Early-stage integration of architectural and structural performance in a parametric multi-objective design tool

N. Brown
Massachusetts Institute of Technology, Cambridge, MA, USA

J.I.F. de Oliveira
Universidade Federal Rural do Semi-Árido, Mossoró, Brazil

J. Ochsendorf
Massachusetts Institute of Technology, Cambridge, MA, USA

C. Mueller
Massachusetts Institute of Technology, Cambridge, MA, USA

ABSTRACT: In conceptual building design, an architect must simultaneously consider a variety of design objectives, including structural efficiency, total energy usage, and aesthetic expression. Multi-objective optimization (MOO) has been demonstrated to adequately account for designers' needs and guide them towards high performing solutions early in the design process. However, conceptual building designers seldom use MOO in practice, and although the use of parametric design tools is widespread, these tools rarely give rapid, multidimensional performance feedback to guide design exploration. In response, this paper describes relevant MOO methods and discusses how architects and engineers can use them to generate diverse, high-performing designs. It also introduces a number of computational tools that support MOO implementation and are embedded in traditional parametric modeling software. Finally, this paper presents a design case study of a cantilevered stadium roof to show how designers can effectively set up and navigate an architectural design space.

1 INTRODUCTION

In the current era of architectural and structural design, computational and numerical methods have expanded the shapes and forms designers are able to analyze and build. However, the software that has enabled this freedom is also largely responsible for removing considerations of performance from the early design phase. Since architects are now able to model complex geometries computationally without reference to gravity or other considerations, a typical workflow of architects coming up with conceptual designs before passing them off to engineers and other specialists for analysis has become commonplace. This process, along with the accompanying difficulty in translating between different software models, severely limits the ability of designers to create a performance feedback loop that influences overall building forms. As a result, sophisticated techniques such as structural optimization, which can reduce the cost and embodied energy of a building design, are often not used in the design process.

Furthermore, architects must simultaneously consider a variety of design objectives early on in the process, including ones than can be quantified, such as structural and energy efficiency, as well as others that cannot, such as aesthetic expression. Many of these objectives may trade off with one another, meaning an optimization process would require designer intuition and input to make overall design decisions. In order to facilitate better designs on a holistic level, optimization techniques must help designers navigate a complicated design space by providing functionality for searching through many designs options, prioritizing different objectives, and presenting rapid and reasonably accurate performance feedback. Multi-objective optimization (MOO) is a methodology designed for this purpose, and if used appropriately, it can account for designers' needs and guide them towards high performing solutions in conceptual design.

MOO is commonly applied in fields as diverse as finance and aerospace engineering, and researchers have developed a wealth of methods and algorithms to apply to various problems. Moreover, parametric modeling is already a commonly used tool in many architectural design firms, which lends itself to optimization. Yet for complex problems with high dimensional design and objective spaces, current architectural design tools lack the functionality and accessibility necessary for enabling widespread use of MOO techniques. Many conceptual designers also have little experience with setting up and using multi-objective workflows, since their development has been geared more towards pure engineering problems than expressive architectural design. In addition, specific architectural problems or designers themselves might require or prefer a wide range of techniques within the broader field of MOO. Thus, there is a need for research contributions that address the accessibility, ease of use, and flexibility of MOO processes for aiding conceptual designers in their search for diverse, creative forms.

2 LITERATURE REVIEW

2.1 *Multi-objective optimization for conceptual design*

Many initial contributions towards creating accessible, interactive optimization tools that consider aesthetics and other non-quantifiable objectives come from the field of structural engineering. Mueller & Ochsendorf (2015) created structureFIT, while von Buelow (2012) developed ParaGEN. The fields of mechanical, aerospace, and product design have also contributed significantly to the theory and application of MOO or Multidisciplinary Design Optimization (MDO), which is related. These contributions including many numerical optimization techniques (Vanderplaats 1999), overviews of simplified optimization workflows for engineering (Arora 2004), and the concept of isoperformance (de Weck & Jones 2006).

