Informing Design through Parametric Integrated Structural Simulation

Iterative structural feedback for design decision support of complex trusses

Michael Makris[1], David Gerber[2], Anders Carlson[3], Doug Noble[4]
School of Architecture, University of Southern California, USA, [2]Sonny Astani Department of Civil and Environmental Engineering, Viterbi School of Engineering, University of Southern California, USA
[2,3,4]http://arch.usc.edu/faculty, [2]http://cee.usc.edu/faculty-staff/faculty-directory
[1]mpmakris@gmail.com, [2]dgerber@usc.edu, [3]andersca@usc.edu, [4]dnoble@usc.edu

Abstract. *The paper describes the development, testing, and initial findings of a design tool that generates parametrically defined, semi-automatically analyzed, and visualized structural performance of specific truss designs. The prototypical design tool provides structural truss solutions for spans of uniform to non-uniform surface curvatures. Real-time visual structural performance feedback enables the designer to more rapidly develop viable and potentially more efficient designs under user defined load conditions. The research methodology is an example of reinforcing structural learning and intuition within the design process. The research presents findings of the impact of iterative and interactive structural feedback through the development of a parametrically integrated structural truss analysis tool for aiding in design decision support.*
Keywords. *Design decision support; structural analysis; parametric design; design optimization; structural design.*

INTRODUCTION

The research is motivated by the notion of *informing form* and of enabling the architectural designer in their native tools, methodologies, and processes. The research is first targeting the need to integrate structural viability and efficiency directly into the form generation process of architectural design in early stage conceptual and formal exploration. Intrinsic to this design decision-making in the early stage are issues of limited domain knowledge, i.e. structural versus architectural; of design and analysis iteration speed and therefore feedback; of access and accuracy to structural analysis; and fundamentally of improving architectural designs and the design process. The work is predicated on the pervasive use of parametric exploration in contemporary design settings and on the architects' desire for formal exploration given the constraints of real world projects, such as fee, time, codes, life safety, and programmatic and contextual fitness. The work is further motivated by the desire to augment the design process with real time visual, qualitative, and quantitative cues for the evaluation of structural fitness in particular. In response, the research approaches the domain of early stage architectural design and its intrinsic formal exploration by building a prototypical tool and working method for more closely coupling geometry with structural loading, analysis, and optimization where real time visualizations provide for reinforced learning and feedback enabling the *informing of form*.

Contemporary architects are embracing efforts to improve designs and design process performance through computational methods and now, prolifically, through parametric design (Gerber, 2009). While parametric design methodologies in architecture have arguably enabled a dramatic increase in design exploration and of geometric and formal complexity, these methodologies have also incurred the need to integrate further intelligent analysis of this proliferation. Evaluating the many design solutions that now can be produced rapidly suggests the need for automated methods rather than by human post facto analysis (Gerber et al., 2012). Providing real time or near simultaneous analysis and feedback of concept design and form is driving informed design decision making earlier into the design process (Toth et al., 2011). Coupling the common practice of parametric form exploration to structural analysis and simulation, the work seeks to further impact design cycle latency as well as to *inform form* by reducing uncertainties such as: structural fitness and efficiencies; and later needs to redesign unsound solutions and their associated costs.

Structural evaluation in a building design is considered integral to the design process and should now be considered more so with respect to the parametric design definition processes (Holzer et al., 2007). Cost and feasibility analyses of structural systems should not be left to the design development stage but should be included and integrated into the evaluation of design solution spaces when architects are still considering as many possible solutions as practical (Sanguinetti et al., 2010). An automated structural analysis tool that can operate within commonly used design interfaces would potentially allow designers to explore solutions iteratively where structural performance and efficiency are increased and cost is reduced, in a more rapid and integral fashion. Having automated structural performance information early in the design process would furthermore allow engineers to collaborate more efficiently and intelligently, enabling them to provide expert knowledge in a more valuable detailed fashion as well in a more timely manner. It is conjectured that the engineer would also be able to provide input and improve design guidance earlier in the project design cycle, reducing the amount of structural revision and augmentation more commonly required later. Without near-instant structural analysis and feedback within the design tool, CAD geometries have no validated physical meaning in terms of their structural performance during initial design. Previous research has argued that the architect lacks meaningful knowledge of the capacity with which his or her design meets the constraints of structural and physics based reality (Bambardekar and Poerschke, 2009).

