than the BMU itself. This corresponds to "rough ordering" at the beginning of the training phase and "fine" tuning near the end.

Since not only the winning node is tuned towards the input pattern but also the neighbouring nodes, it is probable that similar inputs in future training cycles will find their BMU weight vector at nearby nodes on the map. During the learning process, this leads to a spatial arrangement of the input patterns. The more similar two patterns are, the closer their BMUs are likely to be on the final map. It is often said, that SOM folds like an elastic net onto the "cloud" formed by the input data, as can be seen in Figure 3. SOM training ultimately results in a set of weight vectors, which resemble the input data. If one of the components of the weight vectors is reported on top of the grid, each node can be used to display the value of that component for its BMU (algorithm in Figure 2).

3.2 Hierarchical Clustering

Cluster analysis tries to identify homogeneous subgroups of samples in a dataset such that they both minimize within a group variation and maximize between group variations [4]. It can be used to gain insights into the distribution of data

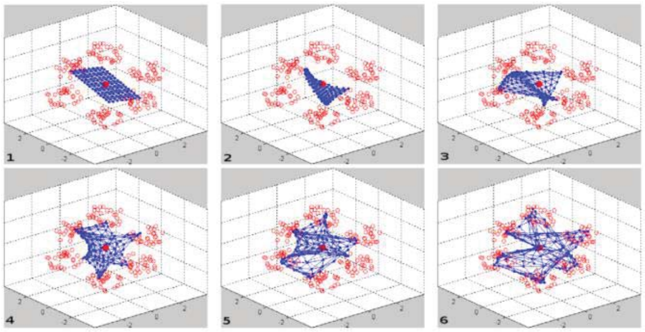
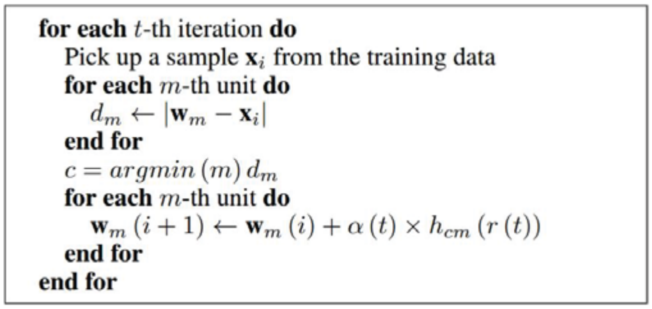


Figure 3. Different phases of SOM training. Three-dimensional input data are represented by empty dots. SOM is the elastic net, which folds to reflect the distribution of input data.



```
for each $t$-th iteration do
    Pick up a sample $\mathbf{x}_i$ from the training data
    for each $m$-th unit do
        $d_m \leftarrow |\mathbf{w}_m - \mathbf{x}_i|$
    end for
    $c = argmin\,(m)\,d_m$
    for each $m$-th unit do
        $\mathbf{w}_m\,(i+1) \leftarrow \mathbf{w}_m\,(i) + \alpha\,(t) \times h_{cm}\,(r\,(t))$
    end for
end for
```

Figure 2. Algorithm of SOM sequential training

within a set, observe characteristics unique to each cluster, and help identifying clusters of interest for further analysis. Graphs like parallelograms, scatter plots and multidimensional scaling algorithms facilitate visualisation of clusters for interpretation purposes.

To perform hierarchical clustering the user must specify a linking method based on which clusters should be merged. It creates nested clusters, not mutually exclusive, where larger clusters in later stages of the agglomeration may contain smaller clusters created at the earlier stages. Different methods exist to merge pairs of clusters at each step, each of which will result in different cluster patterns:

- Single Linkage: the distance between two clusters is the distance between their closest points.
- Complete Linkage: the distance between two clusters is given by the distance between their two furthest points.
- Average Linkage: the distance between two clusters corresponds to the mean distance between all possible inter-cluster pairs.
- Centroid Linkage: the distance between two clusters is the distance between cluster centroids (mean point calculated between all cluster members over all the variables of the data set.
- Ward's method: the distance between two clusters is computed as the increase in the sum of squares of the deviations from the centroid after merging two clusters into a single one. Ward's method minimizes the sum of squares of any pair of clusters to be formed at a given step.

