Integrating Interactive Evolutionary Exploration and Parametric Structural Design

by

Renaud Alexis Pierre Emile Danhaive

B.Sc. in Architecture and Engineering
Université Libre de Bruxelles, 2013

Submitted to the Department of Civil and Environmental Engineering in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Civil and Environmental Engineering

at the

Massachusetts Institute of Technology

June 2015

Signature of Author: Signature redacted
Department of Civil and Environmental Engineering
May 15, 2015

Certified By: Signature redacted
Caitlin T. Mueller
Assistant Professor of Architecture
Thesis Supervisor

Accepted By: Signature redacted
Heidi M. Nepf
Donald and Martha Harleman Professor of Civil and Environmental Engineering
Chair, Departmental Committee for Graduate Students

Integrating Interactive Evolutionary Exploration and Parametric Structural Design

by

Renaud Alexis Pierre Emile Danhaive

Submitted to the Department of Civil and Environmental Engineering on May 15, 2015 in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Civil and Environmental Engineering

Abstract

Current modeling and analysis tools are extremely powerful and allow one to generate and analyze virtually any structural shape. However, most of them do not allow designers to integrate structural performance as an objective during conceptual design. As structural performance is highly linked to architectural geometry, there is a need for computational strategies allowing for performance-oriented structural design in architecture.

In order to address these issues, this research combines interactive evolutionary optimization and parametric modeling to develop a new computational strategy for creative and high-performance conceptual structural design. Parametric modeling allows for quick exploration of complex geometries and can be combined with analysis and optimization algorithms for performance-driven design. However, this methodology often questions the designer's authorship as it is based on the use of black-box optimizers. On the other hand, interactive evolutionary optimization empowers the user by acknowledging his or her input as fundamental and includes it in the evolutionary optimization process. This approach aims at improving the structural performance of a concept without limiting the creative freedom of designers. Taking advantage of the two frameworks, this research implements an interactive evolutionary structural optimization framework in the widely used parametric modeling environment constituted by Rhinoceros and Grasshopper. Previous work has illustrated the benefits of combining parametric modeling and genetic algorithms for design space exploration. Comparatively, the implemented design tool capitalizes on Grasshopper's versatility for geometry generation but supplements the visual programming interface with a flexible portal increasing the designer's creative freedom through enhanced interactivity. The tool can accommodate a wide range of structural typologies and geometrical forms in an integrated environment.

This research offers a versatile, performance- and user-oriented environment for creative and efficient conceptual structural design.

Thesis Supervisor: Caitlin T. Mueller
Title: Assistant Professor of Architecture

Acknowledgements

I wish to thank Professor Caitlin Mueller for her invaluable support and insight throughout this academic year. I am grateful to my friends of the Master of Engineering program for making this year an unforgettable experience that went beyond my expectations.

Finally, I thank my parents for their love and for making this year possible.

Table of Contents

Chapter 1. Introduction 11

1.1. Architectural and Structural Design 12

1.2. Research Goals 12

1.3. Thesis Organization 13

Chapter 2. Literature Review 15

2.1. Optimization and Evolutionary Algorithms 15

2.2. Parametric Modeling and Optimization 16

2.2.1. Revit and Dynamo 17

2.2.2. Rhinoceros and Grasshopper 18

2.3. Current Tools for Interactive Evolutionary Optimization 19

2.3.1. Web-based interactive evolutionary optimization of 2D trusses: structureFIT 19

2.3.2. ParaGen 19

2.3.3. Dreamcatcher (Autodesk 2015) 20

2.4. Potential and Limitations of Existing Tools 20

Chapter 3. Interactive and Parametric Framework 23

3.1. Theory 24

3.1.1. Potential of Connecting Interactive Evolutionary Optimization and Parametric Design 24

3.1.2. Design Variables vs Model Parameters 24

3.1.3. Design Space vs Parametric Space 25

3.2. Implementation 26

3.2.1. Integration within Grasshopper 26

3.2.2. Interactive Evolutionary Optimization (IEO) Component 27

3.2.3. Versatility and Extensibility 27

3.2.4. Graphical User Interface and Features for Enhanced Interactivity 27

3.3. Summary 30

Chapter 4. Design Case Studies 31

4.1. Introduction 31

4.2. 11 Bar Truss (2 design variables) 32

4.3. Three-Dimensional Truss Canopy (18 design variables) 34

Chapter 5. Conclusion 37

5.1. Summary of Contributions 37

5.2. Future Work 38
5.3. Potential Impact and Concluding Remarks 39
Appendices 41
Grasshopper Script for 9 Bar Truss Case Study (part 1) 41
Grasshopper Script for 9 Bar Truss Case Study (part 2) 42
Grasshopper Script for Three-Dimensional Truss Case Study (part 1) 43
Grasshopper Script for Three-Dimensional Truss Case Study (part 2) 44
Grasshopper Script for Three-Dimensional Truss Case Study (part 3) 45
References 47

Figures

Figure 1:Projects successfully integrating architectural and structural design ... **12**
Figure 2: Standard Evolutionary Algorithm vs Interactive Evolutionary Algorithm – adapted from (Mueller and Ochsendorf 2015) ... **16**
Figure 3: Result of the evolutionary search performed on a 2D truss problem using Galapagos for Grasshopper ... **17**
Figure 4: Parametric design and optimization framework illustrated with widely used software and plug-ins ... **17**
Figure 5: Octopus 3D viewport representing the objective space (Vierlinger 2015) ... **18**
Figure 6: structureFIT exploration mode displays a large diversity of sub-optimal results found using evolutionary optimization and user selection (Mueller, structureFIT 2015) ... **19**
Figure 7: ParaGen workflow (Turrin, von Buelow and Stouffs 2011) ... **20**
Figure 8: stormcloud logo ... **23**
Figure 9: Parametric modeling and interactive evolutionary algorithm ... **24**
Figure 10: Distinction between variables and parameters: only parameters that are changed during the interactive evolutionary optimization are called design variables ... **25**
Figure 11: IEO component ... **27**
Figure 12: User interface ... **28**
Figure 13: Main viewport and individual 3d thumbnails are synchronized ... **29**
Figure 14: Geometry generation in Rhino ... **29**
Figure 15: Parametric Model and corresponding parameters and variables ... **32**
Figure 16: Results of evolutionary exploration, x_1 vs score (a), x_2 vs score (b), colormap of the design space in plan view (c), 3d visualization of the design space (results are capped to designs achieving a score lower than 5.00) ... **33**
Figure 17: design case study exploration: the best score is found in one generation and the exploration yields two fundamentally different near-optimal results ... **33**
Figure 18: Earth Canopy, designed by Feilden Clegg and Atelier One (Atelier One 2015) ... **34**
Figure 19: Design parameters (including design variables) ... **34**
Figure 20: Space truss generation process ... **35**
Figure 21: Initial geometry and corresponding loading: perspective (a), side (b), and front (c) views ... **35**
Figure 22: Design exploration using stormcloud: diverse high-performing design options are found ... **36**
Figure 23: 4 iso-performing designs found using stormcloud (plan, side, front and perspective views) ... **36**
Figure 24:Integrating constructability for conceptual design of tall buildings (Horn 2015) ... **38**
Figure 25: Surrogate modeling (Mueller and Ochsendorf 2013) ... **39**

Chapter 1. Introduction

Recent years have seen the development of increasingly powerful modeling and analysis tools that allow architects and engineers to generate and analyze virtually any structural shape. Both generating complex geometries and performing advanced analyses now requires less manual effort and computational time than ever before. However, those tools have generally failed to integrate architectural and structural design. Specifically, existing tools epitomize a design paradigm that entirely excludes structural considerations from the earliest stages of architectural design. Since the performance of a structure is highly attributed to its geometry, such an approach will likely result in poorly performing structures.

Several classes of experimental tools and techniques have recently emerged to address these challenges: Graphic Statics, Thrust Network Analysis, Form-Finding, Integrated Finite Element Analysis, and other related approaches. In order to capitalize on the opportunities offered by computational power while overcoming the issues previously highlighted, these structural design tools have been developed to integrate structural performance as an objective during conceptual design. They are very powerful for specific structures but fail to define an integrated framework for creative conceptual structural design. Mueller (2013) identifies two key features of those tools, namely feedback and guidance. The former offers live quantitative feedback, such as stiffness or mass, allowing users to understand the influence of design choices on the structural performance but is unfortunately limited by the computational power required to perform structural analysis. Guidance-based tools generate more performant geometries on the basis of a design concept. The fundamental benefit of this approach lies in the fact that it aims at improving structural performance of a concept without limiting the creative freedom of designers. In the context of this thesis, two performance-oriented design methodologies are particularly relevant: the parametric design and optimization framework and the interactive evolutionary framework.

1.1. Architectural and Structural Design

Traditionally, in architecture, the conceptual design phase addresses aesthetic and functional objectives while structural design truly starts when the architectural geometry has been determined. This form is often defined based on aesthetic choices, contextual relationships and functional requirements. However, geometry plays a key role not only for a project's architectural success but also for its structural performance.

Integrating structural design during the conceptual design phase has the potential to reconcile structural form and architectural geometry, resulting in efficient and architecturally expressive forms. Many built examples, often designed by teams of architects and engineers (see Figure 1), successfully integrate architectural and structural design, proving that such a design approach can steer creativity and innovation.

Zarzuela Grandstand by Eduardo Torroja.
Image from Flickr user Ximo Michavila

Munich Olympic Stadium by Frei Otto.
Image from Wikipedia

Pedestrian Bridge by Pelli Clarke Pelli Architects and Guy Nordenson and Associates
Image from www.pcparch.com

Figure 1:Projects successfully integrating architectural and structural design

Structural elegance is widely accepted as a timeless principle of architecture and a thoughtful integration of structural features is far more likely to be achieved if considered early on in the design process. In that perspective, a design approach that emphasizes the importance of structural design can benefit a project's performance and greatly contribute to its architectural success. New computational means can help explore design options in a way that integrates structural performance in the conceptual design phase without inhibiting the designer's creative freedom.

This thesis seeks to define a framework that improves and extends existing computational design strategies to help designers integrate structural performance in the conceptual design phase.

1.2. Research Goals

This thesis seeks to integrate parametric design and interactive evolutionary optimization in order to truly empower designers thanks to the interactivity offered by the latter and the versatility of the former. The research goals cover both theoretical aims and practical details, expressed as follows:

(i) Define a framework combining interactive evolutionary optimization and parametric modeling
(ii) Implement the theoretical framework in a tool that facilitate design space exploration in an existing parametric modeling environment, Rhino and Grasshopper (Robert McNeel & Associates 2015),by means of enhanced user interactivity.

1.3. Thesis Organization

This thesis is divided into five chapters: Introduction, Literature Review, Interactive and Parametric Framework, Design Case Studies and Conclusion.

Chapter 2 focuses on existing design space exploration and optimization tools and strategies for structural design and their current limitations.

Chapter 3 expands on the computational strategy proposed in this thesis and presents its implementation, *stormcloud*, for Rhino/Grasshopper.

Chapter 4 presents design case studies to illustrate and discuss the benefits of the method developed for this research.

Chapter 5 offers a summary of the contributions and presents future perspectives for this work.

Chapter 2. Literature Review

Recent research in computational design has demonstrated the need for new computational strategies and tools for creative structural design and optimization (Mueller 2014). After comprehensive reviews of existing tools for conceptual structural design, two key features are extracted in the literature: feedback and guidance. It is then argued that most tools implement feedback features but lack guidance. Guidance-based must allow for performance-oriented, diverse and interactive exploration.

The objective of this section is not to give a detailed overview of all the existing tools available for conceptual structural design, but to focus instead on those which pave the way for the development of more interactive and holistic tools for performance-oriented architectural and structural design. For this purpose, two main strategies are identified: interactive evolutionary optimization and parametric modeling.

