

8-2017

# Integrating Parametric Structural Analysis and Optimization in the Architectural Schematic Design Phase

Nixon Wonoto

Clemson University, [nixonwonoto@gmail.com](mailto:nixonwonoto@gmail.com)

Follow this and additional works at: [https://tigerprints.clemson.edu/all\\_dissertations](https://tigerprints.clemson.edu/all_dissertations)

---

## Recommended Citation

Wonoto, Nixon, "Integrating Parametric Structural Analysis and Optimization in the Architectural Schematic Design Phase" (2017).  
*All Dissertations*. 1963.

[https://tigerprints.clemson.edu/all\\_dissertations/1963](https://tigerprints.clemson.edu/all_dissertations/1963)

This Dissertation is brought to you for free and open access by the Dissertations at TigerPrints. It has been accepted for inclusion in All Dissertations by an authorized administrator of TigerPrints. For more information, please contact [kokeefe@clemson.edu](mailto:kokeefe@clemson.edu).

INTEGRATING PARAMETRIC STRUCTURAL ANALYSIS AND OPTIMIZATION  
IN THE ARCHITECTURAL SCHEMATIC DESIGN PHASE

---

A Dissertation  
Presented to  
the Graduate School of  
Clemson University

---

In Partial Fulfillment  
of the Requirement for the Degree  
Doctor of Philosophy  
Planning, Design, and the Built Environment

---

by  
Nixon Wonoto  
August 2017

---

Accepted by:  
Vincent Blouin, Committee Chair  
Carlos Barrios  
Gregory Mocko  
Matthew Powers  
Matthew Saltzman

## ABSTRACT

The implementation of computational form-finding structural optimization methods has recently mushroomed in the architectural research area. There has been a few emerging architectural parametric Computer-Aided Design (CAD) systems that enable architects to perform early schematic form-finding structural optimization such as the coupling of Grasshopper (a visual programming language), Karamba (a structural analysis plugin) and Galapagos (an optimization plugin). However, the application of the method is very rare in both educational and design practice environments. Also, the architectural schematic design phase is commonly characterized by free-form shapes without the embedded considerations of the material and structural system. On the other hand, the considerations of materiality and structural system are often more properly imposed by structural engineers, who usually prefer to be involved as early as possible in the project. Seen from this perspectives, this research examines the implementation of structural optimization in the architectural schematic design phase; investigate the accessibility and usability of existing architectural structural optimization tools; and study the interoperability and integration of architectural parametric CAD tools and engineering analysis and optimization tools as well as the usability of these tools. This research uses Grounded Theory for data collection and analysis procedure to investigate those research concerns. A comparative study of software is also used to examine the second research concern. Semi-structured interviews are used to acquire in-depth understanding of the participants' responses towards incorporating architectural structural optimization

procedure in the context of the collaboration between architects and engineers. Students and faculty, with years of design practice experience, in Clemson University are used as the target population for the interview process. Five architectural, form-finding structural optimization methods are developed to facilitate the interview process. Improvements of the tools are made based on the participants' responses towards the usefulness of the tools. Finally, guidelines concerning the implementation of the developed architectural structural optimization for the educational and design practice purposes were developed. The design guidelines are developed with the aim to better the communication between architects and engineers during the collaboration process. This research believes that participants' in-depth responses toward the contemporary architectural design issues and the developed methods are the essential driving forces that help this research in finding ways to improve the collaboration between architects and structural engineers.



## DEDICATION

This dissertation is dedicated to my father Jon Wonoto and my mother Liliswati Tiono. Their unconditional love and encouragement have made me the person I am today, and I am eternally grateful for all they have done for me. I would like to also dedicate this dissertation to my oldest brother, Niki Wonoto. Conversing with him about science and life for hours has always been very enjoyable. Last but not least, I would like to also dedicate this dissertation to my second oldest brother, Nikko Wonoto, whose practical advice are always invaluable to me.

## ACKNOWLEDGEMENT

I would like to express my greatest gratitude to my committee members for their guidance and support during my time in the PDBE program. Above all, I would like to deeply express my thankfulness to my advisor and committee chair, Dr. Vincent Blouin, who has patiently guided me throughout the research process. He has always kindly guided and supported me to dive deeper into my passion for studying the world of structural optimization and wholeheartedly responded to any of my intellectual curiosity. I have always been dreaming of becoming a professor in the future, and Dr. Blouin has truly become a role model for me in this endeavor. I would also like to thank Dr. Barrios for agreeing to serve on my review committee and giving me invaluable inputs for improving my research. I would also like to sincerely thank Dr. Mocko for continually guiding me on bringing my research closer to practical environment of design and engineering analysis. My thanks also goes to Dr. Powers who helped me in reviewing the Grounded Theory I used for the qualitative part of this research. I would also like to sincerely thank Dr. Saltzman for guiding me through the beautiful realm of operations research and teaching me optimization theories and techniques that I can conveniently implement for the validity study of the software development in this research.

I would like to acknowledge Dr. Lauria, the director of PDBE program, who taught me doctoral research methods and helped me in many ways during my study. My thanks also go to Connie Robinson for helping me with registrations issues along the way. I would also like to thank all the research participants, Clemson University students,

and faculty whose name I cannot mention one by one, that have tremendously helped me in shaping and answering issues related to my research concerns with their interview responses. Warm thanks also go to my colleagues in the PDBE program. Conversing with them is always a pleasurable experience.

Last but not least, I would like to thank my father Jon Wonoto, my mother Liliswati Tiono, my oldest brother Niki Wonoto and my second oldest brother Nikko Wonoto for all the love and encouragement that they continuously give me.

## TABLE OF CONTENTS

	Page
TITLE PAGE.....	i
ABSTRACT.....	ii
DEDICATION.....	iv
ACKNOWLEDGEMENT.....	v
LIST OF TABLES.....	x
LIST OF FIGURES.....	xii
I. INTRODUCTION.....	1
1.1 Motivation.....	1
1.2 Research concerns.....	6
1.3 Research methods.....	8
1.4 Intellectual merit.....	10
1.5 Broader impacts.....	12
II. REVIEW OF LITERATURE.....	13
2.1 Case Study of Organic Light Weight Structures.....	13
2.2 Parametric Structural Design and Optimization.....	19
2.3 Constructivism and Grounded Theory.....	35
III. RESEARCH DESIGN.....	42
3.1 Qualitative Research Method.....	42
3.2 Grounded Theory for Development of Tools.....	42
3.3 Grounded Theory for Data Collection.....	45
3.4 Grounded Theory for Semi-Structured Interview.....	45

	Page
3.5 Grounded Theory for Data Analysis.....	47
3.6 Grounded Theory for Justifiability and Transferability .....	48
IV. TOOL DEVELOPMENT .....	50
4.1 iMAGv1 .....	50
4.2 iMAGv2 .....	69
4.3 iMAGv3 .....	70
4.4 iGFM.....	72
4.5 iGAG.....	77
4.6 iGKG.....	83
4.7 Comparative Study for the Research Concern Two .....	85
4.8 Case Study - Design of a Twisted Skyscraper .....	87
V. PILOT STUDY .....	107
5.1 Semi-Structured Interview for the Pilot Study.....	107
5.2 Qualitative Data Analysis .....	110
5.3 Theoretical Narrative for the Pilot Study.....	112
VI. FIRST CYCLE OF GROUNDED THEORY .....	114
6.1 Tool Improvement Based on the Pilot Study .....	114
6.2 Modification on the Qualitative Research Tasks .....	114
6.3 Theoretical Narrative for the First Cycle of Grounded Theory .....	121
VII. SECOND CYCLE OF GROUNDED THEORY.....	128
7.1 Modification on the Qualitative Research Tasks .....	128
7.2 Theoretical Narrative for the Second Cycle of Grounded Theory .....	130
VIII. GUIDELINES FOR USING THE DEVELOPED METHODS .....	142

	Page
8.1 Guideline for Introducing the Methods in the Academic Environment.....	142
8.2 Guideline for Implementing the Methods in the Design Practice Environment.....	147
IX. CONCLUSION.....	156
APPENDICES .....	171
A. iMAGv1 User Manual for Interview .....	172
B. Interview Module Transcripts.....	179
C. IRB Approval Document .....	185
D. Sample Interview Transcript A (1 <sup>st</sup> Cycle) .....	186
E. Sample Interview Transcript B (2 <sup>nd</sup> Cycle) .....	191
F. Sample Interview Transcript C (2 <sup>nd</sup> Cycle) .....	195
G. Sample Interview Transcript D (2 <sup>nd</sup> Cycle) .....	201
H. iMAGv1 Source Code.....	207
I. iMAGv2 Source Code.....	228
J. IMAGv3 Source Code .....	231
K. iGFM Source Code .....	234
L. iGAG Source Code .....	251
M. Grasshopper Definition Space Truss .....	264
N. Grasshopper Definition Skyscraper .....	265
REFERENCES .....	266

## LIST OF TABLES

Table	Page
3.1. General format for interview questions .....	47
3.2. General format for organizing qualitative data .....	48
4.1. Comparing analytical and numerical optimization results .....	57
4.2. Comparing analytical and numerical displacement results (non-prismatic beam).....	59
4.3. Comparing analytical and numerical stress results (non-prismatic beam).....	59
4.4. Comparing analytical and numerical displacement results (B31 element).....	61
4.5. Comparing analytical and numerical displacement results (T3D2 elements).....	63
4.6. Results of the initial and optimized condition of each example problem .....	69
4.7. Comparison of structural optimization methods .....	86
4.8. Lower and upper bound state of the optimization .....	90
4.9. Some of the best individuals generated at each generation .....	97
4.10 Two best results when using exhaustive search .....	104
5.1. Interview format for the pilot study.....	109
5.2. Coding of the pilot study interview responses .....	110
6.1. Interview format for interview cycle 1 .....	119

	Page
7.1. Demographic data of the participants.....	128
7.2. Interview format for interview cycle 2.....	129



## LIST OF FIGURES

Figure	Page
1.1. Diagram of contemporary building design process.....	4
1.2. Excerpt of the contemporary building design workflow.....	5
1.3. Proposed building design workflow.....	9
2.1. London Aquatics Center.....	13
2.2. Shenzhen Bao'an Airport.....	14
2.3. National Aquatics Center.....	15
2.4. Eden Project.....	15
2.5. Norman Foster St Mary Axe.....	16
2.6. Capital Gate Abu Dhabi.....	17
2.7. Diagram showing the scope of structures. The subset considered in this research is.....	18
2.8. Rhino/Grasshopper interface.....	21
2.9. M.S. Architecture thesis work (Wonoto, 2013a): parametric structural analysis of deployable tessellated structures.....	30
2.10. British Museum Great Court.....	32
2.11. Qatar Education City Convention Center.....	33
2.12. Rectangular Stadium in Melbourne, Australia.....	34

	Page
3.1. Grounded Theory model used for developing and testing the developed structural optimization methods .....	44
3.2. Semi-structured interview modules and main parameters.....	46
4.1. Integration scheme of iMAGv1 .....	52
4.2. Grasshopper premade components .....	53
4.3. Example tessellated structure .....	55
4.4. Portion of the tessellated structure taken as a statically indeterminate non-prismatic beam problem .....	57
4.5. Cross-checking between Abaqus and analytical analysis results .....	59
4.6. Node numbering on the frame model for verification purpose .....	60
4.7. Karamba results for the frame model .....	60
4.8. Abaqus results for the frame model (B31 elements) .....	61
4.9. Direct stiffness method for the planar frame model .....	61
4.10. Direct stiffness method for the planar truss model.....	62
4.11. Problem 1: Single curvature tessellated frame structure .....	64
4.12. Problem 2: Double curvature tessellated frame structure.....	65
4.13. Problem 3: Double curvature space truss structure .....	66
4.14. Optimized results of each of the three example problems.....	68
4.15. Workflow of iMAGv2 .....	70
4.16. Workflow of iMAGv3 .....	72
4.17. Workflow of iGFM.....	73

	Page
4.18. Validation of the developed customized FEA (top) by comparing the maximum displacement and stress results with Abaqus (bottom left) and Karamba (bottom right) .....	75
4.19. Space truss structure (c) made of two trusses of different section properties: (a) inner and outer polyhedrons and (b) connecting braces .....	76
4.20. Comparing results between customized FEA and Abaqus for a problem with various section properties .....	76
4.21. Workflow of iGAG.....	78
4.22. Booth function .....	79
4.23. Customized GA validation for the Booth function.....	80
4.24. Levi function .....	80
4.25. Customized GA validation for Levi function .....	81
4.26. Easom function .....	82
4.27. Customized GA validation for Easom function .....	83
4.28. Workflow of iGKG.....	84
4.29. Grasshopper interface, Karamba plot, and Galapagos interface .....	85
4.30. Bracing types and variability of the parametric model of the skyscraper .....	89
4.31. Displacement plot of the lower and upper bound state .....	90
4.32. Reduction of the thickness of the cross section.....	92
4.33. Windward and leeward pressure coefficient at sampled sections for the sampled design permutation .....	94

	Page
4.34. iGAG GA plots for tower design.....	96
4.35. Geometries of the best individuals generated at each generation.....	97
4.36. Displacement and load vectors plot of the best individual at iteration 40. ....	98
4.37. Displacement and stress obtained from Abaqus of the best individuals at iteration 40.....	99
4.38. Wireframe and rendered modes of the optimum at iteration 40.....	100
4.39. Cross-section of the beam at the optimum of iteration 40.....	101
4.40. Rendering of the skyscraper design at the optimum of iteration 40.....	102
4.41. Closed up rendering of the skyscraper design at the optimum of iteration 40 .....	103
4.42. Graphical representation of the design space showing feasible designs and the optimum using the exhaustive search analysis.....	105
5.1. Planar frame structure for the pilot demonstration module.....	108
6.1. Difference between performance-based and prescriptive approach.....	115
6.2. Examples of prescriptive vs. performance-based.....	116
6.3. Structural optimization problem used in the educational module .....	117
6.4. iMAGv1 structural optimization interface .....	118
6.5. Repeating ideas and emerging themes using MAXQDA (1 <sup>st</sup> cycle).....	121
6.6. Code-subcode-segments model for the theme “usually do not analyze the structure.”.....	123

	Page
6.7. Code-subcode-segments model for the theme “needs to do structural optimization.” .....	124
6.8. Code-subcode-segments model for the theme “needs design flexibility in structural optimization.” .....	126
6.9. Code-subcode-segments model for the theme “iMAGv1 benefits.” .....	127
7.1. Emerging themes using MAXQDA (2 <sup>nd</sup> cycle).....	131
7.2. Code-subcode-segments model of the theme “architects usually do not analyze the structure.” .....	133
7.3. Code-subcode-segments model of the theme “problems in the collaboration between architects and engineers.” .....	136
7.4. Code-subcode-segments model of the theme “preferred types of optimization in the design process.” .....	139
7.5. Code-subcode-segments model of the theme “iMAGv1 benefits.” .....	141
8.1. Grasshopper course at Clemson University .....	144
8.2. Structures I course at Clemson University .....	145
8.3. M. Arch plan of study.....	146
8.4. Structures II course at Clemson University.....	147
8.5. Proposed building design workflow using the developed methods.....	148
8.6. Design guideline for implementing architectural, structural optimization methods.....	149
8.7. Flowchart of the proposed collaboration process when implementing form-finding structural optimization methods.....	153

# CHAPTER ONE

## INTRODUCTION

### **1.1 Motivation**

Two factors control the success of a building design project, a hard factor and a soft factor. The hard factor is related to the technical dimensions of the project implementation (Pecherskaya et al., 2016). At the operational level, a project success is indicated by good time management, cost-effectivity, and meeting the physical resources and the technical specifications (Phua, 2004). The soft factor is related to people, the emotional intelligence, multiple perspectives, organization aspects, and communication (Koutsikouri et al., 2008). Many researchers agree that the soft dimensions of a project implementation are often the culprit of a project failure (Chen, 2001; Kumar et al., 2003; Markus et al., 2000). Akkermans and Helden (2002) also found that effective communication and collaboration between project members are the determinants to a project success. Without a common 'language,' an effective communication for bridging multiple perspectives is difficult to achieve. Failure in the soft dimension of project implementation often leads to a loss of mutual respect between collaborators and can potentially cost a lot of money and time during the late project resolution. It is possible, however, for a project to succeed in terms of teamwork (i.e., forming a good relationship with other collaborators), even when the project fails in terms of meeting the basic standards such as cost and time (Koutsikouri et al., 2008). Thus, it is reasonable to assume that a project success can be realized if two or more different parties speak the same 'language,' i.e., share as much as possible common knowledge about a phenomenon

under the study to form the communication. Unfortunately, it is not the case in many contemporary architecture-structural engineering collaborations.

Many research endeavors have studied the problems related to the collaboration between architects and structural engineers. The research statements and findings vary, but many of those converge towards the following opinion: *structural engineers and architects often speak different ‘languages’, i.e., one, as a technician, lacks innovation and creativity, and the other, as an artist, focuses merely on appearance, and the differences often lead to frustration on both sides during the collaboration process* (see Koutsikouri et al., 2008; Dougherty, 1992; Charleson & Pirie, 2009; Kieran & Timberlake 2004; Ozmen & Unay, 2011; Peters, 1991; Luyten, 2010; Davison et al., 1998; Hurol, 2013; Von Buelow, 2012). For instance, Tom F. Peters, in his book ‘Bridging the Gap: Rethinking the Relationship of Architect and Engineer,’ describes:

*“... While engineering hopes to be moving toward a more comprehensive approach to design and building, and the very nature of the word ‘design’ in engineering seems to be shifting to mean more ‘configuration’ than ‘dimensioning’, architecture is in danger of diversifying into literary and purely graphic pursuits, on occasion so strongly that some architects become mere aesthetic consultants or even abandon building altogether.”* (Peters, 1991)

Another example is from Charleson and Pirie’s survey which conclude:

*“...Structural engineers’ main concerns focus upon the following: architects’ lack of structural understanding; architects seeking structural advice too late for optimal structural solutions, and the need for architects in general to improve their focus upon collaboration. On the other hand, architects are disappointed by engineers’ lack of both innovation and engagement with architectural design ideas.”* (Charleson and Pirie, 2009)

Working on the same problem with these gaps of knowledge and interest often causes frustration on both sides (Charleson & Pirie, 2009).

Present-day collaboration between architects and engineers is often inseparable from the utilization of CAD and Building Information Modeling (BIM). There are two approaches to the development of building designs: (1) the material agnostic or *amaterial* approach (CAD), and (2) the BIM approach (Gentry, 2013). In the architectural design process, the former is a typical CAD system and is characterized by freeform surface modeling process without the material knowledge and geometrical description of architectural elements using a commercial drawing software such as Rhinoceros (Rhinoceros, 2017). The latter is characterized by a more restraining modeling process in which a model is represented by objects, and is constrained by the characteristics of architectural elements such as walls, beams, columns, and the embedded consideration of materiality (for details about BIM, see Eastman et al., 2008). BIM is commonly considered more convenient to be used during the architectural design development phase (see Figure 1.1), where a more mature architectural articulation of the structural connections, structural elements, member sizes, floor plates, etc. have been defined through several consultations with engineers. Often, in the contemporary design process, the transition from the schematic design to design development, i.e., from the freeform surface model to BIM, is very challenging.



1. Programming	2. Site Analysis	3. Cost & Feasibility	4. Schematic Design
<ol style="list-style-type: none"> <li>1. Develop program</li> <li>2. Client's needs</li> <li>3. Define service scope</li> <li>4. Set goal</li> <li>5. Set design standards</li> </ol>	<ol style="list-style-type: none"> <li>1. Develop master plan</li> <li>2. Environment analysis</li> <li>3. Social analysis</li> <li>4. Economic analysis</li> </ol>	<ol style="list-style-type: none"> <li>1. Define project scope</li> <li>2. Budget analysis</li> <li>3. Scheduling</li> </ol>	<ol style="list-style-type: none"> <li>1. Design concept</li> <li>2. Modelling</li> <li>3. Drawings</li> <li>4. Renderings</li> </ol>
5. Engineering Systems	6. Cost & Code Regulation	7. Design Development	8. Construction documents
<ol style="list-style-type: none"> <li>1. Preliminary structural analysis</li> <li>2. Coordinate electrical, mechanical, fire safety</li> </ol>	<ol style="list-style-type: none"> <li>1. Estimate construction cost</li> <li>2. Code analysis (building, energy)</li> </ol>	<ol style="list-style-type: none"> <li>1. Select building system &amp; materials</li> <li>2. Prepare dev. Documents</li> </ol>	<ol style="list-style-type: none"> <li>1. Prepare construction documents</li> <li>2. Coordinate documents. (architectural, structural, mechanical, electrical)</li> </ol>

Figure 1.1. Diagram of contemporary building design process (Adapted from NCARB, 2016)

Collaboration difficulties emerge because constraints involving structural sizing and materiality are imposed on the design through the involvement of engineers. In other words, through sets of constraints, engineers want to transform the *naïve* design to become more mature. Unfortunately, this transformation process often severs the creative and innovative parts of the design. Holgate describes:

*“...The second design phase... consists of a tremendous battle between the structural engineer and the architect... The result of the struggle is always the same: science prevails, and the final design has generally lost the eventual charm and finesse of detail dreamed by the architect.”* (Holgate, 1986)

This unsmooth transition frequently happens after the schematic design phase in the contemporary design process (Figure 1.2). Design goals and concepts are formulated separately from the structural constraints by different individuals. Architectural innovation is often diminished when structural constraints are introduced by the structural engineers during the preliminary structural analysis.

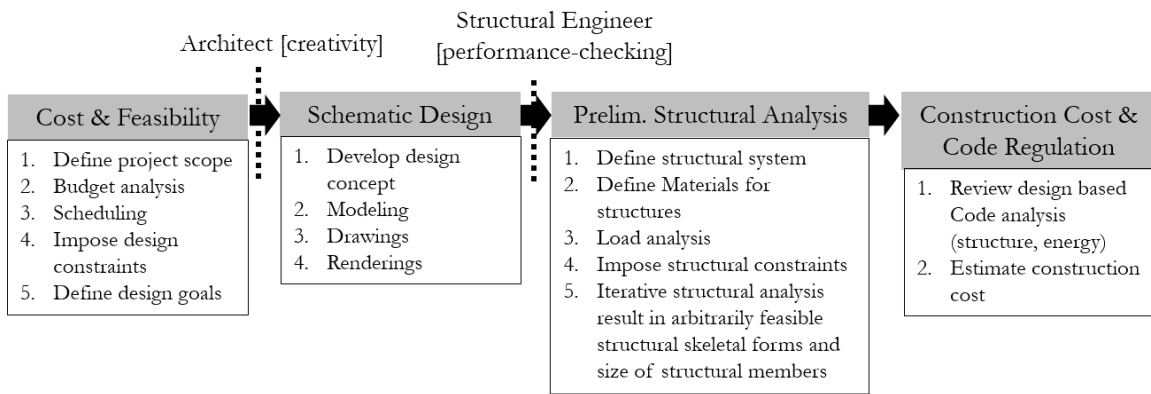


Figure 1.2. Excerpt of the contemporary building design workflow

Recently, several plugins such as Karamba (Karamba, 2016) and Galapagos (Rutten, 2017) have been developed to allow a structural analysis and optimization to be carried out conveniently through a visual programming language (such as Grasshopper) within a surface modeling CAD system (such as Rhino). The emergence of this integrated system allows a form-finding task to be driven by structural performance in the early stage of design, before the design development phase or BIM implementation. However, form-finding requires the evaluation of a series of alternative designs, which is the process of structural optimization. Thus, structural analysis and optimization are inseparable in the form-finding process. Particularly, in this thesis, such type of form-finding process is called architectural structural optimization. The goal of an architectural structural optimization is to automate the search of a rough structural shape and sizes of a building and structural members which finds the balance between design intent and structural performance.

As illustrated in figure 1.2, the incorporation of an early architectural structural optimization would contradict the traditional architectural schematic design workflow

that is commonly characterized by conceptual free-form shapes without consideration of materiality. There is not much research both in architecture and engineering that qualitatively assess the broader impact of incorporating schematic form-finding structural optimization on the architect-engineering collaboration. As a result, despite the availability of commercial architectural structural optimization software, the exploration of structural optimization in the architectural field commonly remains exclusive within the academic research environment and is not vastly applied in the design practice.

This research is interested particularly in the free-form organic structure. The term “organic” is used here to define a structure that is not geometrically repetitive (Nero, 2004), i.e., free-form, unpredictable, flowing and asymmetrical in nature. Traditional buildings with regular geometries and column grids are generally designed based on the available prescriptive codes. On the other hand, the structural performance of non-traditional free-form buildings can only be assessed using performance-based codes such as through computer simulation. This research focuses on the latter in which the goal is to search the form of free-form structures which are generally the results of the architects’ creativity.

## **1.2 Research concerns**

Based on the identified contemporary issues, this research strives to learn more about the respond of architects if the consideration of structural optimization in architecture is incorporated into the schematic design phase. From the motivation above, the research concerns are developed as follows:

Research Concern 1: Examine the implementation of structural optimization in the architectural schematic design phase.

Research Concern 2: Investigate the accessibility and usability of existing architectural structural optimization tools.

Research Concern 3: Study the interoperability and integration of architectural parametric CAD tools and engineering analysis and optimization tools as well as the usability of these tools.

It is important to note that a qualitative research method such as using Grounded Theory does not assume that the researcher knows enough to formulate specific hypotheses (Auerbach and Silverstein, 2003). Thus, unlike in traditional quantitative research methods, research questions are not formulated. Instead, research concerns are used to drive the research process.

The reason for using a qualitative research method as opposed to a quantitative research method is because architects generally do not have sufficient structural knowledge to be able to understand the process involved in structural optimization. Thus, it is assumed in this research that it is necessary to have back and forth communication between participants (architects) and the researcher during the data collection process. The communication is necessary to educate the participants about architectural structural optimization such that the responses from the participants are the mix of the newly acquired education and their academic and design practice experiences.

All the research concerns above are examined using Grounded Theory procedure. However, research concern 2 primarily involves a comparative study method. The first

research concern explores the contemporary architectural schematic design process. The third research concern examines specifically about the structural optimization tool that is developed in this research, and the way the software can be improved. Note that research concern 3 is required to assess the research concern 2, which is done as the comparative study (chapter 4). Those three research concerns are used to construct the interview questions.

### **1.3 Research methods**

Based on the contemporary issue, two actions are formulated to minimize the gap between architects and engineers:

- The first action is to bring the consideration of materiality and structural systems into the early architectural schematic design phase before the collaboration takes place.
- Use the considerations in the first action as part of the larger design system to help the designer generate free-form structure instead of limiting them.

The two actions are realized in this research by conducting an investigation and development of tools involving parametric Computer-Aided Design (CAD) system, numerical structural analysis using Finite Element Analysis software (FEA), and a scientific programming language for linking the parametric CAD system and FEA.

Using both the interview responses and the tools built on those responses, the primary goal of this research is to minimize the gap between architects and structural

engineers regarding both human communication and the software interoperability. In terms of software interoperability, the strategy of efficiently coupling architectural with engineering software are considered necessary such that the operating procedure and results of the developed tools can be conveniently used and interpreted by both architects and engineers as needed. The structural optimization methods are developed as a surrogate to the contemporary engineers' involvement in the preliminary structural analysis phase as shown in figure 1.3.

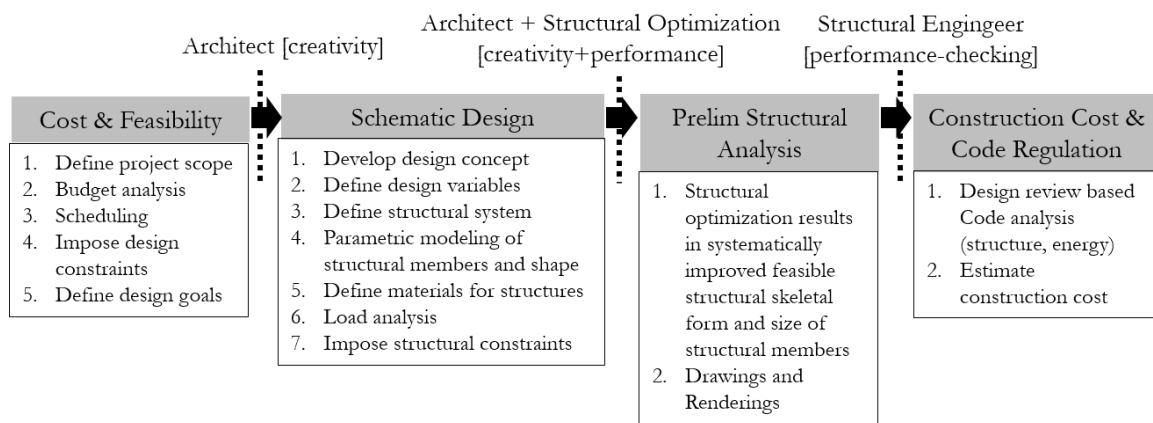


Figure 1.3. Proposed building design workflow

There are five structural optimization methods developed in this research. Each approach uses an interactive visual programming language (Grasshopper) to conveniently generate the parametric model and to parametrically format the structural analysis configuration. Grasshopper is increasingly popular among architects due to its convenience for parametric modeling and thus is considered as the main software component of the developed five methods. The first method, iMAGv1, integrates Grasshopper (Davidson, 2017), finite element analysis software Abaqus (Dassault Systèmes, 2017), and a programming language for technical computing Matlab (Mathwork, 2017) that uses the

coupling of full factorial analysis and deterministic optimization algorithm for performing structural optimization workflow. The second method, iMAGv2, explores an alternative method to iMAGv1's by replacing the optimization algorithm by the discretization and filtering process. The third method, iMAGv3, improves iMAG's workflow to enhance the efficiency through cyclic interoperability between software. The fourth method, iGFM, integrates Grasshopper, customized FEA code, and Matlab's interior point. The fifth method, iGAG, incorporates Grasshopper, customized FEA code, and customized Genetic Algorithm programmed in Matlab. Chapter four discuss the details of these form-finding methods.

This research uses a qualitative method that utilizes Grounded Theory to conduct data collection and analysis for investigating the contemporary design issues and examining the usefulness of the proposed architectural design methodology. Chapter 3 discuss the process for using Grounded Theory in this research. A semi-structured interview is conducted using Clemson architecture students and faculties, and also architectural practitioners. The interview questions are constructed based on the research concern. Each interview is transcribed, and analyze using a qualitative data analysis software MAXQDA (VERBI GmbH, 2017). The findings from the analysis are used to generate theories related to understand further the contemporary design issues and the applicability of the proposed architectural design methodology based on the participants' feedbacks.

#### **1.4 Intellectual merit**

The contributions of this research are as follows:

- This research delivers in-depth responses of architecture faculty, practitioners, and students related to the idea of incorporating structural optimization workflow into the architectural schematic design process. The responses are analyzed and written in the form of a theoretical narrative in chapter 5, 6, and 7 of this dissertation.
- This research develops architectural structural optimization methods that link architectural parametric design software (Grasshopper) with engineering programming and analysis software (Matlab and Abaqus) (chapter 4). The coupling of software is used to improve the communication between architects and engineers and to be able to deal conveniently with complex engineering structural optimization problems (e.g. size, shape, and topology optimization; multimodal optimization; large deformation problem; contact and friction problems; dynamic behavior)
- This research identifies the most appropriate method among the developed and existing methods for the architectural schematic design phase in terms of the operating procedure, computational cost, the convergence of the results, and post-processing capabilities through comparative study (chapter 4) and the analysis of the qualitative interview responses.
- Finally, this research develops guidelines concerning the implementation of the developed architectural, structural optimization methods for the educational purposes and for the utilization in the design practice environment (chapter 8).



## **1.5 Broader impacts**

Architectural structural optimization routines are envisioned to be able to become a surrogate to structural engineer's involvement during the transition from the schematic design to design development phase, and potentially minimize the gap in knowledge for creating a better communication. Common ground is necessary to smoothen the communication in the early design phase and minimize the frustration on both sides. Effective communication (shared information understood by both sides) and efficient communication (quick information transfer) lead to a more cost-effective and time-efficient project. If successfully realized and implemented, the outcome of the collaboration is envisioned to be a design product that is both creative and performance guaranteed.

## CHAPTER TWO

### REVIEW OF LITERATURE

#### 2.1 Case Study of Organic Light Weight Structures

This research is particularly interested in dealing with structural optimization of the lightweight organic structure. Many groundbreaking organic lightweight forms in architecture all over the world have been created thanks to the advancement of construction methods in steel truss systems. Space frames (fixed connection), space trusses (pinned connection) and diagrid systems have been implemented as incredibly large span structures to extremely tall skyscraper. This chapter presents some case studies of such structures designed by well-known architects all over the world.

Zaha Hadid's design of an aquatics center in London is an example of the implementation of truss system for creating a clear 120m long span roof truss. The organic wave-like steel roof truss is only supported by three concrete supports, allowing the design to be column-free on the east and west sides for unobstructed views to the pool (Birch, 2010).

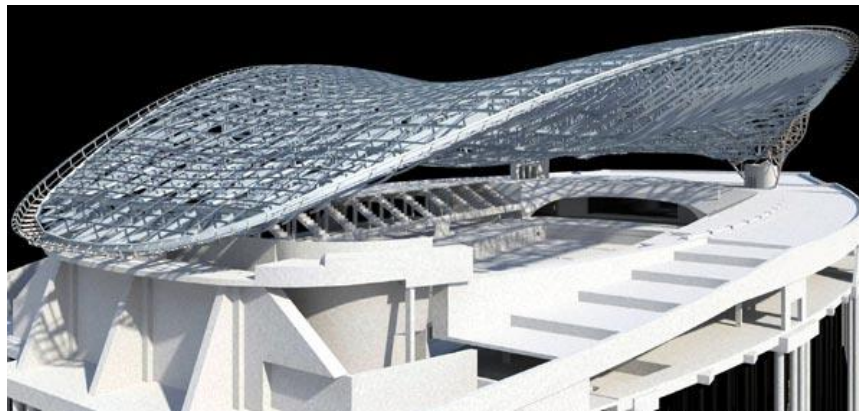


Figure 2.1. London Aquatics Center (Source: Arrow Detailing, 2006)

Fuksas used a double layered space truss for the organic roof design of the Shenzhen Bao'an Airport, which allowed fast erection during the construction phase (Helbig and Kamp, 2014).



Figure 2.2. Shenzhen Bao'an Airport (Source: Welch, 2016)

PTW architect and Arup engineering designed the Beijing National Aquatic Center using a Phelan-Weaire polyhedral system to define the organic geometrical system. The structure uses Vierendeel space frame with fixed connections at the joints. The seemingly random and organic arrangement of the polyhedral modules is actually repetitive to allow mass fabrications. The lightweight ductile space frame is also ideal for the seismic condition of Beijing (Moolman, 2007).

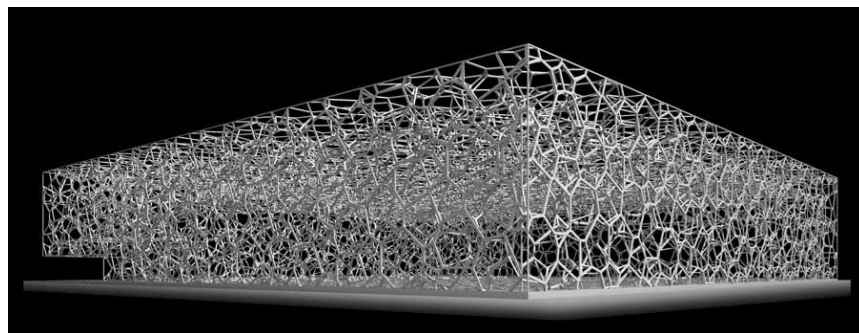


Figure 2.3. National Aquatics Center (Source: Maneval, 2017)

Nicholas Grimshaw's design for the Eden Project interconnects the external hexagonal truss grid with the internal triangular grids. The spatial network of the space truss resembles the molecular organization of silicates in which the crystalline formation has the property of minimal energy paths with minimal material use (Knebel et al., n.d.).



Figure 2.4. Eden Project (Source: Standpoint magazine, 2017)

Norman Foster's design for the St. Mary Axe uses an exterior diagrid structure that provides both bending and shear rigidity for carrying both lateral and gravity loads. The exterior diagrid formation is much more effective for minimizing shear deformation because the members carry shear by the axial action of the diagonal members instead of carrying them by bending of the vertical columns and horizontal spandrels (Moon, 2005). The implementation of the exterior diagrid structural system allows for the construction of an ultra-tall building such as the Lotte Super Tower in Korea (Ali & Moon, 2007).

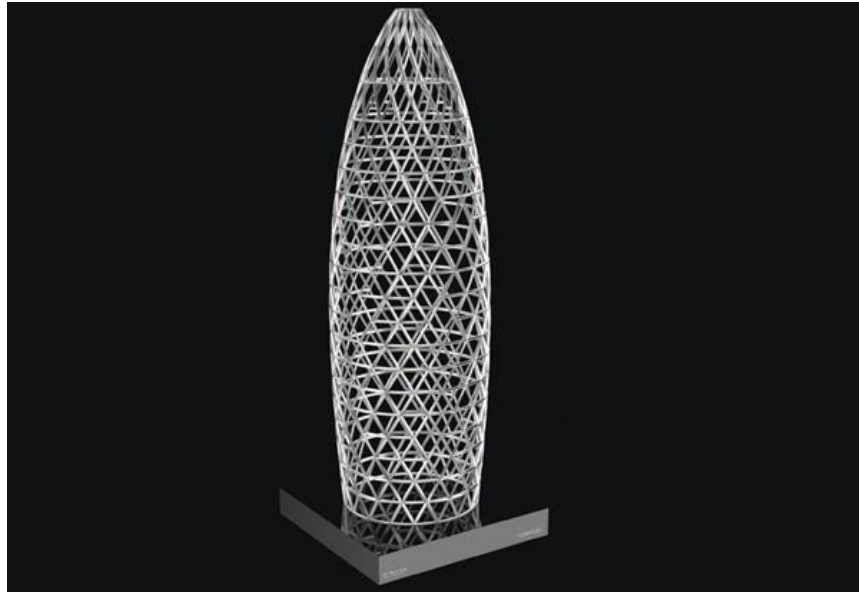


Figure 2.5. Norman Foster St Mary Axe (Source: Indielondon, 2012)

RMJM's design for Capital Gate Abu Dhabi was certified as the "World's furthest leaning man-made tower" (Schofield, 2012). The organic shape leans 18° westwards, and is able to reach the height of 165 meters. The structure uses a double layered diagrid system. The inner diagrid is attached to the core and the steel girders span directly between the external and the internal diagrids. The structural system allows for column-free spaces in the atrium and the thickness of the diagrid members is progressively reduced from the bottom floors (80mm) to the top floors (40mm) to minimize the dead weight (Schofield, 2012).



Figure 2.6. Capital Gate Abu Dhabi (Adapted from Ferradas, 2013)

In line with the sophistication of construction methods and Computer Aided Design (CAD) technology, many architects are intrigued by the artistry of organic lightweight structural systems. While being aesthetically appealing, the utilization of steel truss systems such as diagrid and space grid structures provide benefits that include: the ability to create multipurpose column-free large architectural spaces; use of small elements to allow mass fabrication, handling, and transportation; ease of assembly; light weight to reduce their susceptibility to seismic forces; and an open form that allows easy installation of mechanical and electrical services (Bradshaw, 2002). Through the creative implementation of these systems, architects and structural engineers nowadays are not only able to invent structures that are functional, economical, and efficient, but also capable of creating aesthetically ground-breaking shapes, beyond one can imagine.

These structures have been successfully designed by large renowned architectural firms. Although the buildings are successfully built, the difficulty in the collaboration remains unknown to this research.

This research is particularly interested in creating the link between parametric surface modeling CAD systems and structural optimization methods for designing unconventional lightweight organic structures of any size. The organic structure here is assumed to act as a primary structure that implements lightweight structural systems such as space grids (frame or truss) and diagrid. Figure 2.7 shows the structural features that are considered in this research. For unconventional organic structures, architectural prescriptive codes that prescribe the size of specific structural elements are generally not appropriate, and thus performance-based codes, such as rules that involve structural performance analysis by physical experimentation or numerical simulation, are the only appropriate approach for assessing the integrity of the structure.

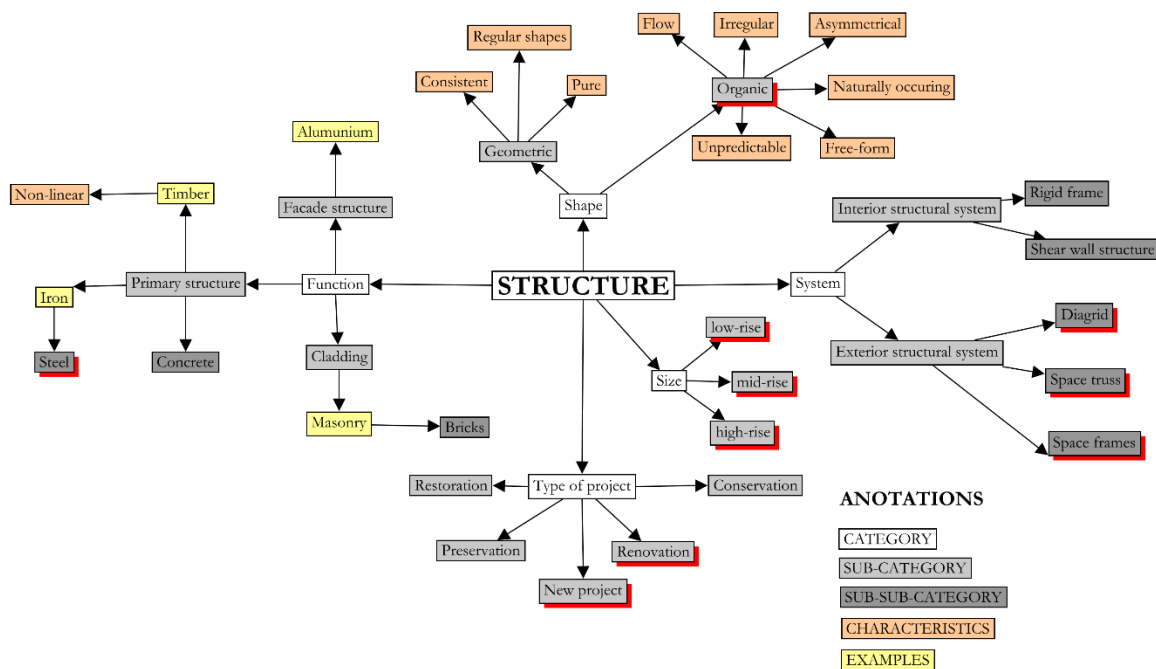


Figure 2.7. Diagram showing the scope of structures. The subset considered in this research is highlighted in red.

## 2.2 Parametric Structural Design and Optimization

Parametric structural design and optimization procedure used to be a complex engineering task in which only highly trained engineers could properly administer the task. Usually, the process involved integrating two or more pieces of software (for examples, see Cazacu and Grama, 2014; Gauchia, 2014). However, the emergence of the architectural research in design computation in 1960 (see Woodbury, 2010) had catapulted the development of various software tools that made parametric design and structural optimization routine accessible to architects. Through the smart integration of these methods, architects can conveniently design a lightweight organic form that is structurally intelligent in the schematic design phase. This chapter presents literature reviews on parametric design, finite element analysis, and optimization.

### 2.2.1 Preprocessing Stage (Rhino/Grasshopper)

Although the term and utilization of CAD in design can be traced back to 1960 (Ross, 1961), the idea of parametric architecture had already been described by Luigi Moretti since 1940 (Bucci and Mulazzani, 2000). Mathematically, the term parametric is defined as a set of equations that express a set of quantities as explicit functions of independent variables, known as parameters (Weisstein 2003). An explicit function is a mathematical function in which the output value is given explicitly in terms of independent variables. This is in contrast to the implicit function in which the outputs are defined in terms of one another.

Equation 2.1. Implicit form for a circle

$$x^2 + y^2 = r^2$$



Mathematically, transforming a problem into its corresponding parametric form requires introducing an additional variable, such as  $\alpha$ , to parameterize the independent variables.

Equation 2.2. Parametric form for a circle

$$r^2(\cos^2 \alpha + \sin^2 \alpha) = r^2$$

Where,

$$0 \leq \alpha \leq 2\pi.$$

However, when using a parametric modeling software, this process of parameterization is done in the background and encapsulated into a function. In the case of a circle above, the user can conveniently input the radius directly into the premade function to construct a circle of radius  $r$ .

Luigi Moretti was the first architect who extensively writes about parametric architecture in the 1940 (Bucci and Mulazzani, 2000). Moretti defines parametric architecture as the study of architecture system in which the goal is to create a relationship between the dimensions that depend on the various parameters (Bucci and Mulazzani, 2000). Thus, in the parametric design process, a set of parameters drive the changes in the form, and relations between geometrical entities are defined by various constraints. In parametric architecture, these parameters and constraints are formulated by architects based on the design requirements, site analysis, performance and aesthetic.

Parametric architecture has recently turned into a more formalized avant-garde architectural style under the name 'parametricism' in which equations dictate forms based on programs and algorithms. Schumacher defines parametricism as the new style

after modernism, while considers post-modernism and deconstructivism only as transitional episodes between those two (Mentegazzi, 2015). Gerber (2007) stated that architectural design is inherently a parametric process and that the architect has always operated in a parametric fashion. All the more, Hudson (2010) said that all design is essentially parametric.

Grasshopper is a plugin for Rhino (Rhinoceros, 2017) that recently gained its fame among architectural academia and practitioners due to its user-friendly parametric modeling capability. Rhino is a CAD application that focuses on producing freeform surfaces, and Grasshopper can be seen as a platform that allows designers to plug and unplug different functions (Brauman et al., 2011).

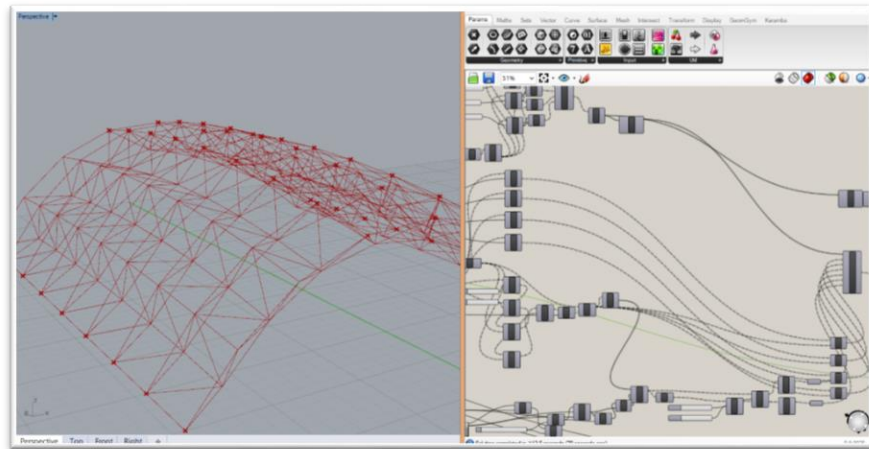


Figure 2.8. Rhino/Grasshopper interface

Recently, Building Information Modelling (BIM) software such as ArchiCAD (Graphisoft, 2017), Autodesk Revit (Autodesk, 2017) and Digital Project (Digital Project, 2015) have been widely used by architects. However, they are not considered as suitable for schematic design as Rhino/Grasshopper. The reason is that architects often dive too deep quickly into the details of construction, cost, and dimensions when using BIM and

cannot easily make changes in the building form. BIM tends to focus more on the details of the design because of its object-based representation instead of raw geometry representation. Although BIM offers closer relation between digital production and realization, Rhino/Grasshopper gives more flexibility for design exploration and is, therefore, more convenient during the schematic design. For instance, although Revit has a broad object library for modeling purpose, it has limitations on parametric rules dealing with angles and does not support complex geometries and curved surfaces (Eastman et al., 2008). Although Digital Project has robust features for modeling and controlling surfaces, new users generally face a steep learning curve to properly use this software (Eastman et al., 2008). These are the reasons for selecting Rhino/Grasshopper for the proposed structural optimization methods.

Grasshopper is considered as user-friendly by many architects due to its geometric visual scripting capability for conducting parametric modeling. As an alternative, the software Generative Component (Bentley, 2017) uses a scripting language for parametric modeling. However, it does not offer the graphic capability of Grasshopper. Also, Grasshopper allows functional customization using script method. The combination of accurate surface representation and geometric visual scripting capability makes Rhino/Grasshopper the ideal tool for design exploration during the schematic design phase.

## 2.2.2 Optimization

### 2.2.2.1 Types of Structural Optimization

There are three types of structural optimization: size, shape, and topology optimization. In size optimization, the domain of the design model and state variables are known a priori and are fixed throughout the optimization process (Bendsoe, 2003). In shape optimization, the purpose is to search the optimum configuration of the domain of the design model such that the shape itself becomes the design variable. Shape optimization is relatively more complicated than size optimization. Finally, topology optimization involves the change of element connectivities and is usually considered as the most complex type of optimization among the three.

The mathematical representation of a structural optimization problem consists of design variables, objective functions, and constraint functions. Geometrical related variables are the design variables. Displacements, stresses, forces, and strains are usually the objective functions representing the goal of the structural optimization (Christensen, 2009). Geometrical or structural quantities such as maximum stress and displacement are commonly used to express the constraint function mathematically. The constraint functions represent the feasible region of the optimal solution.

### 2.2.2.2 Analytical Optimization

Analytical optimization is important in verifying the results obtained from various optimization software. The typical optimization solvers use NLP (nonlinear programming) to deal with several variables and nonlinear constraints. For problems with equalities constraints, it is common to solve the dual problem using Lagrangian multipliers (Rardin, 1998). For problems with mixed equalities and

inequalities constraints, the KKT (Karush-Kuhn Tucker) conditions are commonly used to search for the optimal points. KKT conditions include the consideration of optimality, feasibility, complementary slackness and nonnegativity of the multipliers (Rardin, 1998). The difficulty in extending the Lagrangian stationary conditions to the inequality cases arises in knowing what inequalities will be active at a local optimum. When the inequality is active, then it is treated as an equality and includes it in the Lagrangian (Rardin, 1998). In this case, many inequality constraints in the problem result in many candidates that must be checked with the KKT conditions. This problem becomes cumbersome to solve. Nevertheless, KKT conditions are suitable to be used as the analytical approach for verifying the numerical results of the numerical optimization software (i.e., internal validity check).

#### 2.2.2.3 Black-box Optimization

Black-box optimization is the type of optimization where the analytic form of the objective function or constraint are not known, but can only be given in the form of evaluation process (Ky et al., 2015). Since both the objective and constraint functions are black-box, the gradient and hessian information cannot be analytically calculated to determine whether the problem is a convex, concave, unimodal or multimodal problem. Due to the complex nature of the problem, the black-box optimization problem can only be solved through simulation-based optimization approach. The focus of this thesis is to deal with black-box optimization in which the values of objective and constraint functions are evaluated using Grasshopper and finite element codes.

#### 2.2.2.4 Numerical Optimization (Interior Points)

The interior point algorithm is used in some of the tools developed in this research, namely, iMAGv1, iMAGv3, and iGFM. The interior point method is based on the barrier method. The barrier method adds barrier or the interior penalty function to the objective function such that the search process is penalized when being closed to the boundary. The searching process of the method moves through the interior of the feasible region towards the optimum (Glavic and Wehenkel, 2004). The common penalty functions use a reciprocal function or natural logarithm function weighted by a parameter  $\mu$ . As  $\mu$  decreases, the value of the objective function converges towards the optimum. The KKT condition is then imposed to the modified objective function such that the gradient of the objective function has to equal zero for optimality.

#### 2.2.2.5 Numerical Optimization (Genetic Algorithm)

The developed tool iGAG uses a customized Genetic Algorithm (GA) programmed in Matlab for the optimization. GA is an iterative process that mimics the evolution of living organisms. It consists of encoding the characteristics of a structural design and defining its performance (e.g., maximum displacement, maximum stress, and/or structural weight) as a fitness function that must be maximized. The process starts with a randomly-generated set of individuals (e.g., generally between 20 and 100 structures defined by different sets of design variable values), referred to as the parents. At each iteration, the GA uses the principle of natural selection and stochastic rules for generating new individuals (called offsprings) for the next iteration. The process consists of selecting the fittest parents (i.e., individuals of high fitness values) for mating and

generating offsprings. Iteratively, the population evolves with individuals of increasing fitness values and eventually converges towards an optimum (for more details, see Mitchell, 1998 and Sheta, 2016).

The advantage of the interior point method over GA is that the interior point method is faster when the solution is near. However, GA does not need derivative computation and is more robust when searching for the global optimum such that the global optimum has more chances to be reached even when dealing with complex multimodal functions. GA also can deal with integer programming problems such that it can easily solve structural topological optimization problems.

### 2.2.3 Finite Element Analysis

This section focuses on the theory of Finite Element Analysis (FEA), its complementarity with CAD, and its integration in optimization.

#### 2.2.3.1 Basic Equation in Finite Element Analysis

The Finite Element Method (FEM) is a numerical approach that converts the structural geometry, materials properties, and operating conditions on a set of partial differential equations in an approximate manner. The process also assumes solutions in the form of polynomials.

The basic equation used by FEA for solving structural mechanics problems is  $F = KU$ , where  $F$  is the given external force (usually known),  $K$  is the global stiffness matrix of the structure, and  $U$  is the displacement (often unknown). The domain is discretized into a large number of small elements (i.e., finite elements) whose structural behavior is known and described by a local stiffness matrix  $k$  for each finite element. The local

stiffness matrices are first transformed from the local coordinate system to the global coordinate system and then assembled to form the global stiffness matrix  $K$ . After applying the known boundary conditions, the equation  $F = KU$  is solved numerically by inversion of the matrix  $K$  to determine the displacement vector  $U$ , from which the strains and stresses are determined. Judging the number of reported engineering and design problems solved by FEA, it is undeniable that FEA is a very efficient method to resolve all types of engineering problems.

For large deformation problem, which is a nonlinear analysis, FEM can either use the explicit approach or the implicit approach. For a nonlinear analysis, the load is applied incrementally, and at the end of each increment, the stiffness matrix is updated based on the geometry and material changes (Duczek, 2015). The implicit FEM approach does Newton-Raphson iterations (for details, see DIANA, 2010) to enforce equilibrium of the internal forces with the externally applied load after each increment (Gandhi, 2015). The explicit FEM does not enforce such equilibrium, and relies on increasing the number of small increments to enhance the accuracy, but with higher computational cost. The implicit approach gives much more accuracy for large deformation analysis compared to the explicit. FEA software such as Karamba (Karamba, 2017) uses the explicit approach (Clemens, 2012), while Abaqus/Standard uses the implicit approach by default.

Abaqus is a commercial FEA software developed by Dassault systems and is part of the suite of SIMULIA PLM software tools (Dassault Systèmes, 2017). Abaqus is used in this research as the FEA software to solve structural mechanics problems. Both the



direct stiffness method (for the detail of direct stiffness method, see Felippa, 2000) and the energy method (for the detail of energy method for beam analysis, see Gangamwar et al., 2016) are employed in this research as an analytical approach to verify the results obtained by Abaqus, i.e., internal validity check.

#### 2.2.3.2 Integration of FEA with CAD

Some studies have integrated CAD and FEA software to include analysis in design. For example, the integration of CATIA (a multi-platform software suite developed by Dassault Systèmes (2017)) and MSC NASTRAN (FEA software developed by MSC Software Corporation (2017)) was done by Albers et al. (2007). The initial workflow requires the generation of parametric models (not a mesh model), extraction of all solids' surfaces for setting boundary conditions, meshing the extracted surfaces and generating the volume mesh based on the surface mesh. The intermediate workflow requires controlling the model such as to copy the nodes and volume elements directly to MSC NASTRAN, while the surface elements are used to define boundary conditions. The subsequent workflow includes performing the analysis in MSC NASTRAN (new parameter set is generated) and passing the new parameters to CATIA for automatic updates of the solid model and the associated surfaces. This cycle repeats. This approach of extracting surfaces is also used in all the structural optimization methods developed in this research as a parametric topological selection technique for the boundary and loading condition in Grasshopper.

Al-Haddad (2006) also explored the utilization of Rhino and an FEA software, LS-DYNA (LSTC, 2011), as a form-finding approach. Shape stiffening operations such

as fold, bulge, knot, and subtraction were used to stiffen the structure where significant deflection occurs in the structure of a dome. Several design iterations are done in a view to acquiring the design such that the deflection was minimized based on the observations from FEA. However, in this particular case, the utilization between CAD and FEA is still not computationally integrated. The optimization process is based on the designer's observation instead of responding to the numerical feedback from the FEA.

The Master of Science thesis by the author (Wonoto, 2013a) also experimented with the integration of CAD and FEA by integrating Rhino/Grasshopper and LS-DYNA (Wonoto, 2013b, Baerlecken, 2012, Gentry, 2013). The research was to develop a parametric model of the Miura origami deployable tessellated structure and conducted a structural analysis of the model. This tessellated structure uses the scale of a stadium's retractable roof. Figure 2.9 shows the parametric model and the deflection of the structure. Optimization was done using trial and error. The structural performance of each design permutation was determined by changing folding angles of the origami geometry, and analyzing the deformations using LS-DYNA. From the study, the interoperability from parametric to structural analysis systems is achieved such that all data of nodal coordinates and element construction to materiality are successfully exchangeable.

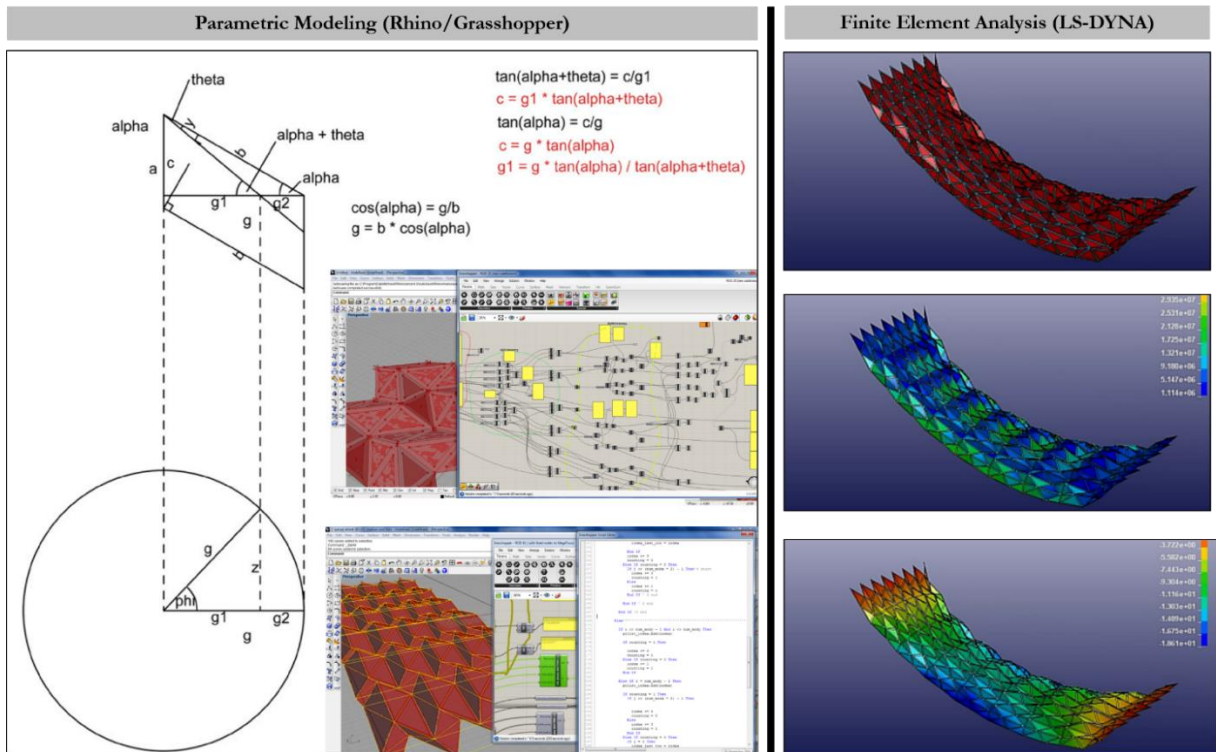


Figure 2.9. M.S. Architecture thesis work (Wonoto, 2013a): parametric structural analysis of deployable tessellated structures

### 2.2.3.3 Integration of FEA with Optimizer

Many studies have experimented on the integration of FEA with an optimizer. Engineers do most of the experimentation on the integration between FEA and optimization. An example of integrating ANSYS (ANSYS Inc, 2017) and Matlab (Genetic Algorithm optimizer) was done by Gauchia et al. (2014) to optimize a complex bus structure that consisted of 2445 beams. At each iteration of the genetic algorithm, Matlab created the FEA input files and called the execution of ANSYS to achieve a minimum in the fitness function using the results from FEA.

The integration of Abaqus and Matlab was done by Anderson (2014) to optimize the ply orientation and stacking sequence of carbon-fiber panels added to the space frame

Formula SAE chassis. In this case, Matlab's `fmincon` (interior point method) function was used as the optimization solver. A python script was used to query the history of the FEA output. These outputs included displacements and reaction forces. These results were tabulated and used in the subsequent optimization loop.

## 2.2.4 Structural Optimization in Architecture

### 2.2.4.1 Structural Optimization in Architectural Practice

An example of structural optimization using prototyping can be seen in Gaudi's work (see Larena, 2009). Gaudi conducted the optimization for the arches of the Convent of Santa by utilizing the notion of the hanging chain model based on the catenary principle. Using this catenary principle, Gaudi produced an optimal arch that closely followed the line of compressive force and became economical in its use of material. For the experiment, Gaudi attached bags of sand to a web of strings, which hung in catenary curves to represent the masses of the static loads of the structure. This was done to reveal the shape of the chapel as a result of the loading. In this case, prototyping and casual inspection were used to optimize the structure intuitively, and the criterion of the optimization was structural performance.

Shape optimization using analytical method can be seen in the design of the British Museum Great Court as seen in figure 2.10. A surface function with a singularity at the boundary was developed mathematically to represent the roof of the museum. A mathematical function defining the surface was developed to satisfy the boundary condition acting as the constraint function in the optimization. The optimization was to find a surface shape that represented a good fit between the architect's early proposal, the

structural loading and bearing criteria. In this case, the optimization problem was solved analytically (Williams, n.d.).

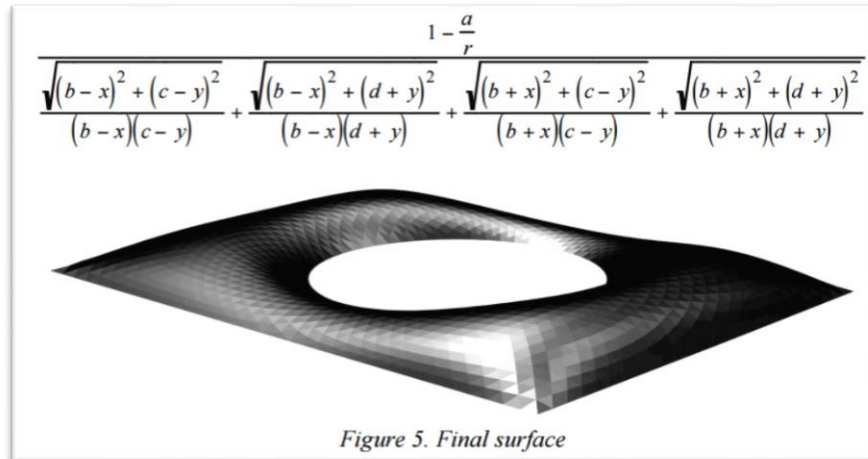


Figure 2.10. British Museum Great Court (Source: Burry et al., 2010)

An example of structural topology optimization can be seen in the design of the Qatar Education City Convention Center (see figure 2.11). The structural optimization problem was solved numerically with the help of the design engineer Mutsuro Sasaki using evolutionary structural optimization (see Pohlheim (2006) for the detail information of evolutionary optimization). An evolutionary structural optimization was conducted to acquire the shapes of the Sidra trees that form the façade of the building. The trees were 'grown' using an optimization solver by iteratively removing the smallest amount of

stressed material from a block until the residual structure used the least amount of material. The objective function of the numerical structural optimization problem was then the weight of the structure to be minimized (Rian et al., 2013).



