Proceedings of the International Association for
Shell and Spatial Structures (IASS) Symposium 2013
„BEYOND THE LIMITS OF MAN"
23-27 September, Wroclaw University of Technology, Poland
J.B. Obrębski and R. Tarczewski (eds.)

An Integrated Computational Approach for Creative Conceptual Structural Design

Caitlin Mueller[1], John Ochsendorf[2]

[1]Research Assistant, Department of Architecture, MIT, Cambridge, United States, *caitlinm@mit.edu*
[2]Professor, Department of Architecture and Department of Civil and Environmental Engineering, MIT, Cambridge, United States, *jao@mit.edu*

Summary: This paper introduces a new computational approach for creative conceptual structural design, synthesizing an interactive evolutionary framework, a structural grammar strategy for trans-typological design, and a performance-focused surrogate modelling technique. By developing and integrating these three strategies into a unified design approach, this research enables architects and structural designers to explore broad ranges of conceptual design alternatives in an interactive way.

Keywords: *conceptual structural design, structural optimization, interactive evolutionary algorithms, structural grammars, surrogate modelling*

1. INTRODUCTION

This paper proposes a new computational approach for incorporating structural considerations into the earliest stages of the architectural design process. Because structural behavior is most affected by geometrical form, the greatest potential for structural efficiency and a harmony of design goals occurs when global formal design decisions are made, in conceptual design. However, most existing computational tools lack the features necessary to take advantage of this potential: architectural modeling tools address geometry in absence of performance, and structural analysis tools require an already determined geometrical form. There is a need for a new type of computational strategy that allows designers to explore the relationship of form and structural performance, with an emphasis on design guidance, diversity of design alternatives, and computational speed.

The work presented here addresses these issues by proposing three design space strategies to be integrated into a new design approach: an interactive evolutionary framework, which balances creative design space exploration with a performance focus, a trans-typology structural grammar methodology, which defines design spaces that broadly span across standard structural types, and an automatic surrogate modeling strategy, which approximates the design space to enable a fast and interactive design environment. The integration of these strategies offers a way to overcome the weaknesses of standard methods, offering a promising alternative for conceptual structural design.

1.1. Existing computational design tools

Today's architecture and engineering practices make widespread use of computational tools throughout the design process, and currently available tools both reflect and enforce existing design paradigms of the architect as form-giver and the engineer as form-verifier [1].

Architecture tools, starting with computer-aided drafting programs in the 1980s, allow users to thoroughly document, and more recently generate, both conceptual and detailed designs. An increasing interest in complex geometry has led to powerful 3D modeling software which, coupled with scripting capabilities, enables the development of impressively complex forms. However, these tools rarely include functionality for legitimate structural performance evaluation, and therefore encourage users to develop geometry in a digital vacuum free of gravity. This reinforces the role of the architect as the form-generator without structural input, preventing integrated design.

Computational tools for structural analysis mirror architecture tools in their power and capacity for complexity, and yet also maintain existing design roles. Finite element analysis (FEA) programs are capable of determining stresses, deflections, and dynamic behavior for highly complicated geometry using very sophisticated techniques. Recent developments focus on increased accuracy and speed under a range of conditions. However, these tools are of little use in conceptual design; they require a geometry be provided to be analyzed, and are incapable of assisting with geometry synthesis. Again, these tools relegate engineers to the tasks of verifying the form and sizing the members, thus limiting or eliminating their involvement in conceptual design.

1.2. Key structural design tool features

The emerging research area of computational structural design tools seeks to bridge the gap between these existing computational approaches, enabling a better integration of structural input during conceptual design. This paper identifies two key types of features for such tools, feedback and guidance. Feedback features offer users a real-time understanding of how design changes affect structural performance according to required material volume, structural stiffness, or other quantitative metrics. This feature has been implemented in a number of applications both in research and practice, but is limited by the speed of computational structural analysis. Guidance features shift engineering software from the existing analysis and verification focus, enabling the software to suggest new geometries to the user in order to improve the structural performance of a design concept. While the field of optimization offers insight into ways to achieve this, there has only been preliminary progress in developing guidance-based tools for conceptual design both in research and practice. To truly encourage integrated conceptual structural design through modern computational tools, it is critical that methodologies that achieve this functionality be further developed.

1.3. Need for guidance-based structural design approach

This paper addresses the problem of integrating structural guidance into conceptual design through computational means. To achieve this, there are three specific and inter-related requirements for which this research offers solutions through original contributions.

