Implementing data-driven parametric building design with a flexible toolbox approach

Nathan C. Brown*[a], Violetta Jusiega[a], and Caitlin Mueller[a]

[a] Building Technology Program, Department of Architecture, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139, USA

* Corresponding author. Present/Permanent Address: Department of Architectural Engineering, The Pennsylvania State University, 104 Engineering Unit A, University Park, PA 16802, USA.

Tel. +1 814 404 9959 Fax. 814-863-4789

Email address: ncb5048@psu.edu

Abstract

Designers in architecture and engineering are increasingly employing parametric models linked to performance simulations to assist in early building design decisions. This context presents a clear opportunity to integrate advanced functionality for engaging with quantitative design objectives directly into computational design environments. This paper presents a toolbox for data-driven design, which draws from data science and optimization methods to enable customized workflows for early design space exploration. It then applies these approaches to a multi-objective conceptual design problem involving structural and energy performance for a long span roof with complex geometry and considerable design freedom. The case study moves from initial brainstorming through design refinement while demonstrating the advantages of flexible workflows for managing design data. Through investigation of a realistic early design prompt, this paper reveals strengths, limitations, potential pitfalls, and future opportunities for data-driven parametric design.

Keywords

Multi-objective optimization; design space exploration; interactive optimization; structural design; sustainable design; conceptual design; surrogate modeling; energy simulation

1 Introduction

Optimization and related techniques have been gaining traction in early building design, especially with increasing access to parametric modeling and the direct link various plug-ins offer between simulation engines and geometry [1–7]. While they have some limitations [8], these design approaches enable the use of building performance simulations to drive early decisions [9], rather than simply confirm or validate initial design choices. This paper is motivated by a desire to increase the accessibility of such design tools, since efforts in this area can multiply the usage of data-driven approaches and their subsequent impact in practice [10,11]. Ideally, these tools should be flexible and easily integrated into existing design approaches. Technologically savvy designers already generate design space catalogs [12,13], conduct architectural optimization [14–21], integrate technical and architectural design goals [22,23], and even implement surrogate modeling and other techniques in their workflows [24–29].

Those designers who are comfortable with coding have found considerable support through open-source libraries, integrated development environments, and various scripting methods [30,31]. A segment of the design community often prefers to work on the cutting edge, frequently manipulating code in a raw form. There have also been efforts in both academia and practice to educate architects and engineers and increase their capacity for computational design through visual programming and coding, which is likely to have a substantial effect on how buildings are designed. This democratization and customization of computational design approaches has many benefits, while creating a constantly evolving and improving shared framework for performance-based parametric design [32,33].

Yet at the same time, many practicing architects and engineers do not have the background, interest, or time to become full-time software developers themselves. Many prefer to spend their time concentrating on creating new designs, rather than improving computational workflows. These designers, both professional and academic, depend on some level of existing software or functionality to suit their needs, and frequently opt for standard visual programming interfaces rather than textual programming [34]. The risk for them, however, is that existing software may not be exactly what is needed, and it may exert more of an influence on their process than they desire, down to the particulars of a digital interface. For these designers, it is worthwhile to find a satisfactory compromise that balances flexibility and accessibility.

This paper describes a computational design toolkit that seeks to achieve such a balance by bringing data-driven approaches directly into a common parametric environment. It acknowledges that visual programming is becoming a viable medium for widespread parametric design exploration in practice, due to its greater level of accessibility compared to raw code. In this context, data-driven approaches can take advantage of the considerable effort within the design community to directly connect parametric geometry to performance simulation. While describing an entire toolbox, this paper reveals how individual components with specific functions involving data science and optimization are combined to enable customized, data-driven design strategies. It then presents an integrated early design example tracing possible workflow progressions through design space formulation, diversity-based brainstorming, and interactive optimization, while making use of various computational components.

The culminating design example involves selecting the geometry for a long span athletic center roof, while considering implications for structural and energy efficiency. The example shows, in practical terms, the advantages and possibilities of using these tools, as well as the limitations and complications that must be overcome in their application to realistic building design problems.

2 Background

Performance-based parametric design operates in the conceptual "design space", which contains all possible options that can be generated by a parametric script, and the "objective space", which locates these designs based on how well they perform. The goal is to explore the design space (by adjusting the variables that control the current design iteration) with reference to the objective space (using simulation to understand performance). In many cases, designers use a systematic approach for "searching" or "exploring" the design space rather than manually controlling each variable. Through this process, designers seek feedback about how different possibilities behave, as well as guidance, which involves suggested directions for modification that correspond to performance improvement. As a result, designers require functionality within parametric design environments that allows for generating options, running simulations, discovering trends, filtering the search, adjusting variables, implementing optimization, and a variety of related actions. Yet the process of parametric design is not linear or perfectly defined—these individual activities may occur iteratively, at different phases, or in combination. Rather than a rigid platform for repeatedly exploring a problem with the same approach, some designers require flexibility.

In response to the widespread need for quantitatively exploring and visualizing the design space, researchers have developed tools that offer different mixtures of these capabilities. To distinguish the specific contributions of this paper towards data-driven design within a shared parametric environment, a brief summary of overlapping functionality is given now. This section will focus on tools in Grasshopper [35], which is the platform used by the authors. In this environment, tools for conducting design space exploration primarily fit into three categories, with some spanning in between: (1) parametric toolboxes, (2) optimization solvers, and (3) sampling interfaces. Parametric toolboxes include Lunchbox [36], TT Toolbox [37], and Dodo [38], which contain components for geometry processing as well as some machine learning and optimization functionality. Optimization solvers include Galapagos (native to Grasshopper), stormcloud [39], Biomorpher [40], Goat [41], Silvereye [42], Opossum [43], and Octopus [44], which includes interactive evolution and supervised learning components. Tools focused on sampling and visualization include Generator [45], Genoform [46], Conduit [47], and Colibri, which is included with TT Toolbox and connects to Design Explorer [48]. Recent machine learning tools include Owl [49] and Crow [50]. Some analogous functionality for exploration and optimization is also available in visual programming tools connected to BIM software, such as Optimo [51] for Dynamo [52], and Project Refinery [53].

These examples are prominent among users of graphical algorithm editors, but this list is not exhaustive. Since parametric design platforms encourage continuous coding and modification, others may have developed similar functionality on their own. Some architecture or engineering firms have in-house developers creating digital tools for design exploration among their own studios or teams. These groups may be formal or informal, and concentrated or distributed across a large firm. However, with a few notable exceptions, many privately developed tools are not publicized by firms or made freely available for broad use during parametric design.

Although there are advantages and corresponding applications for each of these mentioned tools, this paper introduces the DSE toolkit, which combines data-driven components together in a shared format, allowing them to be easily linked together during explorative design processes. Breadth of available methods and easy transfer of information between components is not always possible in parametric design, especially for tools that rely on a specific custom interface. DSE also contains components that offer new functionality not available elsewhere that is focused on data science applications specific to early design, such as variable analysis and transformation. However, in some cases, the components could be used in conjunction with the tools listed above in customized workflows.

Overall, the toolkit approach towards design proposed here addresses many core needs of performance-driven parametric designers. At the same time, it achieves greater flexibility and accessibility than tools that require specialized knowledge to operate in a precise way. The toolkit application on a full-scale conceptual building design example in this paper also reveals nuanced and instructive interactions that can occur between interdisciplinary design goals.

3 Workflow methodology

3.1 Design Space Exploration overview

This section first describes the functionality developed as part of the Design Space Exploration (DSE) toolkit. It then outlines selected workflows and corresponding modes of interacting with parametric design that are enabled by the tools. This list of workflows is not limiting, and others have used components in the DSE toolbox on a variety of applications [54–56], since the intention is to create a flexible mixture of data science and optimization functions to allow designers to create their own approaches.

Design Space Exploration was developed in collaboration with students and researchers affiliated with the Digital Structures Research Group in the MIT Department of Architecture. It is an open-source plug-in for Grasshopper and consists of components for creating design space catalogs and conducting machine learning, optimization, and design space organization. Some of the code, including significant portions related to surrogate modeling, was developed as part of [57]. However, the other workflows listed here were proposed and designed by the authors, with some development support from research assistants. Contributors to the software are listed in the acknowledgements.

DSE is not a simulation engine itself for predicting and understanding building performance, such as EnergyPlus [58] or structural finite element modelers [59]. Nor is it a plug-in that connect geometry to performance simulation engines, such as Diva-for-Rhino [60], Honeybee [61] or parts of the framework in [62]. Rather, it is designed to connect to the combination of any numerical design variables, geometry, and corresponding simulations, as described in Figure 1.

Some DSE components rely on external libraries, including Accord.NET [63], Math.NET [64], and JMetal [65]. Design Space Exploration is freely available online for users of Grasshopper and could be developed for other parametric software in the future. The toolkit is in ongoing development.

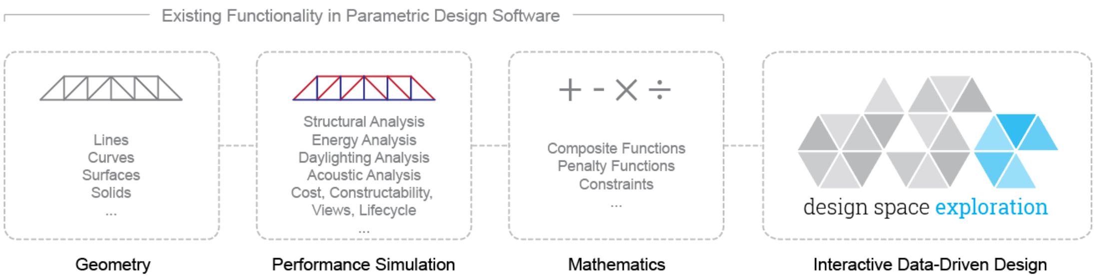

Figure 1: Diagram of the relationships between parametric components in a data-driven workflow. Design Space Exploration is a plug-in for in Grasshopper, which works with other native components and third-party plugins to connect to simulation engines for performance evaluation.

3.2 Possible workflows with DSE

The separate DSE components developed for visual programming can operate on a parametric design space in a variety of ways, described in **Error! Reference source not found.**. What follows is a description of relevant workflows enabled by components, and a contextualization of these workflows within typical designer behaviors. These workflows are not exhaustive due to the customizability of a flexible toolkit. The standard method for interacting with a performance-based parametric model (Workflow 1) is to modify the sliders, view the geometry, run a simulation, and then view the results. For rapid evaluations, the simulation and corresponding visualization may be completed automatically and update every time the slider is adjusted. This base relationship between variables, geometry, simulation, and output is the fundamental building block of all other workflows.

Figure 2: Different workflows enabled by the various data-driven components in DSE. Each blue box (denoted by a component logo) refers to a specific component.

A catalog approach to parametric design repeats the basic structure by automatically generating multiple designs and simulating their geometry in sequence, as in Workflow 2. While conducting this automatic generation, decisions must be made about the method and resolution of sampling in the design space. In addition to basic grid sampling functionality, in which designs are selected at even increments across each design variable's range, DSE adds random and Latin Hypercube sampling procedures and separates the selection of sampling type from its more general iterator, providing increased control over the sampling method, scope, and resolution. At the same time, this separation allows the iterator component to operate on any list of designs recorded in the same format—for example, to take screenshots of every design along a Pareto front as found by a multi-objective optimization algorithm, as described in Workflow 3.

