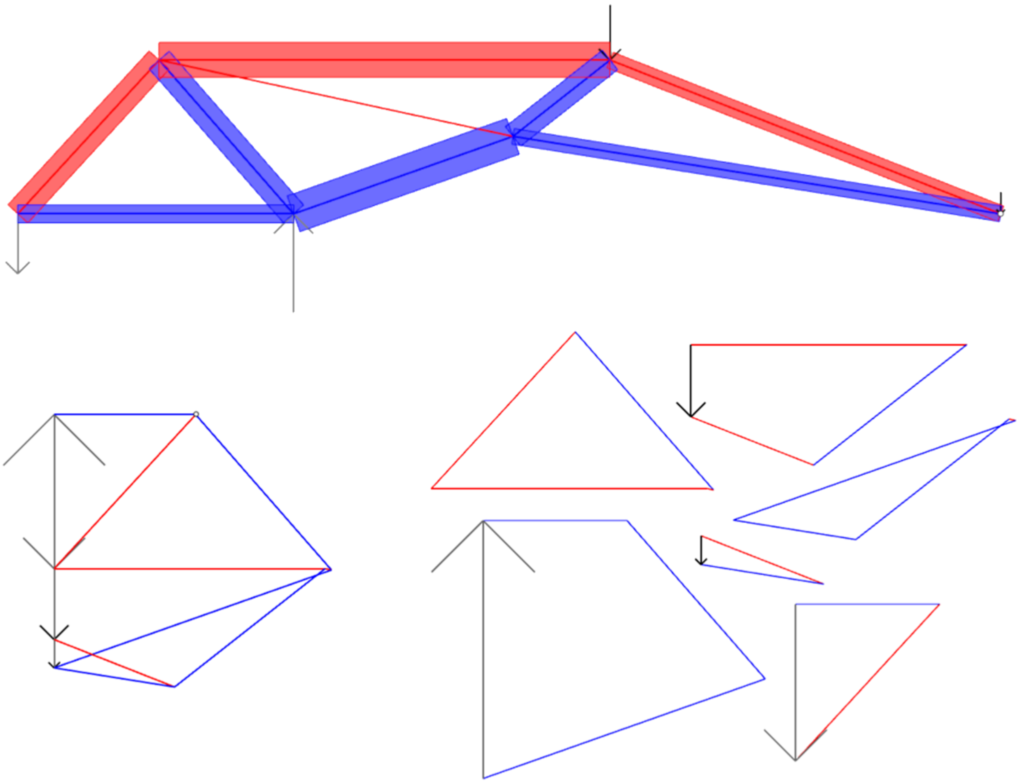
StructuralComponents 8

Facilitating early structural integration in conceptual building design with a force flow design tool

Floris Bruinsma

January 18, 2021



StructuralComponents 8
Facilitating early structural integration in conceptual building design with a force flow design tool

by

Floris Bruinsma
Student number: 4326458

in partial fulfillment of the requirements for the degree of

Master of Science
in Building Engineering

at the Delft University of Technology and
in cooperation with White Lioness technologies,
to be defended publicly on the 28th of January 2021.

Thesis committee:	Prof. dr. ir. J. G. Rots (chairperson)	TU Delft
	Dr. ir. J. Coenders (supervisor)	White Lioness technologies
	Dr. ir. M.A.N. Hendriks	TU Delft
	Ir. R. Crielaard	TU Delft

Preface

This is the report of the Master's thesis of Floris Bruinsma on the development of a force flow design tool for conceptual design of structures. The thesis is part of the ongoing research project StructuralComponents, which focusses on the development of early-stage design tools for buildings. This research project is a collaboration between the faculty of Civil Engineering and Geosciences of the Delft University of Technology and White Lioness technologies. The graduation committee consisted of the following members:

Prof. dr. ir. J.G. Rots (Chairperson)
J.G.Rots@tudelft.nl
Delft University of Technology
Faculty of Civil Engineering and Geosciences

Dr. ir. J.L. Coenders (Supervisor)
jeroencoenders@white-lioness.com
White Lioness technologies
Founder and CEO

Dr. ir. M.A.N. Hendriks
M.A.N.Hendriks@tudelft.nl
Delft University of Technology
Faculty of Civil Engineering and Geosciences

Ir. R. Crielaard
R.Crielaard@tudelft.nl
Delft University of Technology
Faculty of Civil Engineering and Geosciences

Acknowledgments

First of all I would like to express my sincere gratitude to my graduation committee, consisting of prof. dr. ir. J.G. Rots, dr. ir. J.L. Coenders, dr. ir. M.A.N. Hendriks, and ir. R. Crielaard. Their guidance, comments and assistance has helped me navigate through the process and has strengthened this research project. Special thanks go to my daily supervisor Jeroen Coenders who took the time to discuss the topic many times and gave me the initial freedom to define a research project that really interested me. I would also like to particularly thank Roy Crielaard for helping me with the setup of the user experiment.

Many thanks go to Chris Clarke, Giancarlo Torpiano, and Lennert van der Linden of Arup, Diederik Veenendaal of ETH Zürich and Summum Engineering, Rayaan Ajouz of Bouwen met Staal and ABT bv, Roel Schipper and Yi Xia of the TU Delft, David Niño Romero of the Monterrey Institute of Technology, and my colleagues Harm Houwink, Maarten Mathot, and Leah Dierker Viik of White Lioness technologies, who all participated in the user experiment, despite it taking place mostly in the middle of the Christmas holidays, with special thanks to Maarten Mathot who also tested the experiment and provided valuable suggestions.

Finally I thank all colleagues, friends, and family who motivated me throughout the project, my wonderful girlfriend who has supported and helped me continuously, and my parents for believing in me and supporting me for as long as I can remember.

Summary

The architecture, engineering and construction industry is responsible for a large share of the current greenhouse gas pollution, resource scarcity and material waste [International Energy Agency, 2018], and must therefore innovate greatly to ensure a sustainable future. A building design shift towards more efficient structural forms should be one of the vital measures to tackle this challenge. The conventional building design process can often be defined as sequential, meaning that the architect makes the conceptual design, after which a structural engineer designs a structure that fits the established geometry [Verbeeck et al., 2016]. This structurally uninformed and likely inefficient geometry must be compensated with an abundance of structural material, leading to a higher material volume, which is directly related to pollution and resource scarcity. One of the main reasons why this sequential design process can in practice not completely be replaced by a process where structural aspects are involved from the start, is because of the limitations of current digital design tools, and their inability to deal with the limited amount of design information available in early stages [Mueller, 2014].

The objective of this project is to devise and develop a design tool that lets the user explore different force flows during conceptual design, with the goal to inform different geometries, thereby realizing an integrated design process. Key characteristics that this tool must incorporate to be successful are defined as feedback (the ability to provide rapid and reliable analysis results), guidance (the ability to guide the user towards better designs), design freedom (the ability to allow users to make their own decisions and use their own expertise) and structural overview (the ability to make the general structural behaviour clear instantly).

Graphic Statics has been identified as a suitable method to model and compute the force flow with, and has been incorporated as the base of the tool. The method is based on a system of normal force members connected with pin-joints and uses joint equilibrium conditions to solve forces graphically, resulting in an intuitive, fundamental and fast solving process. A less

exact method based on drawing vectors to visualize an equilibrium, and a method revolving around a system of differential equations were also analyzed, but were abandoned respectively because of difficulties in applying a proper feedback mechanism, and an abundance of design variables making optimization and thus guidance impossible to properly incorporate.

A workflow has been devised where the user first identifies and investigates a design problem manually, creating an initial design that must be used as input for the tool, which includes a definition of joints, members. loads, supports and certain boundary conditions. Due to the nature of Graphic Statics, this initial design must be statically determinate.

The tool has been developed in Grasshopper, a parametric and associative modelling plugin of Rhinoceros [McNeel, 2020a], which provides a platform that meets the defined development criteria of accessibility, real-time modelling, geometric flexibility, extensibility, and presentation independence. The developed prototype bearing the name of GSDesign, consists of custom components scripted in C# containing the main functionality, as well as Grasshopper clusters for specific visualization sequences. GSDesign facilitates the design of 2D truss-like structures, which can be interpreted as force flow designs, whose efficiency is quantified in the total load path $\sum |P_i| * L_i$, where P_i and L_i respectively concern the force in and length of member i.

The feedback is generated in real-time, and is presented in a force diagram, form diagram, unified diagram, and/or as numerical values, providing precise and accurate results, if the created truss model is representative for the actual design. Guidance is incorporated intrinsically through Graphic Statics, but also by the incorporation of an optimization process using the total load path value and the genetic optimization component Galapagos native to Grasshopper. Design freedom is ensured through the general setup of the tool, which allows for maximum utilization of the design flexibility that Grasshopper offers. The tool provides a clear structural overview through the visualizations that characterize Graphic Statics, which can be customized to preference by the user. A special cluster has been developed to support the workflow of transferring the manual design to the digital canvas.

The basic functionality of the developed prototype has been validated by two benchmark cases and a case study, which have shown that the computed feedback is correct, and that the design workflow including the optimization process works smoothly and correctly. Additionally, a user experiment has been set up to test the functionality and workflow of the tool in a simple design case carried out by structural engineers to ensure its practical value. The eleven participants were asked to first create an initial design by hand, and to subsequently reproduce that design in GSDesign and improve it with optimization. The use of GSDesign visibly led to new insights towards more

efficient structural forms, which was supported by an average drop of 31% in structural material needs when comparing the fitness data of the optimized design with the initial design. Also the participants on average rated their own optimized design more than 40% higher on structural efficiency than their initial manual design, importantly without compromising on practical feasibility. These results strengthen the claim that GSDesign is an effective tool to create and explore conceptual structural designs, which is a view acknowledged by the participants who rated the practical value for structural design and educational value of the tool respectively on average at 4.5 and 5.3 out of 6.

Contents

Chapter 1

Introduction

This research project aims to facilitate an improvement of the early building design process, making a positive contribution to the global environmental issues that the building industry is partly responsible for. The background of these problems and other ongoing trends and issues in the building industry will be handled in the first section, followed by an analysis of the building industry, focusing on the benefits and characteristics of an early integrated design process compared to the conventional sequential process. This section also notes the necessity of computational design tools to support this process and outlines what characteristics are vital for newly developed design tools. The chapter is concluded with an evaluation of the findings.

1.1 Background

Over the past centuries the rapid advancement of technology has fueled massive developments in society in terms of safety, health and more generally prosperity, making mankind more successful than ever [Bregman, 2019]. However, this success has not been obtained without consequences which is illustrated by major global problems like pollution, resource scarcity and material waste. Such problems force engineers in all fields to improve, optimize and rethink mechanisms, structures and processes in order to sustain our society and provide a bright future for the next generations on planet earth. The building industry is no exception here and must play a big role in the road to a sustainable society, given its relatively large contribution to these global problems [Block, 2018].

However, a deeper analysis of the building industry shows that sustainability is not the only value being pursued. For starters, society is becoming increasingly risk averse, demanding safer structures and less structural failures [Coenders, 2011]. Also building projects are becoming more complex due to more demanding clients and the increasing possibilities of advanced computational design tools. On the contrary clients tend to focus on low costs and quick design and construction of a building, which actually has been identified as a factor that decreases structural safety [Terwel, 2014]. Clearly these trends contradict each other in some ways, meaning there are opposing interests at play which can only be solved with a significant increase in the efficiency of the architecture, engineering and construction (AEC) industry, concerning as well the actual designs as the design processes behind it.

As Coenders [2011] importantly states a “design freedom versus information volume paradox” exists in the design process. This paradox is illustrated in Figure 1.1, and entails that the design freedom is largest when the information volume is smallest, and vice versa. The more popular MacLeamy [2010] curve poses this principle in a more economical way by plotting both the ability to impact design (design freedom) and the costs of design changes against time in the design process. Simple plots like these clearly illustrate that decisions made in early design phases have the biggest impact on the final design and that focusing more on the conceptual phase and acquiring high-quality information early-on generally leads to better and more efficient designs. Because these early design phases are of such importance, its current handling in the building design process will be discussed in the upcoming section. Thereafter, current computational tools that are now used during conceptual design are discussed, including also more academically oriented tools that are not commercially available.

1.2 The building industry

This section aims to describe how the current building industry functions, and will focus on when and how the structural aspects of a design are brought into play. The first part contains an explanation of the early integrated design process compared to the conventional sequential process, and analyzes how computational tools can best aid this integrated process. The section that follows poses the idea of a digital design story and expresses its importance in the building industry. Finally, an overview is given of the problems related to sustainability in relation to the building industry and structural design.