First, guidance-based tools must balance the ability to suggest design changes with freedom of creative exploration within the design environment. There is no single correct answer in architectural design, and it is crucial that such tools allow for a plurality of design options, while nevertheless encouraging the user towards those with better performance. Section 2 offers a new approach to achieve these goals using interactive evolutionary exploration.

For use in conceptual design, a guidance-based methodology should perform like a talented team member in a brainstorming session, generating a broad range of new and unexpected design ideas. This capability is important not only to improve structural performance, but also to discover exciting architectural forms. To accomplish this, the methodology should incorporate a broad and varied design space. Section 3 presents a strategy to achieve diversity and breadth of solutions through structural grammars.

A third hurdle in integrating structure into computational design tools is that structural analysis techniques used in guidance-based features can be time-consuming. A successful approach should include rapid performance prediction strategies to allow for an interactive, real-time user experience. Section 4 addresses this issue by introducing a performance-focused surrogate modeling strategy for design space approximation.

In addition to addressing these three needs, it is necessary to consider how to synthesize a new unified approach from the individual strategies.

Section 5 considers the integration of the three strategies into an original computational approach for design space exploration, definition, and approximation that offers free yet directed exploration of a diverse design space in an instantaneous manner.

2. INTERACTIVE EVOLUTIONARY FRAMEWORK

This section introduces the first of three design space strategies, an interactive evolutionary framework for conceptual structural design. This framework is an extensible and generalized approach for using interactive evolutionary algorithms to explore a broad range of structural design problems, and offers design guidance with enhanced approaches for user interactivity, a new method to promote design quality and diversity, and user interface implementation.

2.1. Design space exploration

In what ways can designers survey the space of possible solutions to a design problem in search of good designs? The most obvious way is through random or educated guessing: think of several possible solutions and evaluate them. Through rapid feedback, design tools offer users a way to map out small portions of the design space in this way. Another approach is to be guided to the best solutions by a computer algorithm. Optimization-based guidance can bring users directly the point of interest, the optimal design.

However, neither of these approaches is completely satisfying. Ideally, a tool should point users in the directions of good designs, but should still allow them the freedom to explore, incorporating qualitative design goals. This type of approach acknowledges the limitations of standard optimization while nevertheless taking advantage of computational power. A promising way to achieve this functionality is through interactive evolutionary algorithms.

2.2. Interactive evolutionary algorithms

Evolutionary algorithms are a general class of optimization strategies that use the principles of Darwinian natural selection to grow and evolve populations of designs. Like other heuristic algorithms, they incorporate randomness, so that they avoid getting stuck in local optima and can effectively navigate around the design space in search of better solutions. The general procedure is to randomly initialize a first generation of candidate designs, evaluate the fitness of each member of the generation, identify the top performers, and use those to create a subsequent generation by combining and mutating them. In standard evolutionary algorithms, the process runs automatically until preset criteria are reached, and a single solution is presented as the optimum. However, it is also possible to take better advantage of the design diversity created by this approach by incorporating human interaction.

Interactive evolutionary algorithms combine principles of evolution with human input to drive design space exploration. The cycle differs from standard evolutionary algorithms at the design selection step. The algorithm identifies top performers, but solicits input from the user to make final choices about which designs to proceed with to form the subsequent generation. This key difference allows the designer to adjust the optimization process based on unformulated goals, such as visual impact or constructability requirements. Furthermore, the user may adapt goals across generations, based on newly realized design criteria discovered in the explorative process.

Interactive evolutionary algorithms have proven effective in a number of disciplines that incorporate both quantitative and qualitative goals, including structural design [2][3]. The work presented here expands on preliminary work by offering a generalized approach and an expanded user experience.

2.3. Framework and implementation

This section introduces a flexible framework that applies an interactive evolutionary algorithm to structural design problems. The framework has been implemented in a web-based design tool for exploring the geometry of planar trusses, structureFIT [4]. The tool uses normalized required volume as the structural performance metric, although the framework supports any quantitative measurement of performance. The implementation has a graphical user interface with three main screens: *set up model*, which allows for open-ended problem definition, *explore solutions*, which enables guided design space exploration using the interactive evolutionary algorithm (shown in Fig. 1), and *refine design*, which allows the user to make fine-tuned adjustments to a selected design and observe resulting performance values through real-time feedback of numerical score and member sizes (shown in Fig. 2). The full sequence of design modes offers the user more control than previous interactive evolutionary implementations, in that he may graphically define the initial geometry, loading, supports, variables, and bounds in the first mode, and may adjust and finalize results found through evolutionary exploration in the third mode.

