Proceedings of the IASS Annual Symposium 2017
"Interfaces: architecture.engineering.science"
25 - 28th September, 2017, Hamburg, Germany
Annette Bögle, Manfred Grohmann (eds.)

Automated performance-based design space simplification for parametric structural design

Nathan C. BROWN*, Caitlin T. MUELLER[a]

*[a] Massachusetts Institute of Technology
77 Massachusetts Ave, Cambridge, MA, USA 02139
*ncbrown@mit.edu

Abstract

As computation has advanced, more designers are becoming familiar with parametric and performance-based design space exploration, techniques that can provide feedback and guidance even in early-stage design. However, two downsides of such techniques are the time and expertise required for problem setup, and the potential for the large volume of generated data to become overwhelming and difficult to absorb. Researchers must find ways to organize performance-based information and simplify exploration so that the design process is more manageable, while ensuring that performance feedback leads to better outcomes. This paper proposes two new applications of traditional optimization methods that can help simplify early-stage architectural or structural parametric design. The first involves analyzing the design variables considered in the problem, ranking their importance, and determining which ones should be eliminated or emphasized during exploration. The second method clusters designs into families and enables designers to cycle through these families during exploration. Two structural design case studies are presented to illustrate the possibilities created by variable analysis and clustering in conceptual, performance-based design.

Keywords: conceptual design, optimization, problem formulation, parametric design, main effects, clustering

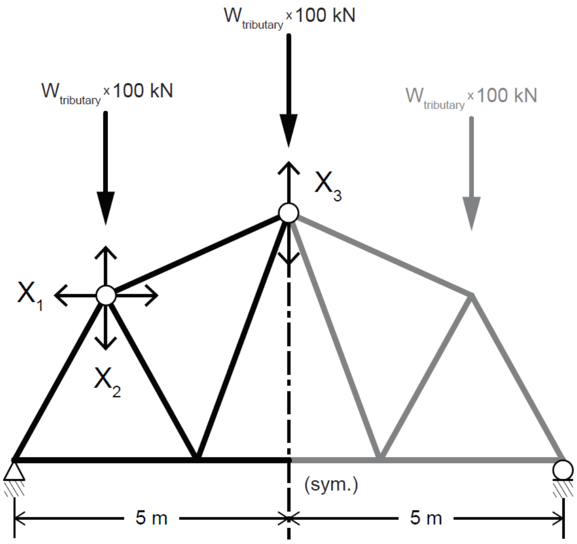
1. Introduction and background

In architecture and engineering, creative designers are increasingly amenable to using optimization techniques, as performance-conscious design has gained both popularity and accessibility. The use of optimization in structural engineering has a long history, but it has only recently become a promising conceptual design strategy for designers looking to achieve high-performance structures. This is partly due to difficulties with conducting optimization in a way that is flexible, interactive, and rapid enough to be valuable for early-stage design. Researchers have created new methods for interactive, performance-based conceptual design (Mueller and Ochsendorf [11]), showed that surrogate modeling can be used to speed up analysis (Tseranidis *et al.* [17]), developed workflows for the efficient creation of design catalogs or databases (Turrin *et al.* [18]), and demonstrated their utility in structural and other performance-based architectural design (Flager *et al.* [5]; Brown and Mueller [2]). However, one drawback to optimization approaches like these, when compared to typical, performance-free geometric design environments, is the time required to set up an optimization workflow and determine which constraints, variables, and bounds are important. Even as computational speed improves, which enables software to generate more performance feedback, architectural and structural designers interested in performance-based design often must commit considerable time to creating a model and testing which parts of the design space are most worth visiting. Additional effort is required to sift through all potential design options and various feedback streams. Consequently, there is a need for methodologies that move beyond the cataloging of parametric designs, allowing designers to quickly and systematically explore both the properties of parametric models themselves and the most consequential types of designs that they produce.

This paper addresses the drawbacks to performance-based modeling by proposing two methods for automatically simplifying a design space given an initial conceptual design. The first method applies statistical techniques to consider potential design variables, eliminate unimportant ones, and allow the user to concentrate on only the variables he or she would like to consider. The second method uses design space clustering to adjust variable bounds and direct users to regions of the design space that are high-performing. Two case studies involving fixed-typology truss structures are presented to demonstrate the utility of these methods. Although both procedures require initial objective function evaluations to link design possibilities with their performance, they are implemented in a way that automates this tedious part and thereafter enables creative, interactive, and rapid exploration of design choices that are pre-selected to be most significant in terms of performance.

2. Methodology

This section describes both simplification techniques in detail, using a case study to illustrate concepts. The problem used is a simply supported eleven bar truss, loaded at its three interior nodes. There are three design variables, which correspond to the width of the corner node, the inverse height of this node, and the height of the central node. The height of the central node is defined with reference to the starting point of the corner node, meaning that high values always generate deeper trusses. This eliminates many unrealistic truss configurations in which the corner and central nodes are on different sides of the straight chord of the truss. The eleven bar truss problem was selected because the outcomes of both variable selection and clustering should be intuitive to structural designers—the depth of the truss is related to its performance, and there are local minimums both above and below the constant chord. The design space and variable bounds for the setup are described in Figure 1.