2.1. Optimization and Evolutionary Algorithms

Evolutionary algorithms constitute a subclass of heuristic search methods. The tools presented in this chapter rely heavily on evolutionary algorithms to perform optimization tasks. In fact, the following tools all use variations of genetic algorithms (GAs). GAs in particular are inspired by Darwinian evolutionary principles and implement them as ways to generate better-performing solutions. Such algorithms are very versatile and well suited to incorporate user interactivity. Based on stochastic processes, they are not guaranteed to find the optimum solution and they work with populations of solutions rather than with a unique solution. They are easily extensible as they only need an objective (fitness) function and coded versions of parameters. Thus, they can theoretically be applied to any problem where a performance parameter can be well-defined. These characteristics explain the GA's robustness (Filomeno Coelho, et al. 2014) as well as their popularity for structural optimization.

Standard GA's operate on the problem using three 'genetic' operators to generate new populations of candidate solutions based on previous populations as follows (Filomeno Coelho, et al. 2014) (see Figure 2):

(i) Selection: best solutions are selected based on their performance or fitness
(ii) Mutation: new candidate solutions are randomly mutated
(iii) Crossover/reproduction: parent solutions are selected and their characteristics (associated parameters) are recombined to create new candidate solutions

On the one hand, standard evolutionary algorithms seek to find the optimal solution of a given optimization using heuristic methods in a closed loop workflow (see Figure 2). On the other hand, an interactive evolutionary algorithm incorporates the designer's input in the optimization workflow for the selection of parent solutions (see Figure 2). This approach accounts for ill-defined objectives, such as aesthetics, which make it very suitable for applications in architectural and structural design where design complexity goes beyond quantifiable metrics. Moreover, it can capitalize on the designer's intuition to interact with evolutionary parameters and steer design space exploration.

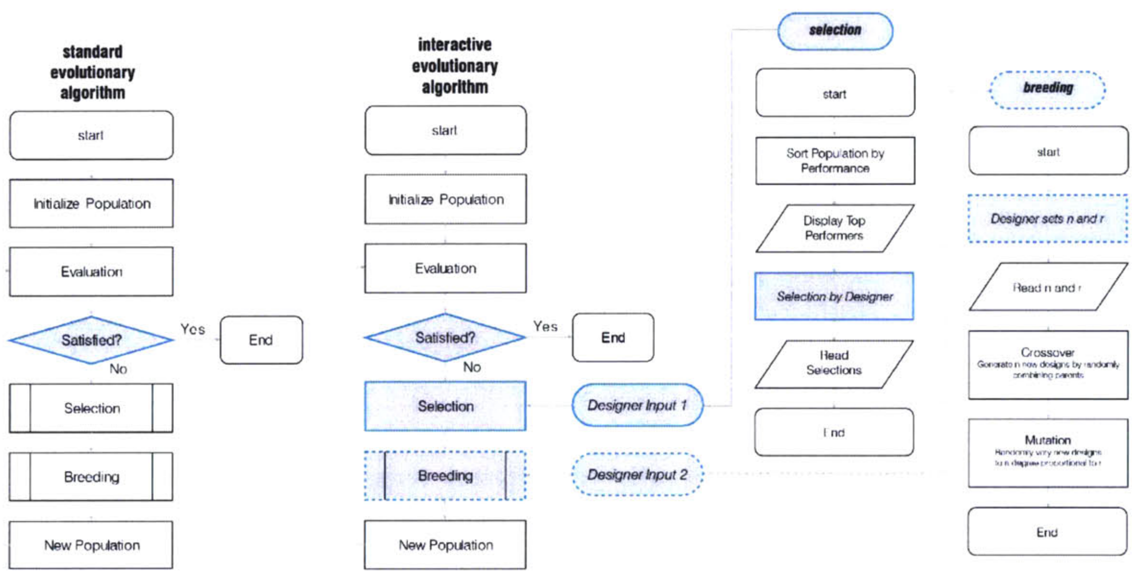


Figure 2: Standard Evolutionary Algorithm vs Interactive Evolutionary Algorithm – adapted from (Mueller and Ochsendorf 2015)

2.2. Parametric Modeling and Optimization

The parametric modeling and optimization (PMO) framework is here defined as the computational methodology for conceptual structural design that has organically emerged from the development of inter-related tools. Specifically, 3D modeling software, such as Rhinoceros or Revit (Autodesk 2015), can be combined with visual programming interfaces, supplementing the modeling workspace to constitute parametric modeling environments. These allow the user to script complex generative algorithms without prior programming knowledge and can help steering design space exploration. Exploring different solutions can be done in a timely manner as the parametric design process is by essence non-destructive, meaning that one model contains all the previously explored solutions as well as the ones yet to evaluate. Furthermore, these parametric modeling environments can be used in combination with analysis and optimization components to constitute an integrated design environment. Thus, such environments are not solely dedicated to computer-aided drawing (CAD) but benefits from the numerous available plug-ins to assess the performance of architectural designs according to a wide range of criteria, ranging from building envelope performance to daylighting availability. This methodology is increasingly used by both engineers and architects. Such integrated environments constitute a compelling common grounds for the two professions. However, little to no control on the optimization process is left to the user and parametric models are optimized using black-box solvers, yielding a single near-optimal solution. As a result, parametric modeling as implemented in existing environments lacks of guidance features and design space exploration usually remains limited to manual manipulation of sliders and initiation of computational search. Automated optimization procedures fail to take advantage of the designer's expertise (Scott, Lesh and Klau 2002) and do not capture the complexity of design which include unquantifiable criteria. These may lead to sub-optimal solutions which are more valuable to the designer in terms of ill-defined criteria such as aesthetics.

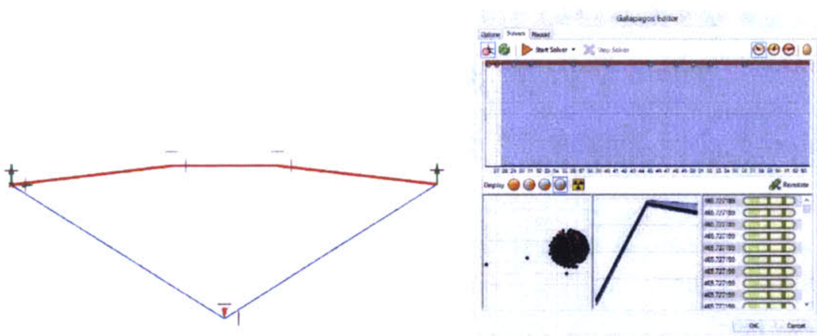


Figure 3: Result of the evolutionary search performed on a 2D truss problem using Galapagos for Grasshopper

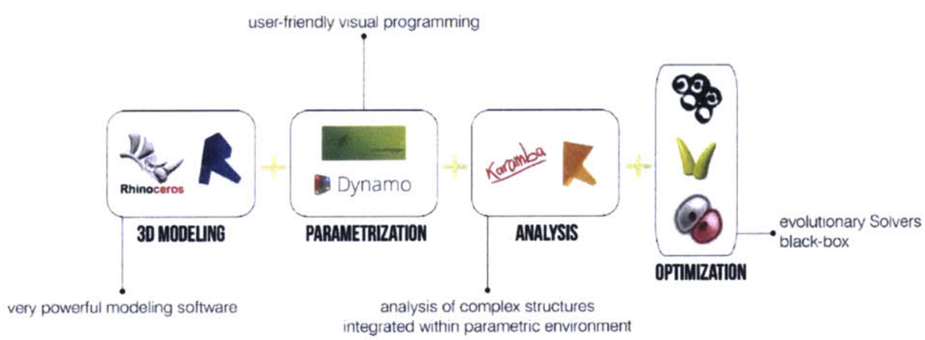


Figure 4: Parametric design and optimization framework illustrated with widely used software and plug-ins

2.2.1. Revit and Dynamo

Autodesk is currently developing an open-source visual programming interface for Revit, a widely used Building Information Modeling (BIM) software tool for architectural and engineering design. Based on the first released versions of Dynamo (Autodesk 2015), its main advantages are: integration in a BIM environment, coding for design made easier (compared to equivalent tools). It also includes an optimization plug-in, Optimo (Rahmani 2015), which is a multi-objective optimization tool implementing an NSGA-II.

2.2.2. *Rhinoceros and Grasshopper*

Rhinoceros is a NURBS-based 3D modeling software, widely used in academia and by engineering and architecture offices. Grasshopper is an algorithmic modeling plug-in for Rhinoceros.

"Grasshopper is a graphical algorithm editor tightly integrated with Rhino's 3D modeling tools. Unlike RhinoScript, Grasshopper requires no knowledge of programming or scripting, but still allows designers to build form generators [...]" (Robert McNeel and Associates 2015)

It extends the modeling capacity of Rhinoceros by allowing the user to define geometries parametrically using built-in components. Those geometries can be previewed in the Rhinoceros user interface and eventually 'baked' as actual 3D objects.

Galapagos

Galapagos is an evolutionary solver included with Grasshopper. Although it extends Grasshopper GUI with an optimization window (see Figure 3), it does not include the designer's input during the evolutionary search.

Goat

Goat (Rechenraum 2015) uses Galapagos UI to interface the NLopt library (Johnson 2015) in Grasshopper. Its main advantage lies in its use of rigorous and widely tested optimization algorithms. It also allows the user to select among several optimization algorithms.

Octopus

Octopus (Vierlinger 2015) is a plug-in for Grasshopper that allows for multi-objective optimization (MOO) and can represent the objective space for up to 3 objectives simultaneously using a 3D viewport. Subjective evaluation is also made possible by the optional display of geometries in place of their corresponding points in the objective space. However, Octopus is computationally expensive and, while it offers promising visualization and MOO features, is limited in terms of user interactivity and design freedom.

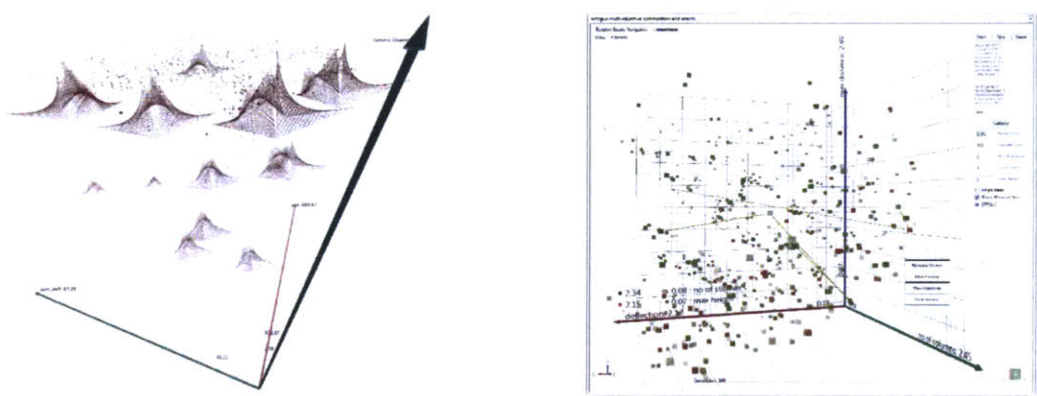


Figure 5: Octopus 3D viewport representing the objective space (Vierlinger 2015)

TT Toolbox

The TT toolbox, developed by Thornton Tomasetti's Core Studio (2015), offers a brute force sampling component for Grasshopper (and Dynamo). Stavros Tseranidis (2015) has implemented more efficient sampling algorithms (grid, random uniform and latin hyper cube) using GhPython (Robert McNeel & Associates 2015).

In summary, it is clear that there are many promising environments and plug-ins for PMO. While no currently available tools directly address the need for performance-based guidance in conceptual design, a fertile environment exists where new tools could capitalize on the many advantages of parametric design approaches.

2.3. Current Tools for Interactive Evolutionary Optimization

As discussed in Section 2.1, interactive evolutionary optimization is an optimization strategy that incorporates the user input in the evolutionary optimization process. Interactive evolutionary algorithms take advantage of the user input to steer design space exploration. Interactive evolutionary optimization usually supposes the parametric definition of a problem as a starting point of the exploration cycle. A distinction between parametric and design space will be made in Chapter 3. Interactive evolutionary optimization, introduced mainly for pure subjective selection (Herdy 1996), lately has gained popularity for optimization in structural design. In the following section, three of the most relevant examples of interactive optimization for structural design are presented.