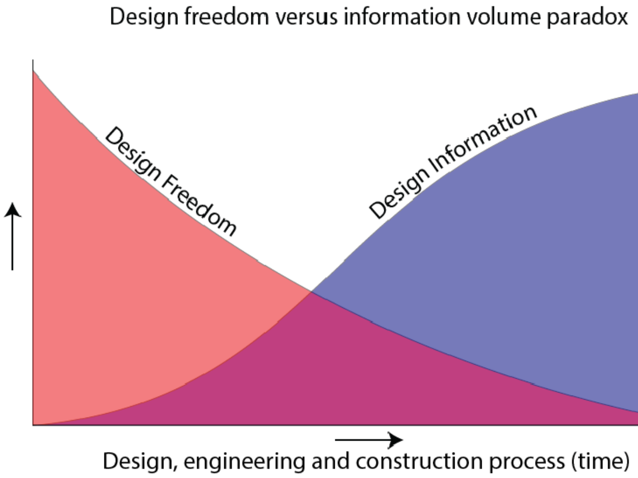


Figure 1.1: An illustration of the design freedom versus information volume paradox, based on Coenders [2011] and MacLeamy [2010]

1.2.1 Early integrated design

The current building design process can be described as sequential. First the architect makes a design, usually delivering a building model with a certain geometry. The structural engineer then designs a main load bearing structure suitable for this geometry and analyzes it, using for example finite element method (FEM) analysis. If it turns out that the initial geometry is structurally inefficient, changes can be made, and other structural concepts or architectural designs may be explored, resulting in a slow and heavy iterative process. In terms of the typical digital tools of the architect and the engineer, a distinction exists where architectural tools focus on geometry generation and often neglect performance, while the tools of the engineer focus on analysis and require a determined form, making the sequential nature of the process inevitable.

This design process has worked reasonably well for decades, but completely misses big opportunities in for example structural efficiency and safety. It is the overall geometric form that most affects the structural behavior, which is formulated most evidently by Mueller [2014]:

> History, theory, and nature show that for structural performance, overall form matters much more than material, member sizing, or internal topology (Thompson, 1942; Zalewski et al., 1998; Larsen & Tyas, 2003; Allen & Zalewski, 2010).

Therefore early involvement of structural aspects in conceptual building design offers the most potential for structural efficiency and achieving the design goals. In terms of clear benefits from early integrated design involvement of the structural designer, the following characteristics apply to the resulting buildings:

- Improved structural efficiency, leading to reduced environmental impact and construction costs due to material savings
- A certain architectural richness and elegance, due to lean shapes and the analogy to forms of nature [Tamai et al., 2019]
- Inherent safety and longevity through designs that are safe by nature [Mueller, 2014]

One of the reasons why a truly integrated design method cannot yet be adopted for large-scale projects is that currently no tools support this method, or are very limiting. The conventional structural tools that are used today have all evolved from the sequential design method and focus on analysis, while the proposed method needs tools with completely different characteristics that assist during the conception of a structural design. In order to assess and develop new design tools, first a formulation must be given of the required characteristics of the these tools to achieve the proposed integrated design process. The main characteristics for such tools are:

1. Feedback
2. Guidance
3. Design freedom
4. Structural overview

Feedback With feedback the ability to rapidly respond to structural forms and to perform real-time analysis is meant. There are some current tools that perform quite well in this area, as will be discussed further on. The feedback itself must be reasonably accurate, but does not have to be very precise; the results serve as an indication to inform early design, while more precise analysis methods can properly verify the structure in later stages using more design information.

Guidance To provide guidance means to actually suggest new forms, optimize geometries and perform useful form-finding. This trait has proven troublesome to apply in computational tools, since it requires a certain adaptability, flexibility and even admission of subjectivity, which generally does not match well with computation [Coenders, 2011]. By properly applying guidance in computational design tools, the role of the structural engineer as conceptual designer will be fully supported, going further than just very fast analysis of uninformed structural shapes. The requirements of feedback and guidance are originally identified as key structural design tool features by Mueller [2014] in her dissertation on the *Computational exploration of the structural design space.*

Design freedom With design freedom the ability to allow the designer freedom during the conceptual design is implied. This entails allowing the designer to follow a certain design direction, incorporating his own knowledge, experience and preferences. In this sense the design responsibilities are shifted from optimization tools and expensive FEM tools back towards the human engineer. This should result in more logical structures, allowing a better understanding of the structure and providing inherent safety. Overall this would also reduce mistakes and improve communication regarding the structure, due to its increased logic and simplicity [Wiltjer et al., 2018].

Structural overview The ability of facilitating structural overview means that a clear and understandable visual overview of the structural behavior of the building can be generated, which can be of great value, but currently often not available during designs. Instead of basing design decisions on tables and single maximum values exported by FEM models, a good visual picture can be extremely helpful to simply grasp the structural behavior of different concepts while considering design variants. Having this behavior clear and available therefore helps to make design decisions, but also improves internal and external communication about the structure.

1.2.2 Digital design story

Nowadays, much of the initial structural design ideas are simply sketched out on pieces of paper. This has been a tried and tested design method for centuries, and is commonly used because of the almost unlimited freedom a blank piece of paper provides and its ease of use. However the information on a piece of paper is lost with the paper itself, making it difficult to learn from it in the future and reuse the obtained knowledge. Moving the conceptual design to a digital environment could change this and revolutionize structural design. Storing the information obtained during conceptual design provides endless possibilities for future applications like machine learning. This would make that structural design should not anymore be started from scratch but gets a head start by using the information of past building projects, enabling concepts like collective intelligence and computational intelligence [Coenders, 2011]. To allow for such developments it is essential to digitalize these processes and properly store the information that is generated throughout the design. Currently, this is insufficiently done, usually only storing the information of the final model, failing to incorporate the design process towards that final design and the different design variants [Bovenberg, 2015].

1.2.3 Sustainability

In order to assess the challenges of sustainability in the building industry, first the definition of sustainable development must be clear. The most widely accepted definition of sustainable development is the one incorporated in the Our Common Future report [Brundtland, 1987], which states:

> Sustainable development is development that meets the needs of the present without compromising the ability of future generations to meet their own needs.

To structure the analysis of the building industry in regard to this definition, the analogy of the three P's, or the triple bottom line is used [Elkington, 1997]. The three P's in this analogy stand for People, Planet and Prosperity (originally defined as Profit), which must be in balance to ensure sustainable development. People considers the societal aspects, and the well-being of the people involved in the processes. Planet concerns environmental aspects and the ecological footprint of a company, industry or sector. Prosperity stands for the economic aspects and represents the financial performance, not only looking at internal profits, but also shared societal prosperity and equality [Brown and Rasmussen, 2019]. Not all three aspects are evenly present in this project, with the focus lying on the environmental and ecological effects. In order to concretize what sustainable development means in the context of this research project, three important sustainability challenges related to the early building design process will be discussed, analyzing the implications and opportunities for People, Planet and Prosperity.

Construction-related carbon emissions 11% of annual energy-related CO_2 emissions originate from the construction industry, and 28% from building operations [International Energy Agency, 2018]. Clearly, buildings are responsible for a big share of the total greenhouse gas emissions, and thus play a big part in minimizing and halting global warming. It has been identified that the manufacturing of cement and steel is responsible for most of the construction-related emissions, since these materials have high manufacturing emissions and are used in large quantities [International Energy Agency, 2018]. In the building sector legislation and initiatives mainly focus on reducing the direct and indirect building operations emissions, while the embodied carbon in buildings remains unaddressed. This makes the relative global share and thus the importance of the embodied carbon in buildings and construction bigger. Building material carbon emissions can be reduced by using more sustainable materials like wood or bamboo, but such materials have their own characteristics and are not suitable for every application. Additionally, it might take a long time to research and develop these new materials and have the industry adapt to it. Research into more sustainable and renewable materials is essential although not the only solution path. Another approach is to save building material by increasing the efficiency structures, decreasing the need of structural material in general. Adopting proper optimization and form-finding during the earliest stages of design can have great impact on material use and is relevant for every chosen material. Currently, tools can effectively optimize parts of the structure, but no practical tool integrates such optimization directly with conceptual design were decisions have most impact on the final design and where most structural material can be saved. In terms of the triple bottom line principle less material use will most notably have benefits for the Planet by reducing emissions, resource scarcity and eventually construction waste, and for Prosperity by reducing material costs and improving building longevity.

Data and sustainability In order to properly assess the implications of early design decisions for environmental factors, it is crucial to have the right information. Such information cannot be defined by stand-alone analysis, but can only be fully obtained by analyzing the data of earlier designs and the implications certain decisions had on the final result. The theory behind this is the same as for the digital design story of Section 1.2.2, meaning that by digitally documenting the conceptual design process today, sustainability based performances can be evaluated and applied on future projects.

Communication and clarity in the design process As discussed in Section 1.2.1 the inefficiency and chaos of the building design process is negatively influencing the ability to achieve design goals. When one regards sustainability just as one of these design goals it becomes clear that the communication and clarity in the design process must be improved to achieve more sustainable designs. The exact same mechanism as discussed in Section 1.2.1 applies here, meaning that early structural design integration, helped by design tools that embrace the four defined characteristics, would enable the improved design process where sustainability-related goals can be effectively incorporated. In terms of People, Planet and Prosperity, all could benefit from a better and clearer design process.

1.3 Current tools

The digital tools used today in the field of structural design have been analyzed to paint a picture of the state of the industry and assess its concepts in the face of conceptual structural design. A distinction is made between the conventional geometry and analysis driven tools, and more recent developments of structural tools that are specifically geared towards early design.

1.3.1 Geometry driven tools

Geometry driven tools are essentially architectural tools and originate from the earliest Computer-Aided Drafting (CAD) tools in the 1980's [Mueller, 2014]. Its functionality has progressed from documenting towards generating designs from the conceptual to the detailed phase, resulting also in the evolved definition of CAD as Computer-Aided Design. High interest in complex geometries has led to big developments in this area, resulting in powerful 3D modelling capabilities which allows the creation of impressive forms. Parametric and Associative Design (PAD) is a relatively new development that couples 3D modelling software with intuitive scripting capabilities. Such software allows for much more control, flexibility and reusability of the model. The most notable current examples of PAD software are Grasshopper and Dynamo, which are respectively embedded in the Rhinoceros [McNeel, 2020a] and Revit [Autodesk, 2020a] platform.

1.3.2 Analysis driven tools

Analysis driven tools are the tools used by engineers to assess structural designs, and necessarily mirror architecture tools in their capacity for handling complex geometries. Almost all analysis tools used in practice are based on the finite element method (FEM), which can among other things calculate the stresses, deflections, and dynamic behavior of a structural model. Most new development efforts are focused on increasing the accuracy and speed of the process.

The conventional FEM analysis workflow can be quite time-consuming, since it requires the generation of a separate structural model based on the geometric model defined by the architect. Also the fact that a complete geometry is required before any feedback is given means that such tools have little use in the conceptual design phase. Additionally, the processing power that computation processes require can be quite big when fine meshes are used on large models, although this is becoming less of an issue with the growing capabilities of modern day computers. In conclusion, FEM tools are specialized in providing accurate feedback for the most complex models, but are in general not suitable for conceptual design since they do not offer fast and flexible explorations and lack capability of providing guidance in geometry and topology generation.

1.3.3 Conceptual structural design tools

In different fields efforts are made trying to integrate structural aspects into conceptual design. Recent developments vary from ready-to-use practical tools, to more theoretically-oriented tools and even conceptual descriptions of a new computational infrastructure. Relevant examples of such developments are presented here, discussing their characteristics, together with their shortcomings, benefits and opportunities. The term conceptual design tool is relatively broad-defined here, since the goal is to explore all structural tools playing any role in early design and investigate their potential in conceptual design. The following developments have been included in this section:

- Real-time numerical analysis
- Integrated numerical analysis
- Form-finding
- Graphic Statics
- Optimization techniques
- Visualization strategies

Real-time numerical analysis

There are multiple tools that have the functionality to provide rapid or real-time feedback. This way the structural modelling and analysis process are more or less intertwined, providing the advantage of speed with respect to conventional methods. It would also raise the structural understanding of the engineer and improve the decision-making process [Hohrath, 2018]. However, most of these tools are still embedded in conventional FEM analysis software, bearing with it their disadvantages.

Prominent examples of such tools are CSI Model Alive [Computers and Structures, 2020] and Arcade [Martini, 2009], of which an analysis is entailed in Appendix A.

Integrated numerical analysis

Instead of providing real-time feedback in analysis-based tools, it is also possible to integrate numerical analysis in geometry-based tools. This allows for a very smooth and direct workflow by making the (manual) transfer of design information from architectural towards engineering software redundant. A practical disadvantage of this method is that each particular application is tied to their own software environment. Theoretically, it is usually undesirable and irrational to use the geometric architectural design directly as the structural design, meaning that an extra definition of certain assumptions, boundary conditions and structural properties is required which generally results in a manual translation or workaround. The most striking examples of tool that offer integrated numerical analysis are Karamba3D [Preisinger, 2020], Robot [Autodesk, 2020b], and compas_fea [Van Mele et al., 2017], BHoM [Fisher, 2020], and Geometry Gym [Mirtschin, 2020], which are described and analyzed in Appendix A.

Form-finding

While the previous categories are focused on a more direct and fast workflow of the current system, form-finding actually embraces the principles of a guidance-based approach. These tools are limited to cable, membrane and shell structures, but do allow the exploration and synthesis of structurally sound geometries. Various techniques and algorithms are used to find such structural forms that rely (almost) fully on axial forces. Appendix A contains an analysis of certain key examples of existing tools and techniques, which include CADenary [Kilian and Ochsendorf, 2005], RhinoVAULT [Rippmann et al., 2012] and Dynamic Relaxation.