Figure 2.11. Qatar Education City Convention Center (Source: Hilal, 2010)

Another example of a structural optimization problem solved numerically is the design of the rectangular stadium of Melbourne, Australia by For Cox Architects with the help of the engineering firm Arup. The goal of the project was to create a dome structure that is as light and stiff as possible and also divisible into easily fabricated units (figure 2.12). Specific protocols were developed to ensure the consistency between flexible surfaces and structural lines of force (Burry, J et al., 2010).



Figure 2.12. Rectangular Stadium in Melbourne, Australia (Source: Windley, 2016)

#### 2.2.4.2 Structural Optimization in Architectural Research

Recently many researchers in the field of architecture have been interested in structural optimization for form finding. An example of structural optimization using numerical trial and error optimization can be seen in the work by Al-Haddad (2006) who conducted a parametric study for shape optimization of a dome structure. The structural analysis was carried out using Finite Element Analysis. The optimization consisted of stiffening the locations of the structure where significant deformation occurred by an iterative trial and error process.

Von Buelow et al. (2010) studied the integration of parametric modeling using Generative Component (Bentley Systems, 2017), STaaD (Bentley Systems, 2017), and Genetic Algorithms. Genetic Algorithm was used as a stochastic optimization approach for conducting structural optimization of the tessellated roof with a hexagonal grid pattern.

Other researchers in architecture used a structural optimization method as a form-finding method to design a truss tower (Felkner et al., 2013), large concrete roofs (Sassone and Pugnale 2007) and Voronoi's cell structure (Friederich, 2008).

## **2.3 Constructivism and Grounded Theory**

### 2.3.1 Constructivism

The research uses a qualitative method for evaluating the effectiveness and efficiency of the developed tools for parametric structural optimization in the schematic architectural design phase. Positivism as a research paradigm is commonly used by researchers using quantitative methods and is referred to as a 'scientific method' that is based on the rationalistic, and empirical philosophy originated from Aristotle, Emmanuel Kant and John Locke (Mertens, 2005). On the other hand, constructivism is the paradigm that is used commonly for conducting qualitative research and argues that reality is socially constructed (Cohen & Manion, 1994). In contrast with positivism, constructivism denies the existence of an objective reality (Guba & Lincoln, 1989). Researchers using constructivism as their theoretical framework rely as much as possible on participants' views of the situations being studied (Creswell, 2006 p.8). In the context of research, the paradigm uses qualitative methods such as interviews and coding to generate theories and hypotheses inductively. This method contrasts with the traditional quantitative method in which the primary purpose is to test the hypothesis. This research uses theoretical framework of constructivism as the guiding theoretical framework.



### 2.3.2 Qualitative Research

According to Margaret Myers (2000), qualitative studies are tools used in understanding and describing the world of human experience. Qualitative research is not interested in causal laws but in individual's experiences, belief and meaning from the perspective of the people (Brink, 1993). A qualitative method is suitable for this research since it investigates the experiences of the individuals who go through the process of design and use the developed tools. According to Coleman and O'Connor (2007), qualitative research usually deals with open-ended questions such as “why” and “how” and thus seems to be suitable for this software development and testing research.

### 2.3.3 Grounded Theory

Grounded Theory is used in this research for data collection and analysis. The term Grounded Theory was originated by Glaser and Strauss (1967) as “the discovery of theory from data that is systematically obtained and analyzed.” Grounded Theory is an inductive methodology (Grounded Theory Institute, 2013). Research using Grounded Theory generates a theory that is kept under cycles of revisions until no revision is further required, thereby reaching theoretical saturation.

In this study, the interview feedbacks from the participants are coded to conceptualize the findings. The coding process of the Grounded Theory used in this research follows Strauss and Corbin levels of coding. The three levels of coding include open coding, axial coding and selective coding (Strauss and Corbin 1998). Open coding breaks down interviews and observations into distinct units of meaning which are labeled to generate a concept (Coleman and O'Connor, 2007). Some of the techniques for the

open coding process include descriptive code (summarizes the topic) and in vivo code (taking the spoken word of the participant directly as a code) (Saldana, 2009). Repeated words, surprising opinions, and statements that are considered to be important are some aspects that are used in the coding method. Similar codes are then grouped together to form themes. This grouping is based on the properties and dimensions of the codes. For instance, color has properties of intensity and hue, and the property of intensity can vary from low to high. Axial coding involves making connections between themes deductively based on the properties and dimensions of each theme. Selective coding involves creating a core category (theoretical construct) and relating other categories to that core category to make a meaningful story or theoretical narratives (Hendrick, 2013).

Qualitative data collection and analysis is a nonlinear cyclic process. Following Auerbach and Silverstein (2003), the procedure can be divided into six cyclic steps including developing a research concern, labelling the relevant text from the interview transcript with the research concern in mind, discovering the repeating ideas (this process involves open coding), grouping repeating ideas into themes (this process includes axial and selective coding), developing theoretical construct by combining themes and literature review to build a more abstract concept, and creating a theoretical narrative by retelling the participant's story in terms of the repeating ideas, themes, and theoretical construct. The subsequent cycles follow the similar process. However, they use the previously generated themes and theoretical construct for developing the new research concern, repeating ideas, themes, and theoretical construct.

Theoretical sampling is discussed in the later section. The process of theoretical sampling and qualitative data analysis cycles iteratively until the research concern is apparent and the theoretical framework no longer changes. The cycle keeps continuing until additional research samples do not add any new information to the understanding of the theoretical constructs, reaching theoretical saturation.

#### 2.3.4 Justifiability and Transferability

The approach of the qualitative research leads to hypothesis-generating research, which is different from the traditional approach of quantitative research that leads to hypothesis-testing (Auerbach and Silverstein, 2003). Instead of dealing with the reliability and validity, the qualitative research deals with the justifiability of interpretation. For a scale to be objective, it must satisfy both reliability and validity. Reliability is determined if the two or more measures have the same or almost the same results. Having the result of *satisfactory* from two survey for responses, for instance, proves reliability. However, the value of *satisfactory* for each person must be different, and ,moreover, it is impossible to know the true value of *satisfactory* especially when dealing with the subjective experience of individuals. This is the reason why Auerbach and Silverstein (2003) proposed the concept of justifiability as an alternative to the idea of validity and reliability.

Rubin and Rubin (1995) proposed three criteria for explaining justifiability. These includes transparency, communicability, and coherence. To realize transparency in qualitative research, the interpretation of the data has to be made transparent to avoid the impression as if the analysis was fake or not trustworthy (Auerbach and Silverstein,

2003). To ensure communicability, the developed themes and theoretical constructs must be communicable to other people (Auerbach and Silverstein, 2003). Last but not least, the coherence can be achieved by ensuring that the developed theoretical constructs fit together and allow the researcher to tell a coherent story (Auerbach and Silverstein, 2003).

While quantitative research seeks generalizability to a population, the qualitative research seeks transferability of the theoretical construct (Auerbach and Silverstein, 2003). The argument is that it is impossible for a researcher to determine if the samples taken are representative enough for the purpose of generalization. The transferability seeks to extend the theoretical construct beyond the sample while maintaining that the themes and repeating ideas are applicable only to the sample under the study.

### 2.3.5 Theoretical Sampling

Theoretical sampling refers to the simultaneous data collection and analysis in Grounded Theory (Strauss and Corbin, 1998). Theoretical sampling involves the iterative process of collecting codes, analyzing data, and making a decision on what data to collect next and where to find such data. Thus, the unique responses acquired from the interview process can be used to modify the questionnaires or target population to get a more qualitative understanding for the future interviews. This theoretical sampling process can thus be used to narrow further down the target population and type of questions to be addressed in subsequent interviews.

### 2.3.6 Semi-Structured Interview

A semi-structured interview is used in this research to validate the effectiveness and efficiency of the developed tools. Bowers (1988) argues that structured interview schedules are inappropriate for grounded theory studies. A semi-structured interview, also known as interview guide approach, is used in this research because it is a flexible method of interview. This approach allows new ideas to be brought up during the interview process based on the responses from the interviewees, while at the same time allowing the interview to stick to the main theme (Berg, 2009). This method is also suitable for a one-time interview with a single interviewee (Bernard, 2006). The use of semi-structured interviews is therefore also congruent with Grounded Theory methodology as it allows the researcher to ask fundamental questions, in the same way, each time but allows flexibility in the sequencing of questions and the depth of exploration (Fielding 1994).

### 2.3.7 Qualitative Method for Software Testing Research

Few research projects have used qualitative research methods for software testing purpose. Qualitative methods such as Grounded Theory are suitable for software testing purposes because they provide rich data for analysis and concept formation for further improvement of the software even with a small number of samples (Konka, 2011). Bertelsen (1997) also argues that Software Engineering research, which includes software development and testing, must include a way to understand psychological, social and cultural phenomena. Using Konka's and Bertelsen's arguments, the research assumes that quantitative methods such as using the Likert scale (Auerbach and Silverstein, 2003) for

operating on the feedback from the interviewing process could eliminate important details of the responses from the interviewees. These helpful responses might include psychological aspects of the respondents when using the developed tools. Thus qualitative research is considered to be more appropriate for this tool development and testing research compared to quantitative research.

## CHAPTER THREE

### RESEARCH DESIGN

#### **3.1 Qualitative Research Method**

The qualitative method used in this research is based on the Grounded Theory. The purpose of the qualitative method in this research is to examine the importance of the early architectural structural optimization as a form-finding technique. iMAGv1 was used for the interview purpose. The qualitative research method is used to generate the themes and theoretical construct corresponding to each of the research concerns in each cycle of the interview.

#### **3.2 Grounded Theory for Development of Tools**

The research workflow show in Figure 3.1 follows the Auerbach and Silverstein (2003) suggested research model for qualitative research using the grounded theory. The model was also adjusted to include the steps involved in developing the software. The workflow starts with literature review related to current collaboration process between architects and structural engineers, and the existing implementation of parametric structural optimization in architecture. The five structural optimization methods were developed to resolve the contemporary issue related to the collaboration between architects and structural engineers. The research concern are designed accordingly. It must be noted that there are two versions of Grounded Theory. The one developed by Glaser (1978) argues that researchers should only review the literature after theories have emerged to avoid prior reading that may influence emerging theories. On the other hand,

the one developed by Strauss and Corbin (1998) argues that it is more realistic that researchers have some prior knowledge in the field of study before entering it, and that research concern need to be pre-set in order to give researchers boundaries and to guide the research process. This research implements the Grounded Theory proposed by Strauss and Corbin (1998). Based on the research concerns, purposive sampling is conducted for determining the interviewees for the first cycle of interviews. The interview questions are then organized based on the research concerns and how iMAGv1 works for demonstration purpose during the interview. The interview is then conducted and transcribed. The coding process then follows. The green colored blocks in figure 3.1 show the steps taken in the coding process of the interview transcript. The cycle repeats until the theoretical saturation is achieved. For the subsequent cycles, the construction, and organization of the research concern, repeating ideas, themes, theoretical construct and theoretical narratives are based on the previously developed themes and theoretical constructs.



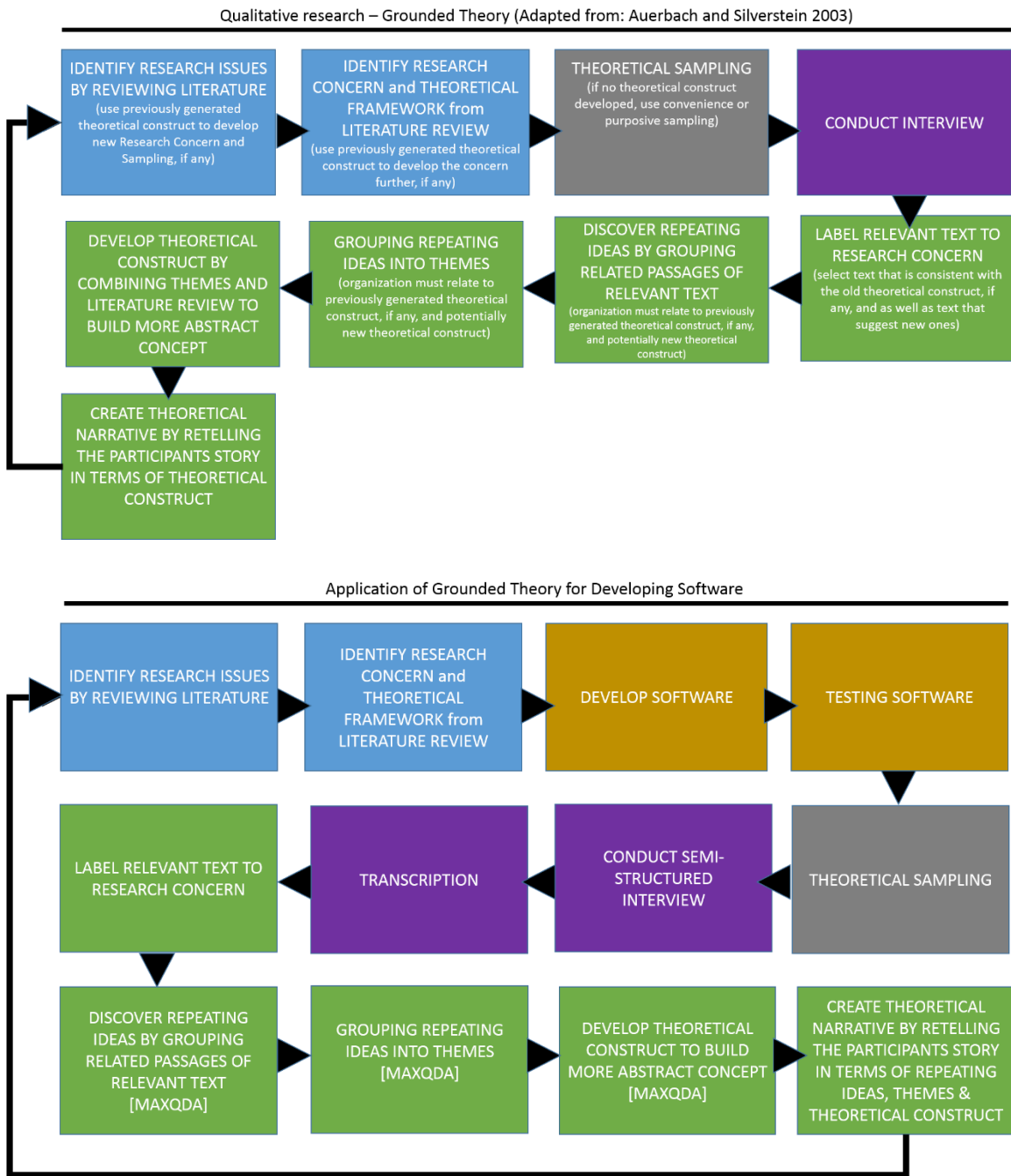


Figure 3.1. Grounded Theory model used for developing and testing the developed structural optimization methods

### **3.3 Grounded Theory for Data Collection**

The purposive, snowball and theoretical sampling are used as the data collection method. Purposive sampling is used only as an initial sampling method in the first cycle of interviews. The targets for the sampling process are students and faculty at Clemson University from the year 2016-2017 who are familiar with the concept of parametric modeling and are familiar with Rhino/Grasshopper. Theoretical sampling is then used for the subsequent cycles of interviews to revise the sampling targets based on the previously generated theoretical construct and additional literature review if any. The sampling process stops when reaching theoretical saturation. In addition, snowball sampling can also be occasionally used. Snowball sampling identifies respondents who are then asked to refer researchers on to other respondents who possess some characteristics that are of research interest (Biernacki and Waldorf, 1981). The snowball sampling method uses an interviewee to select the next sample.

### **3.4 Grounded Theory for Semi-Structured Interview**

For this research, the semi-structured interview process consists of three parts including initial background, education module, and demonstration module (see Appendix B for the interview module transcripts).

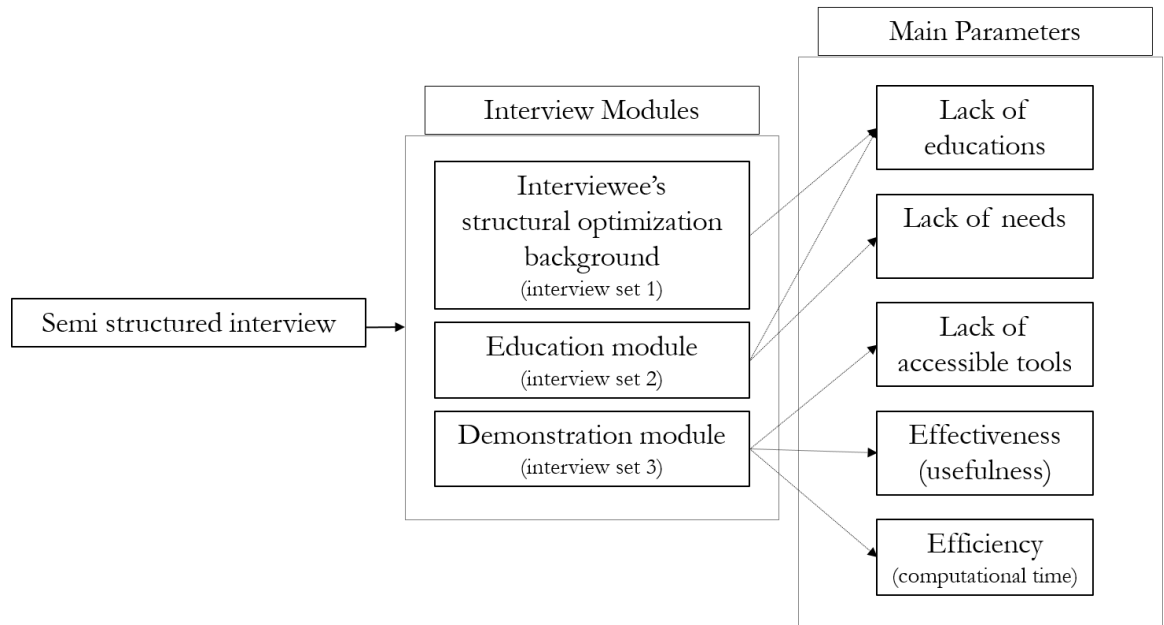


Figure 3.2. Semi-structured interview modules and main parameters

The first module of the interview is to conduct an examination of interviewee's background knowledge and experience in architectural parametric structural optimization. This module introduces the difference between prescriptive and performance-based approach in structural analysis. This introduction is used to contextualize the research. This research is contextualized under the assumption that architects need the freedom of designing complex organic shapes for which prescriptive codes and rules of thumb are not applicable. A performance-based approach is a more realistic approach in this case. The background module is also to find out the importance of structural evaluation as a form-finding method in the architectural design process based on the participant's opinion.

The second module of the interview is to introduce participants with the concept of structural optimization briefly. The education module uses the same recorded video for

each interview for objectivity purpose. The duration of the recorded video is 4 minutes and 26 seconds. This video briefly introduces participants to the concept of structural optimization. A practical example is used to explain the structural optimization concept. Further questions are asked to examine the respond of the participant after watching the video.

The tool demonstration module is used to demonstrate how the developed tools work. The demonstration module uses the same recorded video for each interview for objectivity purpose. The demonstration video includes the step by step process of using the tool following the developed user manual (see Appendix A for the manual). This process is followed by a set of questions used to address research concern 3.

In every interview, the questions and findings are recorded using a recording device, transcribed, analyzed and written into a report format. Table 3.1 shows the general format used in every cycle of the interview.

Table 3.1. General format for interview questions

Interview Cycle: GENERAL FORMAT	
Theoretical Construct	The theoretical construct is based on the theoretical construct generated from the data analysis of the previous interview
Research Concern	This is the Research Concern, and changes in every cycle of interview based on the old generated theoretical constructs and review of literature
Sampling	This sampling method depends on the Research Concern
Sampling Target	The target depends on the sampling method and research concern
Interview Questions	--
IQ	Interview Questions are based on the research concerns

### 3.5 Grounded Theory for Data Analysis

Coding is used for data analysis of the qualitative interview. This process is illustrated in figure 3.1 (labeled in green blocks). A qualitative software MAXQDA is

used to help code the text for developing the theoretical construct, which is then used for preparing the research concern and organizing themes for the subsequent studies. The final form of the theoretical narrative is the end product of the research and is in the form of a story reflecting the participants' subjectivities related to the research concern in each cycle of the interview.

Table 3.2 shows the coding format used to organize the findings at each cycle of the interview.

Table 3.2. General format for organizing qualitative data

Interview Cycle: GENERAL FORMAT	
<b>THEORETICAL CONSTRUCT</b>	<b>memo</b>
Themes	memo
Repeating ideas Repeating ideas Repeating ideas	memo
Samples of relevant text	memo
<b>Theoretical Construct</b>	<b>memo</b>
Themes	memo
Repeating ideas Repeating ideas Repeating ideas	memo
Samples of relevant text	memo

### 3.6 Grounded Theory for Justifiability and Transferability

Transcription of the interview is made, and the corresponding coding process is presented to satisfy the transparency requirement of the justifiability of the interpretation. Sampling targets are kept to be refined based on the research concern, old themes and theoretical constructs in each cycle of the interview to ensure transferability of the theoretical construct.



## CHAPTER FOUR

### TOOL DEVELOPMENT

Five methods were developed in this research to deal with any black-box structural optimization problems of space frames and trusses. iMAGv1, iMAGv2, iMAGv3 and iGFM deal with nonlinear programming problems and iGAG can deal with mixed integer nonlinear programming problems. In addition, the existing tool, referred to as iGKG, which stands for integrated Grasshopper/Karamba/Galapagos, is the architectural structural optimization tool currently available to architects and is explored in this research for the purpose of comparison.

#### 4.1 iMAGv1

This section discusses the components and workflow of the iMAGv1 tool developed in this research. The tool is the integration of three pieces of software: (1) Grasshopper for structural parametric modeling, (2) Abaqus for finite element analysis, and (3) Matlab for file management and optimization.

##### 4.1.1 Workflow of iMAGv1

The workflow of iMAGv1 is an integrated scheme that includes two phases. The first phase is for Matlab to interact with Grasshopper and Abaqus iteratively to conduct a multitude of structural analyses for all predefined parametric values. This full factorial analysis process generates the results of maximum nodal displacements and stresses in a tabulated format. The second phase is for Matlab to generate a type of response surface

from the previously generated data. This response surface is a smooth continuous function (i.e., cosine interpolation) of the design variables for Matlab to be able to use the interior point algorithm for the optimization process. By coupling the interpolation and interior point method, iMAGv1 can also deal with topological optimization. This topological optimization allows changes in the number of nodes in the geometry during the optimization process.

iMAGv1 is inefficient because it requires generating a database of analysis results before finding the optimum. The full factorial analysis takes significant computational time depending on the complexity of the structure and the number of levels for each design parameter. This method was initially used for its simplicity in programming the interaction between the three software, Matlab, Abaqus, and Grasshopper. IMAGv3 was developed afterward to perform cyclic interoperability between the Matlab optimization solver and the other two software.



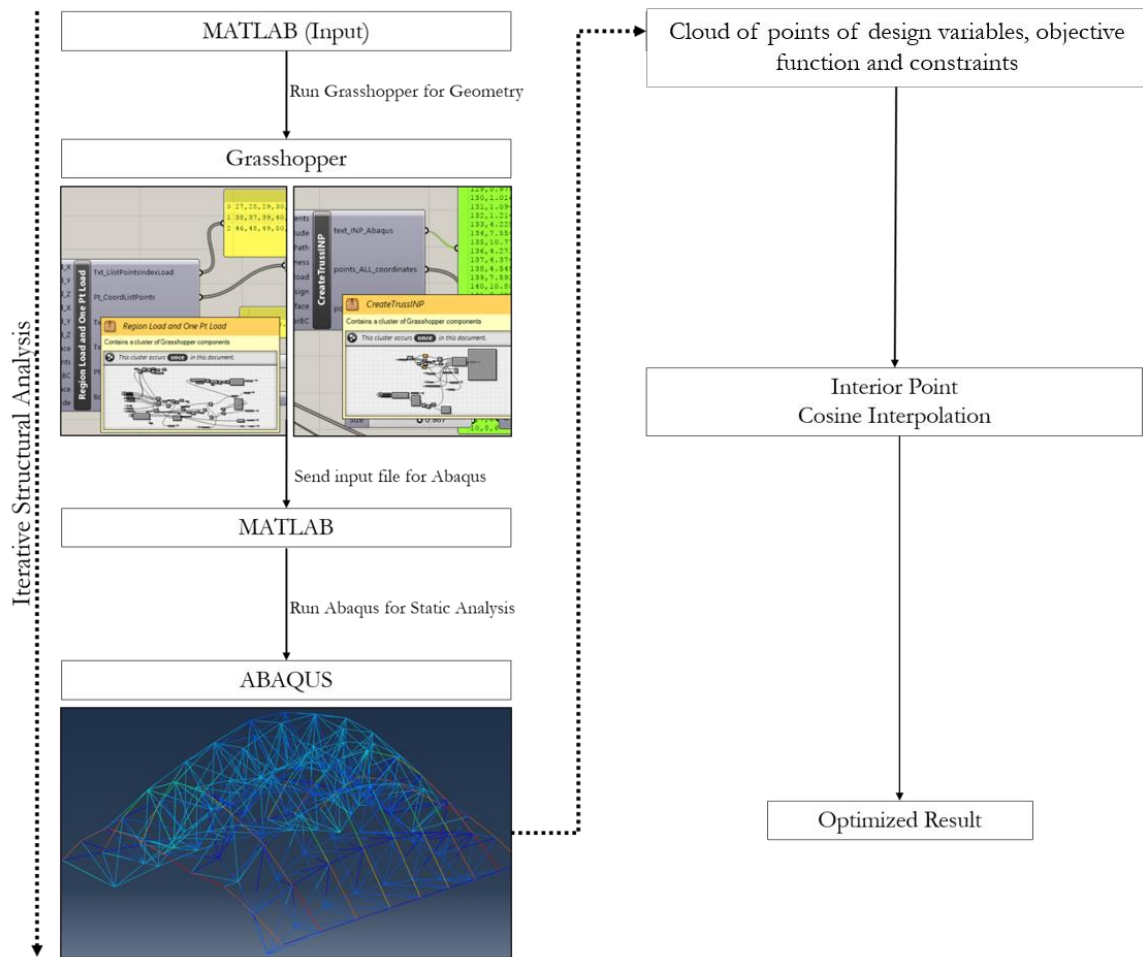


Figure 4.1. Integration scheme of iMAGv1

#### 4.1.2 Grasshopper for Parametric Modeling

Four components (shown in figure 4.2) were created in Grasshopper to realize its integration with Abaqus and Matlab. Each of these components contains several sets of hierarchical components coupled with a scripted component.

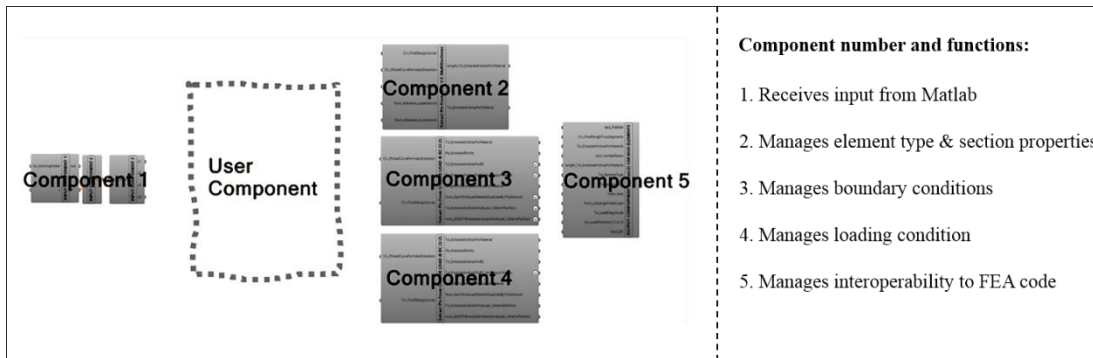


Figure 4.2. Grasshopper premade components

As shown in figure 4.2, the user-defined component, which defines the structure, must be inserted by the user. The first premade component must be located at the beginning of the Grasshopper definition as it takes the numerical inputs from iMAGv1's main interface and passes them to the user-defined component as design variables. The fifth premade component must be located at the end of the Grasshopper definition to manage element connectivity of the user-defined truss model and creates an input file for Abaqus.

The second, third and fourth premade components are optional. The second component is for managing various section properties through parametric selection process of topological entities. The third and fourth components are designed to adjust the boundary and loading conditions at nodes by filtering the nodes on the basic design curves or around the surrounding edges of the basic design surface. These methods are similar to the method used by Albers et al. (2007) previously discussed in Section 2. These methods extract the topological information for setting up the boundary and loading conditions. The three methods allow for topological optimization. However, if no basic design curve or surface are used, then the nodal indices of the structure have to be

specifically defined for applying the boundary conditions and loads. In this case, only sizing and shape optimization are realizable, and these components is not used.

#### 4.1.3 Abaqus for Finite Element Analysis

Abaqus is used in iMAGv1 for conducting the structural analysis. Grasshopper generates the input for Abaqus. This input includes geometrical properties (i.e., nodal coordinates, element connectivity, and cross-sectional properties) and analysis information (i.e., type of analysis, boundary and loading conditions). The output includes the nodal displacements and stresses. The values of the nodal displacements and stresses can be used in the objective and constraint functions of the optimization problem.

#### 4.1.4 Matlab for File Management and Optimization

Matlab is used in iMAGv1 for managing the interactions between Grasshopper and Abaqus, and also to conduct the optimization using the interior point algorithm.

At each analysis, Matlab runs Grasshopper in the background. The numerical inputs of the design variables are set in Matlab by the user. These inputs are used by Grasshopper to generate the geometry of the structure. The geometrical information and analysis setup are passed back to Matlab in the form of a text file that is used by Matlab as input for running Abaqus in the background. Then Abaqus passes the nodal displacement and stress results back to Matlab, and this cycle continues in an iterative manner. Matlab stores all the results from those iterations for conducting optimization using the interior point algorithm.

#### 4.1.5 Internal Validity of iMAGv1

The validation process involves cross-checking the results of the iMAGv1 tool with analytical solutions and the results obtained from various commercial software. The purpose of this validation is to ensure the reliability of the iMAGv1 tool.

##### 4.1.5.1 Validation of Matlab Optimizer

A tessellated structure as shown in figure 4.3 was used to verify the optimization result obtained from the iMAGv1 tool. The tessellated structure is composed of two-dimensional hexagonal and quadrilateral shells connected at particular edges to form a rectangular grid surface.

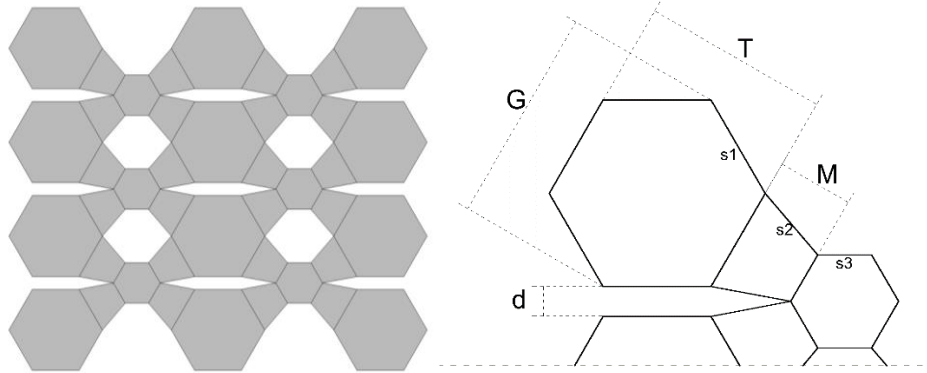


Figure 4.3. Example tessellated structure

Analytical analysis using the KKT conditions and numerical analysis using the MINOS solver in AMPL were conducted (AMPL Optimization inc, 2013). MINOS uses a projected Lagrangian approach to deal with the nonlinear constraints in which a linear approximation to the nonlinear constraints is constructed around the solution at each major iteration, and the Lagrangian and penalty terms are added to modify the objective (AMPL-MINOS, n.d.). AMPL is an algebraic modeling language for describing and solving high-complexity problems for large-scale mathematical computing.

For the geometrical optimization, the objective function is to minimize the surface area of the solid module shaded in gray in figure 4.3. Three nonlinear geometrical constraints imposed by the optimization model. The first constraint is made such that the width of the whole structure must equals 20. The second constraint is made such that the distance  $d$  between hexagons should be less or equal to  $s_1$ . The third constraint is made such that the sum between  $s_1$  and  $s_3$  must be greater or equal to  $s_2$  to maintain the proportionality between modules. Below is the optimization model, and table 4.1 shows that the result using the three methods (KKT, AMPL, and iMAGv1) converge to the same solution. The optimization problem is defined as follows:

Equation 4.1. Optimization model of the tessellated structure

$$\begin{aligned}
 \min : & 12 \left[ \frac{3\sqrt{3}}{2} s_1^2 \right] + 6 \left[ \frac{3\sqrt{3}}{2} s_3^2 \right] + 24 \left[ \sqrt{(u-s_1)(u-s_2)^2(u-s_3)} \right] \\
 \text{st: } & 6 \cos(\pi/3) s_1 + 3s_1 + 2s_3 + 4s_2 \cdot \cos \left[ \frac{2\pi}{3} - \cos^{-1} \left[ \frac{s_1-s_3}{2s_2} \right] \right] - 20 = 0 \\
 & -2s_2 \sin \left[ \frac{1}{2} \left( 2 \left[ \cos^{-1} \left( \frac{s_1-s_3}{2s_2} \right) \right] - \frac{2\pi}{3} \right) \right] + s_1 \geq 0 \\
 & s_1 + s_3 - s_2 \geq 0 \\
 & s_1 s_2 s_3 \geq 0
 \end{aligned}$$

The KKT conditions can be generated as follows:

Equation 4.2. Optimality condition

$$\frac{\partial L}{\partial s_1} = 0, \quad \frac{\partial L}{\partial s_2} = 0, \quad \frac{\partial L}{\partial s_3} = 0, \quad \frac{\partial L}{\partial \lambda_1} = 0,$$

Equation 4.3. Complementary slackness

$$\begin{aligned}
 \lambda_2 \left( 2s_2 \sin \left[ \cos^{-1} \left[ \frac{s_1-s_3}{2s_2} \right] - \frac{\pi}{3} \right] - s_1 \right) &= 0, \\
 \lambda_3 (s_2 - s_1 - s_3) &= 0, \\
 \lambda_4 (s_1) &= 0, \quad \lambda_5 (s_2) = 0, \quad \lambda_6 (s_3) = 0,
 \end{aligned}$$

Equation 4.4. Non-negativity conditions

$$\lambda_2 \geq 0, \quad \lambda_3 \geq 0, \quad \lambda_4 \geq 0, \quad \lambda_5 \geq 0,$$

and  $\lambda_6 \geq 0$

The results from the KKT, AMPL, and iMAGv1 are listed in Table 3.

Table 4.1. Comparing analytical and numerical optimization results

Methods/Optimum	iMAGv1	AMPL	KKT
Optimum S1	1.745	1.745	1.745
Optimum S2	2.576	2.576	2.576
Optimum S3	0.832	0.832	0.832
Minimal Area	184.040	184.041	184.041

As seen in Table 4.1., the results of the three methods are sufficiently close to demonstrate the validity of the optimization algorithm used in iMAGv1 (interior-point algorithm).

#### 4.1.5.2 Validation of Abaqus

For verifying the results from Abaqus, the tessellated structure was simplified into a statically indeterminate non-prismatic beam as shown in Figure 4.4.

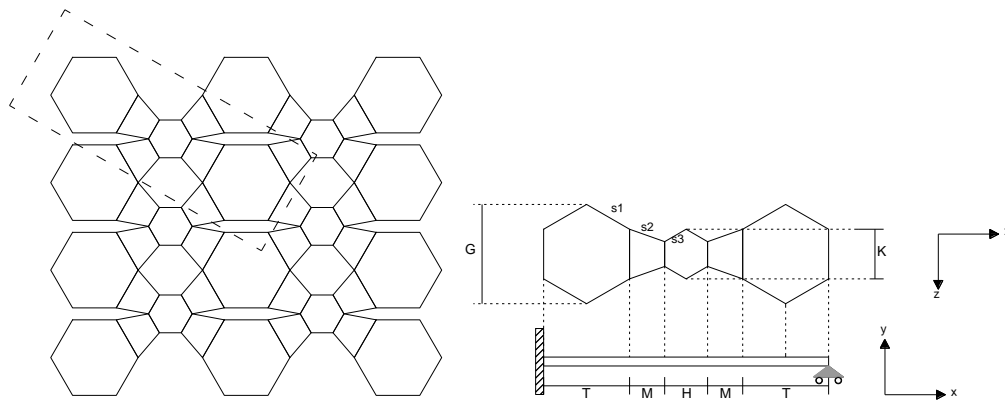


Figure 4.4. Portion of the tessellated structure taken as a statically indeterminate non-prismatic beam problem

The closed form expression for the deflection in the middle of the beam can be derived using the energy method (Castigliano theorem) by initially converting the structure into a statically determinate non-prismatic beam.

Equation 4.5. Solving the redundancy for statically indeterminate non-prismatic beam

$$\left( \int_0^{\frac{T}{2}} \frac{Mz \left[ \frac{\partial Mz}{\partial Q} \right]}{I_1} + \int_{\frac{T}{2}}^T \frac{Mz \left[ \frac{\partial Mz}{\partial Q} \right]}{I_2} + \int_T^{T+M} \frac{Mz \left[ \frac{\partial Mz}{\partial Q} \right]}{I_3} + \int_{T+M}^{T+M+\frac{H}{2}} \frac{Mz \left[ \frac{\partial Mz}{\partial Q} \right]}{I_4} + \int_{T+M+\frac{H}{2}}^{T+M+H} \frac{Mz \left[ \frac{\partial Mz}{\partial Q} \right]}{I_5} + \int_{T+M+H}^{T+2M+H} \frac{Mz \left[ \frac{\partial Mz}{\partial Q} \right]}{I_6} + \int_{T+2M+H}^{\frac{3T}{2}+2M+H} \frac{Mz \left[ \frac{\partial Mz}{\partial Q} \right]}{I_7} + \int_{\frac{3T}{2}+2M+H}^{2T+2M+H} \frac{Mz \left[ \frac{\partial Mz}{\partial Q} \right]}{I_8} \right) dx = 0$$

The derived formula below is used to find the deflection in the middle of the beam.  $Mz$  is the internal bending moment of the beam, and  $mz$  is the partial derivative of  $Mz$ .

Equation 4.6. Solving the mid-displacement for statically indeterminate non-prismatic beam

$$v_B = \frac{1}{E} \left[ \int_0^{\frac{T}{2}} \frac{Mz[mz]}{I_1} + \int_{\frac{T}{2}}^T \frac{Mz[mz]}{I_2} + \int_T^{T+M} \frac{Mz[mz]}{I_3} + \int_{T+M}^{T+M+\frac{H}{2}} \frac{Mz[mz]}{I_4} \right] dx$$

Where:

$$I_1 = h^3 \left[ \frac{2}{T} (G - s_1)x + s_1 \right] / 12 \quad \text{for } 0 < x < T/2$$

$$I_2 = h^3 \left[ \frac{2}{T} (s_1 - G)x + 2G - s_1 \right] / 12 \quad \text{for } T/2 < x < T$$

$$I_3 = h^3 \left[ \left( \frac{s_3 - s_1}{M} \right) x + s_1 - \left( \frac{s_3 - s_1}{M} \right) T \right] / 12 \quad \text{for } T < x < T+M$$

$$I_4 = h^3 \left[ \frac{2(K-s_3)}{H} x + s_3 - \frac{2(K-s_3)}{H} (T + M) \right] / 12 \quad \text{for } T+M < x < T+M+H/2$$

The results from iMAGv1, which come from Abaqus, and the analytical results were found to be approximately the same. Abaqus results are about 5% larger than the analytical results. This difference occurs since the shear terms in the deflection are

neglected for computational efficiency. Using this comparative study, the internal validity related to the instrumentation of Abaqus is verified.

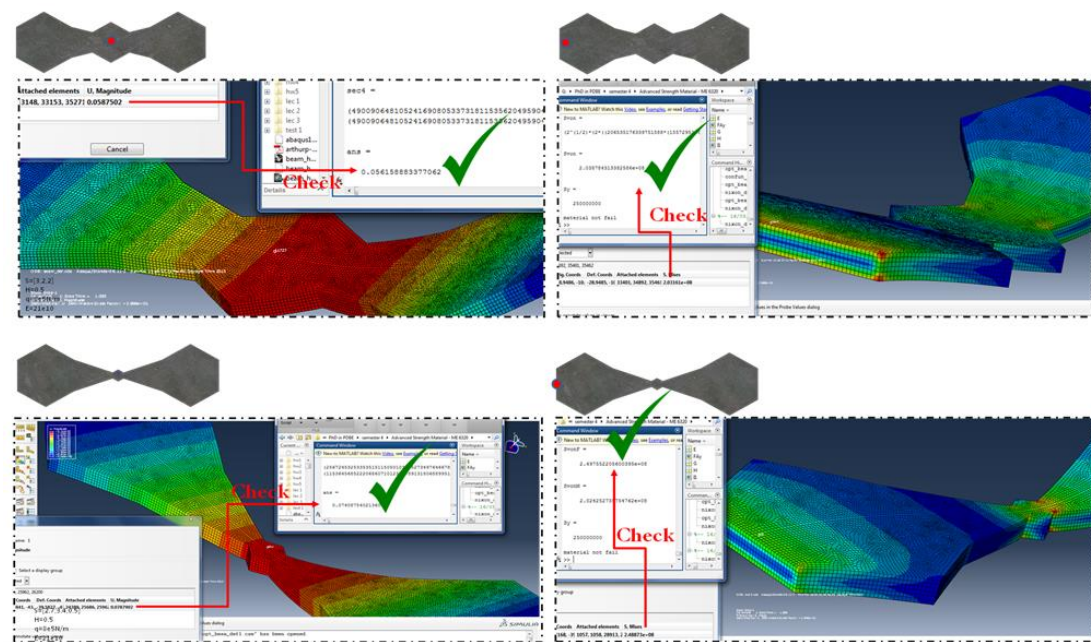


Figure 4.5. Cross-checking between Abaqus and analytical analysis results

Table 4.2. Comparing analytical and numerical displacement results (non-prismatic beam)

Methods/Mid-displacement	Mid-displacement permutation 1 (m)	Mid-displacement permutation 2 (m)
Analytical Method	0.056	0.074
ABAQUS	0.058	0.078

Table 4.3. Comparing analytical and numerical stress results (non-prismatic beam)

Methods/Stress at Boundary	Boundary Stress permutation 1 (MPa)	Boundary Stress permutation 2 (MPa)
Analytical method	2.038	2.498
ABAQUS	2.032	2.488



#### 4.1.5.3 Validation of Karamba

Karamba (Karamba, 2016) is an FEA plugin in Grasshopper. This section conducts a comparative study of the results between Karamba, Abaqus, and direct stiffness method. A simple planar steel frame was modeled in xz-plane (figure 4.6) with seven members, and with an applied load of 1000 lbf at node 3 in the negative z-direction. Node four and five are fully constrained.

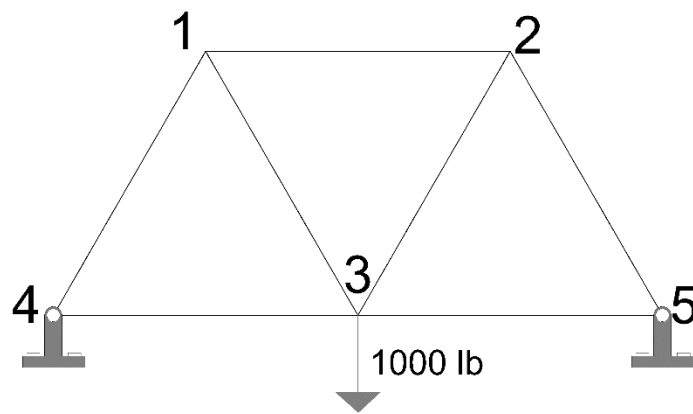


Figure 4.6. Node numbering on the frame model for verification purpose

Figure 4.7 shows the Karamba result for the frame structure using beam elements.

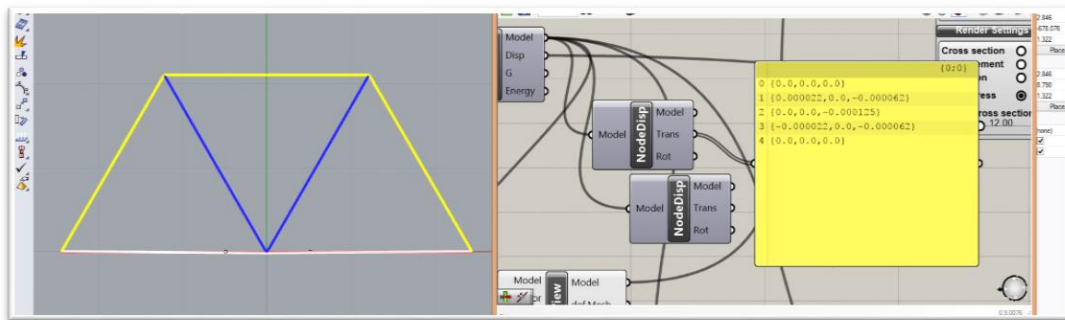


Figure 4.7. Karamba results for the frame model

Figure 4.8 shows the Abaqus results for the frame using B31 beam elements. The value shown in the figure is the deflection of node 3 in the z-direction.

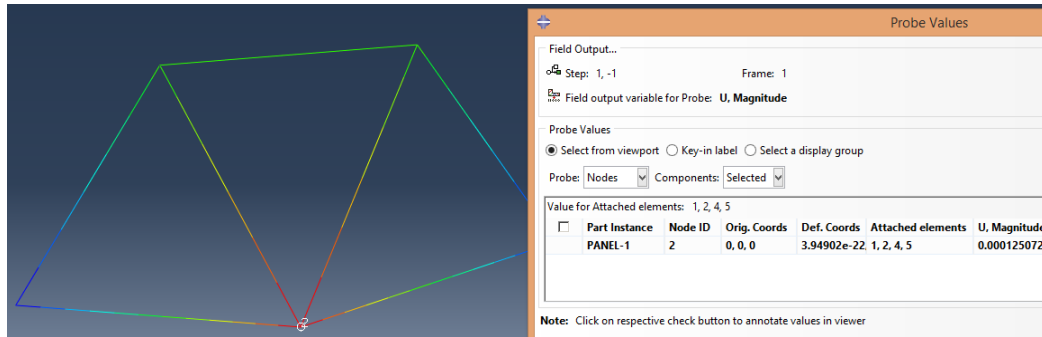


Figure 4.8. Abaqus results for the frame model (B31 elements)

Figure 4.9 illustrates the result of the nodal displacement using the direct stiffness method. B31 beam elements are used such that each node has six degrees of freedoms in the three-dimensional space. The direct stiffness matrix considers the frame problem as two-dimensional and thus having three degrees of freedoms in each node. The global stiffness matrix of the system is also shown in figure 4.9.

1.0e+09 *									Columns 10 through 15						disp =	
Columns 1 through 9																
0.8895	0.0000	0.0002	-0.5929	0	0	-0.1483	0.2567	0.0001	-0.1483	-0.2567	0.0001	0	0	0	0	0
0.0000	0.8895	0.0001	0	-0.0001	0.0001	0.2567	-0.4447	0.0001	-0.2567	-0.4447	-0.0001	0	0	0	0	1.0e-03 *
0.0002	0.0001	0.0005	0	0.8895	0.0000	0.0002	-0.1483	-0.2567	0.0001	-0.0001	0.0001	0.0001	0	0	0	0
-0.5929	0	0	0.8895	0.0000	0.0002	-0.1483	-0.2567	0.0001	0	0	0	-0.1483	0.2567	0.0001	0.0016	0.0216
0	-0.0001	-0.0001	0.0000	0.8895	-0.0001	-0.2567	-0.4447	-0.0001	0	0	0	0.2567	-0.4447	0.0001	-0.0625	-0.0625
0	0.0001	0.0001	0.0002	-0.0001	0.0005	-0.0001	0.0001	0.0001	0	0	0	-0.0001	-0.0001	0.0001	0.0016	-0.0216
-0.1483	0.2567	-0.0001	-0.1483	-0.2567	-0.0001	1.4825	0.0000	-0.0002	-0.5929	0	0	-0.5929	0	0	0	-0.0246
0.2567	-0.4447	-0.0001	-0.2567	-0.4447	0.0001	0.0000	0.8896	0.0000	0	-0.0001	-0.0001	0	-0.0001	0.0001	0.0016	-0.0216
0.0001	0.0001	0.0001	0.0001	-0.0001	0.0001	-0.0002	0.0000	0.0007	0	0.0001	0.0001	0	-0.0001	0.0001	0.0016	-0.0216
-0.1483	-0.2567	-0.0001	0	0	0	0	-0.5929	0	0	0.7412	0.2567	-0.0001	0	0	0	-0.0625
-0.2567	-0.4447	-0.0001	0	0	0	0	0	-0.0001	0.0001	0.2567	0.4448	0.0002	0	0	0	0.0246
0.0001	-0.0001	0.0001	0	0	0	0	-0.0001	0.0001	-0.0001	0.0002	0.0003	0	0	0	0	0.0000
0	0	0	-0.1483	0.2567	-0.0001	-0.5929	0	0	0	0	0	0.7412	-0.2567	-0.0001	-0.1250	-0.1250
0	0	0	0.2567	-0.4447	-0.0001	0	-0.0001	-0.0001	0	0	0	-0.2567	0.4448	-0.0002	0.0000	0.0000
0	0	0	0.0001	0.0001	0.0001	0	0.0001	0.0001	0	0	0	-0.0001	-0.0002	0.0003	0.0000	0.0000

Figure 4.9. Direct stiffness method for the planar frame model

Table 4.4 summarizes the results using the three methods, Karamba, Abaqus, and the direct stiffness method, which converge to the same results.

Table 4.4. Comparing analytical and numerical displacement results (B31 element)

Methods/nodal-displacement	Z-displ. at node 1 (m)	Z-displ. at node 2 (m)	Z-displ. at Nnde 3 (m)	Z-displ. at node 4 (m)	Z-displ at node 5 (m)
<b>Karamba</b>	$-0.62 \cdot 10^{-4}$	$-0.62 \cdot 10^{-4}$	$-0.125 \cdot 10^{-3}$	0	0

<b>Abaqus</b>	-0.62*10 <sup>-4</sup>	-0.62*10 <sup>-4</sup>	-0.125*10 <sup>-3</sup>	0	0
<b>Direct Stiffness Method</b>	-0.62*10 <sup>-4</sup>	-0.62*10 <sup>-4</sup>	-0.125*10 <sup>-3</sup>	0	0

Figure 4.10 illustrates the result of the nodal displacement using the direct stiffness method. T3D2 truss elements are used such that each node has three degrees of freedom in three-dimensional space. The direct stiffness matrix considers the truss problem as two-dimensional and thus having two degrees of freedom in each node. The global stiffness matrix of the system is also shown in the figure 4.10.

```

K =
1.0e+08 *
1.0325  0.0000  -0.6883  0  -0.1721  0.2980  -0.1721  -0.2980  0  0
0.0000  1.0325  0  0  0.2980  -0.5162  -0.2980  -0.5162  0  0
-0.6883  0  1.0325  0.0000  -0.1721  -0.2980  0  0  -0.1721  0.2980
0  0  0.0000  1.0325  -0.2980  -0.5162  0  0  0.2980  -0.5162
-0.1721  0.2980  -0.1721  -0.2980  1.7208  0.0000  -0.6883  0  -0.6883  0
0.2980  -0.5162  -0.2980  -0.5162  0.0000  1.0325  0  0  0  0
-0.1721  -0.2980  0  0  -0.6883  0  0.8604  0.2980  0  0
-0.2980  -0.5162  0  0  0  0  0.2980  0.5162  0  0
0  0  -0.1721  0.2980  -0.6883  0  0  0  0.8604  -0.2980
0  0  0.2980  -0.5162  0  0  0  0  -0.2980  0.5162

disp =
1.0e-03 *
0.018655641999179
-0.053854199650656
-0.018655641999179
-0.053854199650656
0.000000000000000
-0.107708399301313

```

Figure 4.10. Direct stiffness method for the planar truss model

Table 4.5 shows that the two methods, Abaqus and the direct stiffness method, converge to the same results.

Table 4.5. Comparing analytical and numerical displacement results (T3D2 elements)

Methods/nodal-displacement	Z-displ. at node 1 (m)	Z-displ. at node 2 (m)	Z-displ. at node 3 (m)	Z-displ. at node 4 (m)	Z-displ. at node 5 (m)
<b>Abaqus</b>	$-0.539 \cdot 10^{-4}$	$-0.539 \cdot 10^{-4}$	$-0.108 \cdot 10^{-3}$	0	0
<b>Direct Stiffness Method</b>	$-0.539 \cdot 10^{-4}$	$-0.539 \cdot 10^{-4}$	$-0.108 \cdot 10^{-3}$	0	0

#### 4.1.6 iMAGv1 Implementation

Three structural design problems are used for testing iMAGv1 and the demonstration module of the qualitative interview. The description of the models and the structural optimization results are discussed in this chapter.

Each problem has three design variables,  $s_1$ ,  $s_2$ , and  $s_3$ . The first example problem is a single curvature tessellated frame structure consisting of quadrilaterals and irregular hexagons. The second problem is a double curvature tessellated frame structure containing quadrilaterals and irregular hexagons. The third problem is a double curvature space truss structure. Grasshopper is used to generate the model parametrically, and to realize the integration using the premade components.

The single curvature tessellated frame structure (problem 1) has three design variables that control the dimensional scaling of each irregular hexagonal module ( $s_1$ ), the translation of the peak in the transverse direction ( $s_2$ ), and the height of the peak of the curvature ( $s_3$ ).

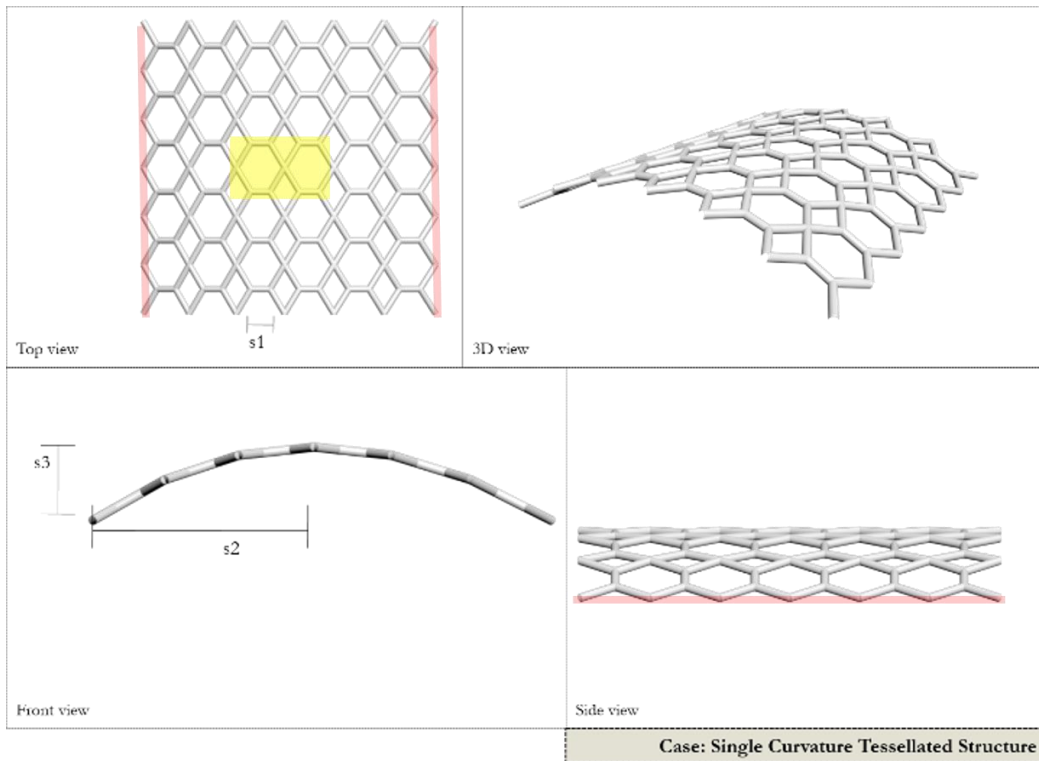
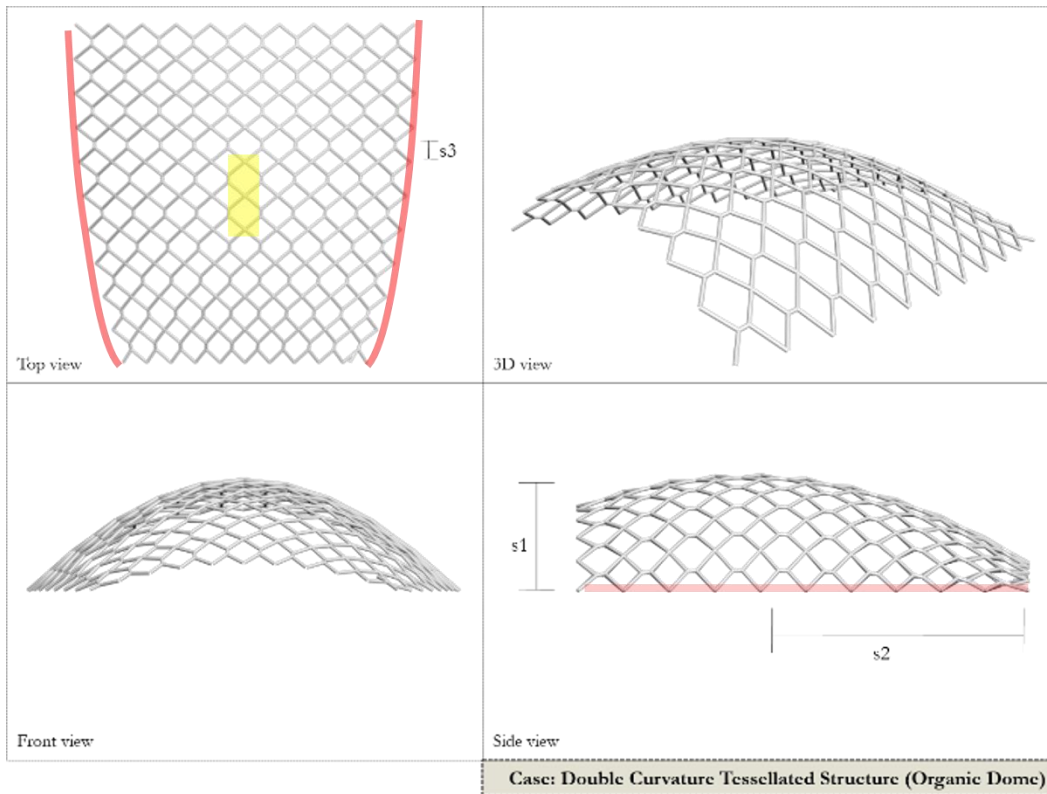


Figure 4.11. Problem 1: Single curvature tessellated frame structure

The organic dome tessellated frame structure (problem 2) has three design variables, including the scale of the modules ( $s1$ ), the shift of the transverse curvature of the dome in the longitudinal direction ( $s2$ ), and the height of the dome represented in scaling quantity ( $s3$ ).



**LEGEND:**

- █ Boundary conditions
- █ Loading conditions

Figure 4.12. Problem 2: Double curvature tessellated frame structure

The double curvature space truss structure (problem 3) has three design variables. Those design variables include the outer radius of the pipe cross section ( $s_1$ ), the height of the dome ( $s_2$ ), and the depth of the structure ( $s_3$ ).

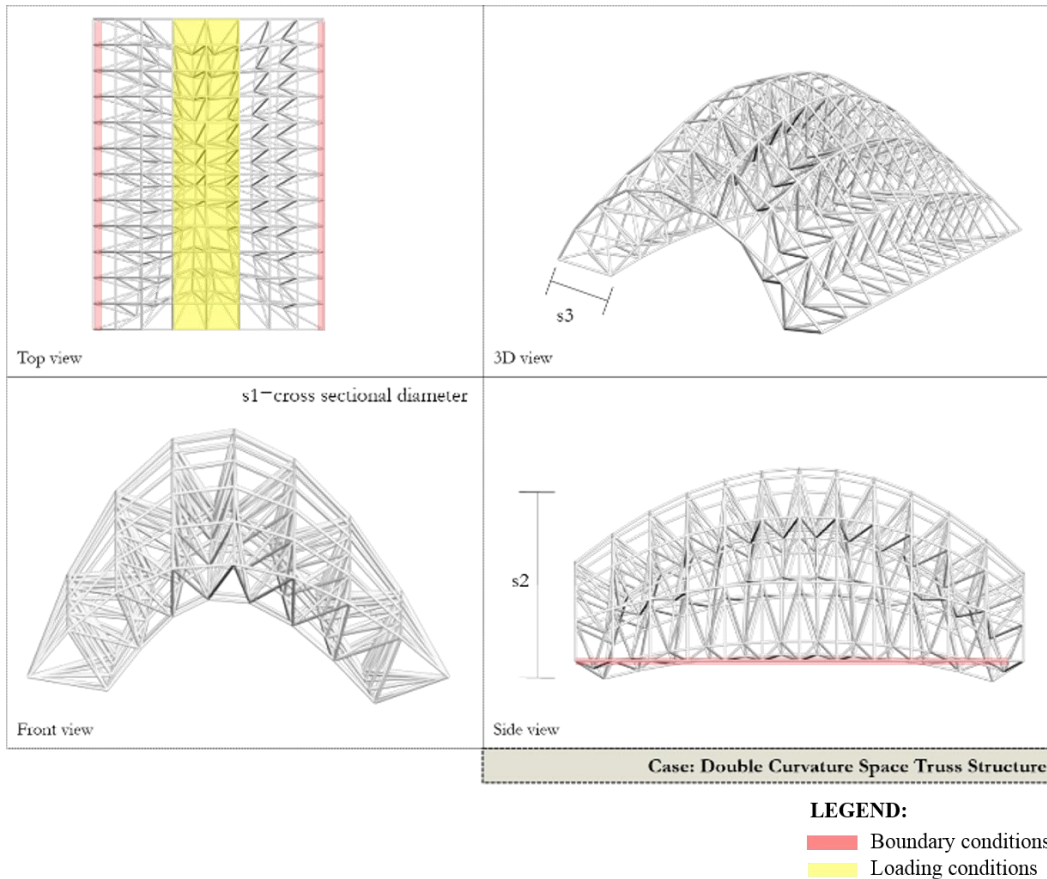


Figure 4.13. Problem 3: Double curvature space truss structure

Each case is treated as a multi-objective optimization problem with the displacement and the weight of the structure as the two conflicting objective functions. They naturally conflict because when the cross-sectional area or the height of the dome is increased, the structure stiffens and thus decreases the displacement at the expense of an increase in weight of the structure. A normalization process is performed such that the displacement and structural weight are of the same order of magnitude. The weighted sum method is used to adjust the respective level of importance of each objective functions.

Equation 4.7. Formula for the total objective function

$$f(s) = \alpha * \textit{normalized\_displacement} + (1 - \alpha) * \textit{normalized\_weight}$$

where  $\alpha$  is a preference coefficient between 0 and 1.

Fixed boundary conditions are applied to all the left and right nodes in the longitudinal direction. For both problems 1 and 2, a load of 100 kN in the negative z-direction was applied at each node at the two hexagonal modules at the center. For problem 3, the selection of the nodes for the load application was controlled by the optional premade functions in Grasshopper (Figure 4.2). The total load of 10,000 kN in the negative z-direction was distributed into all the nodes at the center region around the center longitudinal axis of the structure. The magnitude of the load is adjusted during the optimization loop to give the equivalent resultant of the pressure load at each node within the region if topological optimization is involved. For each of these three problems, the stress constraint was applied such that the maximum normal stress of each member must be less than 250 MPa. The optimized results are graphically shown in Figure 4.14.