2.3.1. Web-based interactive evolutionary optimization of 2D trusses: structureFIT

structureFIT is a web-based tool for interactive evolutionary optimization of 2D trusses. It was developed as a proof-of-concept tool by Mueller (2014) to illustrate the potential of an interactive evolutionary framework for structural design. The tool is extremely user-friendly and uses a very effective user interface. Moreover, it increases user interactivity by allowing the user to interact with evolutionary parameters. Specifically, the user can set the evolutionary parameters of mutation rate and generation size. Mueller and Ochsendorf (2015) show that this increased user interactivity helps to steer design space exploration in diverse ways.

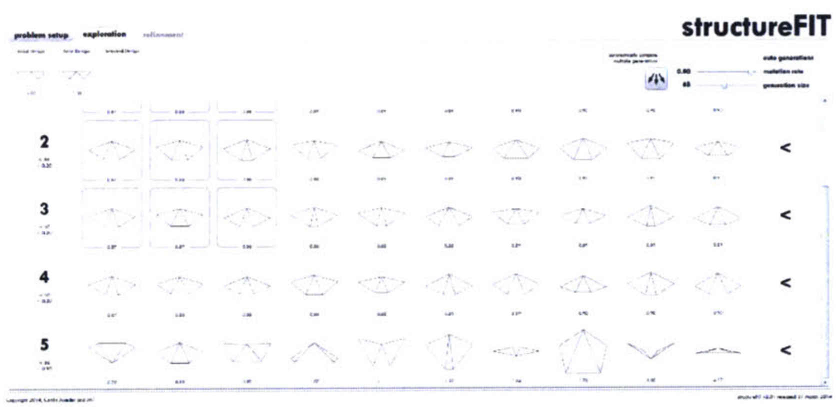


Figure 6: structureFIT exploration mode displays a large diversity of sub-optimal results found using evolutionary optimization and user selection (Mueller, structureFIT 2015)

2.3.2. ParaGen

ParaGen (von Buelow, Paragen: Performative Exploration of Generative Systems 2012) is a tool which effectively combines parametric modeling, GA's, and user input (see Figure 7). Very powerful and flexible, it can deal with different types of problems on complex structure. Von Buelow had already presented earlier iterations of the framework implemented in ParaGen, specifically for truss design (von Buelow 2008), but this tool goes beyond these first attempts in his versatility and ability to work with complex 3D geometries and analyses. ParaGen clearly illustrates the benefits of combining interactive optimization and parametric modeling. However; it lacks user-friendliness, integration, and user interactivity beyond selection to reach its full potential.

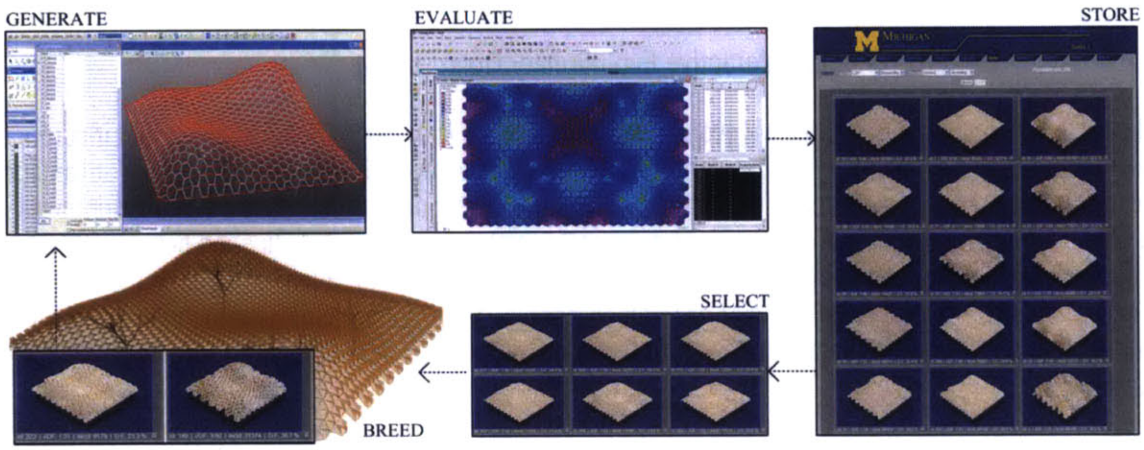


Figure 7: ParaGen workflow (Turrin, von Buelow and Stouffs 2011)

2.3.3. Dreamcatcher (Autodesk 2015)

The Dreamcatcher project an advanced research project underway at Autodesk, with the aim of integrating interactive evolutionary optimization with structural and mechanical design. It uses artificial intelligence and cloud computing principles to generate new populations of designs. While it appears very promising, no official version is currently available to the public for testing.

In summary, while all of three these tools are promising, none are able to fully connect interactive evolutionary exploration with existing parametric modeling environments and plug-ins. These limitations are discussed in the following section.

2.4. Potential and Limitations of Existing Tools

This section has provided an overview of the existing tools for design space exploration for conceptual structural design and drawn conclusions about their potential and limitations. Their potential lies in that they that seek to include the designer in the optimization process. For this purpose, three main aspects make a tool successful: (i) user-friendliness, (ii) algorithms effectiveness and (iii) versatility. structureFIT and ParaGen both succeed in implementing efficient evolutionary algorithms for exploration and, in general, genetic algorithms have proven their usefulness for structural optimization. structureFIT is particularly well designed in terms of user interactivity and for subjective selection of designs. As for versatility, ParaGen shows the most potential for exploring complex problems going beyond structural design. Their main limitation is their lack of integration in environments that allow for versatile parametric formulations and that are already widely used by architects and engineers.

While the tools discussed previously implement very promising computational strategies for conceptual design and constitute proof-of-concepts for their framework, they will not reach their full potential and impact unless providing simultaneously user-friendliness, versatility and extensibility. There is a need for environments that integrate guidance-based exploration of the design space for any problem with user friendliness and versatility.

In addition, the tools that were covered earlier in this chapter are stand-alone and, while it can be argued that implementing such tools as stand-alone software frees the user from having to use commercial software, the approach pursued in this research seeks to overcome the issues that have prevented a more widespread use of optimization in general, and interactive optimization in particular, in practice and in education. Indeed, implementing tools as plug-ins in widely environments that promote the use of third-party plug-ins, often free, can expand the use of new computational ideas considerably. Integrating an interactive evolutionary optimization framework within a visual programming environment that is supported by a strong collaborative community of programmers and designers enables capitalizing on the full potential of the computational strategy developed in this thesis. The following chapter will introduce a new design tool that implements this idea.

Chapter 3. Interactive and Parametric Framework

Previous research has illustrated the benefits of combining parametric modeling and genetic algorithms for design space exploration. Based upon the conclusions made about the existing tools for creative and performance-oriented structural design during the conceptual phase, a computational strategy combining parametric modeling and interactive evolutionary optimization within an a versatile modeling environment is proposed in this chapter. This section defines the conceptual framework developed in this thesis and presents its implementation in Rhino/Grasshopper, *stormcloud* (see Figure 8), with a focus on user interactivity.

Figure 8: stormcloud logo

3.1. Theory

3.1.1. *Potential of Connecting Interactive Evolutionary Optimization and Parametric Design*

An interactive evolutionary optimization tool inherently relies on parametric design. Any problem that can be optimized must be parametrized. In other words, there must be at least one design variable that may change and affect the objective function for the solution not to be trivial. In that perspective, any optimization algorithm is connected to parametric design as it will always act on the parametric model of a problem. For example, structureFIT is effectively a tool that combines parametric modeling, evolutionary optimization and human input. In the following section, however, a new approach is presented that seeks to connect interactive evolutionary optimization to parametric design in such a way that all the features offered by parametric design in terms of geometry generation can be used to explore innovative structural systems. Consequently, interactive evolutionary optimization must not be limited to a set of structural typologies or design problems. Instead, it should build upon parametric modeling environments versatility and built-in features. Doing so, the designer's input goes beyond selection of solution and manipulation of evolutionary parameters as he or she is in charge of the setup of the parametric model with total freedom. Interactive evolutionary optimization must thus be connected to parametric design in the most lightweight way possible and should only know about the geometry, the performance index and the design variables of a given problem, thus allowing the designer to explore any problem easily and interactively. Connecting evolutionary optimization and parametric design inherently extends the designer's input by not limiting it to the modification of predefined parametric formulations of design problems.

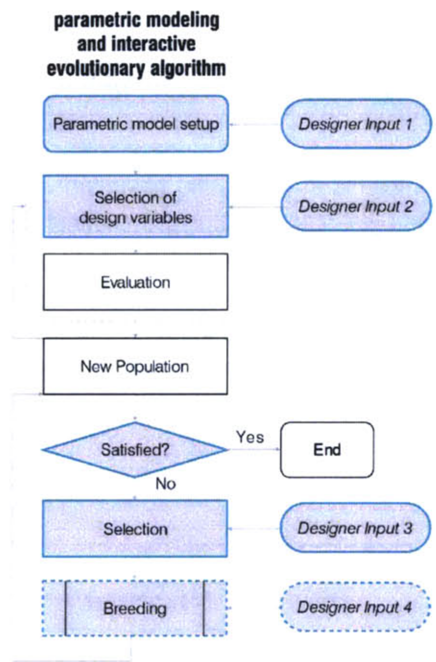


Figure 9: Parametric modeling and interactive evolutionary algorithm

3.1.2. *Design Variables vs Model Parameters*

As mentioned above, optimization acts on parametric models. However, it does not necessarily act on every parameter of the problem. This shows that a distinction needs to be made between a parametric space and a design space, the latter being the space explored by the interactive evolutionary optimization process. This distinction is established in this section and in 3.1.3 below.

In a parametric modeling environment, any data locally (e.g. number) or globally (e.g. Rhino geometry) stored is a parameter. A distinction must thus be made between model parameters and design variables. The model parameters are the different objects that are used to model a geometry but that are not necessarily changed during the interactive evolutionary exploration of the design space. The design variables are the model parameters that may be changed by the evolutionary engine during the exploration process. Hence, according to this definition, all design variables are parameters but not all parameters are variables (see Figure 10). The more parametrized a design problem, the larger the design space. Thus, setting up a model parametrically from the outset will allow the designer to explore more easily the design space in the future and to consider the effect of model parameters that were not deemed important in the earliest design steps. A parametric and interactive environment allows to change the dimension of the design space by including more model parameters as design variables.

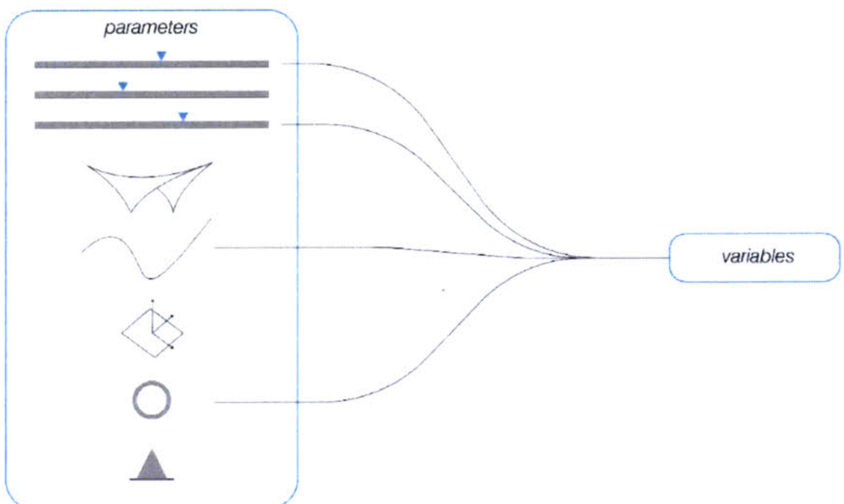


Figure 10: Distinction between variables and parameters: only parameters that are changed during the interactive evolutionary optimization are called design variables

3.1.3. Design Space vs Parametric Space

As stated above, only the design space is explored by optimization algorithms and, since the design variables are selected among parameters defining the parametric space describing a design problem, the design space is included in the parametric space.