Symbol	Variable	Range
X_1	x-location of corner node	$0\text{ m} < x < 3.33\text{ m}$
X_2	y-location (height) of corner node	$-6\text{ m} < x < 6\text{ m}$
X_3	y-location (height) of middle node	$0*X_2 < x < 2*X_2$

Figure 1: A description of the geometry used in the eleven bar truss case study.

2.1. Simplification through variable selection

The first method for simplifying the design space analyzes the relationships between variables and performance in a systematic way using the calculation of Main Effects (Box *et al.* [1]). This procedure is common in optimization outside the fields of structural and architectural design as a first pass in model setup, after which conceptual designers can adjust their design space to include only the variables that matter most. This calculation involves creating an experiment in which different variable combinations are simulated based on an orthogonal matrix, and the performance results for each variable setting are compared to the overall average. The "effect" of the variable setting is defined below, where m_{xi} is the mean of all experiments that contain that setting, and m is the overall mean for the group of experiments:

$$\text{Effect } X_i = m_{x_i} - m \quad (1)$$

Although the raw effects can provide insight into which variable settings are better for performance, the authors propose taking the average magnitude of each level effect for a given variable as a measure of its importance to the design problem. After running the analysis, designers can adjust a variable's bounds or freeze it at a certain setting, effectively removing it from the problem. In the eleven bar truss, the calculation of effects is visualized in Figure 2, where 1.00 is the normalized score for a minimal weight structure, meaning that negative effects (ones that lower the weight) are desirable. The effects clearly show a hierarchy of significance between variables, informing the designer that depth at the middle of the truss is most important, followed by the depth of the corner point, and finally the overall width. This exercise only requires four evaluations, but immediately pushes the performance-conscious designer to suggested areas of the design space.

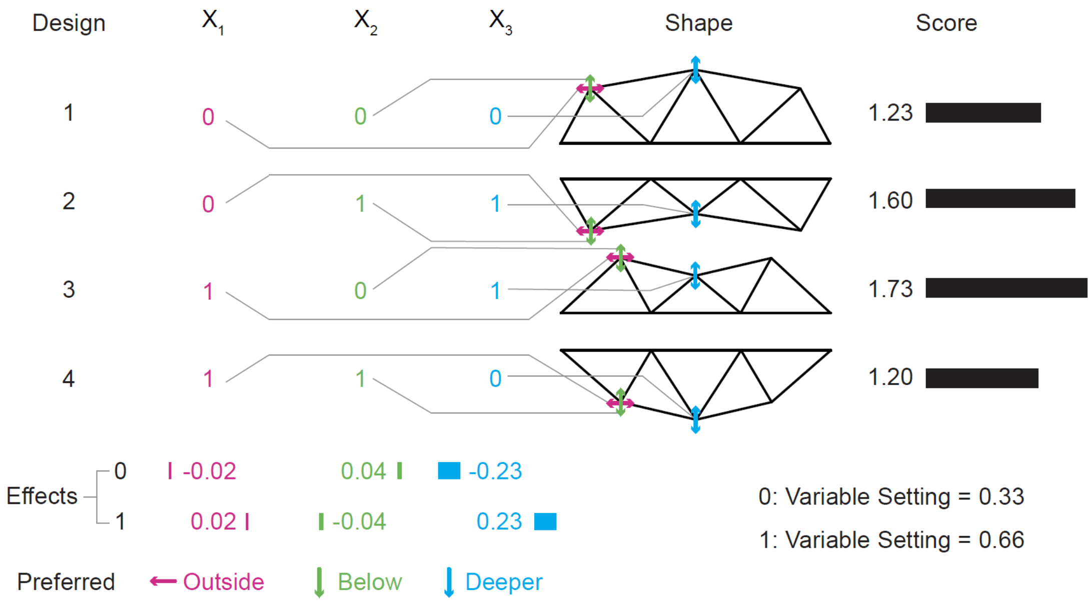


Figure 2: The results of a two-level effects calculation. Based on these samples, the highest performing trusses are deep in the center, have an outer tension chord below the supports, and are wider, in that order of importance.

The calculation of effects is recommended as an efficient first experiment for understanding the design space in a variety of engineering fields (Phadke [12]). The technique does have weaknesses in that the specific settings chosen can have a large influence on the problem, and it does not account for highly curved, volatile design spaces or interactive effects between variables. Specifically, calculating interaction between variables is a planned next research step in this area. In addition to the proposed metric, the authors considered other statistical techniques for measuring variable importance, including raw correlation, stepwise regression, summing coefficients during principle component analysis, and measuring integrals of dense and sparse areas of the design space. After experimentation on test problems, the calculation of effects proved to be the fastest means for obtaining a rough sense of how much each design variable matters, which is the goal of applying such techniques to early-stage design. In the future, the calculation of effects or any other related methodologies that can accurately return a list of which variables matter most could be used to make parametric problem setup much more accessible to less technical designers. For problems with common variable types, such as truss shape optimization, the entire process could be automated, so that a designer need only to sketch an idea in software before a computer tests all possible variables and returns the ones that are important for performance.