Other researchers have proposed and tested MOO workflows specifically for architectural design. Asl et al. (2014) establish and test an optimization method for whole building energy performance and daylighting. Flager et al. (2009) present a methodology and case study results for MOO while considering structure and energy as objectives. Krem et al. (2013) use MOO to study floor plan shapes and the location of the structural core. Quaglia et al. (2014) generate a Pareto optimal solution set for origami-inspired, rapidly deployable shelters. Mendez Echenagucia (2013) presents shape optimizations for a number of competing architectural performance objectives. Although these research contributions show the immense potential of MOO for use in design, few are focused on how conceptual designers can easily and interactively apply MOO to arrive at satisfying design solutions. Many proposed workflows also require a lengthy setup and computationally intense simulations, or complicated information transfers between multiple pieces of software, which may be beyond the scope of some designers.

2.2 *Existing tools*

A number of computational tools that can facilitate some MOO techniques have already been integrated into typical conceptual design environments. Both evolutionary and gradient-based solvers exist in parametric software, including Galapagos and Goat for Grasshopper. The Core Studio at Thornton Tomasetti has developed TT Toolbox and Design Explorer, which include a range of different computational tools that may be useful, such as a brute force solver for enumerating the design space and various visualization components. Another plug-in for Rhino, called Octopus (Vierlinger), was designed to apply evolutionary principles to parametric design problems with multiple objectives. It incudes features such as searching for tradeoffs, forcing diversity of solutions, changing objectives during a search, and visualizing and exporting results. A lighter version, named octopus.E, allows users more flexibility in picking and choosing only some of these functions. A similar tool, called Optimo (Rahmani), has been implemented for the parametric modeling plug-in Dynamo, which interfaces with many different pieces of architecture and engineering software. Users of Optimo can also use multi-objective evolutionary algorithms to help explore tradeoffs and generate optimal designs within a design space.

2.3 *Research contributions*

Some of these existing tools take an all or nothing approach to implementation of MOO, which limits flexibility for designers that prefer particular methods. Many also require sophisticated software knowledge to implement an efficient workflow for moving through a design space and selecting the best options. This greatly reduces the number of designers that can use MOO effectively, or at least increases the time and effort a designer must spend to consider performance. Ideally, future implementations will be as integrated as possible with typical conceptual design software, while building on the functionality of existing tools to create systematic workflows for applying potentially useful MOO techniques. In response, this paper describes a number of multi-objective optimization methods and discusses how conceptual designers can most effectively interact with them, while also considering how specific aspects of each method affect the eventual design solutions it generates. It also introduces a number of computational tools that support MOO implementation in traditional conceptual design software. Finally, a design case study of a cantilevered stadium roof is presented, giving visualizations associated with each interaction mode and illustrating how designers can use these methods to formulate a design space and select a best design.

3 DESIGNER INTERACTION WITH OPTIMIZATION

3.1 *Free exploration with performance feedback*

The first method for integrating performance feedback into the conceptual design phase is simply providing performance "scores" onscreen while a designer is interacting with a parametric model. This method depends on the designer to prioritize objectives and weigh tradeoffs with only the most basic computational assistance. However, due to the reliance on user intuition to move towards better performing designs, this method allows for complete design freedom within the parametric model. In an architectural workflow, it only requires evaluation methods for each dimension of performance, rather than additional optimization algorithms, and it most closely replicates a traditional parametric workflow by using sliders or other means of parametric navigation.

3.2 *Composite functions*

A second prominent method for multi-objective optimization is to form a composite objective function of the weighted sum of the objectives. In this method, the weight of each objective corresponds to a preference factor assigned to that particular objective (Deb 2001). Although this greatly simplifies the problem by reducing it to a single objective optimization and opening up the possibility of a single best solution, a user must still decide on these preference weights. In a conceptual design scenario, a designer would likely have to try a number of different weight combinations to arrive at a satisfying design. This can be done systematically through sampling, or interactively as part of a creative process. Special care must be given to architectural problems in which different objective functions have vastly different units or sensitivities, as certain objectives or combinations of correlating objectives can dominate the problem.