In the parametric space, a design is defined by parameters that can be potentially be included as design variables and change the design space. In other words, the parametric space set up with *n* parameters contains

$$\sum_{i=0}^{n} C(n,i) = \sum_{i=0}^{n} \frac{n!}{(n-i)!\,i!} = 2^n$$

different design spaces as a designer can include 0 up to *n* parameters as design variables and for each number *i* of parameters included as such there are

$$C(n, i) = \frac{n!}{(n-i)!\,i!}$$

different design spaces possible. When creating a parametric model of a problem, a designer thus effectively defines a variety of potential design spaces by including or excluding parameters as design variables. The ability of designers to understand the parametric space has been proven by the increasing use of parametric modeling environments for design. Connecting parametric design and interactive evolutionary exploration offers unprecedented opportunities for design exploration. On the one hand, interactive evolutionary optimization expands parametric design by means of enhanced exploration features. On the other hand, parametric design provides interactive evolutionary optimization with versatility and extensibility. (Microsoft 2015)

3.2. Implementation

The new tool – *stormcloud* - is written in C#/.NET (Microsoft 2015) and uses Windows Presentation Foundation (Microsoft 2015) for tits user interface. Helix 3D Toolkit (Bjorke 2015) is used for the 3D visualizations of generated structures. Since it is integrated within Grasshopper as a plug-in, it is compiled as a .gha file. The following presents the features implemented to meet the goals set for the framework conceptually defined above with a focus on the user experience.

3.2.1. Integration within Grasshopper

The tool is integrated within Grasshopper as a component placed on the canvas. Creating a component for interactive optimization in Grasshopper presents three main challenges. These challenges are amplified by the lack of information in the Grasshopper software development kit, which only provides information on how to code for basic custom components. The following section presents these challenges along with ways to overcome them.

(i) Call the user interface by component events

There are many possible strategies to open a window from a GH component. For *stormcloud*, it was chosen to override the double click event of the GH_Component class to make it open the *stormcloud* window.

(ii) Connect the user interface to Grasshopper

In order to connect the user interface to Grasshopper, the opened window must know about the component that opened it in order to perform change to its input, specifically design variables. Since the window is opened from within the component, the component itself can easily be passed as an argument to the window, which solves the problem.

(iii) Iterate in Grasshopper

The biggest of the three challenges, which arises when one tries to iterate in Grasshopper by changing a slider, lies in the structure of a Grasshopper solution. Indeed, any change of an object will expire all its downstream objects and thus will expire the component changing the object initially, resulting in obvious problems. A way to solve this problem is to store the collected data in the view model of the opened window, which does not get expired by the solution, and to request a new solution after all parameter changes are executed.

3.2.2. *Interactive Evolutionary Optimization (IEO) Component*

The tool is implemented as a single component placed on the Grasshopper canvas (see Figure 11). Double clicking on the component raises an event that opens a WPF window. The component takes three different input – geometry (in the form of lines), score, and design variables - and has no output parameter. The score is normalized according the initial solution score. The code-behind of the WPF window generates populations of candidate solutions by re-computing the Grasshopper script solution after setting the design variables, i.e. sliders, to new values obtained after cross-over and mutation operations. The geometry, values of design variables and score of each solution state are collected by the component at every iteration and are stored in the data context of the window.

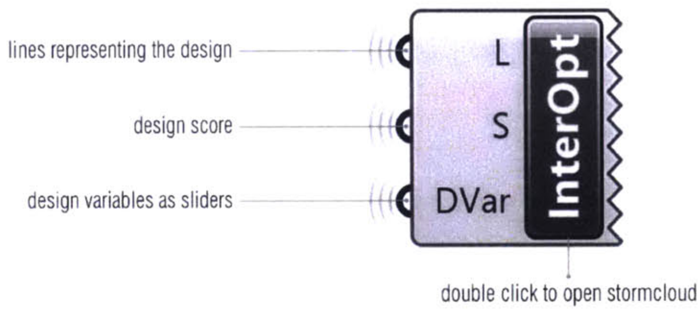


Figure 11: IEO component

3.2.3. *Versatility and Extensibility*

Since the GUI supplements Rhino/Grasshopper, it takes advantage of the versatility offered by the parametric modeling environment both in terms of analysis and geometry generation. Since the IEO component only knows about geometry, score and design variables, the WPF application is extensible to any problem. In other words, the component is blind to the nature of the parametrized model. While this thesis illustrates the benefits of *stormcloud* with structural design examples, it can be used for any design problem for which a fitness function can be set. It can also be used for pure subjective exploration by using a constant fitness function. A surrogate model could be connected to the component, considerably increasing evaluation speed and allowing for the exploration of computationally expansive problems.

3.2.4. *Graphical User Interface and Features for Enhanced Interactivity*

The graphical user interface (GUI) (see Figure 12) is extensively inspired by the exploration mode of structureFIT (Mueller, structureFIT 2015) and uses most of its strategies to enhance user interactivity while implementing new features. The GUI is designed to be as clear and simple as possible with an emphasis of design visualization and selection

Furthermore, in order to provide the user with an interface that promotes undisturbed guidance-based and creative explorations, the user interface includes features for enhanced interactivity which seek to facilitate generation and comparison of solutions.

structureFIT features (Mueller, structureFIT 2015)

It was already mentioned in Chapter 2 that structureFIT already implements many strategies for enhanced interactivity, including interaction with evolutionary parameters and subjective selection of designs. Those strategies are implemented in *stormcloud* to overcome the limitations of existing optimization components and they include:

(i) Interaction with Evolutionary Parameters (mutation rate and population size)
(ii) Return to previous generation
(iii) Selection of parents for next population generation

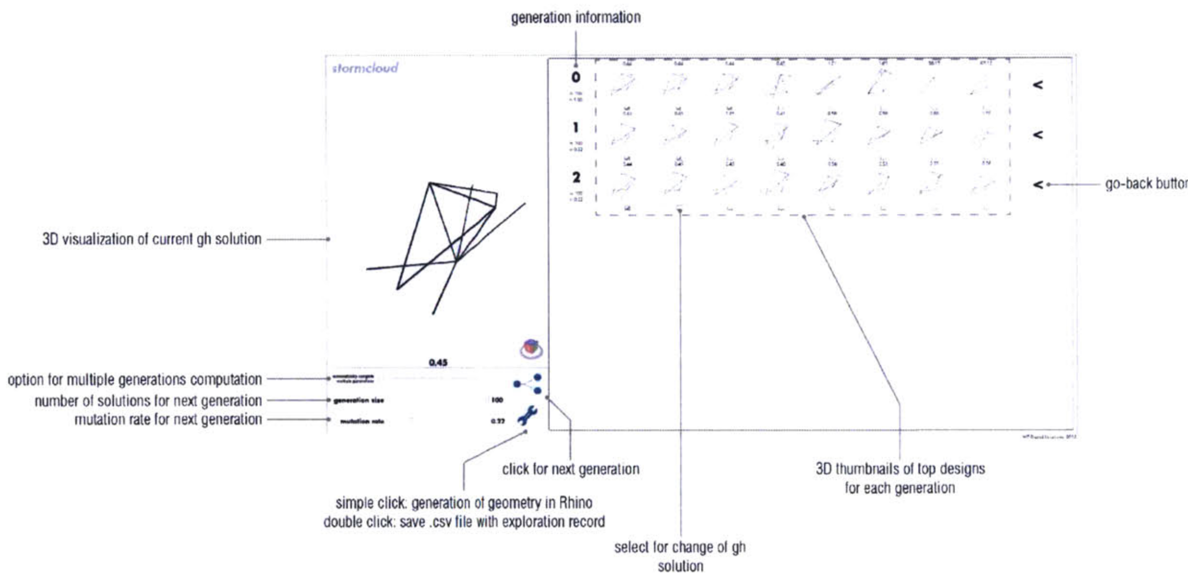


Figure 12: User interface

Grasshopper canvas

Compared to interactive evolutionary optimization tools previously implemented, *stormcloud* capitalizes on the possibilities offered by the Grasshopper GUI in terms of user interactivity. This means that parametrization features that exist in Grasshopper and that have proven to be effective are available *de facto*. This holds true for any visual programming interface, Dynamo for example, within which the framework can be implemented. This also means that *stormcloud* is not bound to any predefined parametric formulation (e.g. trusses) and can be used on a variety of design problems.

Synchronized 3D viewports

3D structures are hard to visualize without dynamic 3D navigation. Static thumbnails of structures are thus replaced by 3D viewports whose cameras are synchronized for better comparison of candidate solutions. Enforcing a camera binding between the viewports allows the user to easily and effectively compare the different solutions for subjective selection. Without this feature, the selection process would be tedious and ineffective. The synchronization is necessary for the fair comparison of the candidate solutions in terms of subjective features. Simultaneously visualizing the different designs dramatically increase the design space exploration capabilities of the parametric environment.

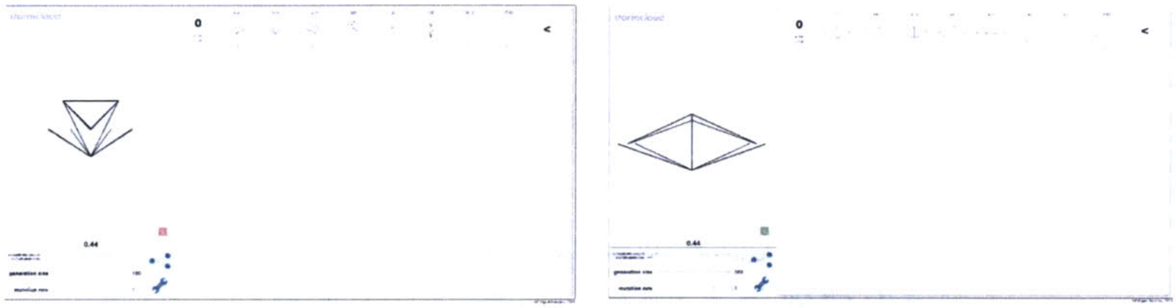


Figure 13: Main viewport and individual 3d thumbnails are synchronized

Selection of design for detailed visualization

Each candidate solution displayed on the design grid can be selected for detailed visualization on the main viewport. This allows the user to assess more easily subjective features of the top designs. Selecting a design for detailed visualization also change the solution state in Grasshopper to correspond with the selected solution, thus allowing for geometry generation in Rhino.

Generation of geometry in Rhino viewport

The user can save the preferred solutions as geometries stored in Rhino. This feature is equivalent to the 'Bake' feature existing in Grasshopper but improves it by making it more accessible through a simple button click. Each solution is assigned a different sublayer of a common 'exploration' layer.