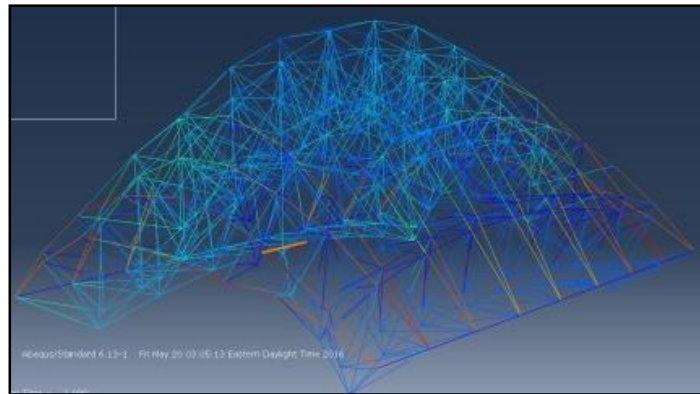
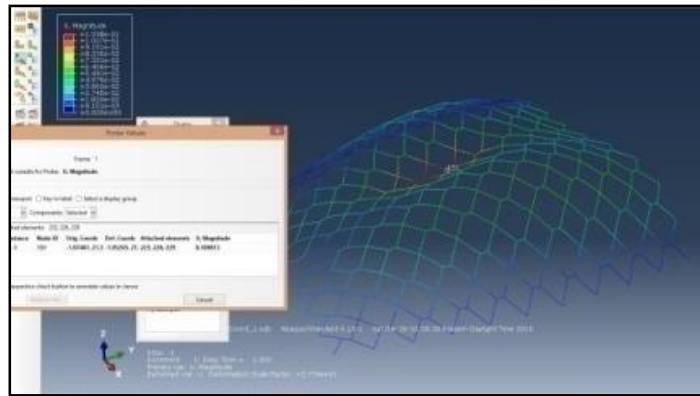
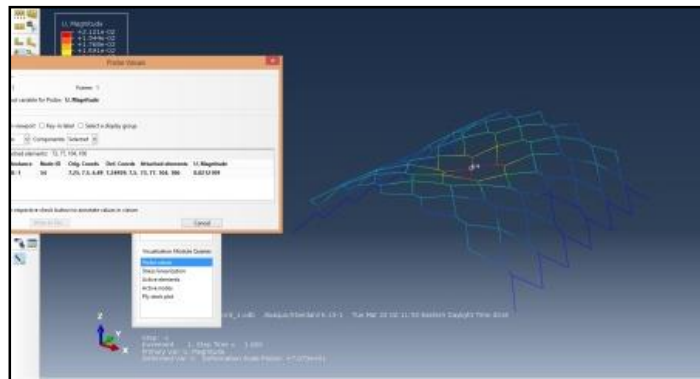


Figure 4.14. Optimized results of each of the three example problems

The initial and optimum values are shown in Table 4.6. The displacement weight is based on the input for the weighted sum optimization using the interior point method.

The weight of the structure is represented by the total length of the truss members since their cross-sectional area is uniform in all three problems.

Table 4.6. Results of the initial and optimized condition of each example problem

Problem Number	Values	$\alpha$	s1	s2	s3	Max Nodal displacement (m)	Max Nodal Stress (MPa)
1	Initial values		0.3	3	3	0.07	304
1	Optimum values	0.9	0.2	7	9	0.021	180
1	Optimum values	0.1	0.4	7	4.8	0.049	200
2	Initial values		0.7	32	0.4	0.11	199
2	Optimum values	0.9	0.5	29.86	0.2	0.11	207
2	Optimum values	0.1	0.5	29.86	0.2	0.11	207
3	Initial values		0.1	25	3	0.02	140
3	Optimum values	0.9	0.35	20	2	0.006	50
3	Optimum values	0.1	0.23	20	2	0.01	80

## 4.2 iMAGv2

iMAGv2 is the same as iMAGv1 except that it uses a subsequent discretization of the cloud of points and filtering process in the second phase instead of the interior point method to identify the optimum. Figure 4.15 shows the workflow of iMAGv2. The second phase discretizes the design space between the data points, interpolates the objective and constraint functions using a cosine interpolation (for cosine interpolation, see Bourke, 1999) and filters the data points to identify the feasible minimum. For iMAGv2, a cosine interpolation is used to create a differentiable function at each iteration of optimization because the interior point can only operate on a smooth function. Note that the exact analysis using FEA is performed only on the cloud of points of the first discretization of

the design space. The function and constraint evaluations of the points of the subsequent discretization are approximated using the cosine interpolation.

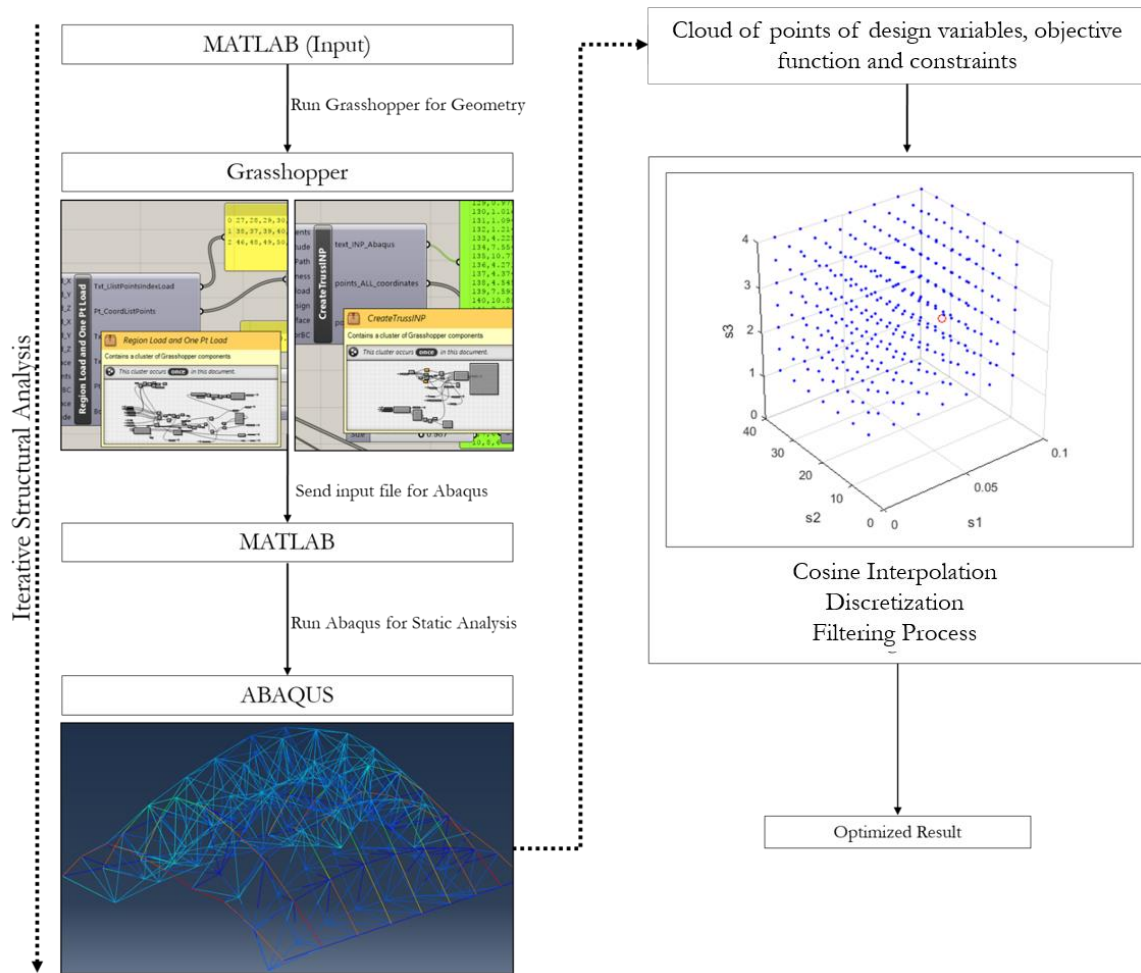


Figure 4.15. Workflow of iMAGv2

### 4.3 iMAGv3

The previous versions, iMAGv1 and iMAGv2, use exhaustive search to integrate Grasshopper, Abaqus, and Matlab to conduct the structural optimization routine. The exhaustive search approach in the previous versions of iMAGv1 and iMAGv2 are

computationally expensive due to the full factorial analysis that has to be conducted before performing the second phase of the optimization. This approach may produce a highly inaccurate data of displacements and stresses, depending on the number of iterations configured by the users. This section discusses a method, iMAGv3, that directly integrates Grasshopper, Abaqus, and Matlab.

iMAGv3 allows for a flexible number of design variables for conducting the optimization. The inputs of the design variables are set up in Matlab and pass into Grasshopper. Grasshopper then generates the geometry and creates the input file for Abaqus using the premade components as shown in figure 4.16. Abaqus receives the input file and generates the maximum displacement and stress values that are used as parameters for Matlab to define the constraint and objective functions. Matlab uses the interior point algorithm to carry out the optimization and provide the new values for the design variables. The cycle repeats until the optimum is reached.

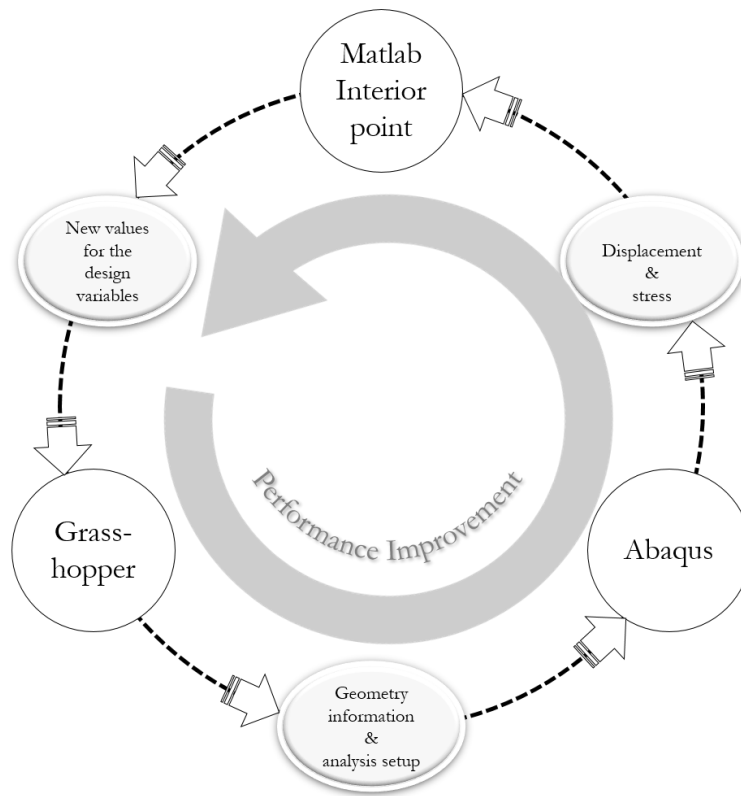


Figure 4.16. Workflow of iMAGv3

## 4.4 iGFM

### 4.4.1 Workflow of iGFM

iGFM is the integration between Grasshopper, customized FEA code, and Matlab interior point algorithm for conducting the structural optimization. Essentially, the workflow of iGFM replaces Abaqus from iMAGv3 with the developed customized FEA code. Thus, at each iteration, Grasshopper generates the input for the customized FEA code in Matlab including the geometrical properties (i.e., nodal coordinates, element connectivity, and cross-sectional properties) and analysis information (i.e., type of

analysis, boundary and loading conditions). The output from the customized FEA includes the nodal displacements and von Mises stresses. The values of the nodal displacement and stresses are then used for constructing the optimization model in another Matlab file for conducting the optimization routine.

One of the main advantages of this workflow compared to iMAGv3 is that iGFM is computationally much more efficient because the workflow only involves two platforms for conducting the whole design, analysis, and optimization process. iMAGv3 needs to call Abaqus at each iteration of the optimization loop, which causes the method to be computationally expensive compared to iGFM.

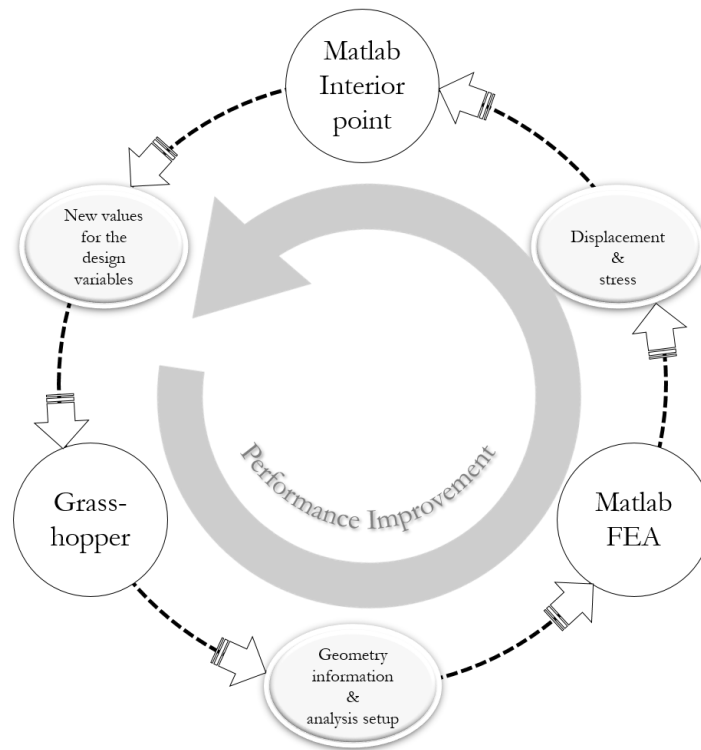


Figure 4.17. Workflow of iGFM

#### 4.4.2 Validation of Customized FEA

iGFM uses a customized FEA code written in Matlab for conducting the space truss and space frame analysis using the direct stiffness method as discussed in section 2.2.3. For each connectivity defined in Grasshopper, the customized FEA code identifies whether it is a truss or beam. The corresponding elemental local stiffness and transformation matrices are imposed to the identified connectivity, and the elemental global stiffness is then assembled to form the global stiffness matrix. The elemental reaction forces at each end node are found by using the global displacement vector of each element multiplied by the elemental global stiffness matrix. The corresponding elemental normal and shear stresses can then be found and utilized to calculate the elemental von Mises stress of each element. Figure 4.18 shows the validation process of the developed customized FEA. The maximum displacement and stress results of the customized FEA code was compared with the results from Abaqus and Karamba as shown in Figure 4.18.

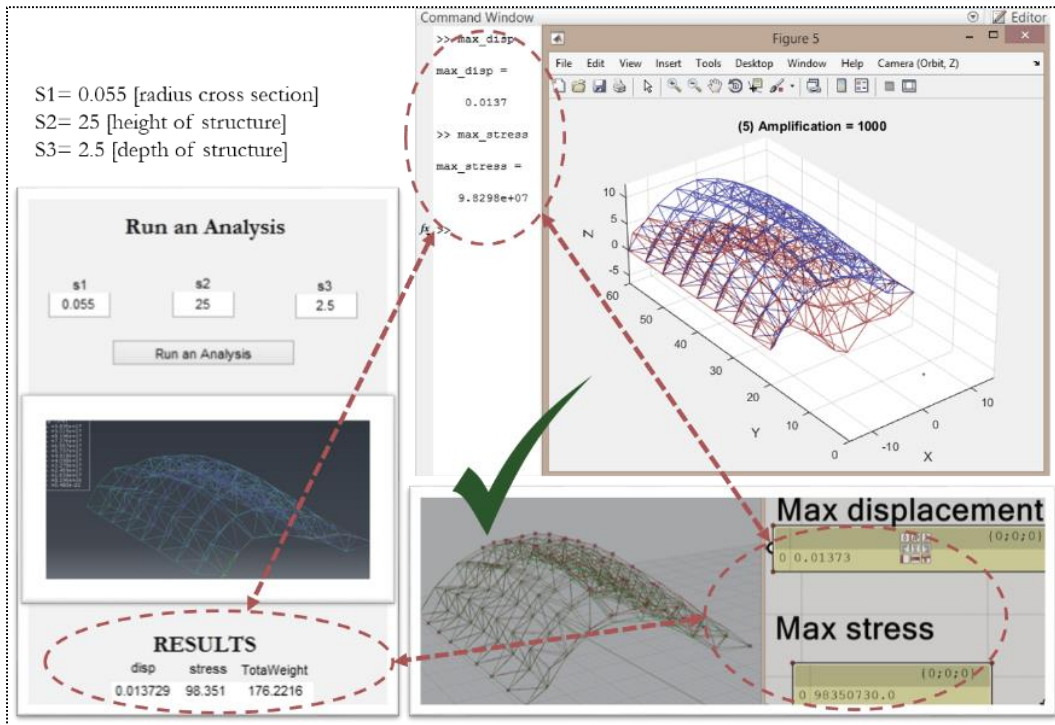


Figure 4.18. Validation of the developed customized FEA (top) by comparing the maximum displacement and stress results with Abaqus (bottom left) and Karamba (bottom right)

The customized FEA code can deal with problems that have various element types and section properties. Simple case studies have been developed in the research to validate the results between the customized FEA and Abaqus when using various section properties in a problem. Figure 4.19 shows a polyhedron space truss structure made of two interconnected trusses of two different section properties. As such, Figure 4.19 shows the two trusses on the left and the assembled structure on the right. The first truss consists of the outer and inner polyhedrons shown on the left of Figure 4.19. The polyhedrons are made of solid steel cylinders of radius 0.01m. The second truss shown in the middle of Figure 4.19 consists of braces connecting the outer and inner polyhedrons. The braces are solid steel cylinders of radius 0.005m.



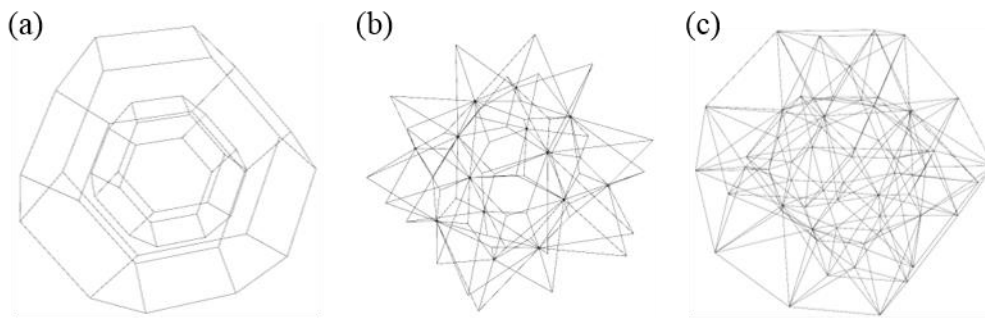


Figure 4.19. Space truss structure (c) made of two trusses of different section properties: (a) inner and outer polyhedrons and (b) connecting braces

Fixed boundary conditions are applied at the four bottom nodes of the structure and concentrated loads are applied at the four top nodes. The results of customized FEA are compared to those obtained with Abaqus. Figure 4.20 shows the results of the maximum displacement and maximum stress when using the two methods. The results are found to be the same, and thus the developed customized FEA code is validated in terms of using various section properties for the space truss problems.

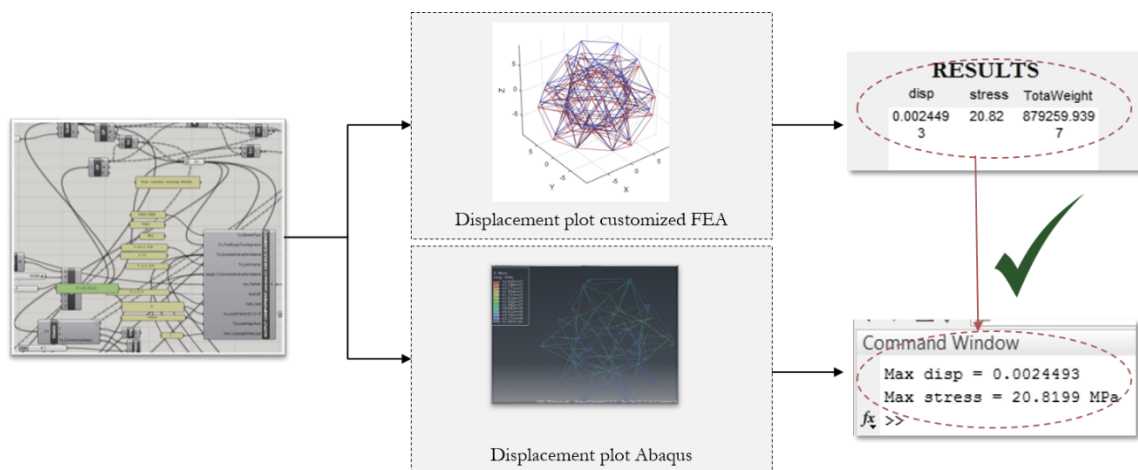


Figure 4.20. Comparing results between customized FEA and Abaqus for a problem with various section properties

## 4.5 iGAG

iGAG includes a Genetic Algorithm (GA) for the optimization process and was developed in response to two demands. The first demand is to be able to deal with discrete design variables (i.e., integer programming), which is necessary for most topology optimization problems. The second demand is based on the need for design flexibility as identified during the first cycle of this research using the grounded theory. The demand is relevant to the participants' opinions on the education module, i.e., "there are many design factors in architecture other than the structural performance." Thus, they desire the ability for carrying out further design tweaking of the optimized solution (see chapter 6 for details on the interview responses). In this case, a set of solutions is preferable as opposed to a single optimum solution.

This section discusses the components and workflow of iGAG developed in this research. The tool is the integration of two pieces of software: (1) Grasshopper for structural parametric modeling using the four premade components discussed in section 4.1, and (2) Matlab for both structural analysis and optimization.

### 4.5.1 Workflow of iGAG

Figure 4.21 shows the integration between Grasshopper and Matlab. In this method, the design variables are defined in Grasshopper, and the values are parameterized by Matlab. The customized FEA code programmed in Matlab using the direct stiffness method was used to carry out the static analysis of space truss or beam structures. A customized Genetic Algorithm (GA) was scripted in Matlab to conduct the optimization. Each individual generated by the algorithm at each iteration is an input

vector that parameterizes the design variables in Grasshopper. Grasshopper generates the formatted information consisting of nodal coordinates, element connectivity, material property, loading and boundary condition describing the geometry and structural analysis setup. This information is used by the customized FEA code to evaluate the fitness and constraint values for each individual in the population at each iteration of the optimization process.

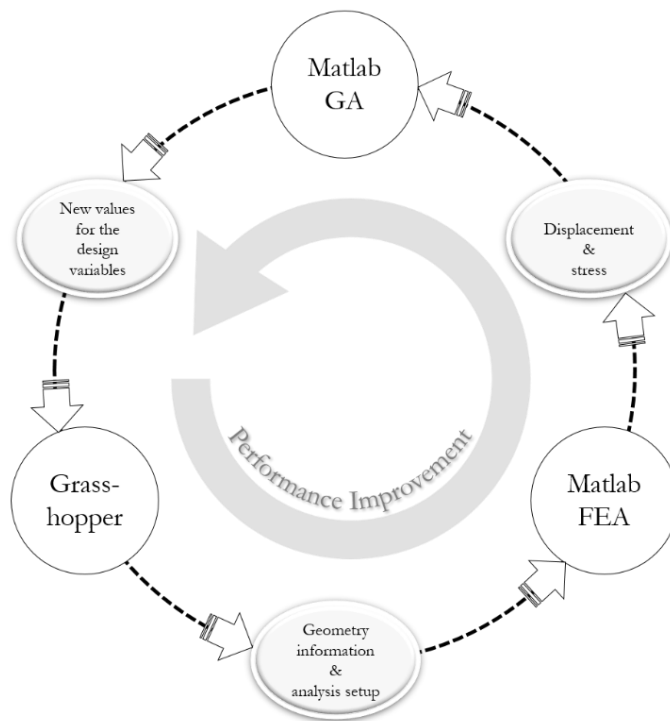


Figure 4.21. Workflow of iGAG

#### 4.5.2 Validation of Customized GA

The customized GA uses a binary encoding process, and performs the selection procedure by using the roulette wheel selection based on the individuals' fitness value (see Mitchell, 1998). Two crossover methods are implemented into the customized GA including the single-point and uniform crossover. The genetic operation for the mutation process uses the bit inversion technique with an adjustable rate of mutation. The user also

has the option to activate the elitism and to configure the rate of the elitism to help ensure convergence. Elitism strategy is vastly utilized to ensure the improvement of the convergence in the individuals' fitness in each subsequent generation (Liang and Leung 2010). The process iteratively continues until reaching the termination criterion.

Three different test functions are used to validate precision and robustness of the developed customized GA: (1) Booth function, (2) Levi function, and (3) Easom function. Figure 4.22 shows the surface of Booth function.

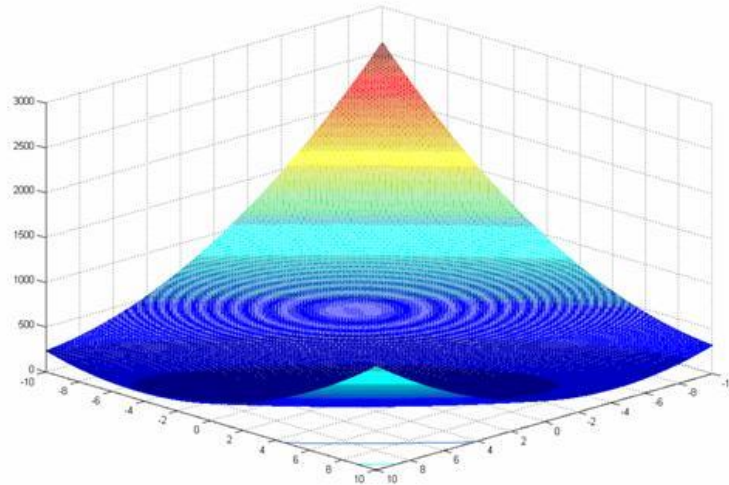


Figure 4.22. Booth function (Source: Hedar, n.d.)

The corresponding equation of the Booth function is as follows (Jamil & Yang, 2013):

Equation 4.8. Booth function

$$f(\mathbf{x}) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$$

The optimum is at  $f = 0$  and  $x^* = [1, 3]$ . Figure 4.23 shows that using a population size of 20 individuals and 20 generations, with the lower bound  $[0 \ 0]$ , and upper bound  $[10 \ 31.01]$ , the customized GA converges to the minimum  $f = 0$ .

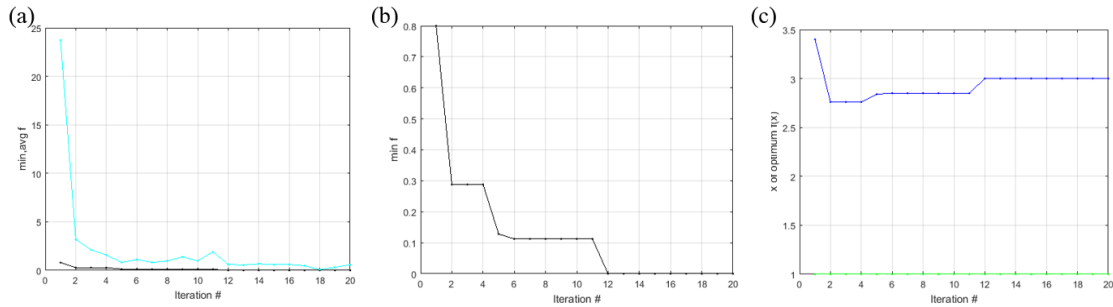


Figure 4.23. Customized GA validation for the Booth function: (a) minimum value (black) and average value (light blue) of the fitness function vs iteration number, (b) minimum value of the fitness function vs iteration number, and (c) design variables of best individual vs iteration number

Figure 4.24 shows the surface of Levi function.

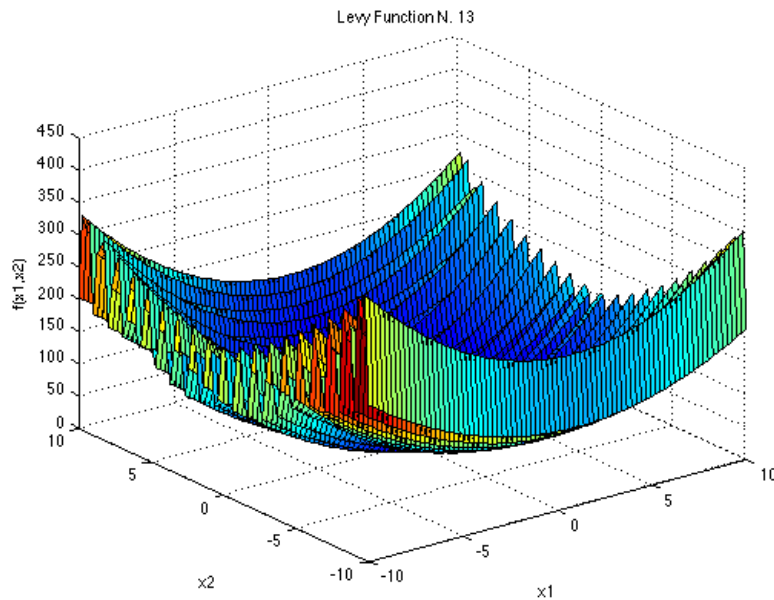


Figure 4.24. Levi function (Source: Surjajonic and Bingham, 2015)

The corresponding equation of the Levi function is as follows (Malherbe, Contal and Vayatis, 2016):

Equation 4.9. Levi function N.13

$$f(\mathbf{x}) = \sin^2(3\pi x_1) + (x_1 - 1)^2 [1 + \sin^2(3\pi x_2)] + (x_2 - 1)^2 [1 + \sin^2(2\pi x_2)]$$

The optimum is at  $f = 0$  and  $\mathbf{x}^* = [1, 1]$ . Figure 4.25 shows that using 20 individuals and 20 generations, with the lower bound [0 0], and upper bound [10.1 10.1], the customized GA converges to the minimum  $f = 0$ .

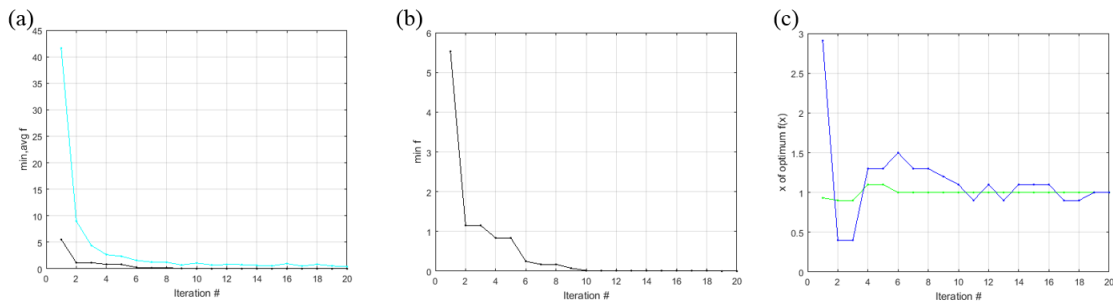


Figure 4.25. Customized GA validation for Levi function: (a) minimum value (black) and average value (light blue) of the fitness function vs iteration number, (b) minimum value of the fitness function vs iteration number, and (c) design variables of best individual vs iteration number

Figure 4.26. shows the surface of Easom function.

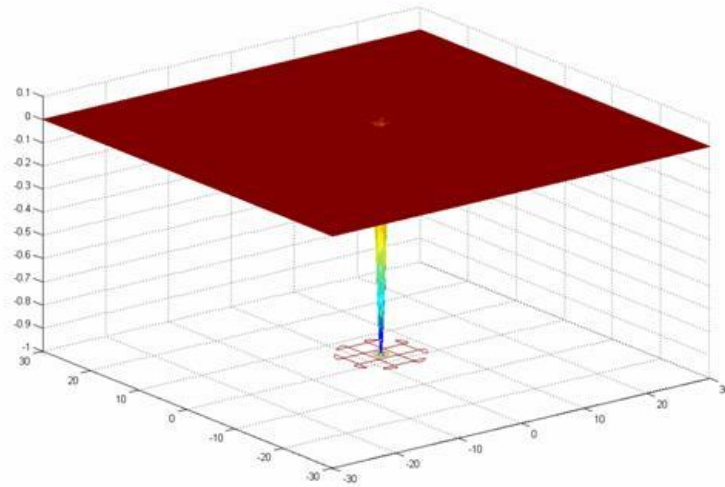


Figure 4.26. Easom function (Source: Hedar, n.d.)

The corresponding equation of the Easom function is as follows (Molga and Smutnicki, 2005):

Equation 4.10. Easom function

$$f(\mathbf{x}) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$$

The optimum is at  $f = -1$  and  $\mathbf{x}^* = [\pi, \pi]$ . Figure 4.27 shows that using 80 individuals and 30 generations, with the lower bound  $[0 \ 0]$ , and upper bound  $[50.11 \ 50.11]$ , the customized GA converges to the minimum  $f = -0.99489$ .

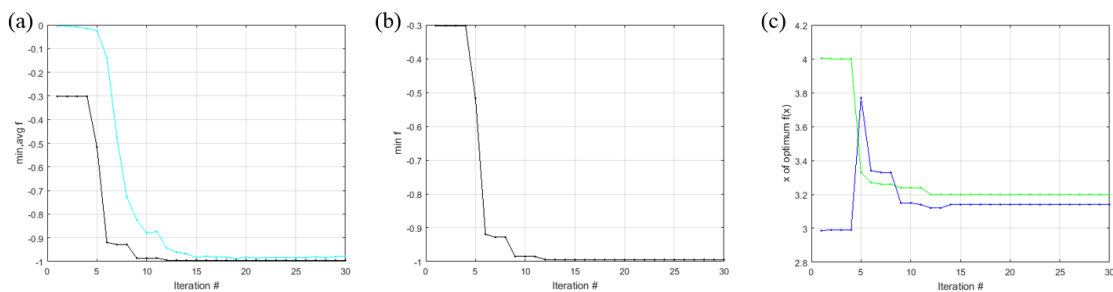


Figure 4.27. Customized GA validation for Easom function: (a) minimum value (black) and average value (light blue) of the fitness function vs iteration number, (b) minimum value of the fitness function vs iteration number, and (c) design variables of best individual vs iteration number

#### **4.6 iGKG**

iGKG is the existing available architectural structural optimization method that integrates Grasshopper, Karamba, and Galapagos. Both Karamba and Galapagos (Rutten, 2017) are plugins for Grasshopper. Karamba can carry out a finite element analysis of trusses and beams. Galapagos uses Genetic Algorithm to conduct the optimization. During the optimization, the designer can stop the process manually if the optimum had been reached. The designer can have the options to configure the number of individuals in the population and define the maximum number of generation as the termination condition of the optimization process. The term “Genome” in Galapagos is used to define the design variables. Constraints can be imposed into the optimization model by adding the constraints to the objective function. Alternatively, constraints can be imposed by creating the constraint conditions using scripting component, or mathematical operator



components in Grasshopper. Figure 4.28 shows the workflow iGKG.

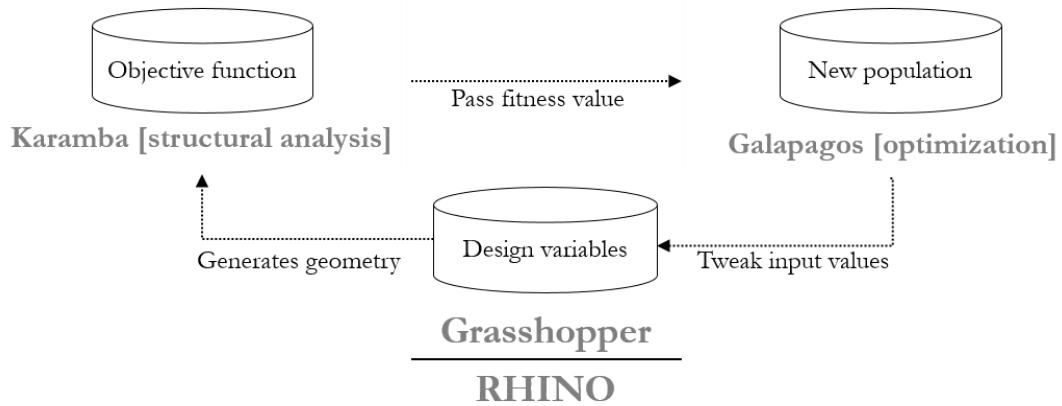


Figure 4.28. Workflow of iGKG

Figure 4.29 shows the Karamba plot of the structure with the loading conditions applied to the structure. The Galapagos interface is also shown in the figure 4.29. All the genomes representing the values of the design variables are shown as scattered dots. All the values of the objective function of each individual are shown in the interface and can be reinstated so that the user can visualize the corresponding design permutation directly in the Rhino interface. In this particular space truss problem, using a population size of 20, the optimum was found at the fourth generation in which eighty function calculations are conducted prior to the emergence of the fittest individual.

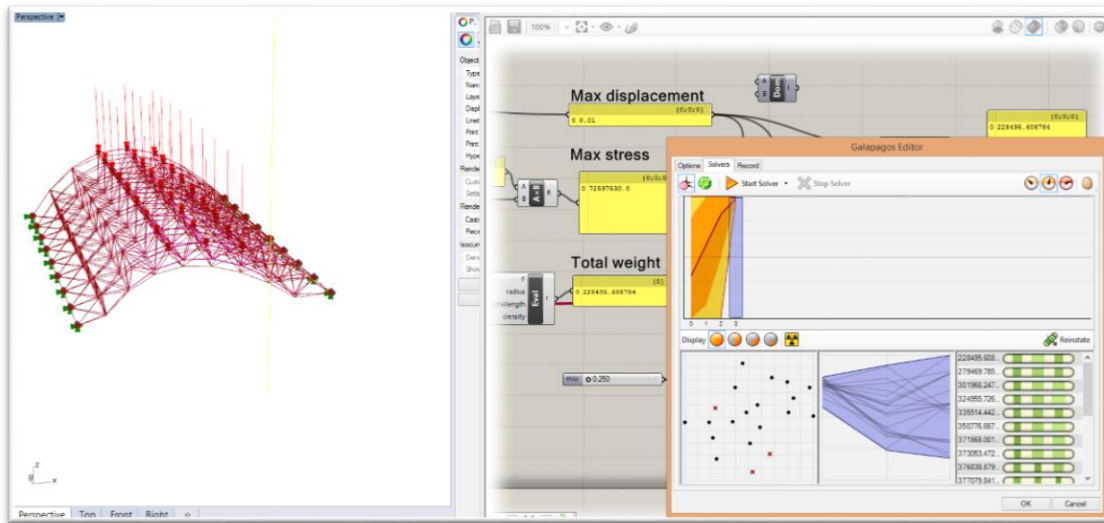


Figure 4.29. Grasshopper interface, Karamba plot, and Galapagos interface

#### 4.7 Comparative Study for the Research Concern Two

The results of iMAG, iMAGv2, iMAGv3, iGFM, and iGKG are compared. This section of this chapter is to investigate research concern 2 and 3. The criteria for the comparative study include computational time, number of function evaluations, and convergence of the results. The double curvature space truss structure (Figure 4.13) is used for comparing the methods. It consists of minimizing the weight of the structure with displacement and stress constraints. This problem is equivalent to minimizing the manufacturing cost of the structure while satisfying prescriptive code requirements on displacement and stress.

iGKG does not require an initial point to be defined for conducting the optimization. Instead, the population size must be defined before beginning to run the optimization task. In this example, 20 individuals were used. IMAGv2 does not use a starting point either since it performs a pre-determined set of analyses. The interior point

optimization algorithm of iMAGv1, iMAGv3, and iGFM require a starting point. In this case, the same starting point is used for the three methods, i.e., [0.095; 15; 3.5].

In this particular optimization problem, the displacement constraint is active at the optimum. In other words, the maximum nodal displacement of the optimum design is the maximum allowable displacement.

The number of function calculations is shown in Table 4.7, which indicates how many times the objective function is called to calculate the weight of the structure. In this case, both the number of calls of the objective and constraint functions are the same for all presented methods. This indicates that the number of finite element analyses is equal to the number of structural weight calculations. Both iMAGv1 and iMAGv2 require the most objective and constraint function evaluations. In this case, the three design variables are each discretized with 8 level, which provides a full factorial analysis of  $7^3 = 343$  designs for the fully crossed design. Table 4.7 shows efficiency and the convergence of each method.

Table 4.7. Comparison of structural optimization methods

Method	#f(x)	Computational cost	S1(m)	S2(m)	S3(m)	max. nodal disp (m)	max stress (MPa)	Total Weight (ton)
iMAGv1	343	282min + 6min	0.064	15.00	2.50	0.01	72.839	231.315
iMAGv2	343	282min + 240min	0.064	14.97	2.56	0.01	73.174	232.174
iMAGv3	104	65min	0.064	15.43	2.62	0.01	73.907	228.743
iGFM	62	5min	0.063	20.94	2.75	0.01	68.448	230.822
iGKG	80	2min	0.063	17.33	2.82	0.01	72.597	228.495

The most time-consuming methods are methods iMAGv1 and iMAGv2 due to the full-factorial iterative analysis of the first phase, which takes about 282 min. The computational time of the two methods varies in the second phase, which is the

interpolation and filtering process for iMAGv2. In this case,  $700^3$  points are used for the secondary discretization of the design space, and thus, an additional 240 min of computational time. For iMAGv1, the second phases use the interior point optimization algorithm to find the optimum in about 6 minutes, which is much faster than the filtering process of method iMAGv2. The computational time of both methods relies heavily on the number of discretization points before the filtering process, and the accuracy of the results can be improved by increasing the number of points, but at a higher computational cost. In terms of the convergence, iGKG is shown to be the most powerful method among the five methods since it manages to find the smallest structural weight in the faster time and least effort. iGKG is also considered the most convenient compared to the developed methods in terms of usability since the whole architectural structural optimization tasks can be conducted conveniently in an interactive visual programming language, i.e., using only Grasshopper.

Thus, regarding research concern two, the existing architectural, structural optimization tool such as iGKG is accessible and appropriate for architects. Unfortunately, the utilization of iGKG is not widespread among architects or even architectural academia. This issue is discussed in chapter 7.

#### **4.8 Case Study - Design of a Twisted Skyscraper**

A parametric model of a twisted skyscraper was made to demonstrate the capability of iGAG to deal with a complex mixed nonlinear integer optimization problem. The parametrically developed skyscraper model employs exterior diagrid system for the braced tubular structural system. The structural system uses the

combination of frames and trusses elements at the outside perimeter as the primary structure. Beam elements are located at each corner of the tower as columns extending from the base of the tower to the top. Spandrel beams at each floor tie the columns. The diagonal braces are utilized to increase the lateral stiffness, allowing more spacing for the vertically spanned truss members. The design employs the twist of the structure as the design variable that needs to be optimized.

#### 4.8.1 Bracing Types and Variability

Various bracing types are considered. Variability such as bracing types and number of corners in the floor plans are programmed in the Grasshopper and can be used as design variables for the optimization. However, they are not used as design variables in this particular case study to simplify the structural optimization process with respect to the computational time. Bracing type one uses only beam elements. Bracing type two uses the combination of beam and truss elements. As shown in Figure 4.30, the tower includes columns at each corner of the structure, and each spandrel beams are all beam elements. The perimeter braces are the truss elements. Bracing type three, four and five use the same configuration for the beam elements, but the setup for the truss elements are different. Bracing type six and seven use beam elements for the entire structure.

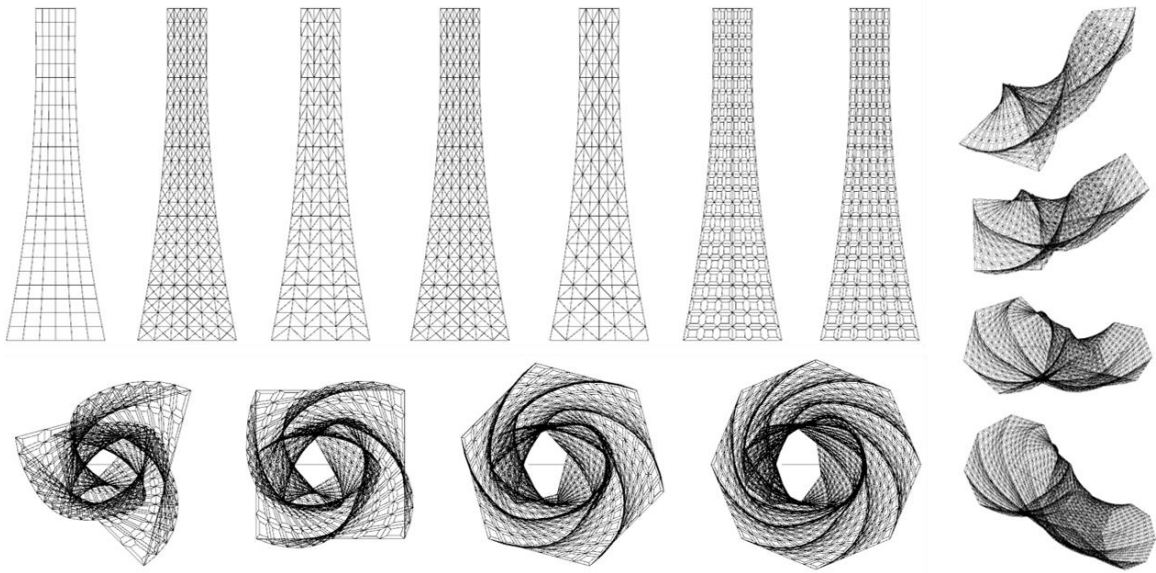


Figure 4.30. Bracing types and variability of the parametric model of the skyscraper

#### 4.8.2 Design Variables

For the optimization, four design variables are used including the radius of the floor located in the mid-section of the skyscraper ( $s_1$ ), the radius of the floor based on the peak of the building ( $s_2$ ), the twist of the tower parameterized from zero to one ( $s_3$ ), and the number of floors ( $s_4$ ). The variable  $s_1$ ,  $s_2$ , and  $s_3$  are continuous, while the variable  $s_4$  is discrete. The radius of the floors between the bottom, middle and peak are taken as the interpolated values of those radii. The thickness of the cross-section of the structural members is a function of the number of floors ( $s_4$ ) such that the thickness reduces from the bottom to the top of the building. The optimization problem is taken as the mixed of non-linear and integer programming problem involving changes in the number of connections and elements. Other variables such as the number of corners of the floor plan, upper and lower bound of the cross-sectional thickness, and the bracing type are

fixed, but could also be used as design variables depending on the design intent. In this case, hexagonal floorplans are used with the bottom floor of 20 x 20 meters.

Table 4.8 shows the values of the design variables at the lower and upper bounds of the design space.

Table 4.8. Lower and upper bound state of the optimization

Boundary	s1 (m)	s2 (m)	s3 (rad)	s4 (#floors)	Disp (m)	stress (MPa)	Total area (m2)
LB	4	3	0	18	0.054	32.7	4131.6
UB	20.1	20.1	5.032	26	1.421	1083.9	28281

Figure 4.31 shows the displacement plot of the lower and upper bound states of the optimization.

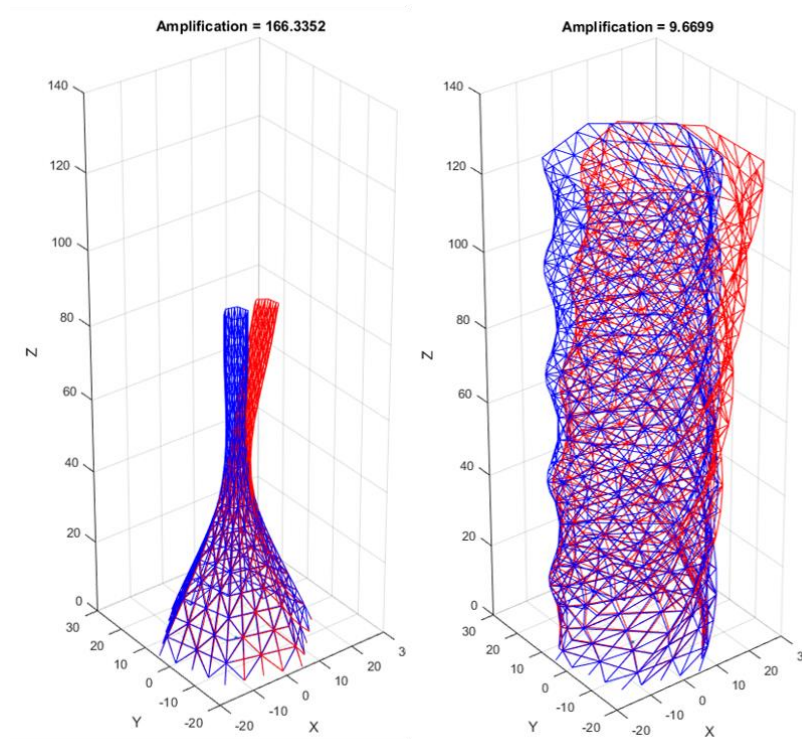


Figure 4.31. Displacement plot of the lower and upper bound state

#### 4.8.3 Parametric Dead and Wind Loads

The structure is subjected to both dead and wind loads. The dead load is a function of  $s_1$ ,  $s_2$ , and  $s_4$ . The dead load is also a function of the size of the cross-section. The reduction of the thickness of the cross-section is to decrease the weight of the building, and thus reduce the stress at the base. Figure 4.32 shows an example of dead load conditions corresponding to the reduction of the thickness of the cross-section of the structural members at a sampled design permutation.



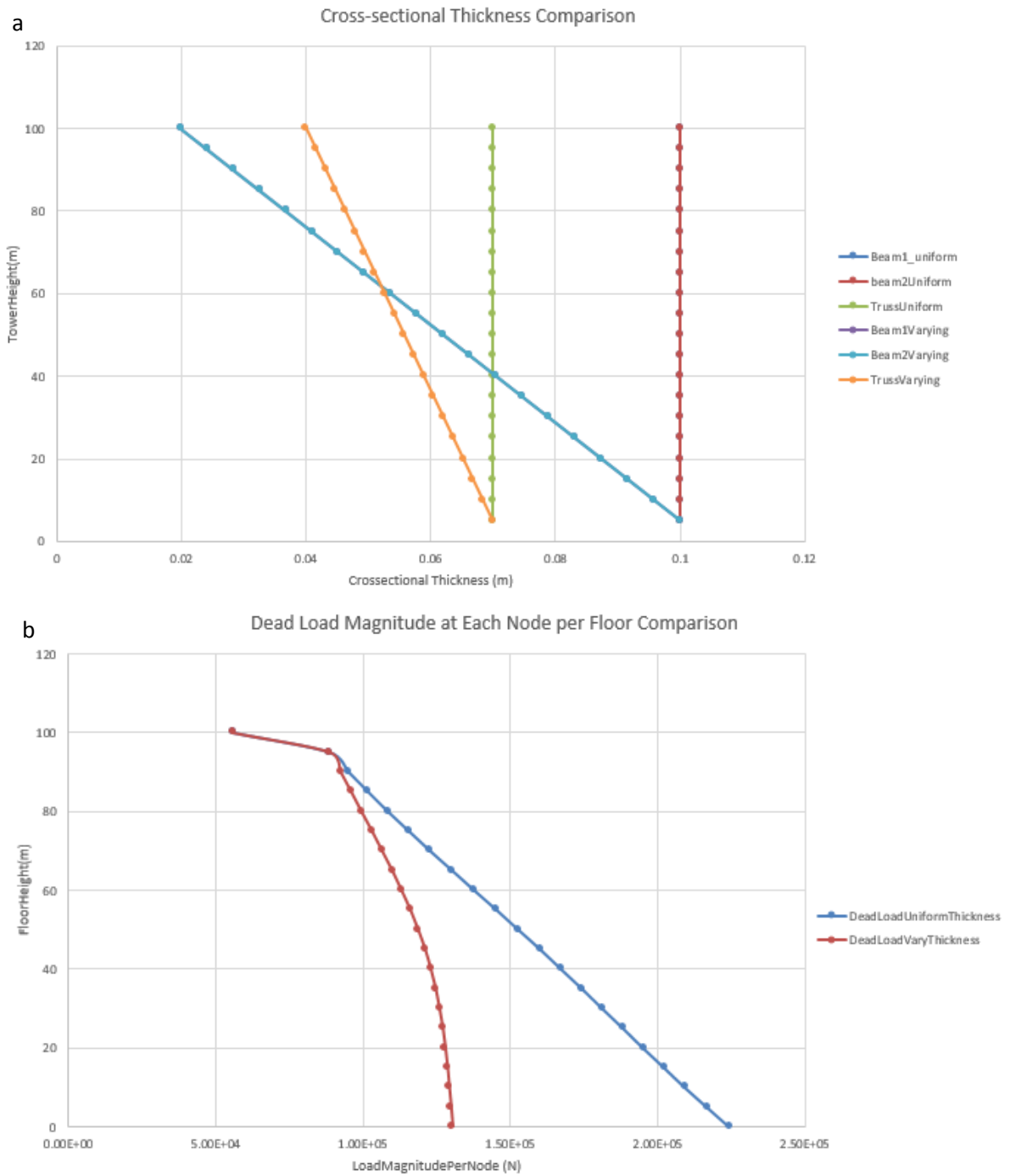


Figure 4.32. Reduction of the thickness of the cross section : (a) thickness of cross-section vs. height of the tower and (b) Load magnitude vs. height of the tower

Two variables are considered to calculate the wind pressure including the wind velocity pressure  $q_z$ , and the leeward and windward pressure coefficient  $c_p$ . The wind velocity pressure is a dependent variable of the building story's height and the surface area of the building of each story. Other parameters for calculating the  $q_z$  such as topographic factor, importance factor, wind directionality, and velocity pressure exposure are taken into account using the available building codes for the design of a tall infrastructure. The mean of the wind speed is set to be 53 meters per second in which, based on Saffir-Simpson scale, the magnitude falls within the range for simulating the typical hurricane wind load (Vickery et al., 2000). The importance factor of 1.15 is used to assume an occupiable building with a substantial hazard to human life in the event of failure. The value of  $c_p$  is a function of the orientation of façade in each story of the building. Figure 4.33 shows the x and y components of the  $c_p$  values at each section of the skyscraper for the sampled design permutation. There is a total of 24 sections

corresponding to four bracing segments at each edge of the hexagonal floor plan.

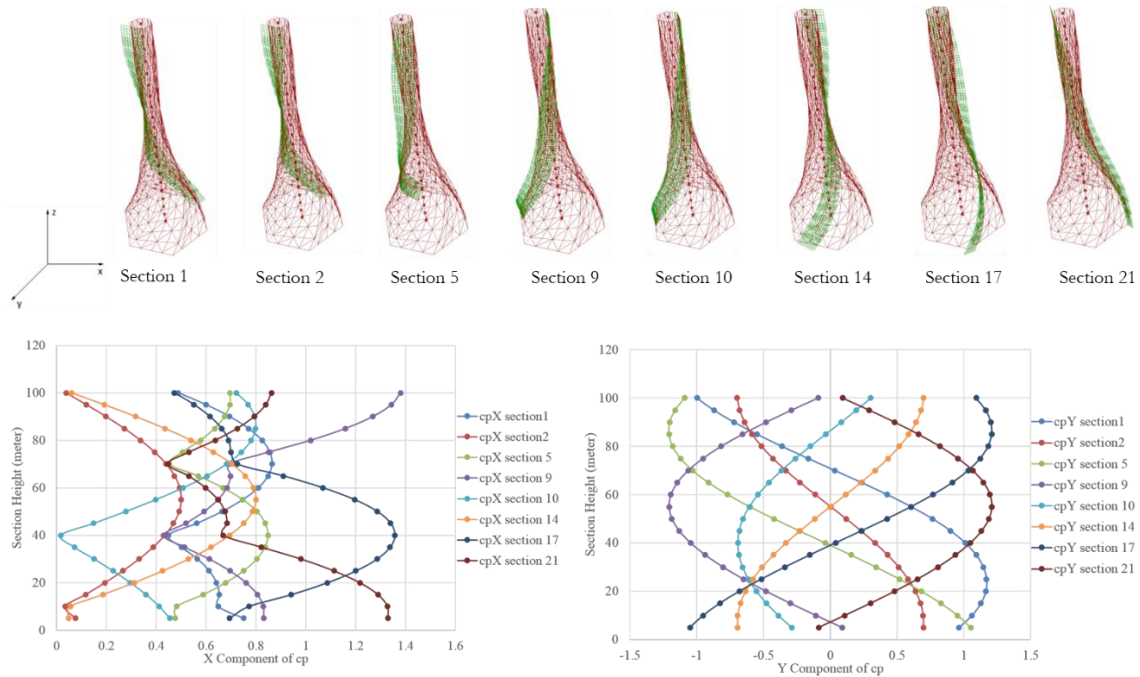


Figure 4.33. Windward and leeward pressure coefficient at sampled sections for the sampled design permutation

#### 4.8.4 Constraint and Objective Functions

The objective function is to maximize the twist angle of the tower ( $s_3$ ). Three constraints are imposed on the optimization model including the maximum allowable elemental von Mises stress, the maximum allowable nodal displacement, and the total occupiable area of the building. The maximum elemental von Mises stress is restricted to be less than 250 MPa for a feasible solution. The maximum nodal displacement in the structure has to be less than 0.2 percent of the building height (see Zhu, 2014). The total occupiable area has to be between 12000m<sup>2</sup> and 14000m<sup>2</sup>. The right-hand side of the displacement constraint changes at each iteration of the optimization depending on the height of the tower, in this case, the value of  $s_4$ . Likewise, the value of the total

occupiable area changes at each iteration depending on the values of  $s_1$  and  $s_2$ . For the design intent, aesthetically the rate of twist should increase to the point where one or more constraints become active. Reducing the height of the tower ( $s_4$ ) can significantly reduce the stress and displacement contributions due to the wind velocity pressure, and reducing both  $s_1$  and  $s_2$  decreases the exposure of the façade to the wind pressure. However, the realistic architectural problem is to have the programmatically related constraint that considers the aspect of occupiable spaces which is, in this case, the total area of the building's stories. Thus, simply reducing  $s_1$ ,  $s_2$  and  $s_4$  would violate the total area constraint.

The computational time of each iteration of the optimization is determined by the execution time of Grasshopper and customized FEA. In this particular case, the execution of the Grasshopper portion and the customized FEA requires an average computational time of 46 seconds and 53 seconds, respectively, at each iteration.

#### 4.8.5 Structural Optimization Results of Skyscraper

Fifty individuals and forty generations are used to perform the structural optimization of the tower by utilizing the customized GA in iGAG. One of the advantages of using the customized GA is that the optimization process can be stopped and continued at any time provided that the population generated at the last stop is the same as the one for continuing the process. Also, unlike deterministic optimization methods, every run of the GA can potentially result in different improved individuals, due to the stochastic nature of the algorithm. Figure 4.34 shows a sample run of the GA for the tower case. The maximum individual fitness value,  $s_3$ , increases drastically at the

second iteration, and improve further at the third iteration. The further slight improvement occurs at the 20<sup>th</sup> iteration. Another variable such as s1 fluctuates more and goes steadily at the optimum where the value lies around half way between the lower and upper bounds. The oscillations of the average fitness values show that the stochastic process involving the crossover and mutation in the GA perform as desired.

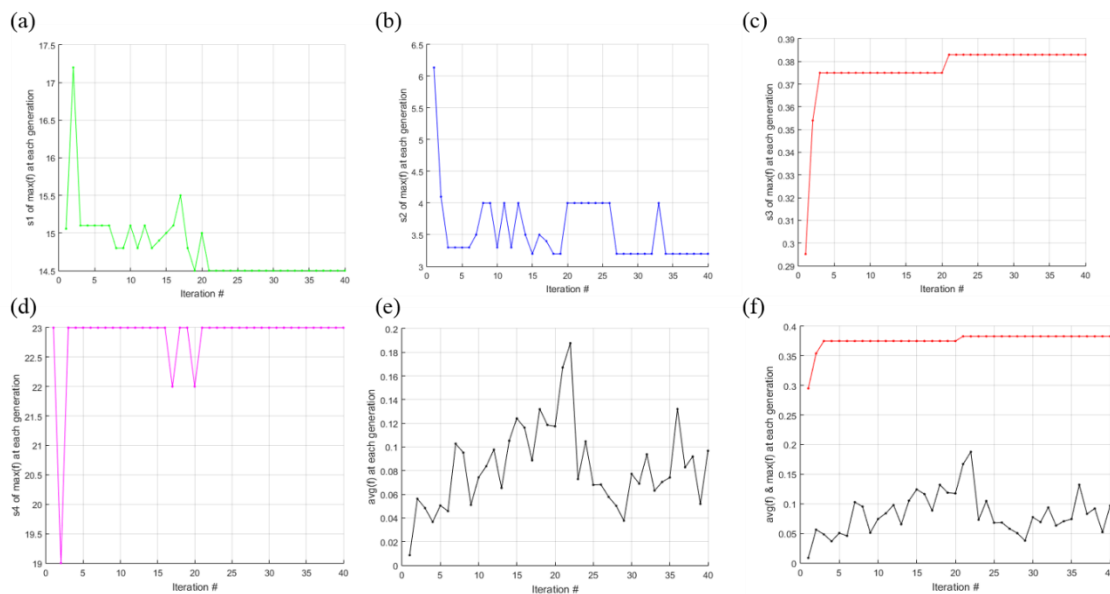


Figure 4.34. iGAG GA plots for tower design: Evolution of s1 (a), s2 (b), s3 (c), s4 (d), Average Fitness (e), and Maximum and Average Fitness (f) as functions of number of iterations

Based on the interview responses, participants prefer a set of improved solutions rather than necessarily a single optimum (see appendix D). Architectural design problems are typically wicked in which an unlimited amount of constraints can be imposed into a single design problem. Measurable constraints such as structural performance and thermal comfort can easily be formulated and quantified. However, aesthetic is subjective in nature. Thus, ideal results of architectural structural optimization method should result

in a narrowed set of solutions that can be used for further design tweaking and exploration. Table 4.9 shows some of the best individuals generated at each generation in the GA that can be used for further design tweaking. Both the best individuals at the 33<sup>rd</sup> and 40<sup>th</sup> iterations almost touch either the stress or the total area constraint.

Table 4.9. Some of the best individuals generated at each generation

Iteration	S1 (m)	S2 (m)	S3 (rad)	S4 (#floors)	Disp (m)	Stress (MPa)	Total area (m <sup>2</sup> )
1	15.1	6.1	1.855	23	0.146	221.07	13211
2	17.2	4.1	2.224	19	0.094	235.46	12296
17	15.5	3.4	2.356	22	0.124	247.82	12467.1
33	14.5	4	2.406	23	0.145	246.7	12193.4
40	14.5	3.2	2.406	23	0.133	239.28	12054

Figure 4.35 shows the geometries of the best individuals generated at each generation. For instance, one may choose the best individual generation at iteration one because it sufficiently has a good rate of twist, but also overall has a larger total occupiable area, especially at the top of the tower. Another may choose the one from the 33<sup>rd</sup> or 40<sup>th</sup> iteration because they have the highest rate of twist that satisfies the design intent.

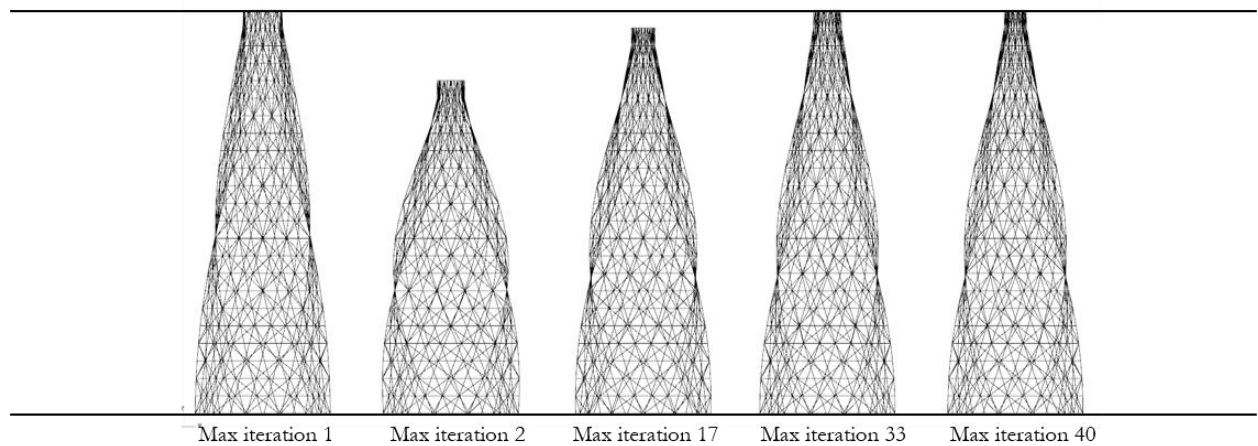


Figure 4.35. Geometries of the best individuals generated at each generation

Figure 4.36 shows the plot of the displacement and load vectors of the best individual generated at 40<sup>th</sup> iteration.