3.3 *Sampling, Pareto optimal design sets, & other methods*

Broadly speaking, the most common methods for multi-objective optimization require two main steps: finding multiple tradeoff solutions with a wide range of objective values, and choosing one of the obtained solutions using higher level information (Deb 2001). One of the most effective methods for first finding the range of best possible solutions uses the concept of Pareto optimality. There are a number of ways to determine or at least approximate the Pareto front of an objective space. The first is to brute-force sample the entire design space and use a Pareto-culling function after each design has been evaluated. The sampling can be random, on a grid, or use a more sophisticated algorithm. One advantage of sampling is that it can generate many different designs a certain distance away the Pareto front, but that an architect may want to consider for non-performance reasons. However, brute-force sampling can waste considerable computational time exploring large portions of the design space that are poor performing. The other method is to use an evolutionary algorithm such as the NSGA-II that approximates the Pareto front more quickly by breeding progressively higher-performing generations (Deb et al. 2002).

Another technique that loosely fits in the family of multi-objective optimization sampling is single variable sampling, where a designer changes one particular design variable while holding all others constant, and then visualizes the performance. This technique is useful when there is a preferred design, determined through some immeasurable architectural criteria, and designers wish to understand the importance of each variable to the problem and the direct effect of changing one of these variables from a given starting point. When the range of this single variable is plotted against each objective function, the slope of the objective function can indicate how sensitive it is to that particular variable. Other promising workflows exist, but many of them are a combination of the ones already mentioned, such as sampling different preference weights in a composite optimization. Thus, the methods listed in this paper are not exhaustive, but provide a useful overview for designers considering MOO in conceptual design.

3.4 *Tools to facilitate these interaction modes*

Although various workflows exist for each of these MOO techniques, they often require multiple transitions between software and manual manipulation of data. This section introduces a suite of tools created within Rhinoceros that are designed to work together to facilitate various MOO methods. Figure 1 gives a flowchart describing the connectivity of the developed components. Components 1-4 have been used in various research, teaching, and design exercises, but components 5 is still in development. Component 1 reads in design variables and their ranges and generates a list of design vectors that represent the design space. Geometry generation and objective functions are design specific, and must be set up individually by the designer. Component 2 records screenshots and performance, while component 3 contains an implementation of the NSGA-II MOO algorithm for use in parametric design (Durillo & Nebro 2011). Component 4 uses surrogate modeling to approximate computationally intense objectives, and component 5 helps organize and visualize overall design data for inspection.

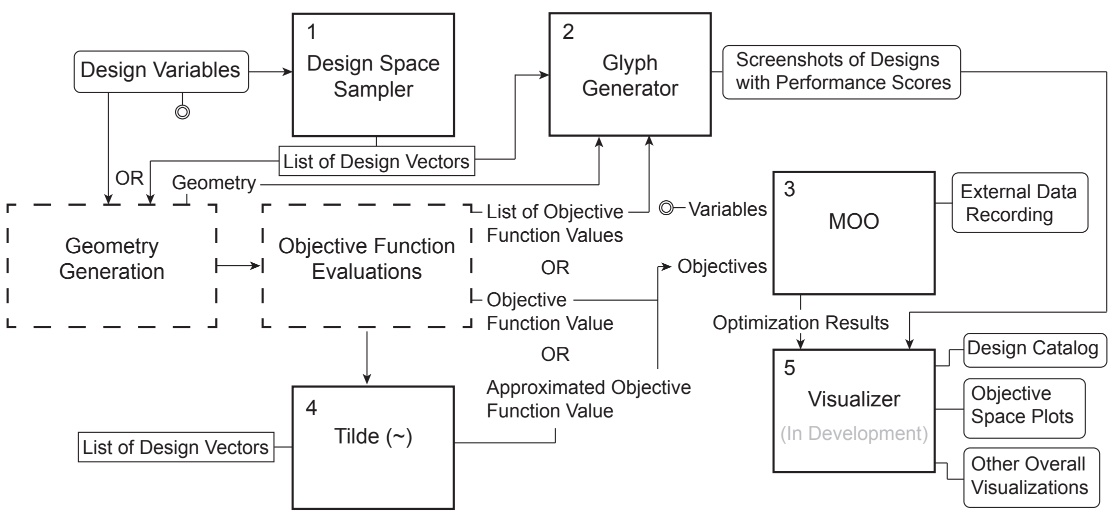

Figure 1. Process flowchart for implementing an MOO workflow using tools developed by the authors