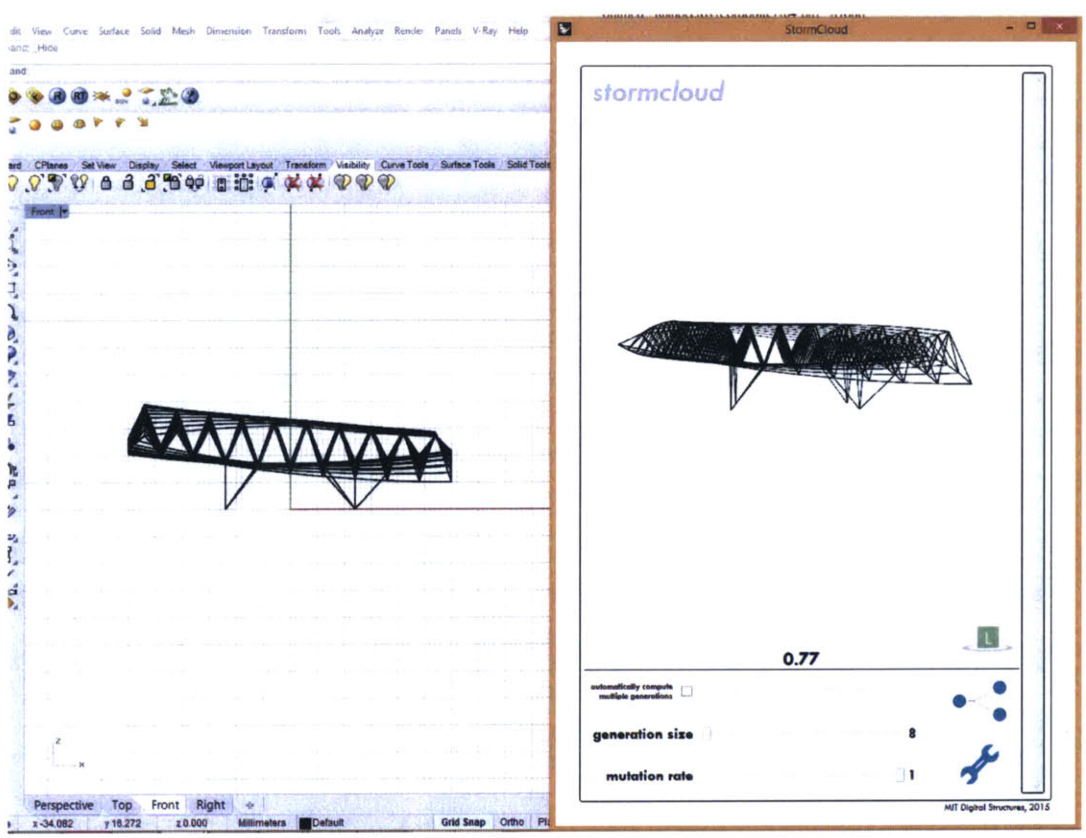


Figure 14: Geometry generation in Rhino

Recording of exploration

The designer is offered the possibility to record his exploration by saving the characteristics of each solution explored, i.e. the values of the design variables and the score, as a comma-separated values (.csv) text file. The file can then be used by the user to extract more knowledge about the design space as it is done for the 11 bar truss design in 4.2.

Exploration visualization

As with Galapagos, geometries that are computed in Grasshopper can be visualized live – changing in real-time – in the Rhino viewport (see Figure 14), which provides additional information to the designer about the possible structural forms that are being explored.

3.3. Summary

This chapter has introduced a new theoretical framework and implemented tool for linking interactive evolutionary exploration with rich, flexible parametric design models. The implemented tool works in the Rhino/Grasshopper environment, but could also be implemented in other environments, such as Revit/Dynamo, in the future. To demonstrate the use of this new approach, the following chapter presents several case studies for conceptual design problems.

Chapter 4. Design Case Studies

4.1. Introduction

This chapter illustrates the applications and benefits of *stormcloud* through several conceptual design case studies. While these examples are chosen in the realm of structural design, *stormcloud* can be used for any discipline in architectural design.

In both case studies, the objective function minimizes volume as follows.

Minimize

$$W = \sum_i A_i L_i$$

$$with\ A_i = \begin{cases} \min(A_{i,allowable\ stress}, A_{i,buckling})\ if\ F_i \geq 0 \\ A_{i,allowable\ stress}\ else \end{cases}$$

The structures are analyzed assuming perfect pin-connections and sized to limit stresses below the material allowable stress and prevent element buckling. In the following, the structural elements are circular rods and element buckling is modeled using Euler's formula. Hence,

$$A_{i,allowable\ stress} = \left|\frac{F_i}{\sigma_y}\right|\ and\ A_{i,buckling} = 2L\sqrt{\frac{F}{E}}$$

The performance index, as measured by W, is normalized according to the initial design.

4.2. 11 Bar Truss (2 design variables)

In order to clearly illustrate the benefits of the framework developed in this thesis, a simply supported 11 bar truss example is developed (see Appendices for the Grasshopper script). The parametric model includes 5 parameters and 2 of them are design variables, namely the maximum width and height of the 3D truss. The truss spans 10 m and is subjected to vertical loads of 10 kN applied on all of its unsupported nodes and horizontal loads of 5 kN are applied on two nodes, as shown in Figure 15. This simple design problem is used to discuss the benefits of the interactive and parametric framework and its implementation as it is very convenient to visualize how the design space was explored for 2 variable design space.

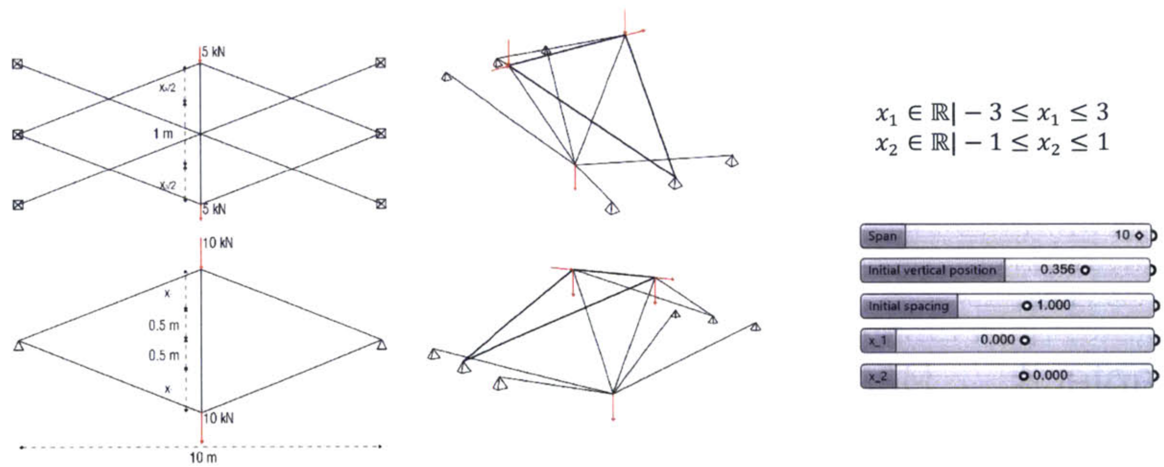


Figure 15: Parametric Model and corresponding parameters and variables

7000 data points were generated using *stormcloud* and the 'automatically compute multiple generations option' 7 times with a mutation rate of 1.00. Hence, the sampling is not a regular grid sampling but is denser near the global optimum (see Figure 16 (c)). Although the problem is a priori mundane, the design space has two local optima and asymptotic behaviors for x_1 approaching -0.5 and for x_2 approaching -1. In both cases, these asymptotic behaviors correspond to flat unstable configurations which cannot carry either vertical or lateral loads. It is interesting to notice that, except near the aforementioned asymptotic regions, the designs space is relatively flat, meaning the designer is free to choose a non-optimal design without losing much performance. Finally, the design space exhibits two bowls whose minima approximately have the same score. Thus, the design can be mirrored around the horizontal plane passing through the supports and attain the same performance (see Figure 17).

This case study illustrates the use of interactive evolutionary optimization for both exploration of the parametric formulation of a structural design problem and its subsequent optimization. While optimization algorithms as currently implemented in parametric design environments would have yielded a single near-optimal solutions the exploration yielded a range of sub-optimal designs saving up to 56% of material (see Figure 17). The denser sampling in the region of the global optimum also shows the effectiveness of the evolutionary search, even with a very high-mutation rate. In conclusion, the tool balances exploration and optimization in a guidance-based approach that is extremely relevant in the context of architectural design where many ill-defined goals must be met by the selected design. Examining the design space gives a lot of information about the importance and the consequences of different design choices. Although the data points were generated automatically using *stormcloud*, the generation of the plots above was done manually and their potential for design space exploration illustrates the future need of including more advanced design space visualization techniques as part of *stormcloud*.

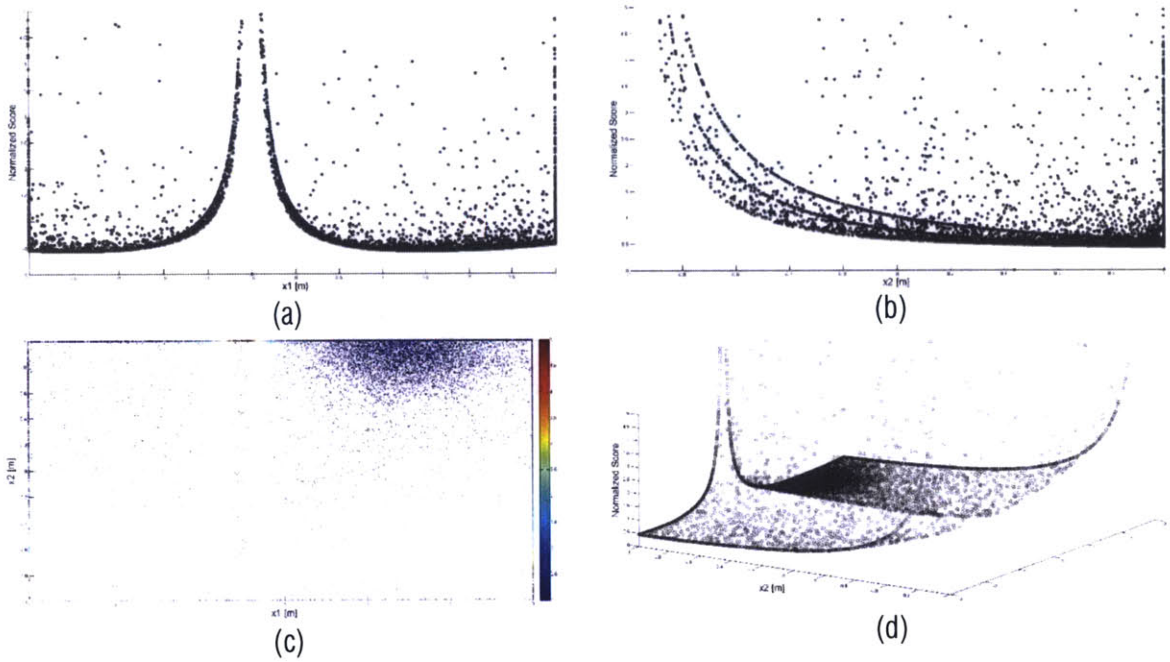


Figure 16: Results of evolutionary exploration, x_1 vs score (a), x_2 vs score (b), colormap of the design space in plan view (c), 3d visualization of the design space (results are capped to designs achieving a score lower than 5.00)

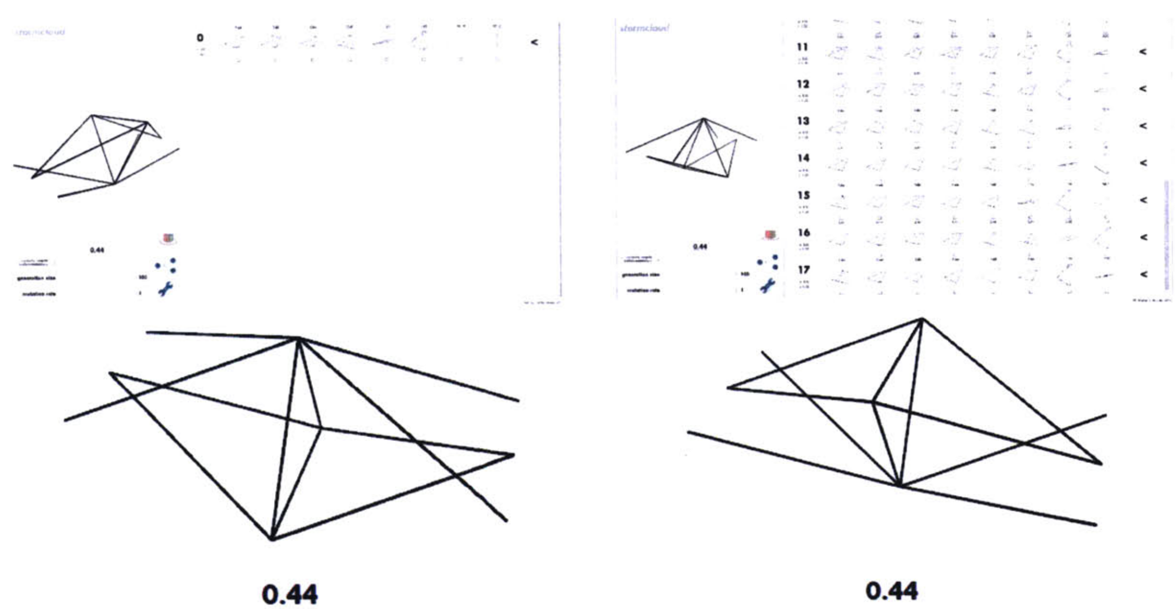


Figure 17: design case study exploration: the best score is found in one generation and the exploration yields two fundamentally different near-optimal results

4.3. Three-Dimensional Truss Canopy (18 design variables)

The design example is a three-dimensional truss covering an area of 60 m x 60 m and pin-supported in four support points. This case study is inspired by the "Earth Canopy" designed by Feilden Clegg and Atelier One (see Figure 18), a timber space frame with a free-form set of bottom chords and upper chords arranged on a flat plane where arrays of solar panels are installed (Atelier One 2015). The goal of this case study is to further illustrate the freedom offered by the tool in terms of creative guidance-based exploration.