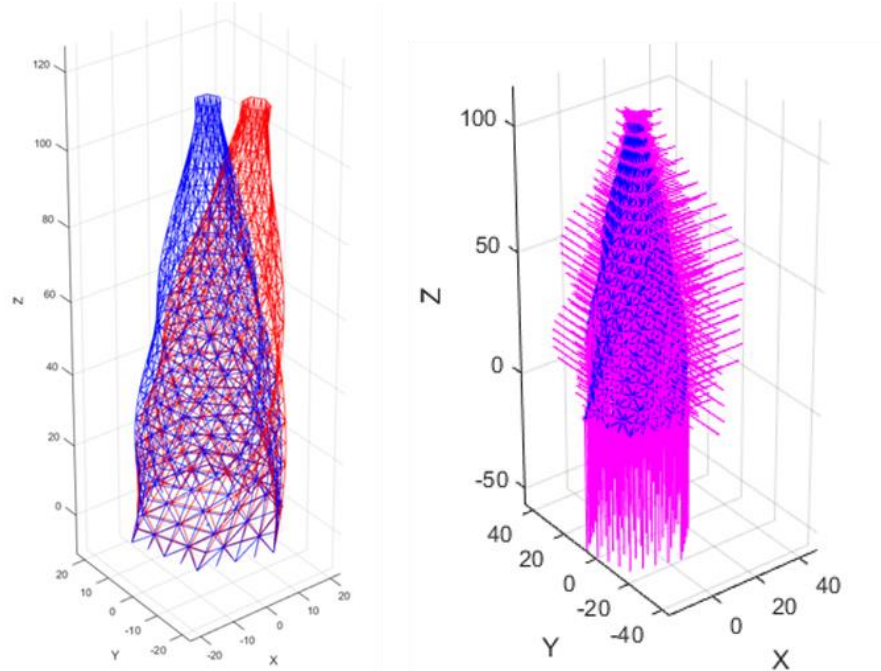


Figure 4.36. Displacement and load vectors plot of the best individual at iteration 40.

Figure 4.37 shows the displacement and stress plot of the best individual (iteration 40) using Abaqus. The plot shows that the maximum elemental von Mises stress occurs on the third floor, and is developed in the diagonal truss element.

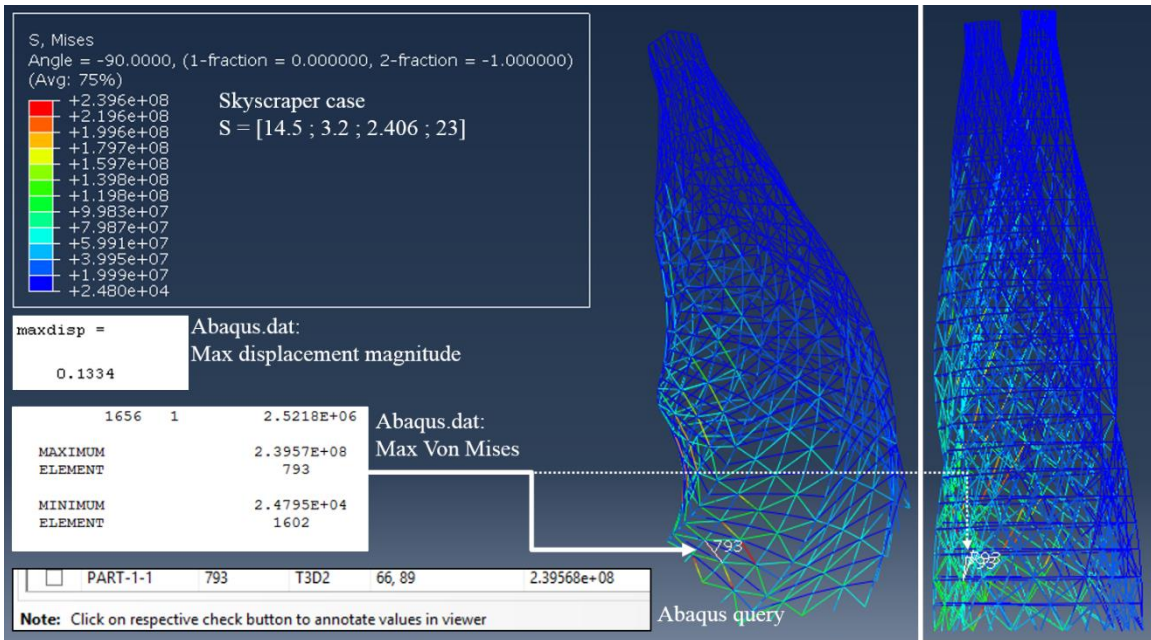


Figure 4.37. Displacement and stress obtained from Abaqus of the best individuals at iteration 40



Figure 4.38 shows the wireframe and the rendered mode of the structural elements at the optimum at iteration 40.

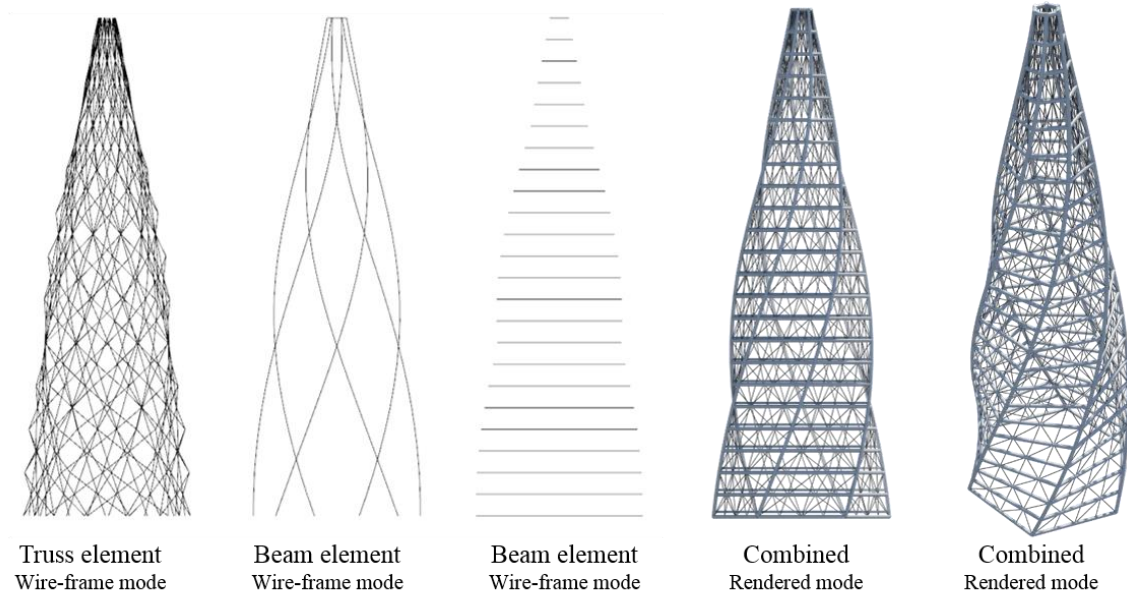


Figure 4.38. Wireframe and rendered modes of the optimum at iteration 40

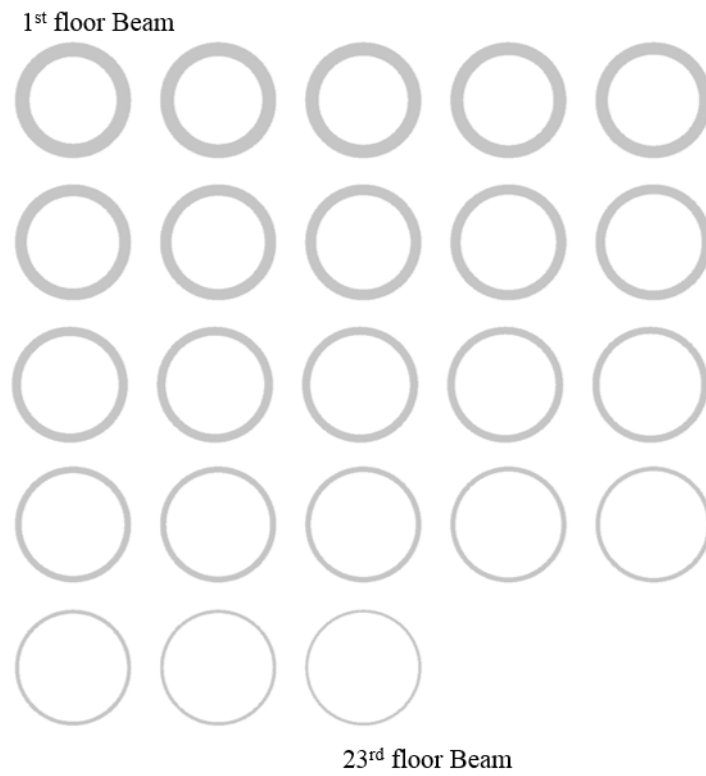


Figure 4.39. Cross-section of the beam at the optimum of iteration 40

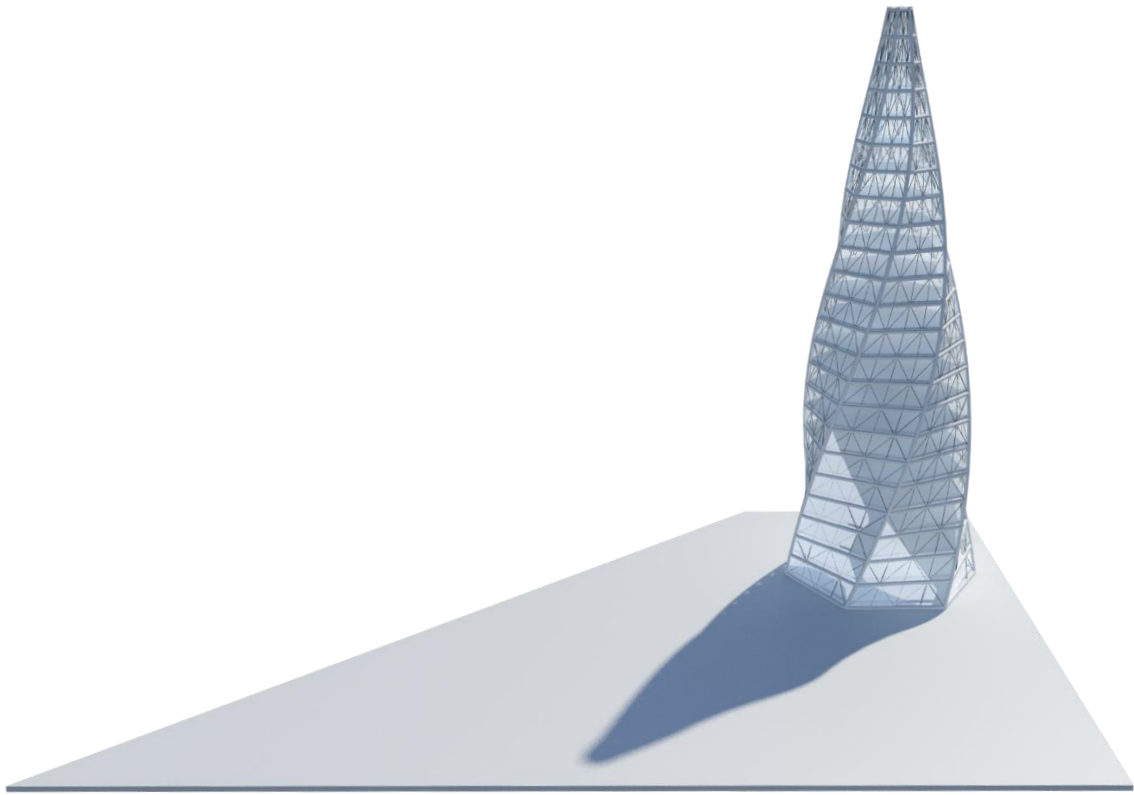


Figure 4.40. Rendering of the skyscraper design at the optimum of iteration 40



Figure 4.41. Closed up rendering of the skyscraper design at the optimum of iteration 40

Since the problem is formulated as a black-box problem, the Hessian cannot be found, and thus the convexity of the function cannot be determined. In fact, it is virtually impossible to ensure that the problem is unimodal (i.e., convex) or multimodal (i.e., non-convex and may include several local optimal solutions). An exhaustive search was then conducted to validate if the obtained result was the global optimum. The full factorial analysis of  $10 \times 10 \times 8 \times 8$  designs was used for the four factors, i.e., a total of 6400 experimental conditions were generated for the fully crossed design. The overall computational time of the exhaustive search is approximately ten times more expensive than iGAG. Table 4.10 shows the two best results when using the exhaustive search.

Table 4.10 Two best results when using exhaustive search

S1 (m)	S2 (m)	S3 (rad)	S4 (#floors)	Disp (m)	Stress (Mpa)	Total area (m <sup>2</sup> )
17.1	2.7	2.476	19	0.091	245.97	12005
16.0	2.0	2.564	21	0.106	248.51	12173

Figure 4.42 shows the feasible points and the optimum solution among the experimental conditions for  $s_4 = 21$  (i.e., 21 floors).

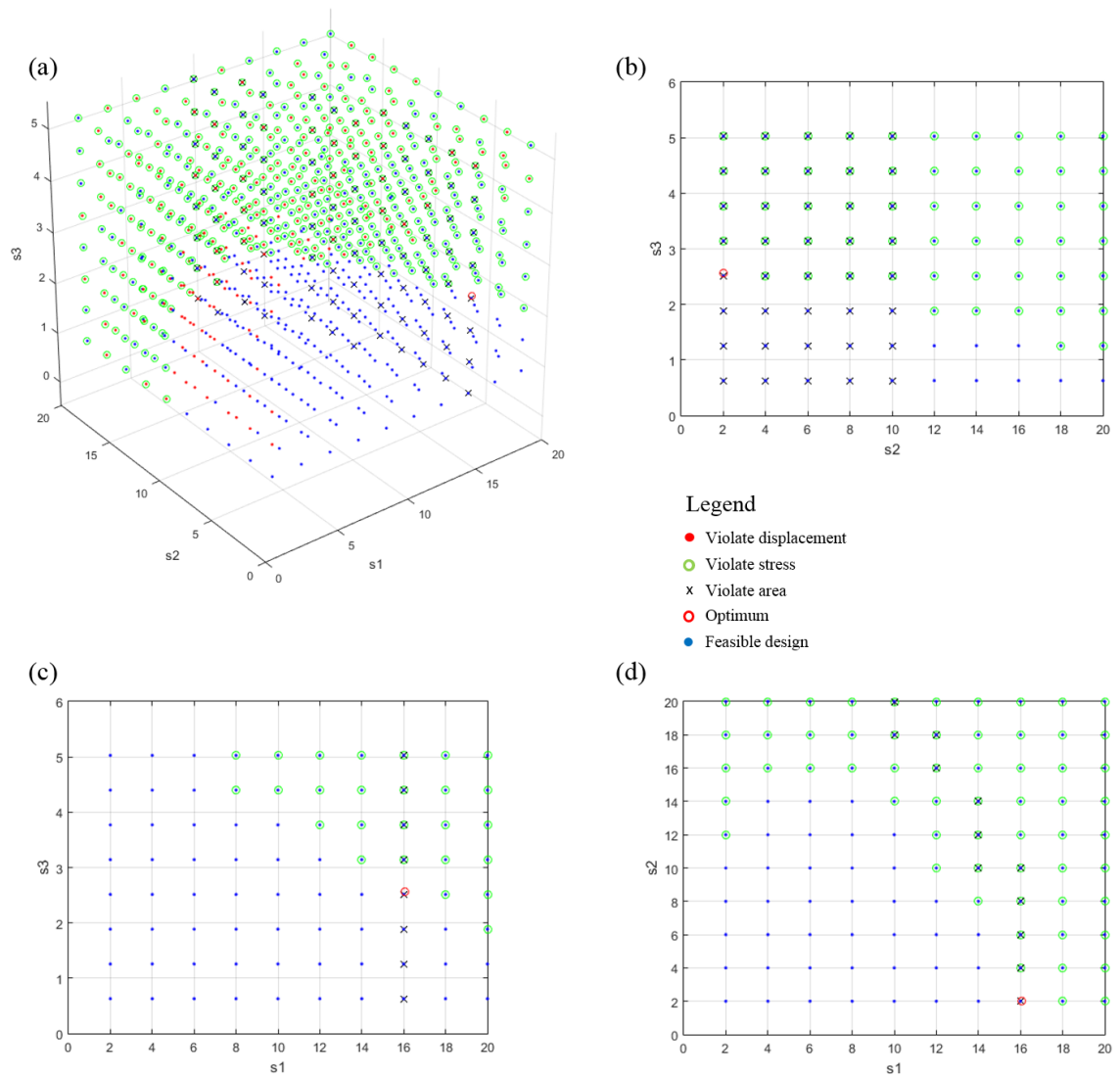


Figure 4.42. Graphical representation of the design space showing feasible designs and the optimum using the exhaustive search analysis : (a)  $s_1$  vs.  $s_2$  vs.  $s_3$  at  $s_4 = 21$ ; (b)  $s_2$  vs.  $s_3$  at  $s_1 = 16$  and  $s_4 = 21$ ; (c)  $s_1$  vs.  $s_3$  at  $s_2 = 2$  and  $s_4 = 21$ ; (d)  $s_1$  vs.  $s_2$  at  $s_3 = 2.513$  and  $s_4 = 21$

The two best candidates were identified when using the exhaustive search. From the results of both the presented method and exhaustive search, the problem was recognized as multimodal. The global optimum was shown to be the permutation with the rate of twist of 2.56 radian and 21 floors. Despite not achieving the global optimum, iGAG proves to be successful to find the improved candidates in a much shorter time compared to the exhaustive search, and thus, it is a viable form-finding technique for dealing with a complex mixed integer nonlinear optimization in the architectural design process.

## CHAPTER FIVE

### PILOT STUDY

The semi-structured interview was conducted using two Clemson University students. During the pilot study, there was only one research concern that were used to construct the interview questions and then later translated into three research concerns in the subsequent cycles of interviews. The modification was made based on the responses of the participants. Note that this type of modification of the research concerns is one of the characteristics of Grounded Theory described in figure 3.1. The responses of the participants were coded to generate the themes and theoretical construct describing the participants' subjective experience of incorporating structural optimization into the architectural schematic design process. The interview questions and findings are presented and discussed in this chapter.

#### **5.1 Semi-Structured Interview for the Pilot Study**

The semi-structured interview presented in the previous Section 3 is divided into three modules including the background module, the education module, and the demonstration module. It should be noted, however, that the interviews of the pilot study were conducted before the development of these three modules. Therefore, the interviews of the pilot included only an initial version of the demonstration module. Even though there was no recorded educational video for the participants to watch, the participants were introduced to the concept of structural optimization casually using an interactive learning process. The demonstration module was also conducted informally by showing how iMAGv1 should be utilized and without recorded educational video. This method is



called the direct demonstration method, which is considered to be less objective than using the recorded video. Brunvand (2010) suggested that using recorded video for educational purpose provide multiple perspectives of the same material rather than relying on a single viewpoint. Also, for certain topics and concepts video can help novice students who have lower prior knowledge process the taught concept more quickly (Reiser & Dempsey, 2007).

The demonstration module uses the planar frame structure as shown in figure 5.1. A pipe cross-section is used with the outside radius of 0.6m and wall thickness of 0.01m. The four-sided planar frame structure is fixed at two opposite sides, and the other two sides are free. Concentrated loads are applied at twelve nodes at the center of the frame system. The premade frame structure consists of two independent variables  $s_1$  and  $s_3$ . The width of the overall frame structure is set as the initial condition and is fixed throughout the optimization. The value of  $s_2$  is a function of  $s_1$ ,  $s_3$  and the width of the frame structure.

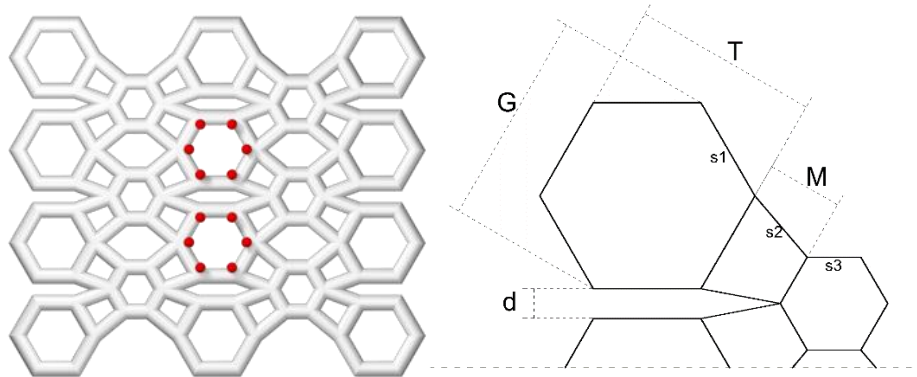


Figure 5.1. Planar frame structure for the pilot demonstration module

Table 5.1 shows the interview format used for the pilot study. The purposive sampling was used to sample Clemson University students from the year of 2015. The

two sampled participants were familiar with Grasshopper for parametric architectural design.

Table 5.1. Interview format for the pilot study

Interview Cycle: Pilot	
Theoretical Construct	None
Research Concern	To learn more about the response of architects if the consideration of structural optimization in architecture is incorporated into schematic design phase
Sampling	Purposive Sampling
Sampling Target	Masters Architecture Clemson students 2015 familiar with Rhino/Grasshopper
Interview Questions	
IQ1	What software do you use for architectural schematic design purpose?
IQ2	What software do you use for architectural developmental design purpose?
IQ3	How often do you conduct a structural evaluation during schematic design process?
IQ4	What is the usual method that you use for conducting preliminary structural evaluation?
IQ5	Do you have experiences in optimization either using a software or analytical method?  <u>BRIEF INTRODUCTION to STRUCTURAL OPTIMIZATION USING PAPER AND PEN</u>
IQ6	What is the usual method that you use for conducting preliminary optimization?  <u>DEMONSTRATION of iMAG using DIRECT DEMONSTRATION method</u>
IQ7	What do you think about <u>iMAG</u> if it would be implemented as a form finding method in architectural schematic design process?
IQ8	How can <u>iMAG</u> be improved ?

During the interview, the interviewee was given the opportunity to interact with the structure by changing the values of the design variables  $s_1$ ,  $s_3$ , and width. This opportunity was provided for the participants to have the general understanding of the geometric behavior of the structure. Then the interviewer ran the analysis-optimization process by setting the initial values ( $s_1$ ,  $s_3$ , and width) of the optimization. During the

iterative analysis process, the student could see the actual changes in the geometric sizes in the Rhino graphical interface at each iteration of the optimization. After the optimization had ended, the student could compare the optimized result with the basic configuration prior the optimization process related to the maximum displacement and stress of the structure. The goal of the optimization was to minimize the deflection of the structure while maintaining the maximum stress smaller than a preselected value.

The theoretical construct was not generated in the interview format of the pilot study. Thus the research concern and interview questions were developed purely based on the review of the literature.

## **5.2 Qualitative Data Analysis**

The answers of the qualitative interviews were analyzed using a coding procedure based on the Grounded Theory. For the pilot study, the interview was not transcribed, and the responses from the participant were recorded using paper and pen. The pilot study was done mainly for experimenting with the interviewing process and for practicing coding procedure on the feedback using Grounded Theory. However, even if the data were not fully transcribed and recorded, important repeating ideas were relatively easy to be captured from the participants.

Table 5.2 shows the relevant repeating ideas captured from the participants' responses.

Table 5.2. Coding of the pilot study interview responses

Interview Cycle: PILOT	
<b>NEED FEWER TECHNICALITIES</b>	
<b>Technical difficulties with iMAG</b>	Students mentioned some technical difficulties in following the demonstration of iMAG. They feel it's difficult because they don't understand the variables corresponding to the optimization process and the Matlab script looked complicated.
The script is confusing I feel confused with the math	
<b>Use casual inspection for structural evaluation</b>	Students prefer casual inspection of the structural performance of their design rather than using analytical method for structural evaluation. Although they mentioned that they had structural analysis courses for conducting the structural evaluation analytically.
I don't use numbers I use reasoning for evaluating structure Such as prototyping for evaluating structure	

For convenience in telling the theoretical narrative of the findings, the repeating ideas are written in italic font, the themes are underlined, and the theoretical construct uses capital letters. The first memo at the right side of the coding explains the students' preferences of casual inspection for structural evaluation. This memo is based on the repeating ideas (*I use reasoning for evaluating structure & I don't use numbers*). This means that both analytical and numerical analysis of structure are usually not conducted by the students, although they admitted that they had taken structural analysis courses in the past. The theme (using casual inspection for structural evaluation) emerged from the repeating ideas involving these issues.

The second memo in the coding explains the phenomenon after the students were introduced to iMAGv1. The students mentioned that the Matlab script for running the structural optimization was confusing. During the demonstration of iMAGv1, certain variables involving setting up the initial values, lower bound and upper bound values of the optimization needed to be adjusted to run the optimization process. To adjust the values, the students had to change the values in the Matlab programming interface. This turned out to be a significant problem for the participants as they repeatedly mentioned

how the Matlab script and the math embedded in the program were confusing to them. The theme (Technical difficulties with iMAGv1) emerges from the repeating ideas related to those issues.

Both of the themes (using casual inspection for structural evaluation and technical difficulties with iMAGv1) described the preference of simplicity and ease for conducting the structural evaluation. These can be seen from how the students dislike the usage of numbers and the technical complexity of the Matlab script. The initial theoretical construct of NEED FEWER TECHNICALITIES is the abstraction and the most general concept of both themes.

### **5.3 Theoretical Narrative for the Pilot Study**

From these two interviews, it is still not clear whether the students consider structural analysis and optimization to be important or not in the architectural schematic design phase. Their answers of saying “yes, it is important” may suggest that they are just polite. The reason for this doubt is that although there are structural analysis courses that teach them the skills for conducting structural analysis as they mentioned, the students much prefer to use reasoning without numbers for structural analysis as described in the repeating ideas (*I use reasoning for evaluating structure & I don't use numbers*). From the architectural curriculum perspective, structural analysis is considered to be important since courses are designed such that students gain the necessary techniques to analyze the structure of their design analytically. However, the students do not seem to be comfortable with applying the taught structural analysis techniques in their design studio courses and prefer to use reasoning for casually evaluating the structure. These

phenomena may be because the design studio courses usually only require students to design buildings in which structural evaluation and decision making can entirely be based on the prescriptive code and rules of thumb. If it is the case, then both analytical and performance based analysis are not useful since they are more advantageous for analyzing unconventional structures or organic shapes.

From the theoretical construct generated by the two themes (using casual inspection for structural evaluation and technical difficulties with iMAGv1), the iMAGv1 tool was altered to reduce the apparent technicality. This improvement mainly includes a new graphical interface to run the structural optimization from Matlab. Using the developed graphical interface, the students do not have to be faced with the technicalities of the Matlab script, and can directly input the numbers corresponding to the design variables of the structural optimization of the problem. This modification was used in subsequent interviews.

Also, the new education and demonstration modules of the structural optimization of iMAGv1 were also used in the subsequent interviews, which will help the participants understand certain concepts of structural analysis and optimization.

## CHAPTER SIX

### FIRST CYCLE OF GROUNDED THEORY

The first cycle of Grounded Theory included two main goals. The first goal was to continue the development of the iMAGv1 tool and improved its user friendliness to increase its efficiency. The second goal was to conduct the qualitative interviews and interpretation of the results to examine the research concerns. Research concerns and interview questions were modified accordingly based on the responses from the pilot study.

#### **6.1 Tool Improvement Based on the Pilot Study**

The development of iMAGv1 continued to increase its efficiency based on the pilot study. The user-friendliness of the iMAGv1 tool was improved by focusing on the graphical user interface, and the modularization of specific user commands to streamline the process of generating a structural analysis and optimization problem.

#### **6.2 Modification on the Qualitative Research Tasks**

##### 6.2.1 Modification of the Interview Format

The interview format was modified based on the pilot study by adding the background, education, and demonstration modules in the form of video recordings (see Appendix B for the interview module transcripts).

##### 6.2.1.1 Background Module

The background module was incorporated into the interview. Based on themes (using casual inspection for structural evaluation and technical difficulties with iMAGv1) generated in the pilot study, the idea was to inform the participants about the context of

the research. As mentioned in Chapter 1, this research is contextualized under the assumption that architects need the freedom of designing complex organic shapes for which prescriptive codes and rules of thumb are not applicable. This statement is explained by describing the difference between the prescriptive and performance based approach to the participants at the beginning of each interview.

Figure 6.1 shows one of the slide presentation used for briefly introducing the difference between a performance-based approach and a prescriptive approach to the participants.

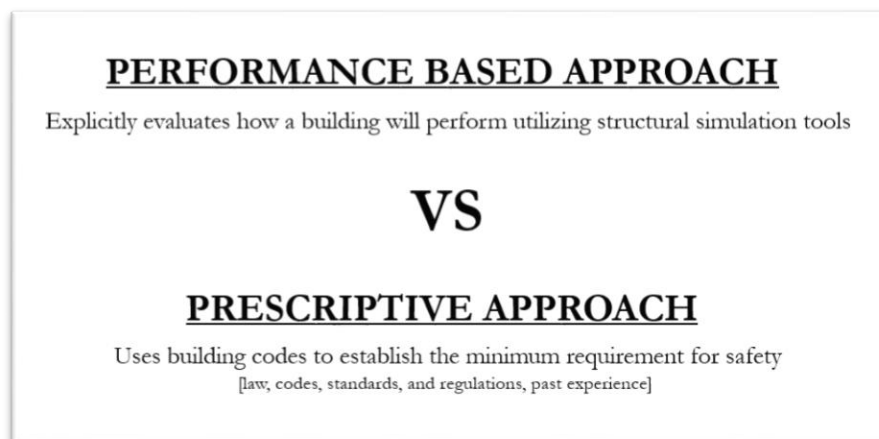


Figure 6.1. Difference between performance-based and prescriptive approach

Brief comparative studies of buildings that typically used either prescriptive or performance-based approaches were presented. The idea of introducing this notion is for the participants to be able to differentiate between the structural systems that can be analyzed using a prescriptive code (i.e., regular grid geometries with conventional and mainstream construction materials) and the structural systems that should be analyzed using a performance-based approach for structural analysis (i.e., complex unconventional



organic structures that can be proven to pass code requirements only by demonstrating their performance). Figure 6.2 shows examples of typical structural systems that use prescriptive and performance-based approaches. The same examples are presented for the participants to understand the difference.

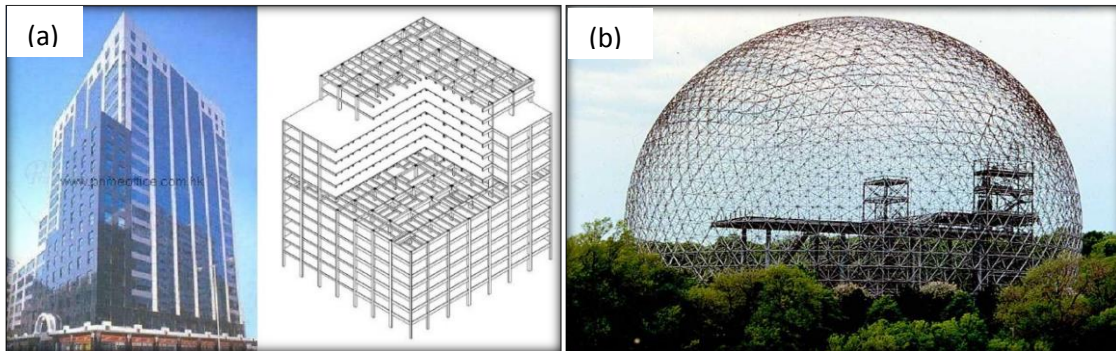


Figure 6.2. Examples of prescriptive vs. performance-based: (a) examples of typical structures using prescriptive approach and (b) performance-based approach

#### 6.2.1.2 Education Module

The education module was incorporated into the interview. In the pilot study, structural optimization was introduced and discussed briefly at the beginning of the interview. Alternatively, a recorded video was presented to each participant to ensure consistency between interviews.

The problem shown in figure 6.3 was used for the educational module. The introduction covers the idea of constructing a one-variable sizing structural optimization model of a simple truss structure. Nodes 4 and 5 of the structure are fully restrained, and a concentrated downward load is applied to node 3. The material for each member is assumed to be a steel pipe with circular cross section. The objective function is to minimize the maximum nodal displacement occurring in node 3 with respect to  $x$ , where  $x$  is the distance between nodes 1 and 2. A stress constraint is imposed by restricting the

normal stress,  $\sigma_e$ , of each member to be less than the yield strength of a given steel (250 MPa). As  $x$  varies, nodes 1 and 2 move horizontally and symmetrically with respect to the position of node 3.

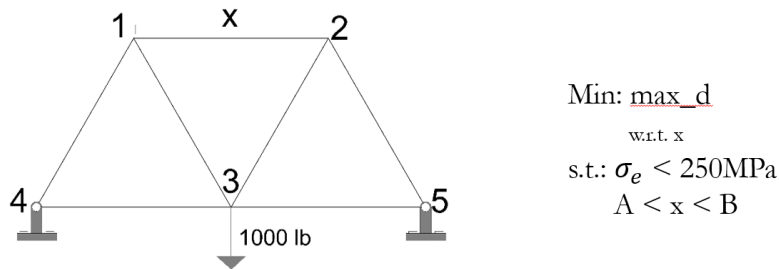


Figure 6.3. Structural optimization problem used in the educational module

A question is asked to the participant after the education module to examine the participant's subjective experience on the matter of structural optimization. This is used to examine issues related to the research concerns 1 and 3.

### 6.2.1.3 Demonstration Module

The demonstration module was modified and included in the interview. In the pilot study, iMAGv1 was introduced casually with a direct demonstration given by the interviewer. In the next interviews, a recorded video is used for the purpose of objectivity and for improving the clarity of how structural optimization works using iMAGv1.

Figure 6.4 shows the structural optimization interface of iMAGv1, which was developed in order to respond to the themes and theoretical construct generated from the pilot study, i.e., to reduce the technicality involved when using the tool. The same interface can be used for any structural truss problems with three design variables. This graphical user interface was developed such that the user never sees the Matlab scripting interface.

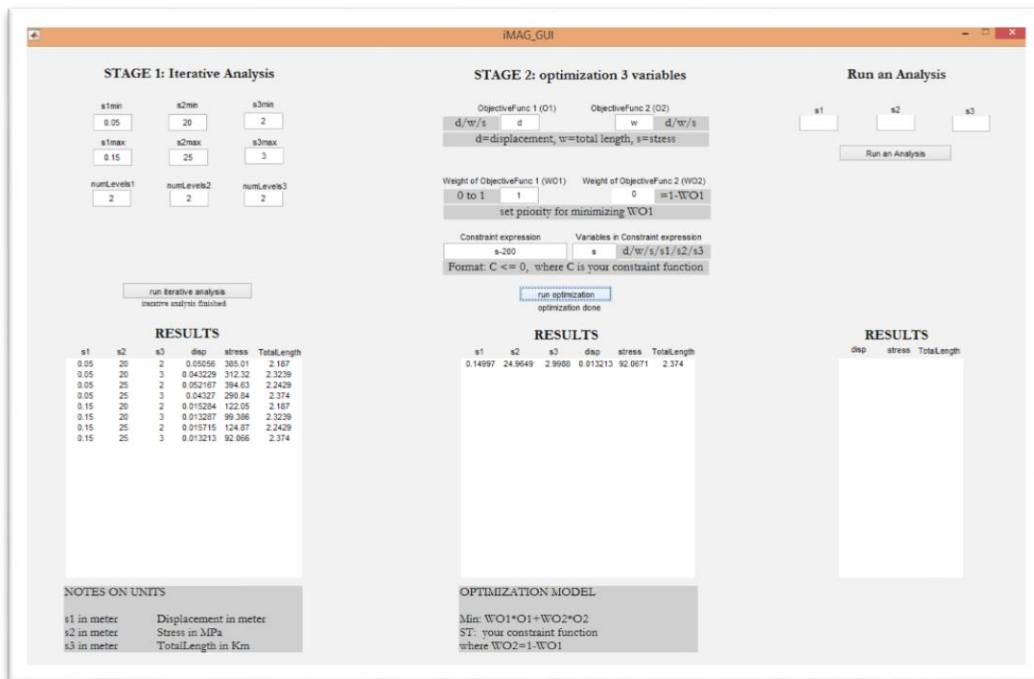


Figure 6.4. iMAGv1 structural optimization interface

As mentioned above, currently, the execution of iMAGv1 includes two steps, namely, the development of the response surface (referred to as “Stage 1: Iterative Analysis”), and the optimization process (referred to as “Stage 2: Optimization”). The second stage can be run only after the first stage has finished generating the analysis results.

## 6.2.2 Qualitative Data Analysis

### 6.2.2.1 Transcription of Interview

For the pilot study, manual coding was used for the data interpretation. This may have reduced the justifiability aspect of the findings. Thus the data transcription of this and subsequent interview cycles were fully made for the actual coding process to ensure justifiability of the interpretations as described in Section 2.2.

#### 6.2.2.2 Using Qualitative Data Analysis Software for Coding

The coding process of the pilot study was done manually. In this and subsequent interviews, the coding process was done with a qualitative data analysis software MAXQDA for a more elaborated qualitative data analysis. Conducting hands-on learning in MAXQDA was part of the research plan before the actual coding process.

#### 6.2.2.3 Interview Format for Interview Cycle 1

The modification of the interview format cycle one was made based on the themes and theoretical construct generated from the pilot study. Table 6.1 shows the interview format for the interview cycle 1. Each module was presented using prerecorded videos, and each module was followed by a set of questions corresponding to the content of the previously watched video.

Table 6.1. Interview format for interview cycle 1

Interview Cycle: 1	
Theoretical Construct	Need fewer technicality
Research Concerns	<p>Examine the implementation of structural optimization in the architectural schematic design phase.</p> <p>Investigate the accessibility and usability of existing architectural structural optimization tools.</p> <p>Study the interoperability and integration of architectural parametric CAD tools and engineering analysis and optimization tools as well as the usability of these tools.</p>
Sampling	Theoretical and snowball sampling
Sampling Target	Architecture Clemson students 2016-2017 familiar with Rhino/Grasshopper
Interview Questions	Cycle 1
	<p><u>[WATCH VIDEO 4.30 to 5 minutes] BACKGROUND MODULE, LIMITED TO BUILDINGS and STRUCTURAL SYSTEM WHERE RULES OF THUMB AND PRESCRIPTIVE APPROACH IS NOT APPLIED</u></p>
IQ1	In which stage(s) of the architectural design process would you perform structural evaluation of your design?
IQ2	How would you perform structural evaluation of your design?
IQ3	If structural analysis was incorporated into the architectural schematic design stage as a form finding method, what would be your thoughts?
	<p><u>[WATCH VIDEO 4.30 to 5 minutes] EDUCATION MODULE of STRUCTURAL OPTIMIZATION for FORM FINDING PROCESS</u></p>
IQ4	What do you think about this concept of structural optimization for form finding process?
	<p><u>[WATCH VIDEO 19 to 20 minutes] DEMONSTRATION of the Research Version of iMAG (the developed structural optimization tool)</u></p>
IQ5	What do you think about the current research version of iMAG, if it would be implemented as a form finding method in architectural schematic design process?
IQ6	Assuming that iMAG is improved to be able to operate on flexible amount of design variables, what do you think if this would be implemented as a form finding method in architectural schematic design process?
IQ7	How can the current research version of iMAG be improved?

### 6.3 Theoretical Narrative for the First Cycle of Grounded Theory

Three Clemson University’s architecture students are sampled for the interview.

MAXQDA software is used to conduct the coding from the interview transcripts.

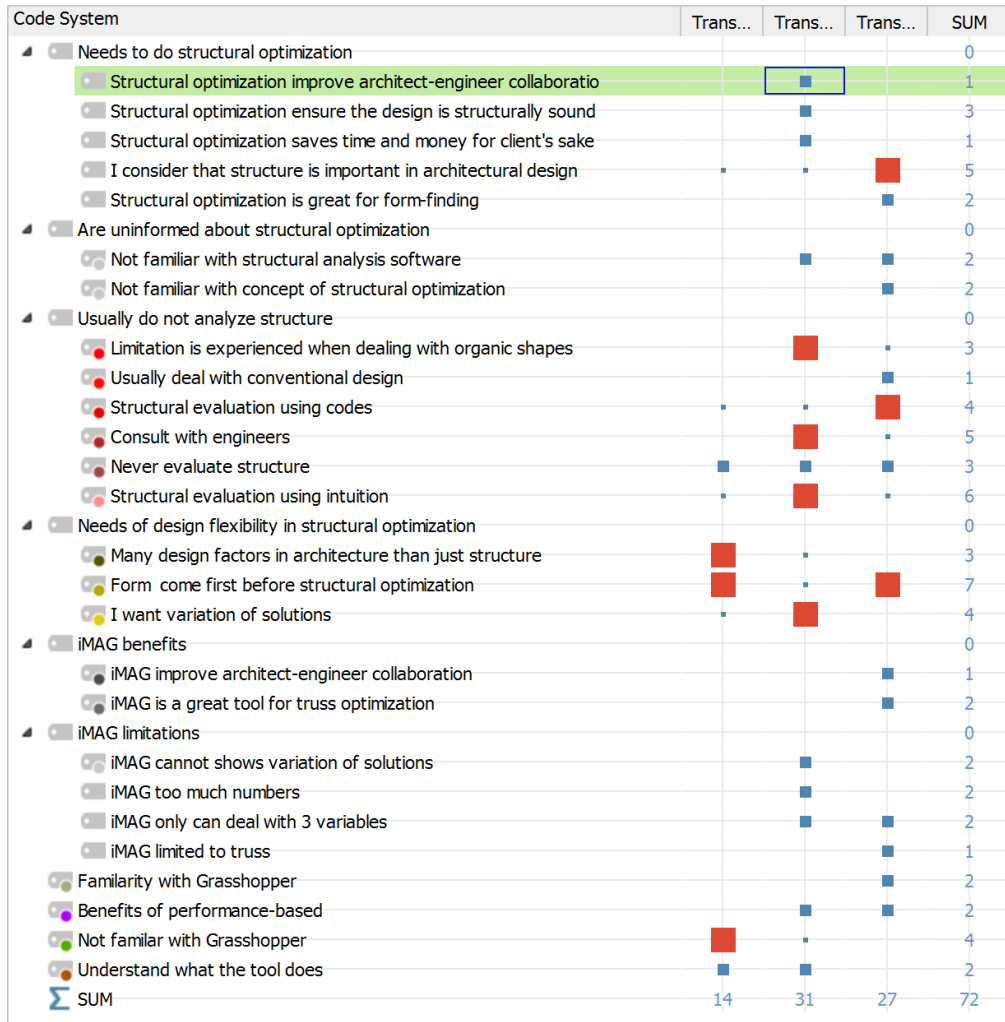


Figure 6.5. Repeating ideas and emerging themes using MAXQDA (1<sup>st</sup> cycle). The three columns labelled “Trans...” correspond to the three interviewees. The column labelled “SUM” correspond to the occurrence frequency. Small dots (few), medium blue squares (several), and large red squares (many) illustrate the number of repetitions of a given entry for a given interviewee.

For convenience in telling the theoretical narrative of the findings, the repeating ideas are written in italic font and the themes are underlined. Students expressed that they are *not familiar with structural analysis software, and with the concept of structural optimization*. Thus they are uninformed about structural optimization both in application and theory, and thus never implement the routine for the design task. In the design studio, they *usually deal with conventional design* in which *structural evaluation using codes* is sufficient to be used. In their design studio, they are *never required to evaluate the structure*. Also, they feel that limitations are experienced when dealing with organic shapes. One of them *consult with engineers when such task occurs*, and another carry out the *structural evaluation using his intuition solely*. Thus, architecture students usually do not analytically analyze structure because they are not academically trained to do that, and they are not required to conduct such task. Figure 6.6 shows the code-subcode-segments model generated from MAXQDA to visualize the connection between the theme do not analytically analyze the structure and its corresponding repeating ideas and

the corresponding samples of retrieved segments.

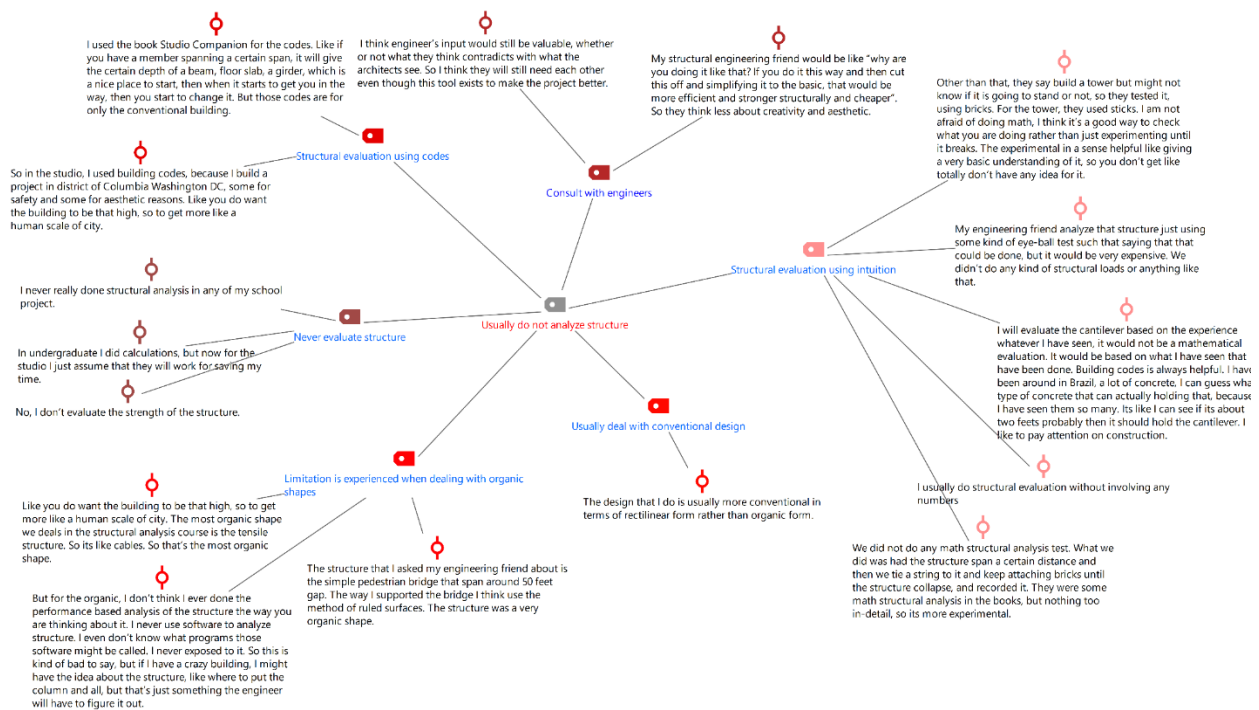


Figure 6.6. Code-subcode-segments model for the theme “usually do not analyze the structure.” The text included in this figure is not meant to be read, but rather to show the organizational structure.

Although the architectural students commonly do not analyze the structure, they acknowledge that *structure is important in the architectural design*. After exposing the structural optimization tool to the students, they seem to get a better insight into what the structural optimization routine does. They become interested in structural optimization by explicitly mentioning that the *structural optimization routine is great for form-finding* because they understand that the approach allows for designing organic structure while at the same time *ensures that the design is structurally sound*. They also mentioned about the way the structural optimization approach for form-finding can *potentially improve architect-engineer collaboration*, and eventually *save time and money for the client’s*



sake. Thus, from both the education and demonstration modules, the students can already see the benefits when the approach is realized in the architectural design practice. Thus, although the students usually do not analytically analyze the structure, after being exposed to the structural optimization approach, they become interested, and see the needs to do structural optimization for form-finding in architecture. Figure 6.7 shows the code-subcode-segments for the theme “needs to do structural optimization.”

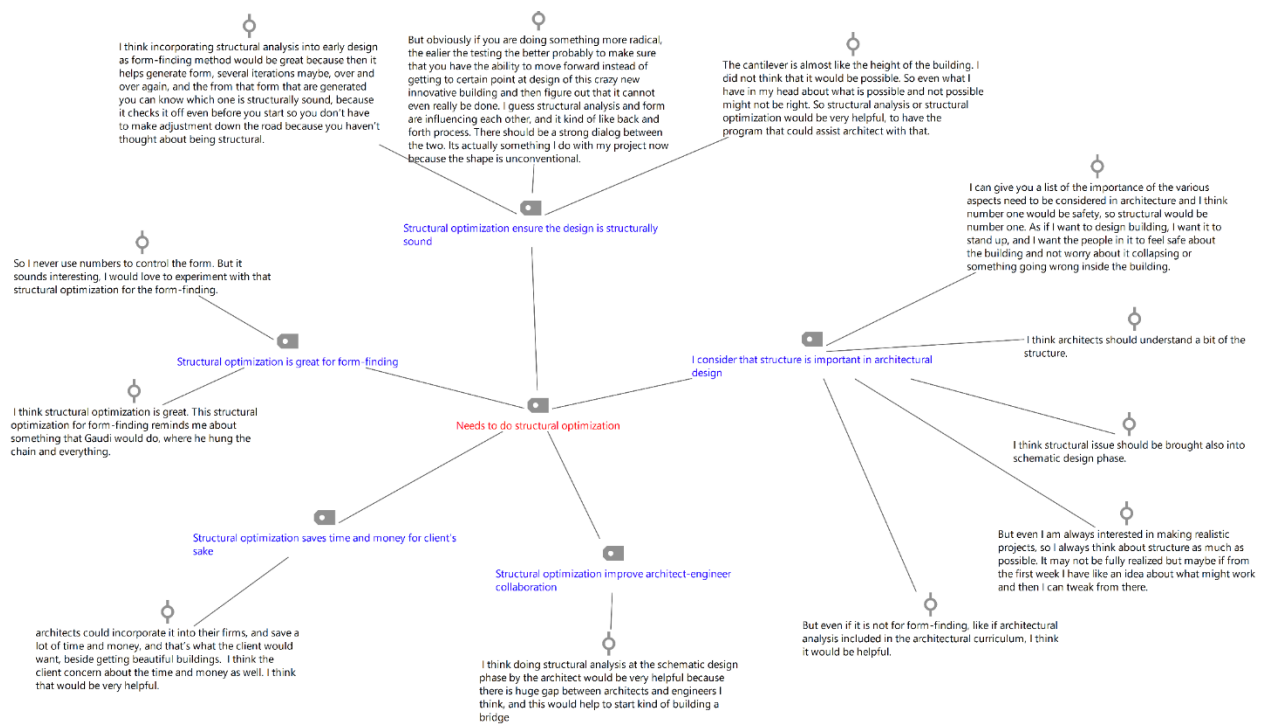


Figure 6.7. Code-subcode-segments model for the theme “needs to do structural optimization.”

After being introduced to the notion of structural optimization through education module, although they are interested in it, the students mentioned that *there are many design aspects other than structural performance*. The students often emphasize that they want *the form to come first before the structural optimization*. They do not want to be constrained by technicalities involved in the structural optimization routine. The students

desire *variation of optimal solutions* in optimization routine so that the innovative design exploration can still be carried out even after the optimization procedure has been conducted. One of the participants describes:

*“... I would like to see the variation. To see what was option one, option two, to see what those different forms were. You might see that, alright, this is the most structurally optimal, but that other design looked way cooler or more aesthetically pleasing... It would be great if the tool can show what would be acceptable, not necessary the top one, but the top five for example. The building does not have to be one hundred percent structurally sound, but just need to pass a certain threshold... I still want to do something more after the optimization considering the style and all within the allowable threshold after the optimization.”* (Appendix D)

Thus, the students can already see the converging nature of structural optimization routine, and they somehow felt constrained with its nature that seems to limit the freedom of the design exploration. In other words, they need design flexibility in the structural optimization. Figure 6.8 shows the code-subcode-segments model for the theme “needs design flexibility in the structural optimization.”

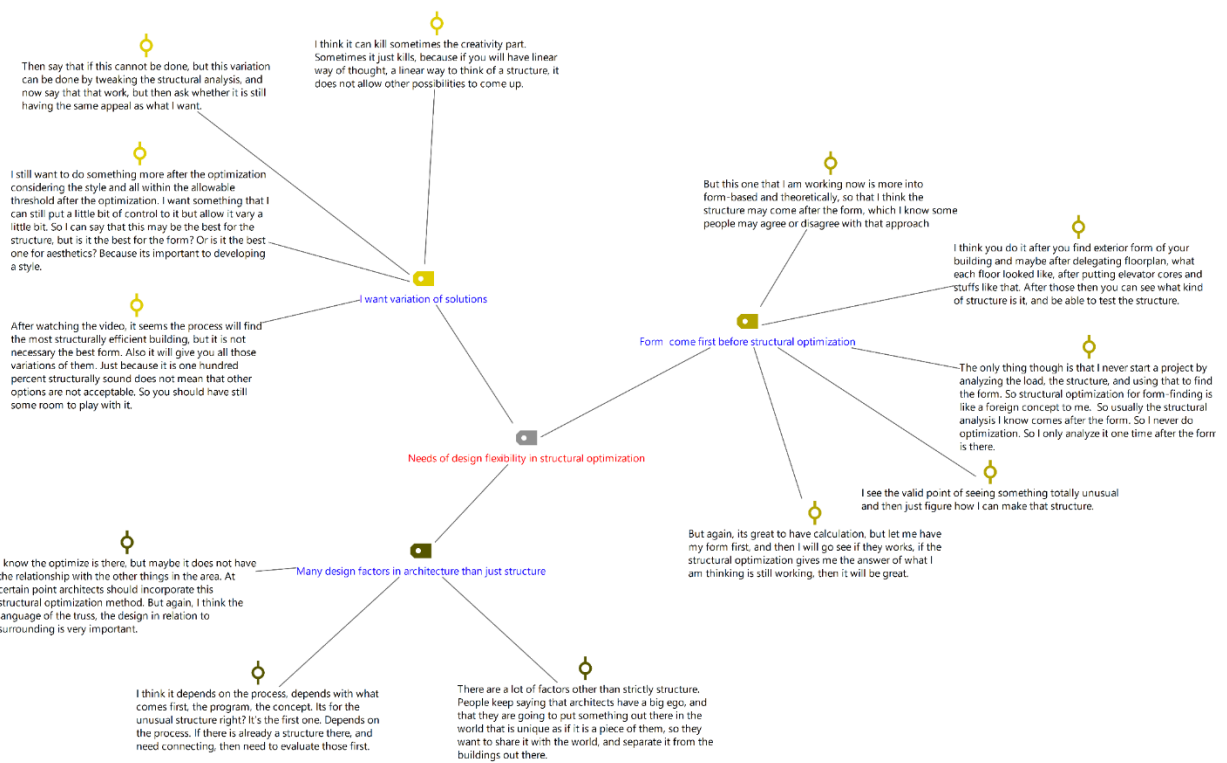


Figure 6.8. Code-subcode-segments model for the theme “needs design flexibility in structural optimization.”

After the demonstration module of the tool, students show their interest in iMAGv1 and mentioned that *it is a great tool for the form-finding of the organic truss structure*. They also can see that the *iMAGv1 can potentially improve architect-engineer collaboration* if implemented in the actual design practice.

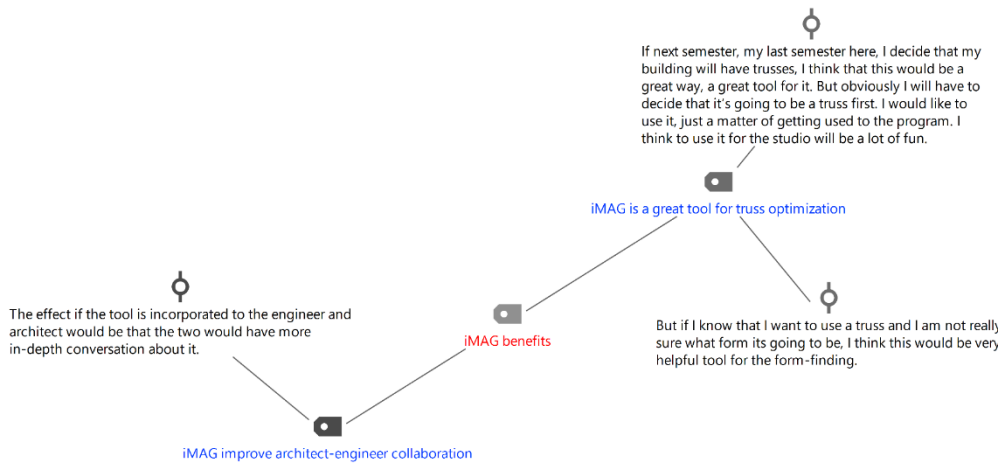


Figure 6.9. Code-subcode-segments model for the theme “iMAGv1 benefits.”

However, despite iMAGv1 benefits, students also described some limitations that they felt regarding the tool. Some of them mentioned how they wanted to have more control over the optimization. However, the research version of *iMAGv1 can only deal with three variables*. Some of them also mentioned that about the way *iMAGv1 involve too many numbers*, despite the fact that the graphical user interface for conducting the optimization had been made to increase the usability of the tool. Perhaps the most notable opinion is that some students mentioned *that iMAGv1 could not show a variation of solutions*. This idea is consistent with their previous notion of *desiring variation of optimal solutions*, after the education module, but before demonstration module.

## CHAPTER SEVEN

### SECOND CYCLE OF GROUNDED THEORY

#### 7.1 Modification on the Qualitative Research Tasks

##### 7.1.1 Modification on the Target Population

Both the pilot study and the first cycle of the Grounded Theory used Clemson University students as the target population. The second cycle of the Grounded Theory continued using Clemson University architectural faculty as the target population. This cycle is also interested in interviewing faculty that have experiences in architectural practice for more than five years. Using a snowball sampling method, seven faculty members were identified for participating in the interview. Table 7.1 shows the demographic data of the participants that include the interviewees from the first cycle of the Grounded Theory but excludes those from the pilot study.

Table 7.1. Demographic data of the participants

Experience / Participants	A	B	C	D	E	F	G	H	I	J
Engineering academic						Ph.D.				Bachelor
Architecture academic	Master	Bachelor	Master	Master	Ph.D.		Ph.D.	Master	Ph.D.	Master
Architecture practice				5 years	7 years		8 years	8 years	9 years	4 years

During the pilot study, only one research concern was used to construct the interview questions. It was then modified into three research concerns in this cycle of the interview. The modification was made based on the responses of the participants. Note that this type of modification of the research concerns is the essential characteristic of Grounded Theory described in figure 3.1

### 7.1.2 Modification of the Interview Format

Changes in the central questions were made based on the first cycle of the Grounded Theory. The questions of the second cycle emphasized more in the aspect of the collaboration between architect and engineer if structural optimization routine was incorporated into the architectural schematic design phase. Table 7.2 below shows the interview format for the second cycle.

Table 7.2. Interview format for interview cycle 2

Interview Cycle: 2	
Theoretical Construct	Although uninformed, there are needs to do structural optimization. iMAG needs variation of solutions for allowing design flexibility.
Research Concern	Examine the implementation of structural optimization in the architectural schematic design phase.  Investigate the accessibility and usability of existing architectural structural optimization tools.  Study the interoperability and integration of architectural parametric CAD tools and engineering analysis and optimization tools as well as the usability of these tools.
Sampling	Theoretical and snowball sampling
Sampling Target	Architecture Clemson faculties 2016-2017 familiar with Rhino/Grasshopper and have architecture practice experiences.
Interview Questions	Cycle 2
IQ1	<u>[WATCH VIDEO 4.30 to 5 minutes] BACKGROUND MODULE, LIMITED TO BUILDINGS and STRUCTURAL SYSTEM WHERE RULES OF THUMB AND PRESCRIPTIVE APPROACH IS NOT APPLIED</u> Based on your experience and knowledge, what do you think about the contemporary collaboration between architects and engineers during the architectural schematic design phase?
IQ2	Please tell me about your knowledge and experience in structural analysis and optimization.
IQ3	<u>[WATCH VIDEO 4.30 to 5 minutes] EDUCATION MODULE of STRUCTURAL OPTIMIZATION for FORM FINDING PROCESS</u> What do you think if structural optimization is used as a form-finding method in the architectural schematic design phase?
IQ4	<u>[WATCH VIDEO 19 to 20 minutes] DEMONSTRATION of the Research Version of iMAG (the developed structural optimization tool)</u> What do you think about the current research version of the tool _if it would be implemented as a form-finding method in architectural schematic design process?
IQ5	How can the current research version of the tool be improved?
IQ6	What do you think if the tool becomes a substitute for engineer's job by postponing their involvement in the architectural design process?

## 7.2 Theoretical Narrative for the Second Cycle of Grounded Theory

Seven Clemson University's faculty members were sampled for the interview. A qualitative data analysis software, MAXQDA, was used to conduct the coding from the interview transcripts. Figure 7.1 shows the list of themes generated from the repeating

ideas. Both the themes and repeating ideas are used to create a theoretical narrative that tells the story about participants' responses to the idea of incorporating the structural optimization into architectural schematic design phase in the context of the contemporary architect-engineer collaboration.

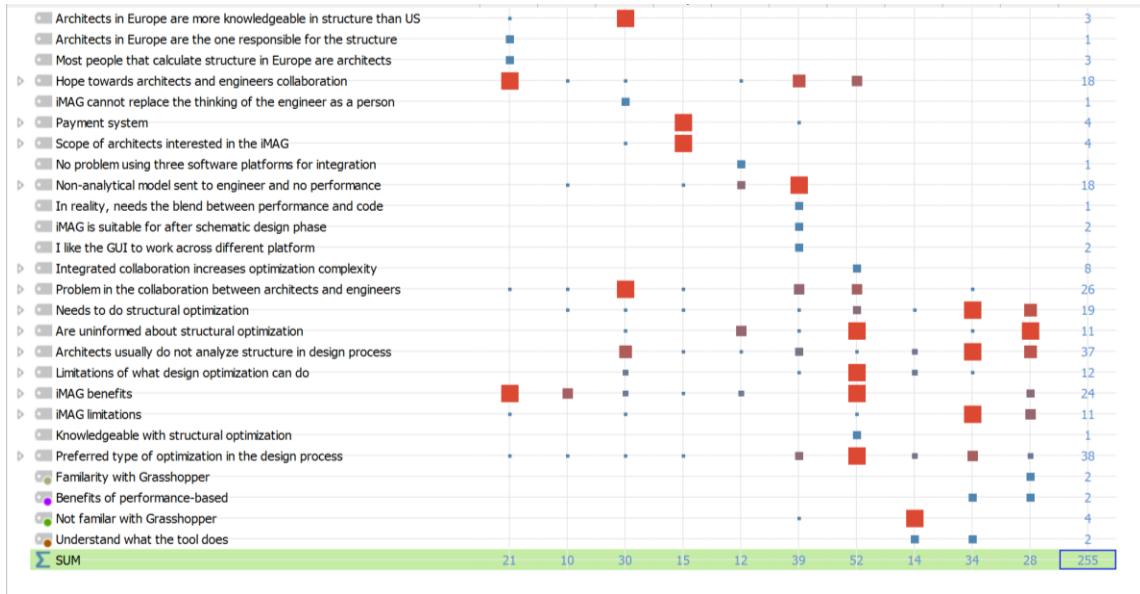


Figure 7.1. Emerging themes using MAXQDA (2<sup>nd</sup> cycle). The ten columns correspond to the ten interviewees. The size and colors of the dots represents the number of repetitions of a given entry for a given interviewee.