4 CASE STUDY DEMONSTRATION: A CANTILEVERED STADIUM ROOF

4.1 *Design space formulation*

The potential outputs and utility of these interactive optimization methods will be demonstrated through their application to the architectural design problem of the cantilevered stadium roof. The cantilevered stadium form is commonly found at racetracks or other programs that require a covered grandstand. Many examples of a creative structural solution to this problem exist, including the reinforced concrete design of the Zarzuela Hippodrome. However, this paper utilizes a steel truss that sits on two columns supports and cantilevers out over the crowd seated below (Fig. 2). A parametric model of a cantilevered roof was created using Rhinoceros and Grasshopper. The model is made of a mixture of 2D and 3D geometry. The structural truss represents one section of the roof, corresponding to a single set of columns. The actual design would include a number of these trusses arrayed longitudinally with additional elements spanning between the trusses, but the reduction of parts of the problem to 2D allows for faster evaluations.

Four design variables drive the parametric logic of the truss: overhang length, backspan length, truss depth ratio denominator, and height of the free edge. These variables were selected because they generate the overall geometry of the roof in a way that allows for design variables to clearly effect performance in multiple architectural objectives. The truss model is projected onto the existing geometry of a crowd seating area and a hospitality suite above, which were previously dictated by the program of the stadium. It is assumed that the surfaces of the roof roughly follow the outer chords of the truss, although a slight offset at the tip location was created to allow for bullnose detail on the edge of the roof.

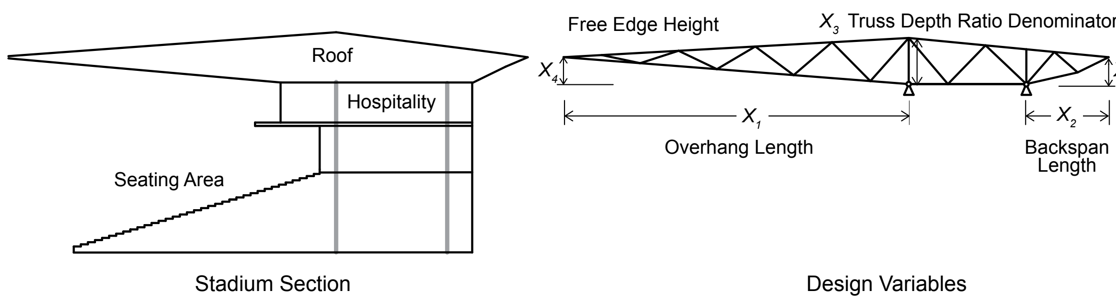

Figure 2. Geometric description of the design space for the cantilevered roof design case study

Table 1. Design variables for the case study

Variables	Lower Limit	Upper Limit
Overhang Length	20 m	30 m
Backspan Length	17 m	25 m
Truss Depth Ratio Denominator	5	9
Free Edge Height	0 m	4 m

4.2 *Objective function selection and optimization*

Four different performance objectives were selected for the multi-objective optimization process (Table 2). The evaluations functions used for each objective were simplified so that the interaction with the model yields nearly real time feedback. Although some of the MOO methods described above can be applied to models with longer evaluation times if an automated way of collecting results is established, the methods that require interactive articulation of design preferences require fast evaluations. The first objective is structural weight, where a design that uses the least amount of steel is considered to be optimal. To determine this weight, a finite element model of the truss was created using Karamba, which is a plug-in for Grasshopper. Karamba's sizing functionality was used, which reads in the applied loads, runs an analysis, and determines the smallest section size for each element that can provide adequate capacity to resist forces and buckling.

The second objective is rain protection, which is measured as the percentage of the crowd that would not be touched by blowing rain at a given windspeed and orientation. Although a computational fluid dynamics simulation could give a more accurate profile of blowing rain penetration into the crowd based on drop size, the effect of this penetration can be approximated geometrically by drawing a line from the tip of the roof at an angle based on the wind speed and droplet vertical velocity. In addition to rain protection, the shaded area of the crowd was also calculated and considered as an objective function, but this metric was deemed too similar to rain protection and less important in certain climates.