The results of a hierarchical clustering can be represented by a tree chart called dendrogram, in which the dissimilarity between two samples can be read from the height at which they join a single group. Cutting the dendrogram at a specific height will produce a clustering with a selected level of resolution (Figure 4). The cutting level of the dendrogram is a critical choice: if it is too low, a trivial clustering will be obtained of a „one to one" kind where the number of clusters is almost equal to the number of observations. If the cutting level is too high, the "true" cluster structure of a dataset can get over-smoothed.

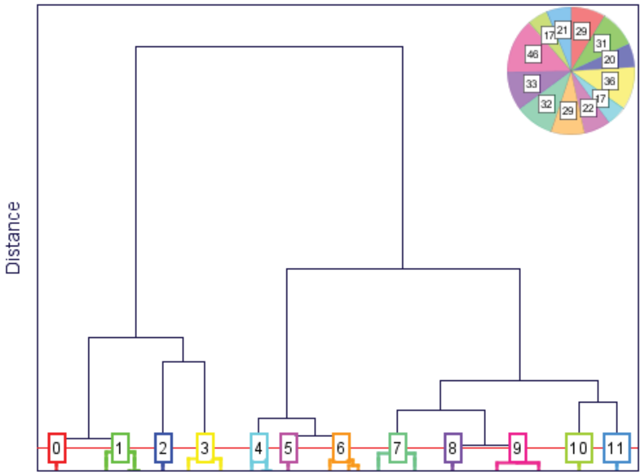


Figure 4. Dendrogram computed by Ward's clustering on the case study dataset based on input perimeters. The pie chart in the corner represents cluster sizes.

A good rule of thumb on where to cut the dendrogram is to look at the largest gap between two successive groupings. Such large gaps stand for good clustering, mainly because adding one more cluster decreases the quality of the global clustering structure, so cutting before such steep decrease occurs is desirable [7].

4 GRASSHOPPER-BASED GEOMETRY VISUALIZATION DASHBOARD

The dashboard has been developed at TU Delft. It is mainly an extension to the previously described multivariate analysis algorithms, which serves as a tool to visualize design alternatives in tandem with the performed analysis. It aims to provide a designer with insight into the relationship between the generated geometry and corresponding performance, as well as facilitate the directing of design space towards solutions that both satisfy architectural and performance-based criteria.

The communication between multivariate analysis performed in ModeFRONTIER and the dashboard is established using PostgreSQL database as an intermediary. The database contains all evaluated design alternatives, including their input parameters, performance values and any additional intermediate information, which may be required by the designer, even if not part of the defined performance values. The solutions are later enhanced with analysis results, indicating the cluster they belong to, Pareto frontier solutions, feasibility information, etc.

4.1 Dashboard functionality

The developed proof of concept of the dashboard includes but is not limited to the following functionalities. All functions are embedded as customized Grasshopper components scripted in Python programming language. Image visualization on canvas is enabled by Embryo plugin [2].

Visualizing typical solutions of SOM

The typical representation of SOM components facilitates understanding of relationships and correlations between all different values and is rather easy to interpret. However, the particularity about dealing with architectural design is that most input values are related to geometry; therefore, each hexagon in SOM can be visualized as a geometrical solution.

Visualization / Exploration of clusters

As mentioned previously, ModeFRONTIER allows various methods of clustering using the same input data. Moreover, clustering can be performed based on various criteria, e.g. clustering based on input parameters will result in having similar designs within one cluster, while clustering based on performance values will allow to explore a range of designs which all yield similar results. Dashboard allows querying the solutions based on a chosen cluster as well as a combination of them.

Additional functionalities

A number of functionalities have been introduced in addition to multivariate analysis.

One of them is exploring Pareto Frontier. Pareto Frontier is a set of solutions, which are not dominated by any of the performance variables. That means that for such set there are no such solutions, which would have better performance in one of the criteria without having a worse one in another. Dashboard allows choosing all the solutions that belong to the Pareto Frontier within the chosen set of solutions (e.g. all values, a single cluster, a combination of clusters, etc.)

Another powerful functionality of the dashboard is its ability to filter the results based on multiple criteria. The criteria can be set based on the additional information (e.g. chosen cluster, Pareto Frontier, feasibility, etc.) as well as based on the desired range of input parameters or performance values, or any combination of those.