Other workflows build on the basics of design space exploration and catalog creation by enabling supplemental methods from data science and engineering. Workflow 4 shows how one might use a diversity filter to assist with meaningful brainstorming within parametric software. In this workflow, the designer can use a dataset generated either through sampling, the history of an optimization run, or another technique. This dataset may contain hundreds or thousands of designs, which are not all worth considering individually. The diversity component first allows the users to select target performance objectives and filter out all designs that are not within a specified range of the target. Then, it asks the designer for a number of representative designs he or she would like to consider, before using the diversity measurements in [66] to generate a highly diverse, curated set of samples for consideration.

The next possible workflow saves time, preserves original parametric relationships, and prevents clutter on a visual scripting canvas while transforming design variables using mapping coefficients, as described in [67]. For this task, a DSE component reads in sliders, a design space scale, and a set of coefficients, which may be calculated within or outside Grasshopper using data science approaches. These coefficients can be numbers, which corresponds to linear mapping of the original variables, or they can be dynamic and depend on the script itself. Once these coefficients are established, a user can create separate synthetic variable sliders to control the design, which override Grasshopper's main solution structure and adjust the original sliders that are still connected to the geometry and simulation. Since new sliders do not have to be reconnected each time, designers can use this workflow to rapidly cycle through synthetic variables.

The remaining workflows described here feature direct applications of classification and supervised learning for design space exploration. Workflow 6 involves calculating effects and cluster-based design exploration—these techniques are described in greater detail in [68]. Workflow 7 depicts how a surrogate modeling component trains a predictive model of objective performance based on an existing dataset. Once the model has been appropriately trained, the simulation can be turned off, and the geometry can be manipulated with only the surrogate model information showing essentially in real-time depending on geometric complexity. In Workflow 8, these live predictions have been plugged into a general interface for gradient-based interactive optimization, as described in [69]. The interactive optimization component of DSE offers the most engaging separate interface for moving in both the design and objective spaces, but has significant requirements, including previous simulation or objective functions that can be approximated with reasonable accuracy.

3.3 Designer behavior with DSE

These workflows span across stages of parametric design and their corresponding behaviors, but trend towards late conceptual design activities. A variety of models exist to describe design behaviors. One of relevance is Gero's FBS model [70], which has been applied in cognitive design studies for parametric design [71,72]. While considering the FBS model, most behaviors enabled by this toolkit involve reasoning about the solution space through synthesis, analysis, evaluation, and potential reformulation. The toolkit is primarily for understanding the behavior of a structure (B_s), comparing it to expected behavior (B_e), and manipulating a design description (D), rather than the initial Formulation itself. Another model for digital design is presented by Oxman [73], which describes four classes of traditional design activities: representation, generation, evaluation, and performance. Based on Oxman, most workflows envisioned with these components activate a performance-based generation model by providing both performance feedback and guidance in various forms.

More specific is Geyer and Beucke's description of interactive cycles for Multidisciplinary Design Optimization in AEC, which proposes a relationship between a designer and an optimization process [74]. A modified version of this framework is provided in Figure 3. Not every behavior enabled in the DSE toolkit involves optimization directly, and so a parallel strand of systematically exploring options has been added. The possible DSE workflows from Figure 2 are labeled in relation to these actions, describing when they might fit into the process. In addition to the applications shown here, individual components within DSE might have broader connections—for example, the function that calculates the set design diversity might be used in the filtering workflow described, or it might be used as an optimization constraint or objective function. The overall goal is to provide support and workflow flexibility for approaching conceptual design, rather than imposing a prescriptive procedure. While Figure 2 provides context for how and when the toolkit might typically be used, further customizable workflows are possible.

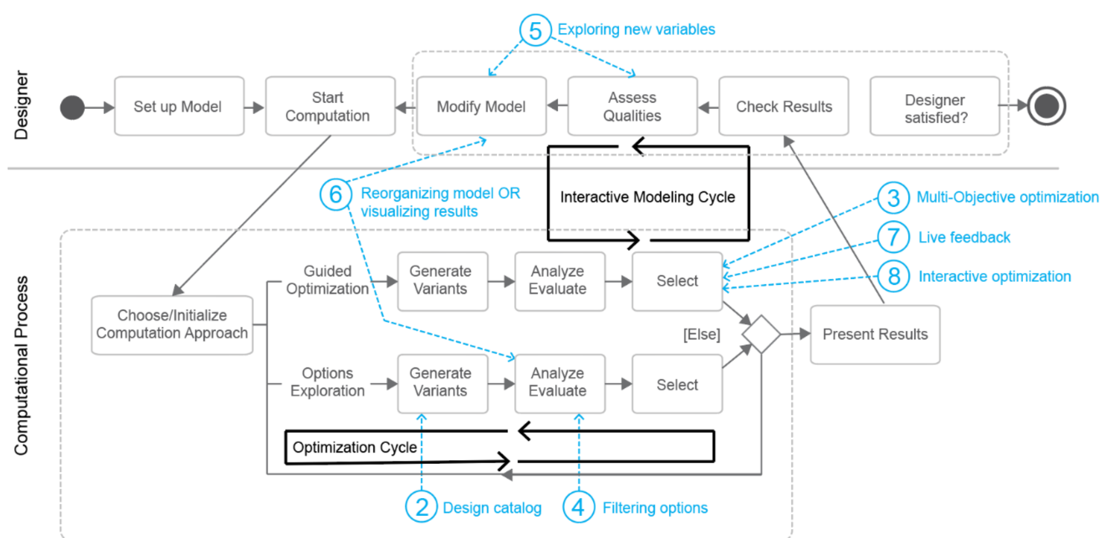

Figure 3: The proposed workflows from Figure 2 projected onto a modified "Interactive workflow for design process and optimization", adapted from [74] *and* [75].

The primary contribution of this section is the mapping of component relationships and how they can be structured to enable effective integration of interactive optimization and data science in parametric design. Next, implementation and testing of these tools on a comprehensive example yields additional insight into data-driven early design processes.

4 Example of conceptual grid shell roof

4.1 Case study description

This section provides an example of how the DSE components could be used in sequence to pursue parametric design. It begins by analyzing and modifying a parametric design space during formulation, then uses a diversity filter to select specific directions for further exploration, and finally demonstrates how interactive optimization can be used locally on a selected working design concept. There are some parallel aspects or repeated steps in the process for demonstrative purposes, as a designer would not need to consider every individual approach simultaneously. However, this example shows how the flexible approach to tool and interface development offers designers a buffet of data-driven methods that can stimulate creativity and support design ideation throughout a computational process.

The selected case study is the design of an athletic center for a campus environment in Boston, MA. For the case study, it is assumed that the design team decided on a hybrid structural system involving a curved grid shell roof, which can be supported on large external columns, as well as directly on the ground. When the edge of the grid shell is lifted off the ground, the resulting gap can be filled with a mixture of opaque wall and transparent glazing. Thus, across the various configurations, the primary structural action may be arching in compression, or it may be spanning in bending between columns. Structural models for this example account for bending by allowing members to grow deeper, which approximates the effect of adding global depth to the roof surface when required.

For many of the possible designs, the columns form clusters or tripods, which assist with lateral loads as well as gravity loads. The column clusters are especially important for variants with flatter roofs that are entirely column-supported, but less so for arching structures that transfer loads directly to the ground. The overall massing is explored by considering different boundary conditions, curvatures, and orientations within the original design concept. Due to its adjustable variables, the design space contains considerable freedom, and the massing and structural system decisions have both performance and visual implications.

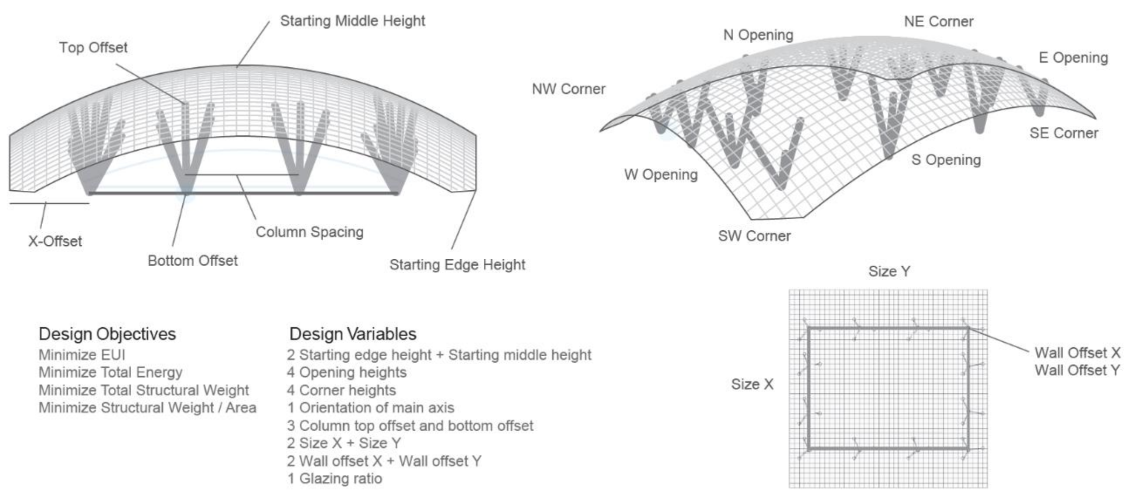

Figure 4: A visualization of design objectives and variables for the case study.

The case study has four numerical objectives, in addition to an acknowledged desire for a visually expressive structure. Two are related to structural performance (total structural weight and structural weight / area), and two are related to energy performance (total annual energy and energy use intensity for the enclosed portion). These objectives are common in parametric structural design and parametric sustainable design, and have been considered together in previous research [55,76–78]. This example primarily focuses on the normalized objectives, which are structural weight / area, abbreviated as structural material quantity (SMQ), and energy use intensity (EUI). The initial structural simulations were conducted using Karamba's optimize cross section feature, which runs a finite element analysis and performs an iterative procedure to size each member for axial, bending, shear, and local buckling. The initial energy simulations used Diva-for-Rhino to generate a multi-zone model adequate to building codes for Boston. Full details concerning model settings can be found in [79], which used the same base simulations. The goal of the case study is to achieve a high-performance design according to these four objectives, but in the context of a natural design process in which other aspects of the design may influence decision-making and should be considered simultaneously.

A few distinct building types within this design space are shown in Figure 5, along with a comparable precedent for each geometry. These possibilities include cantilevered spanning structures, arches, vaults, and other variations of a typical long span design. Although the column configurations change the force flow for some of these examples, many behave in a similar manner to their precedents. As is true in the built environment, some of these designs have clear structural logic and will likely perform well in that domain, while others will not. By considering such a wide design space, it is possible to see how a designer might use the approaches in this paper at different scales for global brainstorming, local optimization within an already sound concept, or a combination of the two.