With this increased access to parametric modelling in the architectural profession, exploring more complex and unconventional structural configurations has become a more common and pressing problem. However, with these non-standard forms come non-standard structural behaviours that are not easy to predict or design for without the incorporation of computer based structural simulation. Most project budgets cannot afford multiple structural analyses and instead resort to a compromised solution in which structural constraints have been imposed on the original design increasing design latency and creating a negative impact on design intent. Given the ability of parametric design to generate expansive solution spaces, the evaluation of these designs for structural feasibility is paramount in the context of form finding through design exploration. This research focuses on the need for designer centric tools to determine structural behaviour quickly during parametric design, eliminating structurally unsolvable or less efficient design solutions and influencing the designer through visual and intuitive feedback to improve towards a more optimal design.

BACKGROUND AND REVIEW

The current design process in practice results in a form that is more often than not disconnected from structural simulation. This inefficient process begins with architectural development and follows with needed structural engineering revision (Nicholas and Burry, 2007). Current modes of practice can leave both architect and engineer dissatisfied with a negotiated and overly compromised result of poorly coordinated design and structural design analysis processes. Architectural and structural designers often take over a month working with a single design solution, producing a very limited set of design alternatives (Flager and Haymaker, 2007). This is primarily a design cycle latency issue that can be solved in part by domain integration (Gerber and Lin, 2013). While it is common knowledge and now almost standard practice in contemporary architectural design that parametric modellers provide designers the ability to generate expansive solution spaces, they do so in design isolation more often than not. However, there are numerous researchers and now practitioners who have begun to integrate structural principles directly into their design exploration processes and technologies (Shea et al., 2005). Current standard practice for collaboration between architectural and structural engineering domains is dependent on architectural models being exported and imported into structural analysis programs, which requires the generation of a new structural model with each new architectural design (Rolvink et al., 2010). While this method permits structural analysis whenever needed, it still represents the slower and limiting linear model of design first followed by analysis and engineering second. In defining a linked and informed design-analysis solution, architectural and structural parameters must continuously update and inform one another. The hurdle to integrate structural simulation into the common designer interface enables an iterative process that improves feedback and the overall design (Turrin et al., 2011).

One prominent precedent for our work into integration of design and simulation analysis operates by coordinating separate computer programs running simultaneously. The work by Sanguinetti et. al. uses a "generative synthesis process (GSP)" to auto-form structures within a design modeller and immediately ports the information to a specialized spreadsheet program for near-instant analysis. However, the researchers concluded this procedure remains dominated by one-way flow, as editing design parameters required a complete regeneration of the form. They then developed another method in which a parametric modeller cedes control to a spreadsheet. Editing the geometric parameters in the spreadsheet instantly updated the model while producing a new structural analysis. This more interactive approach allowed for real-time experimentation while producing instant feedback on structural fitness. More recently researchers have avoided the need for custom spreadsheets by linking commercial parametric design and analysis software so that changes to the architecture are instantly reflected in the analysis model in a second window (Gerber and Lin, 2012). Using computational methods to increase structural performance and decrease cost is a well-established research topic, where cost is inclusive of time, finance, cost of complexity and cost of fabrication and construction to name but a few. Some research has sought to provide architects with methods to better model geometry based on performance criteria. Oxman's research proposed curvature analysis of surfaces as a method to calculate custom thickness, developing an algorithm to take an input curved surface geometry and generate a smoothly transitioning thickness gradient at every point (Oxman, 2007). Significantly, this method generates 3D model geometry as a result of performance information illustrating to the designers real-world consequences of their design as they develop it.