4.2 Dashboard elements

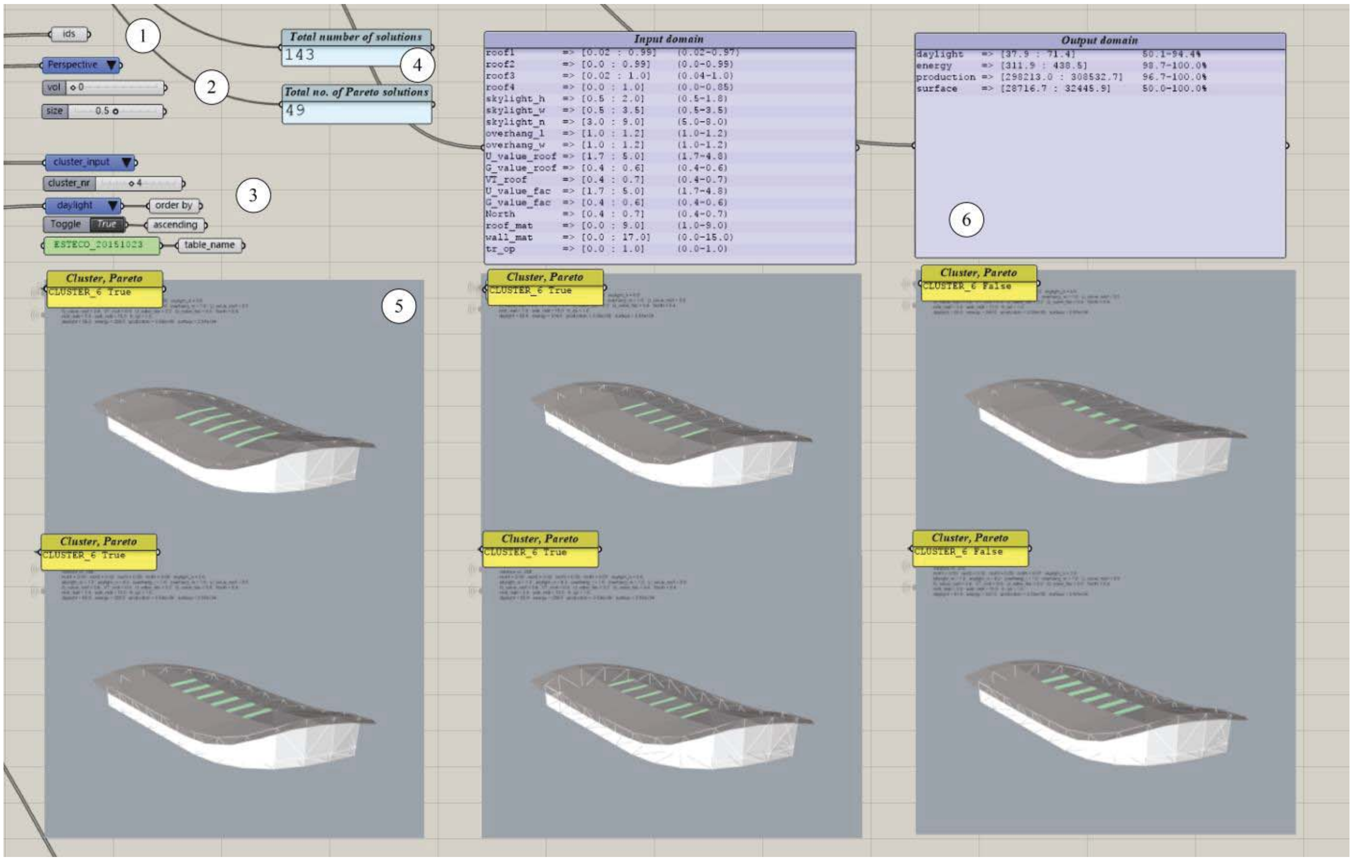
Every set of solutions retrieved from the database by the previously explained functions is portrayed on Grasshopper canvas as shown in Figure 5 together with a number of additional elements.

Querying of design alternatives is performed by customized Grasshopper components based on the designer requirements, retrieving a set of ids (1), which are associated with geometry images. The images are recorded during the evaluation phase of design alternatives. Dashboard user may choose the size and type (perspective, front, back view, etc.) of images to visualize (2).