The third objective is sound dispersion. Acoustics in architecture is a complex field, and measurement of acoustic performance can be difficult to reduce to a single geometric objective function, especially since material selection often matters more than shape. However, it was assumed for this design that sound dispersion was desirable, and a flat or concave shape to the underside of the roof would be worse than an open, convex shape.

The final objective involves daylighting in the hospitality suites above the seating area, which is affected by the shape of the roof. A number of different evaluations and tools were considered for this metric, including those that run full annual simulations or calculate a daylight factor for one specific date and time. In the end, it was decided that the portion of sky visible from the hospitality suite, which can be calculated geometrically, was an acceptable approximation.

Using a combination of existing software, native Grasshopper functionality, and custom tools, each of the potential MOO workflows for design was implemented on the model of the cantilevered stadium roof. The limits of the design space for this problem were based on ranges considered acceptable from an aesthetic point of view, and the performance outputs were normalized, with the worst possible performance receiving a normalized score of 0, and the best performance receiving a score of 1.

Table 2. Objective function assumptions

Objective	Metric	Evaluation Method
Structural Efficiency	Weight of steel (min)	FEM + Member sizing
Rain Protection	Portion of crowd protected (max)	Intersection: rain trajectory and seats
Sound Dispersion	Angle of lower truss chord (max)	Geometric measurement
Portion of Visible Sky	Angle of line: hospitality to roof tip (max)	Geometric measurement

4.3 *Results and Visualizations*

The most basic methods for interacting with multi-objective feedback are real-time model manipulation or sampling the design space to set up a catalogue of generated designs. A small example catalogue for the cantilevered stadium is given in Figure 3, and it contains a glyph of the roof truss and a few typically used performance data visualizations. This type of glyph communicates information rapidly and visually and is appropriate for comparing all sorts of designs in which the variables are primarily geometric. Although this catalogue is presented statically on a page as separate design options, these visuals could be made interactive and dynamic using the tools introduced in Section 3.4. In this case, the graphics would update automatically during free parametric exploration, depending on the evaluations, and allow the designer to weight objective importance, test performance sensitivity, and consider aesthetics of the design while using intuition to drive towards a satisfying design.

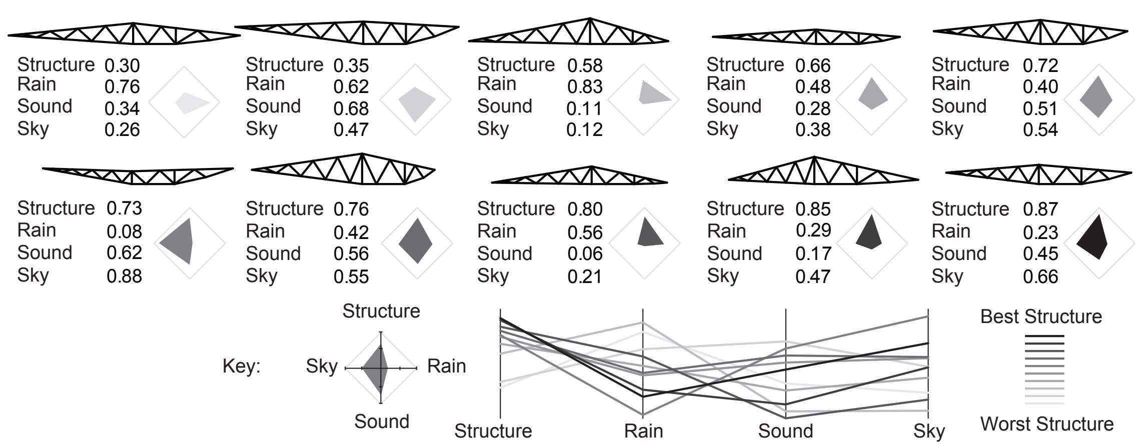

Figure 3. Example design catalogue

If a composite function is used instead, the different objective weights could be projected onto standard glyphs and visualizations. As illustrated by the cantilevered roof examples in Figure 4, performance in certain dimensions may be easy to obtain without significant weighting, suggesting that there are objectives (such as sound dispersion) that are largely independent and do not have major tradeoffs. Other objectives may dominate the composite function if a significant portion of the design space has high performance in that single dimension, depending on the normalization method used. Designers might also notice that certain objectives correlate closely with others (such as sound and sky), and could be eliminated from the problem. Since each of the objectives are quantified in vastly different ranges and units, a designer can use this composite function MOO optimization technique to test various tradeoffs, correlations, and sensitivities. This process can be done gradually to build intuition about the design problem or systematically to generate more design options, but it ensures that each selected design is optimal for a particular combination of designer preferences, ultimately leading to a successful design.