The introduction of CAAD design tools also allowed architects to generate an increasing numbers of design proposals. Whereas limits of time and cost restricted traditional structural engineers from developing more than a few structural models for any project, this increase in design space exploration also required new methods of structural analysis automation such as meshing, centerline modeling and BIM data associated with geometries to simplify iterative modeling between software platforms. Other researchers and practitioners developed methods to automate the process of model generation so that structural analysis programs could be utilized for an increasing, but finite, number of designs proposed by architects. These methods produced adequate one-off analyses in which a finite set of architectural models produced by the architect could be individually analyzed and structural solutions proposed. However, recent advances in computational affordability and the adoption of parametric modeling by structural engineers can now allow architects to fine-tune design parameters in a much more expansive and exploratory iterative process. Researchers such as Rolvink et al. have developed complex computational and interconnected programming to allow near-instant structural analysis within design teams. Architects can refine designs and explore new options in one platform and engineers can provide structural feedback quickly and remotely from their domain specific software platform, allowing architects to further refine their designs in near real-time as illustrated in the work of Holzer et al.

Another set of precedents includes optimization methods once domain integration has been formalized. Researchers have begun to develop computational optimization and form finding of structures in conjunction with parametric design iteration. Across architectural and engineering disciplines, several different methods have been developed for this purpose. Each has shown that computational optimization is a worthwhile next step in the process (Shea et al., 2005; Holzer et al., 2007; Alfaris and Merello, 2008). The major drawbacks to date have been the computational lag of converting architectural to structural models and the relatively high level of computer knowledge required for these analyses. The interoperability and file format exchange process is a continual drawback. While Industry Foundation Classes (IFC) is a partial solution and while some building information modelling (BIM) packages have wire, joint, and surface information embedded in their objects to allow structural properties to be maintained between architectural and structural models, they do not yet adequately define and set loads and constraint definitions and checking of each member for accuracy.

RESEARCH PROBLEM

The research problem is to enable the designer with immediate actionable feedback of a structural solution for a given surface form. The research objective is to further inform form through integrating design process with structural simulation, sustaining iterative analytical feedback for the design decision making of complex trusses. The research gap is seen in the common training and domain expertise of the architect and in their native tool kit to inform their designs through structural analysis in an intuitive and real time fashion. This problem results in architectural solutions that are non-optimized at best, are typically inefficient or un-conservative, or are not physically possible at worst. Additionally, the current work-process of architects and engineers is slow and laborious, requiring architects and their engineers to wait on one another. This design cycle latency and asynchronous workflow limits the number of design iterations that can be considered for further development, and therefore arguably limits the final optimality of a particular design process. Moreover, unlinked architectural and structural models require engineers to spend large amounts of time reworking their analyses when architects make even small changes to these parameters, which occurs often in the early design stages. This inefficiency can be overcome in part by integration and automation. However, in order to achieve maximum structural efficiency, engineers must be engaged in this early design stage process. Linking architectural and structural computer models is key to developing a structurally informed architecture while still in conceptual design, without limiting the designer's ability to explore form and propose non-standard solutions. There are several projects and design teams worldwide that have the resources to achieve a fully integrated and efficient design process, but there are many times more that would like to explore parametric design spaces with limited resources and buy in from the engineer to effectively use this approach.

Our proposed solution and research is to develop a design-assist tool for structural trusses within the designer's native environment. One that can produce structural performance information quickly and in a format comfortable for designers that influences structural performance throughout design exploration, resulting in cost effective, aesthetically appealing, constructible and viable structural solutions. It is a tool that must also allow for an increase in potential solutions to be developed by architect and engineer in closely coupled collaborations. The expected results will be informed architectural designs that are meeting structural performance requirements, minimizing structure costs, and improving collaboration between domain experts. This design-assist intends to foster an increased understanding of structural behaviour for informed decision making of the designers of complex architectural forms.

Ideally, the design-assist tool could be expanded to all structural types but this research was limited to trusses as a starting point. Trusses are a common and efficient structural system to span long distances with minimal materials. Trusses are used in a range of scales and geometric forms from local beam or mullion sizes up to bridge or long-span roof systems. They are comprised of chord and web members and the continuous top and bottom chords are idealized as discrete axially loaded members that resist the bending moments developed along the span due to loading perpendicular to the span. Diagonal and vertical web members collect the shear forces that are proportional to the cumulative perpendicular loading of the truss span. The ability to model all the axial-load only members as springs and their broad application make trusses an ideal starting point for developing the design-assist tool.