2.2. Simplification through clustering

The second methodology aims to simplify the design space by concentrating on types of designs rather than individuals. Since each parametric model is defined by a design vector, the classification of designs into types or families can easily be completed using clustering algorithms. Data clustering, or the unsupervised classification of patterns, has long been an area of fundamental research, with applications in a wide variety of fields (Jain *et al.* [8]; Kaufman and Rousseeuw [9]). This paper proposes using k-means clustering (Hartigan and Wong [7]) to separate the full design space into distinct families for early parametric design exploration. K-means clustering works by randomly selecting *k* cluster centroids in the design space, partitioning the data set with respect to these centroids, updating their location based on the newly created sets, and repeating this sequence until convergence. However, the application to performance-based design space exploration first requires the creation of an initial dataset. Although a random or representative sample of the entire design space could serve this purpose, a performance-directed sampling or optimization is recommended to create this dataset, ensuring more meaningful families than with random initial sampling. Related applications of k-means clustering in architecture include categorized design databases (Sicilia *et al.* [15]) and Biomorpher, an interactive genetic algorithm (Harding and Olsen [6]). However, clustering can also be used while exploring parametric models directly using sliders, as demonstrated in this paper.

Figure 3 shows an example clustering of the eleven bar truss, using a dataset generated from the history of a 20-generation evolutionary algorithm. The best performing cluster has a near-optimal depth on the tension side and the tightest variable bounds. The second cluster performs almost as well, but is not as wide. The third cluster is the most geometrically diverse based on an average of the measurements of design space diversity in Brown and Mueller [3], and contains most of the designs above the supports. The variable plots show the settings that tend to occur for the highest performing designs, which can supplement the initial knowledge gained by the earlier effects calculation. Like the first method for variable analysis, there are some limitations of the clustering procedure. There is randomness in the initial optimization process and k-means itself, meaning that the clusters are not necessarily replicable. As such, the results of the case studies represent only one possible outcome when conducting clustering. The quality of results also depends significantly on the selection of the particular dataset used to perform the clustering. Nevertheless, when conducted appropriately, design space clustering can be a vast improvement to weighing each design individually, whether exploration takes place live on a parametric model or through previously generated options.

2.3. Implementation in parametric design

In order to test the utility of variable analysis and design space clustering, and to make these techniques immediately available to designers, the authors created custom Grasshopper [14] components that calculate variable effects and automatically cluster a given design space. These components were written in C# using Microsoft's Visual Studio. A diagram of the inputs and outputs for each component is given in Figure 4. The first component, called Effects, takes in variable sliders, an objective score, the number of levels, and the desired settings for each level. Since the calculation of effects relies on orthogonal matrices, and there is no automated way of generating such a matrix for any set of variables, experiments are currently limited to 2 or 3 level settings and fewer than 13 problem variables based on matrices defined by Leung and Wang [10]. Once the variables, objective score, and other settings have been established, double-clicking Effects runs the experiment by cycling through the required designs and returning both the magnitude of variable importance and the overall raw effects for each experiment. The designer can use this information to adjust how the variables are being used in the manners described in Section 2.1.