Graphic Statics

Graphic Statics is a rather old method to graphically analyze forces in certain structures, with its first documented use dating back to the 16th century

[Stevin, 1586]. During the 19th century the technique became more widely used by the scientific and engineering elite, and was formally introduced to the field of structural engineering by Karl Culmann [1866]. In this century its popularity rose quickly, due the fact that it was perfectly suitable for the cast-iron pin-jointed structures of the time. Graphic Statics allows structural analysis without the need of manual calculations or intensive numerical methods and visually shows the equilibrium state in a comprehensive manner. Graphic Statics only works on statically determinate structures with linear members with only axial forces, causing a decline in popularity of the method when reinforced concrete construction became more popular in the 20th century. The increasing computational power of computers and the subsequent birth of FEM analysis accelerated this decline even more. Figure 1.2 shows how Graphic Statics works when applied to represent the equilibrium of a 2D rope system.

After a long period of absence in engineering practice, the interest in Graphic Statics is back on the rise due to the potential the technique has when combined with the possibilities of modern computation. There are many developments in different directions that combine Graphic Statics with the computational state of the art to improve the structural design process. Most of the developed tools are only applicable on 2D cases and only work on preset configurations. However, there are also several promising research efforts towards more general 3D computational tools and Graphic Statics based optimization methods. The state of the art is discussed below.

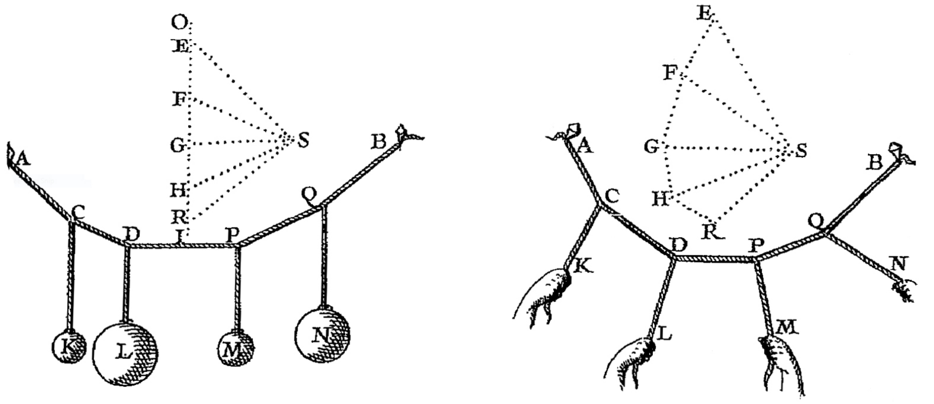


Figure 1.2: The basics of Graphic Statics as illustrated in *Nouvelle Mécanique ou Statique* [Varignon, 1725] representing the static equilibrium of two rope systems in different loading situations, with the force diagrams in dotted lines

Interactive 2D tools Interactive tools make use of computational power by automatically drawing of the reciprocal diagrams, allowing real-time manipulation of the force or form diagram. These tool use the exact same theory that was used in the previous two centuries, but then applied in a digital tool realizing a much more direct workflow with fast feedback and simple form-finding. However, these tools only work on predetermined cases and do not incorporate a general setup to design with Graphic Statics from scratch. Notable developments include the applications Active Statics [Greenwold et al., 2003], RhinoStatics [Shearer, 2010] and eQuilibrium [Van Mele et al., 2009].

3D Graphic Statics 3D Graphic Statics is a more recent development and tries to nullify the general criticism that Graphic Statics can only be applied on 2D structures. In 1864 Maxwell already observed that 2D form and force diagrams are actually the projections of 3D polyhedral diagrams, showing that Graphic Statics is also valid in 3D and that the 2D version is just a special case of 3D Graphic Statics. Several efforts have been made to develop computational 3D Graphic Statics tools, which can fundamentally either be vector-based or polyhedron-based. The most notable work is the computational design framework dissertation on the subject by Lee [2018] accompanied by compas_3gs, the computational implementation of the theory which is based in the COMPAS framework, that is discussed in Appendix A.

Optimization techniques

The impact that structural optimization has had on building design is small compared to its rich research history. The field of optimization originates from the desire to minimize material use in simple beams and frame structures, resulting in analytical solutions by Galileo and Michel in respectively 1638 and 1904. These solutions formed the basis of structural optimization, but do not offer a general approach for the optimization of any structure. In the 1960s a numerical method towards optimization was developed, using systematic iterations to find an optimum. There are three general classes of numerical optimization used today, namely:

- Size optimization
- Shape optimization
- Topology optimization

Size optimization only allows the optimization of member cross-sections for a given configuration. Shape optimization was an important development in the field of structural optimization, since it optimizes the overall structural form. It is therefore more relevant for conceptual design than size optimization. Topology optimization is a method to define the optimal connectivity of members in a structure [Mueller, 2014].

In general optimization algorithms try to minimize a certain objective function, while satisfying certain set conditions. A distinction can be made between gradient-based optimization and heuristic optimization, with the latter being more flexible and suitable for so-called messy problems with multiple local optima [Mueller, 2014]. Heuristic optimization includes genetic or evolutionary algorithms that are the basis for optimization tools like Galapagos and Octopus [Vierlinger, 2018], which are respectively a component and a plugin of Grasshopper.

Despite the big academic interest in optimization, a lot of arguments exist against a pervasive use of it. Most of these arguments come from the fact that optimization has a very rigid nature that requires a clear problem definition with set variables and objectives, while building design problems on the other hand are very flexible, are ill-defined, have many solutions and must be actively explored to understand and define all objectives and constraints. Ideal building design should follow multiple divergence phases [Liu et al., 2003], while conventional optimization only converges. Additionally, it can be difficult or impossible to mathematically formulate certain design objectives. Think only of qualitative requirements like architectural value or spatial experience. Finally, a practical drawback is that the computational process can be very time-consuming for large-scale problems, especially in the context of the high-paced conceptual design process [Mueller, 2014].

Visualization strategies

Although FEM analysis usually does try to display the general behavior of a structural model with neat colored meshes, it does not allow for proper comparison when several variants are to be analyzed. In terms of visualization of analysis results, Figure 1.3 shows how accurate humans can perceive quantitative information through certain visualizations. FEM analysis generally uses color and density in this respect, which are the least accurate options, while a method like Graphic Statics uses position, lengths of lines, and possibly areas to visualize the same information, which are much better scoring tactics. Due to this deficiency of FEM, engineers have to use the exported maximum stresses and deflections to compare variants, leaving the general structural behavior in the shadow. This results in little understanding of the behavior which could result in poor decision-making.

In this light it is important to be able to properly visualize the performance of different variants. Based on the statement that visual displays of data make for better decision-making [Tufte, 1997], Joyce [2015] proposes a method with web-based data visuals with extensive user-freedom in how to display the content. From the same ideology Lennert Loos has developed the Grasshopper plugin Inkbeagle that aims to provide “insight in the structural behavior of multiple design alternatives and to compare these, by means of interactive data visualizations.” [Loos, 2020]

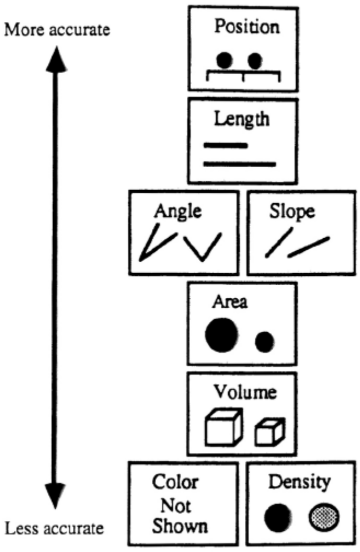


Figure 1.3: *Accuracy ranking of quantitative perceptual tasks* [Mackinlay, 1986], showing that it is easier to perceive quantitative information through positions and lengths than colors and densities

1.4 StructuralComponents

As the title implies this research project is already the eighth version of the ongoing StructuralComponents research project. The goal of StructuralComponents is to make the next step in computational modelling for structural design through the concept of structural design tools (SDT) [Coenders and Wagemans, 2006], which is essentially the idea of a powerful and highly generalized toolbox for structural design. Instead of developing design tools directly inspired by new technology, the development must be coming from the user point-of-view. The feedback and solutions that the tools provide must be rapid and indicative instead of highly accurate.

A brief analysis has been performed on all the previous individual StructuralComponents projects, which can be examined in Appendix B. The previous research done on StructuralComponents is important context for this project, which builds on the same ideology. However, this research project does not directly build on top of any of the previous projects, which is the reason that the analysis in not incorporated in the main report. Additionally, the next section includes an evaluation of the previous StructuralComponents projects in relation to the topic of this project.

1.5 Evaluation

Current tools From the amount of recent innovations in early structural design tools it is clear that the industry is aware of the shortcomings of the conventional separate geometry-based and analysis-based tools. However, most new developments seem to be focused on providing rapid or real-time feedback, basically only improving the current process. Of course integrated and real-time analysis will help with a better integration of the structure in conceptual building design, but it does not take the leap to actually assist in the generation of structurally guided geometries. Form-finding tools clearly do take this step and computational tools based on this feature are increasing in numbers and functionality, but the drawback remains that their applicability range is limited to a very specific set of structures. Graphic Statics as a method shows promise since it is visually oriented, comprehensive, computationally lightweight and could assist in form-finding as well as real-time feedback, but the developed design tools remain narrow in functionality. Its academic popularity is rising and recent developments include 3D implementations and structural optimization, which could be a good base for a more general application in conceptual design. Optimization techniques on the other hand, are in its conventional form not a rational match with the complex process of conceptual design. However, with the use of heuristic algorithms messy problems can be tackled and unexpected designs can be found and explored, hereby guiding the designer. Finally, the positive effect of good visualizations of structural performance on the decision-making process must be mentioned, concluding that conventional analysis software handles this insufficiently, especially when a multitude of structural variants are to be compared.

StructuralComponents The developed StructuralComponents tools also attempt to provide a better alternative for the conventional sequential tools, making use of the structural design tools ideology. With the exception of some more conceptual general tools, most developed tools serve only a very specific application, be it 2D high-rise buildings or certain concrete or rigid frame mid-rise structures. When using such tools a certain structural typology and materialization must be chosen from the start, not allowing later deviation from those choices or combinations of typologies. A tool that is more free in this respect and is placed in the design phase before any materialization takes place might be a valuable addition to the StructuralComponents computational toolbox.

Chapter 2

Objective

This chapter introduces and clearly defines the objective of this research project, which is followed up by defining the research questions and the scope. Lastly, the general methodology applied for this research project will be described.

2.1 Main objective

In Section 1.2.1 it has been determined that a smooth and early structural integration in the building design process has many advantages and should be aspired. An analysis of current tools has resulted in the conclusion that there is still a software gap in supporting this desired process, while newly developed structural engineering tools focus generally on rapid feedback, but lack in the support of structural synthesis and guidance. Therefore the idea is proposed to act on these limitations by the development of a conceptual force flow design tool. The use of the force flow as a form of conceptual design is chosen because it is a very fundamental representation, which can be visualized in an intuitive and understanding way, and can in theory be applied to all structural typologies. It also stays clear of any materialization and its potential issues and time investments, looking only at the essence of what is necessary during conceptual design. The force flow is intended to be both a means and an end for conceptual structural design, meaning that besides its representational qualities, the information contained by force vectors must be used to inform the design. The main research objective is formulated as follows:

> *To develop a prototype of a conceptual force flow design tool supporting the principles of feedback, guidance, design freedom and structural overview.*

2.2 Research questions

To support the main objective and to structure the research project, three research questions have been formulated:

1. *What is an appropriate method to model the force flow?*

   Since the analysis method is very much tied to the modelling method, considerations regarding the analysis or feedback method are inherently included in this research question. A method is appropriate when it meets the requirements stated in the main objective, while also providing the right options for automation making it a viable computational base. Characteristics of different force flow modelling methods and accompanying computation methods must be identified and valued.

2. *How can the computational tool take shape to support conceptual force flow design and meet the defined requirements?*

   This research question follows up on the first one, and assumes that the force flow modelling method has been chosen at this stage. The aim of this research question is to theoretically define the functionality and workflow of the tool, setting the first step for the development stage. The requirements for the computational tool include the principles stated in Section 1.2, computational requirements, but also yet unknown requirements related to the chosen modelling method. The main output after addressing this question is a proper development plan.

3. *What is a suitable framework for the development of the conceptual design tool?*

   This question addresses on which software or infrastructure the tool will be developed. To determine if a framework is suitable the different modelling, visualization and analysis requirements must be taken into account. Since the tool will be a prototype and will not contain all envisioned functionality, it is important to make sure that the tool is easily extensible and is developed in a structured and comprehensive way. Furthermore, the framework must have proper accessibility, and should be able to support real-time modelling and geometric flexibility of designs.