Images can be sorted based on any chosen numerical criteria, ascending or descending and labelled with a chosen type of cluster (3). The dashboard also gives an overview of the number of solutions, which belong to the chosen set, as well as a number of Pareto Frontier solutions within a set (4).

Images are visualized in sets of six (5) together with the label, which indicates the name of the cluster that the solution belongs to and whether it belongs to the Pareto Frontier.

Additionally, two panels summarize the domains of input and performance variables of a chosen set in comparison to all the other solutions, i.e. they indicate the bounds of a chosen set within the explored design space (6).



Total number of solutions
```
143
```

Total no. of Pareto solutions
```
49
```

Input domain
```
roof1         => [0.02 : 0.99]     (0.02-0.97)
roof2         => [0.0 : 0.99]      (0.0-0.99)
roof3         => [0.02 : 1.0]      (0.04-1.0)
roof4         => [0.0 : 1.0]       (0.0-0.85)
skylight_h    => [0.5 : 2.0]       (0.5-1.8)
skylight_w    => [0.5 : 3.5]       (0.5-3.5)
skylight_n    => [3.0 : 9.0]       (5.0-8.0)
overhang_l    => [1.0 : 1.2]       (1.0-1.2)
overhang_w    => [1.0 : 1.2]       (1.0-1.2)
U_value_roof  => [1.7 : 5.0]       (1.7-4.8)
G_value_roof  => [0.4 : 0.6]       (0.4-0.6)
VT_roof       => [0.4 : 0.7]       (0.4-0.7)
U_value_fac   => [1.7 : 5.0]       (1.7-4.8)
G_value_fac   => [0.4 : 0.6]       (0.4-0.6)
North         => [0.4 : 0.7]       (0.4-0.7)
roof_mat      => [0.0 : 9.0]       (1.0-9.0)
wall_mat      => [0.0 : 17.0]      (0.0-15.0)
tr_op         => [0.0 : 1.0]       (0.0-1.0)
```

Output domain
```
daylight   => [37.9 : 71.4]             50.1-94.4%
energy     => [311.9 : 438.5]           98.7-100.0%
production => [298213.0 : 308532.7]     96.7-100.0%
surface    => [28716.7 : 32445.9]       50.0-100.0%
```

Cluster, Pareto
```
CLUSTER_6 True
```
Cluster, Pareto
```
CLUSTER_6 True
```
Cluster, Pareto
```
CLUSTER_6 False
```
Cluster, Pareto
```
CLUSTER_6 True
```
Cluster, Pareto
```
CLUSTER_6 True
```
Cluster, Pareto
```
CLUSTER_6 False
```

Figure 5. Example of Grasshopper-based dashboard for geometry visualization.

5 CASE STUDY

The swimming pool of the Jiangmen Sports Centre is used to demonstrate how the support system works. The project is located in Guangzhou (China) and will be used for sport events on a national level.

5.1 Design space

The input parameter space is composed of 18 variables, including such geometrical parameters as roof curvature, skylight number and their dimensions, construction material properties, building rotation and presence of glazing. The given bounds of parameters are rather wide, aiming to avoid too limited design space at the first exploration.

Four performance values have been set as objectives:

- Maximizing of average UDI (Useful Daylight Illuminance) value; it stands for the percentage of daytime hours with an illuminance level falling between a minimum and maximum threshold set as comfortable, according to the program requirements during the opening hours of defined design period [12].
- Minimizing of the energy need (kWh/m2) in terms of cooling and heating, meaning the energy needed to be supplied to or extracted from a space in order to keep a comfortable thermal environment according to the building program, measured over a defined period.
- Maximizing of energy production value; it is defined as the amount of electrical energy that can be produced by a roof surface if a certain percentage of it is covered with Photovoltaics.
- Minimizing the area of roof surface results in a lower amount of materials needed for the construction of the envelope, meaning less embodied energy and lower construction costs.

An ill-defined criterion is set for building aesthetics, which is decided using the expertise of a designer.

Grasshopper plugins Honeybee and Ladybug [6] together with simple mathematical functions (e.g. surface area calculator) are used as external solvers for the evaluation of performance values. Design space sampling has been performed by ModeFRONTIER using Uniform Latin Hypercube algorithm with 340 alternative designs, out of which 334 were evaluated as feasible.