The interactive evolutionary exploration mode also offers new developments itself. Previous implementations allow the user to communicate qualitative preferences through a somewhat limited interaction: the selection of two parent designs in each generation. In this framework, the user is able to select zero, one, or multiple designs as parents, due to a more flexible crossover computation that relies less literally on a biological metaphor. Critically, the user is also able to control the mutation rate and generation size used to create each new generation. This offers significant control of how quickly the algorithm drives towards optimal solutions versus remaining in a suboptimal but qualitatively interesting region of the design space. Repeated experimentation with these parameters also allows the user to sense the topography of the design space, or in other words, whether changes in design variables have small or large impacts on the performance score.

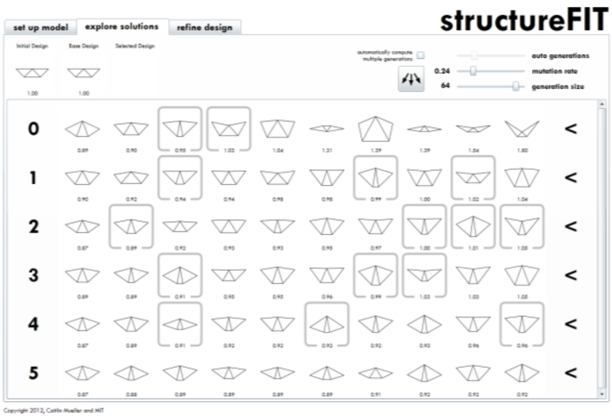

Fig. 1. Screenshot of interactive evolutionary exploration mode.

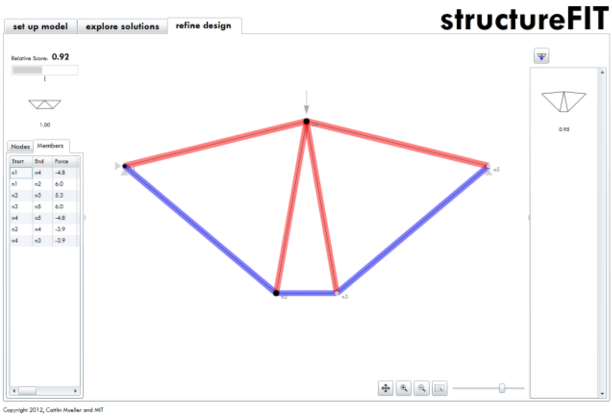

Fig. 2. Screenshot of design refinement mode with real-time feedback.

In summary, this framework and its implementation allow the user to combine the power of computation and optimization with the freedom of creative exploration. Fig. 3 illustrates a range of alternatives found for three design problems that perform well and vary significantly from each other. Designers can use this approach to find a variety of conceptual structural design solutions that perform quantitatively well while also meeting important but unformulated qualitative goals such as aesthetics, constructability, and contextual fit.

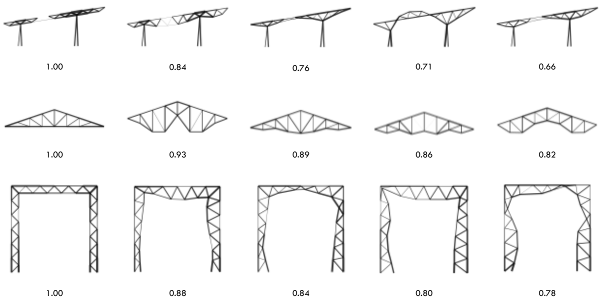

Fig. 3. Examples of explored, with numerical scores indicating normalized required volume.

3. TRANS-TYPOLOGY STRUCTURAL GRAMMARS

In computational design, the design space contains all possible solutions of a system. Optimization methods focus on how to locate the best performer(s) in a given design space, and more nuanced approaches like the interactive evolutionary approach presented above allow freer design space exploration. However, it is also important to consider the design space itself. No matter how well optimization or exploration approaches work, they are limited by the solutions that can be found in the design space of a particular problem formulation. This section motivates the need for ways to define broad and diverse design spaces, discusses types of design spaces for conceptual structural design, and makes the case for rule-based, or grammatical, approaches to design space definition.

3.1. Trans-typological design

The earliest steps in the contemporary conceptual structural design process involve choosing a typology or system. For instance, in a long-span roof design, should the structural action be carried out with an arch, a cable, a fan-like scheme, a bending option, or with a truss? The world's best structural designers are able to brainstorm a range of creative ideas and can intuitively estimate relative performance of competing concepts. For example, the German structural engineer Jörg Schlaich generated a range of exciting conceptual design possibilities for a bridge competition [5], illustrated in part in Fig. 4.