### 7.2.1 Theoretical Narrative Related to the First Research Concern

For convenience in telling the theoretical narrative of the findings, the repeating ideas are written in italic font and the themes are underlined>. Faculty members that have the experiences working in architectural practice *deal mostly with the conventional structure* in which prescriptive code is available for structural sizing. They expressed that in practice, *architectural practitioners never analytically evaluate structure*, and *most architects do not have profound knowledge about structure*. After watching the education



video about the introduction to structural optimization, a faculty mentioned that typically architects could deal with the understanding of the geometrically related constraint in the optimization, but it is not the case for the structurally associated layer of optimization such as stress constraint. Most likely, when the code is not available, *architects use intuition to evaluate the structure or consult with engineers*. This is most often to be found in the case of organic structure, where, unlike the orthogonally assembled rigid frame structure, the structural grid is harder to be defined. In many cases of organic structure, the skin of the buildings becomes the primary structure that supports all the live and dead loads. Faculty expressed that *limitations are experienced when dealing with organic shapes* that require an unconventional method to assess the structure, and the only way to resolve this in practice is usually to *consult with engineers*. This is particularly the case for a *competition project, where time is crucial and organic structure is much more implemented into the design*. Typically, *only shell CAD model sent to a structural engineer in the case of designing organic structures*. Both students and faculty described that there are limitations of what they can do in terms of evaluating structure, and since there are also *many design aspects other than the structure in architecture*, it is common for architects for not going into the technical details such as analyzing the structure, both in practice and academic.

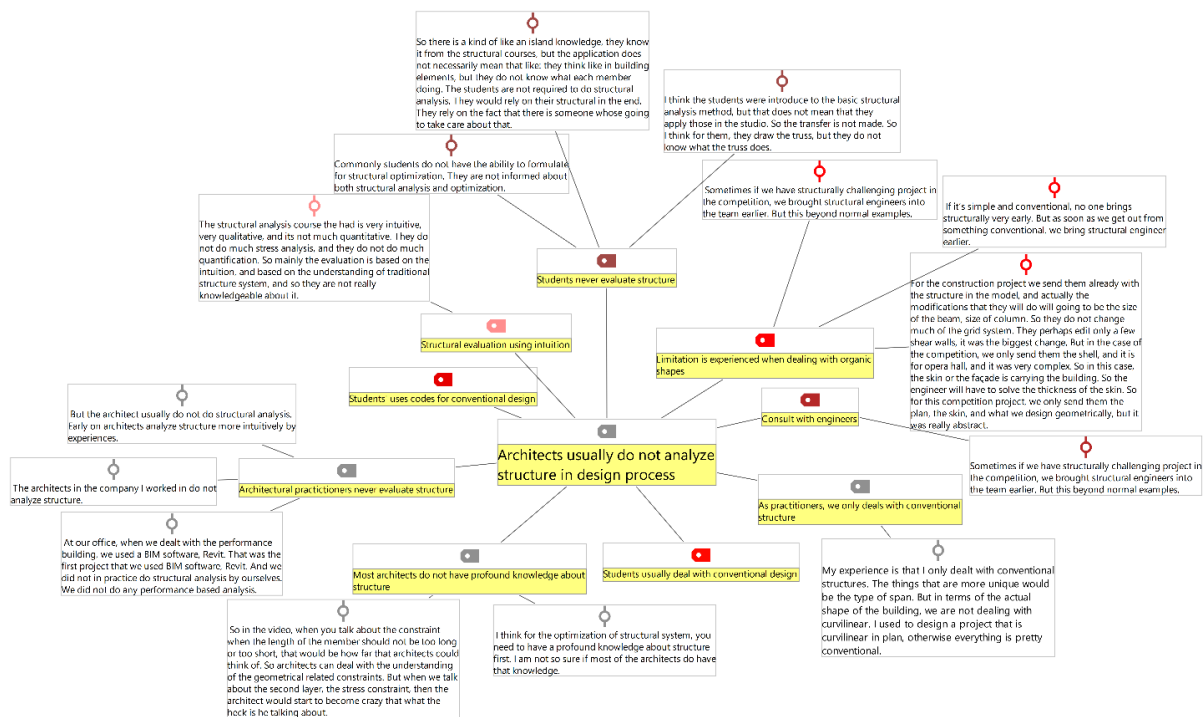


Figure 7.2. Code-subcode-segments model of the theme “architects usually do not analyze the structure.”

On the other hand, in much architectural design practice, design optimization in architecture is commonly done intuitively, if any, similar to *Gaudian optimization for the design of the organic structure*. Architecture students, academia, and practitioners are commonly uninformed about both analytical and numerical structural optimization routine. Academically, at least at Clemson University, *students are not familiar with structural analysis software, and the concept of structural optimization*. A faculty mentioned that the structural analysis courses that the students have are very intuitive and qualitative in nature. At least academically, faculty members expressed that the students do not do structural optimization because they are educated in that way, and they are not expected to do it. These phenomena related to the U.S. architectural educational system are different compared to those in Europe. For instance, the structural analysis courses in European

architectural educational systems are highly technical. A faculty with European architectural degree from Europe describes:

*“... The architecture degree in Europe I got was from Polytechnic, and it was considered equivalent to engineering degree, so the level of difficulty the structure is very high and actually as soon as you were finished from university and got the license you were legally allowed to calculate your own structure, and most of the people that calculate structure there was architect, not engineer.... Architecture in Europe is not in fine art or humanity, but it's more on the polytechnic institute, within the engineering... In my curriculum, we have two years in Calculus, and algebra before we touch any structure... I think not so many architects in the US are exposed with structural optimization compared to Europe.” (Appendix E)*

Another faculty with European architectural degree from another European country:

*“...I am from a European country originally, so educations in architecture there is less artistic and more engineering oriented. So there, we were actually doing similarly to double degree in architecture and civil engineering there... In the US, the students do structural analysis maybe once in their design, which is in the comprehensive studio. But in the Europe, the structural analysis is done in any project to ensure the project has to be buildable in a sense... They do not do hand calculation for the optimization” (Appendix G)*

Thus, based on the participants' responses, architectural students and *architects in Europe are more knowledgeable in structure than in U.S.* and the architects are the licensed engineers for the building design and construction. Regarding the knowledge of structural optimization, both U.S. and European architects commonly do not conduct the structural optimization task. Although the faculty (appendix E) mentioned that European architects are commonly more familiar to structural optimization compared to U.S. These patterns of responses related to the unfamiliarity of architects related to the structural optimization approach for architectural design process become more apparent as more participants were

involved in the interview process. Thus, when relating research concerns 1 and 2, although existing architectural structural optimization method such as iGKG is appropriate for architects as mentioned in chapter 4.7 (comparative study), the limitations on knowledge, education, expectation and design scope commonly prevent architects from exploring the optimization method, especially in the U.S.

So, what are the consequences for lacking the structural knowledge? *Architects and engineers often have different parameters of interest.* A faculty mentioned that while architects are more interested in the perception of the building, engineers are more interested in the structural quality. Faculty members expressed that *architects and engineers do not have the common language, and the gap of knowledge between architect and engineer seems very apparent in many contemporary collaborations.* The gap of knowledge between the two leads to many iterations in the exchange of information, and *iterative exchange of data between architecture and engineering is expensive* which potentially costs a lot of money if payment to the engineer has to be made per design review. The same kind of difficulty, faculty mentioned that *if fixed payment method in the form of contract is made up-front, engineers usually are more reluctant to change the design for efficiency,* because the fee for few design iterations and reviews are the same as multiple. This difficulty in the collaboration process is more often occurring in the case of smaller architectural firms. *Usually, only bigger architecture firms have more connections to engineers,* and thus engineers can show up early in the design process. In most cases, for smaller companies, architects have to rely more on their limited structural knowledge at the initial stage of design before passing their schematic design model to engineers.

One participant mentioned that ideally a good collaboration should be where both can find common ground, and that would require a structural engineer who understands how to design buildings and an architect who knows how to analyze the structure. The lack of performance value in the architectural design often times becomes the reason why *there is the lack of appreciation from engineers to architect's idea*. A faculty mentioned that *architects tend to focus only on the formal exploration and let the engineer solve the structural problem. The gap of knowledge, lack of common ground and lack of mutual appreciation between architects and engineers create problems in the collaboration between architects and engineers.*

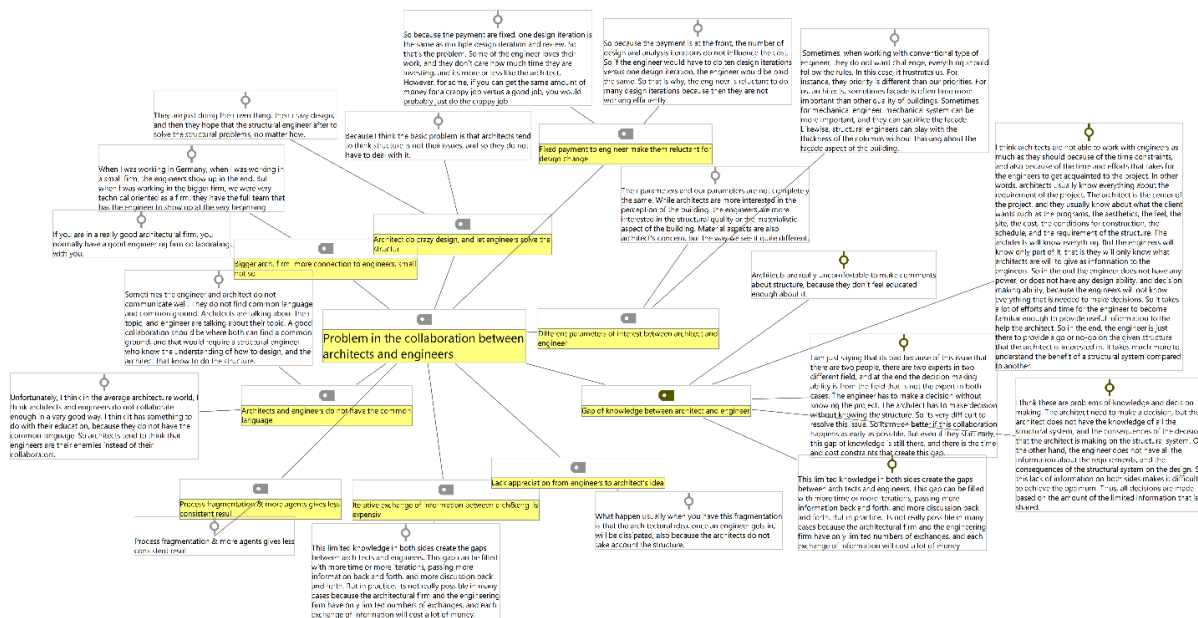


Figure 7.3. Code-subcode-segments model of the theme “problems in the collaboration between architects and engineers.”

Problems related to contemporary collaboration are expressed by participants, but is there a need to do structural optimization, to include performance value into an early

design? A structural optimization concept was introduced to the participants during the interview. A faculty expressed that *structural optimization can potentially help navigate towards a particular solution* and become a significant advantage to *ensure that the design is structurally sound*. Another participant described that *structural optimization approach in architecture could be useful as a rough form-finding process* instead of a refinement method. *There are many design factors in architecture than structure* such as building programs, aesthetics, sustainability aspect, manufacturability, physical condition of the site, social context, budget, schedule, acoustics, mechanical and electrical systems, and much more. As a result, in the design process, *many constraints are not included in the optimization formulation*. Thus, both faculty and students expressed that there are needs to do structural optimization, yet there are limitations of what the design optimization can do in the architectural design context.

There is a hope towards a better architects and engineers' collaboration. A faculty, with eight years of experiences in architectural design practice, expressed that *process fragmentation and multiple agents involved in a process often give the less consistent result*. He describes:

*“... In the design process, I think this fragmentation of the process between architects, engineers, and others are proven to me... What usually happens when you have this fragmentation is that the architectural idea, once an engineer gets in, will be dissipated, also because the architects do not take into account the structure... There would always be an engineer in the end, but, based on my experience, it's always easier if the architect has the capability to move as far as possible... So I think if architects move forward as much as possible, and the architectural product is structurally intelligent, then it would be a more productive collaboration.... Because it's better to work with a high-quality product than working with something that has no quality.”* (Appendix E)

Based on the responses, there are needs to do structural optimization, but what type of feedback are preferred from using this method? Most participants desire a structural optimization process that results *in variation of solutions* for further design tweaking, to give more context, to provide more engagement to the users on the product, and to have more options to adapt the design. However, there is one faculty that *is interested more in a single optimum* to simplify the process. For the criteria of the objective function, one of them are interested in *minimizing the cost and put the stress and displacement as constraints*. He mentioned that the optimization procedure *should become a process where the codes and performance crossover*. He described that this could be done by letting the displacement and stress in the constraints only to satisfy the code and optimize the cost of the structure for the performance. He mentioned that *optimizing the number of connectors in a space truss* could become an interesting example for minimizing the cost of a project.

As for the nature of the result itself, one faculty strongly desired for the *process that results in the design improvement rather than necessarily the optimum*. He mentioned that in the architectural design context, *it would make more sense if the optimization results are feasible low tolerance optima with less concern for the precision*. Some faculty members that are also practitioners desire *layers of optimization with various criteria*, such as the possibility to incorporate acoustic and energy performance into the optimization routine

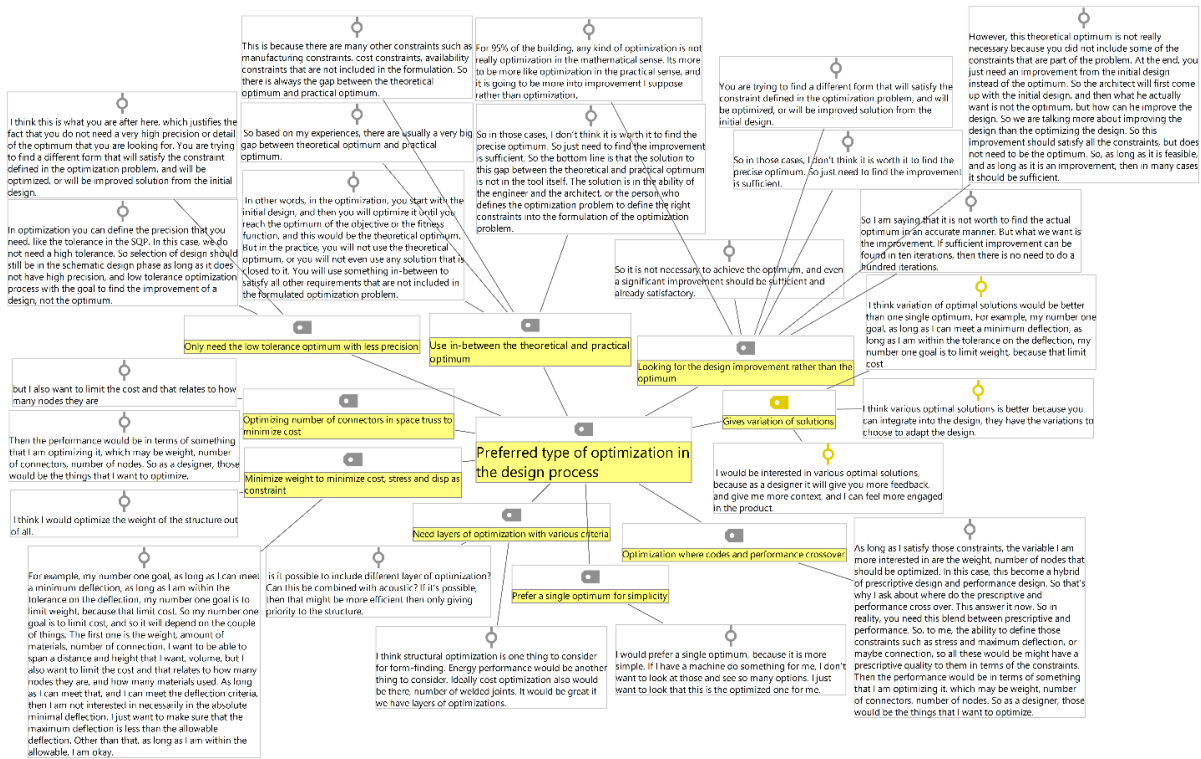


Figure 7.4. Code-subcode-segments model of the theme “preferred types of optimization in the design process.”

### 7.2.2 Theoretical Narrative Related to the Third Research Concern

Participants expressed the benefits and limitations of the demonstrated iMAGv1 after watching the demonstration video during the interview. Regarding the usability aspect, faculty members described that *iMAGv1* is *user-friendly enough*, as long as the users are familiar with using the Grasshopper. A faculty, with an engineering background, mentioned that incorporating *Abaqus into the tool give access to a powerful post-processor* and makes the whole process compatible with the engineering terminology and the engineering practice. Another faculty describes:

“... . I think it's better to have two disciplines coming in the tool with their own software. Because I think each discipline is optimizing their own software specifically... So I think different software from the engineer and architect



*probably will be developed further and further and they will be still tied into your developed system... I think the performance would be better if to keep two or more software from different disciplines. Because of its discipline-specific. Grasshopper is made for architects, and Abaqus is made for engineers.”*  
(Appendix G)

Related to the idea of postponing the engineer’s involvement, one participant mentioned that *the tool helps the architect make an upfront decision without the engineer presence* in which it acts as a knowledge-based system. Another faculty also indicated that the developed structural optimization tool helps the architect produce a more informed design before passing the model to engineers, and as a result, potentially postponing the involvement of engineers in the early schematic design phase. A faculty, who has nine years of experiences in architectural design practice and mostly dealt with competition projects, mentioned that usually when dealing with organic design, white spaces were indicated in an architectural model to signify undefined structural intervention. He expressed that *iMAGv1 allows architects to evaluate the performance of the organic structure*, may help him to at least check the feasibility of the design, and make the organic design more coherent and defined. Faculty members expressed that *iMAGv1 generally can improve architect-engineer collaboration* by enhancing the quality of the architect-engineer communication, and the quality of the product if used properly. Not only structural optimization tool such as *iMAGv1* ensure a more intelligent product to emerge in the early architectural design stage, but also the potential results in a more *innovative artifact that can surpass what the conventional prescriptive design code provide* in terms of the structural design system.

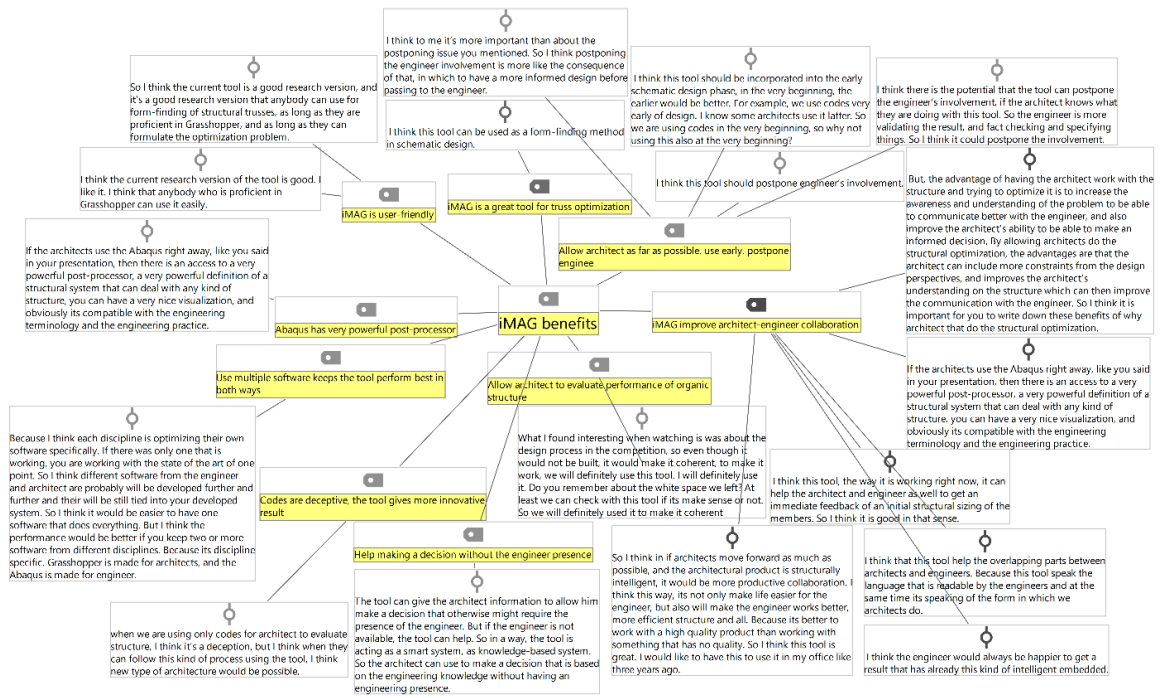


Figure 7.5. Code-subcode-segments model of the theme “iMAGv1 benefits.”

During the whole process, the responses were deemed to converge into a certain pattern of themes or theoretical construct with no further emerging themes and thus indicates that the whole process reaches theoretical saturation.

## CHAPTER EIGHT

### GUIDELINES FOR USING THE DEVELOPED METHODS

#### **8.1 Guideline for Introducing the Methods in the Academic Environment**

From the theoretical narratives of the first and second cycles of the Grounded Theory, participants saw the benefits of performing an early form-finding structural optimization in the design process such as using the developed tools. For educational purpose, this research suggests that the utilization of tools such as iMAGv1, iMAGv2, iMAGv3, iGFM, iGAG or the existing iGKG are incorporated into the structural analysis courses with the prerequisite of a Grasshopper parametric modeling course. Based on the participants' responses, this research partially encourages that U.S. architectural educational systems to incorporate more technical courses (calculus, structural analysis, programming, etc.) into their curriculum, like in Europe, to help the students understand more about the technicality in the structural analysis. However, the research also understands regarding the complexity related to the process of changing the existing programs and plans of study. As a response to this issue, the five developed tools were developed such that the students or architectural practitioners can use them as *black-boxes* without having to spend years for studying the deep theory in typical structural analysis and optimization courses. This research suggests that few lecture materials added to the typical existing structural analysis and computational design courses are considered sufficient to help the students understand how to use the developed tools properly. This section introduces the developed guideline that can be followed in the educational

environment to increase the effectivity for students to understand the implementation of the architectural, structural optimization method.

To maximize the effectivity of using the tools, this section particularly discusses briefly some suggestions to modify the plan of study for M. Arch in Clemson University. The recommendations can also be applied to any plan of study of the architectural program in the U.S that are interested in incorporating architectural, structural optimization as part of the curriculum.

Figure 8.1 shows the timeline of Computational Design Method course in Clemson University for the year 2016 (Lee, 2016). In week 4, students should have the necessary knowledge to develop their own parametric model with the proper understanding of the concept of design variables, design constraints, and design objectives in the context of parametric modeling. The existing Grasshopper component “Galapagos” can be introduced to students to run a geometrical optimization problem. The introduction of Galapagos can help students have a better intuitive understanding of how numerical optimization works.

COMPUTATIONAL DESIGN METHOD COURSE (ARCH 8120) (Clemson University)	
Week 1: Geometry and information: Points, Lines and Surfaces	
Week 2: No class	
Week 3: Computation and form-making: components, aggregation, data sets, ranges/limit	
Week 4: Computational form-finding: Mathematics, biology, physics and materiality in matter	
Week 5: Computation and relational information: Associating design + integration; structuring information	1. Introduce how to model optimization problem (geometric problem or structural if possible).
Week 6: Computation and interaction: Machines, sensors, human occupancy, and feedback	2. Introduce Galapagos to Students.
Week 7: Computation and organization: emergence, biomimicry, efficiency	3. Introduce Karamba if possible.
Week 8: Group presentation	
Week 9-16: research specific topics	

Figure 8.1. Grasshopper course at Clemson University (adapted from: Lee, 2016)

Figure 8.2 shows the timeline of Structure I course in Clemson University for 2016 (Barrios, 2016). In week 10, the students should have the necessary knowledge about the analysis of internal forces and stresses developed in truss members; and nodal displacements. Also, during that time, the students should already have the intuitive understanding for designing truss systems in terms of configuration. If students have taken the Grasshopper course with the suggested appended topic (labeled in orange in figure 8.1), then the students should have the necessary knowledge to formulate their optimization problem for form-finding geometrically, or possibly structurally for the structure I course. The architectural, structural optimization methods such as iMAGv1 can be introduced in the following week by the instructor (see the user manual (Appendix

A) and the corresponding interview modules transcript (Appendix B) to help the teaching process).

STRUCTURES I COURSE (ARCH 8700) (Clemson University)	
Week 1: Introduction	
Week 2: Load tracing	
Week 3: Free body diagrams	
Week 4: Forces and Reactions	
Week 5: Moments and form active	
Week 6: Bulk Active	
Week 7: Analysis of frames	
Week 8: Visiting lecturers	
Week 9: Vector active	
Week 10: Analysis of trusses	Introduce architectural, structural optimization software for truss
Week 11: Vertical structures	
Week 12: Lateral stability	
Week 13-15: Projects and Final exam	

Figure 8.2. Structures I course at Clemson University (adapted from: Barrios, 2016)

Figure 8.3 shows the current plan of study of M. Arch at Clemson University. Coupling the Grasshopper course with a structural course in one semester can potentially increase the effectivity in learning the application of architectural, structural optimization in the design process. Both the research period in the Computational Design Method course (ARCH 8120) (week 9 to 16) and Architecture Studio course (e.g. ARCH 8410) can potentially become perfect moments to explore the architectural, structural optimization method for the design process.

**CLEMSON UNIVERSITY SCHOOL OF ARCHITECTURE**  
**Master of Architecture Plan of Study 2015-16**

**THREE-YEAR (M. Arch. I) TRACK (90 CREDITS) begins here**

**FIRST YEAR: CORE STUDIOS**

**Semester 1 • Fall • Clemson Campus**

6	ARCH 8410	Architecture Studio I
3	ARCH 8100	Visualization I
3	ARCH 8600	Architectural History and Theory I
3	ARCH 8700	Structures I
15	credits	

**Semester 2 • Spring • Clemson Campus**

6	ARCH 8420	Architecture Studio II
3	ARCH 8110	Visualization II
3	ARCH 8610	Architectural History and Theory II
3	ARCH 8710	Structures II
15	credits	

Consider ARCH 8120

Figure 8.3. M. Arch plan of study (adapted from: Clemson University, 2014)

Figure 8.4 shows the timeline of the Structures II course in Clemson University for 2017 (Albright, 2017). In the week 10, the students should have the necessary knowledge about the analysis of the internal forces developed in frame members. The detailed analysis of deflection and stresses in frame members may be considered as quite challenging since it requires prerequisites such as years of calculus courses to be able to properly understand the theory and application of formula related to conservation of energy for the moment of inertia, flexure formula for bending stress, radius of curvature for Euler-Bernoulli Equation and so on. Thus, the understanding of beam theory is often conveyed intuitively to architecture students. However, the utilization of the developed architectural, structural optimization does not require the users to have the knowledge of those theories since all the detail calculations are done in the background by the developed system. It is assumed that during week 10, the students should already have the intuitive understanding for designing frame systems in terms of configuration. Using both the intuitive design knowledge of frames and understanding to use prescriptive code (e.g. the maximum elemental stress < the yield stress of the material used), the students

are considered ready to be introduced to the developed architectural, structural optimization methods such as iMAGv1. The architectural, structural optimization method can be introduced in week 11 by the instructor (see the user manual (Appendix A) and the corresponding interview module transcript (Appendix B) to help the teaching process).

STRUCTURES II COURSE (ARCH 8710) (Clemson University)	
Week 1-2 : Overview of structural system emphasizing spanning systems	
Week 3: Load tracing and statics review	
Week 4: Strength of materials	
Week 5-7: Beam bending and shear	
Week 8: Beam bending and shear with application to structural steel	
Week 9-10: Beam deflections and structural timber	
Week 11-12: Reinforced concrete structures and lateral forces	Introduce architectural, structural optimization software for frame
Week 13: Reinforced concrete and structural connections	
Week 14: Column design	
Week 15: Foundations	

Figure 8.4. Structures II course at Clemson University (adapted from: Albright, 2017)

For the implementation of architectural, structural optimization in the design process, see figure 8.6.

## 8.2 Guideline for Implementing the Methods in the Design Practice Environment

Referring back to figure 1.3, there are five architectural, structural optimization methods developed in this research that can be used during the schematic design phase.



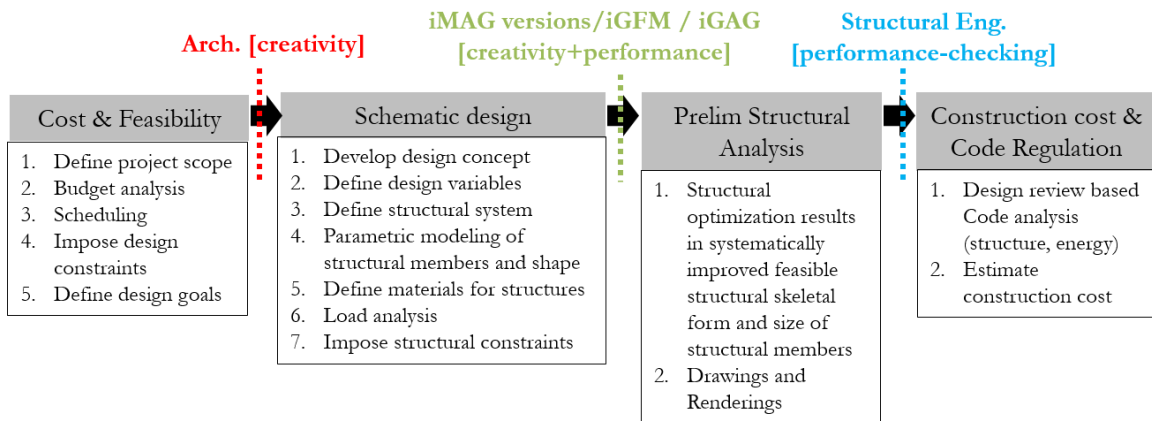


Figure 8.5. Proposed building design workflow using the developed methods

The design variables, design goals, design constraints and structural constraints are formulated by the architect. These formulations must be tied to the parametric modeling system (Grasshopper) defined by the architect. The architect can then use the developed structural optimization methods to conduct the form-finding process. The resulting form can be used as the point of departure in the collaboration between the architect and the structural engineer. Figure 8.6 shows the detail of the proposed design process when implementing the developed structural optimization methods. Note that both appendix A and appendix B should be used as references when following the proposed design process.

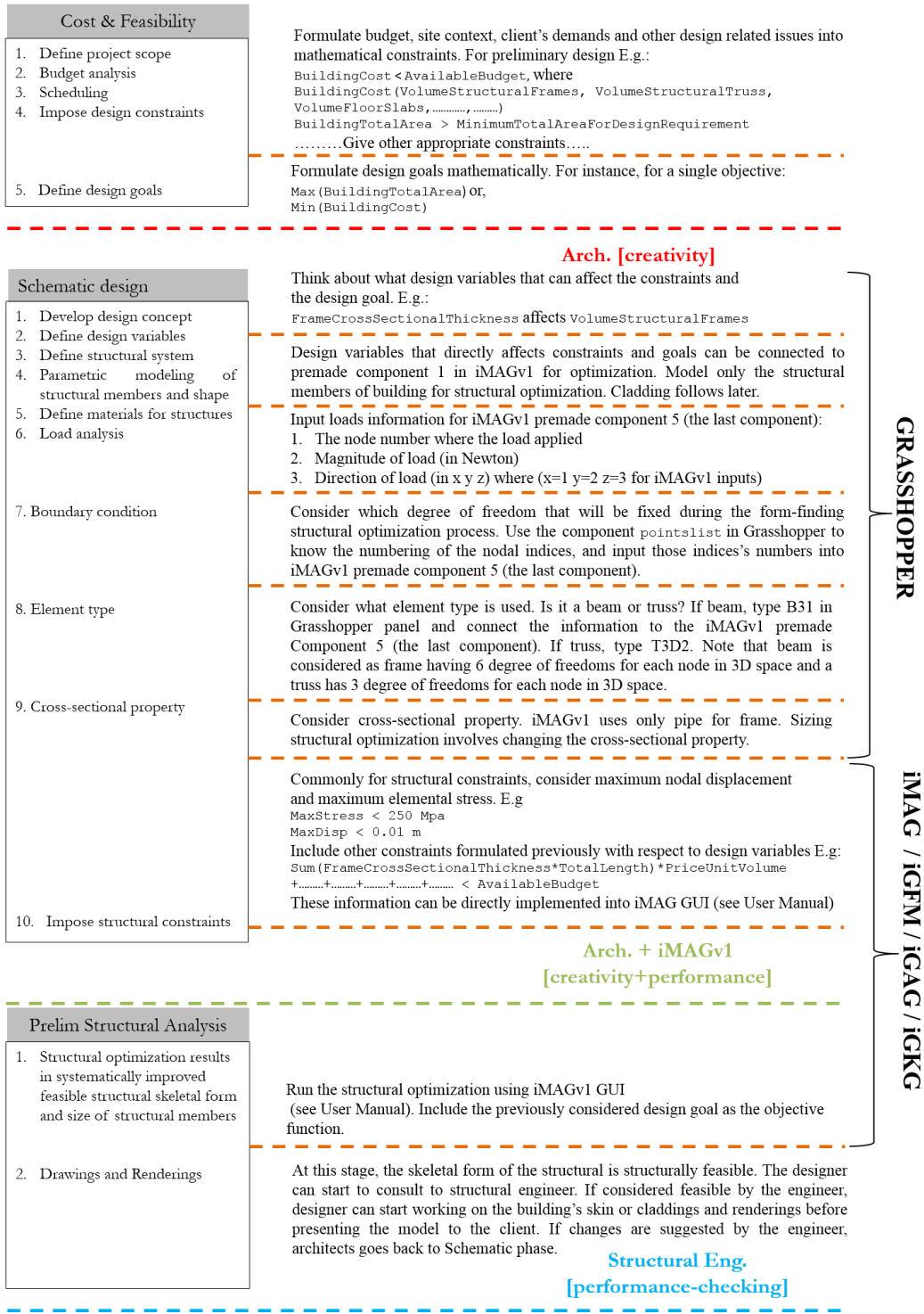


Figure 8.6. Design guideline for implementing architectural, structural optimization methods

Note that the example of the problem formulation in the figure 8.6 is much simplified to meet the purpose of making a simple one page design guideline. Particularly, the simplification is much imposed in the formulation of design constraints. Design constraints, in many cases, are much more difficult to be formulated compared to the structural constraints. Unlike structural constraints, the factors in the design constraints often cannot be reduced into parameters that can be quantified and can simulate the real situation. For instance, aesthetic is qualitative in nature and thus cannot be reduced to numbers. Building cost, for example, involves many factors such as manufacturing cost, material cost, shipping charges, taxes, labor wages, site conditions (soil conditions, conflicting utilities), duration of construction process and many more. Thus, it is often impossible to have a nearly perfect project cost estimation due to such numerous amount of factors in which each of those can be further broken down into other complex sub-factors. In the end, parameters and constraints influencing the estimation of building cost can only be included in the optimization model as extremely rough estimators. However, the motive is always to better consider as much as possible various aspects rather than not at all. The hope is that the inclusion of those parameters into the optimization model can increase the probability for architects to have a better design direction rather than falling down into the limitless design possibilities. A design direction in terms of methods in selecting design candidates are necessary in the architectural design process. Oftentimes, the development of concept in the schematic design phase leads the architect into a very vast design space. Without a well-defined

method for design selection, architects are often tempted to base their decisions purely by artistic intuition and thus the opted design candidates are often detached from the context of reality (cost, feasibility of structural performance, sustainability issues, etc.). As confirmed from the literature review (chapter 1) and conveyed by the research interview participants, a purely artistic design innovation by architects is often not much appreciated by engineers due to lack of performance aspects that relate to reality. The guideline provided in the figure 8.6 can be used by architects to potentially help them reduce the limitless design space into a smaller feasible space, at least structural-wise, by means of an automated form-finding process.

Perhaps the primary challenge when using the proposed design methodology (as illustrated in figure 8.6) is for architects to be willing to shift their schematic design style from the commonly intuitive (contemporary design process) into a more mechanical and scientific approach (proposed design process). The contemporary design process (see Figures 1.1 and 1.2) had been described by many researchers as ineffective and inefficient for realizing project success in terms of the bridging the communication during the architects-engineers collaboration (as discussed in chapter 1). As mentioned in the chapter 1, in the contemporary design process, the cutoff of architectural innovation often happens when structural constraints are introduced by the structural engineer into the design during the preliminary structural analysis. As presented in Chapter 7, some participants mentioned that a more productive collaboration is when the architect can move as far as possible to produce a more structurally intelligent design before the collaboration takes place. Thus, using the proposed design process, parts of the structural

engineer's tasks are shifted to the architect's, i.e., the consideration of materiality and structural performance are brought into the architectural schematic design phase.

Therefore, this research envisions that to better the collaboration process, the inclusion of structural considerations into the schematic design process is necessary. Hence, when architects decide to follow the design guideline (figure 8.6), the idea is to as much as possible impose the creativity into the systematic approach rather than vice-versa. In terms of technical issue, the creativity aspect is reflected by the capability of the architect to transfer as much as possible their design concepts into the construction of parametric model, while ensuring that the formulated design variables, design constraints, structural constraints, and design goals are tied into the parametric definition. Figure 8.7 shows the simplified graphical version of figure 8.6 regarding the collaboration process when implementing the form-finding structural optimization methods.

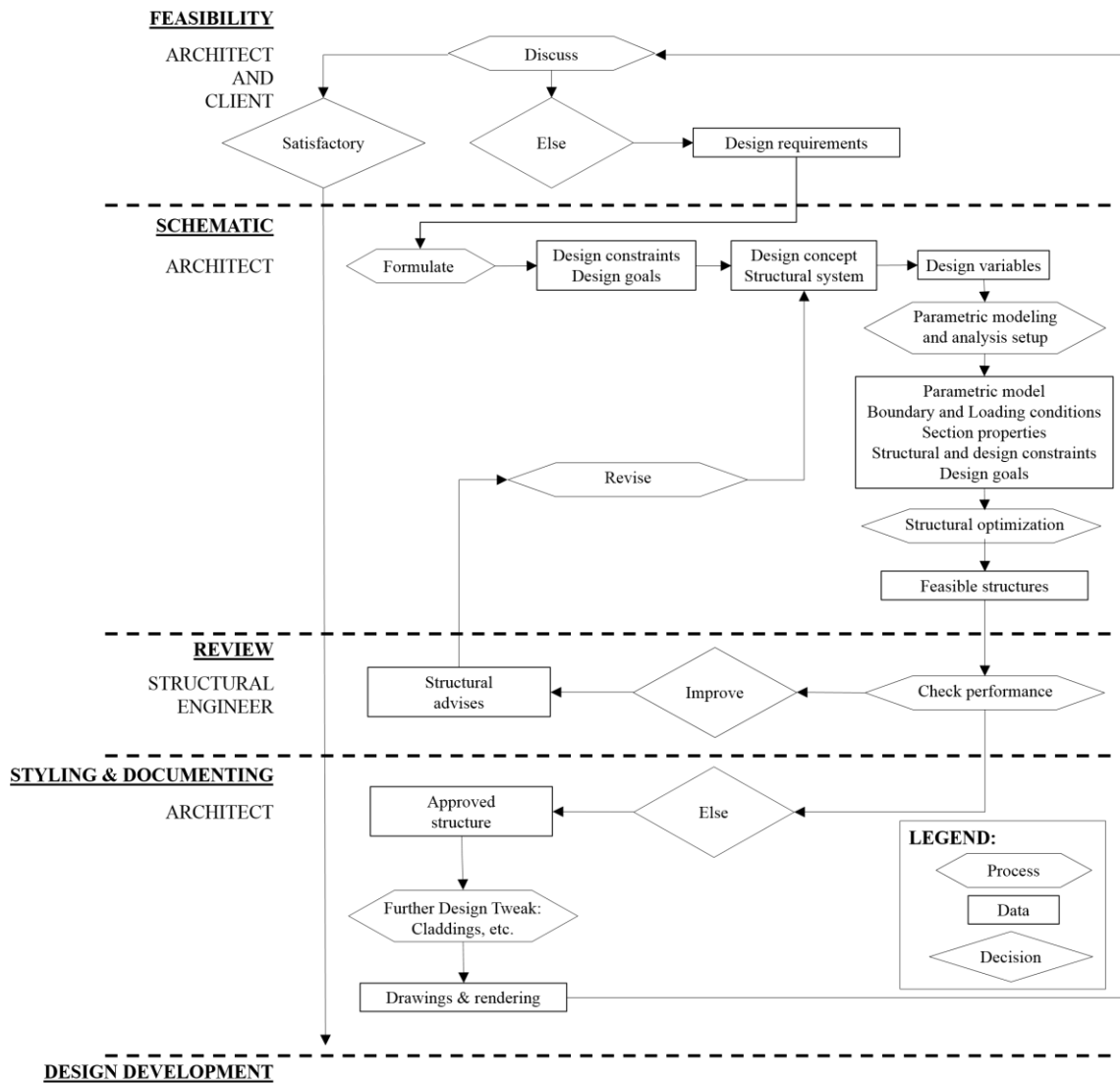


Figure 8.7. Flowchart of the proposed collaboration process when implementing form-finding structural optimization methods

Figure 8.7 shows that the design constraints, design goals, design variables, design concept and structural constraints are formulated by the architect. This information is used to generate the form via structural optimization procedure. It must be noted that the form-finding structural optimization here does not only consider the structural aspect, but it may include variables and constraints that are relevant to the design such as the aspect of

manufacturability, sustainability and so on. The right-hand side of the geometrical related constraints such as total area can be easily evaluated by the geometric modeling system such as Grasshopper. The incorporation of the sustainability aspect into the optimization model may require the inclusion of customized or commercial codes that are able to evaluate the necessary parameters (e.g. calculating daylight factor, thermal loads, etc.).

Regarding the collaboration with the structural engineer, the collaboration starts with the structurally optimized skeletal form consisting of roughly optimized size of structural members, as well as the shape and topology of the building. When either one of the three iMAG versions is used, the structural engineer can directly start post-processing the information using Abaqus (plotting and observing the structural behavior) without having to deal with the software interoperability issues and the necessary further engineering analysis can be conducted (e.g. dynamic analysis, etc.), if necessary. If either iGFM or iGAG is used, the post-processing can be conducted by the structural engineer in the Matlab's environment. The input files generated by Grasshopper for both iGFM and iGAG can also be used to run Abaqus, if preferable.

From the literature review (chapter 1) and research participants' feedbacks, the research predicts that structural engineers should be able to communicate better to architects when the design workflow (figure 8.7) is implemented. Some of the suggestions from the structural engineer to the architect may include modifying or adding some design variables that may have better influence on the structural constraints. Another example of suggestions would be to modify the structural configurations and connectivity to better the performance and innovative aspect of the structure. These

changes are relatively easy to be done in the Grasshopper by architects since tweaking the design can be made by simply removing and adding components and the modification can be made easily in any section of the parametric definition with few adjustments. The architectural, structural optimization can then be processed again by the architect and the iterative process of the collaboration continues until finding the solution agreed by both sides. Cladding and renderings can be done using the solution and the results can then be presented to the client.

Based on the research participants' responses, various improved design candidates are preferable. When using iGAG, architects and engineers can have the back-up feasible improved design permutations that can be used as design alternatives. When the client is unsatisfied with a particular design permutation, those other design alternatives can be presented as other options for the clients to consider.



## CHAPTER NINE

### CONCLUSION

In this research, five methods were developed for conducting black-box structural optimization as a form-finding technique in the architectural schematic design phase. These methods are compared to each other and to hand-calculations for the validity study. The five methods are also compared with the existing architectural structural optimization method such as using Karamba and Galapagos within the Rhino/Grasshopper interface. Several parametric model example problems such as double curved space trusses; double curved frames; polyhedral space frames and trusses; non-prismatic beam; planar frames; and planar trusses were developed for the validity study. A particular case of a twisted skyscraper design under wind and dead loads was parametrically modeled to demonstrate the capability of the presented structural method, such as iGAG, for dealing with a complex mixed nonlinear integer structural optimization problem. iMAGv1 was used in the demonstration module during the interview process. The qualitative data analysis uses Grounded Theory for the data collection process, and use MAXQDA to help with the coding procedure. The qualitative analysis results were written as theoretical narratives that express the interview feedbacks of architecture students and faculty members towards incorporating structural optimization in the architectural schematic design phase. Finally, guidelines concerning the implementation of the developed structural optimization methods in both the academic and design practice environment are developed and proposed.

The incorporation of structural optimization as a form-finding technique opens a new way of thinking regarding the architectural design process. Architects can implement the methods to automate the generation of organic lightweight structural forms based on both performance and creativity together in the schematic design phase, without the engineer's presence. The conventional architectural design approach that focuses solely on the artistry of formal exploration causes the gap of knowledge, lack of common ground, and lack of mutual appreciation in the collaboration between architects and engineers. In many cases, sacrifice has to be made by architects between creativity and structural performance during the schematic design phase. Structural optimization for form-finding allows architects to move beyond the prescriptive code, and to produce a more innovative artifact that is structurally intelligent. Such product not only improves the collaboration (or namely the soft dimension of the project implementation) between the architects and engineers during the later design phase but to also create a more aesthetically innovative product for the client. A good degree of collaboration improves the time management, cost-effectivity, and eventually, resulting in a project success.

### **Contributions and Discussion**

The major elements in this research were the development of the five black-box structural optimization tools for the architectural form-finding process, validity study of the developed tools, a comparative study between developed and existing methods, and the qualitative research process for validating the usefulness of incorporating structural optimization into architectural design workflow.

The five developed tools include iMAGv1, iMAGv2, iMAGv3, iGAG, and iGFM. The idea behind the development of these tool is to couple architectural software with engineering software. The coupling bridges the communication between architects and engineers; provides richer FEA outputs for further engineering analysis (customized FEA and Abaqus); can handle the analysis of large deformation; can handle wider classes of engineering problems; allows parallel computing; gives more access to various optimization algorithms (Matlab); and allows more versatility in configuring the optimization setup (Matlab). The idea is also to use Grasshopper in each of those methods as the generator of the geometry and the configurator of the structural analysis setup. The advantage of using Grasshopper is due to its user-friendly interactive visual programming interface for conducting the parametric modeling tasks. The premade components were made to manage interoperability between Matlab and Grasshopper and to allow parametric control over the structural analysis configuration during the optimization process. The parametric control facilitates programmable structural analysis setup so that the loading conditions, for instance, can be automatically updated at each iteration of the optimization, as demonstrated in the dead and wind load assignment in the twisted skyscraper case (chapter 4). These premade components can be conveniently connected to other Grasshopper components in a node based editor environment. It is considered in this research that an equally complex problem such as the skyscraper case (chapter 4) can only be done in other software by means of text-based programming which is deemed to be unsuitable for architects in terms of the usability aspect.

- IMAGv1 integrates Grasshopper, Abaqus, and Matlab, and uses exhaustive search for performing the structural optimization routine. The advantage of using this method is due to its ease of the programming, and thus this method was the earliest to be developed and used for carrying out the interview. However, this method performs poorly regarding convergence because the method is primarily based on the interpolation method used to create the smooth function for running the interior point method. Also, having to generate an adequate number of data points for the full factorial analysis is computationally expensive, but is required to construct a continuous function to better approximate the optimum.
- IMAGv2 uses the same workflow as iMAGv1, except that it uses the discretization and filtering process to search for the optimum right after the full factorial analysis. Although iMAGv2 takes longer time than iMAGv1, there was a suspicion that this process could be more robust when searching for the optimum. Through a comparative study, iMAGv1 turned out to perform better in terms of computational cost and convergence compared to iMAGv2.
- iMAGv3 integrates Grasshopper, Abaqus, and Matlab, and carry out cyclic interoperability between software and interior-point algorithm for conducting the structural optimization procedure. This method is much more efficient compared to the previous versions due to the direct communication between software. When coupled with the interior-point algorithm, the structural optimization requires much fewer function calculations compared to the other two versions of iMAG. Also, this method gives better convergence in the optimization process. Another

advantage of using this method is that Abaqus is incorporated into the workflow, and thus allows the method to have complete features for post-processing the optimized result, provide a better link to the structural engineering workflow, and is capable of dealing with other classes of problems including dynamic, contact, friction and thermal analysis problems.

- iGFM integrates Grasshopper and Matlab and uses customized FEA code and Matlab interior-point for performing the structural optimization process. This method has a faster structural optimization procedure compared to all iMAG versions because it carries out both structural analysis and optimization on a single platform. This method uses a deterministic algorithm, interior-point, to locate the single best solution. One of the interviewees mentioned that a single optimum is preferable compared to various optimal solutions. In this case, iGFM is a suitable method for them. Also the inclusion of Matlab into the workflow allows access to various available powerful optimization algorithms such as Sequential Quadratic Programming (SQP), Trust-Region-Reflective algorithm, Simplex search method, Quasi-Newton algorithm and many more (Mathwork, 2017). Note that altering the method of optimization in Matlab can be done very easily within few seconds, but it is not explored in this research because the interior point algorithm was considered sufficient for dealing with the presented problems in this thesis.
- iGAG integrates Grasshopper and Matlab, and uses customized FEA code and customized Genetic Algorithm for carrying out the structural optimization

procedure. Perhaps one of the biggest advantages of using this method is that it can deal with the mixed nonlinear integer programming problem, and thus the method can tackle complex problems such as topological optimization. Another unique feature is that this method results in various improved candidates instead of a single optimum in which, based on the interview results, such nature of the results is preferable in the architectural schematic design context. For a multimodal problem, Genetic Algorithm also has higher potential to escape from local optima. Another benefit of using this workflow is that the incorporation of customized FEA code allows a much faster structural optimization procedure because in each iteration Matlab only needs to interact with Grasshopper. The developed case of the structural optimization of a twisted skyscraper design demonstrates iGAG's features for dealing with a complex architectural-engineering problem.

- Validity Study of the Developed Tool: The validity study was to ensure that the developed five methods gave the proper results corresponding to given problems. The process was not only to validate the result from using a specific software or algorithm but also to check if the interoperability between software works correctly, especially as the structural analysis configuration was done in Grasshopper. Hand calculations using the direct stiffness and energy methods were used to validate the FEA results such as those from Abaqus, and the customized FEA code programmed in Matlab. The KKT approach was used to validate the optimization results generated in Matlab. All of the five methods

were validated and were proven to be satisfyingly accurate regarding both structural analysis and optimization capability.

- Comparative Study Between Developed and Existing Methods: Perhaps currently the most popular available architectural structural optimization is Grasshopper coupled with Karamba and Galapagos (iGKG). This method is very efficient because all structural optimization tasks are integrated into a single platform. Regarding operating procedure, the methods are very user-friendly because the designer can deal with the structural optimization task using a single Grasshopper interface. Through comparative studying chapter 4, the existence of iGKG examine the research concern 2 in which the present architectural structural optimization tool is appropriate for architects, but is not widely known. Few benefits of using any iMAG versions compared to iGKG are that all iMAG versions have more powerful post-processing features, analysis features and use the common language as engineers do. Some benefits of using all the developed methods compared to iGKG are that they have more versatility regarding configuring both the analysis and optimization setup including the capability for calculating elemental von Mises Stresses for frames and trusses and outputting the stiffness matrices for further engineering analysis. iGAG also facilitates more versatility in adjusting the optimization features including methods of cross-over, elitism rate, termination criteria, and mutation rate. All iMAG versions, iGFM, and iGAG allows for parallel computing to be performed for a highly complex problem.

- Qualitative Research: The qualitative research process examines the humanity aspect of the utilization of structural optimization in the architectural design phase. There have been emerging researchers in architecture that strive to implement structural optimization workflow into the architectural design process. However, this research is apparently the first that attempts to proceed further from the technicality of software development into the usability aspect in the context of architectural structural optimization. The reason for moving beyond a mere software development is that the implementation of structural optimization for architectural design (such as iGKG) is commonly limited to only develop within academic fields, not in practice. Six interviewed architecture faculty members in Clemson University mentioned that, based on their architectural design practice experiences, structural optimization are not common in architectural design practice. Thus, a question: “why is structural optimization not commonly considered by architects in the schematic design phase?” was raised.

From the qualitative data analysis using Grounded Theory, the responses show that architects are usually uninformed about both analytical and numerical structural optimization routine. The interviewees expressed that in practice, architectural practitioners usually never analytically evaluate structures, and most architects do not have profound knowledge about structures. Many architects, in practice, deal mostly with the conventional structure in which prescriptive codes are available. Thus, the form-finding process is not driven by the performance but only based on the project requirement and the available



building code. In this case, design innovation is suppressed by the lack of knowledge.

Commonly a big architectural firm has a good connection to engineers or has engineers work in-house. Thus, a more developed collaboration workflow is common for a big architectural firm compared to a small firm. This research targets small architectural firms as the users of the five developed structural optimization tools. As acknowledged by the participants, small architectural firms often do not have good connections to engineers. In the case of architectural competition, the small firms will struggle to find engineers, and once the collaboration starts, typically only non-structural shell models are sent to engineers. By using such model as a point of departure, problems such as lack of common ground and lack of mutual appreciation would likely occur during the collaboration. In such situation, innovation, time and cost will be compromised, leading to an unsatisfactory product and collaboration process.

From the interview responses, the core information that can be pulled out is that there are needs for structural optimization in architecture, and there is a hope towards a better architect-engineer collaboration. Structural optimization methods are needed for architects to generate an innovative design that is structurally intelligent, and to use the structurally intelligent model as the point of departure in the collaboration process.

The final section of the interview was the demonstration module, showing how iMAGv1 operated. The participants expressed that the tool can potentially

help architects make upfront decisions, to create a structurally intelligent and innovative product, and to improve the architect-engineer collaboration. Some interview responses are related to the preferable types of optimization models and results. Some preferred optimization models include minimizing construction cost, minimizing the number of connectors, constraining the displacement and the stresses. In terms of the results, participants are interested in design improvements rather than the optimum, various improved candidates rather than a single optimum, low tolerance results rather than precise and layers of optimization with various criteria. iGAG was developed to respond to these demands. The genetic algorithm can deal with all those types of optimization in which deterministic optimization algorithm such as interior point fails to satisfy some of those.

- Guidelines: Finally, the guidelines concerning the implementation of the developed architectural structural optimization methods for both the academic and design practice environment were developed. The contemporary collaboration between architects and structural engineers are found to be problematic based on the interviews and the literature review. This research suggests that changes in the contemporary design process would be beneficial. The changes have to be made starting with how the young generation of architects are taught in the academic environment. This research believes that the mentality of seeing *the design form and the structure as an inseparable entity* must be encouraged. The guidelines can be used to modify architectural educational curriculum and considered as a step-

by-step process to introduce young architects who are interested in free-form finding to the design methodology that uses architectural structural optimization. As for the architectural design practitioners, the guideline in figure 8.6 and the user manual can be used as the shortcut to briefly understand how to incorporate the developed tools into the design process.

### **Future Work**

A number of topics were identified as interesting subjects for future study. They are listed as follows:

- Shell optimization: Currently, the research only deals with the class of optimization that involves size, shape and topology optimization of space trusses and frames. However, many sophisticated structural systems such as in skyscraper design incorporate elements like shear walls for the core. Also, although the dead loads of floor slabs were accounted in the discussed twisted skyscraper, the stiffness of the floor slabs was not incorporated, and it was only taken as the contributing aspect to the factor of safety. For a more accurate structural optimization results, the utilization of shell elements is needed. This will require the stiffness of the shell elements to be constructed and included in the global stiffness matrix of the structural system.
- Dynamic Loads: Currently, the methods successfully demonstrate its capability to handle a nearly realistic simulation of a structure under wind loads. The skyscraper case in chapter 4 shows the form-finding of a structure driven by both the dead and wind loads that changes accordingly at each iteration based on the

changes in the geometry of the structure. Assuming the exposed surface area of the facade remains the same, the wind force increases as height increases.

However, for the case of seismic loads, the base shear value at the bottom is larger, and it decreases as the height increases due to a reduction in cumulative weight of the structure. Also, unlike the wind loads, in seismic load, the stiffness of the structure affects the seismic force that is developed, and the smaller the mass of the structure, the better the structure will behave during a seismic event. Thus, incorporating both wind and seismic loads in the structural optimization problem can potentially give drastic changes to the current improved or optimized geometry of the skyscraper design due to the different and some opposing nature of design requirement when applying both loads.

- Multidisciplinary optimization: The current versions of the methods are able to employ parameters related to both structural performance (e.g. nodal displacements and elemental stresses) and geometrical configuration (e.g. total area and weight of the structure) into the optimization model. However, a more realistic design issue also includes other factors into the optimization model such as the aspect of sustainability. The reduction in non-renewable or finite energy resources (coal, natural gas, etc.) requires design to make more use of renewable resources such as wind and solar energy sources. Energy and daylighting optimization can be done to maximize energy savings (e.g. maximizing total lighting requirement that can be replaced by daylight) while ensuring good daylight penetration with respect to the thermal comfort. For instance, by

integrating Matlab to a building energy simulation programs (e.g. EnergyPlus, Radiance, gbXML) or using Grasshopper's Ladybug and Honeybee, new constraints and objective functions related to building sustainability can be formulated. This process can add a new layer of optimization to the developed existing structural optimization methods as suggested by some participants during the interview regarding multilayer optimization. Incorporating this new sustainability layer of optimization will require not only the building's skeletons to be parametrically modeled but also the skin and shading devices that add a new list of design variables to be controlled during the optimization.

- Include more design variables for the skyscraper case: Despite the already present complexity, the current research only uses four design variables. The presented skyscraper problem is considered as a mixed non-linear integer programming of a black-box structural shape optimization problem. Although the thickness of the cross-section of the structural members is parametrically modeled as the function of the number of floors, the upper and lower bounds of the thickness are fixed during the optimization. The inclusion of variables such as the type of bracing system and upper and lower bounds of the thickness of structural members into the optimization model can further push the problem into a more realistic simulation. In terms of programming, including those variables are trivial and can be done in a few seconds using the presented method. However, the computational time is very expensive when such considerations are incorporated

which essentially upgrade the problem to a size, shape, and topological optimization problem.

- Modification of Target Population: Current target population used for the interview are students and faculty at Clemson University. Most interviewed faculty members have more than five years of experience in architectural practice. A future study may expand the target population to architectural practitioners whose know-how primarily within the intersection field of computational form-finding and structural analysis in architecture (e.g. parametric modeling for structural engineering; parametric modeling for architecture and manufacture; kinetic architecture; finite element analysis; shape grammar; and generative art). The responses from this target population can further increase the research's insight towards the implementation of parametric modeling software in the contemporary architectural design workflow, and the identification of the potential improvement of the developed structural optimization methods to fit into the contemporary design workflow.
- Mixed methods research: The emerging themes (figure 7.1) and repeating ideas (figure 7.2, 7.3, 7.4, 7.5) from the qualitative study can be used to construct meaningful hypotheses. The repeating ideas and themes cannot be used for generalization due to a very small sample size. A future study can be conducted to gather more samples with the same target population and carry out statistical hypothesis testing procedure for generalizing the findings. For instance, a hypothesis "if structural optimization is incorporated into the architectural

schematic design process, then it will decrease the design-review iterations between architects and structural engineers” can be tested. Another hypothesis would be “if structural optimization is incorporated into the architectural schematic design process, then various feasible and improved design candidates are more likely to be preferable during the design practice rather than a single optimum”. For example, using two sampling frames for South Carolina, there are 28 architectural firms recorded in Greenville (SCIWAY.net, 2017) and 14 around Clemson, Pendleton, and Anderson (YP LLC, 2017), giving a population of 42 firms. Using the formula for calculating estimating a sample size and correction for finite population (for formula, see Creative Research Systems, 2016), 21 samples need to be taken from the population. This assumes for 95% confidence level and 15 % confidence interval. This indicates 95% confidence that if for instance, 80% of samples give a certain answer, then the same answer can be acquired from 65% to 95% of the population. This finding can then be generalized within the margin of error, but only among the total of 42 recorded firms around South Carolina. A much broader scope of sampling frame around U.S. needs to be taken for a wider coverage of generalization of the finding.

## APPENDICES



## APPENDIX A: iMAGv1 User Manual for Interview

### iMAGv1

iMAGv1 was the first structural optimization method developed in this research and was the one used for the interview. iMAGv1 is the integration of Matlab, Abaqus, and Rhino/Grasshopper. iMAGv1 is used to conduct parametric structural analysis and optimization of any truss design. iMAGv1 is specifically developed for performing schematic architectural design as a form-finding method using structural performance to drives changes in geometry.



### Matlab

Matlab is a high-performance language that integrates computation, visualization and programming mostly used to deal with various engineering and scientific problems (Mathworks, n.d.).



### Abaqus

Abaqus (Simulia, n.d.) is a suite of robust engineering simulation programs based on the finite element method, developed by Dassault systems as part of their SIMULIA PLM software tools.



### Rhino/Grasshopper

Rhinoceros (Rhino, n.d.) is a CAD application focused on producing freeform surfaces. Grasshopper (Grasshopper, n.d.) is a visual programming language and is a plugin for Rhino for conducting parametric modeling.

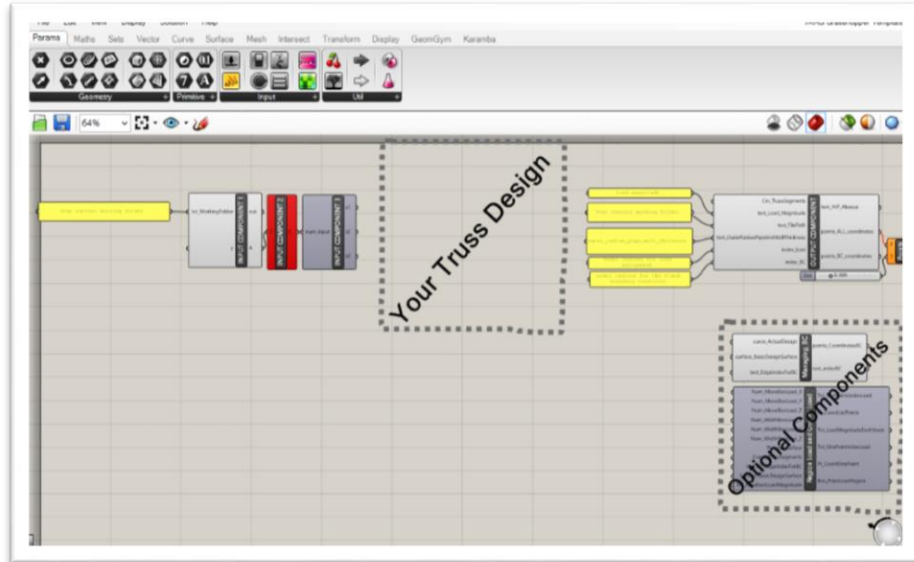


### Preparation:

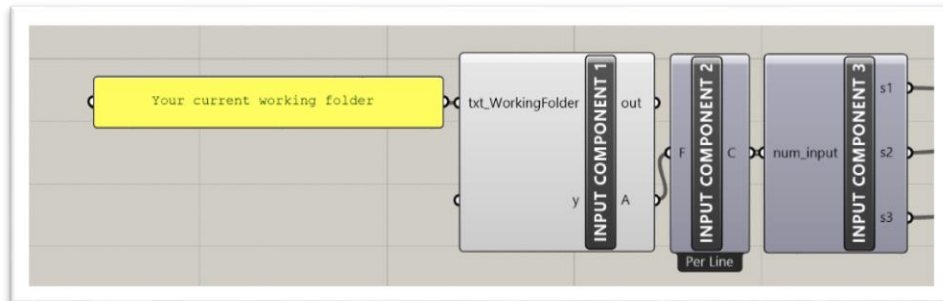
1. Install/Download Abaqus, Rhino/Grasshopper, Matlab to your PC.
2. Make sure you have Karamba plug-in in your Grasshopper.
3. Create a new working folder (This folder will contain any files related to your project).

Grasshopper (Get input from iMAG GUI to Grasshopper):

1. Create your own Grasshopper parametric truss model.
2. Save the Grasshopper file in your working folder.
3. Choose 3 variables to be your design variables for the structural optimization.
4. Open the given iMAG Grasshopper template as shown below and copy the whole components into your Grasshopper definition. Your parametric truss design should be located as shown below.



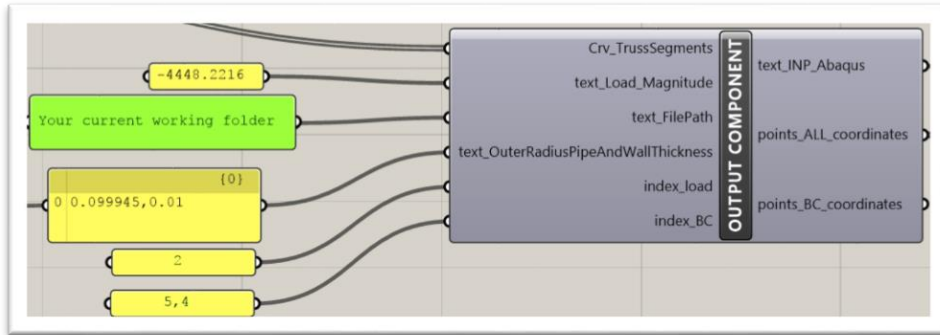
5. Put premade INPUT COMPONENT1, INPUT COMPONENT2 and INPUT COMPONENT3 below at the beginning of your Grasshopper configuration.