2.3 Scope

The following scope applies:

- **Prototype:** The objective is to develop a prototype, not a complete and fully functioning tool. The focus lies on developing the functionality, meaning that the user-friendliness of the prototype and therefore its user interface are of lesser importance. In general the aim is to incorporate the basic functions that are necessary for the tool to perform, and to compliment that basis with some additional features, which means that not all theory discussed in the research project can be incorporated. For a detailed description on what functionality is present in the developed design tool and how this tool actually takes shape, the reader is referred to Section 4.2.
- **Two dimensional:** Since the development of a 3D force flow design tool will likely take too much time, the development scope is set to a tool for 2D structures. A future implementation of 3D structures will be briefly discussed, and the development of the tool should be so to allow a later extension to three dimensional space.
- **Structural typologies:** What set of structural typologies are to be handled by the design tool depends on choices made regarding for example the modelling method. These choices will also determine the scope in this respect, meaning that the tool will therefore not be applicable on all structural typologies. Also a choice might have to be made regarding the structural typologies of the cases that will be used for validating the design tool.

2.4 Methodology

The methodology presents the plan towards and during the development of the design tool. Although the development of this tool is largely an iterative process, the process is divided into four stages for clarity purposes. These stages are:

1. Research
2. Basic functionality
3. Extended functionality
4. Validation

In the research stage the aim is to formulate answers to the research questions. During this process relevant existing tools and researches are analyzed in depth and documented. The information obtained in this stage is processed so that it is directly usable in the next stages.

The second stage is the first development stage and centers around the creation of a first version of the tool with basic functionality, which can also be referred to as the core of the tool. How exactly this core will look like, and which steps need to be followed to develop this is highly dependent on what kind of modelling method is chosen. A basis for the user interface and user interaction will also be created during this stage.

The next stage is aimed to add more functionality to the tool, getting closer to the complete design tool that it is envisioned. A choice on which functionality to add will have to be made beforehand and depends on the findings of the previous two stages and will be a consideration between feasibility and importance. Interesting directions can be the implementation of boundary conditions in the design process, and the addition of optimization techniques.

The last stage concerns the tests and validations of the developed tool. Validation will happen throughout the development process and afterwards. It entails validation using benchmark cases and a more practical case study, validating the feedback the tool provides for different design cases and testing the optimization process. Additionally, a user experiment will be executed to introduce the tool to a professional public, allowing a better verdict on the practical added value of the design tool.

Finally, the obtained results must be evaluated, comparing the outcome against the predefined research questions and objective, outputting clear conclusions and recommendations for future research.

Chapter 3

Proposal force flow design tool

This chapter describes the process and results of the first step of the defined methodology; research. It consists of two sections, aimed to respectively answer the first two research questions.

3.1 Force flow modelling

As the name suggests, a force flow depicts the flow of forces in a structure, in this research project with an emphasis on the overall behavior, disregarding the stress distribution in individual members. A load path is essentially the same thing as a force flow, where the term load path is used more in practical appliances while force flow is usually seen as something more analytical and academic, although no official difference between the terms exist in engineering practice. In this research project the terms are used interchangeably.

This section covers the research initiated by the first research question, and contains an analysis of force flow design methods in general, and an examination of how force flow modelling can be applied in the conceptual design phase. Three force flow methods are investigated and compared, to ultimately choose one as the basis for the computational design tool.

3.1.1 Force flow design

The design of load paths or force flows is an important instrument in the toolbox of structural engineers. It allows them to quickly visualize structural ideas, using the language they know best; forces. This helps them to understand the design problem, to come up with new ideas and to communicate their thoughts. Typically, a load path design consist of a series of vectors depicting the force flow in a structure from an external load to the foundation.

These vectors are displayed as arrows and can be sketched digitally or by hand. Load path designs can also be made for specific parts of a structure, in which case only a part of the load path from external load to foundation is presented.

The nature of the forces that are considered in a load path analysis may differ extensively. In the case of a pure truss structure for example only normal forces are considered, but in more varied cases also shear and bending moments may be implicitly included as mechanisms to transfer the forces. This can be confusing when one tries to analyze a load path analytically, since all load transfers are just displayed by linear arrows along the members. It is therefore important to have in mind that a load path design is not necessarily an analytical method to calculate the forces in each member, but more of a visual way to show which members are active under a load and how these members act together to transfer the external load to the foundation. Differences in the nature of the force vectors can be visualized by color, which could be applied to differentiate external versus internal forces, negative versus positive forces, and main versus secondary and tertiary load-bearing elements.

Since the concept of force flow design can be applied in different ways, two main methods will be discussed that adhere to the characteristics of load path design but which are fundamentally different.

Drawing vectors Although a clear definition does not exist for the method of engineers to draw a load path with vectors without any specific use of informative tools, in this project the term 'drawing vectors' will be used. It is the simplest and most-used form of load path design and its characteristics match the description of load path design given in the previous two paragraphs. This method is used by structural engineers on a daily basis and is critical to understand in order to properly design structures [Drucker, 2014]. Figure 3.1 shows some examples of force flow design by drawing vectors in practice.

Graphic Statics Graphic Statics has been introduced in Section 1.3.3 and can be seen as an analysis and design method for discrete structures based on geometrical relations of internal form and force instead of numerical or analytical computations [Culmann, 1866] [Maxwell, 1864]. The method is based solely on normal force members, and can therefore be a useful tool to design load paths for truss-like structures. Its application is purer and mathematically correct when compared with the previously described drawing vectors method, which is more of a practical engineering approach. The application

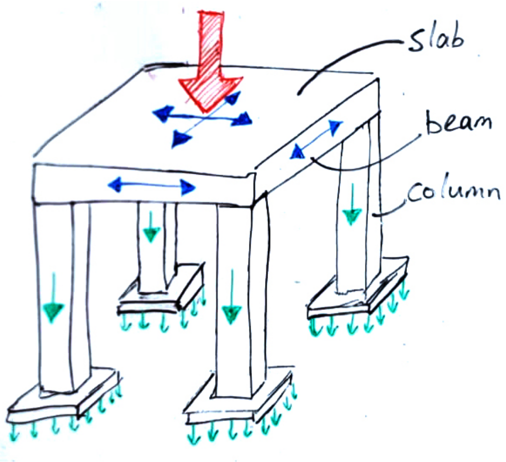


Figure 3.1: The drawing vectors method load path design method illustrated on a simple table structure, where the force flow in the slab and the beam are visualized by blue arrows, and force flows in the column and foundation slab in green

of Graphic Statics for force flow design comes with restrictions and requires more effort to set up, but the great advantage is that the results are more extensive and more reliable. Structures that do not immediately resemble trusses can sometimes be approximated as truss-like structures, making that the principles of Graphic Statics can be used for a broader range of structural typologies than initially expected.

3.1.2 Conceptual force flow design

Load path analysis can be done on existing designs, but can also be used as a conceptual design method for structures. Instead of finding the optimal load path for an existing design, one could reverse this and first design an optimal load path to later find a suitable structure that corresponds with this. Of course structural design is not an exact science, and various processes usually happen simultaneously and iterative. Nevertheless, the possibilities and advantages of using force flow or load path methods for conceptual design are undeniable.

A method like Graphic Statics or more basic principles like ensuring equilibrium can be used to draw out a load path in simple cases. This is done abundantly by structural designers, and corresponds with the way these designers think. Questions like “how is this load transferred?” and “can we guide this force differently?” are aimed to be answered by creating these conceptual load paths. Usually such load paths are simple sketches, created with pen and paper or digitally. However, these drawings are only readable by humans and contain no more information than the ink on a paper, or pixels on a screen. As discussed in Section 1.3 it could be argued that some tools now do possess the capability to help in the creation of informed load path designs, but all have their own limitations and drawbacks, and none are specifically designed for and geared towards this task. Since conceptual design is characterized as rapid and chaotic, it is often difficult for design tools to keep up with the process. Conceptual force flow design tools should therefore combine a fast workflow with the flexibility to quickly change design directions.

3.1.3 Force flow modelling

This section attempts to tell what methods could be used to model the force flow in a computational design tool. Three methods have been identified that could best be used for this, considering their general applicability and suitability for conceptual design. Every method represents a certain design direction and is presented with its limitations and advantages and its workflow is illustrated using a consistent design case. The three methods that will be analyzed and compared are:

1. Drawing vectors
2. Graphic Statics
3. System of differential equations

It is important to note that different approaches also exist within every method, which has impact on the characteristics and suitability. Therefore the three given methods are compared so the method that best suits the main objective can be chosen.

Design case In order to properly compare different modelling methods for the design tool one single design case is used. The design case concerns the auditorium of the Delft University of Technology (Figure 3.2), including the floor and the roof structure up until the service shaft, and has been chosen because its shape is rather uncommon and provides space for intelligent solutions and interesting force flow designs. The structural designs made in this chapter must be considered as if made from scratch in the conceptual design phase of the building, although all designs have been made with the actual designed structure in mind, in order to ensure their soundness. Although the design problem is presented here as two-dimensional, the actual design of the auditorium floor and roof does include three-dimensional structural behavior. The two-dimensional presentation is therefore not directly representative for the actual structure, but is however deemed a reasonably valid simplification and an interesting design case on itself. The simplification to 2D space is necessary to make it suitable for this research project, which only deals with 2D problems. The floor and roof structure of the auditorium are seen as separate structures, which are in the existing design not connected at their ends [Den Hollander, 1964], a trait which has been implemented in most of the designs in this chapter.

Figure 3.2: TU Delft auditorium building, as photographed by Aders [2012]

Figure 3.3 shows the basic setup of the design case, and ultimately presents an overview of the loads on the roof and floor structure and a definition of the resultant vertical forces.

Drawing vectors

This conventional load path design method by drawing vectors is widely used by engineers, and has been discussed in Section 3.1.1. To test its applicability as the basis of a computational design tool it is necessary to analyze how its logic translates to a digital model. Using this method the structural designers would simply digitally draw vectors in a 2D plane, depicting the load path from point of application to the foundation. This way the act of drawing a load path is transferred to a standardized digital environment, where the acquired information is stored and saved to be potentially used later. A possible advantage is that certain means of feedback can be incorporated in the design of the load path. This is however a complicated matter since load path design traditionally does not rely on specific rules, but more on the expertise and vision of the designer. Rules could be formulated to ensure vertical and horizontal equilibrium, but apart from that the design possibilities are still very large, making proper guidance difficult.

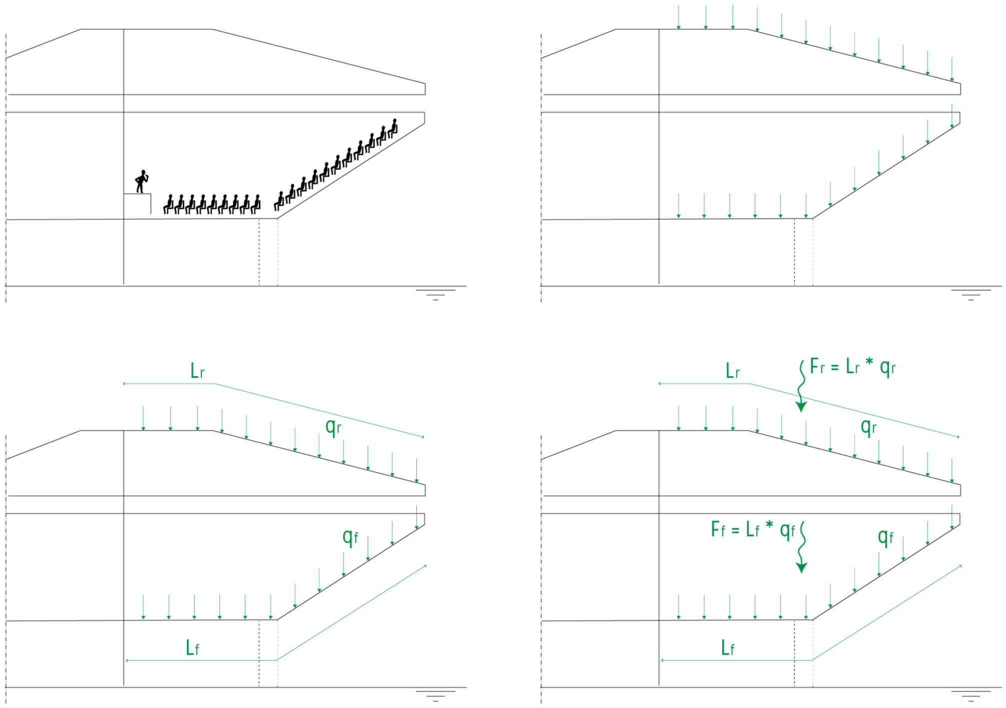


Figure 3.3: Basic setup of the TU Delft auditorium design case

Design case An example of a design with this method can be observed in Figure 3.4. The design workflow does not consist of many steps, but basically just one where the design is drawn. The constructed vectors symbolize the load path of the force flow, containing information about the direction and the magnitude of the force it represents. For elements that act mostly on homogeneous normal force the translation from vector to force is easy to imagine, but elements that transfer force by bending or shear, like the horizontal elements in this example, require more understanding. Still, even if the translation from vector to actual member force is not always straightforward, the diagram is easy to read and understand for those who just want to get an idea of the structural behavior.