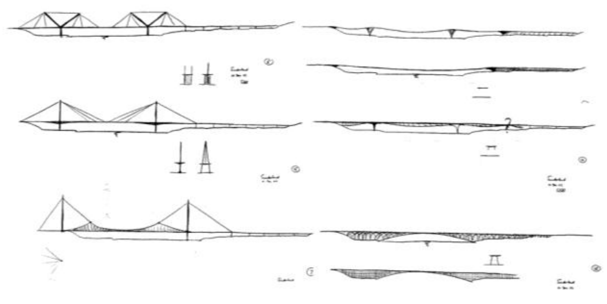

Fig. 4. Conceptual design alternatives developed by Schlaich [5].

Currently, in the most successful examples, the generation of these typological ideas and the selection between them are carried out by expert practitioners with many years of experience and keen intuitions, like Schlaich. In less successful approaches, fewer typological ideas are considered, or an ill-fitting typology is chosen without adequate consideration. There is room for bias and human error to influence this step in the process, which is arguably the most important step because it determines many characteristics of the overall form. There is therefore a strong and unaddressed need to develop computational methodologies for exploring possibilities across typological boundaries. While some masters in the structural design field excel at doing this by hand, the computer can help in several ways. First, given a broad enough design space definition, computational techniques can automatically generate a range of solutions to consider, behaving like a creative brainstorming partner. Second, computation can be used to quantitatively evaluate design options according to structural behavior. This is standard practice as a way to compare designs within a set typology, such as trusses of various configurations, but is rarely used to compare designs across typologies.

3.2. Parametric vs. rule-based design spaces

In both optimization and architectural computation in general, the most obvious type of design space is one that is parameter-based, sometimes called variable-based. Each parameter or design variable constitutes one dimension in the space. Parameters can explicitly relate to particular spatial definitions of a design, or they can more globally control a design's geometry as a whole, which helps limit the design space dimension. In both cases, the designs found in this type of space are parametric variations of each other. Through clever parameter definition, it is possible to define somewhat broad design spaces that exhibit diversity in possible solutions. This type of space can be useful in exploring design decisions once the overall formal strategies and structural systems have been decided upon. However, it is practically impossible to define a parametric design space that covers the range of possibilities that one would like to consider during conceptual structural design. This is related to the fact that one can enumerate a parametric design space– that is, list every possible design it contains−or at least map it exhaustively at a finite resolution.

A compelling way to move beyond the limitations of parametric variation is by using rule-based systems, or grammars, instead of parameter settings to generate designs. Based on Noam Chomsky's theories of generative grammars in language, George Stiny and James Gips proposed generative grammars for geometric shapes, or shape grammars [6]. As Stiny later explained, "[Chomsky's] idea was that a grammar had a limited number of rules that could generate an unlimited number of different things, and that the resulting language was the set of things the rules produced" [7]. Just as there are an unlimited number of new and creative sentences that can be uttered in a language, a grammar for shapes can yield an infinite number of new and creative designs.

3.3. Structural grammars

Because of the breadth and richness of design spaces defined by grammars and rules, they are a better candidate for enabling trans-typological explorations than parametric design spaces. However, the application of geometric shape grammars to the field of conceptual structural design is not trivial. While the generative power of grammars is great, there is a danger that grammars can be too broad, capable of generating forms that make little sense in the physical and structural world. It is therefore critical that grammars used in structural design be sufficiently restrictive and incorporate structural information into rule definitions. Grammars in architectural and engineering domains that move beyond shapes were first suggested by Mitchell [8], who proposed functional grammars with rules that incorporate engineering and fabrication knowledge. More recent research further extends the concept of functional grammars [9], but falls short of defining broad trans-typological design spaces leading to unexpected design possibilities. This existing work suggests that there is a need to further develop a prescription for defining grammatical design spaces that cover a wide range of structurally feasible yet innovative options. A prescription for this type of trans-typology structural grammar is given in the following section.

3.4. Trans-typological grammar features

The trans-typology grammar approach involves three types of computational classes: shapes, grammars, and analysis engines. A particular type of shape is operated upon by a particular grammar, and analyzed for structural performance by a particular analysis engine. In the generalized approach presented here, any shape/grammar/analysis set can be used that follows the same pattern.