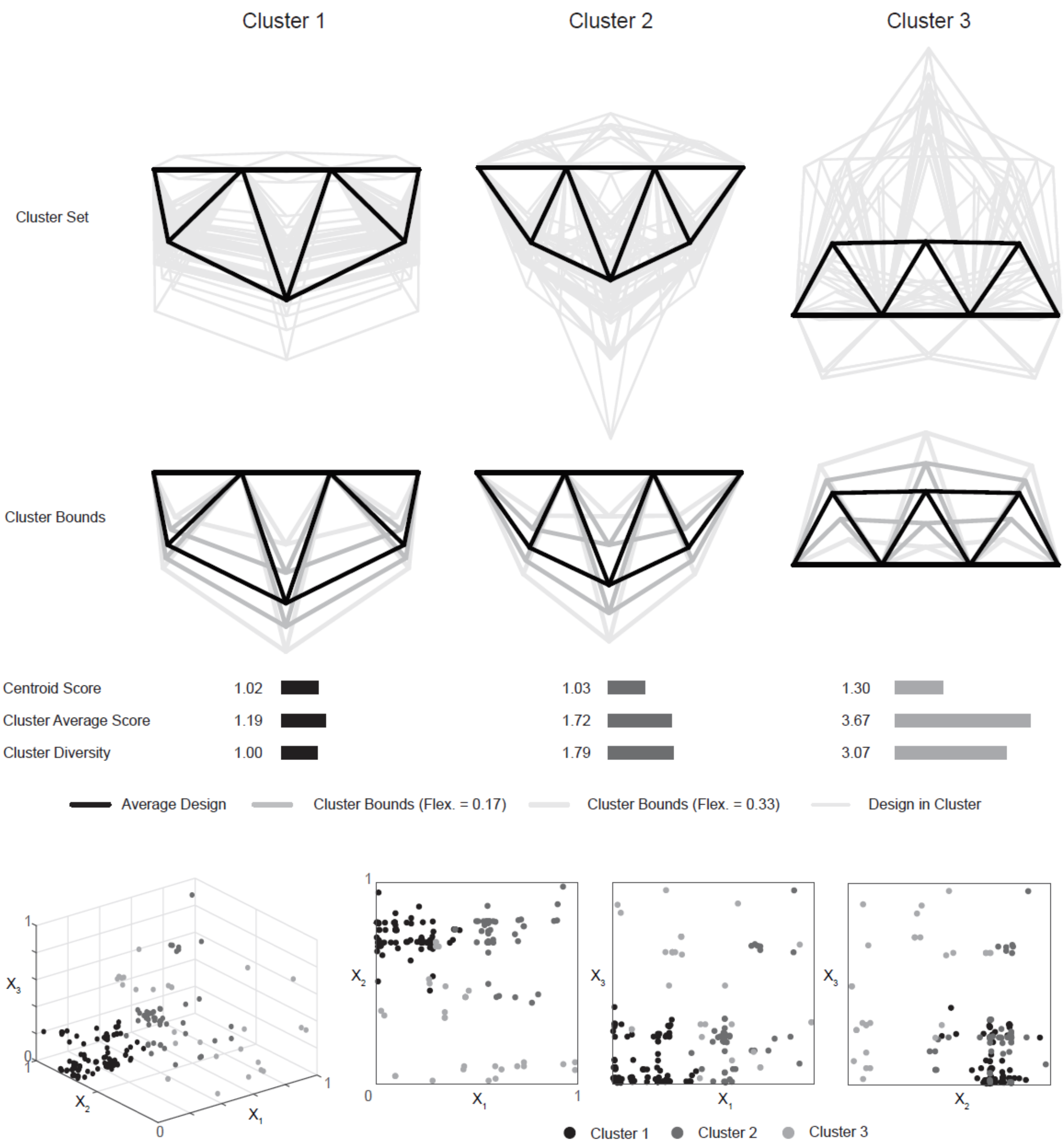


Figure 3: The results of clustering the eleven bar truss. The first row provides every design in the cluster, and the second row shows reset variable bounds for different flexibility settings, which are explained in Section 2.3. Note that higher normalized variable settings correspond to a truss that is narrower (X_1), deeper below the supports (X_2), and deeper in the middle (X_3). The diversity metrics are normalized by the least diverse cluster.

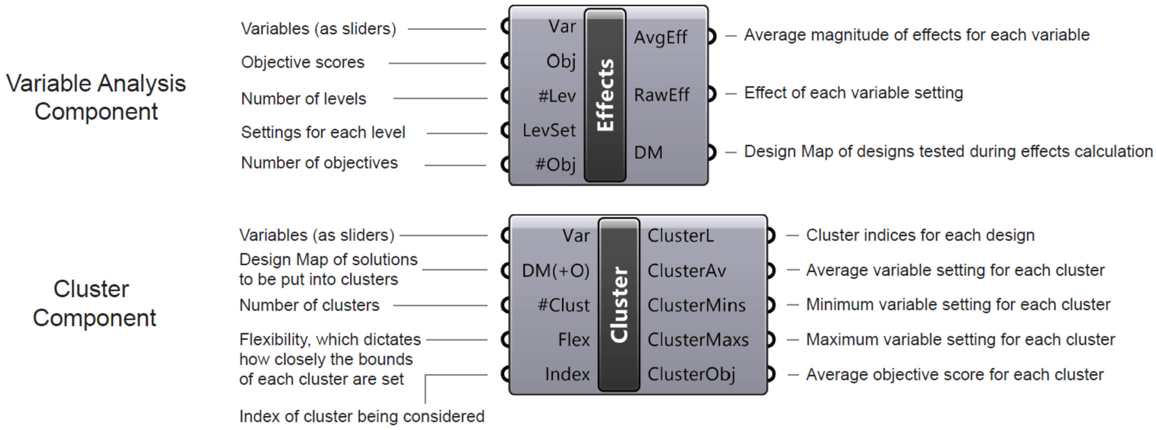


Figure 4: Inputs and outputs for custom Grasshopper components that calculate main effects and cluster the design space.

Cluster takes in variable sliders, a historical map of previously simulated designs, the number of desired clusters, and a flexibility setting that dictates the sensitivity of bounds resetting for cluster-based exploration. Upon double-click, the component runs the k-means clustering algorithm using the Accord library (Souza [16]) and returns the cluster index, average, minimum, maximum, and average objective score for each variable and each cluster. Once the designs have been sorted into clusters, the user can then use the index input to cycle through each cluster for exploration. When resetting the index and double-clicking again, Cluster goes directly to the average for the cluster and resets the bounds for each slider to the flexibility setting multiplied by the distance between the average and maximum/minimum of the cluster. Although the user might have to experiment with which setting offers the desired balance between flexibility and fidelity to the design family, once these bounds have been reset, the designer can quickly explore the portion of the design space that corresponds to the given cluster. Since the component also outputs the cluster objective averages, designers can go right to the highest-performing cluster. Even if another cluster is selected for inspection, the designer will have a better understanding of how the designs being explored are likely to perform.