RESEARCH METHODOLOGY AND PROTOTYPE DESCRIPTION

The primary research methodology is to build a prototypical tool based on technologies common to the contemporary design process. Rhino and Grasshopper are selected as the primary platform for development given the prevalent and easy to use modelling and graphical programming environments. Additionally, Grasshopper components Kangaroo and Galapagos are used for physics based simulations and evolutionary search in the Rhino based parametric design process. The research prototype consists of seven discrete parts in a sequence, that in total comprise a shareable Grasshopper definition, workflow and interface. These components each perform a specialized function and provide designer control over the structural design and simulation process (Figure 1). The seven steps and components are described in detail below.

As a first step, *Design Parameters and Inputs* are required to define the design problem; a surface (described in the next section) and specific trusses are modelled in Rhino. Here the designer specifies a custom form and path to follow or allows the tool to generate one of several automatic truss paths between two points in three dimensions. Other input parameters include the type of truss generated, here limited to the basic Pratt and two versions of Warren trusses; the number of segments; the geometric properties including height, width, and depth width with progressive variation; and connection geometries.

The second step, *Geometry* uses the previous design parameters to automatically generate a wireframe model of the initial truss choice. This component uses the previously input path of the truss to create a series of local orthogonal frames in which the individual joints of the truss are placed, depending on the user defined or default design parameters. With the joints created, a custom node and script calculates and instantiates linear bar segments to form the appropriate truss type.

The third step, *Structural Loads & Parameters* uses the wireframe conceptual model and manages the designer input for structural member sizes, load types and magnitudes, and connection types in order to automate the evaluations of the structural design. To alleviate the complexity or lack of domain expertise of these decision steps for non-technical designers, default choices are provided along with parameter options for customization.

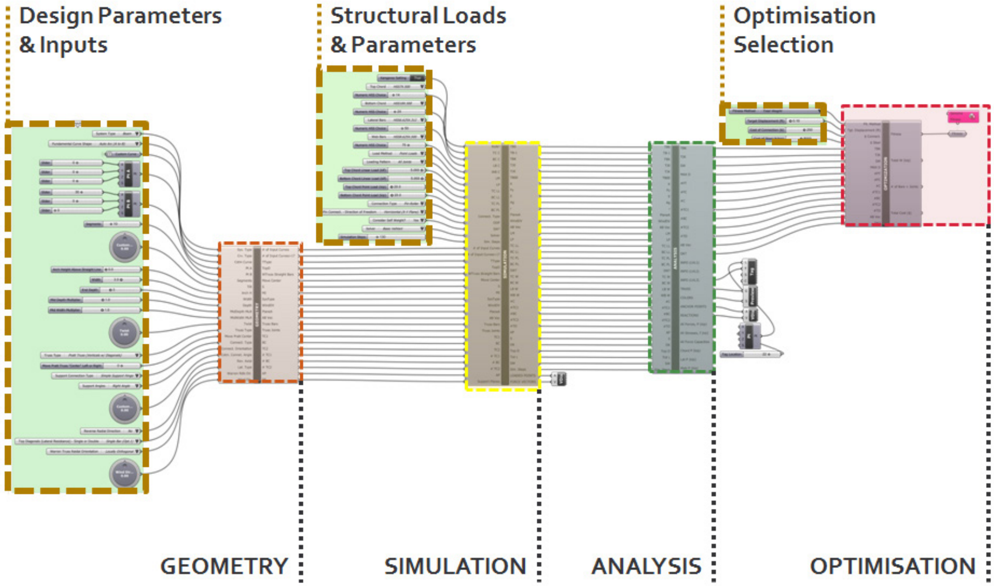

Figure 1
Illustration of the prototypes' graphical program encapsulated as seven steps of workflow; user interface and parameter input fields.

The fourth step, *Simulation* combines the wireframe model with the structural parameters, and creates a structural model through a series of automated steps. Based on elastic spring theory, each bar in the truss is simulated using a Kangaroo spring component and its mechanical properties. Loads are placed on the appropriate joints using a custom placement method that takes into account the load parameters given. Anchor points are constrained to model real-world connection details according to the input choices. This information is fed back into the Kangaroo engine, which performs *n* specified simulation steps. Kangaroo produces an ordered list of the locations of every truss joint as they move due to the loading during each simulation step. These deformed joint locations are culled for the 1^{st}, $n\text{-}1^{th}$, and n^{th} steps, which are reconstituted into deformed wireframe trusses using a similar method as that in the *Geometry* component.