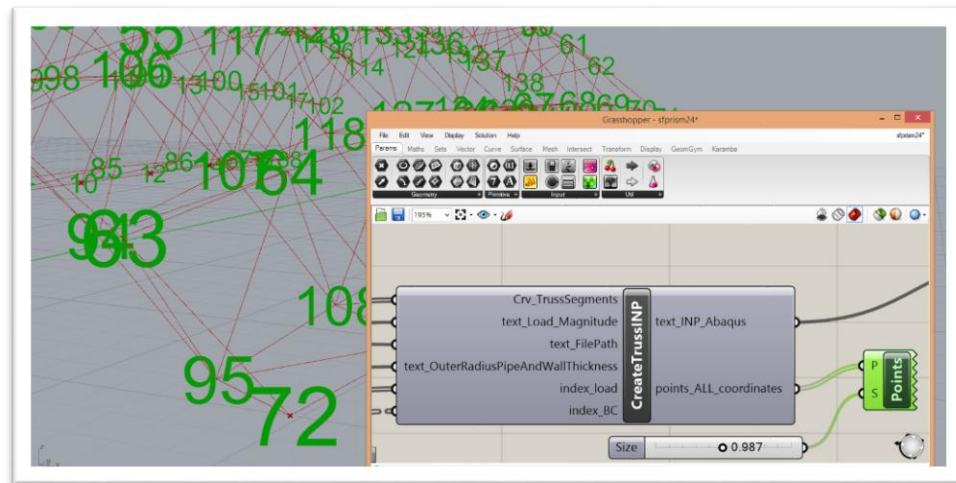
6. Write the directory of your working folder manually in a panel component as shown above and link the panel as input for the INPUT COMPONENT1:txt\_WorkingFolder as shown above.
7. Link the three outputs of the INPUT COMPONENT3 into 3 of your design variables.

Grasshopper (Send geometrical and structural analysis setup from Grasshopper to iMAG):

1. Put premade OUTPUT COMPONENT below at the end of your Grasshopper configuration.



2. Link your truss structure consisting of curves or lines into OUTPUT COMPONENT: Crv\_TrussSegments.
3. Write the directory of your working folder manually in a panel component as shown above and link the panel as input for the OUTPUT COMPONENT:text\_FilePath as shown above.
4. Connect the output of OUTPUT COMPONENT:points\_ALL\_coordinates into points list component to see to nodal indices of your structure as shown below.



5. Type the indices of the nodes where you want to assign the load to a panel component and link the panel to OUTPUT COMPONENT:index\_load.
6. Type the magnitude of the load applied to the previously selected nodal indices into a panel component and linked the panel to OUTPUT COMPONENT:text\_Load\_Magnitude.
7. Type the indices of the nodes where you want to assign the fixed boundary conditions into a panel component and link the panel to OUTPUT COMPONENT:index\_BC.
8. Type the outer radius and a wall thickness of the cross section of the steel pipe used for the truss into a panel component and link the panel to OUTPUT COMPONENT:text\_OuterRadiusPipeAndWallThickness.

Prepare the Files in the Working Folder:

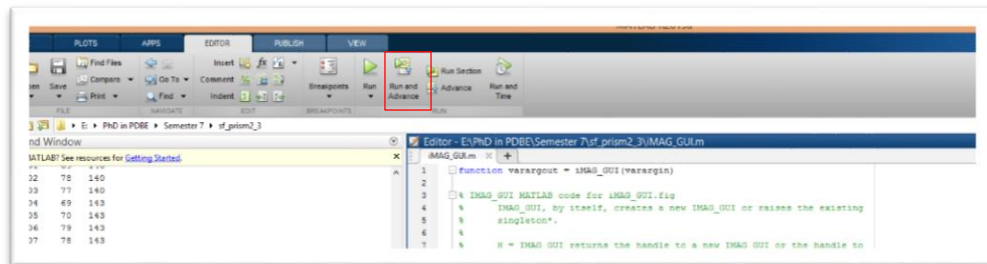
1. Copy and paste all the iMAG files into your working folder as shown below.

Name	Date modified	Type	Size
checkLCK.m	3/7/2015 8:04 PM	M File	1 KB
ComputeWeight01.m	5/3/2016 12:25 AM	M File	3 KB
confun_cosine_interpolation.m	9/24/2016 4:39 PM	M File	2 KB
cosine_interpolation.m	9/24/2016 6:32 PM	M File	1 KB
func_main01_cosine_nix.m	9/24/2016 12:02 AM	M File	2 KB
func_TessCurve_ExhaustiveOPT_Feb20...	9/23/2016 6:59 PM	M File	3 KB
iMAG Grasshopper Template.gh	9/25/2016 11:06 PM	Grasshopper Docu...	91 KB
iMAG_GUI.fig	9/26/2016 12:08 AM	3dsMax biped figu...	44 KB
iMAG_GUI.m	9/26/2016 12:08 AM	M File	34 KB
interpolate3DCosine01.m	3/16/2016 11:22 PM	M File	7 KB
objfun_cosine_interpolation.m	9/24/2016 6:29 PM	M File	2 KB
readoutput.m	10/22/2015 4:04 PM	M File	3 KB
SFDOUBLECurveTess_analysis_3_OBJF...	9/22/2016 6:38 PM	M File	2 KB

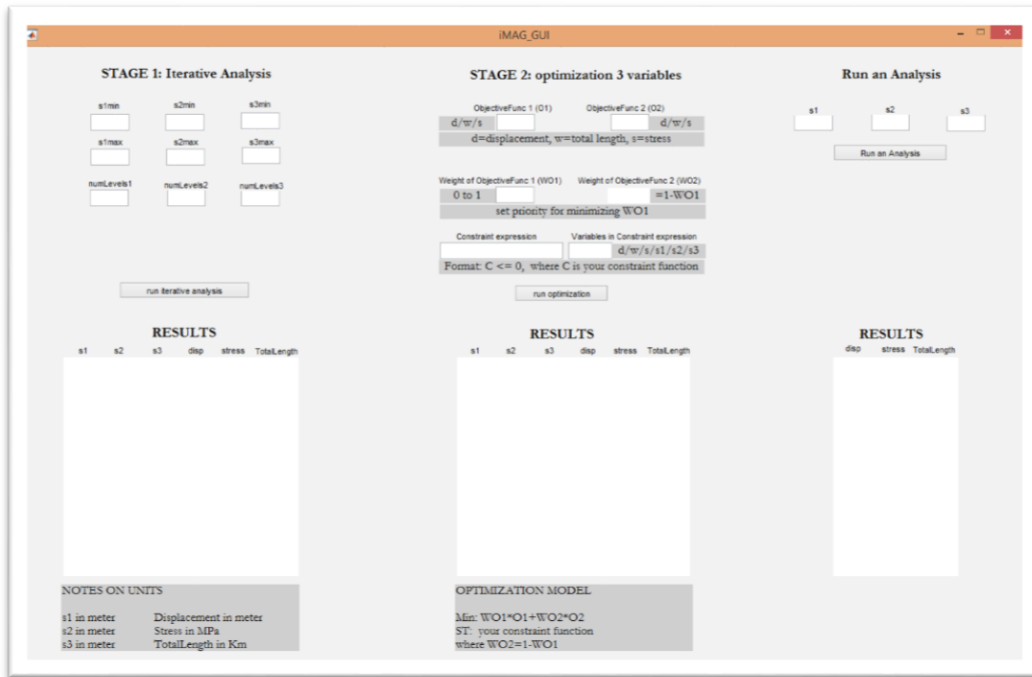
Your working folder should contain all these files

2. The file that is highlighted above is your Grasshopper file saved into the same working folder with the iMAG files  
iMAG Interface (Run Iterative Analysis):

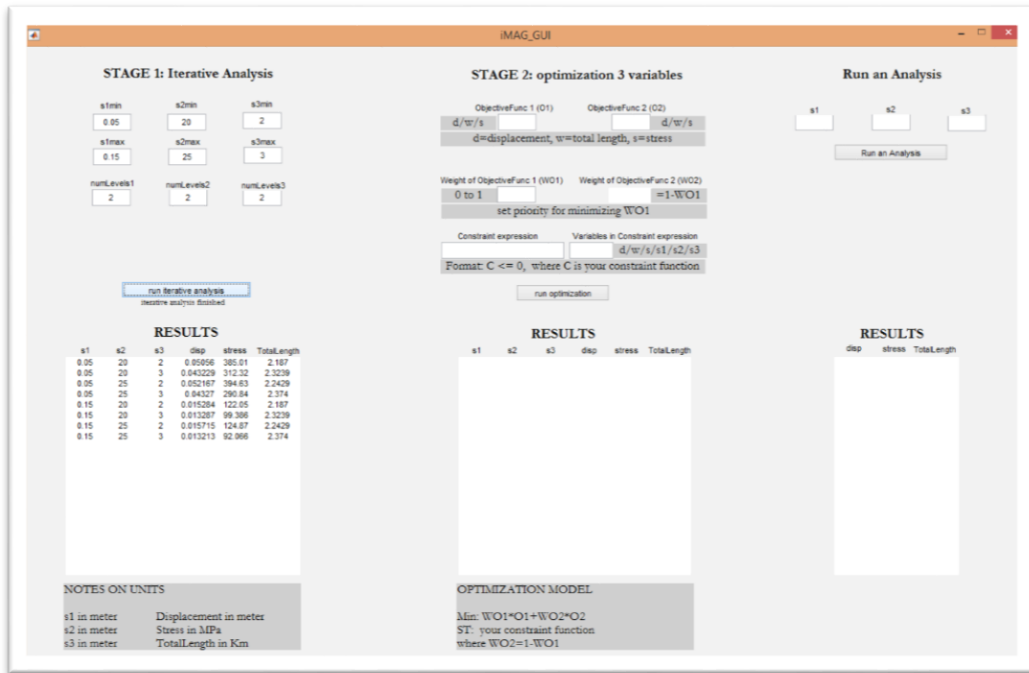
1. Open the iMAG\_GUI.m by double clicking the file or open it in Matlab.
2. Click Run button as shown below.



3. Enter numerical values into s1min, s2min, and s3min as shown below to define the minimum values that your design variables s1, s2, and s3 could take when running iterative structural analysis.

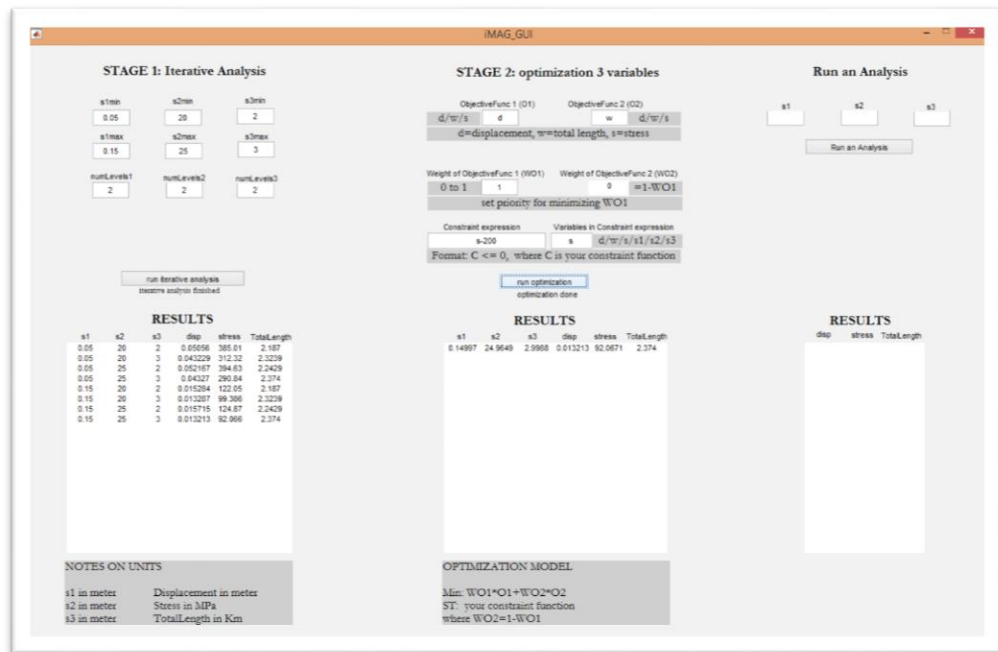


4. Enter numerical values into  $s1_{max}$ ,  $s2_{max}$  and  $s3_{max}$  also to define the maximum values that your design variables  $s1$ ,  $s2$ , and  $s3$  could take when running iterative structural analysis.
5. Enter numerical values into  $numLevels1$ ,  $numLevels2$  and  $numLevels3$  also to define how many levels each of your design variables  $s1$ ,  $s2$  and  $s3$  would take to step from the minimum to maximum values.
6. Click the button: run the iterative analysis.
7. Wait until the result appears as shown below



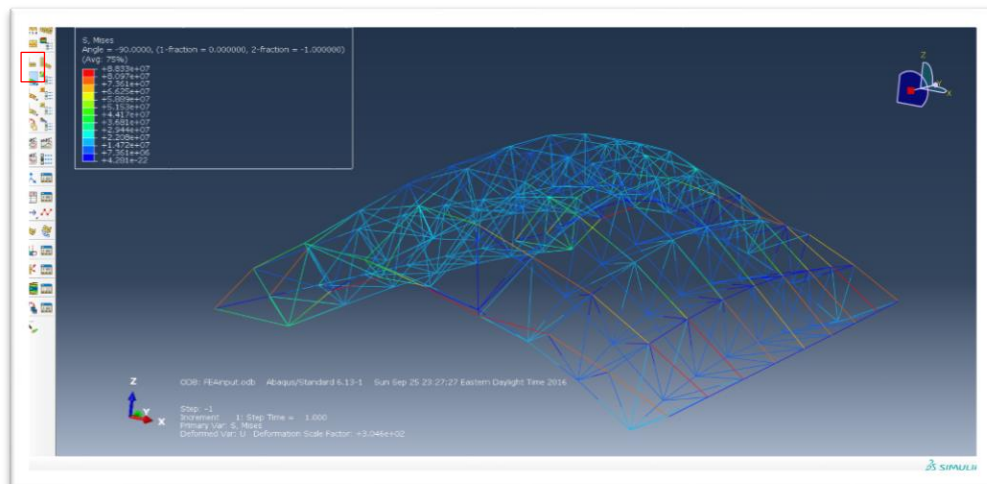
iMAG Interface (Conduct Optimization):

1. Choose two goals for minimization, and type the character d, w or s into the ObjectiveFunc1 and ObjectiveFunc1. (d for minimizing the maximum nodal displacement, w for minimizing total length of the structure, and s for minimizing the maximum stress in member).
2. Decide the priority (level of importance) between the two objective functions, and enter the value between 0 to 1 to Weight of ObjectiveFunc (WO1). (0 for least important and 1 to most important).
3. Type a mathematical expression into the Constraint Expression textbox. The format is  $C \leq 0$  where C is your constraint function.  
(e.g., If we wanted  $s \leq 200MPa$  then we write  $s - 200MPa \leq 0$ ).
4. Type the variables used in your Constraint expression textbox into Variables in the Constraint expression text box (separated by ',' if variables more than 1). The variables should be between s1, s2, s3, d, w, and s. (e.g., if constraint expression  $s1 + s3 - 20 \leq 0$  then we type s1,s3 into the textbox).
5. Click the button: run optimization to get the optimized result.
6. Wait until the result appears as shown below



iMAG Interface (Run an Analysis and Check the Plot of any Structural Analysis):

1. Enter the numerical values of s1, s2, and s3 under Run an Analysis Stage.
2. Click the button Run an Analysis.
3. Open your working folder and open the ODB file to access Abaqus user interface.
4. Click the icon as shown below to see the plot of the displaced structure based on the values of s1, s2, and s3 that you entered.

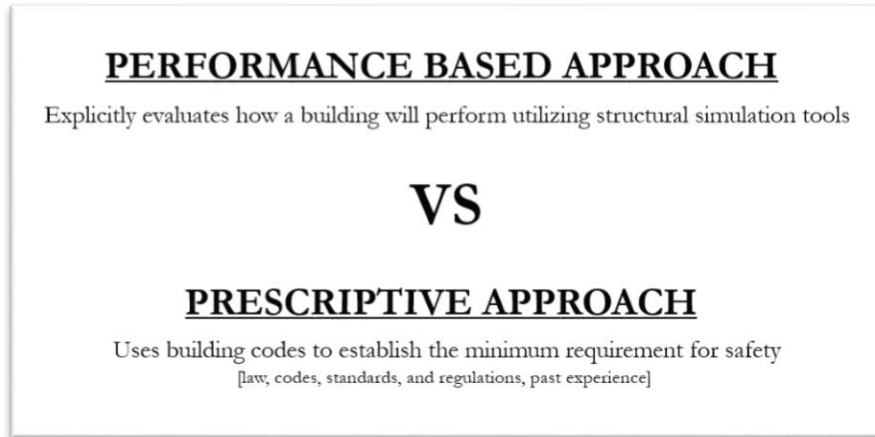




## APPENDIX B: Interview Module Transcripts

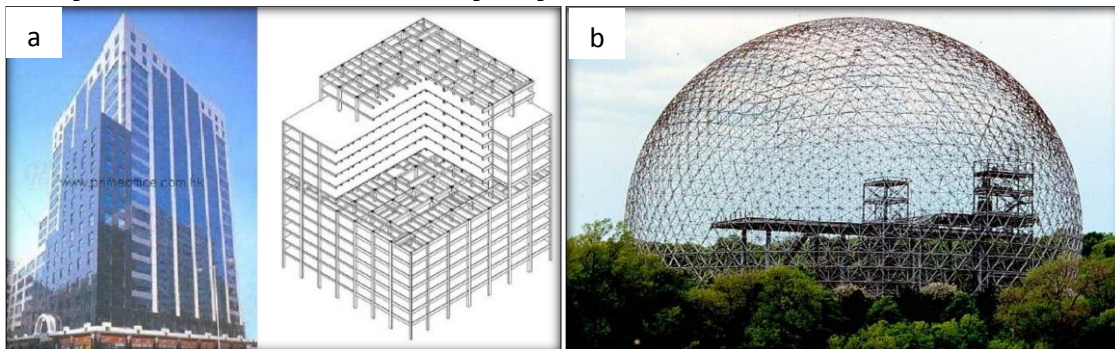
### **Background Module**

This video is for the purpose to introduce the difference between prescriptive and the performance-based approach for conducting the qualitative interview as part of the research process for developing iMAGv1, for architectural parametric structural optimization in the schematic design process. So in the background module, we start with the difference between performance-based approach and prescriptive approach.



What is the difference between these? So, the performance-based approach explicitly evaluates how a building will perform such as utilizing structural analysis software. On the other hand, the prescriptive approach uses building codes to establish the minimum requirement for safety. The prescriptive approach mainly based on the law, codes, standards, regulations, and past experience. The performance-based approach aims for accuracy and can be used to deal with the new type of structures in which codes are not available. The performance-based approach is often used as the basis for improvement of the existing codes.

For example, we can kind of see what type of approach that architects or engineers will use for designing the structure.



The project's name here is Stanhope House, designed by Lee and Associates, and the location is in Hong Kong. The building's structure uses rigid frame system, the design is rather conventional, and we can

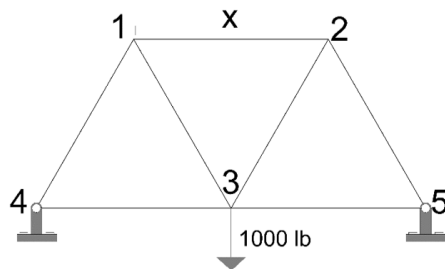


see this kind of building everywhere in our daily lives. So I would probably guess that this building uses prescriptive approach due to its conventionality.

Compared to this, the project name is Montreal Biosphere, the architect is Buckminster Fuller, and the location is in Quebec, Canada. We can see that this building is more unconventional than Stanhope House. It uses geodesic configuration. The inner and outer shell uses icosahedron trussed tessellation with a different frequency to form a space truss system. What type of approach that this building will use most likely? Probably, the performance-based approach, right? Because of this building in unconventional, it's really rare to see this type of structure, and we probably have to go to Canada to see this kind of structure. So, for the structural evaluation, due to this unconventionality, its uniqueness, the designer has to analyze the structure, because there are no available codes that can be used to design and analyze this type of unconventional structure. Based on this information, this research is contextualized under the assumption that architects are in need for the freedom of designing complex organic shapes such that prescriptive codes and rules of thumb are not applicable for the structural evaluation, and that performance-based approach is a more realistic approach in this case.

#### Education Module

This video is for the purpose of briefly introducing structural optimization for conducting the qualitative interview as part of the research process for developing iMAGv1, for architectural parametric structural optimization in the schematic design process. So, here, as we can see, we have a planar truss model.



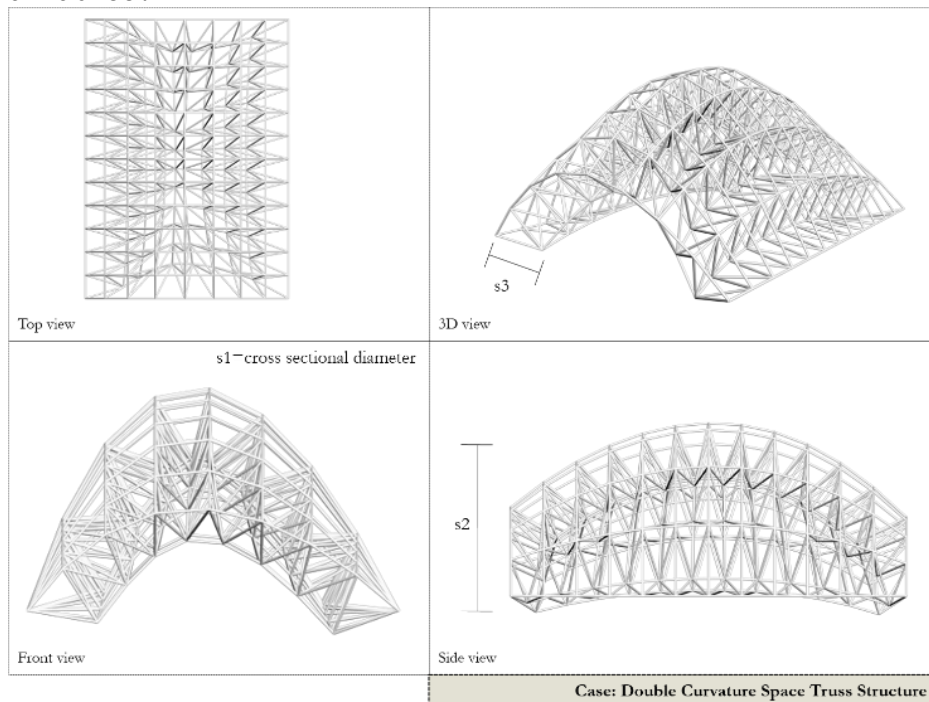
$$\begin{aligned} \text{Min: } & \max_d \\ & \text{w.r.t. } x \\ \text{s.t.: } & \sigma_e < 250\text{MPa} \\ & A < x < B \end{aligned}$$

We can see that node one, two and three are unrestrained, and node four and five are fixed. We apply a load downward at node number three of one thousand pounds, let's say, it can be any number. In this problem, we have only one design variable, which is  $x$ . The variable  $x$  is the distance between node one and node two. So, from here, let's say for the design requirement, we limit the length  $x$  for not to be too long, or not to be too short. So, we kind of restrict the value of  $x$  such that it should be between two values, namely  $A$  and  $B$ . Also, note that the truss uses steel pipe, and for the steel material, as we know, the yield stress of the material has to be less than 250 mega Pascal. So, if the normal axial stress of any of these members is more than the yield stress, then basically when the member deformed, it will not go back to the original form, because it has already passed the yield point. Then, as we can see here, our goal is to minimize the maximum nodal deflection, which most likely to occur in node three. So, we can kind of see that there is the goal, we also have stress constraint, and here we also have a geometric constraint. Here, we can call this as

shape optimization. We can kind of see, for example, if we run the optimization, such as using software, basically it generates some solution. Then, after running the process, maybe the design could change from here to this one below. You see? This is node number one, two, three, four and five. The distance  $x$  between node one and two changes, so the shape is changed, and so it is called shape optimization. Perhaps this one below possibly performs better than the design above in terms of minimizing the maximum nodal displacement, and restricting that maximum stress should be less than the yield stress. Or, maybe this one, the design on the right. You see, the topology is the same here, but the lengths in some members change because we are playing around with the distance  $x$ , and it affects other lengths to change as well. So, this is what structural shape optimization basically do, to find a form that is optimum in terms of the structural performance.

### Demonstration Module

Currently, we are continuing with the demonstration module of iMAGv1, for conducting architectural, structural optimization in the schematic design process. So, this is the example that we will use to conduct the structural optimization. We have a space truss structure, and we have three variables.



The first variable  $s_1$  is the outer radius of the steel pipe's cross-section for any of the truss member shown here. The second variable  $s_2$  is that height of the structure. The third variable  $s_3$  is the depth of the space truss structure.

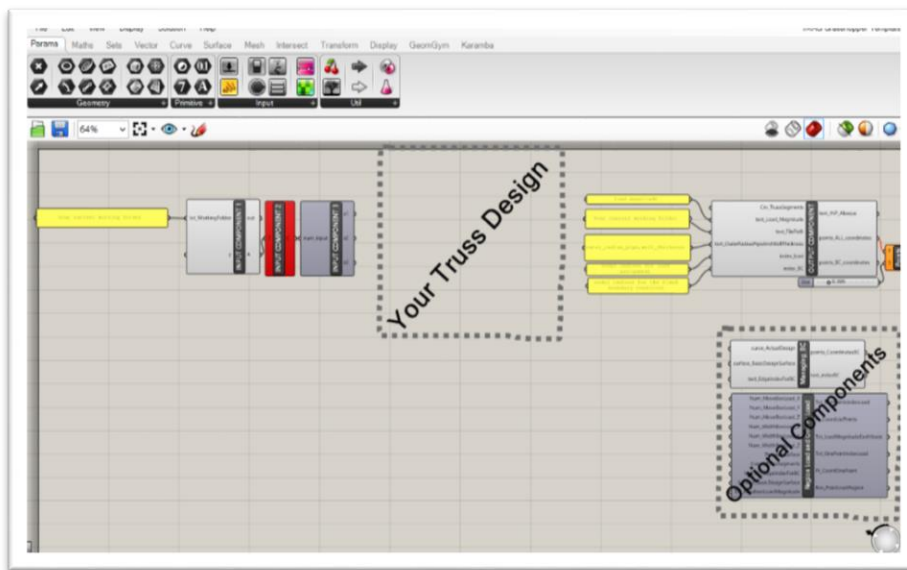
Here I have the user manual for iMAGv1. We will follow this step by step. Basically, this iMAGv1 is the integration of Rhino Grasshopper, Abaqus, and Matlab. Grasshopper is for the parametric modeling, Abaqus is for the structural analysis, and we use Matlab for managing this two software. So for the preparation, we should have all these three

software, and we have to create a new working folder. So in here, I have my new working folder.

Name	Date modified	Type	Size
checkLCK.m	3/7/2015 8:04 PM	M File	1 KB
ComputeWeight01.m	5/3/2016 12:25 AM	M File	3 KB
confun_cosine_interpolation.m	9/24/2016 4:39 PM	M File	2 KB
cosine_interpolation.m	9/24/2016 6:32 PM	M File	1 KB
func_main01_cosine_nix.m	9/24/2016 12:02 AM	M File	2 KB
func_TessCurve_ExhaustiveOPT_Feb20...	9/23/2016 6:59 PM	M File	3 KB
iMAG Grasshopper Template.gh	9/25/2016 11:06 PM	Grasshopper Docu...	91 KB
iMAG_GUI.fig	9/26/2016 12:08 AM	3dsMax biped figu...	44 KB
iMAG_GUI.m	9/26/2016 12:08 AM	M File	34 KB
interpolate3DCosine01.m	3/16/2016 11:22 PM	M File	7 KB
objfun_cosine_interpolation.m	9/24/2016 6:29 PM	M File	2 KB
readoutput.m	10/22/2015 4:04 PM	M File	3 KB
SFDOUBLECurveTess_analysis_3_OBJF...	9/22/2016 6:38 PM	M File	2 KB

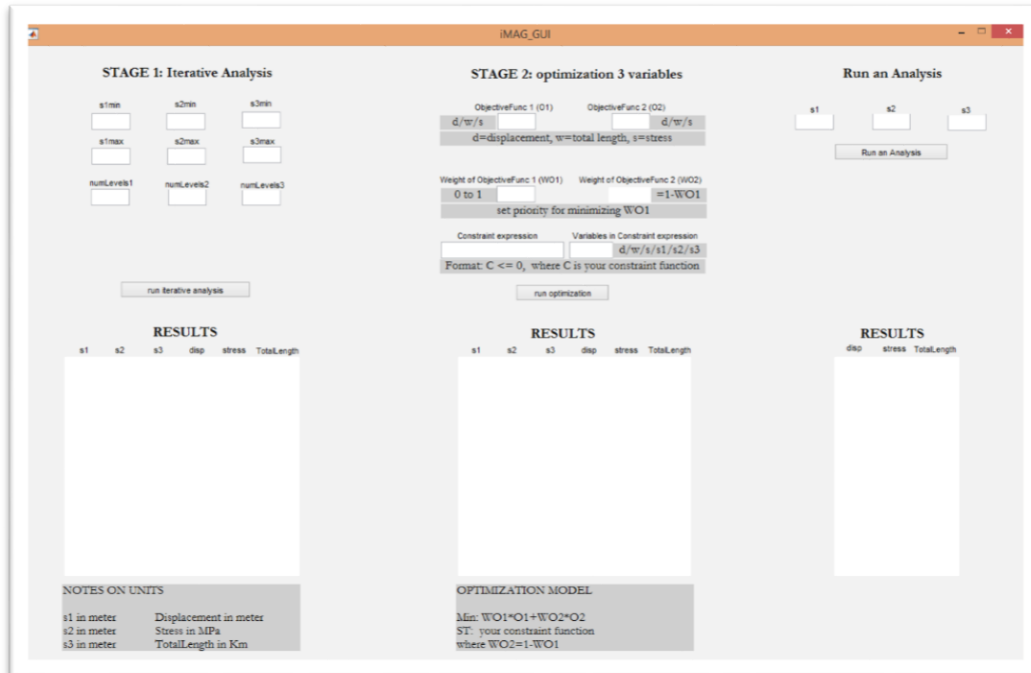
Your working folder should contain all these files

So, everything that we will do will be in this folder. Then, after making this working folder, what we have to do is to create our own parametric model in Grasshopper. So, here I have an example of the parametric space truss model. So, here we need to choose three design variables, namely s1, s2, and s3. Then, for the step number four, we need to open the iMAGv1's Grasshopper template as shown and copy the whole components into our parametric definition.



So in the template here, our parametric model will be located in the middle of these two premade components. The optional components can be used or not; it's up to you. These optional components can be used to ease the structural analysis setup in managing the boundary and loading condition. The next step is to write the directory of our working folder manually. Then what we do next is to link the three outputs here into our design variables to change our geometry parametrically. We also want to copy paste the last component here into the last section of our parametric definition. We also want to write our working folder here in the text panel. We also want to write the indices where we want to assign the concentrated load and boundary in text panels. Then we connect these panels into the last component here. We also type the magnitude of the load in the text panel and connect it to the last component. In here, I use the optional premade components to help me

managing the loading and boundary conditions, so I don't have to type them manually. But you can also type it manually using text panels. The next step, we want to copy all iMAGv1 files to our working folder. Then, we open the file named iMAG GUI. Then what we want to do, is to run it. The interface will pop up; then we enter the values for s1, s2, and s3.

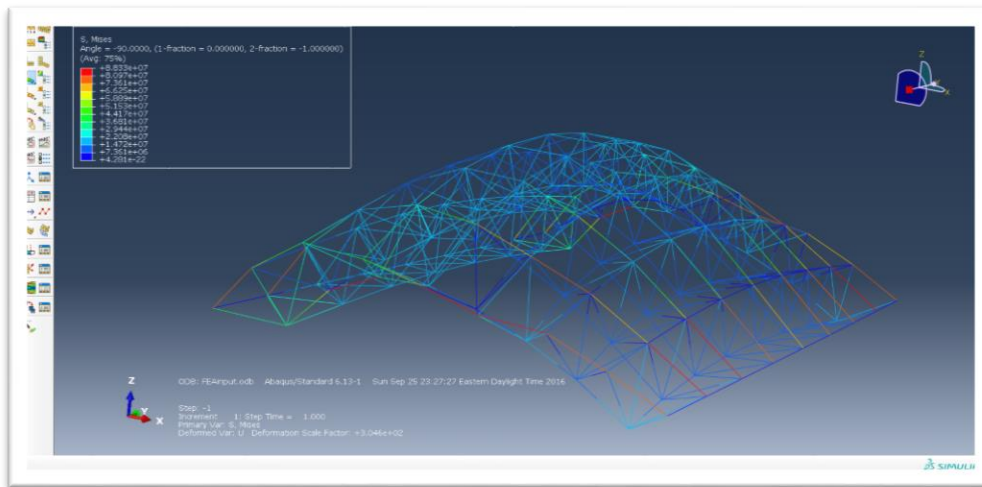


So these text boxes here define the lower and upper bound of your s1, s2, and s3. So our results will be between these numbers. Then we define how many levels you want between the lower and upper bounds. So in stage one, basically it runs the iterative analysis by calling Grasshopper and Abaqus at each iteration. It keeps doing that, and it will generate some results. In this case, since we have two levels for each variable, so there will be eight results corresponding to the changing values of s1, s2, and s3. The unit of displacement here will be in meter, stress in mega Pascal and the total length is in kilometers. While waiting for stage one to finish, we proceed to stage two.

So in stage two, we have to choose two goals for minimization. We can type the character d, w, or s into the text boxes. The letter d is for minimizing the maximum nodal displacement, w is for minimizing the total length of the structure, and s is for minimizing the maximum elemental stress. Let's say, I want to minimize the maximum nodal displacement, so I will type d, and I will also type w for minimizing total length. Then let's say I want to put more priority to minimize the displacement, so I will put one in this text, and the other text box here will automatically set up to be zero. If I put the value zero point eight here in the left text box, then the right text box will be zero points two. So, anyway, I want to prioritize to minimize the displacement, so I put the value one here. Then, we put the constraint. We want the maximum elemental stress to be less than two hundred mega

Pascal. So we type it in this text box. Then we type the variable  $s$  here, indicating stress. So take a look here at the bottom. The objective function of the multi-objective optimization here is to minimize this formula. We can plug the values in the text boxes above to this formula here; then we get minimizing one times displacement plus zero times the total length, which will result only to minimize the displacement. So here we have already the result on the left. Then we can push this button to run the optimization in stage two. So here, we have the optimum values of  $s_1$ ,  $s_2$ , and  $s_3$  that minimize the maximum nodal displacement, and we see that the stress satisfies the constraint.

So, what we want to do next, we want to visualize our structure. So we just need to type these optimum values into the text boxes here. Then we push this button. So, what it will do is that an odb file will appear, and we will open it to visualize our structure. We open the odb here. So this is our best structure now based on the goal and the given constraint.



So here is the displacement plot of our structure. So if you see here in the Grasshopper, the load in here is applied in here in all the green nodes, and the boundary condition is applied in these green nodes. And that's it. Basically, we have the optimum design here based on the given condition.

## APPENDIX C: IRB Approval Document

### Exempt Determination for IRB2017-190 | Integrating Parametric Structural Analysis...

Amy F Smitherman <[smithe3@clemson.edu](mailto:smithe3@clemson.edu)>

Tue 5/23/2017 1:22 PM

To: VINCENT BLOUIN <[vblouin@clemson.edu](mailto:vblouin@clemson.edu)>;

Cc: Naline Patin <[npatin@clemson.edu](mailto:npatin@clemson.edu)>; nixon wonoto <[nixon\\_arch@hotmail.com](mailto:nixon_arch@hotmail.com)>;

Dear Dr. Blouin,

The Clemson University Institutional Review Board (IRB) reviewed the protocol "Integrating Parametric Structural Analysis and Optimization for Architectural Schematic Design Phase" using exempt review procedures and a **determination was made on May 23, 2017** that the proposed activities involving human participants qualify as **Exempt under category B2** in accordance with federal regulations 45 CFR 46.101.

No further action or IRB oversight of the protocol is required except in the following situations:

1. Substantial changes made to the protocol that could potentially change the review level. Researchers who modify the study purpose, study sample, or research methods and instruments in ways not covered by the exempt categories will need to submit an expedited or full board review application.
2. Occurrence of unanticipated problem or adverse event; any unanticipated problems involving risk to subjects, complications, and/or adverse events must be reported to the Office of Research Compliance immediately.
3. Change in Principal Investigator (PI)

All research involving human participants must maintain an ethically appropriate standard, which serves to protect the rights and welfare of the participants. This involves obtaining informed consent and maintaining confidentiality of data. Research related records should be retained for a minimum of three (3) years after completion of the study.

The Clemson University IRB is committed to facilitating ethical research and protecting the rights of human subjects.

Please contact us if you have any questions and use the IRB number and title when referencing the study in future correspondence.

Good luck with your study.

Best,

Amy Smitherman

IRB Coordinator

Office of Research Compliance, Clemson University

[391 College Avenue, Suite 406K-1, Clemson, SC 29631](#)

T: 864.656.6460

[smithe3@clemson.edu](mailto:smithe3@clemson.edu)

IRB Email: [IRB@clemson.edu](mailto:IRB@clemson.edu) (send all new request to IRB inbox)

Website: <http://www.clemson.edu/research/compliance/irb/>



<https://outlook.live.com/owa/?realm=hotmail.com&path=/mail/inbox/tp>

1/2

## APPENDIX C: Sample Interview Transcript A (1<sup>st</sup> Cycle)

### **Answer to question 1**

I think you do it after you find an exterior form of your building and maybe after delegating floorplan, what each floor looked like, after putting elevator cores and stuffs like that. After those then you can see what kind of structure is it, and be able to test the structure. But obviously, if you are doing something more radical, the earlier the testing, the better probably to make sure that you have the ability to move forward instead of getting to the certain point in the design of this crazy new innovative building and then figure out that it cannot even really be done. I guess structural analysis and form are influencing each other, and it kind of like back and forth processes. There should be a strong dialog between the two. It's actually something I do with my project now because the shape is unconventional. So you can play with the form forever but then you will kind of need to start adding structure to say that this can be done and that cannot be done. Then say that if this cannot be done, but this variation can be done by tweaking the structural analysis, and now say that that work, but then ask whether it is still having the same appeal as what I want.

### **Answer to question 2**

I never really done structural analysis in any of my school projects. I know that there is much different software that you can use. But for the scope of our project, we don't have to go too much into structural detail really. But I guess if I can work on the project with some engineers, I have some engineering friends; I can talk with them and see if they knew a way to do it. But even in the engineering department, they are not teaching them to use the performance-based method. They are using prescriptive. So I think it's pretty prevalent in engineering school to teach the basic in which you see every day rather than pushing it into a new innovative way and more efficient way to do it. I don't know any name of the software that does the structural test. I just saw portfolios in professional work that has the map that looks like almost like a heat map that has yellow, red and green that kind of show which part that are structurally sound and other parts that are red maybe need some works and that they are not gonna be as effective. My structural engineering friend would be like "why are you doing it like that? If you do it this way and then cut this off and simplifying it to the basic, that would be more efficient and stronger structurally and cheaper." So they think less about creativity and aesthetic. Sometimes the suggested simplified design is not what I want, but sometimes you just have to make compromises so that it can become structurally sound. But again I think that is predominantly used prescriptive approach and not the performance-based. I think with the performance-based, it's kind of more innovative and can take the form that has not already been done before. Normally people will say that since it cannot be done before so it cannot be done, and the prescriptive method says that it cannot be done. But I think the performance-based method will allow more freedom. The structure that I asked my engineering friend about is the simple pedestrian bridge that spans around 50 feet gap. The way I supported the bridge, I think to use the method of ruled surfaces. The structure was a very organic shape. My engineering friend analyzes that

structure just using some kind of eye-ball test such that saying that that could be done, but it would be very expensive. We didn't do any kind of structural loads or anything like that.

**Answer to question 3**

I think incorporating structural analysis into early design as form-finding method would be great because then it helps generate form, several iterations maybe, over and over again, and from that form that is generated you can know which one is structurally sound, because it checks it off even before you start, so you don't have to make adjustment down the road because you haven't thought about being structural. I think doing structural analysis at the schematic design phase by the architect would be very helpful because there is huge gap between architects and engineers I think, and this would help to start kind of building a bridge, and even if it is not a very specific structural analysis, the interface is not too hard to operate, architects could incorporate it into their firms, and save a lot of time and money, and that's what the client would want, besides getting beautiful buildings. I think the client concern about the time and money as well. I think that would be very helpful.

**Answer to question 4**

After watching the video, it seems the process will find the most structurally efficient building, but it is not necessary the best form. Also, it will give you all those variations of them. Just because it is one hundred percent structurally sound does not mean that other options are not acceptable. So you should have still some room to play with it. But also I forget what building it was, but I think it is in Shanghai. The building is like a vertical city, has ten city blocks. They have an interior core and the exterior cores, and it actually moves within the wind, and it's like twisting a little bit. When watching the video it, they used performance-based to see which one is the most structurally sound and that they do now want to rotate it that much. The twist is a bit to find the form that would stand up and not having the problem. But if something really catastrophic maybe something would happen, but they set a threshold and say that it can be safe enough and that it matches with the form that we want. So in a sense like that, it would be good for form-finding and optimize. So if there is this point where the design would not be structural at all, and the other point which is one hundred percent structural such that no problems with this building, but the threshold would be somewhere in-between about what they need to do by using prescriptive-method. Other criteria that I use for form-finding are an inspiration. I think the inspiration for that building was there is an area with a lot of typhoons. So the twist of the building was inspired by the wind or something like that. So, in this case, it's the climate become the inspiration. Also, the program of the building, like you don't want to do something too different, depends on the setting of the building, what it's for. But you don't want it to become a duck architecture like there is a doughnut store which a giant donut with it. Also, may you do not want it to open. There are a lot of factors other than strict structure. People keep saying that architects have a big ego and that they are going to put something out there in the world that is unique as if it is a piece of them, so they want to share it with the world and separate it from the buildings out there. I can give you a list of the importance of the various aspects need to be considered in architecture and I think



number one would be safety, so structural would be number one. As if I want to design a building, I want it to stand up, and I want the people in it to feel safe about the building and not worry about it collapsing or something going wrong inside the building. So safety is always a top priority, and from that you can set the parameter of what you can or cannot do based on that safety one. Because if it is total freedom, it would be hard to design something because you can just go on forever. But once you are starting to set parameters for yourself in design, that is when you can start to use those rules to your advantage. It is kind of like a game, and you play it by the rules of the game, but you can still win the game, and winning the game would be designing an aesthetically beautiful building. I also think that you do not want a plain building because sometimes the most structural building is straight everywhere. You can really put the style with that plain building like putting glass and concrete, but it will still look like not inspiring and does not show emotion to me. The second building in your presentation, the one in Montreal, I can say in my head that I have never seen anything like that before. So I think innovation would also be up there somewhere, something new, something different, something better. Sometimes I do eye-ball test for the structural evaluation, like looking at it whether it can be done or not. Like for example I know I cannot make a building that is floating in the air, and I definitely know that it cannot be done. But I used to see a building with a cantilever that is way longer than any cantilever I have ever seen. The cantilever is almost like the height of the building. I did not think that it would be possible. So even what I have in my head about what is possible and not possible might not be right. So structural analysis or structural optimization would be very helpful, to have the program that could assist architect with that. I used to take one structural course, and it is not as in-depth as the course here in Clemson University in which I will be taking that in the next Spring. In that past course, they teach about tower or bridge. We did not do any math structural analysis test. What we did was had the structure span a certain distance, and then we tie a string to it and keep attaching bricks until the structure collapse, and recorded it. They were some math structural analysis in the books, but nothing too in detail, so it's more experimental. I think here the school architecture is more theory-based and not very practice-based. So it gives the buildings and the students here more artistic freedom of design but not necessarily the best ground for structural knowledge. So that course was full experimental. So in the studio, I used building codes, because I build a project in the district of Columbia Washington DC, some for safety and some for aesthetic reasons. Like you do want the building to be that high, so to get more like a human scale of the city. The most organic shape we deal in the structural analysis course is the tensile structure. So it's like cables. So that's the most organic shape. Other than that, they say build a tower but might not know if it is going to stand or not, so they tested it, using bricks. For the tower, they used sticks. I am not afraid of doing the math; I think it's a good way to check what you are doing rather than just experimenting until it breaks. The experimental in a sense helpful like giving a very basic understanding of it, so you don't get like totally don't have any idea for it.

**Answer to question 5**

The video was pretty clear. Three different programs generate these results. Like, all the math maybe be unclear in my head, but I can understand how it's done. Like the table varying certain aspects of it within the threshold to generate the best. As for a Clemson Undergraduate program, I don't see it being frequently used. But what I think it can be helpful is that I have friends in engineering who are thinking to graduate school in architecture. So someone who has an engineering background and is familiar with those programs, then coming to the architectural program could be very helpful to help them generate the forms that are also structurally sound. Because with their background they would not just design a building without knowing that it could stand. I think in their head they might always think the way to emphasize that for validating it. I think when architect use that can be confused. Also, I don't see other options of how the structures will look like. I think it would be good if I can know how much does it vary. Because if it only slight variation within sets of parameters, like it would be structurally sound, but it would not affect that form that much. So I would like to see the variation. To see what was option one, option 2, to see those different forms were. You might see that, alright, this is the most structurally optimal, but that other design looked way cooler or more aesthetically pleasing. So it was like the previous case I mentioned about like it does not change the form drastically, but they cut the form. So then, they could know that anything past this point would be structurally sound. So it's like knowing which one you want, and then can take your pick, rather than generating this one, and deciding that I'm going to take this one it's the most structurally sound. I think the tool is effective; the math is a bit confusing. Like I would not use this now for my project. I think the problem is when it shows the structural analysis program. Like I am also confused by the Grasshopper as well. The table is pretty clear, and the video shows a good reason for the numbers when they were plugged in. I don't know where the numbers come from. You told what the numbers were, but as far as why you picked those numbers were a bit confusing. I mean I know you pick the one, like s3, the actual structural integrity of material used, and it's understandable, and then you talked about certain ones to minimize distance for the cost of the material. So I mean yeah, I have the basic understanding of when you varied that that will change the structure and the structural integrity. But as far as how it will change that, I have don't have the idea. Like how the structural performance will change corresponding to the variations. If the tool allows more than 3 variables, then it will be more complex. But at the same time, it will allow more control over what is generated, so I think that's a plus. But also the learning curves to be able to know what each of the ten variables does and sometimes maybe it could be overwhelming. Maybe you do only need three to get a certain amount, but it would be even greater to have more option.

#### **Answer to question 6**

It would be great if the tool can show what would be acceptable, not necessary the top one, but the top five for example. The building does not have to be one hundred percent structurally sound, but just need to pass a certain threshold. So then I can get several different forms corresponding to those. I still want to do something more after the optimization considering the style and all within the allowable

threshold after the optimization. I want something that I can still put a little bit of control to it but allow it vary a little bit. So I can say that this may be the best for the structure, but is it the best for the form? Or is it the best one for aesthetics? Because it's important to develop a style.

**Answer to question 7**

If there is a visualization of the results, I think it can be useful to architects. But the way I look it right now it seems its more towards numbers and that kind of scares me to use the program.

## APPENDIX D: Sample Interview Transcript B (2<sup>nd</sup> Cycle)

### **Answer to question 1**

The architecture degree in Spain I got was from Polytechnic, and it was considered equivalent to engineering degree, so the level of difficulty the structure is very high and actually as soon as you were finished from university and got the license you were legally allowed to calculate your own structure, and most of the people that calculate structure there was architect, not engineer. I did calculate some of my own structure, and certainly, legally can calculate my own structure. But, my degree is M. Arch, but also have that technicality. I was working in my office, leading my office, and we calculated our own structure. I also have Ph.D. in architecture. What I think is that for an architect is essential to establish a conversation with the engineer. According to my experience, as a designer, we did not have an awareness of the possibility of structures, what entails, what can be done, how it affects budget and size of things. I think to me, as an architect, you need to be able to talk to an engineer and to know what is possible. And I think the engineer, to take the possibility of the structure much further. I think the conversation is demanding on both sides and needs to overlap. I think the engineer is a bit savvier, and architects more into innovation. We worked in different materials, typologies, complex shape, a wide span, cantilever, that requires very precise and performance oriented methodology. We always have to simulate and modeling structure. I don't agree with the term organic and conventional structure, because a conventional structure can also very difficult, and they can have another type of difficulty. There is this Brazilian architect, that is very structurally oriented architect, he won the Pritzker price years ago, he worked with concrete, he optimizes, and he did a lot of technicalities. We were using BIM, I don't remember the software, Rhino for modeling, and most of the time those are the files directly used for evaluating the structure. So I could not tell you the software because I did not do it myself. For the structural analysis software, I have the documentation of the project, but I forget. The architects in my consultant are the ones do the structural analysis. Always in practice, we calculated things, we more or less already know the size. The structural problematic also part of the design. So we don't work with the shell that we don't know it structurally. We always know where it's going to be thicker, lighter, and we always adapt the form to it. We calculated the structure ourselves. For example, if we have a simple form, we cut, carving out, creating discontinuity, and we would know which part in compression and tension, which one is thicker and thinner, and that kind of logic will be in the design, we will do those by hands, then goes to the simulation. We used section methods for the beam, and have the approximation of the loads that we will have, for a truss, we will joints method. Right now in Spain, the structural analysis is more matrix oriented, every piece is considered. In our times, when computers not so used, we used more the joints or section methods like how we did in our office. I think now it's more like matrix approach. There were like 4000 students, one of the biggest ones was the department of structures in Spain. Architecture in Spain is not in fine art or humanity, but it's more on the polytechnic institute, within the

engineering. For your first questions, is more in the US, architects not that much aware about structure, and that will affect the design, the design will have lower quality for that reason. The thing is that the structural consultant in Spain is an architect, not an engineer. The engineer will do bigger things like a bridge. The rules there would be architects are the best structural engineers. For the payment method, usually all the money goes to the architect, and then the architect hires the consultants. The finished project signed only by the architects usually, and then he hires other people, except if the project is very complex, he may hire someone to do the structure in terms of the insurance. But in my consultant, all the structure fully covered by my insurance. So if the building falls, I go to jail, not the engineer.

**Answer to question 2**

There is a gap in terminology. Is optimization to take one joint and analyze it separately? In my curriculum, we have two years in Calculus, and algebra before we touch any structure. I am not familiar with the SQP and Genetic Algorithm. In cantilever, we take the maximum deflection, and we limit them and will get the size of the structure. Optimization is not used by us for form-finding. I have the documentation of the project here; I could show you if you want.

**Answer to question 3**

I think structural optimization can become a consistent approach to taking a decision in architecture. I am interested in architecture that is formally non-conventional. I think structural optimization makes thing better if we know how to use it. I think there are not so many architects in the US are exposed with structural optimization compared to those in Europe. I used Grasshopper to work with several elements of tensioned structure, and have to find balance, sizes, the weight of elements could be fully dependent on structural balance, and we play with the possibilities, like the position of elements and sizes. We directly used Grasshopper in the consultant. In this case, we obtain certain model then we can choose. Grasshopper is for the schematic, feasibility study for competition; it gave us for a design that was structurally feasible. I don't know what structural analysis they use. I don't remember what files that are sent from architecture software to structural analysis software. I was not involved in the production; they were just tools. I am not familiar with structural analysis software developed by architects.

**Answer to question 4**

I think the tool has the potential to make the architecture better. If you want the architect to use it, specifically in the US, there are still space to make it user friendly, making it more direct, so that you don't have to type, so need less input of data, and will be good if the GUI shows the structure available in all of the steps, so fewer numbers, and more drawings. So most of the problem are on the table, and if you know how to use Grasshopper, it's easy in the Grasshopper part. I think the tool is really good as form-finding. I think the architect when using this tool will be able to find a more efficient material in terms of weight, less material and use it more efficiently, but I also think, when we are using only codes for architect to evaluate structure, I think it's a deception, but I think when they can follow this kind of process using the tool, I think new type of architecture would be possible. Another question, when this would go

for construction, how would this be understood by the builders? But I think for architects, I think this will be liberating and positive, and for architecture students, this tool will help them to do a better project. I think that this tool helps the overlapping parts between architects and engineers. Because this tool speaks the language that is readable by the engineers and at the same time it's speaking of the form in which we architects do. I think this tool should be incorporated into the early schematic design phase, in the very beginning, the earlier would be better. For example, we use codes in the very early of design. I know some architects use it later. So we are using codes in the very beginning, so why not using this also at the very beginning? For American architects, I think this tool can be used at the very beginning. I am a little skeptical about architectural students can interpret the results properly from the tool. The technical level of knowledge the students have here is very low. In Spain, a lot of young architects use Grasshopper a lot; they can interpret the structure more properly.

**Answer to question 5**

I think it would be good to have the graphic results since we are dealing with the size of the bars, it would be good to see the form and the size of the bars. You can have beautiful forms, but if the bar too thick or too thin, it would be different. If it's the roof, I am not worried about displacement, if its floor, I would worry about optimizing displacement. About the stress, it's associated with the type of material you are using, so I think depends on how the tool used for. Ideally, if we would have a structure like a roof, I think to have the lightest structure would be best, but you cannot have the floor that moves, you need to consider. Ideally, the lightest perhaps the best structure, it would be cheapest and easy to bring them to the site. I think various optimal solutions are better because you can integrate them into the design. They have the variations to choose to adapt the design.

**Answer to question 6**

I think this tool should postpone engineer's involvement. In the design process, I think this fragmentation of the process between architects, engineers, and others are proved to me. As a European that live in the US, to me, it's clear that the more agent, the less consistent is the project, and the less consistent is the result. If you were able to have one agent that could have the first idea and move it as far as possible, it's easier, and would be more difficult to shortcut the qualities in the beginning. What usually happens when you have this fragmentation is that the architectural idea, once an engineer gets in, will be dissipated, also because the architects do not take account the structure. If the architect can take account the structure, and thermal behavior and all of that, and have this kind of tool, I think it will be good and can take the project far ahead in the process. There would always be an engineer in the end, but, based on my experience, it's always easier if the architect has the capability to move as far as possible. In my experience, I have a system to work like that. So there is no frustration because there is no other expectation because people know that it works that way. If you have a very experimental project, and the architects move really forward, and he needs an engineer, I think the engineer would always be happier to get a result that has already this kind of intelligent embedded. When we architects say that

the engineers betray our project, the problem is in the concept themselves, because they don't have the intelligence in the first place. So I think if architects move forward as much as possible, and the architectural product is structurally intelligent, then it would be a more productive collaboration. I think this way; it does not only make life easier for the engineer but also will make the engineers' works better, and to make the more efficient structure and all. Because it's better to work with a high-quality product than working with something that has no quality. So I think this tool is great. I would like to have this to use it in my office like three years ago.

## APPENDIX E: Sample Interview Transcript C (2<sup>nd</sup> Cycle)

### **Answer to question 1**

Collaboration is inevitable. Specifically, in engineering and architecture, we have quite different perspectives in approaching design. Their parameters and our parameters are not completely the same. While architects are more interested in the perception of the building, the engineers are more interested in the structural quality or the materialistic aspect of the building. Material aspects are also architect's concern, but the way we see it quite different. We, the architect, may be more experimental sometimes, but sometimes even the recent engineers become more and more experimental. So depends on the engineer and architect. I have worked with two or different engineers in my practice. One of the engineers I worked with prefers conventional structures, because they are more interested in the codes, and they always follow the rules and would ensure the building function well, would stand up, the structure will stand up, and there would be no problem. But I work with another engineer, for example, when we draw a simple plan, the engineer would then complain that we, the architect, werenot challenging him enough. So that was such a simple structure. So the engineer can be frustrated because of that simple design. It is, of course, a different type of engineers. So the later one is a more experimental oriented type of engineer. So for the conventional structural engineer, we always use them for construction type of project in which building that will be immediately be applied. For the more experimental engineer, we always asked him for the competition. The engineer that is more experimental is for the design that is more crystal form, not organic. Sometimes, that structural engineer could help us to solve a façade that can at the same time perform as a structure. So there are different types of engineers. So one of them has experience worked in Arup, and the other one is just a local engineer. Both of the structural engineers I worked with are academia also. One of them is more interested in the codes, rules while the other one is more challenging himself and is more experimental. Sometimes, when working with the conventional type of engineer, they do not want a challenge, everything should follow the rules. In this case, it frustrates us. For instance, they priority is different than our priorities. For us, architects, sometimes façade is often time more important than other quality of buildings. Sometimes, for a mechanical engineer, the mechanical system can be more important, and they can sacrifice the façade. Likewise, structural engineers can play with the thickness of the columns without thinking about the façade aspect of the building. But, in the case of the experimental structural engineer that I mentioned earlier, he was trying to follow what we were asking and discussed with us. In my view, he is very exceptional, and one of the best engineer I worked with. Both of the architects are professors, working in the school. One of them is more challenging and form-oriented compared to the other which is more performance oriented.

### **Answer to question 2**

I was a consultant and a partner in Turkey before. So I was the consultant of an architectural consultant in Turkey, and I also at the same time a partner later on. The firm works both in construction and competition. But I was more working in the competition part of the



design. So the engineers we are working with were working separately from our consultant. They were actually changing engineers from project to project. So this also related to the cost of the engineer. For the local project, the architectural consultant would not risk for an expensive engineer, but for a challenging project, they would go for an expensive engineer. I have three different examples of the payment method for the engineer. In the first case, a housing project, the engineer was charging the owner who assigned us, the architect. So in this case, the client pays for the engineer. For the second project, a factory plant project, the engineer was paid by the architect. The third case, for the competition, the engineer was not paid, but the engineer will take a percentage from the grant prize from the competition price. So the engineer will get twenty percent of the prize. So for the second example, one design iteration and review would be the same to any number of design iteration and reviews. So what they are doing is more like a long-term process. Before preparing the final engineering project, during the design process, every week the head of the engineer or the assistant of their office will come to take a look at our sketches and comment them to give a recommendation. So in the design process, they are already involved, and they will come once a week to give recommendations and discuss with the use of what they are thinking. And eventually, after the design has been completed, they will work with the project and make all the calculation and make the final decision. The technology is now different. In the last case, we were working, we would send them digitally of what we are working, and they will send us the comments, call us and speak on the phone. And at the time, for the competition, we were sending our project, the engineers are in London, and we the architect are in Turkey. So long distance is not a problem. And then when the project is finished, they will finish with the detail. In our case, the model starts from the beginning, from the first stage using like Rhino 3D models in the computer, and we always send them. So they will not do any kind of evaluation until the last part, and that phase is after the schematic design phase. So this is for the competition project. For the construction project, there would be two or three stages of the design. The first would be the idea section, which was making decisions, the concept gets settled and done, making the first proposal for the client, and before the proposal sent to the client, the engineers will take a look at it. After the proposal was accepted by the client, then the engineer will start with the simulation to find out if roughly it will work or not. This is, for instance, for one of the projects we worked on, it's for the concert hall, and in that case, we won the competition. So before we submit to the competition, the engineer makes their first simulation. But after the competition, the engineer will start to make multiple of options, and start to explore. Because we won the competition, and the construction would start, and in this case, the engineer will start to look if there were other options for the acoustics or structural system. So mostly we are using Rhino. We do not use Grasshopper or parametric system. So the type of model that we send to engineer depends on the project. For the construction project we send them already with the structure in the model, and actually, the modifications that they will do will going to be the size of the beam, the size of the column. So they do not change much of the grid system. They perhaps edit only a few shear walls, it was the biggest change.

But in the case of the competition, we only send them the shell, and it is for opera hall, and it was very complex. So in this case, the skin or the façade is carrying the building. So the engineer will have to solve the thickness of the skin. So for this competition project, we only send them the plan, the skin, and what we design geometrically, but it was really abstract. So we do not send them thickness because we discuss the material together. We know it would be a steel structure, but we haven't decided, it was very complex, all angled, a very crystallographic form. So the engineer comes up with the idea about skin structure. So some of the skins are three-dimensional, and some only surfaces, but load bearing surfaces. We have structural courses, we are familiar with dimensions, and we use mostly the German standards for the beam size and column size. But the actual structural analysis will not be done by the architect, the only analysis and final decision, type of concrete, density, sizes will be by the engineer. So the analysis is done by the engineer. So for the optimization, in our cases, everything is done by eye. So it's more like trial and error, but we don't have the chance to build a mock-up because the material will be different than what we were using. So there is the time for the competition, the engineer was involved in the simulation, not us. So specifically for the case of structure and acoustics of the concert hall, they were working using simulation. We were giving instructions to the engineer, and get comments. Our focus as the architect is the spatial quality. So what the acoustics guys sometimes asked not the same as what the structural engineer did. So for instance, sometimes the material the structural engineer recommend does not match for the acoustics. Here is the technique, we were aware that there would be some changes in the design process. For instance, we use the method in the Baroque period, which is called poche drawing. Poche drawing is when you draw the thickness of the structure; the drawing separates what is visible and what not visible. What is invisible, we are giving tolerance as changing spaces, so it can act as an unknown structure ranging from one foot to three feet. So we are not sure that the size can change, so we would just go for the maximum. So the engineer will go to the place where we left empty and start to speculate that. So the white section of the drawings, the engineer can touch it, but the black sectioned one they should not touch it. If the engineer wants to modify the white zone, it's okay. So it's like color-coded. So for example, when we are working in the place where it has first-degree earthquake zone, ranging 6 to 8 Richters. So, in this case, we will very carefully follow the engineer, their codes. So everything depends on the project. But for the competition project, we, the architect, has more authority for the changes. But in the case of the construction project, we give more authority to the engineer in the final.

### **Answer to question 3**

So structural optimization for form-finding can be useful depending on the architect. If you look at the history of architecture, art form and structural form are two different approaches in architecture. There are two types of form, structural form, or core form, and the art form which covers the structural form. In some architects' approaches, the structure should not be exposed; the structure should be coated and covered. If you see Mies Van Der Rohe's columns, his columns are all covered by another layer, which will tell you a completely different story. So even though he will use an optimized column, which is quite

thin, he will cover it again with another thicker layer to psychologically give the feeling of security. So, for them, the structure is not the priority. In that case, they will not go for optimization. The optimization may be there, but they will cover them again. But the architect like Cecil Belmond, he will give all the authority to the structure, and his optimization will determine the form. So depends on the architect. In my view, coming to the final decision, sometimes structural results are not so aesthetically appealing. In my view, those visual representations, art form, are necessary. If you look at Toyo Ito, the skin is carrying the building, so the skin is the structure. So there is a different approach. There are limits of tectonic also. If the structure is there and is carrying the building, then it should be visible to an extent. So we have to choose whether the art form or the structure form is the priority. In the gothic architecture, its really difficult to separate the aesthetic with the structure, because all pieces are organically structured. I don't think the architect is searching for an optimum structure. The final result is an optimum condition. So what is the best solution? Because if you have a structure that works completely perfect, all sizes are perfect, systems work perfect, then how about the climatic condition, acoustic and so on. So there are multiple layers of design. In the case of a concert hall design, what the structural engineer sometimes said the opposite to what the acoustic guys said. The acoustic guy said that the engineer would have to use special concrete bricks that would absorb the voice. But that material was overloading the beam, and thus the structural engineer was so angry and said: "are you insane?." But it was necessary, so the beam would then have to increase the size. The structure was not optimized, but the acoustic is necessary. So there are many aspects. Depending on the offices of architecture, some offices in San Fransisco is fascinated by the idea of affordable construction. So for this office, I think they would be very interested in the optimization. But in another extreme, Frank Gehry, only focus on the look than the structure. So they have no constraints at all on the optimum structure. So if I was invited to build a country where everything is based on the cost of the construction, I would definitely think about using optimization. Then in this case, if the architect has a structural optimization tool, I think it would be a big advantage because they will be able to see if the structure work or not. So this structural optimization can be part of the design process. Then they will start to think about the material in terms of weight of the building. Look at Renzo Piano; he is very conscious about structural optimization, he knows structural optimization. This because Renzo Piano has a structural engineering background. The background of the architect is also important. I never use parametric modeling or Grasshopper. I don't use structural analysis software and so on. I am a theory and history person. I am more interested in the social and humanistic aspect of architecture. So there is a little bit distance between me and that technical field.