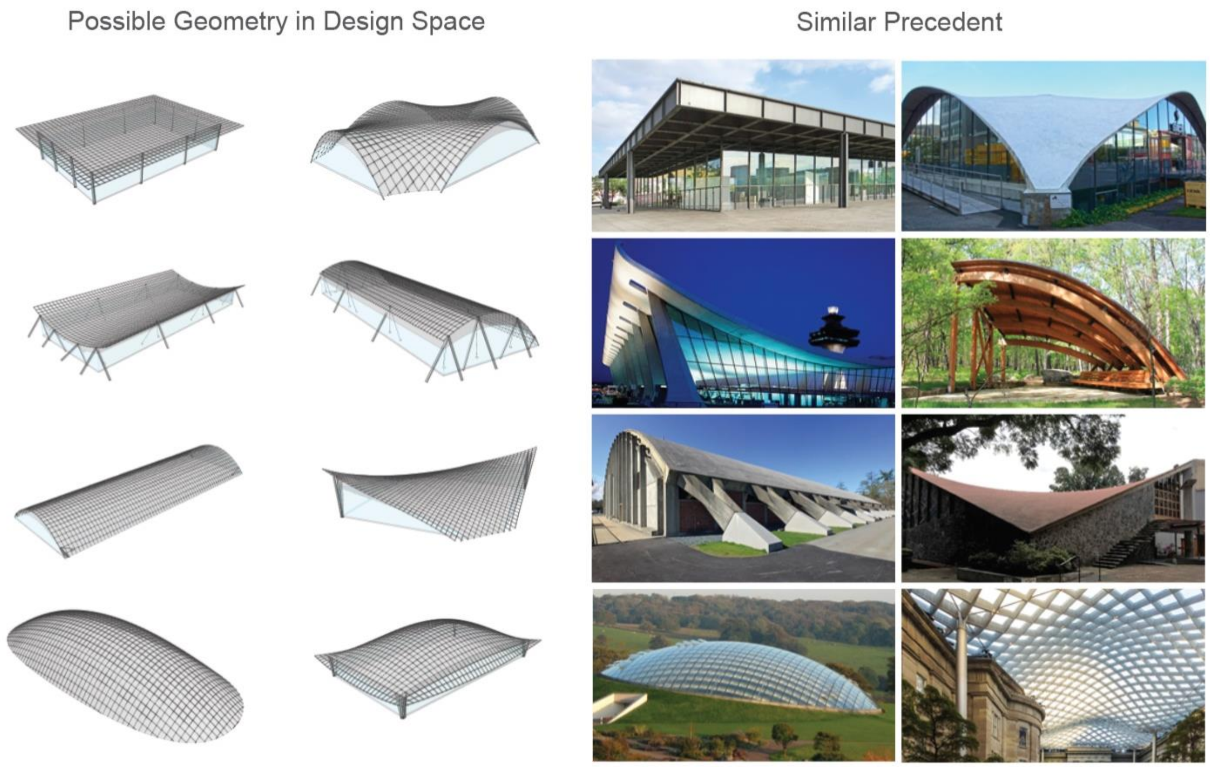

Figure 5: Potential options within the design space for this example, along with building precedents they resemble. Clockwise from top left are the Neue Nationalgalerie in Berlin [80]*; SICLI Company Building in Switzerland* [81]*; Tulip Tree Shelter in Bentonville, AR ; El Altillo in Mexico City; Kogod Courtyard in D.C.; Great Glasshouse in the National Botanic Garden of Wales ; Thompson Area in Hanover, NH* [82]*; and Dulles International Airport in Virginia* [83]*. Photos not credited are by the authors.*

4.2 Early design space analysis

Initially, a designer might begin exploring the design space through direct slider manipulation, sampling, or optimization, in accordance with common methodologies. Prior to a more exhaustive sampling or optimization procedure, when variables are still being established, one useful approach is to gain a quick understanding of how the variables affect the problem in terms of performance. Intuitively, the designer might decide to begin this exploration by focusing on one objective first, due to experience, interest, or prioritization. In this case, structure is considered first, since there is a strong relationship between geometry and structural weight that will become clearer as the example progresses. A first pass method for calculating variable importance for structure is demonstrated in Figure 6. This effects calculation was conducted using three levels corresponding to variable settings at 0.25, 0.5, and 0.75 of the available range.

Since there were more than 13 variables, which is a limit of the orthogonal array implemented in DSE, two separate calculations were conducted and then normalized using an effects calculation that included variables from both original groups. The results of this analysis indicate that column spacing, edge start, and overall size have a considerable influence over the structural performance. Column spacing mostly dictates the largest span for a given footprint, and it can remove intermediate columns for arching geometries, while the edge start generally prescribes the boundary condition along the outside of the roof. Conceptually, the overall size variables should control structural performance, as larger structures and corresponding spans require more area. However, designers must be careful to not assume too much from this analysis, as there is clearly noise in the data, even as knowledge of the main relationships can be useful.

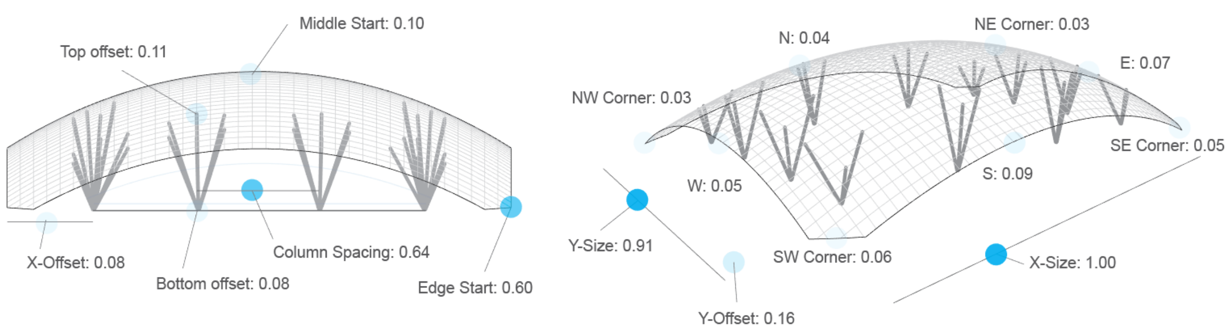

Figure 6: An initial estimation of variable importance to the problem by calculating variable effects

4.3 Variable transformation

After understanding variable importance, the designer might seek creative solutions within the design space. At this point, he or she could sample the design space at a resolution that fits the practical pressures on the design process. In this example, a Latin Hypercube sampling methodology (n = 1000) was used to provides the underlying dataset used for the following workflows. Beyond viewing these samples as a design catalog, there are other ways to use a dataset to pursue more natural, interactive, and flexible approaches to design exploration. One method is to mine the existing dataset for patterns and find new ways of manipulating geometry that connect more directly with performance. Figure 7 provides two example directions for morphing geometry that are meant to correlate with performance for both structure and energy, based on Canonical Correlation Analysis (CCA). These synthetic variables are found by conducting CCA, extracting the coefficients for each original variable, and then creating a slider that controls all original variables simultaneously through multiplication with these coefficients. In the structural direction, very large spans and a relatively flat roof give way to a much smaller, curved roof that is supported at the corners. Along this continuum, the structure transitions from acting primarily in bending to behaving primarily in compression, which is more efficient. For energy, a tall, high-surface area design transitions towards larger, lower surface area structures before finding a design that is too low to be feasible for the programmatic requirements at the edges of the design space.

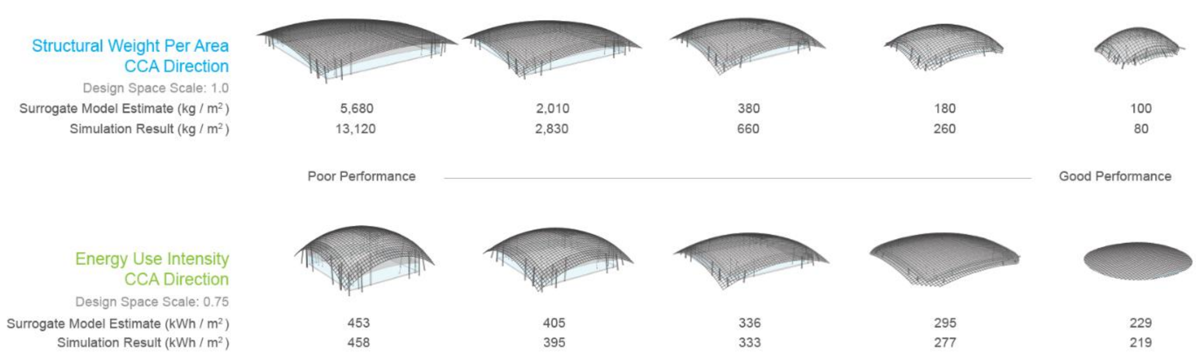

Figure 7: CCA variable directions based on SMQ and EUI for the design space

The creation of synthetic variables like those shown in Figure 6 has a few potential benefits. First, the designer might use these directions during live exploration, mostly likely in conjunction with the original sliders to provide more flexibility while moving from global to local. In addition, this slider essentially provides a composite visualization of which design variables matter, and how they should trend in order to improve performance. While this does not tell the whole story for complex objective functions, it can provide additional information and in some cases give greater control over the design. In future cases with more variables, or when variable generation might be automated from an initial sketch, such methods for interpreting the morphing design space could see further applications.

4.4 Filtering options

Next, or independently at another point in the process, a designer might want to brainstorm and find a diverse range of possibilities, before choosing and further refining a single option. Working again from a generated dataset that contains either original or synthetic variables, a designer could achieve greater diversity while considering fewer results by using a diversity filter. In this example, the designer first starts with the large dataset that was employed for creating synthetic variables. To obtain an initial starting point for further design, the designer first puts in a set of target numerical objectives. Rather than consider all designs in the dataset that meet this qualification simultaneously, which are often too numerous for a human to fully consider without systematic organization [84], the designer can use a diversity filter to return 12 sufficiently different designs from the set. The goal is to produce a wide range of potential options for further refinement, within a small enough group of designs to meaningfully consider each one. The outcomes of this section are thus not final solutions, but intermediate geometries meant to inspire creativity.

As an exploratory procedure, this paper considers progressively lower isoperformance levels for the design space to understand how much diversity is sacrificed by moving towards better structural performance within the dataset. Figure 8 shows the number of qualified designs, diversity of a random sample within each performance level, and the diversity of a specifically filtered set for structural performance targets ranging from 80-600 kg / m^2. The number of qualified designs shows a downward trend with lower structural material quantities—for example, within this dataset, fewer designs can be found at 100 kg / m^2 than at 200 or 300 kg / m^2.

However, no similarly clear trend exists for design diversity. To arrive at the diversity ranges in Figure 8, a random sample of 12 designs was first taken from within each qualified set 5 times, and the diversity of this culled set was measured repeatedly. In this paper, an average of the two sparseness methods and the outlier method was used as a unitless diversity metric for relative comparison. Next, the DSE diversity filter was used to find 5 sets of 12 qualified designs that have measurably higher diversity, using the same metrics. There is randomness in each procedure, hence the ranges and repeated sampling for research interest. Yet in each case, the diversity filter leads to more diverse designs than a random selection, which might be the default way of initiating a similar interactive workflow. Comparing across performance targets indicates that little diversity is sacrificed when moving towards 100-150 kg / m^2 from the poorer performing levels. Armed with this knowledge, a performance-conscious designer still in the creative brainstorming phase might begin by considering potential options at these lower ranges.

A visual comparison of example isoperformance sets from this exploratory analysis is provided in Figure 9. This image shows 12-design sets at three separate performance levels: 400, 200, and 150 kg / m^2, along with corresponding overall steel quantities. Following from the measured diversity, each of these sets provides noticeably dissimilar directions for further exploration and refinement. Moving down the levels, there are other trends, such as increasing curvature and eventually the appearance of infeasible solutions, which must be managed. While each of these levels satisfy the need for geometric diversity, it may be wisest to initiate further exploration at one of the lower performance levels, depending on the needs of a project.