3. Full design case study – cantilevered canopy

3.1. Problem setup

In this section, the methods described in Section 2 are employed on a more extensive case study to demonstrate their utility in parametric, early-stage structural design. This case study is based on the design problem proposed by Danhaive and Muller [4]—more details about the structural model can be found by consulting this reference. In the case study, the overall shape of a cantilevered roof canopy is determined by defining the coordinates of a steel space truss. The structure is supported at four locations, and the design variables are the z-direction displacements at certain positions along the top and bottom of the space truss. Similar to the procedure for the example eleven bar truss, the authors implemented a parametric definition of the truss in Grasshopper, using Karamba (Preisinger [13]) for the structural analysis and a sizer to calculate section sizes that are sufficient for the axial, bending, torsion, sheer, and buckling loads imposed on each member. The objective score is the weight of the entire steel structure based on these member sizes. All results have been normalized so that the lowest weight truss generated during the optimization has a score of 1.00. Every other design score shows how much worse that design performs than this optimum.

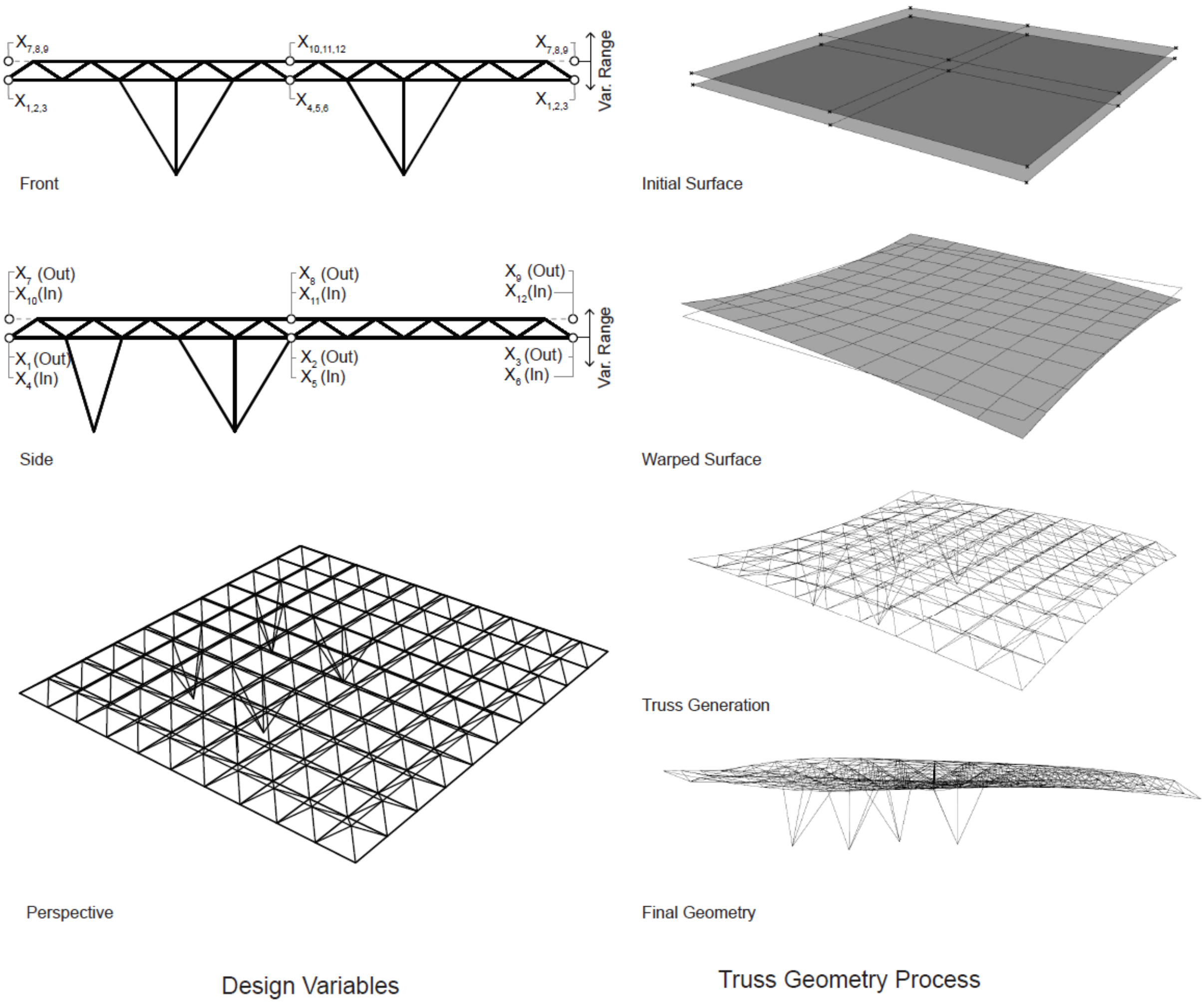
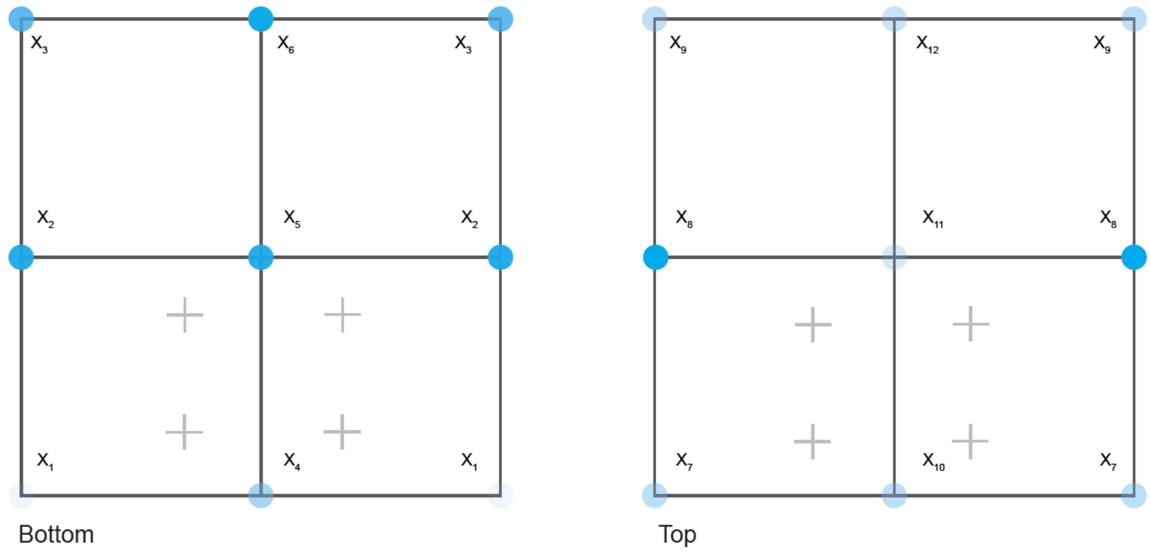
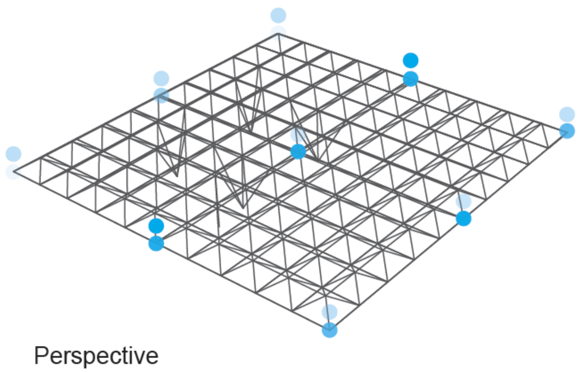