#### **Answer to question 4**

What I found interesting when watching this was about the design process in the competition, so even though it would not be built, it would make it coherent, to make it work, we will definitely use this tool. I will definitely use it. Do you remember about the white space we left? At least we can check with this tool if its make sense or not.

So we will definitely use it to make it coherent. You know, for instance, in the architecture competition, they do not pay attention so much to how the structure works, they are more interested in the spatial effects of the project, but it should be coherent. So this tool will give a great sense of security to our office in terms of what they are doing. So the usage of it will depend on the case. So in the last competition, this tool might not work, because the skin is carrying the building, so the structure is part of the design, that is why we include the engineer into the design process. But in the previous case, when we were doing the concert hall, this tool would work to give the outline of the structure roughly. This tool would be useful depending on the size of the office. In the small office, this tool would work, because small office sometimes is very difficult to find an engineer, and this tool would work for their reference for the engineer. I don't think this tool would be used as a design tool, but more like to confirm their design. It depends again on the type of project. If they are making competition for a bridge, then yes, this tool might work. But if the project is for the concert hall, I doubt it. If they are designing the monument, I doubt it. So if the structural optimization is the part of the design criteria, then the tool would work. But if the design is not measured in that sense, then it would not be used perhaps.

**Answer to question 5**

What I would be interested, instead of only including the structural optimization, is it possible to include a different layer of optimization? Can this be combined with acoustic? If it's possible, then that might be more efficient than only giving priority to the structure. I would not incorporate aesthetic into constraints. So going back to the art form and core form, I would give optimization more into the core form which is the working part of the building. Regarding the usability, architects are always interested to find the most user-friendly way. Most architects are interested in AutoCAD or some parametric tools, so if it can read this information, and get these data immediately, they would really be "pew," attracted to it. Because again, I always think in the mindset of a competition architect. A competition architect, the project is expected to be finished by three months. So this means that time is everything. So the size of office I used to work is quite small, only maximum twelve people or less. So in this case, giving one person all the job to do it would cost perhaps one day, and they may put questions marks. But if someone can send the file as they are, and if they can read the file and do it, then they would be very attracted to it. When we were working with the engineer, sometimes we got very stressful and expressed the stress because we were waiting for the simulation that they were making, and we were losing two days, and we were angry because we only had six days left. In the case of the competition, the skin is very important, and we would need to change the design based on the engineering feedback to adjust the opacity and so on. I don't know if the engineer has to remodel the architectural model. But we send them Rhino. They can use Rhino, ArchiCAD and so on. Not all of the students here are familiar with Grasshopper. I can see that all of them know about Grasshopper, but only some use it. Most undergraduates know how to use it, but from outside the school that comes here not use it. We have a course for the comprehensive studio which is required for them to graduate. In that

case, they evaluate structure using codes. But not evaluating the deflection and so on. So they are limited if they have to deal with the organic structure because they only rely on the code. The students are more focused on design.

**Answer to question 6**

I think this tool can allow architects for making first design decision depending on the project. I feel like we are making a design, and this tool becomes a confirmation tool that will give use self-confidence about our proposal. But if the engineer is part of the innovation procedure, then I have doubts about this tool, so far. If the structure is part of the art form, then I think it's better to have the engineer early in the process rather than relying on the tool. If it is more like a more norm structural method, it would be functional. For a more explorative type of project, I will still go for the engineer. Because the engineer is also inventive because they can give advice about the material that we should use for the design in the early design phase. The structural engineer is more familiar with the material system. But, I think this tool can give more coherence into the design before passing the design into the engineer. The engineer's imagination is different than the architect's. The engineers' optimal ideas are about structures; they are always thinking about structure. There was one time when we designed parking garage, need four hundred cars, we decided to use frame structure, but the engineer suggested precast structure, which is more innovative. So using this tool has to have practice, knowing the industry, knowing what is available, knowing the materials, existing production systems. The conventional architects are more cost-sensitive, and always prefer cheaper and simple way to do it than the unconventional. That experimental architect is more about the project. The conventional architect has more project, working with so many different firms, so they want the fastest way. Also, I would be more interested in multiple optimal solutions. So everything depends on how much time I should use this tool to do the optimization. It does not matter whether it uses a single or multiple platforms. Remember that my perspective comes from the competition type of architect, which is time, speed and idea are the most important for making a fast decision.

## APPENDIX F: Sample Interview Transcript D (2<sup>nd</sup> Cycle)

### **Answer to question 1**

Unfortunately, I think in the average architecture world, I think architects and engineers do not collaborate enough in a very good way. I think it has something to do with their education because they do not have the common language. So architects tend to think that engineers are their enemies instead of their collaborators. So then the architect actually designs first, then ask the engineer to join them later, instead of fixing the problem. So coming from my background, I am from Germany originally, so educations in architecture there is less artistic and more engineering oriented. So there, we were actually doing similarly to double degree in architecture and civil engineering there. So for me, some of the very few faculties here actually like to collaborate with the civil engineers. I used to set up a couple of seminars where fifty percent are architecture students, and fifty percent are engineers. So I think in the ideal world, which is not happening everywhere, is architects and engineers sit together from the very beginning. So there is an initial sketch from the architect describing what they want to do, and the collaboration starts immediately. So the structure in this way not become an after-thought, but becoming an integral element of the whole design process. Thus this is a comeback and forth process, where the engineer will comment the sketch, and eventually, it will be optimized. So in this way, the architectural product will be more exciting architectural work than the average structure. So it's not the case currently. If you are in a really good architectural firm, you normally have a good engineering firm collaborating with you. If there is a good collaboration going on, and good respect to each other, then those good things will happen. But I would say, in a whole world, not a handful; there are very few firms that are literally doing that. Some of the architecture consultants has an in-house structural engineer there, and you can see that in their works. But the average architecture firm does not do that. They are just doing their own thing, their crazy design, and then they hope that the structural engineer after to solve the structural problems, no matter how. The studio that I am teaching now, I am not a practitioner in the US, I am not a licensed architecture in the US, I stopped working as an architect when I moved to the US. But, the way we are trying to set up here, like the class I am teaching now, which is the comprehensive studio, we try to actually make structures an integral part of the project. So what we are doing now in the studio, so there are initial architectural ideas, then the students have to solve the structural problems on their own without the structural engineer. The students have to solve the structure, through the advice of the faculty who has the structural background, and so they have to start thinking how do I communicate the design to the structural engineer that they have the knowledge. Because I think the basic problem is that architects tend to think structure is not their issues, and so they do not have to deal with it. Architects are really uncomfortable to make comments about structure because they don't feel educated enough about it. The students learn a little about how to calculate the deflection of beams, but very little because there is not the time for that. Because we are squeezing the full design of the building, which is not

a small building, into one full semester, and they technically have to resolve that at the same time. So there is no way you can put those into the computer and optimize it. I have not seen that. But actually, when the collaboration comes in the comprehensive studio, with the civil engineer department, we have civil engineering students who do that for us at the same time, which is super successful. But it requires the faculties from the civil engineering to agree with our curriculum, like accepting some of their civil engineering students come here with us and follow our schedule, which is not easy. So the university environment is not set up for cross-disciplinary at research, so it's really difficult. Our curriculum here in architecture is more flexible than others, but we still cannot apply our ideas into their curriculum. So the collaboration here is limited in classroom. When I was working in Germany, when I was working in a small firm, the engineers show up in the end. But when I was working in the bigger firm, we were very technical oriented as a firm, they have the full team that has the engineer to show up at the very beginning. But you do normally, to get a job to build a building as an architect, you have to go through a competition first, and then as soon as the competition finish with the initial design, the engineer comes in. Normally, the project that won in the competition does not stay like that after winning the competition, so there has to be adapt to the reality, and more or less like start it over again. Sometimes if we have structurally challenging project in the competition, we brought structural engineers into the team earlier. But this beyond normal examples. For example, we designed an office building which is covered by huge glass waves, and somewhat produce an intermediate climate around the building. But the wave was shape in the geometry of an airplane's wing, so that natural undulation was forced through that building, and due to the fact that we do not want to have a bulky structure there, the structural engineer has to resolve that. But this is when you have an architect that is particularly interested in good structures, that really want to bring the project into other level by bringing expertise from another discipline. If it's simple and conventional, no one brings structurally very early. But as soon as we get out from something conventional, we bring structural engineer earlier. I think bringing engineer early into the design phase increase the innovation. But that depends also on the structural engineer, like some of them are curious and participate in the adventure, and some of them just want to make it simple. So it depends on the mentality of the structural engineer. If the structural engineer is actually willing to design, and do something new, and push their discipline as much as the architect, then it will be innovative. But if they only want to do something normal, then it would not be innovative anymore. When I was working in the consultant, engineers that would do the structural analysis. We would give the engineers the idea like "why don't you do something like this or that?". The engineers can say either "this is too crazy, we cannot do that" or "let's go and do it, let's make a simulation of that". Then the engineer might come next week after the simulation and said that its working. So that's more or less it worked. Regarding the payment, when we do competition together, we already know that we got the amount x for the work that we already invested into it. Depending on the contribution of the structural engineer, they participate in that money, so they will get their share. For the

competition, pretty much everybody just provides their time and ideas, and you don't expect to get any money for that, so really investing everything for the future project. And there from there on, it's kind of different than here in U.S., in Germany, the kind of percentage they got is clearly defined, like they will receive the same percentage amount of money you have to build for the whole project in the end. So it's a percentage share. So because the payments are fixed, one design iteration is the same as multiple design iteration and review. So that's the problem. Some of the engineer loves their work, and they don't care how much time they are investing, and its more or less like the architect. However, for some, if you can get the same amount of money for a crappy job versus a good job, you would probably just do the crappy job. So that depends on the motivation, if you would want to become a good architect or engineer, you probably, some don't go for efficiency, you would not mind investing more time. It's a strange calculation. If you were doing something outstanding, people will be interested to use you more, but if you do just average job, maybe no one wants to hire you. You can see that young architects do crazy stuffs, until somewhat they are settled, slowing down, and become more average in their works. About the model that we send to engineer, if the structural engineer is included in the competition, and require some modeling through that, they will go back and forth between their CAD files, their computer drawings. So architects work with the AutoCAD. So for the engineer, simulation is going on in the structural engineer's office, they add the information that the engineer need to add, then the architect can deal with it afterwards. So the whole part of the simulation is more like an in-house part within the structural engineering office, and its not going to be handed over to architect. The engineers will just evaluate the output of the simulation, and bring it into the CAD files of the architect, or just make sketches, and the architects that will change. So depending on how complex the building is, the architects make suggestion, and then it will be modified. Or even if its too complicated for the architect to think about structural grid or whatever, the engineer will fill the information and comes back to the architect. We use AutoCad. But here in US, most of them use Revit. In Germany, we use AutoCAD. In the construction project, the files that we sent to the engineer, if its early phase of the project, we sent AutoCad. If it's later in the project, we sent BIM model. I don't know if the structural engineer has to remodeled the model after the architect passed the CAD model. The AutoCad file, can be converted to dxf file, and then the engineer can convert it into their software as far as I remember. In the competition project, usually the work is more graphic. Actually the firm that I worked with worked with are not working with AutoCad for the competition, but they were working with Vectorworks, which is really graphic, but you can still draw dxf file from those. But because they are so graphic, points are not necessarily positioned in precise ways, so they might slightly off. So in the competition project, the model that is sent to engineer will be more organic, but less precise. So there would not be any structural performance in that model.

#### **Answer to question 2**

In the US, the students do structural analysis maybe once in their design, which is in the comprehensive studio. But in the Germany, the structural analysis is done in any project to ensure the project has to



be buildable in a sense. The architecture students in Germany do not use simulation programs. The students are familiar with the hand calculation for the structural analysis. In Germany, the architectural students do structural analysis by hand, but it uses the rule of thumb and codes. So you would not find any flying objects because there are missing columns in the students' works in Germany. They do not do hand calculation for the optimization.

**Answer to question 3**

Honestly, it's great if this structural optimization method can be used as a form-finding method in the schematic design. But it will require initial idea about structures. I think, for the optimization of a structural system, you need to have a profound knowledge about structure first. I am not so sure if most of the architects do have that knowledge. So, I am not so sure when architects design, they can convert their idea into that mathematical optimization model you showed in the video. So, they might learn that very abstractly, but it does not mean that they can apply that concept. So, having that said, I think all these tools about optimization can make a lot more sense if you already had basic knowledge about structure, or if you were someone who sneaks into the engineer classes, this tool can become a very dangerous tool, because then you are starting to rely on something that you do not know how to evaluate that. So, I think it's great as a tool because I know what you are talking about. I might not be able to normally to transfer my idea into something like that mathematical optimization model in the video. I think it's amazing. I would love to have more time to do that. But I think that is a certain danger of basic knowledge, and have someone buying into something that doesn't know how to evaluate that afterward. So in the video, when you talk about the constraint when the length of the member should not be too long or too short, that would be how far that architect could think of. So architects can deal with the understanding of the geometrically related constraints. But when we talk about the second layer, the stress constraint, then the architect would start to become crazy that what the heck is he talking about. So, once, in the whole career of the students here in Clemson, they actually take their design and try to bring it into a structural diagram, and the structural diagram talks about the forces, shear and so on. But honestly, the structural engineer would probably freak out about that, because it's the skeleton of a building with a couple of errors. The reason is that because they have such bare information, and the funny thing is that the architect does not even think about materials when they do that. They just bring it down, like a comic version of the project, and it's the skeleton. When you are talking about tension here, compression there, need reinforcement here and there, shear element there, make sure that the building not falling over this or that direction, but it's all very basic, and it does not deal with the idea of material, like if I use this material, I can deal with these forces in here or there. I think the students were introduced to the basic structural analysis method, but that does not mean that they apply those in the studio. So the transfer is not made. So I think for them, they draw the truss, but they do not know what the truss does. So if you are pointing at a truss configuration, and ask what each member is doing in this structure, they would not know. It would make sense on the diagram that the architectural students made, but it would not make sense in the drawing

they made afterward about their building. So there is a kind of like an island knowledge, they know it from the structural courses, but the application does not necessarily mean the real thing in building elements, so they do not know what each member is doing. The students are not required to do structural analysis. They would rely on their structural engineers in the end. They rely on the fact that there is someone who is going to take care about that. Sometimes the engineer and architect do not communicate well. They do not find a common language and common ground. Architects are talking about their topic, and engineer is talking about their topic. A good collaboration should be where both can find common ground, and that would require a structural engineer who knows the understanding of how to design and the architect that know to do the structure.

#### **Answer to question 4**

I think it would be nice if the architect has more clue about the tool. If they know how to use it, I think the architectural design will go into a more different direction. The critical questions are that architectural education is not just about structural design, there are many aspects. So it is tough to someone just to push one topic that much. I think we somewhat like to offer a structurally oriented studio, and for those, it would be awesome. Like we have a task to span over a certain distance and optimize it. If you had that question, without questions like where the restrooms go, where would be the entrance, what kind of envelope we would use, and so just focus on the structure, then that would be great. Normally, our task is touching so many topics, and so it's kind of hard to push into this detail. So that is the tough part. But if you could teach the students to use the tool, it would be a super interesting tool for them to use later on. The students are usually familiar with the Grasshopper. The students are not familiar with the structural analysis or optimization software. I think they use Grasshopper differently than how you guys use. Grasshopper is a form-finding tool for them. I think they know how to do that. I am from a different generation, so I lost that already. But, I think they do not know to do with the Grasshopper afterward. I am more a performance person, so I think the students are more interested in the building performance aspect than structure. I think usually architects do not do structural optimization because of the time issue. There are certain critical moments about the architect have their own field, and the engineer has their own, and usually, they tend not to like if someone is reaching out into their field. So it's like the profession that they have learned. I think it's always good to talk to them about it and understand the process. But it is not necessarily that the architect needs to do structural optimization. Imagine if I am working with the mechanical engineer, and if they tell me how to design a building, I am freaking out. So if telling the engineer how to do the structural optimization, and maybe to say that "your structure is not as optimized as mine right now," then the structural engineer will be freak out. If I knew something about structural optimization, and I talked about it with the structural engineer, I think it would raise the respect of the engineer towards me, and the engineer will feel "Okay, you know what you are talking about, that is good." It is always an improvement to have a common ground.

#### **Answer to question 5**

About the tool, honestly, if you know Grasshopper, I think it would be easier for someone to follow that. The moment you start to put the variables and that stuff, you are losing someone who does not have knowledge about that. So when you are starting to fill something that relates to formulas and numbers, I think that where the people are just getting lost. I think I would optimize the weight of the structure out of all. I would prefer a single optimum because it is simpler. If I have a machine do something for me, then I don't want to look at those and see so many options. I just want to look that this is the optimized one for me. I think it's okay to use multiple software. We have much software where we have to use plugins and so on. I think it's better to have two disciplines coming in the tool with their own software because I think each discipline is optimizing their own software specifically. If there was only one that is working, you are working with the state of the art of one point. So I think different software from the engineer and architect are probably will be developed further and further and they will be still tied into your developed system. So I think it would be easier to have one software that does everything. But I think the performance would be better if to keep two or more software from different disciplines. Because of its discipline-specific. Grasshopper is made for architects, and the Abaqus is made for an engineer.

**Answer to question 6**

I think this tool should not postpone the engineer involvement because I think no matter how much you are simulating, as an architect, you still would not think like an engineer. I think the machine cannot take over the engineer as a person. So for me "no." This tool can be useful if the architect understands the output. I think the architect and the engineer should use this tool when they are collaborating. You cannot do a good structure without a structural engineer. So that is the common ground. I like the tool as it raises awareness, but the collaboration should happen as early as possible.

## APPENDIX G: iMAGv1 Source Code

### **iMAGv1 Grasshopper send output:**

```
Private Sub RunScript(ByVal coordinates As List(Of Object), ByVal
connectivities As List(Of Object), ByVal index_load As List(Of String),
ByVal magnitude_load As Object, ByVal num_members As String, ByVal
text_FilePath As String, ByVal txt_OuterRadiusPipe As String, ByVal
input_BC As List(Of Object), ByRef A As Object, ByRef b As Object)
    Dim delim As String
    Dim i As Integer
    Dim startBC As Integer
    Dim startLoad As Integer
    Dim Crossarea As Double
    delim = ","
    Dim s As String
    s = ""
    s += "*** PARTS" + Constants.vbNewLine
    s += "***" + Constants.vbNewLine
    s += "*Node" + Constants.vbNewLine
    For i = 0 To coordinates.Count - 1
        If i = (coordinates.Count - 1) Then
            delim = ""
        End If
        s += coordinates(i).tostring() + delim
    Next
    startLoad = 5
    For i = 0 To index_load.count - 1
        startLoad = startLoad + 1
        s += "*Nset, nset = _PickedSet" + startLoad.ToString() +
Constants.vbNewLine
        s += index_load(i) + Constants.vbNewLine
    Next
    startBC = 5 + index_load.count
    For i = 0 To input_BC.count - 1
        startBC = startBC + 1
        s += "*Nset, nset = _PickedSet" + startBC.ToString() +
Constants.vbNewLine
        s += input_BC(i) + Constants.vbNewLine
    Next
    s += "*" + "Element, type=T3D2, elset=frame" + Constants.vbNewLine
    For i = 0 To connectivities.Count - 1
        If i = (connectivities.Count - 1) Then
            delim = ""
            ' s += Constants.vbNewLine
        End If
        s += connectivities(i).tostring() + delim
    Next
    s += "*** Section: Section-1 Profile: Profile-1" +
Constants.vbNewLine
```

```

s += "*Solid Section, elset = frame, material=Material - 1" +
Constants.vbNewLine
Crossarea = 3.14159 * txt_OuterRadiusPipe ^ 2
s += Crossarea.ToString() + Constants.vbNewLine
s += "***" + Constants.vbNewLine
s += "***MATERIALS" + Constants.vbNewLine
s += "***" + Constants.vbNewLine
s += "*Material, name = Material - 1" + Constants.vbNewLine
s += "*Elastic" + Constants.vbNewLine
s += "2.1e+11, 0.26" + Constants.vbNewLine
s += "*** -----"
-----" + Constants.vbNewLine
s += "***" + Constants.vbNewLine
s += "*** STEP: Step-1" + Constants.vbNewLine
s += "***" + Constants.vbNewLine
s += "*Step, name = -1" + Constants.vbNewLine
s += "*Static" + Constants.vbNewLine
s += "1., 1., 1e-05, 1." + Constants.vbNewLine
s += "***" + Constants.vbNewLine
s += "*** BOUNDARY CONDITIONS" + Constants.vbNewLine
s += "*** " + Constants.vbNewLine
startBC = 5 + index_load.count
For i = 0 To input_BC.count - 1
    startBC = startBC + 1
    s += "*** Name: BC-1 Type: Symmetry/Antisymmetry/Encastre" +
Constants.vbNewLine
    s += "*Boundary" + Constants.vbNewLine
    s += "_PickedSet" + startBC.ToString() + ", ENCASTRE" +
Constants.vbNewLine
    s += "***" + Constants.vbNewLine
Next
s += "***LOADS" + Constants.vbNewLine
s += "***" + Constants.vbNewLine
startLoad = 5
For i = 0 To index_load.count - 1
    startLoad = startLoad + 1
    s += "***Name: Load-1 Type: Concentrated force" +
Constants.vbNewLine
    s += "*Cload" + Constants.vbNewLine
    s += "_PickedSet" + startLoad.ToString() + " ,3, " +
magnitude_load + Constants.vbNewLine
    s += "***" + Constants.vbNewLine
Next
s += "***OUTPUT REQUESTS" + Constants.vbNewLine
s += "*** " + Constants.vbNewLine
s += "*NODE PRINT" + Constants.vbNewLine
s += "U" + Constants.vbNewLine
s += "*EL PRINT" + Constants.vbNewLine
s += "MISES" + Constants.vbNewLine
s += "*Restart, write, frequency = 0" + Constants.vbNewLine
s += "***" + Constants.vbNewLine
s += "*** FIELD OUTPUT: F-Output-1" + Constants.vbNewLine
s += "***" + Constants.vbNewLine
s += "*Output, field, variable = PRESELECT" + Constants.vbNewLine

```

```

s += "*** " + Constants.vbNewLine
s += "*** HISTORY OUTPUT: H-Output-1" + Constants.vbNewLine
s += "***" + Constants.vbNewLine
s += "*Output, history, variable = PRESELECT" + Constants.vbNewLine
s += "*End Step" + Constants.vbNewLine
Dim g As String
g = ""
g += text_FilePath
g += "\FEAinput.inp"
file.WriteAllText(g, s)
a = s

```

#### **iMAGv1 GUI:**

```

function varargout = iMAG_GUI(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @iMAG_GUI_OpeningFcn, ...
                  'gui_OutputFcn',  @iMAG_GUI_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function iMAG_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = iMAG_GUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function s1min_Callback(hObject, eventdata, handles)
function s1min_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function s2min_Callback(hObject, eventdata, handles)
function s2min_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function s3min_Callback(hObject, eventdata, handles)
function s3min_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

```

function s1max_Callback(hObject, eventdata, handles)
function s1max_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function s2max_Callback(hObject, eventdata, handles)
function s2max_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function s3max_Callback(hObject, eventdata, handles)
function s3max_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function runIterative_Callback(hObject, eventdata, handles)
gets1min=str2num(get(handles.s1min,'String'));
gets2min=str2num(get(handles.s2min,'String'));
gets3min=str2num(get(handles.s3min,'String'));
gets1max=str2num(get(handles.s1max,'String'));
gets2max=str2num(get(handles.s2max,'String'));
gets3max=str2num(get(handles.s3max,'String'));
getnumLevels1=str2num(get(handles.numLevels1,'String'));
getnumLevels2=str2num(get(handles.numLevels2,'String'));
getnumLevels3=str2num(get(handles.numLevels3,'String'));
set(handles.processing_indicator,'String','currently running the
analysis.....please wait....')
start_clock=fix(clock);
set(handles.start_time, 'String', num2str(start_clock))
func_TessCurve_ExhaustiveOPT_Feb2016(gets1min,gets2min,gets3min,gets1ma
x,gets2max,gets3max,getnumLevels1,getnumLevels2,getnumLevels3)
set(handles.processing_indicator,'String','iterative analysis
finished')
end_clock=fix(clock);
set(handles.end_time, 'String', num2str(end_clock))
current_folder=pwd
file_IterativeResult='\SFTessDOUBLECurveResult.txt'
directory_IterativeResult=strcat(current_folder,file_IterativeResult)
M=dlmread(directory_IterativeResult)
N=zeros(size(M,1),6);
N(1:size(M,1),1:6)=M(:,4:9)
s1_R=N(1:size(N,1),1)
s2_R=N(1:size(N,1),2)
s3_R=N(1:size(N,1),3)
disp_R=N(1:size(N,1),4)
stress_R=N(1:size(N,1),5)
totallength_R=N(1:size(N,1),6)
set(handles.result_s1,'String',num2str(s1_R));
set(handles.result_s2,'String',num2str(s2_R));

```

```

set(handles.result_s3, 'String', num2str(s3_R));
set(handles.result_disp, 'String', num2str(dispatch_R));
set(handles.result_stress, 'String', num2str(stress_R));
set(handles.result_totallength, 'String', num2str(totallength_R));
function x1_Callback(hObject, eventdata, handles)
function x1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function x2_Callback(hObject, eventdata, handles)
function x2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function x3_Callback(hObject, eventdata, handles)
function x3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function LB_x1_Callback(hObject, eventdata, handles)
function LB_x1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function LB_x2_Callback(hObject, eventdata, handles)
function LB_x2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function LB_x3_Callback(hObject, eventdata, handles)
function LB_x3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function UB_x1_Callback(hObject, eventdata, handles)
function UB_x1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function UB_x2_Callback(hObject, eventdata, handles)
function UB_x2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
end

```



```

function UB_x3_Callback(hObject, eventdata, handles)
function UB_x3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function d_w_Callback(hObject, eventdata, handles)
function d_w_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function run_optimization_Callback(hObject, eventdata, handles)
gets1min=str2num(get(handles.s1min,'String'));
gets2min=str2num(get(handles.s2min,'String'));
gets3min=str2num(get(handles.s3min,'String'));
gets1max=str2num(get(handles.s1max,'String'));
gets2max=str2num(get(handles.s2max,'String'));
gets3max=str2num(get(handles.s3max,'String'));
initialx1=gets1min%(gets1min+gets1max)/2
initialx2=gets2min%(gets2min+gets2max)/2
initialx3=gets3min%(gets3min+gets3max)/2
getdw=str2num(get(handles.d_w,'String'));
gettw=1-getdw;
set(handles.text84,'String',num2str(gettw));
getobj1=get(handles.obj1,'String');
getobj2=get(handles.obj2,'String');
getconstraint_exp=get(handles.constraint_exp,'String');
getvar_ce=get(handles.var_constraintexp,'String');
set(handles.processing_indicator2,'String','currently running the
optimization.....please wait....')
[x_opt,f_opt]=func_main01_cosine_nix(initialx1,initialx2,initialx3,gets
1min,gets2min,gets3min,gets1max,gets2max,gets3max,getdw,getobj1,getobj2
,getconstraint_exp,getvar_ce)
set(handles.processing_indicator2,'String','optimization done')
xx=x_opt(1)
params(1)=getdw;
params(2)=2;
output1=objfun_cosine_interpolation(x_opt,params,getobj1,getobj2,getcon
straint_exp,getvar_ce);
disp(['displacement = ',num2str(output1(1))])
disp(['weight = ',num2str(output1(2))])
[C,Ceq] =
confun_cosine_interpolation(x_opt,params,getobj1,getobj2,getconstraint_
exp,getvar_ce);
disp(['stress = ',num2str(C)])
set(handles.opt_s1,'String',num2str(x_opt(1)));
set(handles.opt_s2,'String',num2str(x_opt(2)));
set(handles.opt_s3,'String',num2str(x_opt(3)));
set(handles.opt_disp,'String',num2str(output1(1)));
set(handles.opt_stress,'String',num2str(C));
set(handles.opt_totallength,'String',num2str(output1(2)));

```

```

function s_1_Callback(hObject, eventdata, handles)
function s_1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function s_2_Callback(hObject, eventdata, handles)
function s_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function s_3_Callback(hObject, eventdata, handles)
function s_3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function RunOneAnalysis_Callback(hObject, eventdata, handles)
gets_1=str2num(get(handles.s_1,'String'));
gets_2=str2num(get(handles.s_2,'String'));
gets_3=str2num(get(handles.s_3,'String'));
[maxdisp,maxstress] =
SFDOUBLECurveTess_analysis_3_OBJFUNexhaustive(gets_1,gets_2,gets_3)
total_length=ComputeWeight01();
cross_area=pi*gets_1*gets_1;
material_density=8050;
elem_weight=total_length*cross_area*material_density;
set(handles.onetime_disp,'String',num2str(maxdisp));
set(handles.onetime_stress,'String',num2str(maxstress/1e6));
set(handles.onetime_TotalLength,'String',num2str(elem_weight/1e3));
function numLevels1_Callback(hObject, eventdata, handles)
function numLevels1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function numLevels2_Callback(hObject, eventdata, handles)
function numLevels2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function numLevels3_Callback(hObject, eventdata, handles)
function numLevels3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function obj1_Callback(hObject, eventdata, handles)
function obj1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');

```

```

end
function constraint_exp_Callback(hObject, eventdata, handles)
function constraint_exp_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function obj2_Callback(hObject, eventdata, handles)
function obj2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function var_constraintexp_Callback(hObject, eventdata, handles)
function var_constraintexp_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

#### **iMAGv1 check LCK file:**

```

function checkLCK(filename)
done = 0;
flag = 0;
while ~done
    lck = dir([filename, '.lck']);
    if flag == 0
        if ~isempty(lck)    % lck file was created
            flag = 1;
        end
    else
        if isempty(lck)    % lck file was removed
            done = 1;
        end
    end
    end
    pause(2)
end
end

```

#### **iMAGv1 compute weight:**

```

function weight= ComputeWeight01()
filename = 'FEAinput.inp';
fid = fopen(filename,'r');
done = 0;
start = 0;
nodes = [];
while ~done
    tline = fgetl(fid);
    if ~ischar(tline)
        done = 1;
    end
end

```

```

if ~done
    if length(tline) == 5
        if tline(1) == '*'
            if tline(1:5) == '*Node'
                start = 1;
            end
        end
    end
    if start == 1
        nodes = [nodes;str2num(tline)];
    end
end
if ~isempty(nodes)
    if start == 1
        if tline(1) == '*'
            done = 1;
        end
    end
end
end
fclose(fid);
filename = 'FEAinput.inp';
fid = fopen(filename,'r');
done = 0;
start = 0;
elems = [];
while ~done
    tline = fgetl(fid);
    if ~ischar(tline)
        done = 1;
    end
    if ~done
        if length(tline) >= 5
            if tline(1) == '*'
                if tline(1:5) == '*Elem'
                    start = 1;
                end
            end
        end
        if start == 1
            elems = [elems;str2num(tline)];
        end
    end
    if ~isempty(elems)
        if start == 1
            if tline(1) == '*'
                done = 1;
            end
        end
    end
end
end
fclose(fid);
nodes;
elems;

```

```

Nnodes = size(nodes,1);
Nelems = size(elems,1);
total_length = 0;
for i=1:Nelems
    found = 0;
    j = 0;
    while ~found
        j = j+1;
        if elems(i,2) == nodes(j,1)
            found = 1;
            ifound1 = j;
        end
    end
    found = 0;
    j = 0;
    while ~found
        j = j+1;
        if elems(i,3) == nodes(j,1)
            found = 1;
            ifound2 = j;
        end
    end
    coords1 = nodes(ifound1,2:4);
    coords2 = nodes(ifound2,2:4);
    vector = coords2-coords1;
    total_length = total_length + norm(vector);
end
total_length
weight=total_length

```

**IMAGv1 constraint function:**

```

function [C,Ceq] =
confun_cosine_interpolation(x,params,obj1,obj2,constraint_expression,va
r_ce)
cosine_interpolation;
d=y(1)
s=y(2)
w=y(3)
C2=y(1)-0.01;
var=strsplit(var_ce,',')
var_join=strjoin(var,',')
E=inline(constraint_expression)
var_size=size(var,2)
for i=1:var_size
    if var{1,i}=='s1'
        ar{i,1}=x(1)
    elseif var{1,i}=='s2'
        ar{i,1}=x(2)
    elseif var{1,i}=='s3'
        ar{i,1}=x(3)
    elseif var{1,i}=='d'
        ar{i,1}=d
    end
end

```

```

        elseif var{1,i}=='s'
            ar{i,1}=s
        elseif var{1,i}=='w'
            ar{i,1}=w
        end
    end
end
art=transpose(ar)
C=[E(art{:}) C2]
Ceq=[];
if params(2)==2
    C=s;
End
End

```

**iMAGv1 calling cosine interpolation:**

```

filename = 'SFTessDOUBLECurveResult.txt';
fid = fopen(filename,'r');
done = 0;
store_data = [];
while ~done
    tline = fgetl(fid);
    if ~ischar(tline)
        done = 1;
    end
    if ~done
        store_data = [store_data;str2num(tline)];
    end
end
fclose(fid);
S = store_data(:,4:6);
d = store_data(:,7);
sigma = store_data(:,8)
weight = store_data(:,9)
stress=sigma
y = interpolate3DCosine01(x,S,d,sigma,weight);

```

**iMAGv1 cosine interpolation:**

```

function y=interpolate3DCosine01(x,S,d,sigma,weight)
N = size(S,1);
Smin = min(S);
Smax = max(S);
levels1 = [];
for i=1:N
    found = 0;
    for j=1:size(levels1,1)
        if S(i,1) == levels1(j)
            found = 1;
        end
    end
    if ~found
        levels1 = [levels1;S(i,1)];
    end
end

```

```

end
levels2 = [];
for i=1:N
    found = 0;
    for j=1:size(levels2,1)
        if S(i,2) == levels2(j)
            found = 1;
        end
    end
    if ~found
        levels2 = [levels2;S(i,2)]; [levels1;S(i,1)];
    end
end
levels3 = [];
for i=1:N
    found = 0;
    for j=1:size(levels3,1)
        if S(i,3) == levels3(j)
            found = 1;
        end
    end
    if ~found
        levels3 = [levels3;S(i,3)];
    end
end
Nlevels(1) = size(levels1,1);
Nlevels(2) = size(levels2,1);
Nlevels(3) = size(levels3,1);
sla = Smin(1);
done = 0;
for i=2:Nlevels(1)
    if ~done
        if x(1) < levels1(i)
            if i>2; slz = levels1(i-2); end;
            sla = levels1(i-1);
            done = 1;
        end
    end
end
if x(1) == levels1(Nlevels(1))
    slz = levels1(Nlevels(1)-1);
    sla = levels1(Nlevels(1)-1);
end
s1b = Smax(1);
done = 0;
for i=2:Nlevels(1)
    if ~done
        if x(1) >= levels1(i-1,1) & x(1) < levels1(i,1)
            s1b = levels1(i,1);
            if i<Nlevels(1); s1c = levels1(i+1); end;
            done = 1;
        end
    end
end
end

```

```

if x(1) == levels1(Nlevels(1))
    s1b = levels1(Nlevels(1));
    s1c = levels1(Nlevels(1));
end
s2a = Smin(2);
done = 0;
for i=2:Nlevels(2)
    if ~done
        if x(2) < levels2(i)
            if i>2; s2z = levels2(i-2); end;
            s2a = levels2(i-1);
            done = 1;
        end
    end
end
if x(2) == levels2(Nlevels(2))
    s2z = levels2(Nlevels(2)-1);
    s2a = levels2(Nlevels(2)-1);
end
s2b = Smax(2);
done = 0;
for i=2:Nlevels(2)
    if ~done
        if x(2) >= levels2(i-1,1) & x(2) < levels2(i,1)
            s2b = levels2(i,1);
            if i<Nlevels(2); s2c = levels2(i+1); end;
            done = 1;
        end
    end
end
if x(2) == levels2(Nlevels(2))
    s2b = levels2(Nlevels(2));
    s2c = levels2(Nlevels(2));
end
s3a = Smin(3);
done = 0;
for i=2:Nlevels(3)
    if ~done
        if x(3) < levels3(i)
            if i>2; s3z = levels3(i-2); end;
            s3a = levels3(i-1);
            done = 1;
        end
    end
end
if x(3) == levels3(Nlevels(3))
    s3z = levels3(Nlevels(3)-1);
    s3a = levels3(Nlevels(3)-1);
end
s3b = Smax(3);
done = 0;
for i=3:Nlevels(3)
    if ~done
        if x(3) >= levels3(i-1,1) & x(3) < levels3(i,1)

```



```

        s3b = levels3(i,1);
        if i<Nlevels(3); s3c = levels3(i+1); end;
        done = 1;
    end
end
end
if x(3) == levels3(Nlevels(3))
    s3b = levels3(Nlevels(3));
    s3c = levels3(Nlevels(3));
end
for i=1:N
    if S(i,1)==s1a & S(i,2)==s2a & S(i,3)==s3a
        d_aaa = d(i);
        sigma_aaa = sigma(i);
        weight_aaa=weight(i);
    end
    if S(i,1)==s1b & S(i,2)==s2a & S(i,3)==s3a
        d_baa = d(i);
        sigma_baa = sigma(i);
        weight_baa=weight(i);
    end
    if S(i,1)==s1a & S(i,2)==s2b & S(i,3)==s3a
        d_aba = d(i);
        sigma_aba = sigma(i);
        weight_aba=weight(i);
    end
    if S(i,1)==s1b & S(i,2)==s2b & S(i,3)==s3a
        d_bba = d(i);
        sigma_bba = sigma(i);
        weight_bba=weight(i);
    end
    if S(i,1)==s1a & S(i,2)==s2a & S(i,3)==s3b
        d_aab = d(i);
        sigma_aab = sigma(i);
        weight_aab=weight(i);
    end
    if S(i,1)==s1b & S(i,2)==s2a & S(i,3)==s3b
        d_bab = d(i);
        sigma_bab = sigma(i);
        weight_bab=weight(i);
    end
    if S(i,1)==s1a & S(i,2)==s2b & S(i,3)==s3b
        d_abb = d(i);
        sigma_abb = sigma(i);
        weight_abb=weight(i);
    end
    if S(i,1)==s1b & S(i,2)==s2b & S(i,3)==s3b
        d_bbb = d(i);
        sigma_bbb = sigma(i);
        weight_bbb=weight(i);
    end
end
end
% s1a <= x(1) < s1b at s2a and s3a
mu = (x(1)-s1a)/(s1b-s1a);

```

```

mu2 = (1-cos(mu*3.14159))/2;
y1 = d_aaa;
y2 = d_baa;
d_1aa = y1*(1-mu2)+y2*mu2;
y1 = sigma_aaa;
y2 = sigma_baa;
sigma_1aa = y1*(1-mu2)+y2*mu2;
y1 = weight_aaa;
y2 = weight_baa;
weight_1aa = y1*(1-mu2)+y2*mu2;
% sla <= x(1) < slb at s2b and s3a
mu = (x(1)-sla)/(slb-sla);
mu2 = (1-cos(mu*3.14159))/2;
y1 = d_aba;
y2 = d_bba;
d_1ba = y1*(1-mu2)+y2*mu2;
y1 = sigma_aba;
y2 = sigma_bba;
sigma_1ba = y1*(1-mu2)+y2*mu2;
y1 = weight_aba;
y2 = weight_bba;
weight_1ba = y1*(1-mu2)+y2*mu2;
% sla <= x(1) < slb at s2a and s3b
mu = (x(1)-sla)/(slb-sla);
mu2 = (1-cos(mu*3.14159))/2;
y1 = d_aab;
y2 = d_bab;
d_1ab = y1*(1-mu2)+y2*mu2;
y1 = sigma_aab;
y2 = sigma_bab;
sigma_1ab = y1*(1-mu2)+y2*mu2;
y1 = weight_aab;
y2 = weight_bab;
weight_1ab = y1*(1-mu2)+y2*mu2;
% sla <= x(1) < slb at s2b and s3b
mu = (x(1)-sla)/(slb-sla);
mu2 = (1-cos(mu*3.14159))/2;
y1 = d_abb;
y2 = d_bbb;
d_1bb = y1*(1-mu2)+y2*mu2;
y1 = sigma_abb;
y2 = sigma_bbb;
sigma_1bb = y1*(1-mu2)+y2*mu2;
y1 = weight_abb;
y2 = weight_bbb;
weight_1bb = y1*(1-mu2)+y2*mu2;
% x(1) s2a <= x(2) < s2b and s3a
mu = (x(2)-s2a)/(s2b-s2a);
mu2 = (1-cos(mu*3.14159))/2;
y1 = d_1aa;
y2 = d_1ba;
d_2a = y1*(1-mu2)+y2*mu2;
y1 = sigma_1aa;
y2 = sigma_1ba;

```

```

sigma_2a = y1*(1-mu2)+y2*mu2;
y1 = weight_1aa;
y2 = weight_1ba;
weight_2a = y1*(1-mu2)+y2*mu2;
% x(1) s2a <= x(2) < s2b and s3b
mu = (x(2)-s2a)/(s2b-s2a);
mu2 = (1-cos(mu*3.14159))/2;
y1 = d_1ab;
y2 = d_1bb;
d_2b = y1*(1-mu2)+y2*mu2;
y1 = sigma_1ab;
y2 = sigma_1bb;
sigma_2b = y1*(1-mu2)+y2*mu2;
y1 = weight_1ab;
y2 = weight_1bb;
weight_2b = y1*(1-mu2)+y2*mu2;
% x(1) x(2) s3a <= x(3) < s3b
mu = (x(3)-s3a)/(s3b-s3a);
mu2 = (1-cos(mu*3.14159))/2;
y1 = d_2a;
y2 = d_2b;
d_3 = y1*(1-mu2)+y2*mu2;
y1 = sigma_2a;
y2 = sigma_2b;
sigma_3 = y1*(1-mu2)+y2*mu2;
y1 = weight_2a;
y2 = weight_2b;
weight_3 = y1*(1-mu2)+y2*mu2;
y = [d_3,sigma_3,weight_3];

```

**iMAGv1 objective function:**

```

function objfun =
objfun_cosine_interpolation(x,params,obj1,obj2,constraint_expression,va
r_ce)
cosine_interpolation;
dw=params(1,1);
ww=1-dw
if obj1=='d'
    interpol_obj1=y(1)
    obj1max=max(d)%
    obj1min=min(d)%
elseif obj1=='s'
    interpol_obj1=y(2)
    obj1max=max(stress)%
    obj1min=min(stress)%
elseif obj1=='w'
    interpol_obj1=y(3)
    obj1max=max(weight)%
    obj1min=min(weight)%
end
if obj2=='d'
    interpol_obj2=y(1)

```

```

        obj2max=max(d)%
        obj2min=min(d)%
elseif obj2=='s'
    interpol_obj2=y(2)
    obj2max=max(stress)%
    obj2min=min(stress)%
elseif obj2=='w'
    interpol_obj2=y(3)
    obj2max=max(weight)%
    obj2min=min(weight)%
end
if params(2)==1
    objfun=dw*((interpol_obj1-obj1min)/(obj1max-
obj1min))+ww*((interpol_obj2-obj2min)/(obj2max-obj2min));
else
    objfun=[interpol_obj1,interpol_obj2];
end

```

**iMAGv1 main optimization function:**

```

function
[x_optim,f_optim]=func_main01_cosine_nix(x1,x2,x3,LB_x1,LB_x2,LB_x3,UB_
x1,UB_x2,UB_x3,getdw,obj1,obj2,constraint_expression,var_ce)
x = [x1, x2, x3]
LB = [LB_x1;LB_x2;LB_x3];
UB = [UB_x1;UB_x2;UB_x3];
FUN = 'objfun_cosine_interpolation'; %
NONLCON = 'confun_cosine_interpolation'; %
options=[];
params(1)=getdw;
params(2)=1;
[x_optim,f_optim] =
fmincon(FUN,x,[],[],[],[],LB,UB,NONLCON,options,params,obj1,obj2,constr
aint_expression,var_ce)
params(2)=2;
output1=objfun_cosine_interpolation(x_optim,params,obj1,obj2,constraint
_expression,var_ce);
disp(['displacement = ',num2str(output1(1))])
disp(['weight = ',num2str(output1(2))])
[C,Ceq] =
confun_cosine_interpolation(x_optim,params,obj1,obj2,constraint_express
ion,var_ce);
disp(['stress = ',num2str(C)])

```

**iMAGv1 full factorial analysis:**

```

function
func_TessCurve_ExhaustiveOPT_Feb2016(s1min,s2min,s3min,s1max,s2max,s3ma
x,numLVs1,numLVs2,numLVs3)
s(1)=s1min;
s(2)=s2min;
s(3)=s3min;
N = 1;

```

```

store_data = [];
disp_vector=[];
lengths1=s1max-s1min
lengths2=s2max-s2min
lengths3=s3max-s3min
if numLVs1==0
    ds1=0
    numLVs1=1
elseif numLVs1==1
    r=0
    ds1=lengths1/(numLVs1-r)
else
    r=1
    ds1=lengths1/(numLVs1-r)
end
if numLVs2==0
    ds2=0
    numLVs2=1
elseif numLVs2==1
    r=0
    ds2=lengths2/(numLVs2-r)
else
    r=1
    ds2=lengths2/(numLVs2-r)
end
if numLVs3==0
    ds3=0
    numLVs3=1
elseif numLVs3==1
    r=0
    ds3=lengths3/(numLVs3-r)
else
    r=1
    ds3=lengths3/(numLVs3-r)
end
for i1=1:numLVs1%8
    s(2) = s2min;
    for j2=1:numLVs2%8
        s(3) = s3min;
        for k3=1:numLVs3%8
            c = clock;
            [maxdisp,maxstress] =
SFDOUBLECurveTess_analysis_3_OBJFUNexhaustive(s(1),s(2),s(3));
            total_length=ComputeWeight01();
            cross_area=pi*s(1)*s(1);
            material_density=8050;
            elem_weight=total_length*cross_area*material_density;
            disp([c(4:6),s,maxdisp,maxstress,elem_weight]);
            store_data =
[store_data;c(4:6),s,maxdisp,maxstress,elem_weight];
            disp_vector=[s,maxdisp, elem_weight/1e3];
            [M,I] = min(disp_vector(:,4)) ;
            best_permutation=disp_vector(I,:);
            s = s+[0,0,ds3];

```

```

        end
        s = s+[0,ds2,0];
    end
    s = s+[ds1,0,0];
end
store_data
fid=fopen('SFTessDOUBLECurveResult.txt','w');
for i = 1:size(store_data,1)
fprintf(fid,'%s\n',num2str([store_data(i,1:7),store_data(i,8)/1e6,store
_data(i,9)/1e3 ]))
end
fclose(fid);

```

#### **iMAGv1 call Abaqus:**

```

function [maxdisp,maxstress] =
SFDOUBLECurveTess_analysis_3_OBJFUNexhaustive(s1,s2,s3)
g(1)=s1
g(2)=s2
g(3)=s3
GHinput=[g(1);g(2);g(3)]
csvwrite('SFDOUBLECurveTessInitialValues.csv', GHinput);
pause(5);
done=0;
while ~done
    listoffiles = dir;
    for i=1:size(listoffiles,1)
        if size(listoffiles(i).name,2) == 12
            if listoffiles(i).name == 'FEAinput.inp'
                done = 1;
            end
        end
    end
end
end
dos('abaqus job=FEAinput');
disp('Abaqus is running');
checkLCK('FEAinput');
disp('Abaqus is done');
[maxdisp,maxstress] = readoutput('FEAinput')
disp(['    'Maximum displacement = ',num2str(maxdisp),' m'])
disp(['    'Maximum stress = ',num2str(maxstress),' MPa'])
y=maxdisp;

```

#### **iMAGv1 read output from Abaqus:**

```

function [maxdisp,maxstress] = readoutput(filename)
fid = fopen([filename,'.dat'],'r');
done = 0;
store_maxlines = [];
for i=1:200
    store_maxlines = [store_maxlines,' '];
end
while ~done

```

```

tline = fgetl(fid);
if ~ischar(tline)
    done = 1;
else
    if length(tline) > 22
        if tline(1:18) == '    ELEMENT  PT FO'
            tline = fgetl(fid);
            tline = fgetl(fid);
            done2 = 0;
            max_mises = 0;
            while ~done2
                tline = fgetl(fid);
                if length(tline) == 0
                    done2 = 1;
                else
                    mises = str2num(tline);
                    if mises(3) > max_mises
                        max_mises = mises(3);
                        node_mises = mises(1);
                    end
                end
            end
            end
            maxline1 = ['S ', num2str([node_mises, max_mises])];
            for i=1:200-length(maxline1)
                maxline1 = [maxline1, ' '];
            end
            store_maxlines = [store_maxlines; maxline1];
        end
    end
    if length(tline) > 22
        if tline(1:22) == '    NODE FOOT-  U1'
            tline = fgetl(fid);
            tline = fgetl(fid);
            done2 = 0;
            max_amplit = 0;
            while ~done2
                tline = fgetl(fid);
                if length(tline) == 0
                    done2 = 1;
                else
                    amplit1 = str2num(tline);
                    amplit2 = norm(amplit1(2:4));
                    if amplit2 > max_amplit
                        max_amplit = amplit2;
                        node_amplit = amplit1(1);
                    end
                end
            end
            end
            maxline1 = ['U ', num2str([node_amplit, max_amplit])];
            for i=1:200-length(maxline1)
                maxline1 = [maxline1, ' '];
            end
            store_maxlines = [store_maxlines; maxline1];
        end
    end
end

```

```

        end
    end
end
fclose(fid);
store_maxlines
maxstress = 0;
maxdisp = 0;
for i=1:size(store_maxlines,1)
    if store_maxlines(i,1) == 'S'
        line_num = str2num(store_maxlines(i,2:size(store_maxlines,2)));
        if line_num(2) > maxstress
            maxstress = line_num(2);
        end
    end
    if store_maxlines(i,1) == 'U'
        line_num = str2num(store_maxlines(i,2:size(store_maxlines,2)));
        if line_num(2) > maxdisp
            maxdisp = line_num(2);
        end
    end
end
end
end

```



## APPENDIX H: iMAGv2 Source Code

iMAGv2 uses the following iMAGv1's files: (1) check LCK, (2) compute weight, (3) cosine interpolation, (4) full factorial analysis, (5) call Abaqus, and (6) read output from Abaqus.

**iMAGv2 performs discretization and filtering process:**

```
S = data1(:,1:3);
d = data1(:,4);
sigma = data1(:,5);
weight = data1(:,6);
N = size(S,1);
Smins = min(S);
Smaxs = max(S);
levels1 = [];
for i=1:N
    found = 0;
    for j=1:size(levels1,1)
        if S(i,1) == levels1(j)
            found = 1;
        end
    end
    if ~found
        levels1 = [levels1;S(i,1)];
    end
end
levels2 = [];
for i=1:N
    found = 0;
    for j=1:size(levels2,1)
        if S(i,2) == levels2(j)
            found = 1;
        end
    end
    if ~found
        levels2 = [levels2;S(i,2)];
    end
end
levels3 = [];
for i=1:N
    found = 0;
    for j=1:size(levels3,1)
        if S(i,3) == levels3(j)
            found = 1;
        end
    end
    if ~found
        levels3 = [levels3;S(i,3)];
    end
end
Nlevels(1) = size(levels1,1);
Nlevels(2) = size(levels2,1);
Nlevels(3) = size(levels3,1);
Nlevels
sa = Smins;
sb = Smaxs;
M = 100*max(Nlevels);
deltas = (sb-sa)/M;
```

```

x = sa;
weight_min = 1e10;
for i3=1:M+1
    x(2) = sa(2);
    for i2=1:M+1
        x(1) = sa(1);
        for i1=1:M+1
            y=interpolate3DCosine01(x,S,d,sigma,weight);
            if y(3) < weight_min
                if y(2) <= 250
                    if y(1) <= 0.01
                        i1_min = i;
                        weight_min = y(3);
                        x_star = x;
                    end
                end
            end
            x(1) = x(1)+deltas(1);
        end
        x(2) = x(2)+deltas(2);
    end
    disp(num2str([x,y,weight_min]))
    x(3) = x(3)+deltas(3);
end
end

```

## APPENDIX I: IMAGv3 Source Code

iMAGv3 uses the following iMAGv1's files: (1) check LCK, (2) compute weight, and (3) read output from Abaqus.

**iMAGv3 constraint function:**

```
function [C,Ceq] = nix_confun3(s)
global confun_calculation
global objfun_calculation
dos('del FEAinput.inp');
csvwrite('FEAinput.csv', s');
done = 0;
while ~done
    lls = ls;
    pause(0.1)
    for i=1:size(lls,1)
        if lls(i,1) == 'F'
            if lls(i,2:3) == 'EA'
                if lls(i,9:10) == '.i'
                    done = 1;
                end
            end
        end
    end
end
disp('updated file in nix_confun1');
dos('abaqus job=FEAinput');
disp('Abaqus is running');
checkLCK('FEAinput');
disp('Abaqus is done');
[maxdisp,maxstress] = readoutput_trussbeam('FEAinput');
num2str([s,maxdisp,maxstress])
c1= maxdisp-0.01;
c2=maxstress-200e6;
C = [c1;c2];
Ceq = [];
confun_calculation=confun_calculation+1
fid5=fopen('out5.m', 'a');
fprintf(fid5,'%s\n', num2str([s, C]))
fclose(fid5);
```

**iMAGv3 objective function:**

```
function y = nix_objfun3(s)
global confun_calculation
global objfun_calculation
dos('del FEAinput.inp')
csvwrite('FEAinput.csv', s');
done = 0;
while ~done
    lls = ls;
    pause(0.1)
    for i=1:size(lls,1)
        if lls(i,1) == 'F'
            if lls(i,2:3) == 'EA'
                if lls(i,9:10) == '.i'
                    done = 1;
                end
            end
        end
    end
end
```

```

                end
            end
        end
    end
end
total_length=ComputeWeight01();
cross_area=pi*s(1)*s(1);
material_density=8050;
elem_weight=total_length*cross_area*material_density;
y=elem_weight
objfun_calculation=objfun_calculation+1
fid5=fopen('out5.m', 'a');
fprintf(fid5,'%s\n', num2str([s, y]));
fclose(fid5);

```

**iMAGv3 main optimization function:**

```

clc
clear all
close all
format long
time_start = clock;
global confun_calculation
global objfun_calculation
confun_calculation=0
objfun_calculation=0
FUN = 'nix_objfun3';
NONLCON = 'nix_confun3';
options=optimoptions('fmincon','Display', 'iter',
'Algorithm','interior-point', 'DiffMinChange',0.001, 'tolfun', 0.001);
X0 = [0.095,15,3.5];
LB = [0.01,10,1];
UB = [0.1,40,4];
[x_optim,f_optim] = fmincon(FUN,X0,[],[],[],[],LB,UB,NONLCON,options)
time_start
time_end = clock

```

## APPENDIX J: iGFM Source Code

iGFM has the same workflow as iMAGv3 but replaces “call Abaqus” with “call main customized FEA code.”