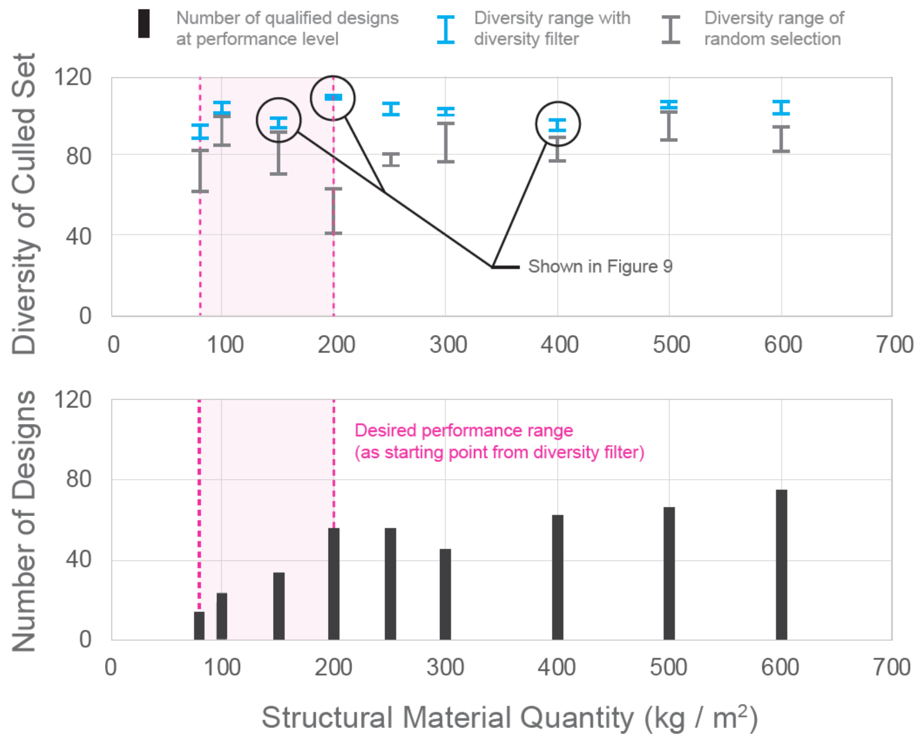

Figure 8: A comparison of number of qualified designs and diversity of diversity culled sets for brainstorming, at different structural performance targets

Regardless of which individual designs would be selected, it is clear from the large differences and unorthodox geometries that these designs represent non-standard solutions, which expands the possible options during a brainstorming phase, albeit dependent on the resolution of previous manual steps. Designers would have had trouble rapidly generating each of these options with only a chosen target threshold and an automated optimization process. When compared to a pure sampling technique, the diversity filter makes sure to eliminate designs that are too similar from consideration and allow designers to meaningfully engage with their preferred number of options.

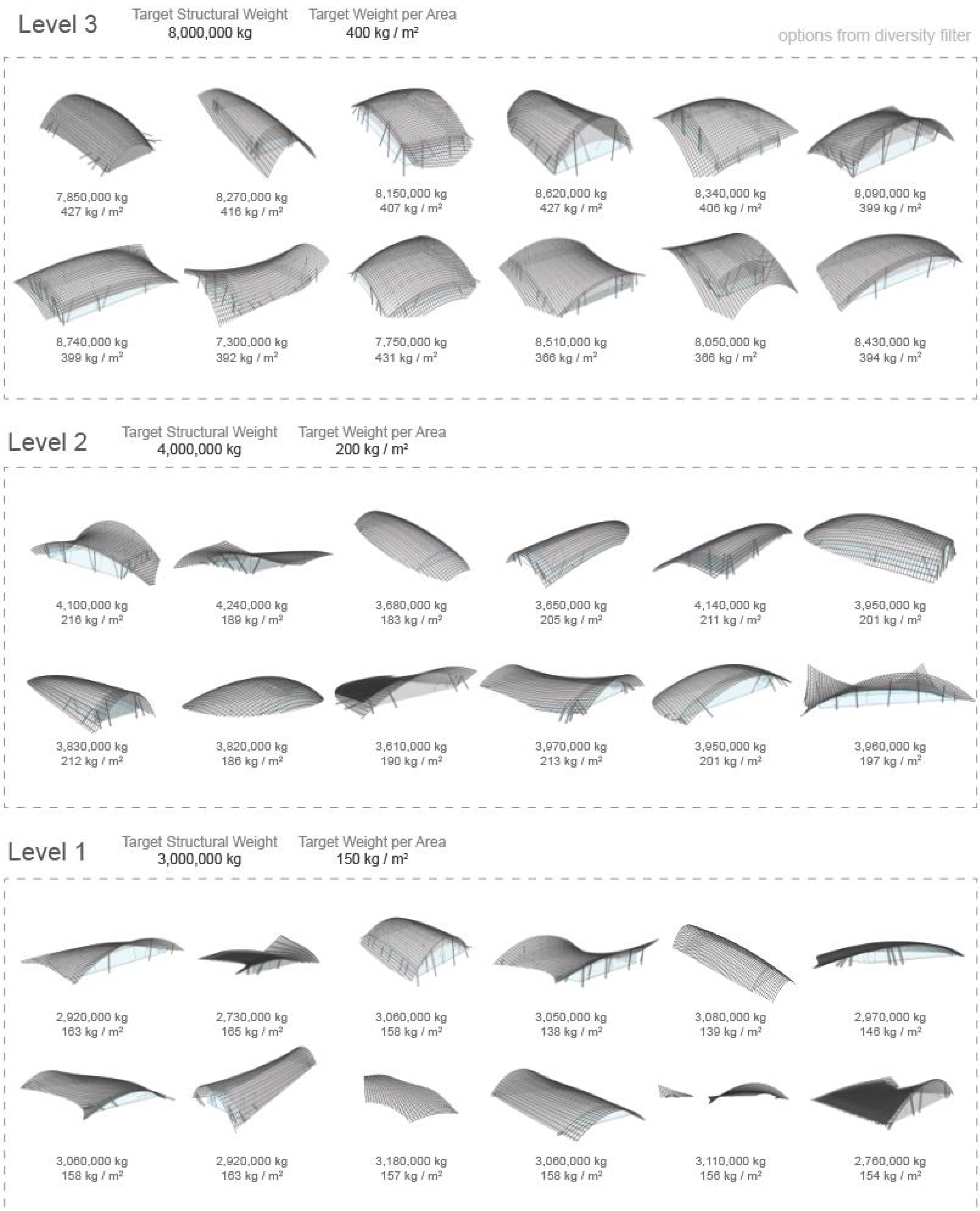

Figure 9: Example sets of 12 diverse design possibilities for three different structural performance levels. Moving towards Level 1 improves performance without meaningfully sacrificing too much diversity, but also leads to some geometrically unqualified designs.

4.5 Live feedback and interactive optimization

After formulating and analyzing a design space, or selecting a design from a brainstorming activity, a user would likely continue refining the design. With traditional parametric methods, it is only possible to adjust the original sliders. Furthermore, simulations must be completed at each iteration, or turned off for the sake of live geometric changes. Using the interactive optimization framework enabled in Workflow 8, designers can adjust the design while moving in the design space and objective space simultaneously. The first step is to use the previously generated dataset to train surrogate models that predict in real-time the estimated performance of the building.

For this example, both Random Forrest and Ensemble Neural Network surrogate models were attempted by splitting the dataset into training and validation data. After testing, a Random Forest with 100 trees and 0.6 training set ratio was most accurate for the structural surrogate model, and a Random Forest with 200 trees and 0.6 training set ratio was best for energy. The structural model used 12,000 initial simulations for training and validation, while the energy model used 1,000. Example plots of actual versus predicted performance for structural weight per area (500 test points) and Energy Use Intensity (288 test points) are provided in Figure 10. In this figure, the dotted lines represent 10% difference from the actual simulation values. The energy surrogate model is more accurate than the structural model overall. However, in the feasible structural range for the design problem, there is a strong relationship between predicted and actual data, which can still help support designers making live geometric decisions.

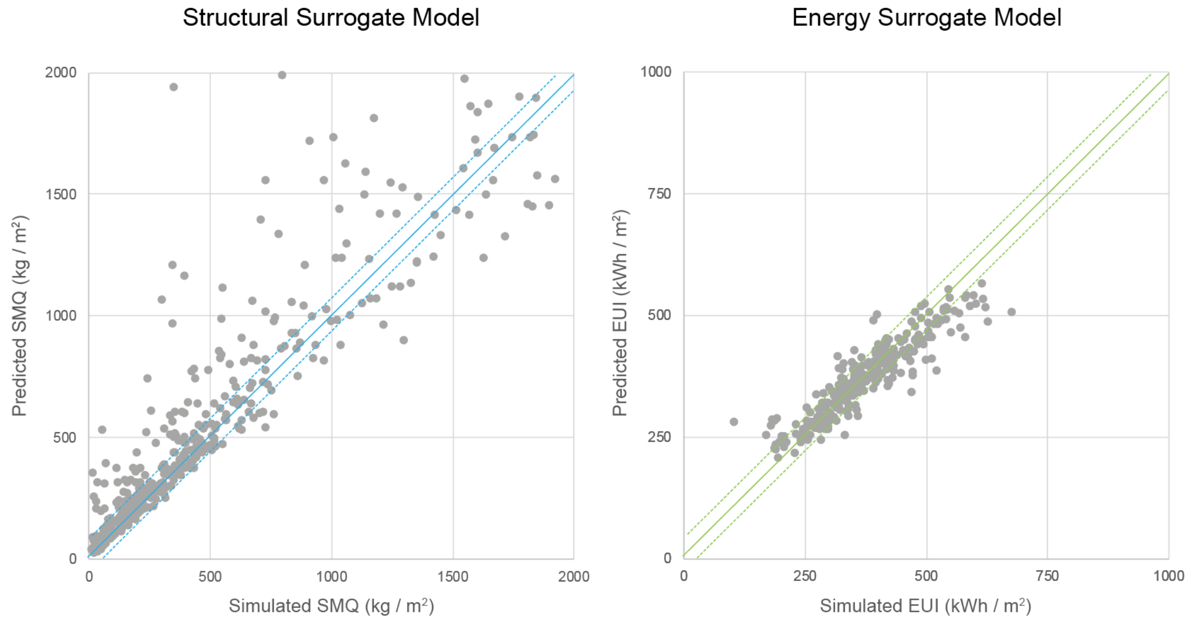

Figure 10: Visualization of the accuracy of surrogate models for structure and energy

The output of these surrogate models can then be projected onto the screen along with other geometric design information, as shown in Figure 11. As sliders are moved, the results of the surrogate models update in real-time, which the designer can use to further build intuition about the design space and understand if local movements improve or worsen performance. Various visualization techniques can provide the performance feedback—these example graphics use native Grasshopper components.