3.4.1. Structural shapes

Structural shapes are defined by their properties. A shape object is an instantiation of the general shape class that has particular property settings. Properties are shape-specific and include single and group functional designations for geometric elements like lines, points, and areas that dictate their behavior. Properties also include a state label, which will be discussed later. These designations allow rules and analysis engines to identify and act on certain parts of the structural shape. Structural shapes must include more than pure geometric data. Important structural information, such as loading, material properties, support conditions, and allowable structural behavior should be encoded and accessed by rules and analysis engines. This is accomplished by incorporating non-visual data into the computational representation structural shape. While the graphical depiction shows the geometry, the underlying formulation contains a richer set of properties.

3.4.2. Types of rules

A trans-typology structural grammar can be described by the list of rules that it contains and an initial structural shape to begin rule application with. Rules adjust the structural shape through addition, subtraction, subdivision, and other modifications to geometric or structural properties. A rule has a left-hand side, or LHS (the structural shape prior to rule application) and a right-hand side, or RHS (the structural shape after application), and can only apply to a structural shape that matches its left-hand side.

When rules can be applied recursively, there are an infinite number of rule application sequences that determine unique designs, a key principle for defining a broad design space. Another important feature of the structural grammar's rules is the use of state labels. A state label is a way to control which rules can be applied to structural shapes at various times in the rule application process. In the trans-typological structural grammar approach presented here, a structural shape is always in a particular state, and a rule can only apply to structural shapes in one or more specified states. Rules can change the state of a structural shape, thereby changing the rules that can subsequently apply to it, although they may also maintain the current state. To allow for maximum flexibility in rule applications, trans-typology structural grammars also include parametric rules. The application of a parametric rule is dependent on one or more parameters that help to define its behavior. Parameters can be continuous numerical values, integers, binary values, or members of a discrete set. Finally, a critical class of rules in structural grammars incorporates structural awareness and insight. This is important because it limits the results to those that are structurally feasible. In contrast, a rule that chose an arch or cable shapes arbitrarily would likely yield highly irrational or impossible forms.

3.4.3. Structural performance evaluation

While structurally aware rules in trans-typology structural grammars are useful for restriction, it is still necessary to compare among structurally feasible design possibilities. The evaluation method is necessarily grammar-specific, since different grammars include different assumptions about structural behavior. In general, the evaluation method should utilize some kind of analysis engine that produces a numerical score for a given structural shape.

3.5. Example: pedestrian bridge grammar

To demonstrate the power of this approach to generate diverse and interesting designs, this section introduces a more realistic and complex trans-typology structural grammar developed to generate designs for short- and medium-span pedestrian bridges. The grammar is inspired by creative and innovative bridge designs involving a variety of types of cable solutions, including suspension bridges, cable-stayed bridges, and solutions between and beyond. This section outlines the grammar and illustrates a range of generated designs. There are 21 rules in the pedestrian bridge grammar, and according to the trans-typological prescription, these rules use recursion, parameters, and state labels, and often incorporate structural logic and knowledge. A sample rule is given in Fig. 5. Fig. 6 shows 15 designs randomly generated by the pedestrian bridge grammar. These designs demonstrate the breadth of the grammatical design space, including both the cable-stayed bridge typology, the suspension bridge typology, and space in between the two. There are many unexpected results that emerge from a relatively small set of rules, potentially suggesting innovative and creative solutions that have yet to be built.

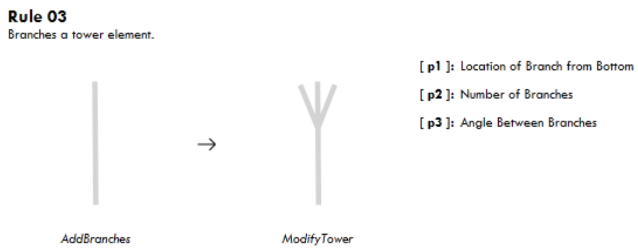

Fig. 5. A rule from the pedestrian bridge grammar that parametrically branches a vertical tower structural shape.

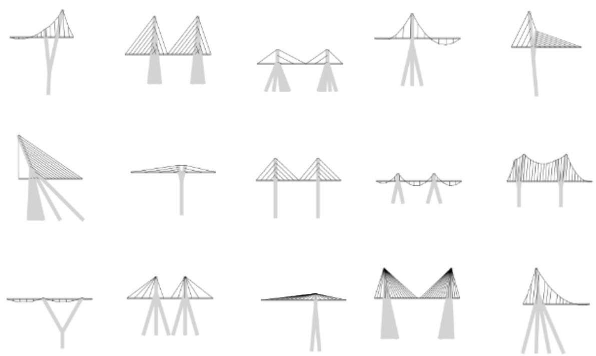

Fig. 6. Example pedestrian bridges generated randomly using the grammar.