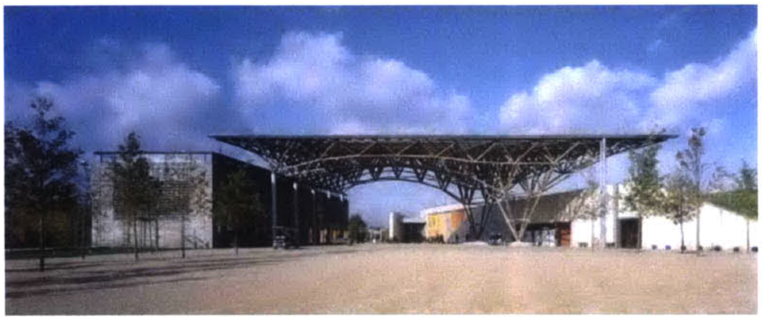


Figure 18: Earth Canopy, designed by Feilden Clegg and Atelier One (Atelier One 2015)

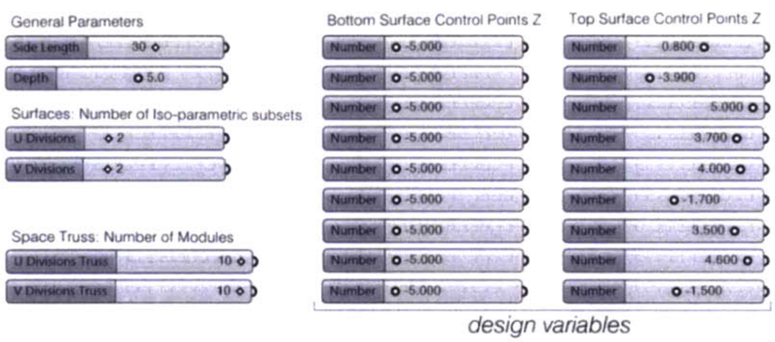


Figure 19: Design parameters (including design variables)

The structural system explored is a three-dimensional truss canopy covering an area of 60 m x 60 m. The top and bottom chords are laid out on two different NURBS surfaces that are parametrically defined in Grasshopper . The truss connectivity can be defined manually by extracting iso-parametric subsets of the top and bottom surfaces and connect their corners appropriately to define a stable space truss. Another option is to use the Lunchbox plug-in that implements the method more efficiently (Miller 2015), which is the method used in this case. The structural geometry is generated in multiple steps described in Figure 20: . Thanks to the visual programming interface, the designer can easily set up the parametric model describing the problem explored here. The problem is parametrized with 24 parameters, 18 of them are design variables (see Figure 19).

The structure is designed to resist gravity loading which is applied through point loads of 1 kN applied on every node (see Figure 21). Since the control points can only move vertically, the tributary area of each node is equal –except for the edge nodes – and the loading condition is realistic for projected loads.

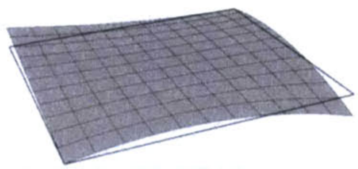
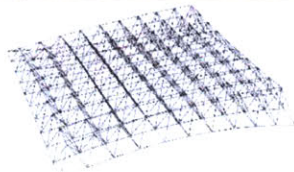
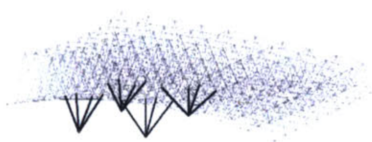
Geometry	*Step*	*Parameters*
	rectangular surface generation	side length depth
	surface manipulation through vertical translation of control points	$x_i \in \mathbb{R} \mid -5 \leq x_i \leq 5$ with i=1,2,…,18
	surfaces used as input for the space truss component	
	lines describing space truss output by component	u divisions truss v divisions truss
	4 supports points are connected with bars to the four closest truss nodes	support points positions

Figure 20: Space truss generation process

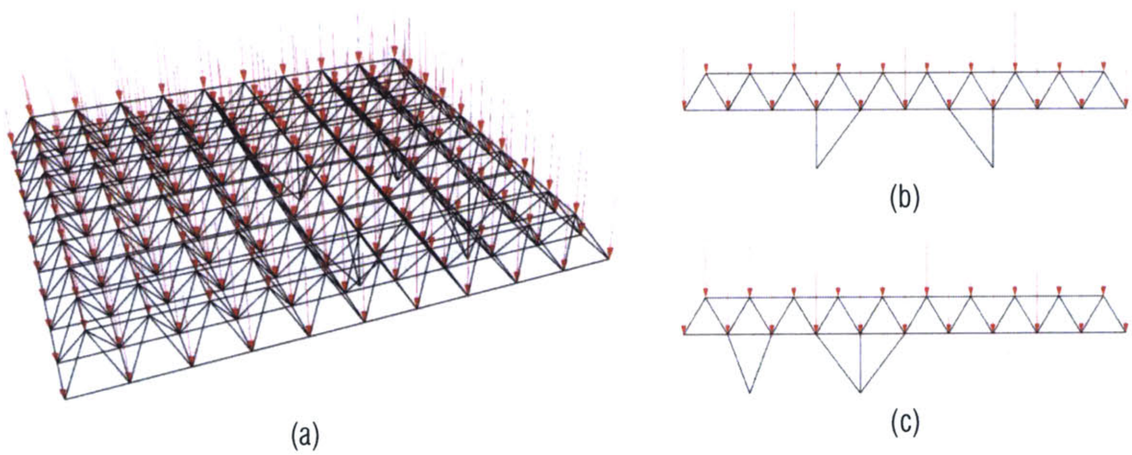


Figure 21: Initial geometry and corresponding loading: perspective (a), side (b), and front (c) views

While the problem parametrization is limited – node positions are changed indirectly - and will not yield the absolute optimal solution, the design example shows that the tool allows to explore very diverse and expressive forms which are also high-performing (see Figure 22 and Figure 23). This case study also illustrates the benefits of the different UI features implemented. Indeed, given the complexity of the structure, the 3D visualization and selection interactions are necessary to assess designs effectively. Finally, it presents the potential of integrating parametric modeling and interactive evolutionary optimization for structural design: a very complex design can easily be explored in a timely manner and result in both performant and creative designs.

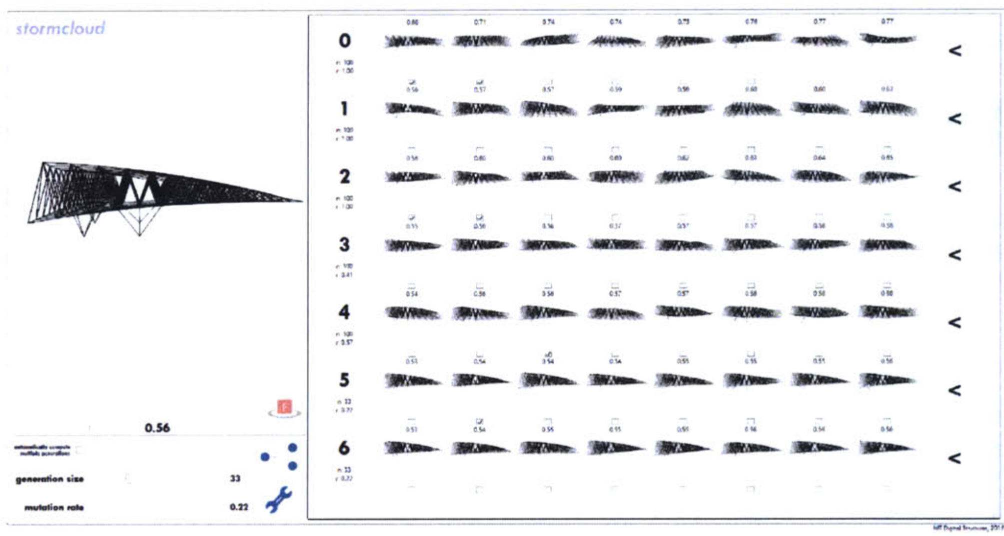


Figure 22: Design exploration using stormcloud: diverse high-performing design options are found

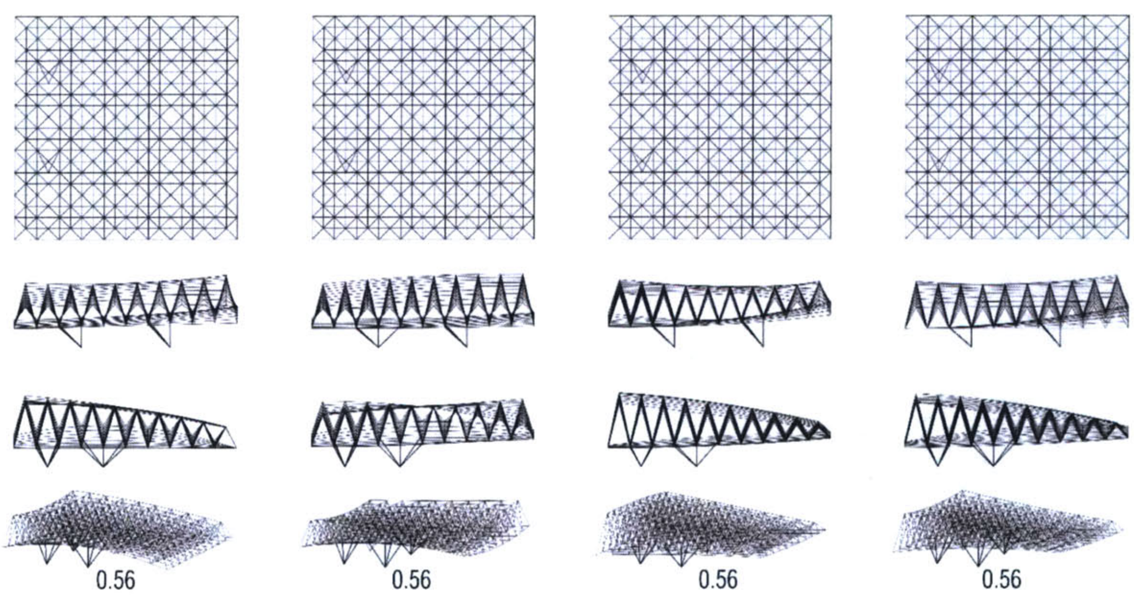


Figure 23: 4 iso-performing designs found using stormcloud (plan, side, front and perspective views)

Chapter 5. Conclusion

5.1. Summary of Contributions

This thesis has presented a new strategy for creative and high-performance structural design. After examining trends among interactive structural optimization tools, it was proposed to combine parametric design and interactive evolutionary optimization to produce a new framework. The design space itself can incorporate any objective and can describe any problem thanks to the versatility of parametric modeling. The framework was implemented in a proof-of-concept tool – *stormcloud* – which enhances Grashopper's capabilities in terms of design exploration and optimization. User interface features for enhanced interactivity were developed and presented in this thesis. Two case studies were conducted to illustrate the benefits of the framework and the tool. This research offers a versatile, performance- and user-oriented environment for creative and efficient conceptual structural design.