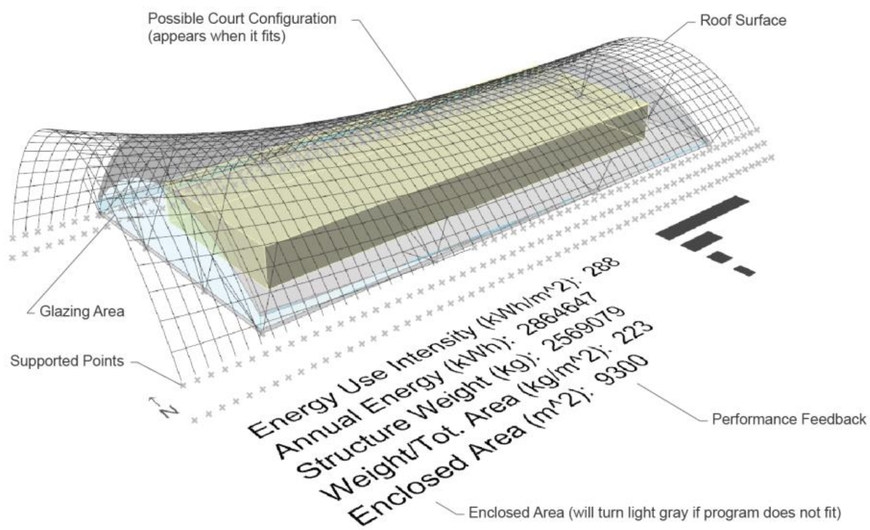

Figure 11: Live visualization of geometry and performance (according to the surrogate models) together

The design can also be manipulated interactively using gradient information from the objectives, which provides the user with specific directions to improve performance. The starting point for this local exploration can be a design already optimized for one objective, a general concept with room for improvement selected using the diversity filter, or any other preferred point in the design space. Using the Stepper component, which has a separate interface, designers are able to select an objective, pick a step size, and click to move in the direction of the gradient or its opposite. Figure 12 shows a history of how the objective functions have changed throughout the exploration. In this stage, the designer can meaningfully engage with all design objectives simultaneously, to understand their relationships while making subtle adjustments and ultimately refining the design.

In this design example, the changing objective functions tend to trend together, but various tradeoffs are often typical in early stage design. Yet even so, the designers might not want to keep improving a given objective until it flattens out, since following a single path through the design space could lead to better performing designs that deviate too far from the original design intent, fail to balance competing objectives, or violate obvious spatial constraints. In an automated optimization, such geometric constraints must be manually coded into the problem, which takes time and expertise to do properly. The threshold at which a design might deviate from the original intent is also ignored by a computer. By gradually optimizing step-by-step, these issues can be directly managed by the designer without any additional scripting.

Through this interface, it is also possible to attempt to move in isoperformance directions, which can be used to traverse between or depart certain areas of the design space or more generally for brainstorming support. In addition, users can select only certain variables to include in the gradient calculation and subsequent movement through the design space. These tools, when used sequentially or partially in parallel, enable a rich, data-driven, multi-objective approach to early computational design.

4.6 Comparison of designer types and lessons learned

Building designers vary in their needs, approaches, and desired outcomes. In some cases, data-driven methods are used to explore large geometric design differences for expressive, non-traditional applications, with an eye towards quantitative performance goals along with outside design criteria. In these settings, designers are open to a wide range of possibilities, and do not yet know which aspects of the design should remain constant as it is adjusted. For other situations, designers might be refining an already sound concept, while considering multiple objectives simultaneously, perhaps because secondary objectives were ignored during the initial ideation. In either case, there are potential benefits and challenges to using these interactive methods compared to other available parametric workflows.

Figure 12 presents two paths through the design and objective spaces for this case study, based on the hypothetical needs of two different design teams. The dotted lines represent the changes in objective functions through discrete steps taken in the objective space, and the start and end points represent actual simulations of these two designs, which give a more accurate evaluation of overall performance changes. These paths were generated using a combination of design space sliding and objective space stepping, which is enabled through DSE, allowing for both direct and indirect geometric manipulation.

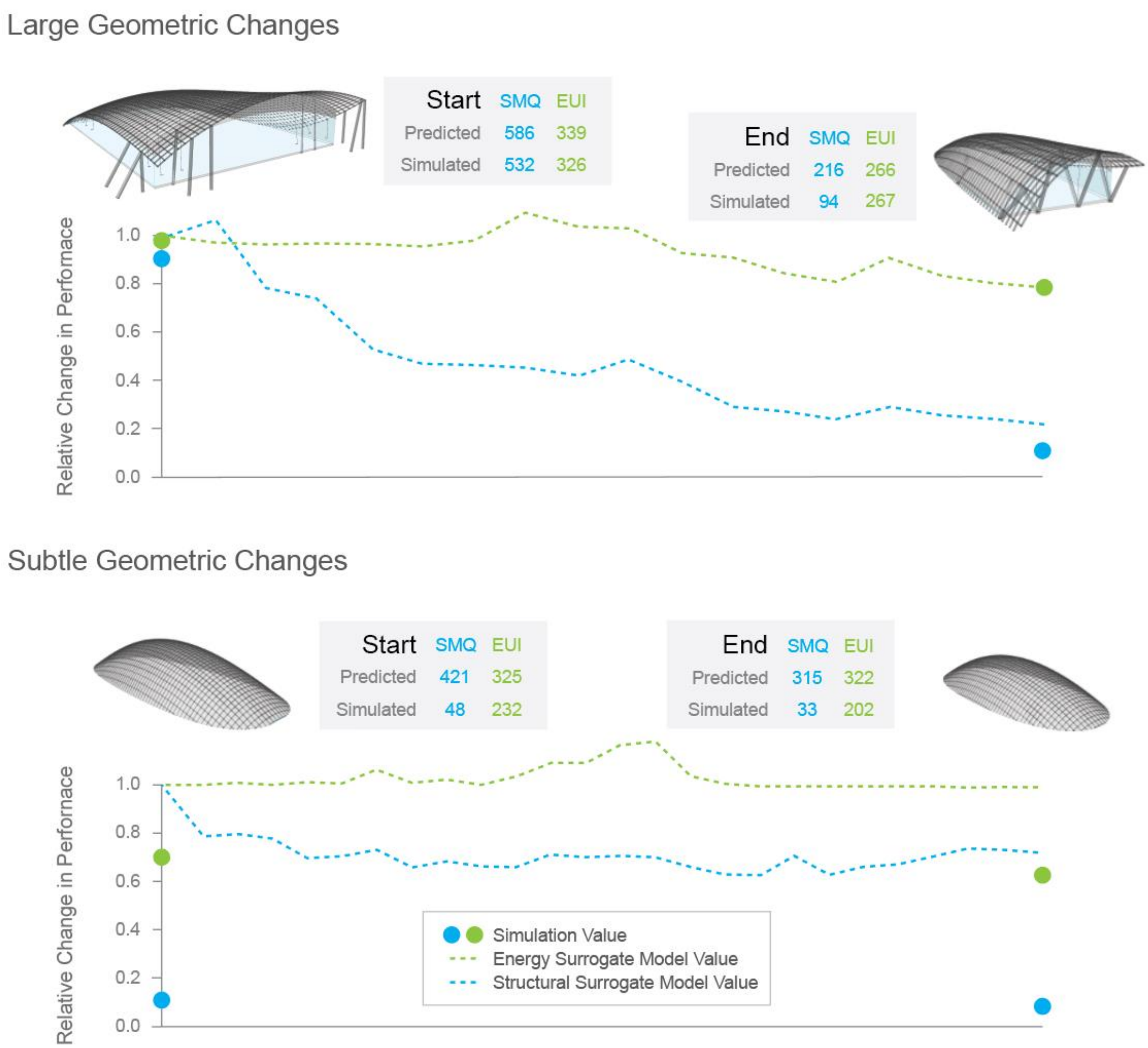

Figure 12: Possible paths through the design space using interactive optimization, for two different designer types

In the first example, the starting point is a form that seemed compelling from a visual perspective but does not perform well. By moving through the design space, the designer notices how SMQ and EUI can both be reduced, while still maintaining visual aspects of the original concept. While conducting this method, the designers can also consider qualitative feedback and even hard constraints, such as required programmatic area, without needing to code them directly for an automated solver. In this region of the design space, the surrogate model tends to be fairly accurate, and the result is a substantially better performing concept as measured by simulation.

In the second design space path, a structurally efficient concept has already been developed. The initial oval grid shell, which is symmetrical and supported along all edges, is a higher performing design than most other options in the space, weighing in at under 50 kg / m^2. This structural surrogate model, which maps relationships between variables and objectives across a much broader design space, severely overestimates the amount of steel required to build it compared to the simulation. The energy surrogate model provides a similar overestimate. These inaccuracies make the stepping itself more difficult—attempting to improve performance in what is clearly near a local minimum leads to more back and forth. In the end, the simulated performance turns out better for both objectives after moving through the design space, but not with the change in magnitude for the freer exercise above.

Although the paths show improved performance, this exercise raises some issues. It is obvious from the second example that surrogate model accuracy matters. The goals of this case study necessitated an extremely broad design space, which made it difficult for a surrogate model to properly capture structural performance at the extremes. Such methods must be properly tuned to the resolution of the question at hand, which was not the case for the second design path. Even for problems in which surrogate models are generally more accurate, the risks propagated by their uncertainty never go away. In other words, this approach does not always lead to an efficient form, due to either user preference, user error, or relationships that are simply too difficult to model.

This is also true of using complicated performance models at all in the early design stages. Especially for structural and energy modeling, which many practitioners and researchers now do parametrically, complex design spaces require care to ensure meaningful results. In a context where subtle design decisions are being made and one or two performance metrics dominate the conversation, it may make more sense to employ a catalog or run a long optimization instead of using live surrogate modeling. The choice of proper tool or method can also be discipline specific, and a general multi-objective data framework is not always the answer.

Yet the utility of these methods for supplementing creative brainstorming processes, understanding and interpreting the design space, and making important design decisions about overall geometry is clearly demonstrated by this design example.

5 Discussion

5.1 Design example performance compared to benchmarks

While the main goals of this paper relate to demonstrating, justifying, and applying the exploration of relative options within a (possibly dynamic) design space, it is worthwhile to compare the performance of the options mentioned here to databases of actual buildings. This comparison illuminates the magnitudes at which it is possible and practical to improve performance in the domains of structure and energy, which has implications for which performance objectives should be considered in early parametric design and how they might be prioritized. It also helps contextualize and even quantify the potential benefits of a toolkit for considering geometric and technical goals simultaneously, which was not possible for many existing buildings. Finally, it clarifies ways in which performance-conscious designers might apply these tools to supplement their workflows with a variety of intentions—finding optimal shapes, hitting generally accepted performance targets, or making minor improvements to an already preferred shape.

First, the range of structural options in this paper is compared to buildings in the DeQo database [85], which contains an extensive record of structural material quantities for built projects around the world [86,87]. Figure 13 groups examples in this case study with other buildings based on construction material, number of stories, and longest clear span. The meaningful range of the case study contains an upper bound of around 600 kg/m^2, with the knowledge that general concepts at this performance level could be substantially improved through interactive optimization. The lower bound considered is around 100 kg/m^2, and a few carefully chosen structures in the design example are even lower. While such high-performance designs are possible in the design space, it must be stressed that the simulations are only estimates and are most useful for relative importance.

Nevertheless, the results of this example indicate that such computational exploration can meaningfully move designs towards high performance. This result must be understood cautiously, since a single-story roof that can behave primarily in tension or compression should be lighter per usable square meter than a tall tower, and the DeQo database only allows for basic segmentations. In the design context of shells or other lightweight geometries, 100 kg/m^2 is a more reasonable target for a long span structure, compared to the variety of buildings in DeQo. Capable structural designers around the world have pushed even lower on SMQ for high-performance roofs, which is desirable for economic and environmental reasons.

Figure 13 supports another major conclusion: for structural designs, even with a given footprint and parametric model, it is easy to find options that are 10x, 20x, or even 30x worse in terms of material quantities. Since stakes are quite high for designers in the structural domain, it is important that these data-driven tools be used by or in conjunction with experienced designers, who can identify potentially good and bad designs based on an understanding of structural behavior. In these situations, data-driven techniques can help discover potentially new forms that perform similarly to well-known shapes, locally optimize geometry, or consider structural decisions with the benefit of multidisciplinary performance feedback.