In summary, this section has presented a novel way to define broad, diverse design spaces that can generate creative and exciting design alternatives for conceptual structural design. Through the use of trans-typology structural grammars, designers can explore concepts that range across traditional typologies in an automated, computational manner. This is important because both because new and unexpected forms can be discovered, and because a broad range of forms can be quickly and quantitatively compared.

4. PERFORMANCE-FOCUSED SURROGATE MODELING

This section presents the third of three design space strategies, a surrogate modeling approximation approach that greatly reduces the computational speed required to evaluate performance in conceptual structural design tools. Surrogate modeling substitutes a low fidelity, computationally inexpensive model, or surrogate, for an original high fidelity model [10]. In general, the challenge of this method is to find a surrogate model that is sufficiently accurate. This section proposes an approach that focuses on accuracy in high-performing design space regions, tunes models automatically, and adapts to fit user preferences.

4.1. Design space approximation

Even in conceptual design, mathematical models of structural designs can become unwieldy and difficult to evaluate in a manner rapid enough for a fast-paced, interactive design tool. This is because evaluation methods for structures, such as finite element analysis, typically involve solving large linear systems. While the wait time for a single analysis run is tolerable in a traditional application, newer design space exploration approaches, such as the interactive evolutionary framework presented previously, require the evaluation of tens or hundreds of designs at once, and demand increased computational performance.

To facilitate such exploration, an approximation of the design space can be used: the performance of a design concept at a particular point in the design space is predicted using a data-based response surface. The benefit of an approximation approach like this is that compared to more accurate analysis-based performance evaluation, performance prediction takes negligible computation time. Therefore, hundreds or thousands of design points can be visited and approximately evaluated by the computer nearly instantly. The drawback of this type of approach is that the performance prediction may be quite inaccurate, so design decisions made based on the predictions may be ill-informed. The key to design space approximation is navigating this tradeoff between response time and accuracy. In structural design applications, it is also important that the approximation strategy be accessible and easy to apply.

4.2. Surrogate modeling strategies

Surrogate modeling is a compelling approach to design space approximation that uses statistical, or data-based, models to estimate performance. Fig. 7 illustrates the basic concept of surrogate modeling: statistical models are built to attempt to fit a curve or surface to a set of data points generated through computer simulation (usually randomly generated and then actually evaluated), called training data. This curve or surface is then used to nearly instantly predict the performance of newly generated data points, avoiding computationally expensive simulation. The curve or surface generally includes some degree of error, both at the points it is trained on, and the newly generated points it is tested on.

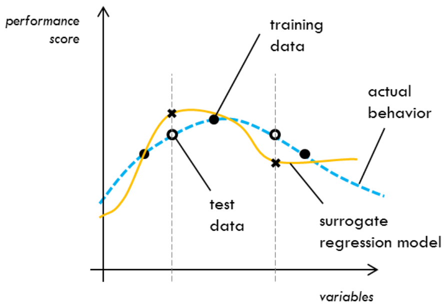

Fig. 7. Illustration of surrogate modeling approach.

Existing surrogate modeling strategies use polynomial or other parametric regression functions as models, and involve choosing coefficients and other parameters through careful tuning, which is problem-specific and requires considerable expertise. For practical application in structural design by practitioners with limited backgrounds in surrogate modeling, it is necessary to adapt existing approaches to be more robust and automatic.

4.3. Surrogate model types

Because polynomial and other function-based modeling strategies are difficult to apply consistently to a wide range of problems in an automated way, it is necessary to consider other regression model types as potential surrogates. In machine learning, significant study has been given to off-the-shelf or black-box methods that work well on many problem types without much tuning or expertise required [11]. Furthermore, the machine learning technique of bagging, or bootstrap aggregating, has been found to be an effective way to increase model robustness by averaging the results of an ensemble of regression models [11].

The combination of these methods results in regression modeling techniques work very well as off-the-shelf approaches for automatic predictive modeling. Two different modeling strategies of this type, ensemble neural networks and random forests, have become popular in the machine learning realm for their combination of robustness and predictive power. Despite their common use in machine learning, they have not been frequently applied in surrogate modeling applications. It is proposed that they be used instead of standard surrogate modeling types in cases where systematic, non-expert model building is important, such as in a conceptual structural design environment.