### **iGFM FEA read analysis setup:**

```
done = 0;
j = 0;
index_blank = 0;
while ~done
    j = j+1;
    if j>length(abaqus_input_file)
        done = 1;
    else
        if abaqus_input_file(j) == ' '
            done = 1;
            index_blank = j;
        end
    end
end
if index_blank > 0
    abaqus_input_file = abaqus_input_file(1:index_blank-1);
end
fid = fopen(abaqus_input_file,'r');
inpdata = [fgetl(fid)];
count1 = 1;
card_lines = [];
cards = [];
done = 0;
while ~done
    tline = fgetl(fid);
    if ~ischar(tline)
        done = 1;
    else
        if length(tline) < size(inpdata,2)
            for jj=1:size(inpdata,2)-length(tline)
                tline = [tline,' '];
            end
        else
            column1 = [];
            for ii=1:size(inpdata,1)
                column1 = [column1;' '];
            end
            for jj=1:length(tline)-size(inpdata,2)
                inpdata = [inpdata,column1];
            end
        end
        if length(tline) > 1
            if tline(1) == '*'
                if tline(2) ~= '*'
                    count1 = count1+1;
                    inpdata = [inpdata;tline];
                    card_lines = [card_lines;count1];
                    if length(tline) < size(cards,2)
                        for jj=1:size(cards,2)-length(tline)
                            tline = [tline,' '];
                        end
                    end
                end
            end
        end
    end
end
```



```

        else
            column1 = [];
            for ii=1:size(cards,1)
                column1 = [column1;' '];
            end
            for jj=1:length(tline)-size(cards,2)
                cards = [cards,column1];
            end
        end
        cards = [cards;tline];
    end
else
    count1 = count1+1;
    inpdata = [inpdata;tline];
end
end
end
end
nodes = [];
elems = [];
props = [];
setsN = [];
setsE = [];
BCs = [];
forces = [];
PiGs = [];
for i=1:size(cards,1)
    if cards(i,1:5) == '*Node'
        n1 = i;
    end
end
for i=card_lines(n1)+1:card_lines(n1+1)-1
    nodes = [nodes;str2num(inpdata(i,:))];
end
if nodes(1,1) == nodes(size(nodes,1),1)+1
    nodes1 = [];
    for i=2:size(nodes,1)
        nodes1 = [nodes1;nodes(i,:)];
    end
    nodes1 = [nodes1;nodes(1,:)];
    nodes = nodes1;
end
n1 = [];
elem_types = [];
elem_elemsets = [];
for i=1:size(cards,1)
    if cards(i,1:5) == '*Elem'
        if cards(i,11:14) == 'type'
            done1 = 0;
            j = 14;
            while ~done1
                j = j+1;
                if j==size(cards,2)-1
                    error('Should have found a comma!')
                end
            end
        end
    end
end

```

```

        else
            if cards(i,j) == ','
                done1 = 1;
                j_comma2 = j;
            end
        end
    end
    type1 = cards(i,16:j-1);
    if length(type1) <= size(elem_types,2)
        for j=1:size(elem_types,2)-length(type1)
            type1 = [type1,' '];
        end
    else
        column1 = [];
        for k=1:size(elem_types,1)
            column1 = [column1;' '];
        end
        for j=1:length(type1)-size(elem_types,2)
            elem_types = [elem_types,column1];
        end
    end
    elem_types = [elem_types;type1];
    done1 = 0;
    j = j_comma2;
    while ~done1
        j = j+1;
        if j==size(cards,2)-1
            error('Should have found a equal sign!')
        else
            if cards(i,j) == '='
                done1 = 1;
                j_equal = j;
            end
        end
    end
    done1 = 0;
    j = j_equal+1;
    while ~done1
        j = j+1;
        if j==size(cards,2)
            done1 = 1;
            j_done1 = j+1;
        else
            if cards(i,j) == ' '
                done1 = 1;
                j_done1 = j;
            end
        end
    end
    elset1 = cards(i,j_equal+1:j_done1-1);
    if length(elset1) <= size(elem_elemsets,2)
        for j=1:size(elem_elemsets,2)-length(elset1)
            elset1 = [elset1,' '];
        end
    end
end

```

```

        else
            column1 = [];
            for k=1:size(elem_elemsets,1)
                column1 = [column1;' '];
            end
            for j=1:length(elset1)-size(elem_elemsets,2)
                elem_elemsets = [elem_elemsets,column1];
            end
            end
            elem_elemsets = [elem_elemsets;elset1];
        else
            error('There should be "type" here!')
        end
        n1 = [n1;i];
    end
end
for il=1:size(n1,1)
    for i=card_lines(n1(il))+1:card_lines(n1(il)+1)-1
        elems = [elems;str2num(inpdata(i,:)),il];
    end
end
if elems(1,1) == elems(size(elems,1),1)+1
    elems1 = [];
    for i=2:size(elems,1)
        elems1 = [elems1;elems(i,:)];
    end
    elems1 = [elems1;elems(1,:)];
    elems = elems1;
end
nset_names = [];
nsets = [];
for i=1:size(cards,1)
    if cards(i,1:5) == '*Nset'
        if cards(i,26) == ' '
            nset_names = [nset_names;cards(i,15:25),' '];
        elseif cards(i,27) == ' '
            nset_names = [nset_names;cards(i,15:26),' '];
        elseif cards(i,28) == ' '
            nset_names = [nset_names;cards(i,15:27),' '];
        elseif cards(i,29) == ' '
            nset_names = [nset_names;cards(i,15:28)];
        end
        nsets = [nsets;inpdata(card_lines(i)+1,:)];
    end
end
props1 = [];
props1_elemsets = [];
for i=1:size(cards,1)
    if cards(i,1:5) == '*Beam'
        if cards(i,16:20) == 'elset'
            done1 = 0;
            j = 23;
            while ~done1
                j = j+1;
            end
        end
    end
end

```

```

        if j==size(cards,2)-1
            error('Should have found a comma!')
        else
            if cards(i,j) == ','
                done1 = 1;
            end
        end
        elemset1 = cards(i,24:j-1);
    else
        error('There should be "elset" here!')
    end
end
if inpdata(card_lines(i),32) == ','
    props1_elemsets = [props1_elemsets;elemset1,'  '];
elseif inpdata(card_lines(i),33) == ','
    props1_elemsets = [props1_elemsets;elemset1,'  '];
elseif inpdata(card_lines(i),34) == ','
    props1_elemsets = [props1_elemsets;elemset1,'  '];
elseif inpdata(card_lines(i),35) == ','
    props1_elemsets = [props1_elemsets;elemset1];
end
    props1 = [props1;str2num(inpdata(card_lines(i)+1,:)),...
        str2num(inpdata(card_lines(i)+2,:))];
end
end
for i=1:size(cards,1)
    if cards(i,1:5) == '*Soli'
        if cards(i,17:21) == 'elset'
            done1 = 0;
            j = 24;
            while ~done1
                j = j+1;
                if j==size(cards,2)-1
                    error('Should have found a comma!')
                else
                    if cards(i,j) == ','
                        done1 = 1;
                    end
                end
            end
            elemset1 = cards(i,25:j-1);
        else
            error('There should be "elset" here!')
        end

        if inpdata(card_lines(i),33) == ','
            props1_elemsets = [props1_elemsets;elemset1,'  '];
        elseif inpdata(card_lines(i),34) == ','
            props1_elemsets = [props1_elemsets;elemset1,'  '];
        elseif inpdata(card_lines(i),35) == ','
            props1_elemsets = [props1_elemsets;elemset1,'  '];
        elseif inpdata(card_lines(i),36) == ','
            props1_elemsets = [props1_elemsets;elemset1];
        end
    end
end

```

```

        end
        props1 = [props1;str2num(inpdata(card_lines(i)+1,:)),0,0,0,0];
    end
end
for i=1:size(cards,1)
    if cards(i,1:5) == '*Elas'
        stiff1 = str2num(inpdata(card_lines(i)+1,:));
    end
end
BC_names = [];
for i=1:size(cards,1)
    if cards(i,1:5) == '*Boun'
        if inpdata(card_lines(i)+1,12) == ','
            BC_names = [BC_names;inpdata(card_lines(i)+1,1:11),' '];
            if inpdata(card_lines(i)+1,14:16) ~= 'ENC'
                error('ERROR IN READING BOUNDARY CONDITIONS')
            end
        elseif inpdata(card_lines(i)+1,13) == ','
            BC_names = [BC_names;inpdata(card_lines(i)+1,1:12),' '];
            if inpdata(card_lines(i)+1,15:17) ~= 'ENC'
                error('ERROR IN READING BOUNDARY CONDITIONS')
            end
        elseif inpdata(card_lines(i)+1,14) == ','
            BC_names = [BC_names;inpdata(card_lines(i)+1,1:13),' '];
            if inpdata(card_lines(i)+1,16:18) ~= 'ENC'
                error('ERROR IN READING BOUNDARY CONDITIONS')
            end
        elseif inpdata(card_lines(i)+1,15) == ','
            BC_names = [BC_names;inpdata(card_lines(i)+1,1:14)];
            if inpdata(card_lines(i)+1,17:19) ~= 'ENC'
                error('ERROR IN READING BOUNDARY CONDITIONS')
            end
        end
    end
end
end
Cloud_names = [];
Cloud_values = [];
for i=1:size(cards,1)
    if cards(i,1:5) == '*Cloa'
        if inpdata(card_lines(i)+1,12) == ' '
            Cloud_names = [Cloud_names;inpdata(card_lines(i)+1,1:11),'
'];
            Cloud_values =
[Cloud_values;str2num(inpdata(card_lines(i)+1,12:size(inpdata,2)))];
        elseif inpdata(card_lines(i)+1,13) == ' '
            Cloud_names = [Cloud_names;inpdata(card_lines(i)+1,1:12),'
'];
            Cloud_values =
[Cloud_values;str2num(inpdata(card_lines(i)+1,13:size(inpdata,2)))];
        elseif inpdata(card_lines(i)+1,14) == ' '
            Cloud_names = [Cloud_names;inpdata(card_lines(i)+1,1:13),'
'];
            Cloud_values =
[Cloud_values;str2num(inpdata(card_lines(i)+1,14:size(inpdata,2)))];

```

```

        elseif inpdata(card_lines(i)+1,15) == ' '
            Cload_names = [Cload_names;inpdata(card_lines(i)+1,1:14)];
            Cload_values =
[Cload_values;str2num(inpdata(card_lines(i)+1,15:size(inpdata,2)))];
        end
    end
end

guynodes = [];
guyelements = [];
BCs = [1 1001 1 0 2 0 3 0 4 0 5 0 6 0];
setsN = [1001];
for i=1:size(BC_names,1)
    index1 = 0;
    for k=1:size(nset_names,1)
        if BC_names(i,:) == nset_names(k,:)
            index1 = k;
        end
    end
    if index1 == 0
        error('THERE IS A PROBLEM IN NODE SETS AND BC''S')
    else
        setsN = [setsN,str2num(nsets(index1,:))];
    end
end
end
forces = [];
count2 = 0;
for i=1:size(Cload_names,1)
    index1 = 0;
    for k=1:size(nset_names,1)
        if Cload_names(i,:) == nset_names(k,:)
            index1 = k;
        end
    end
    if index1 == 0
        error('THERE IS A PROBLEM IN NODE SETS AND CLOADS')
    else
        nodelist2 = [];
        forces12 = [];
        nset_str = str2num(nsets(index1,:));
        for j=1:size(nset_str,2)
            count2 = count2+1;
            nodelist2 = [nodelist2;[2000+count2,nset_str(j)]];
            comps2 = [0 0 0];
            comps2(Cload_values(i,1)) = Cload_values(i,2);
            forces12 = [forces12;[count2,2000+count2,comps2]];
        end
    end
    nodelist2 = [nodelist2,zeros(size(nodelist2,1),size(setsN,2)-2)];
    setsN = [setsN;nodelist2];
    forces = [forces;forces12];
end
setsE = [];

```

```

PiGs = [];
nodes0 = nodes;
[A,I] = sort(nodes0(:,1));
nodes = nodes0(I,:);
if size(nodes,1)~=nodes(size(nodes,1),1)
    error('There is a gap in node numbering')
end
elems0 = elems;
[A,I] = sort(elems0(:,1));
elems = elems0(I,:);
if size(elems,1)~=elems(size(elems,1),1)
    error('There is a gap in element numbering')
end
if size(props1_elemsets,2) > size(elem_elemsets,2)
    lack=size(props1_elemsets(end,:))-size(elem_elemsets(end,:));

props1_elemsets(:,size(elem_elemsets,2)+1:size(elem_elemsets,2)+lack(2)
)=[];
end
for i=1:size(elems,1)
    foundit = 0;
    for k=1:size(props1_elemsets,1)
        if elem_elemsets(elems(i,4),:) == props1_elemsets(k,:)
            foundit = k;
        end
    end
    if ~foundit
        error('Could not find the element set')
    else
        if foundit ~= elems(i,4)
            error('This number should be different!')
        else
            if elem_types(foundit,1:3) == 'B31'
                elems(i,5:8) = [2,props1(foundit,3:5)];
            else
                if elem_types(foundit,1:3) == 'T3D'
                    elems(i,5:8) = [1 0 1 0];
                else
                    error('Strange element type!')
                end
            end
        end
    end
end
end
for i=1:size(props1,1)
    done1 = 0;
    k = 0;
    while ~done1
        k = k+1;
        if k > size(elem_elemsets,1)
            error('Should have found an element set!')
        else
            if elem_elemsets(k,:) == props1_elemsets(i,:)
                done1 = 1;
            end
        end
    end
end

```

```

        if elem_types(k,1:3) == 'B31'
            area1 = pi*(props1(i,1)^2-(props1(i,1)-
props1(i,2))^2);
            inertial1 = pi*(props1(i,1)^4-(props1(i,1)-
props1(i,2))^4)/4;
            props(k,:) = [k,area1,inertial1,7000,stiff1(1)];
        else
            if elem_types(k,1:3) == 'T3D'
                props(k,:) = [k,props1(i,1),9e-
5,7000,stiff1(1)];
            else
                error('This element type is not valid. Need
upgrade!')
            end
        end
    end
end
end
end
end
end
fclose(fid);
nnodes = size(nodes,1);
nelems = size(elems,1);
display('NOW in FEA_read_input_ABAQUS05')

```

**iGFM FEA create load vectors:**

```

load_vector = zeros(6*nnodes,1);
for i=1:size(forces,1)
    done = 0;
    counter = 0;
    while ~done
        counter = counter+1;
        if forces(i,2) == setsN(counter)
            done = 1;
        end
    end
    end
    forcesnodes = setsN(counter,2:size(setsN,2));
    for j=1:length(forcesnodes)
        if forcesnodes(j) ~= 0
            load_vector(6*(forcesnodes(j)-1)+1) = ...
load_vector(6*(forcesnodes(j)-1)+1)+forces(i,3);
            load_vector(6*(forcesnodes(j)-1)+2) = ...
load_vector(6*(forcesnodes(j)-1)+2)+forces(i,4);
            load_vector(6*(forcesnodes(j)-1)+3) = ...
load_vector(6*(forcesnodes(j)-1)+3)+forces(i,5);
        end
    end
end
end
display('DONE in FEA_create_loads_vector01')

```

**iGFM FEA create global stiffness matrix:**

```

K = zeros(6*nnodes,6*nnodes);

```



```

for i=1:nelems
    if elems(i,5) == 2
        a = nodes(elems(i,2),2:4);
        b = nodes(elems(i,3),2:4);
        v1 = b-a;
        e1 = v1/norm(v1);
        v2_inter = elems(i,6:8);
        e2 = v2_inter/norm(v2_inter);
        v3 = cross(v1,v2_inter);
        if norm(v3) == 0
            v2_inter = [1 0 0];
            v3 = cross(v1,v2_inter);
            if norm(v3) == 0
                v2_inter = [0 1 0];
                v3 = cross(v1,v2_inter);
            end
        end
        e3 = v3/norm(v3);
        e2 = cross(e3,e1);
        T1 = [e1;e2;e3];
        T = [
            T1 zeros(3) zeros(3) zeros(3)
            zeros(3) T1 zeros(3) zeros(3)
            zeros(3) zeros(3) T1 zeros(3)
            zeros(3) zeros(3) zeros(3) T1
        ];
        propnumber = elems(i,4);
        A = props(propnumber,2);
        I = props(propnumber,3);
        E = props(propnumber,5);
        L = norm(b-a);
        G = E/2/(1+0.26);
        J = 2*I;
        Iz=I;
        Iy=I;
        k1 = E*A/L;
        k2 = 12*E*Iz/(L*L*L);
        k3 = 6*E*Iz/(L*L);
        k4 = 4*E*Iz/L;
        k5 = 2*E*Iz/L;
        k6 = 12*E*Iy/(L*L*L);
        k7 = 6*E*Iy/(L*L);
        k8 = 4*E*Iy/L;
        k9 = 2*E*Iy/L;
        k10 = G*J/L;
        a=[k1 0 0; 0 k2 0; 0 0 k6];
        b=[ 0 0 0;0 0 k3; 0 -k7 0];
        c=[k10 0 0;0 k8 0; 0 0 k4];
        d=[-k10 0 0;0 k9 0;0 0 k5];
        ke_local = [a b -a b;b' c b d; (-a)' b' a -b;b' d' (-b)' c];
        ke = T'*ke_local*T;
        n1 = elems(i,2);
        n2 = elems(i,3);
        i1 = (n1-1)*6;
    end
end

```

```

j1 = (n1-1)*6;
for j=1:6
    for k=1:6
        K(i1+k,j1+j) = K(i1+k,j1+j)+ke(k,j);
    end
end
i1 = (n1-1)*6;
j1 = (n2-1)*6;
for j=1:6
    for k=1:6
        K(i1+k,j1+j) = K(i1+k,j1+j)+ke(k,6+j);
    end
end
i1 = (n2-1)*6;
j1 = (n1-1)*6;
for j=1:6
    for k=1:6
        K(i1+k,j1+j) = K(i1+k,j1+j)+ke(6+k,j);
    end
end
i1 = (n2-1)*6;
j1 = (n2-1)*6;
for j=1:6
    for k=1:6
        K(i1+k,j1+j) = K(i1+k,j1+j)+ke(6+k,6+j);
    end
end
end

else
    if elems(i,5) == 1
        a = nodes(elems(i,2),2:4);
        b = nodes(elems(i,3),2:4);
        c(1) = dot(b-a,[1,0,0])/norm(b-a);
        c(2) = dot(b-a,[0,1,0])/norm(b-a);
        c(3) = dot(b-a,[0,0,1])/norm(b-a);
        propnumber = elems(i,4);
        A = props(propnumber,2);
        E = props(propnumber,5);
        ke = [
            c(1)^2 c(1)*c(2) c(1)*c(3) 0 0 0 -c(1)^2 -c(1)*c(2) -
c(1)*c(3) 0 0 0
            c(1)*c(2) c(2)^2 c(2)*c(3) 0 0 0 -c(1)*c(2) -c(2)^2 -
c(2)*c(3) 0 0 0
            c(1)*c(3) c(2)*c(3) c(3)^2 0 0 0 -c(1)*c(3) -c(2)*c(3)
-c(3)^2 0 0 0
            0 0 0 0 0 0 0 0 0 0 0 0
            0 0 0 0 0 0 0 0 0 0 0 0
            0 0 0 0 0 0 0 0 0 0 0 0
            -c(1)^2 -c(1)*c(2) -c(1)*c(3) 0 0 0 c(1)^2 c(1)*c(2)
c(1)*c(3) 0 0 0
            -c(1)*c(2) -c(2)^2 -c(2)*c(3) 0 0 0 c(1)*c(2) c(2)^2
c(2)*c(3) 0 0 0

```

```

c(3)^2 0 0 0      -c(1)*c(3) -c(2)*c(3) -c(3)^2 0 0 0 c(1)*c(3) c(2)*c(3)
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
];
ke = A*E/norm(b-a)*ke;
n1 = elems(i,2);
n2 = elems(i,3);
i1 = (n1-1)*6;
j1 = (n1-1)*6;
for j=1:6
    for k=1:6
        K(i1+k,j1+j) = K(i1+k,j1+j)+ke(k,j);
    end
end
i1 = (n1-1)*6;
j1 = (n2-1)*6;
for j=1:6
    for k=1:6
        K(i1+k,j1+j) = K(i1+k,j1+j)+ke(k,6+j);
    end
end
i1 = (n2-1)*6;
j1 = (n1-1)*6;
for j=1:6
    for k=1:6
        K(i1+k,j1+j) = K(i1+k,j1+j)+ke(6+k,j);
    end
end
i1 = (n2-1)*6;
j1 = (n2-1)*6;
for j=1:6
    for k=1:6
        K(i1+k,j1+j) = K(i1+k,j1+j)+ke(6+k,6+j);
    end
end

else
    error('Can only handle trusses and beams')
end
end
ke_ALL(i,,:) = ke;
end

```

**iGFM FEA apply boundary condition:**

```

constraineddofs = [];
for i=1:size(BCs,1)
    done = 0;
    counter = 0;
    while ~done
        counter = counter+1;

```

```

        if BCs(i,2) == setsN(counter)
            done = 1;
        end
    end
end
BCnodes = setsN(counter,2:size(setsN,2));
for j=1:length(BCnodes)
    if BCnodes(j) ~= 0
        BC_line = BCs(counter,:);
        for k=3:length(BC_line)
            if floor(k/2)*2 ~= k
                constraineddofs = [constraineddofs; (BCnodes(j)-
1)*6+...
                                BC_line(k)];
            end
        end
    end
end
end
constraineddofs=constraineddofs';
sizenodes=size(nodes);
sizenodes1=sizenodes(1);
ndofs=sizenodes1*6;
alldofs=[1:ndofs];
freedofs=setdiff(alldofs,constraineddofs);
K11=K(freedofs,freedofs);
K12=K(freedofs,constraineddofs);
K21=K(constraineddofs,freedofs);
K22=K(constraineddofs,constraineddofs);
load_vector5=load_vector(freedofs);
RemovedRows = all(K11 == 0, 2);
IndexRemoved=find(RemovedRows);
UnremovedRows = ~all(K11 == 0, 2);
IndexUnremoved=find(UnremovedRows);
K2=K11(any(K11,2),any(K11,2));
load_vector2=[];
load_vector2=[load_vector2;load_vector5(IndexUnremoved)];
columns_to_be_removed=IndexRemoved;
columns_to_be_kept=IndexUnremoved;

```

#### **iGFM FEA calculate displacement**

```

displacement3 = inv(K2)*load_vector2;
displacement2 =
zeros(length(displacement3)+length(columns_to_be_removed),1);
displacement2(columns_to_be_kept(1:length(columns_to_be_kept)),1)=displ
acement3((1:length(columns_to_be_kept)),1);
displacement=zeros(length(alldofs),1);
    displacement(freedofs)=displacement2;
[displacements6,added]=vec2mat(displacement,6);
displacements_translation=displacements6(1:nnodes,1:3);
displacements_norm=sqrt(sum(displacements_translation.^2, 2));

```

#### **iGFM FEA calculate stresses:**

```

reactions6=zeros(nnodes,6);
[reactions6,added2]=vec2mat(reactions,6);
elem_VM_stresses = [];
for i=1:nelems
    if elems(i,5) == 1
        ke(:, :) = ke_ALL(i, :, :);
        ue = [displacement((elems(i,2)-1)*6+1:(elems(i,2)-1)*6+6);...
            displacement((elems(i,3)-1)*6+1:(elems(i,3)-1)*6+6)];
        external_loads = ke*ue;
        v1 = nodes(elems(i,3),2:4)-nodes(elems(i,2),2:4);
        e1 = v1/norm(v1);
        v2_inter = elems(i,6:8);
        e2 = v2_inter/norm(v2_inter);
        v3 = cross(v1,v2_inter);
        if norm(v3) == 0
            v2_inter = [1 0 0];
            v3 = cross(v1,v2_inter);
            if norm(v3) == 0
                v2_inter = [0 1 0];
                v3 = cross(v1,v2_inter);
            end
        end
        e3 = v3/norm(v3);
        e2 = cross(e3,e1);
        T1 = [e1;e2;e3];
        T = [
            T1 zeros(3) zeros(3) zeros(3)
            zeros(3) T1 zeros(3) zeros(3)
            zeros(3) zeros(3) T1 zeros(3)
            zeros(3) zeros(3) zeros(3) T1
        ];
        internal_loads = -T*external_loads;
        areal = props(elems(i,4),2);
        stress_VMABCD = [];
        for k1=1:2
            N11 = internal_loads((k1-1)*6+1)
            stress_normal_N11 = N11/areal;
            s11 = stress_normal_N11;
            stress_VM = abs(s11);
            stress_VMABCD = [stress_VMABCD, stress_VM, 0, 0, 0];
        end
    else
        if elems(i,5) == 2
            ke(:, :) = ke_ALL(i, :, :);
            ue = [displacement((elems(i,2)-1)*6+1:(elems(i,2)-
1)*6+6);...
                displacement((elems(i,3)-1)*6+1:(elems(i,3)-1)*6+6)];
            external_loads = ke*ue;
            v1 = nodes(elems(i,3),2:4)-nodes(elems(i,2),2:4);
            e1 = v1/norm(v1);
            v2_inter = elems(i,6:8);
            e2 = v2_inter/norm(v2_inter);
            v3 = cross(v1,v2_inter);
            if norm(v3) == 0

```

```

        v2_inter = [1 0 0];
        v3 = cross(v1,v2_inter);
        if norm(v3) == 0
            v2_inter = [0 1 0];
            v3 = cross(v1,v2_inter);
        end
    end
end
e3 = v3/norm(v3);
e2 = cross(e3,e1);
T1 = [e1;e2;e3];
T = [
    T1 zeros(3) zeros(3) zeros(3)
    zeros(3) T1 zeros(3) zeros(3)
    zeros(3) zeros(3) T1 zeros(3)
    zeros(3) zeros(3) zeros(3) T1
];
internal_loads = T*external_loads;
areal = props(elems(i,4),2);
inertial = props(elems(i,4),3);
diameter1 = sqrt(2*areal/3.14159+8*inertial/areal);
stress_VMABCD = [];
for k1=1:2
    N11 = internal_loads((k1-1)*6+1);
    S12 = internal_loads((k1-1)*6+2);
    S13 = internal_loads((k1-1)*6+3);
    T11 = internal_loads((k1-1)*6+4);
    M12 = internal_loads((k1-1)*6+5);
    M13 = internal_loads((k1-1)*6+6);
    stress_normal_N11 = N11/areal;
    stress_shear_S12 = 1.5*S12/areal;
    stress_shear_S13 = 1.5*S13/areal;
    stress_shear_T11 = T11*diameter1/2/2/inertial;
    stress_normal_M12 = M12*diameter1/2/inertial;
    stress_normal_M13 = M13*diameter1/2/inertial;
    s11A = stress_normal_N11-stress_normal_M13;
    s11B = stress_normal_N11+stress_normal_M12;
    s11C = stress_normal_N11+stress_normal_M13;
    s11D = stress_normal_N11-stress_normal_M12;
    s12A = 0;
    s12B = -stress_shear_T11+stress_shear_S12;
    s12C = 0;
    s12D = stress_shear_T11+stress_shear_S12;
    s13A = stress_shear_T11+stress_shear_S13;
    s13B = 0;
    s13C = -stress_shear_T11+stress_shear_S13;
    s13D = 0;
    stress_VMA = sqrt(s11A^2+3*(s12A^2+s13A^2));
    stress_VMB = sqrt(s11B^2+3*(s12B^2+s13B^2));
    stress_VMC = sqrt(s11C^2+3*(s12C^2+s13C^2));
    stress_VMD = sqrt(s11D^2+3*(s12D^2+s13D^2));
    stress_VMABCD =
[stress_VMABCD, stress_VMA, stress_VMB, stress_VMC, stress_VMD];
end
else

```

```
        error('Error here!')
    end
end
elem_VM_stresses = [elem_VM_stresses;i,stress_VMABCD];
end
```

**iGFM FEA main customized FEA:**

```
abaqus_input_file = 'FEAinput.inp';
plotYN = 0;
FEA_read_input_file_ABAQUS05
FEA_create_global_stiffness_matrix03
FEA_create_loads_vector01
FEA_apply_BCs03
FEA_calculate_displacements02
reactions = K*displacement;
FEA_calculate_stresses02
max_disp = max(displacements_norm);
max_stress = max(elem_VM_stresses(:,2));
```

## APPENDIX K: iGAG Source Code

iGAG has the same workflow as iGFM, but replace “Matlab’s interior point” with the “customized GA code.”



**iGAG Grasshopper updated premade component for sending output (adapted for tower case):**

```
Private Sub RunScript(ByVal coordinates As List(Of Object), ByVal
connectivities As List(Of String), ByVal num_numberfloors As Integer,
ByVal ListLengthSectionType As List(Of Integer), ByVal num_ElementType
As List(Of Object), ByVal txt_ElementType As List(Of String), ByVal
section_prop As List(Of String), ByVal index_load As List(Of String),
ByVal ListLengthIndexLoad As List(Of Object), ByVal magnitude_load As
List(Of String), ByVal LoadDirection As List(Of String), ByVal
text_FilePath As String, ByVal input_BC As List(Of Object), ByRef A As
Object, ByRef b As Object)
```

```
    Dim delim As String
    Dim i As Integer
    Dim j As Integer
    Dim v As Integer
    Dim startBC As Integer
    Dim startLoad As Integer
    Dim Crossarea As Double
    delim = ","
    Dim s As String
    s = ""
    s += "*** PARTS" + Constants.vbNewLine
    s += "***" + Constants.vbNewLine
    s += "*Node" + Constants.vbNewLine
    Dim delim_node As String
    delim_node = delim
    For i = 0 To coordinates.Count - 1
        If i = (coordinates.Count - 1) Then
            delim_node = ""
        End If
        s += coordinates(i).ToString() + delim_node
    Next
    startLoad = 5
    v = 0
    For j = 0 To ListLengthIndexLoad.Count - 1
        For i = 0 To ListLengthIndexLoad(j) - 1
            startLoad = startLoad + 1
            s += "*Nset, nset = _PickedSet" + startLoad.ToString() +
Constants.vbNewLine
            s += index_load(v) + Constants.vbNewLine
            v = v + 1
        Next
    Next
    startBC = 5 + index_load.Count
    For i = 0 To input_BC.Count - 1
        startBC = startBC + 1
        s += "*Nset, nset = _PickedSet" + startBC.ToString() +
Constants.vbNewLine
        s += input_BC(i) + Constants.vbNewLine
    Next
    i = 0
```

```

v = 0
Dim delim_connectivities As String
delim_connectivities = delim
Dim k As Integer
Dim elem_n As Integer
elem_n = 0
For j = 0 To txt_ElementType.count - 1
    For k = 0 To num_numberfloors - 1
        elem_n += 1
        s += "*" + "Element, type=" + txt_ElementType(j) + ",
elset=element" + elem_n.toString() + Constants.vbNewLine
        For i = v To v + num_ElementType(j) / num_numberfloors - 1
            v = v + 1
            s += v.toString() + delim_connectivities + connectivities(i)
+ Constants.vbNewLine
        Next
    Next
Next
elem_n = 0
v = 0
For j = 0 To txt_ElementType.count - 1
    For k = 0 To num_numberfloors - 1
        s += "*** Section: Section-" + elem_n.toString() + " Profile:
Profile-" + elem_n.toString() + Constants.vbNewLine
        If txt_ElementType(j) = "T3D2" Then
            s += "*Solid Section, elset = element" + elem_n.toString() +
", material=Material - 1" + Constants.vbNewLine
            Crossarea = 3.14159 * section_prop(elem_n) ^ 2
            s += Crossarea.ToString() + Constants.vbNewLine
        ElseIf txt_ElementType(j) = "B31" Then
            s += "*Beam Section, elset = element" + elem_n.toString() +
", material = Material - 1, section = PIPE" + Constants.vbNewLine
            s += section_prop(elem_n) + Constants.vbNewLine
            s += "0.,0.,-1." + Constants.vbNewLine
        End If
        elem_n += 1
    Next
Next
a = s
End Sub

```

**iGAG customized GA (settings, objective function and constraints are adapted for the tower case):**

```

clc
clear all
close all
dos('del Area.txt')
dos('del TowerHeight.txt');
start_time=clock;
lb=[4 3 0 18]
ub=[20.1 20.1 0.801 26];
optimization_type=1

```

```

max_generation=40;
num_population=50;
crossover_method=1
elitism_active=1
elitism_rate=1;
mutation_rate=0.1
plotresult=1
ub_int_numdigits_vect=[];
power_n_vect=[];
for i=1:size(ub,2)
    ub_numdigits=numel(num2str(ub(i)));
    ub_coma=strfind(num2str(ub(i)),'.')
    power_n=ub_numdigits-ub_coma;
    if isempty(ub_coma)
        power_n=0;
    end
    ub_integer=ub(i)*(10^power_n);
    ub_int=dec2bin(ub_integer);
    ub_int_numdigits=numel(num2str(ub_int));
    ub_int_numdigits_vect=[ub_int_numdigits_vect;ub_int_numdigits]
    power_n_vect=[power_n_vect;power_n]
end
individual=[];
pop=[];
for i=1:num_population
    individual=[];
    for j=1:size(ub,2)
        ub_coma=strfind(num2str(ub(j)),'.')
        if isempty(ub_coma)
            var_j=randi([lb(j) ub(j)]);
        else
            rr=rand(1,1);
            var_j=ub(j)*rr+(1-rr)*lb(j);
        end
        individual=[individual,var_j]
    end
    pop=[pop;individual];
end
if optimization_type==2
    f=1/(1+f)
end
GA_done=0;
iteration=0;
store_data = [];
store_data_analysis=[];
kept_individuals=[];
while ~GA_done
    iteration=iteration+1;
    round_pop=[];
    round_pop_vect=[];
    int_round_pop_vect=[];
    bin_pop=[];
    for i=1:size(ub,2)
        round_pop=pop(:,i);

```

```

round_pop=round(round_pop,power_n_vect(i));
round_pop_vect=[round_pop_vect,round_pop]
int_round_pop=round_pop*10.^power_n_vect(i);
int_round_pop_vect=[int_round_pop_vect,int_round_pop];
int_round_pop_vect_j=int_round_pop_vect(:,i)

bin_pop_i=dec2bin(int_round_pop_vect_j,ub_int_numdigits_vect(i));
cell_bin_pop_i={bin_pop_i}
bin_pop=[bin_pop, cell_bin_pop_i]
end
objfun_set=[];
for i_obj = 1:size(round_pop_vect,1)
    objfun=round_pop_vect(i_obj,3)
    if round_pop_vect(i_obj,1) < lb(1)
        round_pop_vect(i_obj,1)=lb(1)
        objfun=0;
    end
    if round_pop_vect(i_obj,2) < lb(2)
        round_pop_vect(i_obj,2)=lb(2)
        objfun=0;
    end
    if round_pop_vect(i_obj,4) < lb(4)
        round_pop_vect(i_obj,4) = lb(4)
        objfun=0;
    end
    if round_pop_vect(i_obj,1) > ub(1)
        round_pop_vect(i_obj,1)=ub(1)
        objfun=0;
    end
    if round_pop_vect(i_obj,2) > ub(2)
        round_pop_vect(i_obj,2) = ub(2)
        objfun=0;
    end
    if round_pop_vect(i_obj,4) > ub(4)
        round_pop_vect(i_obj,4) = ub(4)
        objfun=0;
    end
    s=round_pop_vect(i_obj,:);
    GA_confun
    if max_stress > 250e6
        objfun=0;
    end
    if max_disp > (0.8/100)* TowerHeight
        objfun=0;
    end
    if TotalFloorArea > 14000
        objfun=0;
    end
    if TotalFloorArea < 12000
        objfun=0;
    end
    data_analysis={num2str([iteration,
s,max_disp,max_stress,TotalFloorArea,TowerHeight])};
    store_data_analysis=[store_data_analysis;data_analysis]

```

```

        objfun_set=[objfun_set;objfun];
    end
    objfun_set
    [A1,I1]=max(objfun_set);
    if optimization_type==1
        optimum_value=max(objfun_set)
    elseif optimization_type==2
        optimum_value=(1/max(objfun_set))-1
    end
    data1 =
[iteration,min(pop),mean(pop),max(pop),min(objfun_set),mean(objfun_set)
,max(objfun_set),optimum_value,pop(I1,:)];
    store_data = [store_data;data1]
    fitness_set=objfun_set(:,1);
    relative_fit_set=[];
    for i=1:size(fitness_set,1);
        relative_fit=1*(fitness_set(i,1)/sum(fitness_set));
        relative_fit_set=[relative_fit_set;relative_fit];
    end
    bin_pop_nplets=[];
    if elitism_active==1
        sorted_fit_set=sort(relative_fit_set)
        top_quartile_value=sorted_fit_set(floor(((100-
elitism_rate)/100)*size(sorted_fit_set,1)), 1)
        top_quartile_index=find(relative_fit_set >= top_quartile_value)
        for i=1:size(bin_pop{1},1)
            row_bin_set=[];
            for j=1:size(ub,2)
                row_bin={bin_pop{j}(i,:)};
                row_bin_set=[row_bin_set,row_bin];
            end
            bin_pop_nplets=[bin_pop_nplets;row_bin_set];
        end
        kept_individuals= bin_pop_nplets(top_quartile_index,:)
        kept_individuals2=kept_individuals;
        char_individuals=[];
        for i = 1: size(kept_individuals2,1)
            individual1=kept_individuals2(i,:);
            string_individual=strcat(individual1{1,:});
            char_individuals=[char_individuals;string_individual];
        end
        char_individuals;
        [C,ia,ic]=unique(char_individuals,'rows');
        kept_individuals=kept_individuals2(ia,:)
    end
    matingpool=[];
    percent_fitness=floor(100*num_population*relative_fit_set);
    for i=1: size(percent_fitness,1)
        for j=1:percent_fitness(i,1)
            matingpool=[matingpool;bin_pop_nplets(i,:)];
        end
    end
    fitparent=[];
    for i = 1:size(percent_fitness,1)

```

```

    pick1 = randi([1 size(matingpool,1)]);
    fitparent=[fitparent;matingpool(pick1,:)];
end
if crossover_method==1
    child_set=[];
    for i=1:size(fitparent,1)/2 - size(kept_individuals,1)/2
        pickparent1=randi([1 size(fitparent,1)]);
        parent1=fitparent(pickparent1,:);
        pickparent2=randi([1 size(fitparent,1)]);
        parent2=fitparent(pickparent2,:);
        child_i=[];
        for j =1:size(fitparent,2)
            child_ij=[];
            split_index=ceil(rand()*(ub_int_numdigits_vect(j)-1));
            child1=[parent1{j}(1, 1:split_index), parent2{j}(1,
split_index+1:ub_int_numdigits_vect(j))];
            child2=[parent2{j}(1, 1:split_index), parent1{j}(1,
split_index+1:ub_int_numdigits_vect(j))];
            child_ij=[child_ij; {child1}; {child2}];
            child_i=[child_i,child_ij];
        end
        child_set=[child_set;child_i];
    end
    child_set
end
if crossover_method==3
    fitparent
    child_set=[];
    for i=1:size(fitparent,1)/2 - size(kept_individuals,1)/2
        pickparent1=randi([1 size(fitparent,1)]);
        parent1=fitparent(pickparent1,:);
        pickparent2=randi([1 size(fitparent,1)]);
        parent2=fitparent(pickparent2,:);
        child1=[];
        child2=[];
        for j=1:size(fitparent,2)
            child_1k=[];
            child_2k=[];
            for k=1:size(fitparent{i,j},2)
                crossover_prob=rand(1,1);
                if crossover_prob < 0.5
                    child1_gene=parent1{j}(1,k);
                    child2_gene=parent2{j}(1,k);
                elseif crossover_prob >= 0.5
                    child1_gene=parent2{j}(1,k);
                    child2_gene=parent1{j}(1,k);
                end
                child_1k=[child_1k,child1_gene];
                child_2k=[child_2k,child2_gene];
            end
            child1=[child1,{child_1k}];
            child2=[child2,{child_2k}];
        end
        child_set=[child_set;child1;child2];
    end
end

```

```

        end
        child_set
    end
    child_set
    for i = 1:size(child_set,1)
        for j=1:size(child_set,2)
            for k=1:size(child_set{i,j},2)
                mutation_prob=rand(1,1);
                if mutation_prob < mutation_rate
                    if child_set{i,j}(k)=='0'
                        child_set{i,j}(k)=='1';
                    elseif child_set{i,j}(k)=='1'
                        child_set{i,j}(k)=='0';
                    end
                end
            end
        end
    end
    end
    child_set
    if elitism_active==1
        child_set=[child_set;kept_individuals]
    end
    child_set
    pop=[];
    for i=1:size(child_set,1)
        pop_row=[];
        for j=1:size(child_set,2)
            child_int=bin2dec(child_set{i,j});
            int_round_pop=child_int;
            pop_ij=int_round_pop*10^-power_n_vect(j);
            pop_row=[pop_row,pop_ij];
        end
        pop=[pop;pop_row];
    end
    pop
    if iteration==max_generation
        GA_done=1;
    end
end
if plotresult==1
    PlottingGA_only
end
store_data_index_optobjfun=1+size(ub,2)*3+4;
if optimization_type==1
    [A2,I2]=max(store_data(:,store_data_index_optobjfun));
elseif optimization_type==2
    [A2,I2]=min(store_data(:,store_data_index_optobjfun));
end
store_data(I2,store_data_index_optobjfun:size(store_data,2))
start_time
end_time=clock

```

**iGAG customized GA constraint function:**

```
s;
dos('del FEAinput.inp');
dos('del FEAinput.csv');
csvwrite('FEAinput.csv', s');
done = 0;
while ~done
    lls = ls;
    pause(0.1)
    for i=1:size(lls,1)
        if lls(i,1) == 'F'
            if lls(i,2:3) == 'EA'
                if lls(i,9:10) == '.i'
                    done = 1;
                end
            end
        end
    end
end
done = 0;
while ~done
    lls = ls;
    pause(0.1)
    for i=1:size(lls,1)
        if lls(i,1) == 'A'
            if lls(i,2:3) == 're'
                if lls(i,5:6) == '.t'
                    done = 1;
                end
            end
        end
    end
end
end
area_input_file='Area.txt';
fid2 = fopen(area_input_file,'r');
TotalFloorArea=str2double(fgetl(fid2));
fclose(fid2);
done = 0;
while ~done
    lls = ls;
    pause(0.1)
    for i=1:size(lls,1)
        if lls(i,1) == 'T'
            if lls(i,2:3) == 'ow'
                if lls(i,12:13) == '.t'
                    done = 1;
                end
            end
        end
    end
end
end
TowerHeight_input_file='TowerHeight.txt';
```



```

fid3 = fopen(TowerHeight_input_file,'r');
TowerHeight=str2double(fgetl(fid3));
fclose(fid3);
main03_automatic;
dos('del Area.txt')
dos('del TowerHeight.txt');

```

**Exhaustive search with filtering process:**

```

clc
clear all
close all
format compact
format short
dos('del Area.txt')
dos('del TowerHeight.txt');
load('Tower6400iterations_Exhaustive');
new_store_exhaustive;
string_copy1_store_exhaustive;
new_store_exhaustive;
s4value=new_store_exhaustive(:,5);
values4=19;
sorted_new_store_exhaustive=[];
string_sorted_new_store_exhaustive=[];
for i=1:8
    indexs4=find(s4value==values4);
    data_of_indexs4={new_store_exhaustive(indexs4,:)};

sorted_new_store_exhaustive=[sorted_new_store_exhaustive;data_of_indexs4];
    string_data_of_indexs4={num2str(new_store_exhaustive(indexs4,:))};

string_sorted_new_store_exhaustive=[string_sorted_new_store_exhaustive;
string_data_of_indexs4];
    values4=values4+1
end
x_star_array=[];
store_feasible_x=[];
string_store_feasible_x=[];
actual_store_feasible_x=[];
string_actual_store_feasible_x=[];
twist_max=0.0001;
for i4=1:8
    S=sorted_new_store_exhaustive{i4}(:,2:4);
    number_floors=sorted_new_store_exhaustive{i4}(1,5);
    d_exhaustive=sorted_new_store_exhaustive{i4}(:,6);
    sigma=sorted_new_store_exhaustive{i4}(:,7);
    total_area=sorted_new_store_exhaustive{i4}(:,8);
    tower_height=sorted_new_store_exhaustive{i4}(1,9);
    N = size(S,1);
    Smins = min(S);
    Smaxs = max(S);
    levels1 = [];

```

```

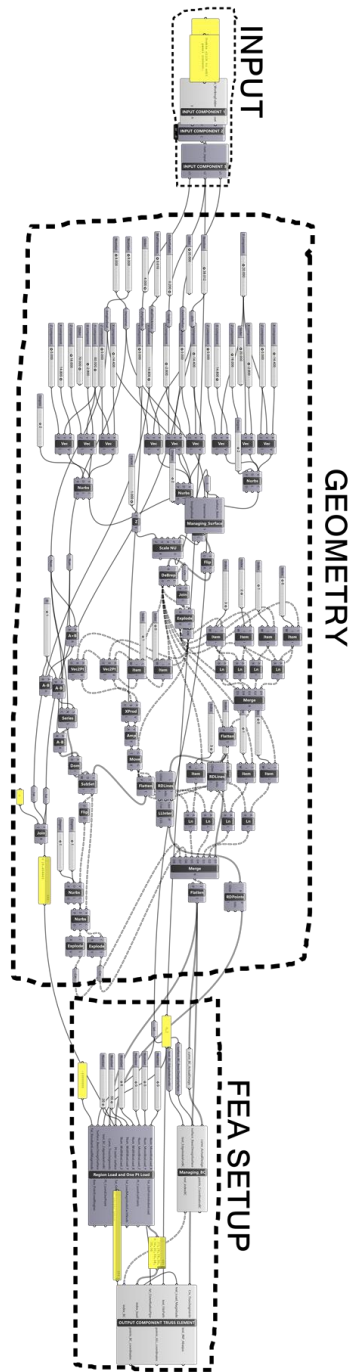
for i=1:N
    found = 0;
    for j=1:size(levels1,1)
        if S(i,1) == levels1(j)
            found = 1;
        end
    end
    if ~found
        levels1 = [levels1;S(i,1)];
    end
end
levels2 = [];
for i=1:N
    found = 0;
    for j=1:size(levels2,1)
        if S(i,2) == levels2(j)
            found = 1;
        end
    end
    if ~found
        levels2 = [levels2;S(i,2)];
    end
end
levels3 = [];
for i=1:N
    found = 0;
    for j=1:size(levels3,1)
        if S(i,3) == levels3(j)
            found = 1;
        end
    end
    if ~found
        levels3 = [levels3;S(i,3)];
    end
end
Nlevels(1) = size(levels1,1);
Nlevels(2) = size(levels2,1);
Nlevels(3) = size(levels3,1);
Nlevels;
sa = Smins;
sb = Smaxs;
M = 5*max(Nlevels);
deltas = (sb-sa)/M;
x_exhaustive = sa;
for i3=1:M+1
    x_exhaustive(2) = sa(2);
    for i2=1:M+1
        x_exhaustive(1) = sa(1);
        for i1=1:M+1
y=interpolate3DCosine01(x_exhaustive,S,d_exhaustive,sigma,total_area);
        if x_exhaustive(3)> twist_max
            if y(3) <= 14000
                if y(3) >= 12000

```

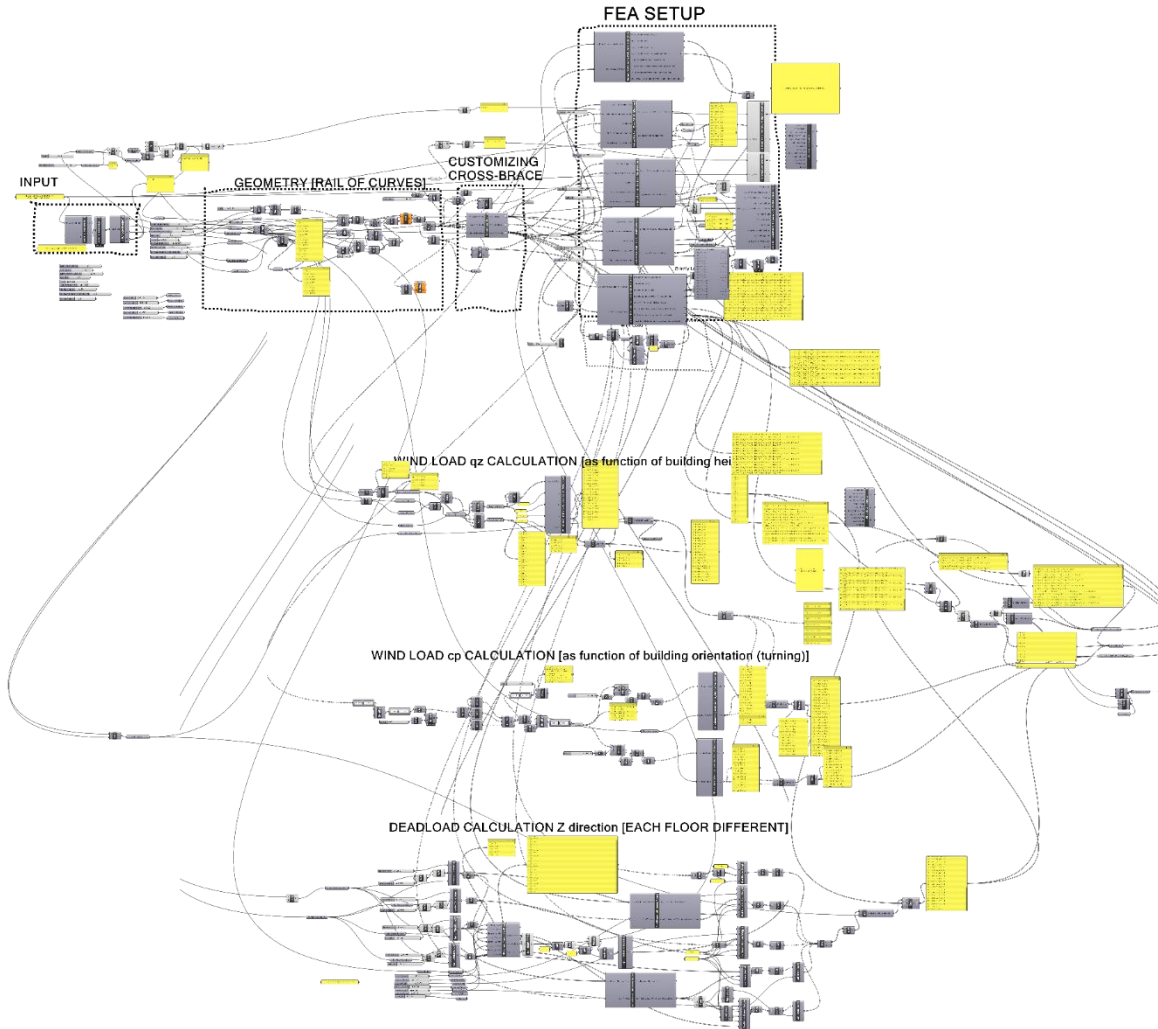


```
        x_exhaustive(1) = x_exhaustive(1)+deltas(1);
    end
    x_exhaustive(2) = x_exhaustive(2)+deltas(2);
end
    x_exhaustive(3) = x_exhaustive(3)+deltas(3);
end
x_star_array=[x_star_array;x_star];
end
```

# APPENDIX L: Grasshopper Definition Space Truss



# APPENDIX M: Grasshopper Definition Skyscraper



## REFERENCES

- Akkermans, H., & Helden, K. (2002). Vicious and virtuous cycles in ERP implementation: A case study of interrelations between critical success factors. *European Journal of Information Systems*, 11, 35-46.
- Albers, A., Enkler, H., Maier, T., & Weiler, H. (2007). Robust design processes with cad based finite element models. *International Conference on Engineering Design*, 531
- Albright, D. (2017). Syllabus ARCH 8710. Retrieved from from <https://etpr.app.clemson.edu/data/spring2017/dalbrig56832spring2017.pdf>
- Al-Haddad, T. (2006). *PerFORMance: Integrating structural feedback into design processes for complex surface-active form* (Master of Architecture).
- Ali, M. M., & Moon, K. (2007). Structural developments in tall buildings: Current trends and future prospects. *Architecture Science Review*, 50(3), 205-223.
- AMPL Optimization inc. (2013). AMPL: Streamlined modeling for real optimization. Retrieved from <http://ampl.com/>
- AMPL-MINOS.Using AMPL/MINOS. Retrieved from [ampl.com/BOOKLETS/ampl-minos.pdf](http://ampl.com/BOOKLETS/ampl-minos.pdf)
- Anderson, E. (2014). *Design and optimization of carbon-fiber chassis panels* (Master of Science in Mechanical Engineering).
- ANSYS Inc. (2017). Pervasive engineering simulation is here. Retrieved from <http://www.ansys.com/>
- Arrow Detailing. (2006). London olympic aquatic center. Retrieved from <http://www.arrowdetail.com/Featured%20Articles/>
- Auerbach, C., & Silverstein, L. (2003). *Qualitative data an introduction to coding and analysis*
- Autodesk. (2017). Revit: Built for BIM. Retrieved from <https://www.autodesk.com/products/revit-family/overview>
- Baerlecken, D., Gentry, R., Swarts, M., & Wonoto, N. (2014). Structural, deployable folds — design and simulation of biological inspired folded structures. *International Journal of Architectural Computing*, 12(3)

- Baerlecken, D., Swarts, M., Gentry, R., & Wonoto, N. (2012). Bio-origami: Form finding and evaluation of origami structures. *Digital Physicality: Proceedings of the 30th eCAADe Conference, 1*, 497-504.
- Barrios, C. (2016). Syllabus ARCH 8700. Retrieved from <https://etpr.app.clemson.edu/data/fall2016/crbh53865fall2016.pdf>
- Bendsoe, M. P., & Sigmund, O. (2003). *Topology optimization: Theory, methods and applications* Springer.
- Bentley. (2017). Generative component: Design and explore the unimaginable. Retrieved from <https://www.bentley.com/en/%20products/product-line/modeling-and-visualization-software/generativecomponents>
- Bentley Systems. (2017). STAAD.pro: Strength meets inspiration. Retrieved from <https://www.bentley.com/en/products/product-line/structural-analysis-software/staadpro>
- Berg, B., & Lune, H. (2011). *Qualitative research methods for the social sciences (8th edition)* (8th ed.) Pearson.
- Bernard, R. (2006). *Research methods in anthropology: Qualitative and quantitative approaches* (4th ed.) Altamira Press.
- Bertelsen, O. (1997). Towards a unified field of SE research and practice. *IEEE Software*, 14(6), 87-88. doi:10.1109/MS.1997.636682
- Biernacki, P., & Waldorf, D. (1981). Snowball sampling: Problems and techniques of chain referral sampling. *Sociological Methods and Research*, 10(2), 141-163.
- Birch, A. (2010). Steel structure of zaha hadid's aquatics center. Retrieved from <http://www.bdonline.co.uk/steel-structure-of-zaha-hadids-aquatics-centre/5005265.article>
- Bourke, P. (1999). Interpolation methods. Retrieved from <http://paulbourke.net/miscellaneous/interpolation/>
- Bowers, B. (1988). Grounded theory: Path to knowledge. *Innovative Research Methods for Nursing*,
- Brauman, J., & Cokcan, S. (2011). Parametric robot control. *Proceedings of the Annual Conference of the Association for Computer Aided Design in Architecture*, Calgary/Banff. 244-251.



- Brink, H. I. L. (1993). Validity and reliability in qualitative research. *Journal of the Democratic Nursing Organisation of South Africa*, 16(2), 35-38.  
doi:10.4102/curationis.v16i2.1396
- Brunvand, S. (2010). Best practices for producing video content for teacher education. *Contemporary Issues in Technology and Teacher Education*, 10(2), 247-256.
- Bucci, F., & Mulazzani, M. (2002). *Luigi moretti works and writings* Princeton Architectural Press.
- Buelow, V., Falk, A., & Turrin, M. (2010). Optimization of structural form using a genetic algorithm to search associative parametric geometry. *Proceedings of the International Conference on Structures Architecture*, , 1 195-196.  
doi:10.1201/b10428-93
- Burry, J., & Burry, M. (2010). *The new mathematics of architecture* (1st ed.) Thames & Hudson.
- Cazacu, R., & Grama, L. (2014). Comparative study for the size optimization of a tapered pipe cantilever beam using genetic algorithms. *Fascicle of Management and Technological Engineering*, (1), 59-62.
- Charleson, A., W., & Pirie, S. (2009). An investigation of structural engineer-architect collaboration. *Journal of the Structural Engineering Society*, 22(1)
- Chen, I. J. (2001). Planning for ERP systems: Analysis and future trend. *Business Process Management Journal*, 7(5), 374-386.
- Christensen, P. W., & Klarbring, A. (2009). *An introduction to structural optimization* (1st ed.). Netherlands: Springer. doi:10.1007/978-1-4020-8666-3
- Clemens, P. (2012). Beam deflection stresses. Retrieved from <http://www.grashopper3d.com/group/karamba/forum/topics/beam-deflection-stresses>
- Clemson University. (2014). M. arch plan of study. Retrieved from <http://www.clemson.edu/caah/departments/architecture/files/files-programs/files-graduate/m-arch-grad-curriculum-plan-2015-16-080115.pdf>
- Cohen, L., & Manion, L. (1994). *Research methods in education* (4th ed.). London: Routledge.
- Coleman, G., & Connor, R. (2007). Using grounded theory to understand software process improvement: A study of irish software product companies. *Journal of*

*Information and Software Technology*, 49(6), 654-667.  
doi:10.1016/j.infsof.2007.02.011

- Creative Research System. (2016). Sample size formula. Retrieved from <https://www.surveysystem.com/sample-size-formula.htm>
- Creswell, J. W. (2006). *Qualitative inquiry and research design: Choosing among five approaches* (2nd ed.) SAGE Publications.
- Dassault Systèmes. (2017). Abaqus unified FEA. Retrieved from <https://www.3ds.com/products-services/simulia/products/abaqus/latest-release/>
- Dassault Systèmes. (2017). Catia: Shape the world we live in. Retrieved from <https://www.3ds.com/products-services/catia/>
- Davidson, S. (2017). Grasshopper.: Algorithmic modeling for rhino. Retrieved from [www.grasshopper3d.com](http://www.grasshopper3d.com)
- Davison, J., Popovic, O., & Tyas, A. (1998). A new degree program in structural engineering and architecture. *Proceedings of the International Conference on Engineering Education*,
- DIANA. (2010). Newton-raphson. Retrieved from <https://dianafea.com/manuals/d943/Analys/node386.html>
- Digital Project. (2015). High performance BIM: Design, engineering, and fabrication BIM for the world's most demanding projects. Retrieved from <http://www.digitalproject3d.com/>
- Divisare. (2016). Zaha hadid architects london aquatic center. Retrieved from <https://divisare.com/projects/195689-zaha-hadid-architects-hufton-crow-london-aquatic-centre>
- Dougherty, D. (1992). Interpretative barriers to successful product innovation in large firms. *Organization Science*, 3(2), 179-202.
- Duczek, S. (2017). Results difference between abaqus standard and abaqus explicit. Retrieved from [https://www.researchgate.net/post/Why\\_the\\_result\\_is\\_different\\_between\\_abaqus\\_standard\\_and\\_abaqus\\_explicit2](https://www.researchgate.net/post/Why_the_result_is_different_between_abaqus_standard_and_abaqus_explicit2)
- Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2008). *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors* (1st ed.) John Wiley & Sons.

- Faridah, A., Sani, A., & Ismail, R. (2004). Bridging the gap: Architecture and structural engineering. *Conference on Engineering Education (CEE 2004)*, Kuala Lumpur.
- Felippa, C. A. (2000). *A historical outline of matrix structural analysis: A play in three acts*. (No. CU-CAS-00-13). Computers & Structures.
- Felkner, J., Chatzi, E., & Kotnik, T. (2013). Architectural feedback in the structural optimization process. *Structures and Architecture: Concepts, Applications, and Challenges*, , 676-684.
- Ferradas, A. (2013). Capital gate abu dhabi. Retrieved from <http://cargocollective.com/Angelferradas/filter/Abu-Dhabi-capital-gate/Capital-Gate-Abu-Dhabi>
- Fielding, N. . (1994). Varieties of research interviews. *Journal of Nurse Researcher*, 1(3), 4-13. doi:0.7748/nr.1.3.4.s2
- Friedrich, E. (2008). *The voronoi diagram in structural optimization* (Master of Science in Adaptive Architecture & Computation).
- Gandhi, S. (2015). Results difference between abaqus standard and abaqus explicit. Retrieved from [https://www.researchgate.net/post/Why\\_the\\_result\\_is\\_different\\_between\\_abaqus\\_standard\\_and\\_abaqus\\_explicit2](https://www.researchgate.net/post/Why_the_result_is_different_between_abaqus_standard_and_abaqus_explicit2)
- Gangamwar, Y., Deo, V., Chate, S., Bhandare, M., & Deshpande, H. N. (2016). Determination of curved beam deflection by using Castigliano's theorem. *International Journal for Research in Emerging Science and Technology*, 3(5), 19-24.
- Gauchia, A., Boada, B. L., Boada, M. J. L., & Díaz, V. (2014). Integration of MATLAB and ANSYS for advanced analysis of vehicle structures. *MATLAB Applications for the Practical Engineer*, doi:10.5772/57390
- Gentry, R., Baerlecken, D., Swarts, M., & Wonoto, N. (2013). Parametric design and Non-Linear analysis of a Large-Scale deployable roof structure based on action origami. *Proceedings of the ICSA 2013*, Portugal. 771-778. doi:10.1201/b15267-108
- Gentry, T. R. (2013). Digital tools for masonry design and construction. *ARCC Journal of Architectural Research*,
- Gerber, D. (2007). *Parametric practices: Models for design exploration in architecture* (Doctor of Design).

- Glaser, B. (1978). *Theoretical sensitivity: Advances in the methodology of grounded theory* (1st ed.) The Sociology Press.
- Glaser, B., & Strauss, A. (1967). *The discovery of grounded theory: Strategies for qualitative research*. New Brunswick, U.S.A: AldineTransaction.
- Glavic, M., & Wehenkel, L. (2004). *Interior point methods: A survey, short survey of applications to power systems, and research opportunities*. (Technical Report). Belgium: The university of Liege in Belgium.
- Graphisoft. (2017). Archicad. Retrieved from <http://www.graphisoft.com/archicad>
- Grounded Theory Institute. (2014). What is grounded theory? . Retrieved from <http://www.groundedtheory.com/what-is-gt.aspx>
- Guba, E., & Lincoln, Y. (1994). Competing paradigms in qualitative research. *Handbook of qualitative research* (pp. 105-117). London: Sage.
- Hedar, A. Booth function. Retrieved from [http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar\\_files/TestGO\\_files/Page816.htm](http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page816.htm)
- Hedar, A. Easom function. Retrieved from [http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar\\_files/TestGO\\_files/Page1361.htm](http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page1361.htm)
- Helbig, T., & Kamp, F. (2014). New terminal 3 for shenzhen bao' an airport: A 1250 m long structure. Retrieved from [http://www.knippershelbig.com/sites/default/files/pdf/new\\_terminal\\_3\\_shenzhen\\_bo\\_a\\_an\\_airport\\_th\\_fk1.pdf](http://www.knippershelbig.com/sites/default/files/pdf/new_terminal_3_shenzhen_bo_a_an_airport_th_fk1.pdf)
- Hendrick, D. (2013). *Using grounded theory to develop a framework for software testing best practice in a telecommunications company* (Master of Science in Computing (Information Technology)).
- Hilal, A. (2010). Qatar education city convention center by yamasaki architects. Retrieved from <http://www.architecturelist.com/2010/02/09/qatar-education-city-convention-centre-by-yamasaki-architects/>
- Holgate, A. (1986). *The art in structural design* Clarendon Press.
- Hudson, R. (2010). *Strategies for parametric design in architecture: An application of practice led research*. (Unpublished Doctor of Philosophy). University of Bath,
- Hurol, Y. (2013). Ethical considerations for a better collaboration between architects and structural engineers: Design of buildings with reinforced concrete frame systems in

earthquake zones. *Journal of Science and Engineering Ethics*, 20(2), 597.  
doi:10.1007/s11948-013-9453-4

IndieLondon.co.uk. (2012). Norman foster. 30 st mary axe. Retrieved from  
<http://www.indielondon.co.uk/gallery/ra-now-gallery?imagenum=10>

Jamil, M., & Yang, X. (2013). A literature survey of benchmark functions for global optimization problems. *Int. Journal of Mathematical Modelling and Numerical Optimisation*, 4(2), 150-194. doi:10.1504/IJMMNO.2013.055204

Karamba. (2016). Karamba. Retrieved from  
[http://www.food4rhino.com/app/karamba?ufh\\_ =](http://www.food4rhino.com/app/karamba?ufh_=)

Kieran, S., & Timberlake, J. (2003). *Refabricating architecture: How manufacturing methodologies are poised to transform building construction* (1st ed.) McGraw-Hill Education.

Knebel, K., Sanchez, J., & Zimmerman, S. *The structural making of the eden domes*. Unpublished manuscript.

Konka, B. (2011). *A case study on software testing methods and tools* (Master of Science Thesis in Software Engineering and Management).

Koutsikouri, D., Austin, S. A., & Dainty, A. R. J. (2008). Critical success factors in collaborative multi-disciplinary. *Journal of Engineer- Ing, Design and Technology*, 6(3), 198-226.

Kumar, V., Maheshwari, B., & Kumar, U. (2003). An investigation of critical management issues in ERP implementation: Empirical evidence from canadian organizations. *International Journal of Technological Innovation, Entrepreneurship and Technology Management*, 23(10), 793-807.

Ky, V., Ambrosio, C., Hamadi, Y., & Liberti, L. (2015). Surrogate-based methods for black-box optimization. *International Transactions in Operational Research*, 24(3)

Larena, A. (2009). Shape design methods based on the optimization of the structure. historical background and application to contemporary architecture. *Proceedings of the Third International Congress on Construction History*, Cottbus.

Lee, D. (2016). Syllabus ARCH 8120. Retrieved from  
<https://etpr.app.clemson.edu/data/spring2016/dlee251515spring2016.pdf>

LSTC. (2011). Ls-dyna. Retrieved from <http://www.lstc.com/products/ls-dyna>

- Luyten, L. (2010). Architect and structural engineer communicating in multidisciplinary creativity. *Structures and Architecture*, Portugal.
- Malherbe, C., Contal, E., & Vayatis, N. (2016). A ranking approach to global optimization. *Proceedings of the 33rd International Conference on Machine Learning (ICML)*,
- Maneval, V. (2017). Architecture bulle - water cube (2003-2008) beijing - china. Retrieved from <http://bubblemania.fr/en/architecture-bulle-cube-deau-2003-2008-pekin-chine/>
- Markus, M. L., Axline, S., Petrie, D., & Tanis, C. (2000). Learning from the adapters' experiences with ERP: Problems encountered and success achieved. *Journal of Information Technology*, 15, 245-265.
- Mentegazzi, E. (2015). *Parametricism: The act of change in architecture*. Unpublished manuscript.
- Meredith, J. R., & Mantel, S. (2006). *Project management: A managerialist approach* (6th ed.). New York: Wiley.
- Mertens, D. M. (2005). *Research methods in education and psychology: Integrating diversity with quantitative and qualitative approaches* (2nd ed.) Thousand Oaks: Sage.
- Mitchell, M. (1998). *An introduction to genetic algorithms* MIT Press.
- Molga, M., & Smutnicki, C. (2005). Test functions for optimization needs. Retrieved from <http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf>
- Moolman, S. (2007). A magic cube. Retrieved from [www.idc-online.com/technical\\_references/pdfs/civil\\_engineering/A\\_magic\\_cube.pdf](http://www.idc-online.com/technical_references/pdfs/civil_engineering/A_magic_cube.pdf)
- Moon, K. (2005). *Dynamic interrelationship between technology and architecture in tall buildings*. (Unpublished PhD Dissertation). Massachusetts Institute of Technology,
- MSC Software Corporation. (2017). MSC nastran: Multidisciplinary structural analysis. Retrieved from <http://www.mscsoftware.com/product/msc-nastran>
- Myers, M. (2000). Qualitative research and the generalizability question: Standing firm with proteus. *The Qualitative Report*, 4(3)
- NCARB - National Council of Architectural Registration. (2016). Gain AXP experience. Retrieved from <http://www.ncarb.org/gain-axp-experience>

- Nero, I. (2004). *Computers, cladding, and curves: The techno-morphism of frank gehry's guggenheim museum in bilbao spain* (Doctor of Philosophy).
- Ozmen, C., & Unay, A. I. (2011). Architect – structural engineer collaboration in sustainable structural system design. *Journal of Science*, 24(4), 919-925.
- Pecherskaya, E. P., Averina, L. V., Kamaletdinov, Y. A., Tretyakova, N. V., & Magomadova, T. L. (2016). Assessment of critical success factors transformation in ERP projects. *International Electronic Journal of Mathematics Education*, 11(7), 2608-2625.
- Peters, T. F. (1991). *Bridging the gap: Rethinking the relationship of architect and engineer*. New York: the Relationship of Architect and Engineer”, Van Nostrand Reinhold.
- Phua, F. T. T. (2004). Modelling the determinants of multifirm project success: A grounded exploration of differing participant perspectives. *Construction Management and Economics*, 22, 451-459.
- Pohlheim, H. (2006). *Genetic and evolutionary algorithm toolbox for matlab*. ().
- Rardin, R. (1997). *Optimization in operations research*. Prentice Hall: Pearson.
- Reiser, R., & Dempsey, J. (2007). *Trends and issues in instructional design and technology* (2nd ed.). Saddle River, NJ: Pearson.
- Rian, L., & Sassone, M. (2014). Tree-inspired dendriforms and fractal-like branching structures in architecture: A brief historical overview. *Frontiers of Architectural Research*, 3(3), 298-323. doi:<http://doi.org/10.1016/j.foar.2014.03.006>
- Robert McNeel & Associates. (2017). Rhinoceros: Design, model, present, analyze, realize. Retrieved from <https://www.rhino3d.com>
- Ross, D. (1961). *Computer-aided design: A statement of objectives*. (Technical Memorandum No. 8436-TM-5). Mas saChufetts: Massachusetts Institute of Tecp.nology.
- Rubin, H., & Rubin, I. (1995). *Qualitative interviewing: The art of hearing data*. CA: Sage.
- Rutten, D. (2017). Galapagos. Retrieved from <http://www.grasshopper3d.com/group/galapagos>
- Saldana, J. (2009). *The coding manual for qualitative researchers* SAGE.

- Sassone, M., & Pugnale, A. (2007). Evolutionary structural optimization in shell designs. *Advanced Numerical Analysis of Shell-Like Structures: Special Workshop*, Zagreb. 247-257.
- Schofield, J. (2012). Capital gate, abu dhabi. *Council on Tall Buildings and Urban Habitat (CTBUH)*, (2), 12-17.
- SCIWAY.net. (2017). Greenville SC architects and architecture firms. Retrieved from <http://www.sciway.net/sc-architects/greenville-architects.html>
- Sheta, A., & Turabieh, H. (2006). Sheta, A.: A comparison between genetic algorithms and sequential quadratic programming in solving constrained optimization. *Journal of Artificial Intelligence and Machine Learning (AIML)*, 6(1), 67-74.
- Social Affairs Unit Magazines Limited. (2017). Nicholas grimshaw PPRa, eden project biomes. Retrieved from [http://www.standpointmag.co.uk/node/4728?page\\_num=2](http://www.standpointmag.co.uk/node/4728?page_num=2)
- Strauss, A., & Corbin, J. (1990). *Basics of qualitative research: Grounded theory procedures and techniques*. Newbury Park: SAGE.
- Strauss, A., & Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory* (2nd ed.) Sage Publications.
- Surjajovic, S., & Bingham, D. (2015). Levy function N.13. Retrieved from <https://www.sfu.ca/~ssurjano/levy13.html>
- The MathWorks. (2017). Matlab: The language of technical computing. Retrieved from <https://www.mathworks.com/products/matlab.html>
- VERBI GmbH. (2017). MAXQDA: The art of data analysis. Retrieved from <http://www.maxqda.com/>
- Vickery, P., Skerlj, P. F., & Twisdale, L. (2000). Simulation of hurricane risk in the U.S. using empirical track model. *Journal of Structural Engineering*, 126(10)
- Weisstein, E. (2002). *CRC concise encyclopedia of mathematics* (2nd ed.) CRC Press.
- Welch, A. (2016). Massimiliano fuksas, architect. Retrieved from <https://www.e-architect.co.uk/architects/massimiliano-fuksas>
- Williams, C. *Definition of the geometry of the british museum great court roof*. Unpublished manuscript.



Windley, M. (2016). Melbourne victory staying loyal to AAMI park despite being on verge of 'outgrowing' home. Retrieved from [www.heraldsun.com](http://www.heraldsun.com)

Wonoto, N., Baerlecken, D., Gentry, R., & Swarts, M. (2013). Parametric design and structural analysis of deployable origami tessellation. *Computer-Aided Design and Applications*, 10(6), 939-951. doi:10.3722/cadaps.2013.939-951

Woodbury, R. (2010). *Elements of parametric design*. New York: Routledge.

YP LLC. (2017). Anderson, SC architects. Retrieved from <https://www.yellowpages.com/anderson-sc/architects>

Zhu, X. (2014). *Wind load analysis on a high-rise square-plan building* (Master of Science).