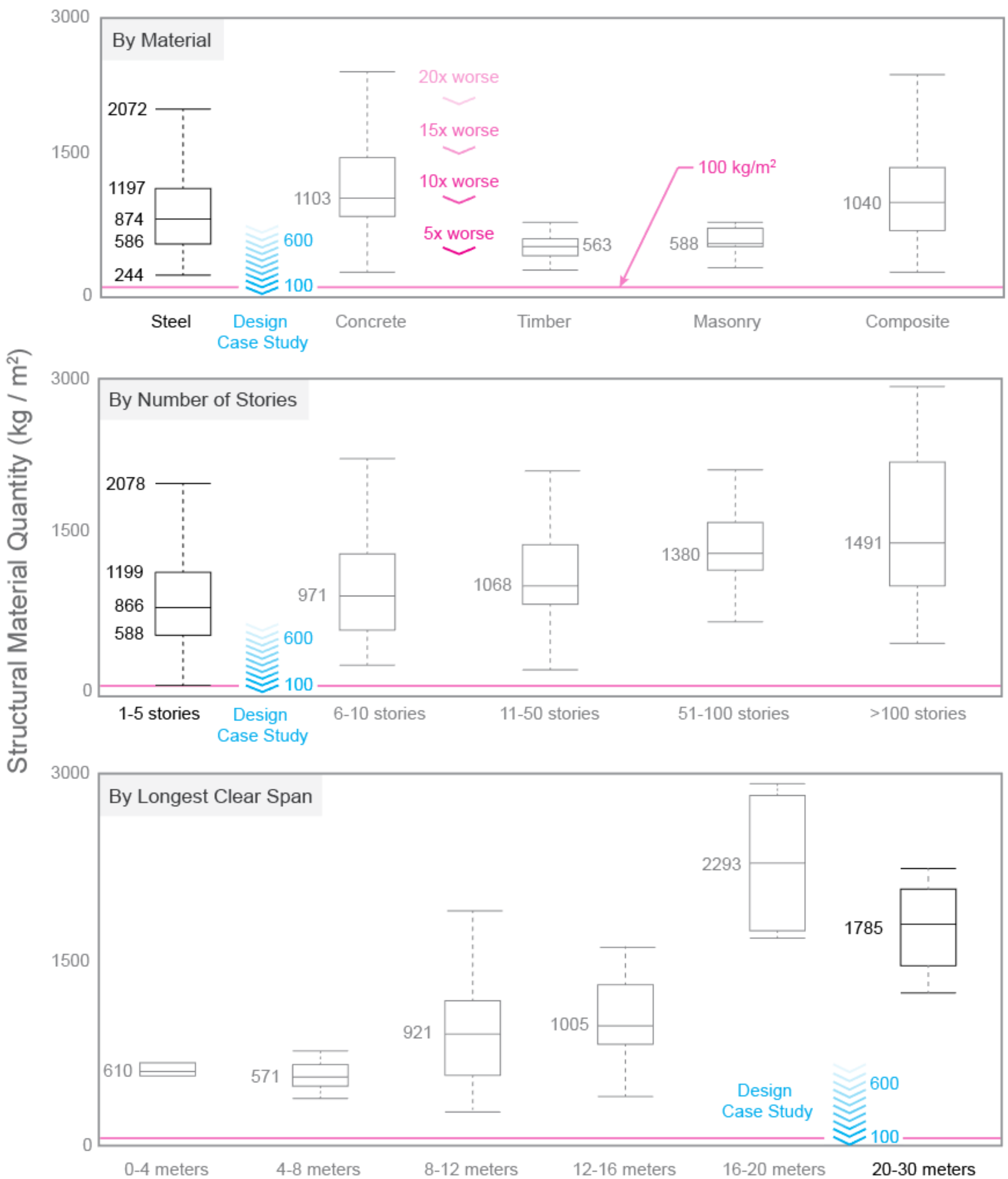

Figure 13: The range of performance improvement generally considered for structural weight per area in this case study, compared to similar buildings from the DeQo database, after De Wolf (2016)

Two similar comparisons for the energy model reveal that geometric design decisions have a smaller performance multiplier in this domain than for structure. The first comparison is to similar buildings found in the Building Performance Database (BPD) [88], which was created by Lawrence Berkeley National Laboratory and is the largest dataset of information about the energy-related characteristics of commercial and residential buildings in the United States [89]. Figure 14 includes all buildings of comparable use in the BPD constructed after 2000, in the same climate zone (5) as Boston. Although there is no specific category for an athletic center, the selection included buildings categorized as education, healthcare, lodging, nursing home, office, public assembly, retail, service, and transformation, while avoiding energy outliers such as convenient stores, data centers, grocery stores, laboratories, and parking garages.

Based on the interactive design space paths and sampled bounds, it is generally possible to move from slightly over the median building to substantially better than the 25^{th} percentile, by making geometric changes using data-driven tools in DSE. As with structure, being considerably better than the median does not in itself indicate a high-performance building. Depending on use, climate, and other factors, contemporary practicing designers routinely set more ambitious source EUI targets to reduce the environmental impacts of their buildings. Often, these targets are a direct response to the Architecture 2030 Challenge, or a similar push for better performance in buildings. For example, the Zero Tool from Architecture 2030 indicates a baseline source EUI of 305 kWh / m^2 for a new construction fitness center in Boston with a similar square footage [90]. A 70% reduction would set a target of 92 kWh /m^2, and an 80% reduction would leave a target of 61.

The geometric changes explored in this example alone cannot reach these targets. Nevertheless, it does appear that early stage energy modeling during geometric exploration has some noticeable benefit, in that it allows designers to move towards geometries that are relatively more efficient and would likely lead to lower energy uses as the design is further refined.

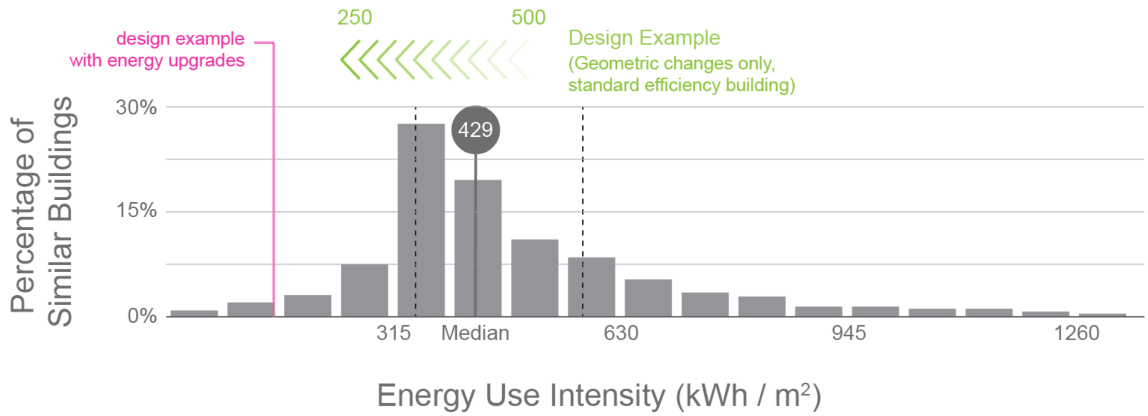

Figure 14: The range of performance improvement generally considered for EUI in this design example, using only geometric variables, compared to similar buildings from Lawrence Berkeley National Lab's Building Performance Database

The second comparison considers the final design obtained from interactive stepping as it was modeled, versus how much the EUI could be reduced by upgrading non-geometric settings for energy performance. The case study was conducted with a code-compliant building rather than one with many energy-efficient features, such that users could more easily perceive differences in energy performance during geometric exploration. To understand the consequences of magnifying geometric differences with a mid-efficiency model, another test model was created with better insulation and glazing, more efficient lighting, and less substantial requirements for showers and equipment power density. Figure 15 shows consecutive upgrades in these domains, according to the values in Table 1. Overall, the energy efficiency features drop the baseline EUI of the selected design to ~150 kWh/m^2**,** or around 55% of what was originally shown in the model. These design decisions have a substantial influence over the performance of the building and may be worthwhile to contemplate using data-driven approaches. However, they can be made independently of geometric exploration, and they often involve discrete options with associated costs, which is a different conversation than an exploration of form.

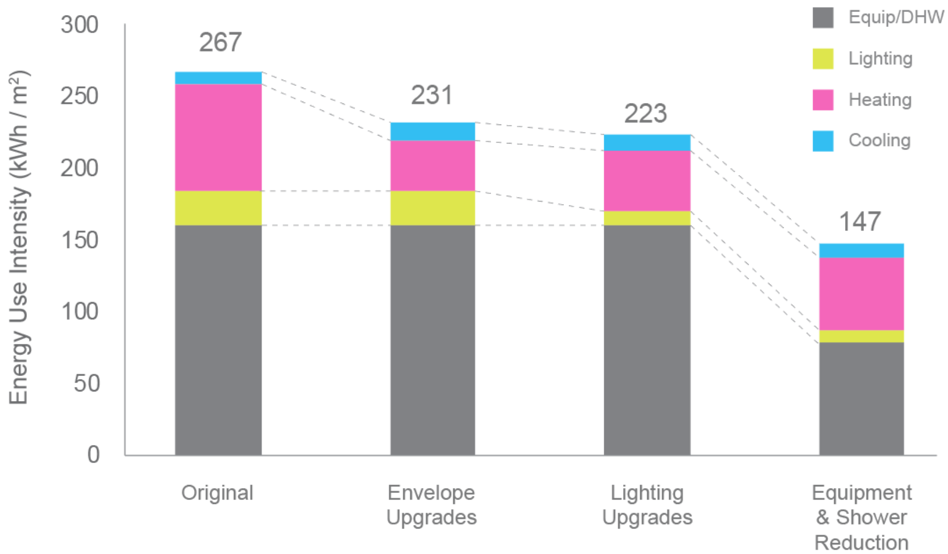

Figure 15: History of energy upgrades for the design example, which are here explored independent of geometry

5.2 Prioritization in multi-objective design

These comparisons to performance databases and an alternative energy modeling approach puts the improvements enabled by data-driven design of overall building form in context. If designers did not have access to such tools and intended to "optimize" within an established geometry, or adjusted geometry without simulation feedback, they would miss out on potential savings in material usage, energy consumption, and other performance considerations that interact with geometry. A comprehensive design study by Brown [79] quantifies differences in the simulated performance outcomes of designers with access to performance-free environments, versus those with access to several workflows described in this paper. However, the main impact of the toolkit described here is its demonstrated flexibility and accessibility, which makes it more likely that designers can reap the benefits shown by providing context to the case study.

At the same time, these comparisons also point to certain quantitative relationships that should be managed during conceptual design. For example, while structural material usage changes more substantially with geometry, annual energy over the lifetime of a building has a larger lifecycle effect than the initial structural embodied energy for typical buildings, although this relationship is changing with the movement towards net-zero operation. Yet energy models are not always as effective "form-generators" as models for daylight, structure, and other objectives, depending on the design variables in question and the resolution of the model. In some cases, a design team might be better off with a simplified shoebox model to set all energy parameters, and then explore form with those settings already established, along with the knowledge that subtle geometric differences will not greatly affect overall EUI. Prioritization between objective functions may also change for different projects. One may want to increase roof surface area for PV at the expense of additional heat transfer, or the embodied energy in the structural material may be less concerning for wood structures versus concrete and steel. For this reason, data-driven design tools that leave decisions about specific workflow to the user have advantages.