Figure 5: The design variables and procedure for defining the geometry of the steel space truss canopy.

3.2. Variable analysis

Once the design space was constructed, the authors initiated a three-level main effects experiment, with variable settings of -3.0, 0.0, and 3.0 for the three levels. For a 12 variable problem with three levels, a larger orthogonal sampling matrix with 27 designs was required. The results of this effects simulation are given in Figure 6. As with the simpler truss, the analysis shows a hierarchy of importance between the different variables, although the relationships are more difficult to explain completely since the problem is more complicated. The variable settings for depth nearest to the interior supports appear most significant, since the highest global moment occurs at these locations, requiring structural depth. In general, the variables in the bottom of the roof seemed to affect the performance score most, although this could be a consequence of the specific orthogonal matrix used. The bottom nodes prefer lower settings and the top nodes prefer higher settings, since both of these geometric translations add overall depth. In terms of magnitude, adjusting the single, highest importance variable could affect the weight of the structure by over 25% compared to the optimal, if all other settings are constant. It must again be stressed that these results depend on a largely stochastic workflow and ignore the potentially substantial interactive effects between variables. Nevertheless, with only a limited amount of computational effort, a designer has considerably more information about which variables to emphasize and which to ignore when conducting exploration.





	Effects			
Variable	X = -3.0	X = 0.0	X = 3.0	Avg. Magnitude
X1	0.017	-0.034	0.017	0.023
X2	-0.314	-0.047	0.360	0.240
X3	-0.094	-0.200	0.294	0.196
X4	-0.144	-0.017	0.161	0.108
X5	-0.359	0.020	0.339	0.239
X6	-0.339	-0.042	0.381	0.254
X7	0.123	0.001	-0.124	0.083
X8	0.390	-0.020	-0.370	0.260
X9	0.117	-0.112	-0.005	0.078
X10	0.014	0.112	-0.126	0.084
X11	0.038	0.039	-0.078	0.052
X12	-0.087	0.096	-0.009	0.064

Figure 6: In this figure, the blue color gradient corresponds to the importance of each variable, with a darker blue representing a higher average effect.