The implemented design tool capitalizes on Grasshopper's versatility for geometry generation but supplements the visual programming interface with a flexible portal increasing the designer's creative freedom through enhanced interactivity. The tool can accommodate a wide range of structural typologies and geometrical forms in an integrated environment.

5.2. Future Work

It is envisioned that the framework and the tool will evolve to incorporate more features for increased interactivity and design space exploration. Some of these features are presented in the following.

Design Space Visualization and Interaction

It was already mentioned in 4.2 that the interface would benefit from design space visualization and interaction features. Design space visualization is particularly important to understand the influence of design variables on performance and offers insight about a problem. Given the importance of visualization for design exploration, next versions of the tool should include tabs that allow users to interact with design space visualization.

Multi-Objective Optimization

The framework developed in this thesis should be expanded to multi-objective optimization. Human-computer optimization constitutes a form of multi-objective optimization as it accounts for ill-defined objectives that can only be evaluated subjectively by the designer in addition to a quantitative objective. However, the parametric and interactive optimization framework presented here must go beyond single-quantifiable-objective optimization or the use of composite functions. Other objectives, such as constructability (see Figure 24) or energetic performance, must be included during the conceptual design phase to reduce cost and resources consumption. While conceptually possible, interactive multi-objective optimization presents many challenges in terms of visualization, selection and implementation.

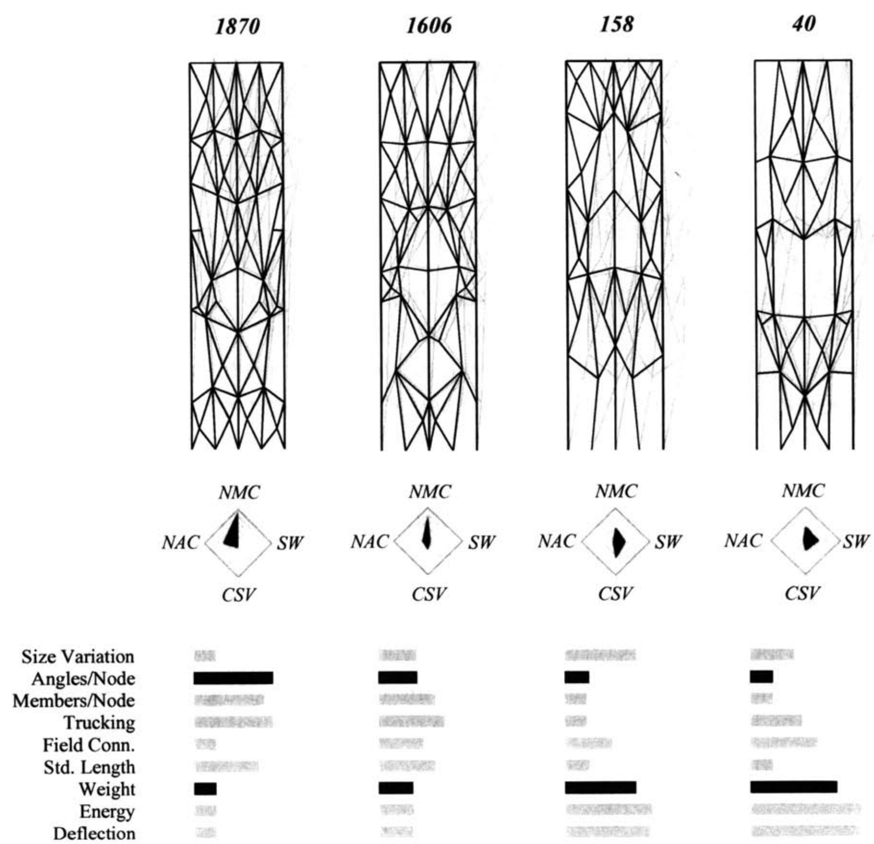


Figure 24: Integrating constructability for conceptual design of tall buildings (Horn 2015)

Surrogate Modeling

Surrogate modeling consists in approximating the design space with a model constructed using discrete data points (see Figure 25) in order to predict a design performance in a computationally efficient fashion. While structural analysis of linear elements is relatively efficient, this approach becomes necessary for many types of analysis relevant to performance-focused architectural design – daylighting analysis for example – which are very computationally expensive and cannot possibly be integrated in interactive tools as is.

Coupling surrogate models with exploration tools would dramatically increase speed of evaluation and interactivity. Since the implemented tool is blind to the analysis method used, a surrogate model can be used as an input for the design score. The development of surrogate modeling engines has the potential to greatly improve interactivity of guidance-based tools.

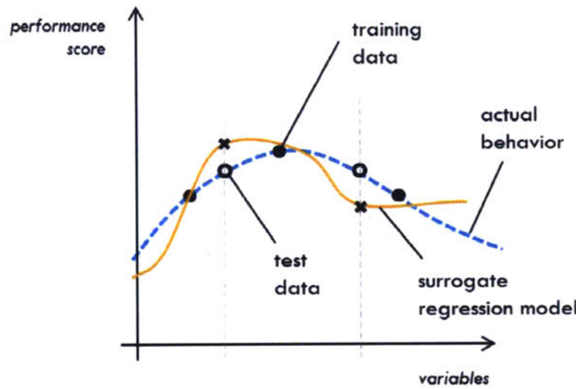


Figure 25: Surrogate modeling (Mueller and Ochsendorf 2013)

Automatic Identification and Generation of Parameters

Setting up a parametric model can be a cumbersome process. The designer's creativity must not be hampered by the parametrization capabilities of a modeling environment. There is a need for strategies that help the designer identify the key parameters of his architectural concept and generate the corresponding variables accordingly.

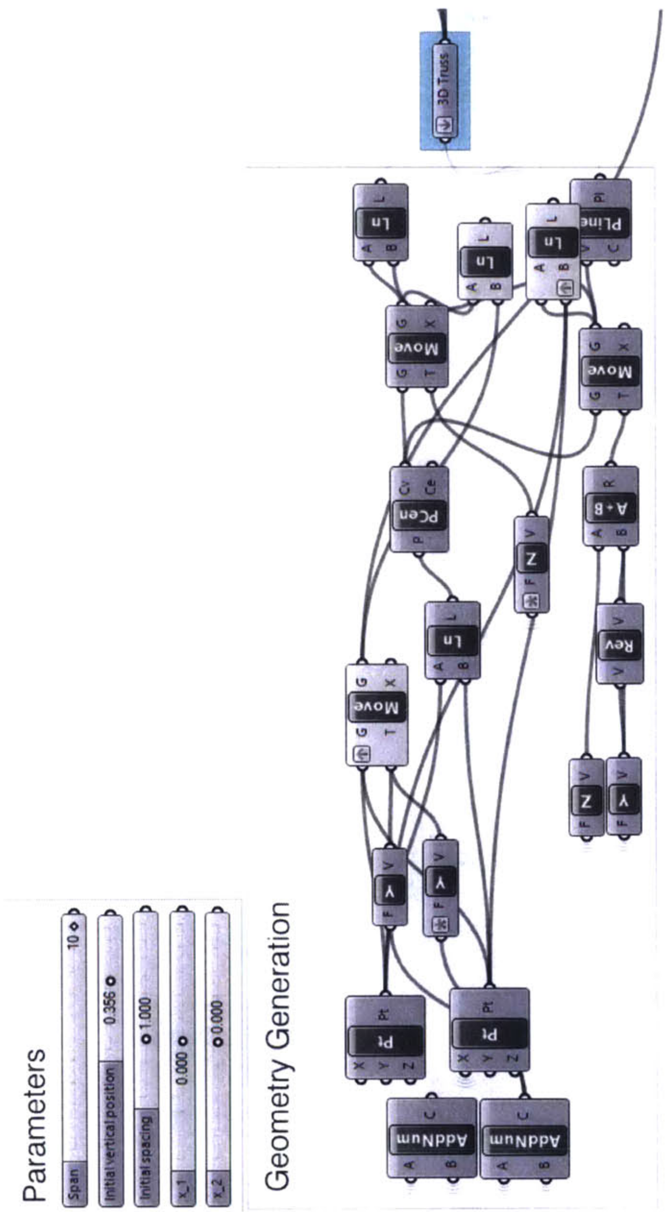
Structural Grammars

The use of rule-based search engines and trans-typology structural grammars (Mueller 2014) allows to go beyond typologies and help designers generate wide range of design options. It offers tremendous opportunities for creative and unexpected structural design and overcomes limitations of parametric space formulations which have to be defined based on well-defined design intentions.

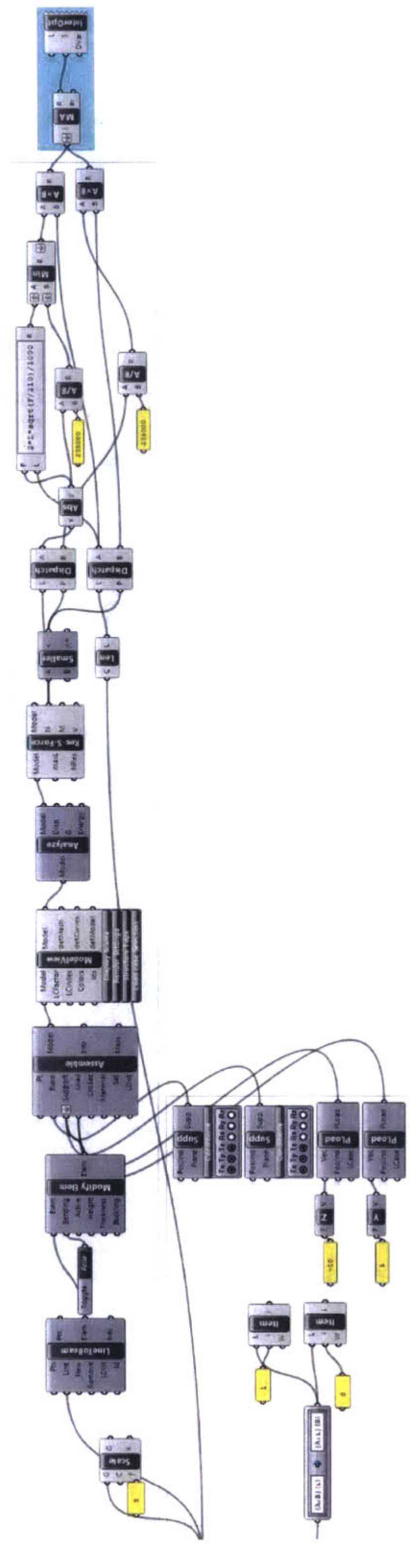
5.3. Potential Impact and Concluding Remarks

The framework developed in this research has the potential to help designers adopt interactive evolutionary optimization as a performance-oriented methodology for design. Indeed, not only the framework developed goes beyond what is available in parametric environments in terms of design space exploration but, for the first time, interactive evolutionary optimization is actually implemented in a widely used environment that has the potential to make the strategy adopted by architects and engineers. This thesis illustrates that parametric modeling can be extended using novel computational ideas to explore performance-focused design in creative ways. With such frameworks, computational design can reconcile the too often antagonistic concepts that are creativity and performance. While this work has focused on structural design, the computational methodology developed here and its implementation are inherently versatile.