Simulation-based computational design tools are thus meant to enhance or supplement designers' abilities rather than replace them. These tools can improve the performance of standard designs or push high-performance designs to use even less embodied and operational energy, while also illuminating how these objectives might interact with a long list of related goals. The design space might need to be adjusted and refined to lead to good solutions, which is often a human task supported by computational feedback. This is true of most early stage modeling platforms or comparable design tools, even as they begin to employ techniques like data science and machine learning—most interactive systems cannot yet complete all work alone, from problem formulation through final selection. In the search for non-standard forms and creative freedom, designers still bring their own experiences and sensibilities to the process, often in a more direct, tangible way than they would to automated optimization.

6 Conclusions

6.1 Contributions towards flexibility & accessibility in data-driven parametric design

This paper describes several contributions towards data-driven, multi-objective, parametric design. Primarily, it proposes and justifies a toolbox approach, called Design Space Exploration, by explaining natural relationships between data science and optimization components that enable new data-driven strategies beyond the typical creation of design catalogs. These approaches are demonstrated on a case study involving complex geometry, considerable design freedom, and a mixture of qualitative and quantitative design goals. Advantages of the toolbox approach include better workflow flexibility and the possibility of customizing how design data is used without needing to manipulate text-based code. Novel functionality within the DSE toolkit can also increase creative possibilities an enable more meaningful and directed consideration of the early design space. By enabling the workflows described throughout, these freely available tools provide the basis for further implementation of data-driven methods within both research and building design practice, which increases their potential for broader impact.

6.2 Future work and concluding remarks

Since the toolbox is in constant development, there are clear areas for future work. The tools can gradually be improved in terms of user interfaces, robustness, and additional functionality. Another topic of research and software development would include connecting these methods directly to a generalized data visualization platform. Although used to clarify ideas, data visualization techniques themselves are largely out of the scope of this paper. There are existing tools for design space visualization as mentioned in the literature review [44,47,48,91], and some have been connected to Grasshopper or related interfaces in compelling ways. While there are opportunities to develop new methods for effective data visualization related to computational design, this part of the process was separated from the components that control and operate on the information found throughout the design space, which was the focus of this paper. It is also worth mentioning that the ideas around data-driven design proposed here are platform agnostic and not married to specific software. As such, future work could extend this functionality to the programs that designers adopt in the upcoming years. More broadly, ongoing efforts to generalize surrogate modeling techniques beyond individual parametric models, extend interfaces to facilitate multi-user interaction, and further adapt optimization for creative applications could be implemented as part of a toolbox. Each of these areas for future work would build on a conceptual framework for data-driven, multi-objective design.

7 Acknowledgements

The authors would like to thank those who contributed to software development of the design tools described in this paper, including Renaud Danhaive, Jonathas Felipe, Stavros Tseranidis, and Alicia Nimrick.

8 References

[1] L.G. Caldas, L.K. Norford, A design optimization tool based on a genetic algorithm, Automation in Construction. 11 (2002) 173–184. doi:10.1016/S0926-5805(00)00096-0.

[2] K. Shea, R. Aish, M. Gourtovaia, Towards integrated performance-driven generative design tools, Automation in Construction. 14 (2005) 253–264. doi:10.1016/j.autcon.2004.07.002.

[3] M. Turrin, P. Von Buelow, R. Stouffs, Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms, Advanced Engineering Informatics. 25 (2011) 656–675. doi:10.1016/j.aei.2011.07.009.

[4] T. Wortmann, Surveying design spaces with performance maps: a multivariate visualization method for parametric design and architectural design optimization, International Journal of Architectural Computing. 15 (2017) 38–53. doi:10.1177/1478077117691600.

[5] M. Tsigkari, A. Chronis, S. Joyce, A. Davis, S. Feng, F. Aish, Integrated design in the simulation process, in: L. O'Brien, B. Gunay, A. Khan (Eds.), Proceedings of the Symposium on Simulation for Architecture & Urban Design, San Diego, 2013: pp. 153–162. http://simaud.org/proceedings/index.php#2013.

[6] E. Bradner, F. Iorio, M. Davis, Parameters tell the design story: ideation and abstraction in design optimization, in: D. Gerber, R. Goldstein (Eds.), Proceedings of the Symposium on Simulation for Architecture and Urban Design, 2014: pp. 77–84. http://simaud.org/proceedings/index.php#2014.

[7] J. Lee, N. Gu, A.P. Williams, Parametric design strategies for the generation of creative designs, International Journal of Architectural Computing. 12 (2014) 263–282. doi:10.1260/1478-0771.12.3.263.

[8] J. Harding, S. Jorce, P. Shepherd, C. Williams, Thinking topologically at early stage parametric design, in: Advances in Architectural Geometry 2012, Springer, Vienna, 2012: pp. 67–76. https://www.researchgate.net/profile/Sam_Joyce2/publication/269634846_Thinking_Topologically_at_Early_Stage_Parametric_Design/links/5592934908aed7453d462776.pdf.

[9] J. Nembrini, S. Samberger, G. Labelle, Parametric scripting for early design performance simulation, Energy and Buildings. 68 (2014) 786–798. doi:10.1016/j.enbuild.2013.09.044.

[10] J. Haymaker, M. Bernal, M.T. Marshall, V. Okhoya, A. Szilasi, R. Rezaee, C. Chen, A. Salveson, J. Brechtel, L. Deckinga, H. Hasan, P. Ewing, B. Welle, Design space construction: A framework to support collaborative, parametric decision making, Journal of Information Technology in Construction. 23 (2018) 157–178. issn: 18744753.

[11] D.J. Gerber, S.H.E. Lin, Designing in complexity: simulation, integration, and multidisciplinary design optimization for architecture, Simulation. 90 (2014) 936–959. doi:10.1177/0037549713482027.

[12] N.C. Brown, C.T. Mueller, Designing with data: moving beyond the design space catalog, in: Proceedings of ACADIA 2017: Disciplines and Disruption, Boston, MA, 2017. isbn: 978-0-692-96506-1.

[13] F. Ritter, P. Geyer, A. Borrmann, The Design Space Exploration Assistance Method: constraints and objectives, in: Proc. of the 13th Conference on Construction and Virtual Reality (CONVR), London, 2013: pp. 72–81. http://itc.scix.net/pdfs/convr-2013-8.pdf.

[14] P. von Buelow, Paragen: performative exploration of generative systems, Journal of the

International Association For Shell and Spatial Structures. 53 (2012) 271–284. https://www.ingentaconnect.com/content/iass/jiass/2012/00000053/00000004/art00010.

[15] C.T. Mueller, J.A. Ochsendorf, Combining structural performance and designer preferences in evolutionary design space exploration, Automation in Construction. 52 (2015) 70–82. doi:10.1016/j.autcon.2015.02.011.

[16] M.R. Asl, M. Bergin, A. Menter, W. Yan, BIM-based parametric building energy performance multi- objective optimization, in: E. Thompson (Ed.), Proceedings of the 32nd ECAADe Conference, Newcastle upon Tyne, England, 2014: pp. 455–464. http://papers.cumincad.org/cgi-bin/works/paper/ecaade2014_224.

[17] T. Méndez Echenagucia, Computational search in architectural design, Ph.D. Thesis, Politecnico di Torino, 2014. doi:10.6092/polito/porto/2543137.

[18] R. Evins, A review of computational optimisation methods applied to sustainable building design, Renewable and Sustainable Energy Reviews. 22 (2013) 230–245. doi:10.1016/j.rser.2013.02.004.

[19] S. Asadi, Z.W. Geem, Sustainable building design: a review on recent meta-heuristic methods, in: X.-S. Yang (Ed.), Recent Advances in Swarm Intelligence and Evolutionary Computation. Studies in Computational Intelligence, Springer, 2015: pp. 203–223. doi:10.1007/978-3-319-13826-8_11.

[20] C. Waibel, Simulation-based optimization of buildings and multi-energy systems, Dr. Sc. Thesis, ETH Zurich, 2018. doi:10.3929/ethz-b- 000307674.

[21] X. Shi, Z. Tian, W. Chen, B. Si, X. Jin, A review on building energy efficient design optimization from the perspective of architects, Renewable and Sustainable Energy Reviews. 65 (2016) 872–884. doi:10.1016/j.rser.2016.07.050.

[22] H. Hofmeyer, J.M. Davila Delgado, Coevolutionary and genetic algorithm based building spatial and structural design, in: Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, Cambridge University Press, 2015: pp. 351–370. doi:10.1017/S0890060415000384.

[23] B. Steiner, E. Mousavian, F.M. Saradj, M. Wimmer, P. Musialski, Integrated structural-architectural design for interactive planning, Computer Graphics Forum. 36 (2017) 80–94. doi:10.1111/cgf.12996.

[24] T. Wortmann, A. Costa, G. Nannicini, T. Schroepfer, Advantages of surrogate models for architectural design optimization, Artificial Intelligence for Engineering Design, Analysis and Manufacturing. 29 (2015) 471–481. doi:10.1017/S0890060415000451.

[25] R. Danhaive, C. Mueller, Structural metamodelling of shells, in: Proceedings of the IASS Symposium 2018, Boston, MA, 2018. https://www.ingentaconnect.com/content/iass/piass/2018/00002018/00000025/art00002.

[26] S. Tseranidis, N.C. Brown, C.T. Mueller, Data-driven approximation algorithms for rapid performance evaluation and optimization of civil structures, Automation in Construction. (2016). doi:10.1016/j.autcon.2016.02.002.

[27] T. Wortmann, Efficient, visual, and interactive architectural design optimization with model-based methods, Ph.D. Thesis, Singapore University of Technology and Design, 2018. doi:10.13140/RG.2.2.15380.55685.

[28] D. Reynolds, K. Ghantous, R.K. Otani, Performance measures from architectural massing using machine learning, in: Proceedings of the International Association for Shell and

Spatial Structures (IASS) Symposium 2015, Future Visions, Amsterdam, 2015. https://www.ingentaconnect.com/content/iass/piass/2015/00002015/00000016/art00009.

[29] P. Westermann, R. Evins, Surrogate modelling for sustainable building design – A review, Energy and Buildings. 198 (2019) 170–186. doi:10.1016/j.enbuild.2019.05.057.

[30] M. Burry, Scripting Cultures: Architectural Design and Programming, John Wiley & Sons, 2011. isbn: 978-0-470-74641-7.

[31] D. Davis, B. Peters, Design Ecosystems: Customising the Architectural Design Environment with Software Plug-ins, Architectural Design. 83 (2013) 124–131. doi:10.1002/ad.1567.

[32] R. Oxman, Performance-based design: current practices and research issues, International Journal of Architectural Computing. 06 (2008) 1–17. doi:10.1260/147807708784640090.

[33] R. Woodbury, Elements of parametric design, Routledge, London, 2010. isbn: 9780415779876.

[34] G. Celani, C. Eduardo Verzola Vaz, CAD scripting and visual programming languages for implementing computational design concepts: A comparison from a pedagogical point of view, International Journal of Architectural Computing. 10 (2012) 121–137. doi:10.1260/1478-0771.10.1.121.

[35] Robert McNeel & Associates, Grasshopper, (2007). www.grasshopper3d.com.

[36] Proving Ground, Lunchbox, (2011). provingground.io/tools/lunchbox/.

[37] Thornton Tomasetti, TT Toolbox, (2017). www.food4rhino.com/app/tt-toolbox.