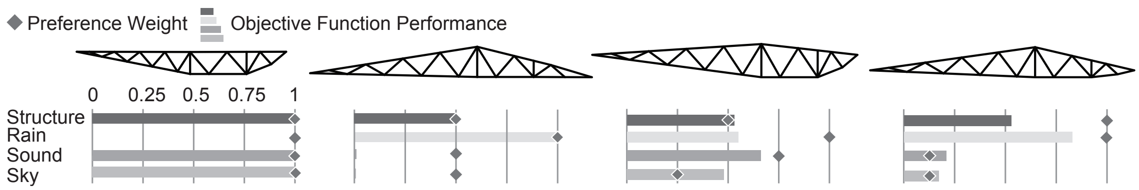

Figure 4. Optimal designs using different objective functions

Another typical visualization, especially useful while exploring a sampled design space or a generated Pareto front, is to plot the objective space with each design represented as a point. When more than three objectives exist, this can be done using projections to various bi-objective plots, such as the ones shown in Figure 5. At this scale, only the trends can be viewed effectively, but these plots could be made interactive in a parametric design workflow so that a user could click on one design and have it highlighted on every other plot. Overall, these plots give context to the set of highest performing designs, as well as indicating clear correlations (visible light and sound dispersion), almost linear tradeoffs (rain versus both of these), and curved Pareto fronts (structure and rain protection). By setting up a Pareto front, a conceptual designer is already moving towards high-performing designs, but likely has a variety of visual options to choose from within the set. In the case where a designer already has a preferred design for aesthetic, geometric, or other reasons, single variable sampling can prove useful. This technique, sometimes called one at a time analysis, can show the relationship between design variables and performance sensitivities by visualizing the performance gradient at a particular point (Fig. 6).

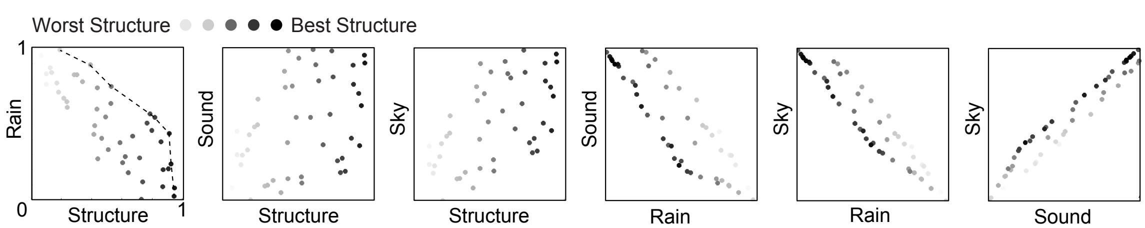

Figure 5. Automatically generated Pareto fronts for the cantilevered roof design example

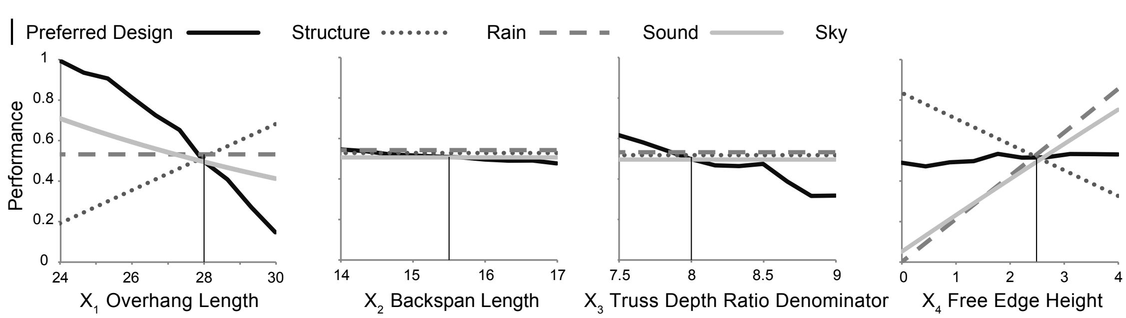

Figure 6. Results of a single variable sampling exercise for a preferred design within that design space