The fifth step, *Analysis* takes the now loaded and deformed wireframe truss from the Kangaroo simulation and, through an analysis of the differential between the original and n^{th} truss, produces and visualizes deformation information for each bar. Using the mechanical properties that defined the structural model, these deformations are converted to strain, stress, and axial force measurements. Simultaneously, a custom code-check algorithm calculates the strength capacity of each bar using AISC Steel Manual compression and tension specifications. A strength-demand/strength-capacity calculation determines the efficiency of each bar used and determines which fail to meet their strength requirements. A colour gradient is modelled over a shaded version of the truss to communicate to designers which bars have excess, appropriate, and insufficient strength to meet their demands. Additionally, a deflection assessment is performed to measure truss displacements under the applied loads.

The sixth step, *Optimization Selection* enables designers to choose which metric determines the relative fitness of their truss and form design. The metrics provided include weight, number of members, cost, weight with deflection limits, and cost with deflection limits. Fitness is evolved and described in the last step.

The seventh and final step, *Optimization* uses another Grasshopper component, Galapagos to automate the use of an evolutionary algorithm to optimize the design for a singular fitness metric chosen. This automated fitness-function component evaluates the original and deformed truss geometries as well as the structural input information to determine the fitness value for the current design solution. The Galapagos engine records this fitness value for the current solution and runs an optimization routine, it then proceeds to change any designated input parameters in steps one and three, at which point a new progression proceeds. It should be noted the design assist tool uses only one metric as the primary fitness function and collapses all metrics into one through weighting factors, a limit of Galapagos discussed in greater detail in conclusions.

RESEARCH EXPERIMENTS, PROTOTYPE TESTING AND RESULTS

In order to evaluate the accuracy and validity of this tool, a series of five experiments were performed using an increasing level of complexity of truss to replicate both standard and highly unusual configurations. One primary research metric was that of the design assist tool's accuracy measured for all geometries. Industry standard SAP2000 was used as the engineering analysis benchmark for accuracy verification. Another critical research metric is that of impact on design process understood through speed of design exploration and value of the in-process analytical cues and visualization.

For each experimental case the accuracy benchmark process is as follows; a truss was created, loaded, and analyzed within the design assist tool. The wireframe model is 'baked' (saved as a fixed geometry) into Rhino, from which a series of .dxf files modeling the different bars in the truss were exported and then imported into SAP2000. Within SAP2000, with the geometries of the truss automatically placed, member sizes, bar numbers, load placements, and base constraints were assigned manually to match the Grasshopper tool setup. A batch file could be written to create a SAP2000 model from Rhino to automate the process, but the goal of this methodology is to make a native and inexpensive tool for designers that does not rely on expensive black box software. An analysis was run and internal force data was exported for each of the truss bars. Similarly, internal forces were exported from the Grasshopper tool to an Excel spreadsheet for comparison. Initially, results of the individual member forces comparing the tool and SAP2000 varied by as much as an average of 112.6% and a median of 2.1%. Average absolute member force error was as high as 2.61 kips (1 kip = 4.448 kN), an unacceptable level of error when designing structural members. However, once SAP2000's non-linear analysis mode was enabled to more realistically model second order deflections, errors dropped significantly to acceptable levels. While average error remained as high as 14.4% for one trial, median error across the trials was at or below 0.1% (Table 1). The high percentage errors are deceiving as they occur in close-to-zero load members. The absolute errors are small for all members. Average absolute bar force error never exceeded 0.02 kips, or 20 lbs, a statistically insignificant amount for structural design of this magnitude.