5.2 Multivariate Analysis

Two types of hierarchical clustering have been applied on all feasible solutions. Both types used Ward’s approach, since it was proven to provide clusters the most similar in size. One approach has clustered solutions based on input parameters with 17 to 46 solutions in each of the twelve clusters, while another one has been applied on performance values (Figure 6) with 32 to 54 solutions in each of the eight clusters.

SOM has been trained using the four performance values of all feasible solutions. Although the SOM is trained using only the objectives as training components, ModeFRONTIER also records the variation of other parameters as separate component maps. The resulting SOM hexagonal grids for each of the input parameters and design objectives can be seen in Figure 7.

Every hexagon in a grid corresponds to a set of real or virtual solutions in a way the nearby hexagons are more similar than the distant ones. Every hexagon holds a single value for each of the parameters and objectives; therefore, it is possible to visualize every point on a grid using the proposed dashboard. It must be noted that the colours in hexagonal grids always correspond to the numerically normalised values rather than the actual range of parameters, i.e. even a Boolean value would be displayed as blue corresponding to False (or 0), red to True (or 1) and a gradient in between, which would rather display the fuzziness between the two.

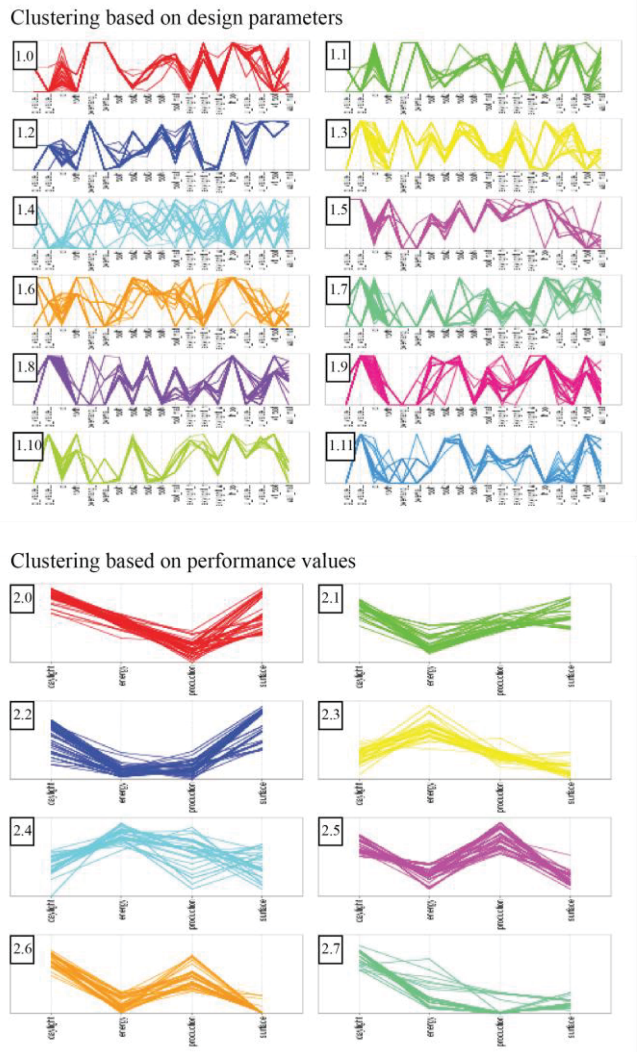


Figure 6. Every cluster is represented by its own Parallel Chart, helping to discover which variables determine the cluster structure as indicated by the internal and external similarity values. These charts represent visual means for assessing the compactness and the uniqueness of clusters.