Visually these vectors can differ in color, thickness and length to show differences in force magnitude, type or origin. In this example the color of the displayed vectors correspond with the origin of the load and the length with the magnitude, although this could have been done more consistent.

Additionally, the numerical magnitude of the forces in vertical elements are shown in this image next to the vectors, making it clear that in every horizontal cross-section the summation of all forces is equal to the relevant resultant load; an obligatory condition for equilibrium.

It is important to note that this method has a major downside, namely that it is difficult to actually design with. The method is suited for an analysis of a given structure and provides the ends to draw down certain considerations and thoughts, but designing and generating feedback on multiple designs would be a slow and arbitrary process. In principle, it might be more correct to define this method as more of an illustrative tool, without enough potential to be exact enough to provide trustworthy informative feedback.

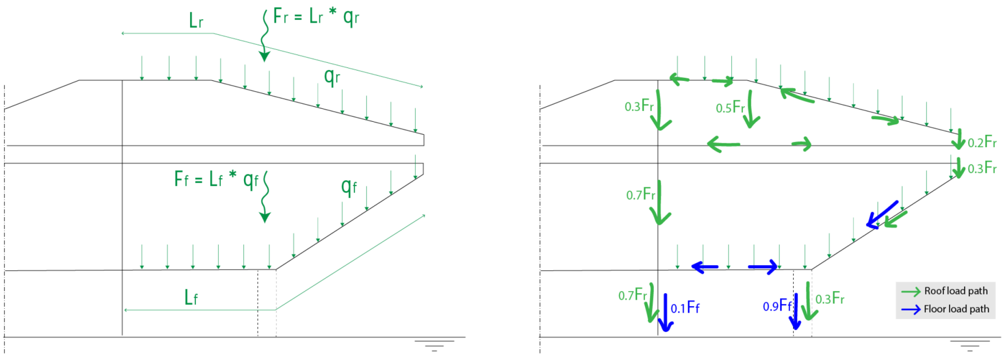


Figure 3.4: The drawing vectors method applied to the TU Delft auditorium design case, with the roof and the floor structure here combined in one structural solution

Graphic Statics

The theory of Graphic Statics and the current computational tools adopting this theory have been introduced in Section 1.3.3, and its application as a force flow design method has been discussed in Section 3.1.1. A computational force flow design tool based on Graphic Statics would be based on discrete structures based on axial forces, leaving no place for beam-like elements. The workflow based on this method would start with a design of a form or force diagram, or a combination of the two. Since the form is essentially always the starting point of a design, it is also assumed that this workflow starts with a sketch of the form of the structure, which must be converted to a valid form diagram. This form diagram is constructed in the tool with input of the designer, alongside with a reciprocal force diagram. These force and form diagrams contain all information of the force flow design, and is represented additionally in a unified diagram, which can be seen as modified form diagram that contains also the forces and visualizes them in an intuitive and insightful way. In the theory of Graphic Statics changes can be made to the form diagram from which the force diagram can be updated, and vice versa. This would be ideal to incorporate in the computational too to allow for much design freedom, but a simpler form can also be chosen where the user designs only from the form diagram while using the force diagram for solving the forces and visualizing the results. Changes to designs can be made manually and can be assisted by (genetic) optimization algorithms. The method generates feedback by using the rules of Graphic Statics, constructing the form and force diagram of a design in equilibrium based on graphical relations, and computing forces by measuring the lengths of the lines in the force diagram.

Next to a proper computation and display of the reciprocal diagrams, it is also important to link these diagrams to the right context. Forces and forms that are explored must be placed in the environment of the original design problem, making it possible to truly design a force flow for a specific design problem inside the tool, making the process faster and the results more comprehensive. This idea can be realized by incorporating an overlay of the basic design outline in the viewer, together with a proper visualization of the created designs.

Design case Using Graphic Statics as a computation method poses certain requirements on the setup of the problem. In this case that means that the uniformly distributed loads must be converted to one or multiple point loads. Also an initial design must be made which will serve as a first form diagram, which can only be based on discrete normal force-only members, and must be statically determinate. Figure 3.5 shows how these setup requirements take shape for the design case, for both the roof and the auditorium floor structure.

The initial design is used to create the form, force and unified diagrams, of which the latter two are shown in Figure 3.6. In the unified diagram a clear visual distinction is made between compression and tension forces, where the most common conception of red representing tension and blue representing compression is upheld. All bars have a fill with an opacity of 50% to ensure the visibility of all bars regardless of overlapping, and have a thickness linearly related to the force. The force lines in the force diagram have a set given thickness, while the coloring rules of the lines regarding tension and compression are equal to the unified diagram. Also the colors of the external forces are kept consistent for optimal readability.

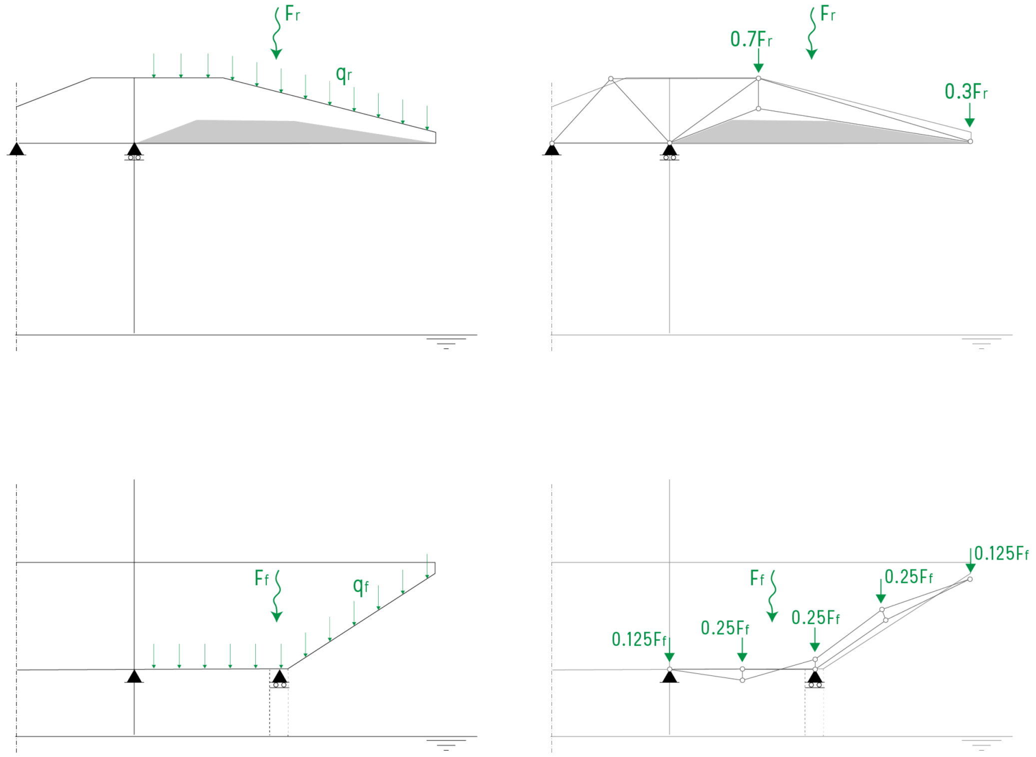


Figure 3.5: Setup of the TU Delft auditorium design problem for the Graphic Statics method: **top** The roof structure, **bottom** The floor structure

System of differential equations

With this system, the idea is that the user can design a structure, consisting of beam, tie, strut or other structural elements. Every element is represented with a programmed differential equation that describes its behavior, with boundary and interface connections taking care of the joint and load conditions. Solving the complete system with the right boundary conditions will provide the force distribution in the members, which can be visualized as a force flow. After the initial design and analysis the user can change the connectivity, node locations, element type and element properties, like for example the bending stiffness of a beam. The great variety in different

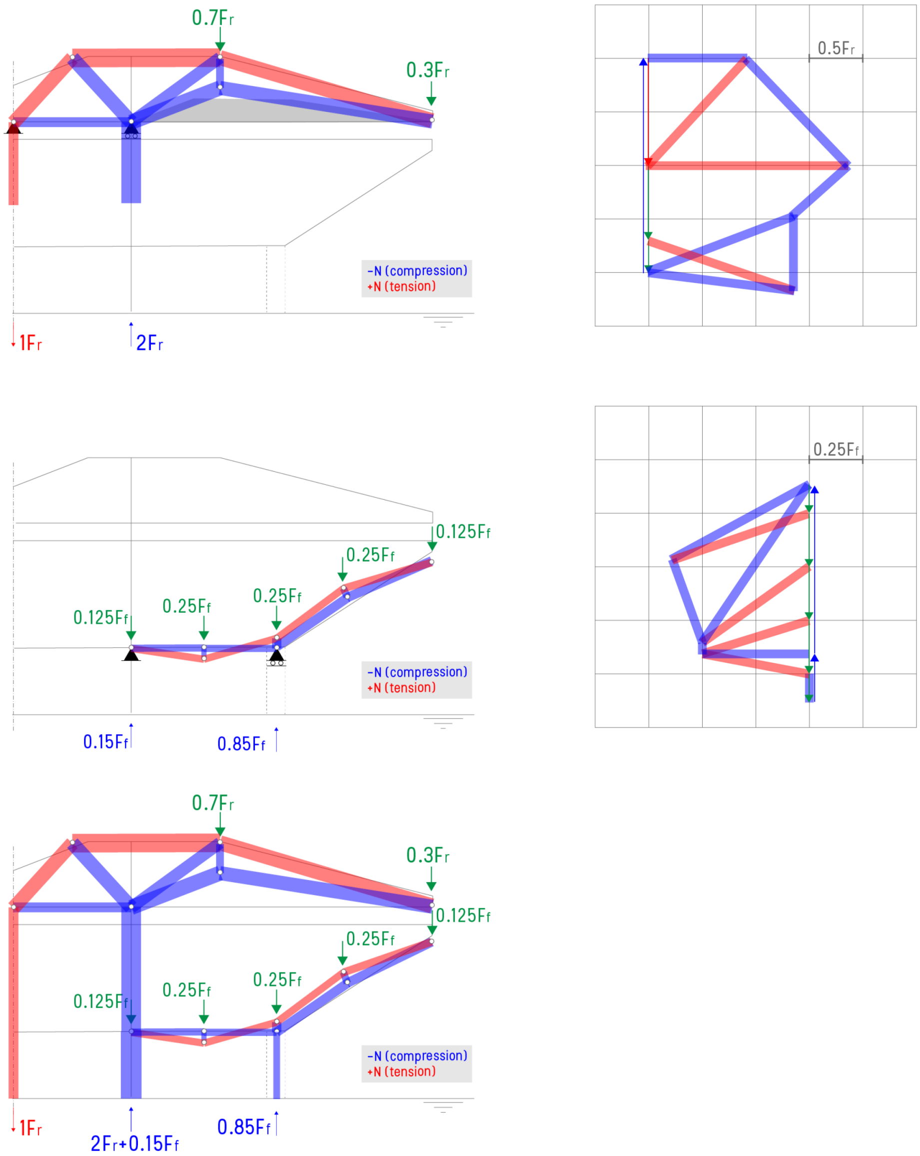


Figure 3.6: The visualization of the design of the TU Delft auditorium design case solved and visualized with Graphic Statics: **top** The roof structure unified and force diagram, **middle** The floor structure unified and force diagram, **bottom** A combination of both unified diagrams

designs and settings of variables may seem advantageous, since many designs are possible and the structural feedback will be closer to reality. However, it also leads to a slower workflow, requires more information as inputs, and provides fewer possibilities to guide the user towards efficient designs.

Design case A visualization of the system of differential equations method in action can be observed in Figure 3.6, showing the basic design setup and the visualization of the force flow as computed by the method. Because the solved model holds a lot of information, it is the most difficult to properly visualize of the three methods. The choice is made to only show the normal forces and the bending moments and leave out the shear forces, since that information is already implicitly contained in the bending moments. The basic visualization rules are the same as the Graphic Statics method, complimented by a bending moment line shown in pink, with the maximum bending moment in that element also visualized by the thickness and proportions of that line. If for a certain design the shear forces are considered to be important for understanding the force flow, the options should be available to add the shear lines in some manner to the visualization.

3.1.4 Conclusion

All of the presented methods have their own characteristics and applications, resulting in advantages and disadvantages regarding the main objective of this research project. An overview of the strong and weak aspects of the three methods as a force flow modelling tool is presented fully in Table 3.1. This evaluation is carried out with the idea of a computational tool as the main implementation, meaning that the computational aspects and possibilities for automation are taken into account. In order to make a decision on the choice of modelling and calculation method, these findings must be used and collated against the defined requirements.