3.2. Clustering

Following the calculation of variable effects, the design space was further simplified using the Cluster component. In order to perform the clustering and achieve meaningful results, a performance-based design map was again generated using the history of a single optimization run. An evolutionary algorithm with a generation size of 20 was simulated for 10 generations, for a total of 200 example designs. The results of the design space clustering are provided in Figures 7-8. As with the eleven bar truss case study, the score of the centroid design and the cluster average score are provided, as well as a measure of design space diversity. In general, these clusters can be ranked in terms of performance, with the best performing clusters often showing the tightest variable bounds and lowest design diversity. Although the centroid is shown in the visualization, the average performance score of each cluster is used to rank the clusters by performance, since average score is the initial result generated by the Cluster component and gives a more general idea of how the entire cluster performs. Amongst the clusters, more depth generally leads to lower weight structures. As such, exploring within Cluster 1 can lead to better design outcomes, depending on how flexibly the designer tunes the bounds. Like the other clusters, however, this group still offers curved, visually elegant solutions, even if they tend to be higher performing on average.

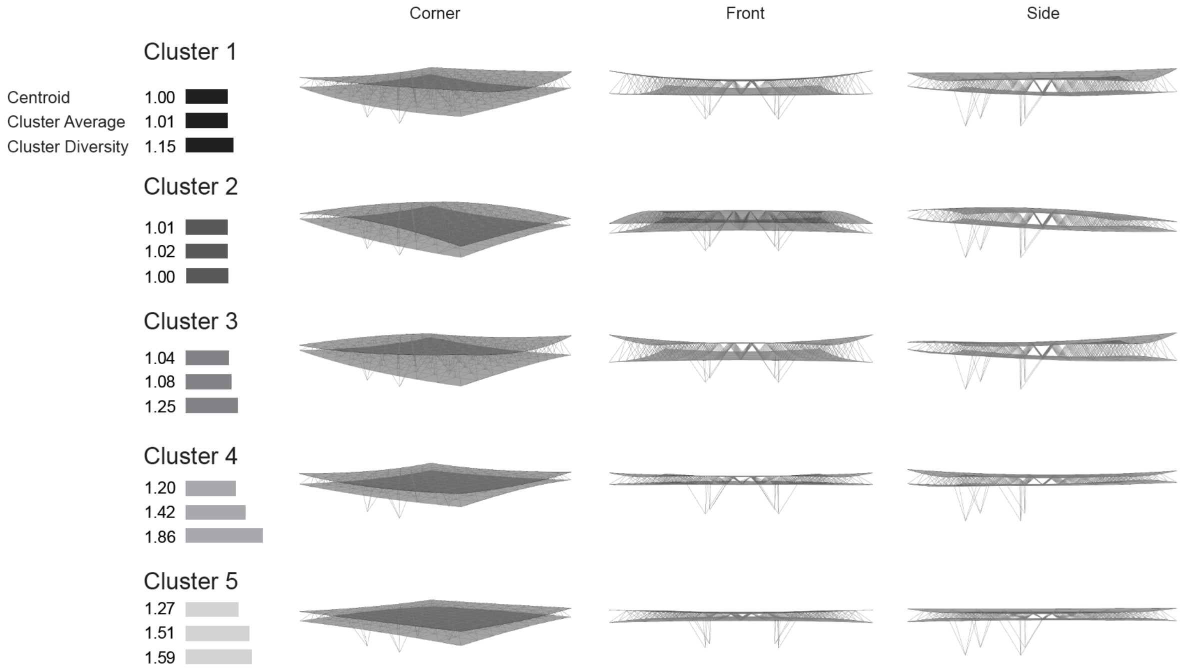


Figure 7: The various design space clusters for the canopy model, in order of performance.