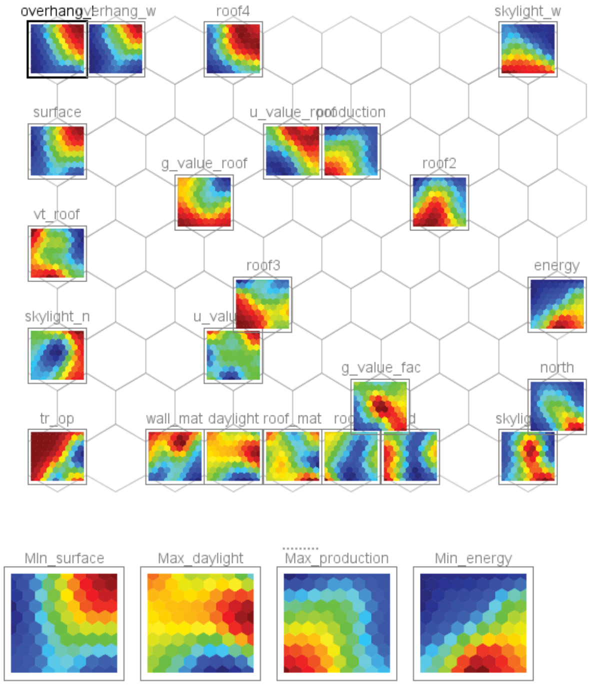


Figure 7. The chart shows how input parameters and performance values are distributed on the SOM hexagonal grid for the described case study. Similar component maps are placed in adjacent positions in order to spot correlations. The 4 rectangles at the bottom refer to the design objectives (as described in section 5.1), blue color corresponding to low numerical values, red – high numerical values and gradient colors referring to the in-between space.

5.3 Geometry Visualization

Four types of geometry images have been recorded while running the evaluation of design alternatives: perspective, top, side and front view. All images contain text with the solution's ID and both values of input parameters and performance.

Additional images are generated for each of the grid cells of SOM in order to visualize chosen geometries quickly.

5.4 Results

After exploring the results of multivariate analysis combined with the geometry visualization on the dashboard, a number of conclusions were drawn about the relationships between the parameters and performance values. It has also facilitated modifying the initial broad design space could also be modified in such a way that it would lead the subsequent optimization towards the preferred direction. Part of these insights is presented hereafter.

While browsing through the clusters based on specified input variables (Figure 6, top), it has been noted that cluster 1.3 which can be characterized by small skylights, no facade glazing and non-expressive roof, results in poor aesthetical performance. The same can be noted about clusters 1.7 and 1.9, which have bigger skylights. However, they display a box-like appeal not compliant with the initial concept of the building. By looking at the performance domains of these clusters, it is obvious that they are not falling into the range of the highest performance values. Therefore, such sets can be safely discarded.

Clusters based on output variables (Figure 6, bottom) suggest that the most desired behaviour can be found in clusters 2.5 and 2.6, where many designs with expressive roof curvature can be found that is also preferred by the designer. It can be noticed that only certain types of wall and roof materials fall within these clusters, however, since they are mostly not influencing building aesthetics (with some exceptions), they can be chosen purely based on numerically expressed performance values. Clusters 2.3 and 2.4, which both have very poor performance, suggest that a combination of small skylights and glazed facade, as well as big skylights concentrated close to each other, results into poor performance and should be avoided in future designs.

SOM components chart (Figure 7) very clearly demonstrates that surface area and energy production are contradicting values, which have high correlation with overhang length and width. Since designers' preference is to have as large overhang as possible, it is suggested that the value can simply be decided in a way that it somewhat satisfies both of the criteria, especially taking into account designer's opinion. Even though it is expected that overhang size should influence energy need for cooling and heating and UDI, because it protects the facades from direct solar radiation, it is obvious that in this case there is almost no influence on these objectives, therefore the parameters can be kept stable for further optimization.

Furthermore, while analysing all of the available chart and the dashboard, it has been noticed that only some certain configurations of roof curvature values result in both aesthetically appealing and well-performing designs. Since these combinations of curvature values are not bounded within a single domain, varying constraints can be set to capture the preferred design direction. For example, if values „roof1" and „roof2" are similar, values „roof3" and „roof4" should be as high as possible; if there is a high difference between value „roof1" and „roof2", then „roof3" and roof4" should stay the same. It is also important that the difference between „roof1" and „roof3" is not bigger than the one between „roof2" and „roof4".

All the mentioned insights have been used to adjust the initial design space before running an optimization process. Additionally, a number of important dependencies have been noted which value may exceed the specificities of a particular project.

6 DISCUSSION AND FUTURE WORK

The developed proposal is work in progress; therefore, many more tests need to be performed, including more case studies with varying disciplines and multiple objectives, different sampling algorithms and a number of design alternatives. The work should also be validated towards its benefits for the optimization results compared to results obtained while using only human expertise for setting up the design space.