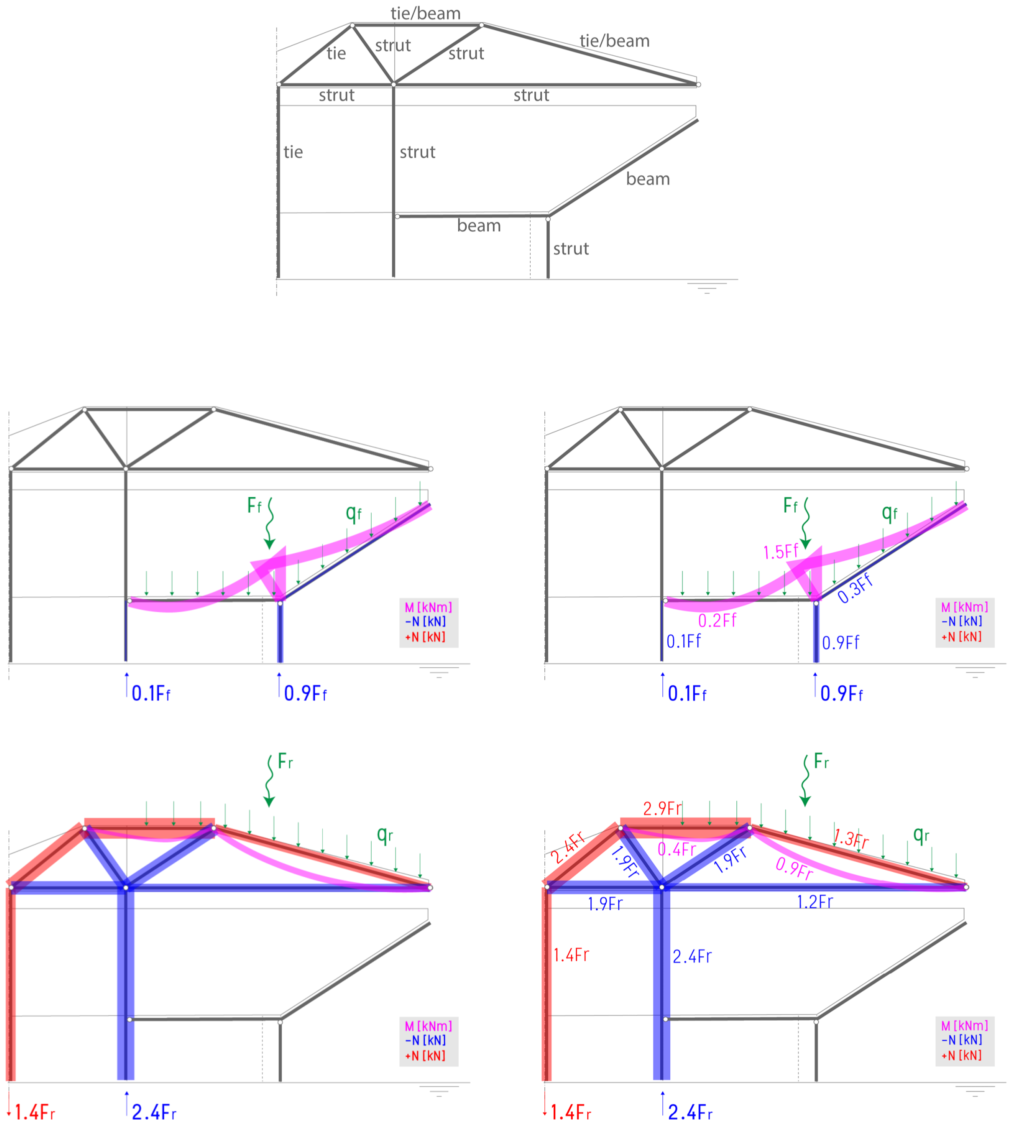


Figure 3.7: The system of differential equations method applied to the TU Delft auditorium design case: **top** Setup of the full design, **middle** Floor structure solution, **bottom** Roof structure solution.

Table 3.1: Comparison of the analyzed force flow modelling methods

	Drawing Vectors	Graphic Statics	System of DE's
Strong aspects	• Much freedom • Wide applicability • Relatively similar to current methods • Easy to understand	• Fundamental, simple and intuitive principles • Fast setup, design and solving • Suitable for optimization • Guide towards efficiency	• Much freedom • Wide applicability • High quality feedback • Steer the forces with stiffness • Equilibrium easily guaranteed
Weak aspects	• Low quality or possibly incorrect feedback • Not suitable for complex designs • Prone to human error	• Limited applicability • Strict setup requirements • Bi-directionality is difficult to achieve • Translation between structure and design	• Slow setup, design and solving • Information overload • Challenging to visualize the force flow • Risk of inefficient designs • Limited in guidance

The drawing vectors method has the attractiveness that it provides much freedom and has a very wide applicability range. However, these advantages come with a big price; it doesn't provide enough feedback to be considered an informative structural design tool in itself. It can be of great value when simply trying to visualize ideas from a structural designer on a digital canvas, but it does not use the power that computation has to offer to move the design along much further. Also since a feedback mechanism is difficult to incorporate, the method is in general prone to human errors and definitely unsuitable for more complicated designs. This method does not match the goals of this research project sufficiently and is therefore dropped.

The System of DE method can be found on the complete other end of the spectrum and can provide a great deal of information about the structural design. This method also offers a lot of freedom in choice of structural elements, connections, and placement of the elements. Therefore a lot of different structures can be made and analyzed, and unlike Graphic Statics it is very easy to reach a valid design that is in equilibrium. However, here also lies the culprit of the method. The unlimited freedom available could prove a negative factor in early design. How would it be possible to find a suitable simple solution for a design problem if so many options are available? One would have to be a very skillful and experienced structural designer to achieve that, in which case the tool would be used more for verification than design. Since the options are so endless, the method also does not match well with optimization, and a risk would be for structural designers to get lazy and just go with easy known solutions. Surely, detailed modelling options are important for later design, but during early design phases they can prove a blockage for creative innovating design. To conclude, this method does not match the objectives of this project and is not suitable as a conceptual force flow design tool.

Graphic Statics on the other hand seems to walk the fine line between proper feedback of design information, guidance and design freedom. Since the method is quite limited and fundamentally simple, it could be used efficiently in combination with optimization, which would greatly benefit the guidance capacity of the tool. Of course the to be designed structures would have to rely for a major part on normal force action, and the method is therefore not applicable for every structural typology. However, normal force action is in a structural sense the most efficient way to transfer forces, and it would not be a bad development if more structural designs would be based on normal force behavior. Buckling is an aspect that is in the basis not present in Graphic Statics but very important in structural design of axially loaded members, and must therefore be handled carefully and incorporated in the design tool as good as possible.

Another restriction of this method is that it only supports discrete structures, with a finite number of straight axially loaded elements, meaning that no structurally optimal shapes like arches and catenaries can be created. Additionally, in order for Graphic Statics to work, the designs would have to be statically determinate. Despite the narrower applicability that comes with these restrictions, Graphic Statics does provide endless structural possibilities, allowing a free exploration of statically determinate truss structures that can be solved and visualized in real-time. Also it is worth noting that in many cases a structural design that technically does not meet the right requirements for the Graphic Statics framework can be simplified in such a way that it can be analyzed using Graphic Statics while also generating meaningful results.

In conclusion, Graphic Statics is chosen as the force modelling method for the tool that is to be developed. The matter of how exactly this tool should be developed, and how to let the user interact with the tool is an important question which will be addressed in the remainder of this chapter and Chapter 4.

3.2 Requirements of Graphic Statics-based tool

Moving forward from the choice of Graphic Statics as the computational base of the force flow design tool, a more refined definition of its shape and functionality must be provided. This section exists to theoretically explore and establish how the tool is set up and used by the designer, and entails a proposal for the workflow and a commentary on how certain challenges can be overcome to meet the defined requirements.

3.2.1 Proposal workflow

An illustration containing the proposal for the workflow of the tool is presented in Figure 3.8, which consists out of five steps that will be discussed here.

In the first step a design problem exists where the force flow tool might be able to contribute. It is up to the designer to identify if the problem or sub-problem indeed matches the functionality of the tool. It is important in this stage that the designer is familiar with Graphic Statics and knows its limitations. If the tool is found suitable for the design problem, a proper

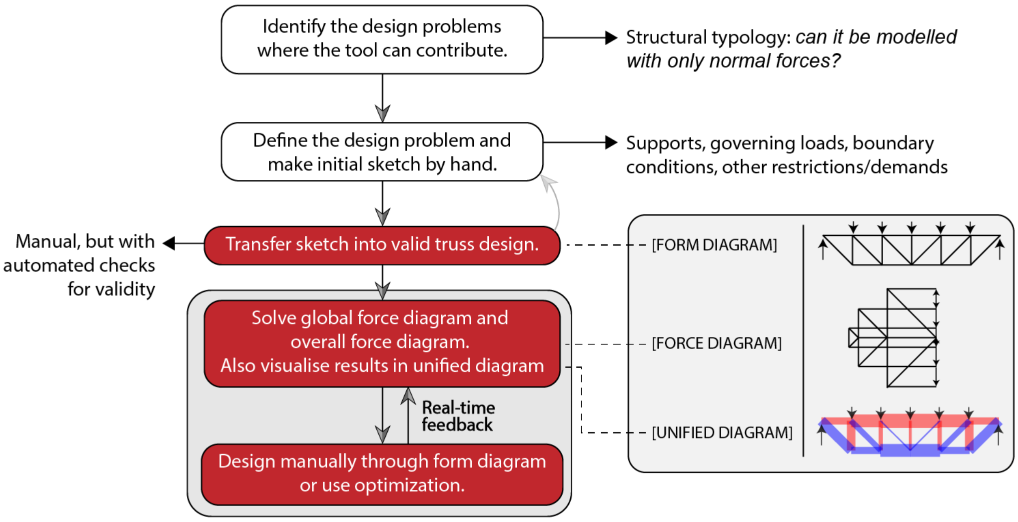


Figure 3.8: Proposal for the workflow of the Graphic Statics-based force flow design tool; all red boxes represent actions involving the force flow design tool, while white boxes represent the preparatory steps

definition of the problem must be created. Support conditions, governing load conditions, boundary conditions, and other restrictions or requirements of the design must be formulated. Then an initial design sketch must be made by hand, taking into account the whole problem definition, and making sure that the loads, supports and elements are defined in such a way that the theory of discrete 2D Graphic Statics can deal with them. Depending on the design problem, this step may require a certain expertise from the engineer. Different design directions can be explored and the designer is free to use any available method to properly inform the initial design.

Once the initial sketch design is finished the designer moves from the preparatory work to interacting with the computational tool. The characteristics of the sketch design must be implemented as inputs for the computational tool, which must be defined manually in a predefined way. The time frame necessary for this first setup could be less than five minutes if the user is familiar with the platform, but depends of course on the complexity and scale of the design. If the defined inputs return an in some way invalid design, this information is looped back to the user who must use the response to improve the design and ensure its validity. If the user inputs, however, lead to a valid design, this design is immediately solved and visualized, providing the user with multiple visualization options, customization and numerical and geometric outputs.

Fundamentally Graphic Statics is bidirectional, meaning that a change in the form diagram can be used to update the reciprocal force diagram, and vice versa. However, these relations appeared impossible to implement in the intended framework described in detail in Section 4.1.2, which had led to the choice of only allowing design changes through the form diagram. Advantageous of this workflow is that is follows the way designers logically think and design, while keeping the format of the tool structured and consistent. Even with this configuration the force diagram is still very important in the workflow; it is the instrument that is responsible for the computation of the forces in the members and is a uniquely illustrative display of these forces, which is a great addition to more conventional force visualization methods.

3.2.2 Incorporated principles

The four basic principles for any conceptual structural design tool that were introduced in Section 1.2.1 need to hold up for this tool and its workflow. This section provides a description on how these principles are incorporated in the Graphic Statics-based tool as presented here.

Feedback Feedback concerns when and how actual feedback is created, which is here achieved using the geometrical rules of Graphic Statics to inform the design. This is done at two stages in the design process, namely at:

1. The creation of the first (in equilibrium) form and force diagrams corresponding to the initial design.
2. Any manual change of the design, which automatically updates the form and force diagrams and updates the solution.

Because the computations are based on 2D Graphic Statics they are very lightweight, leading to virtually instant feedback. The forces are quantitatively visualized using line lengths, line thicknesses and also by providing the numerical value in text next to the element or force. On top of that the qualitative differentiation between compression and tension will be made with color, with blue representing compression and red representing tension. Additional feedback is generated when the design that is created by the definition of the inputs is invalid in any way. This feedback must clearly state the error that is occurring and should point the user in the right direction to solve it.

Guidance When the tool guides the user effectively towards better designs, one can speak of guidance. The principle is first of all ensured intrinsically in the method of Graphic Statics. By using Graphic Statics and adhering to its rules the designer is already steered towards efficient load transfer and minimal material use. Designing this way makes it more probable for the user to design an efficient structure than compared to using a tool with limitless freedom or without immediate feedback. The clear visualizations of the solved forces also help the designer to immediately grasp the efficiency of their design, which is another key factor for this kind of guidance.