Where individual bars experienced unexpected and unusually high force errors when other bars in the truss were deemed accurate, these errors were traced to occur in bars with absolute forces of small magnitudes. It is clear that the largest percentage errors between SAP2000 and this tool occur in members that help to laterally brace the main chord members but take little force themselves, whereas critical members that carry the majority of the forces in each truss regularly have errors of less than 0.1%. For all members, the absolute errors are small. The results of these test trials show that the tool is accurate and reliable to within tenths of a kip for both standard and highly irregular truss geometries.

In parallel to ensuring structural accuracy, the research sought to qualify the design assist tool's impact on design process. Exhaustive end user tests have yet to be conducted, but metrics for continued

SUMMARY OF RESULTS							
Test Case	Number of Members	Average Error (%)	Median Error (%)	Average Absolute Error (kip)	Average Error (%) (Nonlinear)	Median Error (%) (Nonlinear)	Average Absolute Error (kip) (Nonlinear)
1	111	6.7	1.7	1.33	5.4	0.0	0.00
2	42	1.4	1.0	0.28	14.4	0.0	0.00
3	78	7.7	0.7	0.15	0.3	0.1	0.03
4	87	112.6	0.6	0.18	0.3	0.1	0.02
5	111	8.7	2.1	2.61	0.0	0.0	0.00

Table 1
Error Analysis for the five accuracy benchmark cases comparing the design assist tool to industry standard SAP2000.

research have been formulated. These include aspects of improving design exploration in terms of speed, e.g. iterations over time; quality of solution space, e.g. number of viable solutions within the constraint space; and designer computer interaction, e.g. ease of use and more so impact and integration of the analytical visualization in the design process. The tool already can evaluate designs for their cost and structural efficiencies and we believe through continued user experiments and protocol analysis we will be able to demonstrate and measure reinforced learning and impacts on design cognition as it relates to integration, problem scale (complexity) and coupling of architectural and structural design parameter spaces (Figure 2).

CONCLUSIONS

Linking commercial software does not alone achieve the goal of providing informed iterative feedback to designers without the need for advanced engineering expertise and the incurred design cycle latency. The design assist tool described here provides intuitive structural feedback directly on the design model and illustrates a novel approach to combining analytical data and visualization in real time design exploration process. In addition to the design assist tool's generation of truss geometries that follow complex surface forms, it allows the designer to observe structural simulation in real time. The success or failure of the tool is measured first by its ability to decrease the number of failing design proposals generated by users under a given load condition; and second by the time taken to reach a successful design solution. This time is understood to be a critical metric for determining design assist tools' ability to help designers iterate and search for a viable structural design solution while rejecting those that fail testing, thereby minimizing traditional design cycle latency. A third measure of success of the tool and method is truss efficiency; where the designer assisted truss outputs can be shown to reduce cost and/or weight over standard alternatives while fulfilling all requirements, project geometry and structural viability. In pursuit of these measures of success further research and development of the design tool to work with other structural systems are to be explored in an open source approach to expand the designers' access to integrated structural feedback in an intuitive and iterative manner.

REFERENCES

Alfaris, A and Merello, R 2008, 'The Generative Multi-Performance Design System' in *ACADIA 08 õ Silicon + Skin õ Biological Processes and Computation*, pp. 448-457.

Bambardekar, S and Poerschke, U 2009, 'The Architect as Performer of Energy Simulation in the Early Design Stage' in *Building Simulation 2009: Eleventh International IBPSA Conference*, Glasgow, Scotland, pp. 1306-1313.

Flager, F and Haymaker, J 2007, 'A comparison of multidisciplinary design, analysis and optimization processes in the building construction and aerospace industries' in I Smith (ed), *24th W78 Conference on Bringing ITC knowledge to work*, Maribor, Slovenia, pp. 625-630.

Gerber, DJ 2009, *The parametric affect: Computation, innovation and models for design exploration in contemporary architectural practice*, Design and Technology Report Series, Harvard Design School, Cambridge, MA.