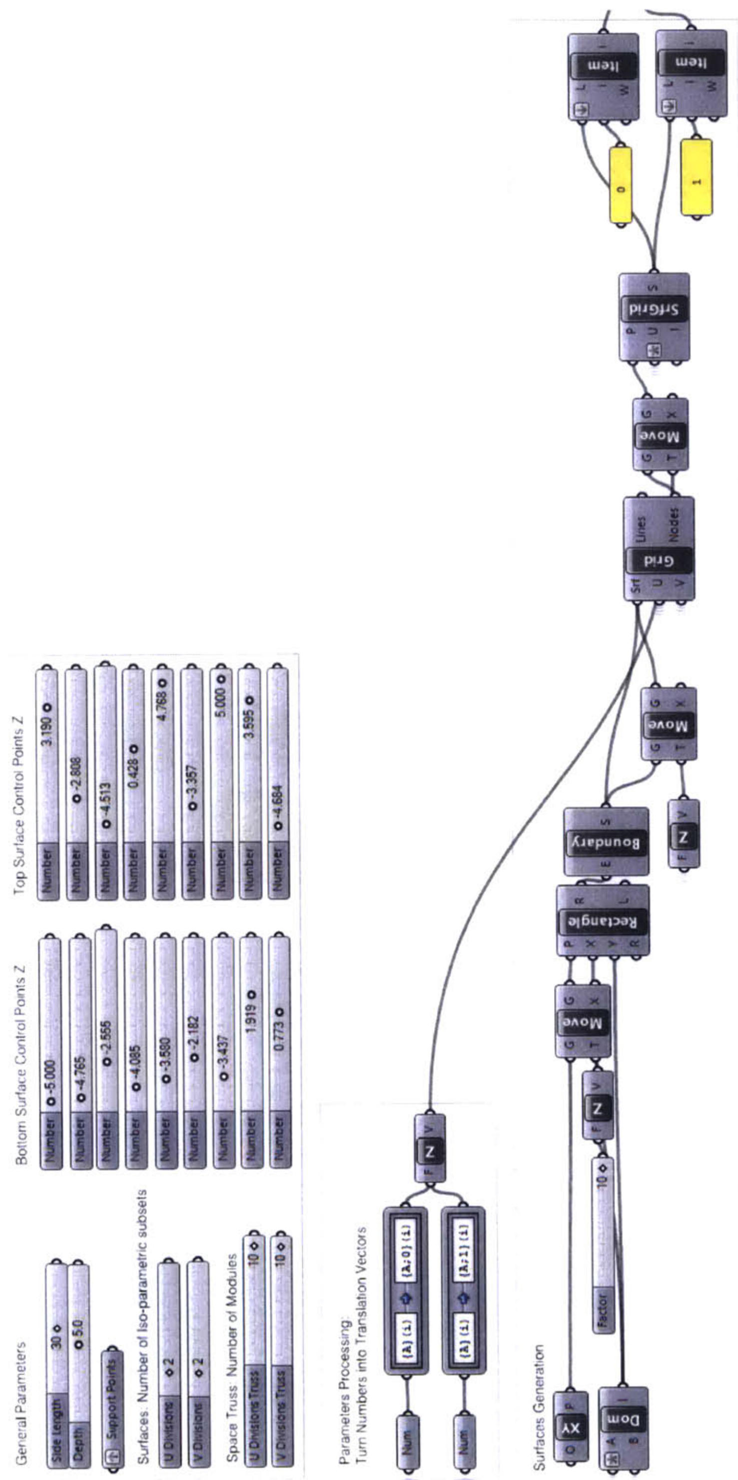
Grasshopper Script for 9 Bar Truss Case Study (part 1)



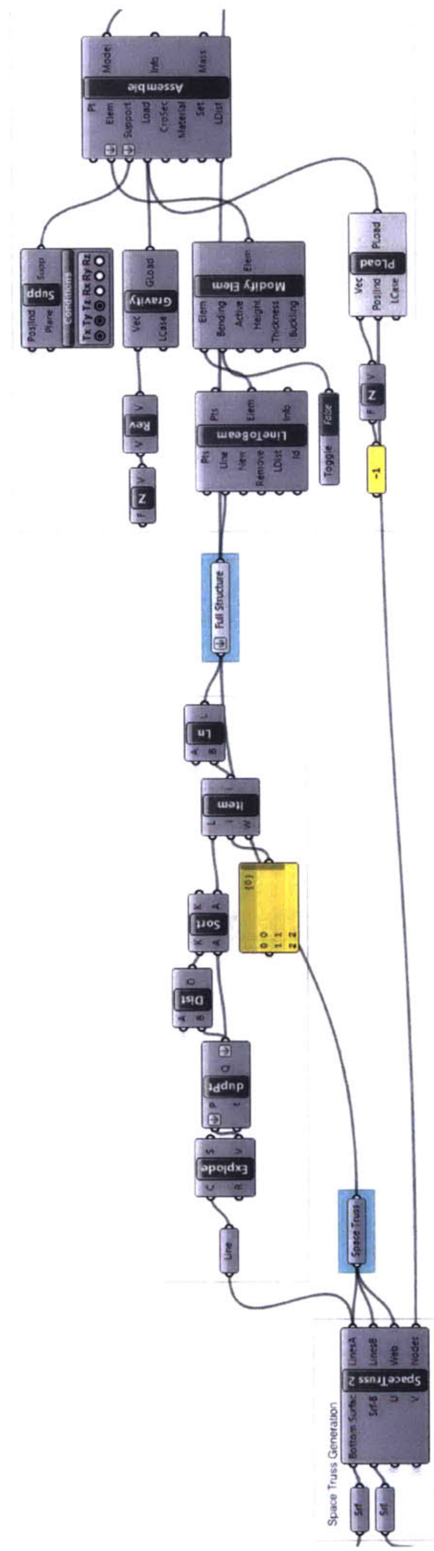
Grasshopper Script for 9 Bar Truss Case Study (part 2)



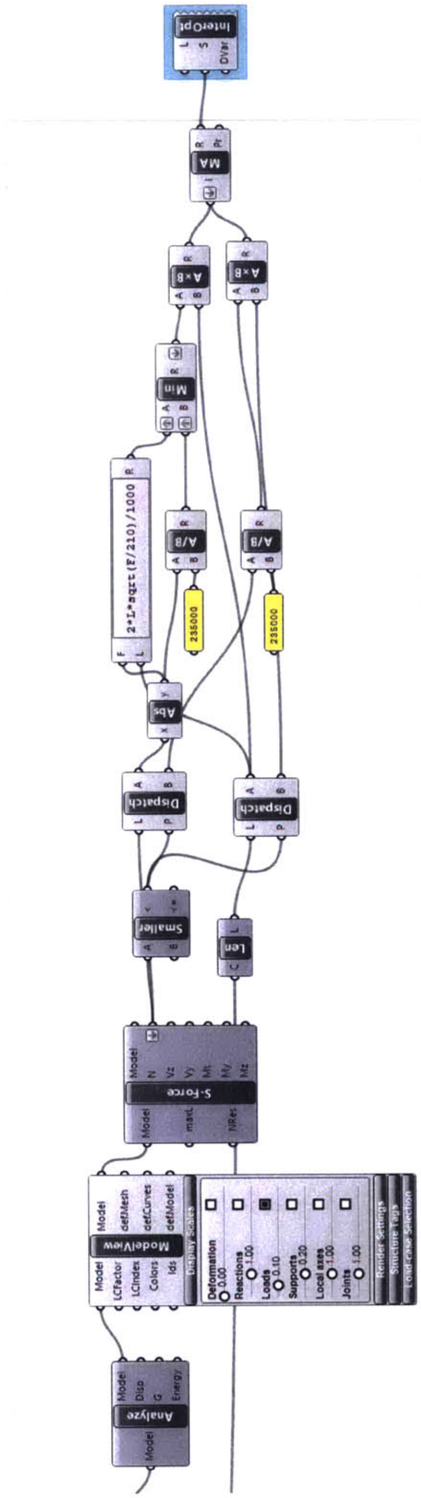
Grasshopper Script for Three-Dimensional Truss Case Study (part 1)



Grasshopper Script for Three-Dimensional Truss Case Study (part 2)



Grasshopper Script for Three-Dimensional Truss Case Study (part 3)



References

Arnaud, Virginie. 2013. *Quantifying Architects' and Engineers' Use of Structural Design Software.* M.Eng. Thesis, Massachusetts Institute of Technology.

Atelier One. 2015. *Projects - atelier one.* Accessed May 2, 2015. http://www.atelierone.com/projects/.

Autodesk. 2015. *Dynamo BIM.* http://dynamobim.com/.

—. 2015. *Project Dreamcatcher.* http://autodeskresearch.com/projects/dreamcatcher.

—. 2015. *Revit.* http://www.autodesk.com/products/revit-family/overview.

Bjorke, Oystein. 2015. *Helix 3D Toolkit.* https://github.com/helix-toolkit.

Descamps, Benoît. 2013. *Optimal shaping of lightweight structures.* Ph.D. Dissertation, Université Libre de Bruxelles.

Filomeno Coelho, Rajan, Tomas Méndez Echenagucia, Alberto Pugnale, and James N. Richardson. 2014. "Genetic algorithms for structural design." In *Shell Structures for Architecture: Form Finding and Optimization*, by Sigrid Adriaenssens, Philippe Block, Diederik Veenendaal and Chris Williams, 290-294. New York: Routledge.

Herdy, Michael. 1996. "Evolution Strategies with Subjective Selection." *Parallel Problem Solving from Nature - PPSN IV.* Berlin: Springer Berlin Heidelberg. 22-31.

Horn, Abbigayle. 2015. *Integrating Constructability into Conceptual Structural Design and Optimization.* M.Eng. Thesis, Massachusetts Institue of Technology.

Johnson, Steven G. 2015. *NLopt.* http://ab-initio.mit.edu/nlopt.

Microsoft. 2015. *C#/.NET.* https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx.

—. 2015. *Windows Presenation Foundation.* https://msdn.microsoft.com/en-us/library/ms754130%28v=vs.110%29.aspx.

Miller, Nathan. 2015. *LunchBox.* http://www.grasshopper3d.com/group/lunchbox.

Mueller, Caitlin. 2014. *Computational Exploration of the Structural Design Space.* Ph.D. Dissertation, Cambridge: Massachusetts Institute of Technology.

—. 2015. *structureFIT.* http://www.caitlinmueller.com/structurefit/.

Mueller, Caitlin, and John Ochsendorf. 2013. "An Integrated Computational Approach for Creative Conceptual Structural Design." *Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium.* Wroclaw.

—. 2011. "An Interactive Evolutionary Framework for Structural Design." *7th International Seminar of the Structural Morphology Group (SMG), IASS Working Group.* London, UK.

Mueller, Caitlin, and John Ochsendorf. 2015. "Combining structural performance and designer preferences in evolutionary design space exploration." *Automation in Construction 52* 70-82.

Rahmani, Mohammad. 2015. *Optimo.* http://dynamobim.com/optimo/.

Rechenraum. 2015. *goat.* http://www.rechenraum.com/en/goat/overview.html.

Robert McNeel & Associates. 2015. *GhPython.* http://www.food4rhino.com/project/ghpython?ufh.

—. 2015. *Rhinoceros.* https://www.rhino3d.com/.

Robert McNeel and Associates. 2015. *Grasshopper 3D.* http://www.grasshopper3d.com/.

Scott, Stacey D., Neal Lesh, and Gunnar W. Klau. 2002. "Investigating Human-Computer Optimization." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* . New York, USA. 155-162.

Thornton Tomasetti. 2015. *TT Toolbox.* http://www.food4rhino.com/project/tttoolbox?ufh.

Tseranidis, Stavros. 2015. *Approximation Algorithms for Rapid Performance Evaluation and Optimization of Structures.* M.S. Thesis, Massachusetts Institute of Technology.

Turrin, Michela, Peter von Buelow, and Rudi Stouffs. 2011. "Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms." *Advanced Engineering Informatics.*

Vierlinger, Robert. 2015. *Octopus.* http://www.food4rhino.com/project/octopus?ufh.

von Buelow, Peter. 2012. "Paragen: Performative Exploration of Generative Systems." *Journal of the International Association for Shell and Spatial Structures*, December 4.

von Buelow, Peter. 2008. "Suitability of genetic based exploration in the creative design process." *Digital Creativity* 19 (1): 51-61.