On the other hand, using Graphic Statics to obtain a conceptual structural design is no guarantee for perfection on the first or second try. To help guide the designer towards efficient designs, the possibility exists to combine the feedback and the defined parameters with an evolutionary optimization tool like Octopus or Galapagos. Evolutionary optimization uses a genetic algorithm, as discussed in Section 1.3.3, and can help in the exploration of new directions with the capability to not only search for one (local) optimum but to randomly explore different directions to obtain a wider range of solutions, and to effectively map the design field. It is very well suited for the messy nature of building design problems and forms a powerful match with the fast solving process and flexibile parameter definition possible with Graphic Statics. However it is worth noting that heuristic optimization is not an exact method but contains imperfections and does not always find the same solution for the same problem.

Design freedom The guidance capacity of a tool is important, but the designer should in the end always be able to choose his own directions, and make his own choices. Within this tool the designer is logically limited to the principles of Graphic Statics, and the designer can ask for guidance using optimization. However following the absolute optimum is not at all required; design directions are free to be explored according to any personal preference. When an optimization set-up attempts to guide the designer in a certain direction, he or she can still be stubborn and choose otherwise. This is an important feature of the tool and should be considered with care during development.

Structural overview Here structural overview is mainly to be provided by proper and understandable visualizations. Also important is the possibility of customization of the visuals by the user, since everybody perceives visual information in a different way, and since different cases might ask for different visualizations. Also this will guarantee that the visual information can be used in different ways and will remain usable in later design stages. Furthermore, the overview must be available at all stages of the design process, and the meaning of the different visual aspects must be clear, so that the complete design information can be referenced at all times.

3.2.3 Statical determinacy

A practical requirement for the application of Graphic Statics is that the concerning truss structure is statically determinate. To find the degree of statical determinacy Equation 3.1 can be used [Hartsuijker, 2012], where n is equal to the difference of the amount of unknowns and the amount of available equilibrium conditions. if $n < 0$ the structure is kinematically indeterminate, which means that it is unstable, while if $n > 0$ the structure is classified as statically indeterminate, meaning that there is an infinite amount of possible solutions, so the unknowns in the structure cannot be determined from the equilibrium conditions alone. In the case that $n = 0$ the structure is statically determinate; the structure is kinematically determinate and the unknowns can be solved using the equilibrium conditions, meaning that Graphic Statics can be applied to the design. In the presented equation r stands for the amount of support reactions, c stands for amount of connection forces, and e depicts the amount of equilibrium conditions.

$$n = r + c - e \tag{3.1}$$

For truss systems with members that only transfer axial forces, connected only with pinned joints, the formula for statical determinacy can be simplified to equation 3.2. Here the connection forces, which together with the support reactions form the unknowns, are represented by the amount of members in the system, depicted with the letter m. The equilibrium conditions are simplified as $2j$, meaning two times the total amount of joints, since every joint must have equilibrium in horizontal and vertical direction.

$$n = r + m - 2j \tag{3.2}$$

Chapter 4

Development results

This chapter discusses the software system behind the developed force flow design tool. It focuses on the system architecture, including its concepts and data flow, and the characteristics and components of the tool prototype called GSDesign.

4.1 System architecture

This section aims to conceptually describe the structure and behavior of the developed system. First specific development requirements are listed, after which the basic system components and the data flow are presented.

4.1.1 Requirements

Next to the restrictions imposed by the general research objectives the following requirements have been defined for the development process:

- Accessibility
- Real-time modelling
- Geometric flexibility
- Extensibility
- Presentation independence

Accessibility In the first place the design tool is intended to be used by structural designers, which makes it important that the chosen software environment for the tool is widely used in that discipline. Furthermore, it must be easy to use, and the underlying concepts must be apparent and comprehensive. This applies also to the architect, other members of the design team and legislative authorities, because they also must understand the structural behavior of design alternatives. The principles of Graphic Statics offer great opportunities in this respect, because it is fundamentally easy to explain and understand. It is important that the tool conveys and utilizes this simplicity.

Real-time modelling In the conceptual design process rapid exploration of design variants is very important. A key factor is the computational speed of the analysis. For this tool the aim is to achieve real-time modelling, meaning that a change in the design will update the geometry and analysis results in real-time. An additional requirement is that a new design can be developed during a brainstorm session, meaning that the setup of such a design can also be done in a relatively short period of time.

Geometric flexibility Unlike current 2D graphic tools that only support a limited number of specific truss designs, it is important that this design tool supports any valid geometry. This requires a very general setup of the Graphic Statics rules, almost coding it as how a structured engineer would approach the problem. Additionally, the user must be able to alter the geometry rapidly, not requiring a whole new setup for one design change. This requirement mostly concerns the inputs of the tool, especially the data type and data structure of the inputs must be thoroughly thought out.

Extensibility The developed tool is only a prototype and could incorporate many more functions in the future like an upgraded optimization sequence or a version for 3D structures. The extensibility of the tool would be ensured by developing in such a manner to allow for these future extension without requiring fundamental changes to the code or software platform.

Presentation independence To support other implementations and reuse of the code, a division must be upheld between the computational logic of the analysis and the presentation of these results.

4.1.2 Concept

This section discusses the main concepts of the software system of the design tool. The main system components are introduced, as well as the data flow in the system.

System components

The conceptual system can be divided in three components, these are:

1. Parametric modelling
2. Custom components
3. Display

Parametric modelling It is key for the concept of the tool that a parametric setup of the force flow design is supported. Grasshopper, a plugin of Rhinoceros as previously discussed in Section 1.3.1, provides an environment that supports parametric modelling. Additionally, this platform satisfies the requirement of accessibility giving its widespread and increasing use by building engineers and architects. For these reasons Grasshopper is chosen as the software on which the force flow design tool will be developed.

The basic setup of a design will take place in Grasshopper, where the inputs are defined and parameterized as desired. This makes the geometric flexibility of a design partly the responsibility of the user, instead of a pure characteristic of the tool.

Custom components The core functionality of the tool will be embodied by a number of custom components, stored in a Grasshopper plugin file. These components are to be coded in C# making use of the .NET framework and the RhinoCommon API [McNeel, 2020b]. Such a configuration with custom components is preferred above using multiple general C# Script components because it allows for more control over the code, a better development workflow, a clearer end product and better shareability of the tool. Additionally, the C# language is preferred above Python, which could be used in the GhPython Script component, because of its superior performance. C# is a compiled language, while Python is an interpreted one, which is one of the main reasons that applications can run up to 44 times faster in C#. Also, the logic of the envisioned Graphic Statics functionality is considered too complicated to be modelled completely in Grasshopper; it might be possible, but would be very inefficient.

The custom components will use the predefined inputs to generate the geometry of the design, after which the forces in the design will be computed using Graphic Statics. Outputs of the components include - but are not limited to - a form diagram of the overall design, the individual force diagrams per joint, an overall force diagram and specific data concerning the visualizations.

Display The solved geometry that flows out of the components is visualized in the 3D viewer of Rhinoceros, usingonly the two dimensions of the XY-plane. Since Grasshopper contains certain custom display components which perfectly match with the display objectives, the choice is made to use those instead of coding these sequences as extra C# components. Certain visualization clusters are to be developed that contain the display logic, which ensures the accessibility and at the same time satisfies the presentation independence requirement.

Data flow

The data flow of the system has been mapped in Figure 4.1, which can be qualified as a Level 0 data flow diagram. For the creation of the data flow diagram the Yourdon and Coad notation system has been used. This means that rectangles represent external entities, circles represent processes and arrows represent data flows. A fourth and final component of the data flow diagram, that is absent in this one, is the data store, which would be displayed by a rectangular shape with one open edge [Coad and Yourdon, 1990].

The system initializes with the user defining the design inputs. These inputs together essentially form a truss design case, where nodes, members, supports and external forces are defined. Not only the actual data that is passed through, but also how this data is produced is of importance. A proper setup, parameterized specifically for the design case, is essential during designing and optimization. The definition of the design input will be done in Grasshopper, which gives access to many possibilities, including the option to reference geometry drawn manually in the Rhinoceros interface.

The Graphic Statics-based solver components use the input to create the design geometry and to solve the forces in the structure, using only the rules of Graphic Statics. The analysis results can be directly interpreted by the user, but are also send into a visualization process, which makes the results comprehensive and transparent for the user. Via the design inputs the user can also customize the analysis output which triggers different visualization

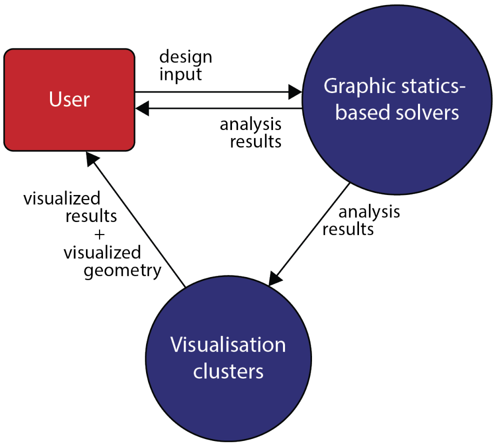


Figure 4.1: Level 0 data flow diagram of StructuralComponents 8

sequences, allowing users to customize the visualization to their own preferences.

4.2 GSDesign

GSDesign stands for Graphic Statics Design and is the working name given to the developed force flow design tool. The content of this section builds upon the described system architecture, but aims to provide a more in-depth picture by analyzing all processes in the prototype design tool, presenting a deconstruction of its workflow.

4.2.1 Code structure

The components have been developed using object-oriented programming (OOP), which is a programming paradigm based on the concepts of classes and objects. Important reasons for the use of this paradigm are that it allows for a more organized and reusable code, and that it follows the Don't Repeat Yourself (DRY) principle, resulting in efficient coding [Kappert, 2020].

Figure 4.2 shows the Unified Modeling Language (UML) diagram of the developed tool, visualizing the classes in the system and their interrelationships. As visible in the diagram no use is made of the inheritance connection, meaning that all classes are in principle stand-alone and none are subclasses of another class. Relationships that have been applied are aggregation (open diamond) and composition (filled diamond). The distinction between those two is that the latter indicates a relationship where an object of a certain class can only exist as part of an object of another class, while the aggregation relation is mostly similar but also includes the possibility of the first object existing independently. The numbers shown at the start and endpoints of the relationship lines depict the multiplicity, which sets numerical constraints on the described relationships. Additionally, it is important to note that all attributes and methods of every class have the '+' prefix, indicating that their visibility is public, meaning that all methods and attributes can be accessed freely by any other class or subclass.

4.2.2 Custom components

The core functionality of the tool has been developed in the shape of three custom Grasshopper components, presented in Figure 4.3, available under the plugin that bears the name of GSDesign. Of these the Solve Member Forces and Solve Global Forces component take care of the computation of the forces, while the Compute Fitness Function component governs post processing of these results for proper comparison and ultimately optimization purposes. In the remainder of this section the composition of each individual component will be discusses in detail, including the inputs, outputs and more specifically the internal processes.

Solve Global Forces

The Solve Global Forces component essentially computes the support reactions, and visualizes the process of doing so. This makes that this component does not behave as a black box, but can actually create understanding in this process. The inputs of the component include the following, with the C# or RhinoCommon data type included between brackets:

- The joints of the design (Point3d).
- The member start and end indices where each pair results in a member created between the joints with these indices (Integer).
- A force definition, containing a force index (Integer) depicting on which joint the force applies, a force magnitude (Double), and a force rotation (Double) which sets the clockwise rotation in degrees relative to the straight downwards direction.
- A support definition, containing the support index (Integer), the horizontal and vertical constraints (Boolean) and the support rotation (Double) in degrees relative to the default position of a horizontal base.