Gerber, DJ and Lin, S-HE 2012, 'Designing-in performance

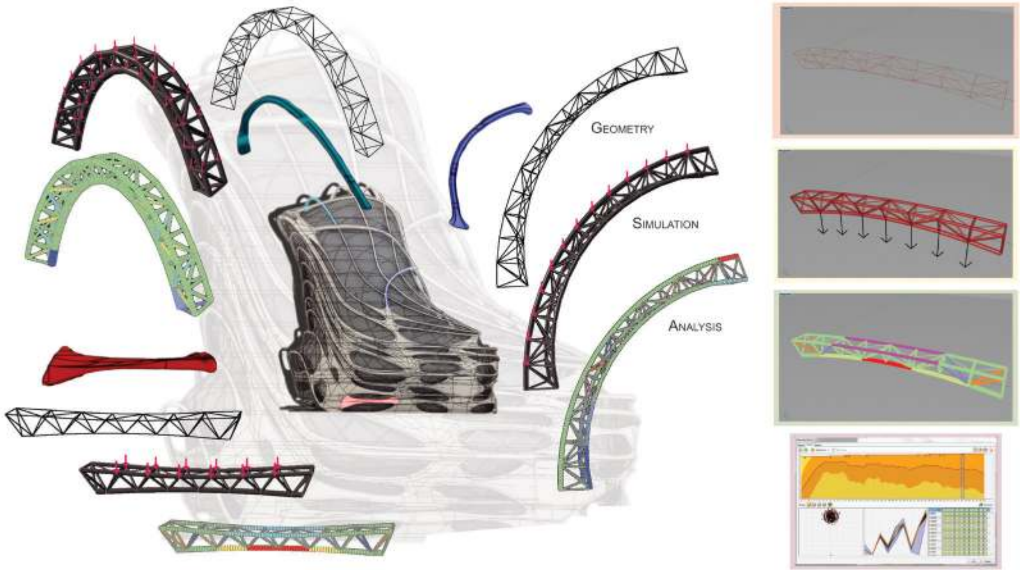

Figure 2
Illustration of common architectural design problem where structure and design are not closely coupled. The diagram shows three truss configurations extracted for testing the design assist tools with images of four steps along the right hand side including the graphical output and designer interaction with analytical visualization.

through parameterization, automation, and evolutionary algorithms: 'H.D.S. BEAGLE 1.0" in T Fischer et al. (eds), *CAADRIA 2012: Beyond Codes and Pixels*, Chennai, India, pp. 141-150.

Gerber, DJ and Lin, S-HE 2013, 'Designing in complexity: Simulation, integration, and multidisciplinary design optimization for architecture', *Simulation*.

Gerber, DJ, Lin, S-HE, Pan, BP and Solmaz, AS 2012, 'Design optioneering: Multi-disciplinary design optimization through parameterization, domain integration and automation of a genetic algorithm' in L Nikolovska and R Attar (eds), *SimAUD 2012*, Orlando, Florida, USA, pp. 23-30.

Holzer, D, Tengono, Y and Downing, S 2007, 'Developing a framework for linking design intelligence from multiple professions in the AEC industry' in A Dong, AV Moere and JS Gero (eds), *Computer-Aided Architectural Design Futures (CAADFutures) 2007*, Springer Netherlands, pp. 303-316.

Nicholas, P and Burry, M 2007, 'Import As: Interpretation and Precision Tools' in Y Gang, Z Qi and D Wei (eds), *CAADRIA 2007*, Southeast University, Nanjing, China: Southeast University and Nanjing University, pp. 249-257.

Oxman, N 2007, 'Get real towards performance-driven computational geometry', *International Journal of Architectural Computing*, 5(4), pp. 663-684.

Rolvink, A, van de Straat, R and Coenders, J 2010, 'Parametric structural design and beyond', *International Journal of Architectural Computing*, 8(3), pp. 319-336.

Sanguinetti, P, Bernal, M, El-Khaldi, M and Erwin, M 2010, 'Real-time design feedback: Coupling performance-knowledge with design' in A Khan (ed), *SimAUD 2010*, Orlando FL, USA, pp. 23-30.

Shea, K, Aish, R and Gourtovaia, M 2005, 'Towards integrated performance-driven generative design tools', *Automation in Construction*, 14(2), pp. 253-264.

Toth, B, Salim, FD, Drogemuller, R, Frazer, JH and Burry, J 2011, 'Closing the loop of design and analysis: Parametric modelling tools for early decision support' in *CAADRIA 2011*, Newcastle, Australia, pp. 525-534.

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