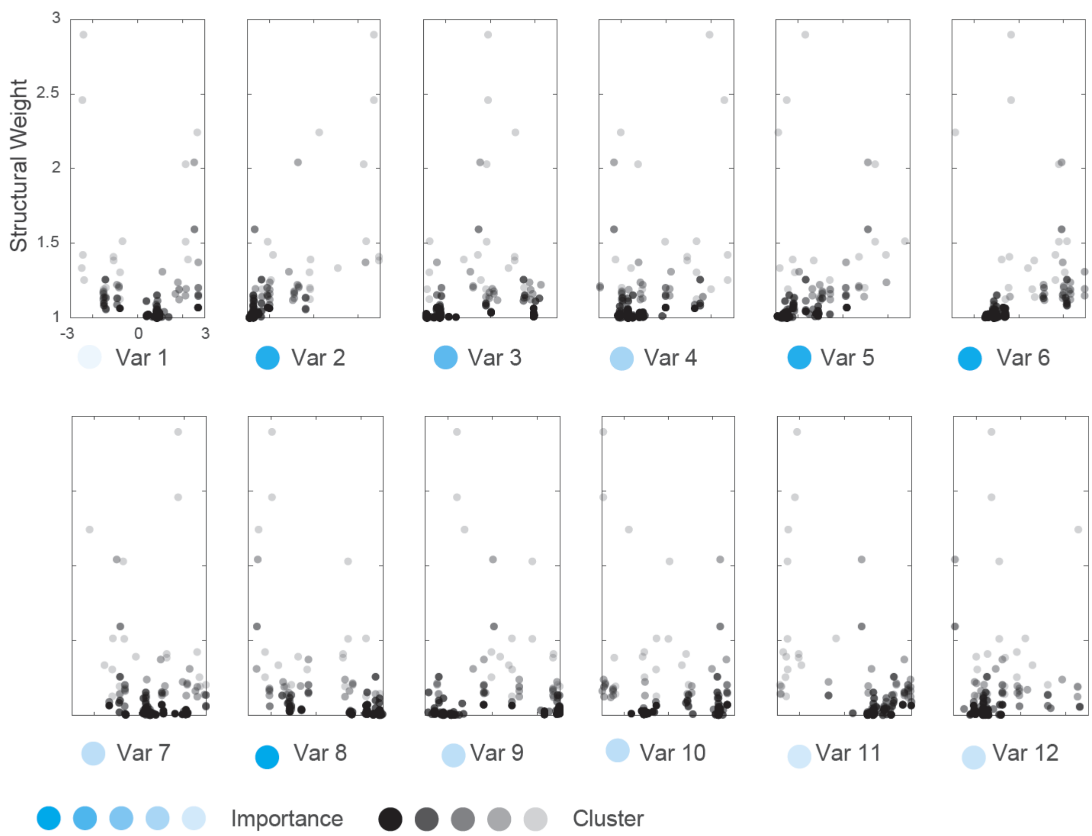


Figure 8: A projection of the variable settings for each cluster. For some variables, high-performing designs clearly prefer a certain setting, such as in Variable 2 and Variable 6.

4. Summary of contributions and conclusions

In conclusion, this paper has proposed the application of two statistical analysis techniques, variable effects calculation and clustering, to the conceptual design of buildings and structures. It has also introduced components developed by the authors that enable the direct use of these methods within typical design software. Finally, it has presented two structural case studies that demonstrate the benefits, limitations, and future possibilities of calculating effects and design space clustering. Although future research can extend these methods by accounting for interactive effects and increasing automation of design space formulation, this paper moves beyond existing techniques for interactive design and catalog generation and enables designers to better focus on high-performance regions of the structural and architectural design space.

References

[1] Box G.E.P., Hunter J.S., and Hunter W.G., *Statistics for Experimenters: Design, Innovation, and Discovery*. (2nd ed.), John Wiley & Sons, 2005.

[2] Brown N.C. and Mueller C.T., Design for structural and energy performance of long span buildings using geometric multi-objective optimization, *Energy Build.*, 2016; **127**; 748-761.

[3] Brown N.C. and Mueller C.T., Quantifying diversity in parametric design, 2017 (manuscript under review).

[4] Danhaive, R.A. and Mueller C.T., Combining parametric modeling and interactive optimization for high-performance and creative structural design, in *Proceed. of the IASS Ann. Symp.*, 2015.

[5] Flager F., Welle B., Bansal P., Soremekun G., Haymaker J., Multidisciplinary process integration and design optimization of a classroom building, *Electron. J. Inf. Technol. Constr., 2009;* **14**; 595-612.

[6] Harding J and Olsen, C.B., Biomorpher, 2017, www.grasshopper3d.com/group/biomorpher.

[7] Hartigan J.A. and Wong M.A., Algorithm AS 136: A K-Means Clustering Algorithm, *J. R. Stat. Soc. Ser. C, Applied Statistics*, 1979; **28**; 100-108.

[8] Jain A.K., Murty M.N., and Flynn P.J., Data clustering: a review, *ACM Comput. Surv.*, 1999; **31** 264-323.

[9] Kaufman L. and Rousseeuw P.J., *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 2005.

[10] Leung Y.W. and Wang, Y., An orthogonal genetic algorithm with quantization for global numerical optimization, in *IEEE Trans. Evolutionary Computing*, 2001; **5**.

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

[12] Phadke M.S., *Quality Engineering Using Robust Design*. Prentice Hall, 1995.

[13] Preisinger, C., Linking structure and parametric geometry, *Arch. Design*, 2013; **83**; 110-113.

[14] Robert McNeel and Associates, Grasshopper, www.grasshopper3d.com.

[15] Sicilia, Á., Madrazo, L., and González, M., Applying clustering techniques to retrieve housing units from a repository, in *Design Computing and Cognition '10*, 2011; 387-401.

[16] Souza, C., Accord.NET, 2008-2015. www.accord-framework.net/index.html.

[17] Tseranidis S., Brown N.C., and Muller C.T., Data-driven approximation algorithms for rapid performance evaluation and optimization of civil structures, *Automat. Constr.* 2016; **72;** 279-293.

[18] Turrin M., Von Buelow P., and Stouffs R., Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms, *Adv. Eng. Informatics,* 2011; **25;** 656-675.