4.1. Sampling plans and error measures

Beyond the modeling strategy itself, this new approach proposes modifications to the sampling scheme used to generate training data points and the error measures used to evaluate the accuracy of models. Because conceptual design tools aim to help designers find high-quality design alternatives, it is more important that an approximate model be accurate in high-performing regions of the design space than overall. Poorly performing design space regions often contain discontinuities or especially steep topography that can be particularly difficult to match with regression models. It is therefore proposed that the data points generated to train the surrogate model over-represent high-performing design space regions and only minimally represent low-performing regions. This can be achieved by weighted sampling schemes that include duplicate copies of data points that perform well. Because model-building techniques work to minimize error over all training data points, the duplicated high-performing designs will effectively weight the surrogate model towards accuracy where it is most critical.

The measurement of error should also be reconsidered to account for conceptual design needs. Specifically, this approach places an emphasis on the predicted rank of a candidate design over its predicted score; in other words, it's more important for the approximation strategy to correctly identify the better of two conceptual design alternatives than to estimate the performance value itself. Additionally, it is more important to consider error in top-performing candidate designs than in those that perform poorly. Traditional surrogate modeling approaches use error measures like root mean square error (RMSE), which is the square root of the average squared difference between predicted and actual design performance. A new error measure, top mean rank error (TMRE) is proposed here, which finds the average absolute difference of predicted and actual design ranks over a top-performing subset of all the designs in a set of test data points. This new error measure can be used to evaluate the accuracy of a surrogate model, but can also be used in building surrogate models themselves in the parameter-tuning step, which tries multiple model parameters and selects the value that leads to the smallest error. Along with the weighted sampling plan, this can lead to approximate models that perform better at evaluating high-performing design alternatives quickly.

4.2. Case studies

To test the impact of the proposed approach, a case study considered four conceptual structural design problems involving exploration of nodal positions in planar trusses of increasing complexity, labeled (a) through (d). For each problem, surrogate models were trained using an increasing number of samples and two approaches: standard, which uniformly sampled the design space for training data points and used the traditional RMSE error measure, and proposed, which used the weighted sampling plan discussed above and the new TMRE measure.

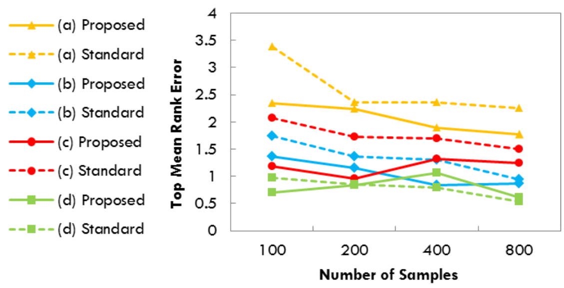

Fig. 8. Surrogate model performance for case study problems with increasing training data set size.

The results are illustrates in Fig. 8, which shows that the proposed approach was more accurately able to predict the relative performance of top-performing candidate designs than the standard method. This difference was generally more pronounced for a small number of training samples, indicating that this new approach is even more effective when there is less time to generate training data.

In summary, this design space approximation approach offers a way to quickly compare the performance of many design alternatives in conceptual design without relying on costly structural analysis. This is important because designers can use the saved time to consider a much broader range of design alternatives, leading to better conceptual design decisions.

5. INTEGRATION OF DESIGN SPACE STRATEGIES

Three complementary computational strategies for conceptual structural design have been introduced in this paper, dealing with exploration, definition, and approximation of the design space of possible solutions. These three strategies help to facilitate a focus on structural performance, design creativity, and immediate, interactive results, and together comprise a new strategy for using computation in the early-stage design of architectural structures. This section will briefly outline the ways in which the three strategies can be combined into a new, unified computational design approach by considering three pairwise combinations of the three approaches.

5.1.1. Evolutionary framework and structural grammars

Interactive evolutionary exploration of design spaces based on structural grammars raises the key challenge of creating “crossed over” offspring from parents whose underlying formulation is not a design vector, but a variable-length sequence of rules and parameter settings. While crossover can be implemented using averages of design variables for parametric design spaces, the variable length of rule derivations makes this approach impossible for rule-based design spaces. Instead, an approach is proposed that implements crossover by splicing and swapping rule sequences at mutually allowable points for two parents. An example of this concept is illustrated in Fig. 9.