5 FUTURE WORK & CONCLUDING REMARKS

Currently, many of these generated visualizations require the export of data from the parametric modeler to another tool. Extensions to the already completed tools, which are being developed by the authors and are presented in Section 3.4, would add this functionality as its own interface within the parametric modeler. Another significant consideration when applying MOO is computational speed, which can limit interactivity. In light of this, the authors are currently developing and testing surrogate modeling tools for evaluations created in parametric modelers. These tools will approximate longer simulations with a less accurate but much faster evaluation, enabling free play, composite functions, and generally enhancing overall interactivity.

This future work will build on the contributions of this paper, which has discussed multi-objective optimization methods that are well within reach of conceptual designers who are already using parametric design tools. This paper has also presented a number of computational tools developed to facilitate these methods, while demonstrating their usefulness through visualizations, which are fundamental to interactive design methods that must consider and balance both quantitative and qualitative design goals. These contributions have the potential to encourage more widespread and effective usage of MOO in architectural and structural design.

6 ACKNOWLEDGEMENTS

The case study used in this paper was inspired by the conceptual design of the Curragh Racecourse redevelopment in County Kildare, Ireland. The authors are grateful for the opportunity to collaborate with Grimshaw Architects in their design process for this project. The authors would also like to thank Carl Rosenberg of Acentech Inc. and Alejandra Menchaca of Payette for their contributions concerning the acoustic and rain protection evaluation methods, respectively. This research and collaboration was facilitated by the Aga Khan Award for Architecture. Additional support for the research was provided by Science Without Borders / CAPES – Brazil.

REFERENCES

Arora, J.S. 2004. *Introduction to Optimum Design* 2nd ed., San Diego, CA: Elsevier Academic Press.

Asl, M.R. et al. 2014. BIM-based Parametric Building Energy Performance Multi- Objective Optimization. *Proceedings of the 32nd eCAADe Conference*, 224: 1–10.

von Buelow, P. 2012. Paragen: Performative Exploration of Generative Systems. *Journal of the International Association For Shell and Spatial Structures*, 53(4).

Deb, K. et al. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197.

Deb, K. 2001. *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons.

Durillo, J.J. & Nebro, A.J. 2011. jMetal: a Java Framework for Multi-Objective Optimization. *Advances in Engineering Software*, 42: 760–771.

Flager, F. et al. 2009. Multidisciplinary process integration and design optimization of a classroom building. *Electronic Journal of Information Technology in Construction*, 14: 595–612.

Krem, M. et al. 2013. Structural Configuration and Building Energy Performance. *Journal of Architectural Engineering*, 19(1): 29–40.

Mendez Echenagucia, T. 2013. *Computational Search in Architectural Design*. Politecnico di Torino.

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

Quaglia, C.P. et al. 2014. Balancing energy efficiency and structural performance through multi-objective shape optimization : Case study of a rapidly deployable origami-inspired shelter. *Energy and Buildings*, 82: 733–745.

Rahmani, M. Optimo. Available at: dynamobim.com/optimo/.

The Core Studio at Thorton Tomasetti, TT Toolbox. Available at: www.food4rhino.com/project/tttoolbox?ufh.

Vanderplaats, G.N. 1999. *Numerical Optimization Techniques for Engineering Design* 3rd ed., Colorado Springs, CO: Vanderplaats Research and Development.

Vierlinger, R., Octopus. Available at: www.food4rhino.com/project/octopus?ufh.

de Weck, O.L. & Jones, M.B. 2006. Isoperformance: Analysis and Design of Complex Systems with Desired Outcomes. *Systems Engineering*, 9(1): 45–61.