Currently, the method is designed for the Rhino/Grasshopper environment, but may also be implemented in other environments, such as Revit/Dynamo.

Considering the multivariate analysis, the drawback of clustering is that it does not provide clear insights into the correlations. On the contrary, the drawback of SOM is the visualisation of geometries, since in case there are 100 hexagons, 100 geometries need to be visualized, overburdening the gain of insights. A good solution for these problems seems to be clustering performed on SOM. When SOM is trained based on outputs, clustering can be based on inputs that allow grouping similar geometries and exploring the relationships between them all at once.

7 CONCLUSIONS

This paper has presented an interactive environment that integrates guidance-based support for exploration of the design space. The proposed method is computationally efficient and integrated into an environment familiar to architects. It relies on algorithms available in ModeFRONTIER software together with database querying capabilities available in PostgreSQL and a developed dashboard, which uses the Grasshopper interface.

The proposed method has demonstrated that it is able to support the exploration of design space and facilitate its definition in order to lead the design towards the preferred direction. At the current stage of the method's development, it is possible to tell which design variables can be discarded and which parameter domains need to be limited, considering both numerical and ill-defined performance values. In addition, the method provides insights about introducing varying constraints or their combinations.

Finally, a designer is provided with additional knowledge about dependencies between design variables and their combinations, which yield particular performance values.

ACKNOWLEDGMENTS

This research was supported by the 2015 Open Project of State Key Lab of Subtropical Building Science, South China University of Technology (Grant No. 2015ZA01). It was also supported by the Sports Engineering Development Fund, TU Delft Sports Engineering Institute, and the Urban System and Environment (USE) Joint Research Centre between SCUT and TU Delft.

The Jiangmen Sports Centre has been designed by Sun Yimin Studio of the Architectural Design & Research Institute of South China University of Technology.

REFERENCES

1.Danhaive, R.A., Mueller C.T. Combining parametric modeling and interactive optimization for high-performance and creative structural design. Proc. *IASS Symposium*. As of August 2015.

2. Embryo, http://www.grasshopper3d.com/group/embryo as November 20, 2015

3. Grasshopper 3D, http://www.grasshopper3d.com/ as of November 20, 2015

4.Jain, A., Murty, M., and Flynn, P. Data clustering a review. *ACM Computing Surveys 31* (1999).

5.Kohonen, T. Self-Organizing Maps. *Springer*, 1997.

6. Ladybug + Honeybee, http://www.grasshopper3d.com/group/ladybug as of November 20, 2015

7.Manning, C., Raghavan, P., Schtze, H. Introduction to Information Retrieval. *Cambridge Universtiy Press*, 2008.

8.Martens, B, Wurzer, G, Grasl T, et al. Analysing Populations of Design Variants Using Clustering and Archetypal Analysis. Proc. *33rd eCAADe Conference* – Vol. 1, pp. 251-260. As of 16-18 September 2015

9. modeFRONTIER® is a product of ESTECO SpA http://www.esteco.com/modefrontier as of November 20, 2015

10.Mourshed, M., Shikder, S., Price, A. Phi-array: a novel method for fitness visualization and decision making in design optimization. *Journal of Advanced Engineering Informatics*, Elsevier, Vol.25, pp. 676–687, 2011.

11.Mueller, C. T., & Ochsendorf, J. A. Combining structural performance and designer preferences in evolutionary design space exploration. *Automation in Construction*, 52, 70-82, 2015

12.Nabil A., Mardaljevic J. Useful Daylight Factors. *Energy and Buildings*, 38(7), 2005.

13. PostgreSQL, http://www.enterprisedb.com/ as of November 20, 2015

14.Scott S.D., Lesh N., Klau G.W., Investigating human–computer optimization, Proc. SIGCHI Conference, pp. 155–162, 2002.

15.Turrin, M., von Buelow, P., & Stouffs, R. Design explorations of performance driven geometry in architectural design using parametric modelling and genetic algorithms. *Advanced Engineering Informatics*, 25(4), 656-675, 2011.

16.Yang, D., Sun, Y., Turrin, M., von Buelow, P., & Paul, J. C. Multi-objective and multidisciplinary design optimization of large sports building envelopes: a case study. Proc. *IASS* . As of August 2015