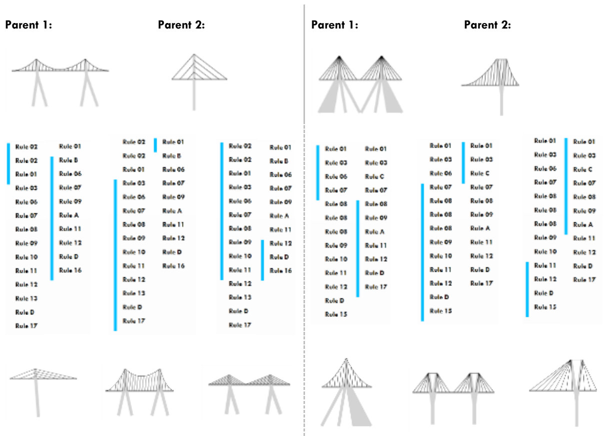

Fig. 9. Examples of crossover for two sets of parent designs.

5.1.2. Evolutionary framework and surrogate modeling

Using surrogate modeling to approximate the design space explored by the interactive evolutionary framework requires adding a step in the user experience between design problem setup and interactive exploration: the surrogate model building step. In this phase, the user must wait while training and testing sets of data points are generated and evaluated using full structural analysis to build the surrogate model. The tradeoff is that the user can then experience nearly instant results in the subsequent interactive exploration mode, due to the approximation. The importance of this development is that system response time during interactive exploration is decoupled from the complexity of the design problem and the size of the design space, eliminating impractical or prohibitive disruptions in the creative process caused by minutes- or hours-long waits between generations of candidate designs.

5.1.3. Structural grammars and surrogate modeling

The greatest of the three integration challenges is the application of surrogate modeling, which has been developed exclusively for parametric design spaces, to those which are rule-based. Surrogate modeling approaches build predictive systems that produce an output, given a vector of inputs. In standard surrogate modeling, the natural candidate for the input vector is the design vector. When applying surrogate modeling to grammar-generated designs, it is necessary to generate a reasonable input vector based on the design without directly using its rule sequence. Once the rule sequence is transformed into a constant-length design vector, it can be used to build surrogate models, and to predict performance of new designs using the surrogate. The transformed vector can be based on derived geometrical and structural properties based on the specifics of the grammar, such as dimensions of various elements. The vector can also include variables indicating whether certain rules were used and their parameter settings.

6. CONCLUSIONS

This paper has presented an overview of a new computational approach to creatively integrate structural considerations into conceptual design. The approach develops three individual strategies with original contributions: interactive evolutionary exploration with flexible problem definition and design refinement capabilities, trans-typological grammars yielding unexpected design alternatives across standard structural types, and surrogate modeling approximation adapted to focus on high-performing design space regions. In combination, these strategies comprise a promising new way for designers to use computation to quickly explore a wide range of exciting structural concepts at the start of the design process. Future work includes further development and study of this approach to improve the implementation and to test its effectiveness on more complex and realistic design problems.

7. REFERENCES

[1] Wang L., Shen W., Xie H., Neelamkavil J., and Pardasani A., *Collaborative conceptual design—state of the art and future trends*, Computer-Aided Design, Vol. 34, No. 13 (2002) pp. 981-996.

[2] von Buelow P., *Suitability of genetic based exploration in the creative design process*, Digital Creativity, Vol. 19, No. 1 (2008) pp. 51-61.

[3] Mueller C., Ochsendorf J., *An Interactive Evolutionary Framework for Structural Design*, 7th International Seminar of the IASS Structural Morphology Group, London (2011).

[4] Mueller C., *structureFIT*, www.caitlinmueller.com/structurefit (2012).

[5] The Happy Pontist, *Bridge competition debris part 12: River Wear*, http://happypontist.blogspot.com/2009/03/bridge-competition-debris-part-12-river.html (2012).

[6] Stiny G., and Gips J., *Shape Grammars and the Generative Specification of Painting and Sculpture*, Information Processing '71 (1972) pp. 1460-1465.

[7] Stiny G. *Shape: Talking about Seeing and Doing*. MIT Press (2006).

[8] Mitchell W. J., *Functional Grammars: An Introduction*, ACADIA Reality and Virtual Reality (1991) pp. 167-176.

[9] Geyer, P., *Multidisciplinary grammars supporting design optimization of building*, Research in Engineering Design, Vol. 18, No. 4 (2008) pp. 197-216.

[10] Forrester A., Sobester A., and Keane, A., *Engineering Design via Surrogate Modeling: A Practical Guide*, John Wiley & Sons (2008).

[11] Hastie T., Tibshirani R., and Friedman J. *Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.)*, Springer, New York (2009).