[38] L. Greco, Dodo, (2015). www.food4rhino.com/app/dodo.

[39] R.A. Danhaive, C.T. Mueller, Combining parametric modeling and interactive optimization for high-performance and creative structural design, in: Proceedings of the IASS Annual Symposium 2015, Amsterdam, 2015. https://www.ingentaconnect.com/content/iass/piass/2015/00002015/00000020/art00006.

[40] J. Harding, C.B. Olsen, Biomorpher, (2017). www.food4rhino.com/app/biomorpher.

[41] S. Flöry, Goat, (2013). www.rechenraum.com/goat/download.html.

[42] J. Cichocka, A. Migalska, W.N. Browne, E.R. Ramirez, SILVEREYE - the implementation of Particle Swarm Optimization algorithm in a design optimization tool, in: G. Çağdaş, M. Özkar, L. Gül, E. Gürer (Eds.), Computer-Aided Architectural Design. Future Trajectories. CAADFutures 2017. Communications in Computer and Information Science, Springer Science & Business Media, Singapore, 2017. doi:10.1007/978-981-10-5197-5_9.

[43] T. Wortmann, Opossum: introducing and evaluating a model-based optimization tool for grasshopper, in: P. Janssen, P. Loh, A. Raonic, M.A. Schnabel (Eds.), Protocols, Flows and Glitches, Proceedings of the 22nd International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Hong Kong, 2017: pp. 283–292. http://papers.cumincad.org/cgi-bin/works/Show?caadria2017_124.

[44] University of Applied Arts Vienna, Bollinger+Grohmann Engineers, Octopus 0.4, (2018). www.food4rhino.com/app/octopus.

[45] A. Turiello, Generator, (2013). www.food4rhino.com/app/generation. (accessed February 10, 2019).

[46] Genometri Ltd., Genoform, (2013). https://www.food4rhino.com/app/genoform. (accessed

February 10, 2019).

[47] Proving Ground, Conduit, (2018). https://www.food4rhino.com/app/conduit. (accessed February 10, 2019).

[48] Thornton Tomasetti, Design Explorer, (2016). core.thorntontomasetti.com/design-explorer/. (accessed February 10, 2019).

[49] M. Zwierzycki, Owl, (2018). https://www.food4rhino.com/app/owl. (accessed February 10, 2019).

[50] B. Felbrich, Crow, (2016). https://www.food4rhino.com/app/crow-artificial-neural-networks. (accessed February 10, 2019).

[51] M. Rahmani, Optimo, (2014). dynamobim.org/optimo/. (accessed February 10, 2019).

[52] Autodesk, Dynamo, (2011). dynamobim.org. (accessed February 10, 2019).

[53] Autodesk, Project Refinery, (2018). www.autodesk.com/solutions/refinery-beta. (accessed February 10, 2019).

[54] S. Nagpal, C. Mueller, A. Aijazi, C.F. Reinhart, A methodology for auto-calibrating urban building energy models using surrogate modeling techniques, Journal of Building Performance Simulation. 12 (2019) 1–16. doi:10.1080/19401493.2018.1457722.

[55] N.C. Brown, C.T. Mueller, Design for structural and energy performance of long span buildings using geometric multi-objective optimization, Energy and Buildings. 127 (2016) 748–761. doi:10.1016/j.enbuild.2016.05.090.

[56] N.K. Khouri, Structural grid shell design with islamic pattern topologies, S.M. Thesis, Massachusetts Institute of Technology, 2017. http://hdl.handle.net/1721.1/111282.

[57] C.T. Mueller, Computational exploration of the structural design space, Ph.D. Thesis, Massachusetts Institute of Technology, 2014. http://hdl.handle.net/1721.1/91293.

[58] D.B. Crawley, L.K. Lawrie, F.C. Winkelmann, W.F. Buhl, Y.J. Huang, C.O. Pedersen, R.K. Strand, R.J. Liesen, D.E. Fisher, M.J. Witte, J. Glazer, EnergyPlus: Creating a new-generation building energy simulation program, Energy and Buildings. 33 (2001) 319–331. doi:10.1016/S0378-7788(00)00114-6.

[59] C. Preisinger, Karamba, (2014). www.karamba3d.com. (accessed February 10, 2019).

[60] Solemma, Diva-for-Rhino, (2012). solemma.net/Diva.html. (accessed February 10, 2019).

[61] M. Roudsari, C. Mackey, Ladybug Tools, (2013). www.ladybug.tools. (accessed February 10, 2019).

[62] S. Boonstra, K. van der Blom, H. Hofmeyer, M.T.M. Emmerich, J. van Schijndel, P. de Wilde, Toolbox for super-structured and super-structure free multi-disciplinary building spatial design optimisation, Advanced Engineering Informatics. 36 (2018) 86–100. doi:10.1016/j.aei.2018.01.003.

[63] C.R. Souza, The Accord.NET Framework, (2014). accord-framework.net/index.html. (accessed February 10, 2019).

[64] C. Ruegg, J. Larsson, J. Van Gael, Math.NET, (2009). www.mathdotnet.com. (accessed February 10, 2019).

[65] J.J. Durillo, A.J. Nebro, jMetal: a Java framework for multi-objective optimization, Advances in Engineering Software. 42 (2011) 760–771. doi:10.1016/j.advengsoft.2011.05.014.

[66] N.C. Brown, C.T. Mueller, Quantifying diversity in parametric design: a comparison of possible metrics, Artificial Intelligence for Engineering Design, Analysis and Manufacturing. 33 (2019) 40–53. doi:10.1017/S0890060418000033.

[67] N.C. Brown, C.T. Mueller, Design variable analysis and generation for performance-based parametric modeling in architecture, International Journal of Architectural Computing. 17 (2019) 36–52. doi:10.1177/1478077118799491.

[68] N.C. Brown, C.T. Mueller, Automated performance-based design space simplification for parametric structural design, in: Proceedings of the IASS Annual Symposium 2017, Hamburg, Germany, 2017. https://www.ingentaconnect.com/content/iass/piass/2017/00002017/00000015/art00008.

[69] N.C. Brown, C.T. Mueller, Gradient-based guidance for controlling performance in early design exploration, in: Proceedings of the IASS Annual Symposium 2018, Cambridge, MA, 2018. https://www.ingentaconnect.com/content/iass/piass/2018/00002018/00000002/art00018.

[70] J.S. Gero, Design prototypes: A knowledge representation schema for design, AI Magazine. 11 (1990) 26–26. doi:10.1609/AIMAG.V11I4.854.

[71] R. Yu, N. Gu, M. Ostwald, Comparing designers' problem-solving behavior in a parametric design environment and a geometric modeling environment, Buildings. 3 (2013) 621–638. doi:10.3390/buildings3030621.

[72] R. Yu, J. Gero, N. Gu, Architects' cognitive behaviour in parametric design, International Journal of Architectural Computing. 13 (2015) 83–101. doi:10.1260/1478-0771.13.1.83.

[73] R. Oxman, Theory and design in the first digital age, Design Studies. 27 (2006) 229–265. doi:10.1016/j.destud.2005.11.002.

[74] P. Geyer, K. Beucke, An integrative approach for using multidisciplinary design optimization in AEC, in: W. Tizani (Ed.), Proceedings of the International Conference on Computing in Civil and Building Engineering, 2010. pdfs.semanticscholar.org/a2d3/fe6b1669b6ad1bb6553ac66b6c1b747e043b.pdf (accessed January 2, 2020).

[75] P. Geyer, Multidisciplinary grammars supporting design optimization of buildings, Research in Engineering Design. 18 (2008) 197–216. doi:10.1007/s00163-007-0038-6.

[76] F. Flager, B. Welle, P. Bansal, G. Soremekun, J. Haymaker, Multidisciplinary process integration and design optimization of a classroom building, Electronic Journal of Information Technology in Construction. 14 (2009) 595–612. issn: 1874-4753.

[77] M. Krem, S.T. Hoque, S.R. Arwade, S.F. Breña, M. Asce, Structural Configuration and Building Energy Performance, Journal of Architectural Engineering. 19 (2013) 29–40. doi:10.1061/(ASCE)AE.1943-5568.0000103.

[78] C.P. Quaglia, N. Yu, A.P. Thrall, S. Paolucci, Balancing energy efficiency and structural performance through multi-objective shape optimization : Case study of a rapidly deployable origami-inspired shelter, Energy and Buildings. 82 (2014) 733–745. doi:10.1016/j.enbuild.2014.07.063.

[79] N. Brown, Design performance and designer preference in an interactive, data-driven conceptual building design scenario, Design Studies. (n.d.). (accepted 24 Jan 2020).

[80] J.-P. Dalbera, Photo of the Neue Nationalgalerie, (2013). commons.wikimedia.org/wiki/File:La_nouvelle_galerie_nationale_(Berlin)_(11478130635)

.jpg.

[81] N. Janberg, Photo of the SICLI Company Building, (2015). structurae.net/photos/242468-sicli-company-building.

[82] G. Klack, Photo of Thompson Ice Arena, (2018). commons.wikimedia.org/wiki/File:Rupert-Thompson-Ice-Arena-Dartmouth-College-Hanover-New-Hampshire-05-2018a.jpg.

[83] Falkenpost, Photos of Dulles International Airport, (2016). pixabay.com/photos/dulles-airport-building-1651554/.

[84] J. Harding, Dimensionality reduction for parametric design exploration, in: S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (Eds.), Advances in Architectural Geometry 2016, Zurich, Switzerland, 2016: pp. 204–221. doi:10.3218/3778-4.

[85] C. De Wolf, Database of embodied Quantity outputs (DeQo), (2016). www.carbondeqo.com (accessed April 10, 2019).

[86] C. De Wolf, Low carbon pathways for structural design: embodied life cycle impacts of building structures, Ph.D. Thesis, Massachusetts Institute of Technology, 2017. http://hdl.handle.net/1721.1/111491.

[87] C. De Wolf, F. Yang, D. Cox, A. Charlson, A.S. Hattan, J. Ochsendorf, Material quantities and embodied carbon dioxide in structures, Proceedings of the Institution of Civil Engineers: Engineering Sustainability. 169 (2016) 150–161. doi:10.1680/jensu.15.00033.

[88] P. Mathew, Building Performance Database, (2015). bpd.lbl.gov (accessed April 10, 2019).

[89] P.A. Mathew, L.N. Dunn, M.D. Sohn, A. Mercado, C. Custudio, T. Walter, Big-data for building energy performance: lessons from assembling a very large national database of building energy use, Applied Energy. 140 (2015) 85–93. doi:10.1016/j.apenergy.2014.11.042.

[90] Architecture - 2030, Architecture 2030 Challenge, (2006). architecture2030.org/2030_challenges/2030-challenge/.

[91] L.B. Liang, J.A. Jakubiec, A three-part visualisation framework to navigate complex multi-objective (>3) building performance optimisation design space, in: Proceedings of BSO 2018: 4th Building Simulation and Optimization Conference, Cambridge, UK, 2018. www.ibpsa.org/bso-2018-proceedings/.

Tables

Table 1: Energy upgrades and corresponding model settings for the example

Setting	Original	Upgrade
Envelope Upgrades	R-2.75 Walls + R-3.67 Roof Double Pane Clear Glass	R-5.5 Walls + R-9.17 Roof Double Pane Low-E Glazing
Lighting Upgrades	12 w/m^2	5 w/m^2
Equipment & Shower Reduction	12 w/m^2	5 w/m^2, 50% reduction in showers