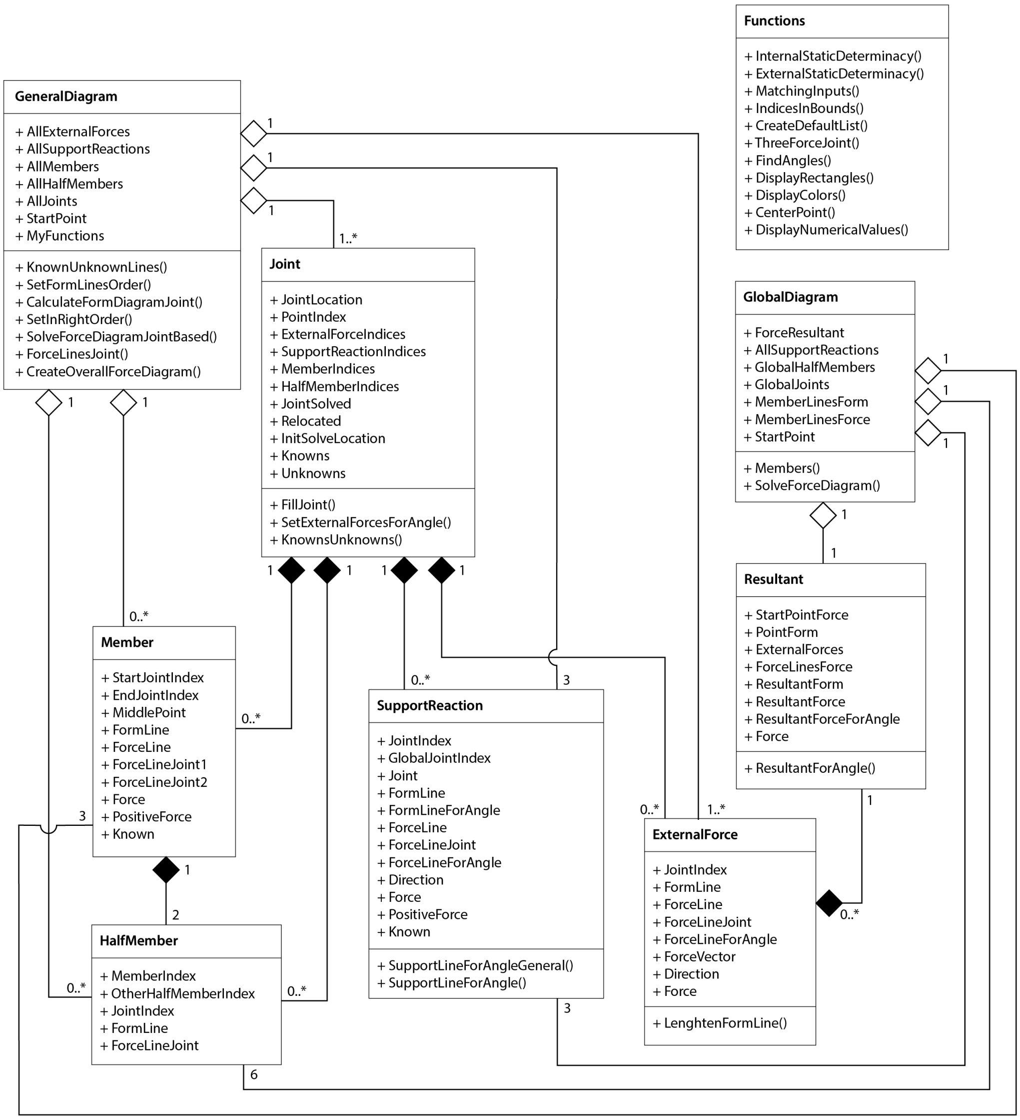


Figure 4.2: UML Diagram of GSDesign

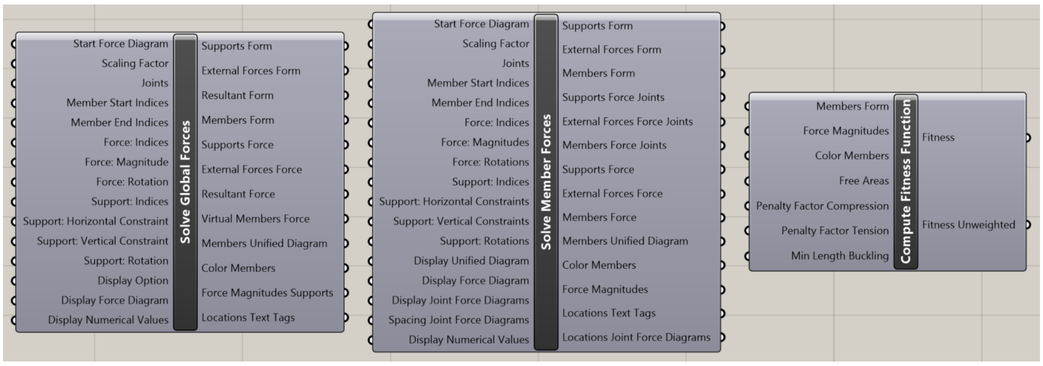


Figure 4.3: Overview of the three custom components of GSDesign

- Certain visualization options, including the start of the force diagram (Point3d), a scaling factor (Integer), a display option (Integer) which determines what kind of result visualizations the user sees, a display force diagram option (Boolean) and a display numerical values option (Boolean).

It is important to note that because all three force inputs together create the definition of the forces, the list lengths of all these inputs must be equal. The same principle holds up for the support inputs. Regarding the display option input, three options are available numbered 0 to 2, the first option the user can choose is 0 which shows just the overall structural form, the external forces and the computed support reactions and leaves the virtual global members out of the form and force diagrams. Option 1 on the other hand displays the global (virtual) members instead of the actual members in the form diagram, and also includes those members in the force diagram. Option 2 is the last option and provides essentially the same image as option 1, with the distinction that the form diagram has changed into the so-called unified diagram, meaning that the forces in the global members are visualized by the thickness of the members, forming translucent rectangles instead of solid lines.

Like all main functionality of the tool, the resultant force is computed graphically. Figure 4.4 shows the first three steps in the computation of the resultant for an example case with three arbitrary external forces. The final diagram of this figure contains the information that allows for the next two steps of the process as illustrated in Figure 4.5, where the action line of the resultant force is computed and placed in the form diagram.

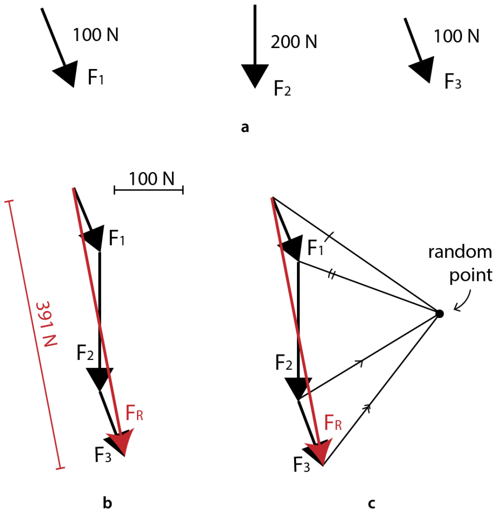


Figure 4.4: The first three steps of the resultant definition: **a** Three external forces of which the resultant must be found, **b** Addition of the three vectors in the force diagram, finding the direction and magnitude of the resultant force, **c** An arbitrary force diagram, created by connecting any random point to the start- and endpoints of the forces

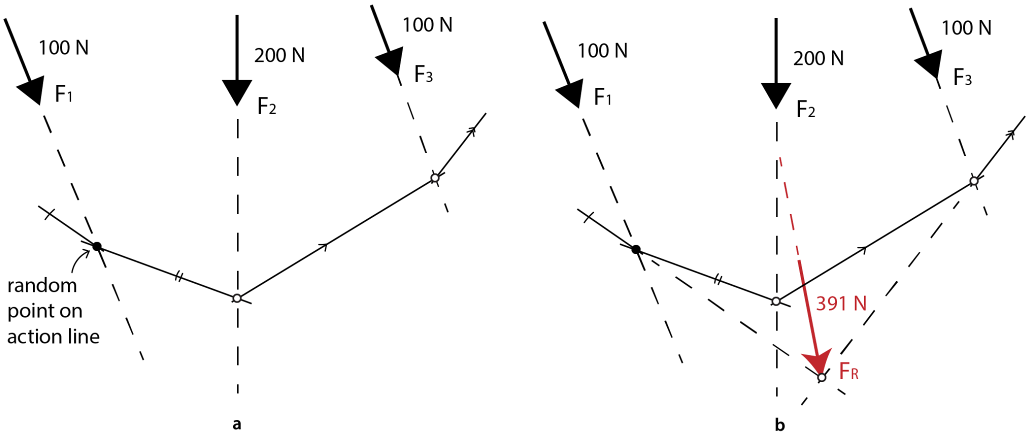


Figure 4.5: The final two steps of the resultant definition: **a** Using a random point on the action line of F_1, action lines are connected by drawing lines parallel to those in the global force diagram, **b** The action line of the F_R is found by finding the intersection point of the far most left and right lines of the diagram

With the resultant force known, the support reactions can be solved by constructing a global force diagram. It is important to note that this diagram is an instrument to compute the support reactions, and has a somewhat arbitrary shape that does not represent any physical structure. To reduce the workload of the development the choice has been made to limit the tool to only accept structures with two supports, one pinned and one roller, which makes that the global form diagram always has the shape of a triangle of which two points are given by the supports and the other one being an arbitrary point on the action line of the resultant force. Since this shape is fixed, the solving process of the three unknown support reactions is rather simple, and follows this order:

1. Solve the top joint. The resultant force is the only known force, which can by constructing a closed force polygon solve the forces in the two connected global members.

2. Solve the support with only one support reaction. Using the solved member force from the previous step the final member force and the first support reaction can be computed

3. Solve the remaining support. Since all global member forces are known by now, the support reactions in the pinned joint can be computed.

Illustrations of a global form diagram and its corresponding solved joint force diagrams are presented in Figure 4.6. The choice of any other point on the action line of the resultant force would result in a different global form and force diagram, but the proportions would be exactly the same, leading always to the same support reactions. The only really *wrong* choice for this arbitrary point would be on the line between the supports, because in that case equilibrium cannot be achieved. This occurrence case is prevented by picking a random enough point, resulting in an infinitely small chance that this point exactly coincides with the line between the supports.

An extension of the tool to also allow structures with three roller supports could easily be implemented due the extensibility that the object oriented setup of the code ensures. It is only not implemented in this prototype due to time limitations and a different prioritization.

The output of the component is made up of the lines of the global form and force diagram, divided in supports, external forces, resultant and (global) members with the unified diagram members outputted as rectangles. Additional outputs contain the color of the global members (Color), the force magnitudes of the supports (Double) and the locations of the corresponding

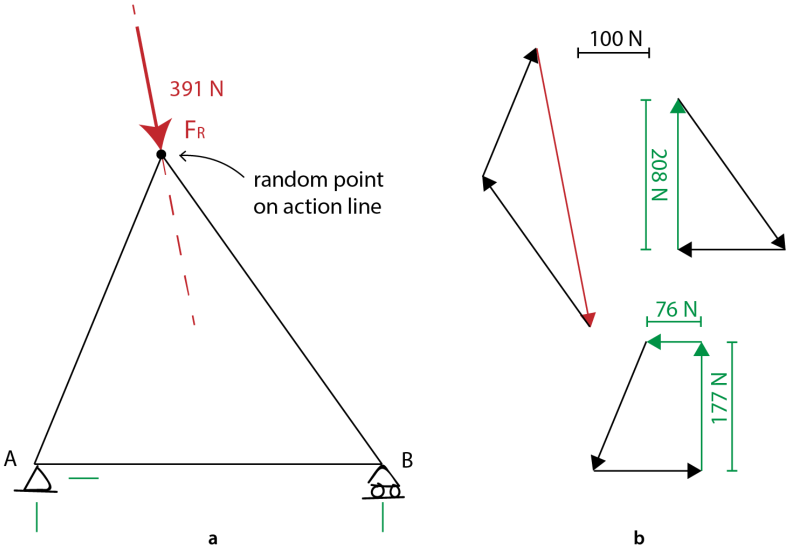


Figure 4.6: The solving process of the support reactions: **a** An arbitrary global form diagram, created by connecting a random point on the action line of F_R and the two supports to each other, **b** The solved force diagrams of the solved, starting with the top joint (top left), then support B (top right), and finally support A (bottom), resulting in the definition of all three support reactions

text tags in the form diagram for these support magnitudes. More information on how the visualizations are implemented is available in Section 4.2.3.

Solve Member Forces

The Solve Member Forces component can be seen as the main component of the tool. Essentially it contains the functionality of the Solve Global Forces component in a condensed form plus the general Graphic Statics logic that solves the forces in all members of the design. This means that the Solve Global Forces and Solve Members Forces components can be used independently of each other, with the first focusing on the computation of the support reactions, while the latter regards the support reactions as a given and focuses on the member forces. The inputs of the component are similar to those of the Solve Global Forces component except for some differences in the display options part, including the addition of a number input called Spacing Joint Force Diagrams (Double) which governs the spacing between the joint force diagrams. This is because the Solve Member Forces introduces the visualization of individual force diagrams per joint next to the overall force diagram.

The solving process of all the members in the design is somewhat similar to that of the global form diagram, but requires a more general approach to accommodate all different possible designs. To explain this process the example case of Figure 4.7 is used, which continues on the earlier illustrations. The situation in this image is considered as the starting point where the external forces and the support reactions are known, and all member forces yet unknown. Figure 4.8 contains the setup of two joints of the structure at different stages of the overall process, whose solving process will be analyzed in depth. These joints have been chosen because they have a different amount of knowns and unknowns which require a slightly different solving process, making sure that all computational logic is explained.

During the solve process every joint is analyzed in ascending numerical order and is solved if the following conditions are met:

- Joint is yet unsolved
- Joint contains less than three unknowns

This means that, provided that conditions are met, the joint has either two, one or zero unknowns. In the case of zero unknowns, all corresponding

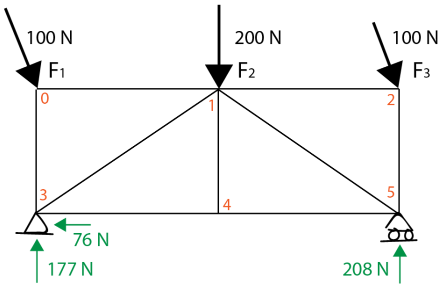


Figure 4.7: A simple Howe truss structure with the external forces and support reactions as determined in figures 4.4 and 4.6, with joints numbered 0 to 5

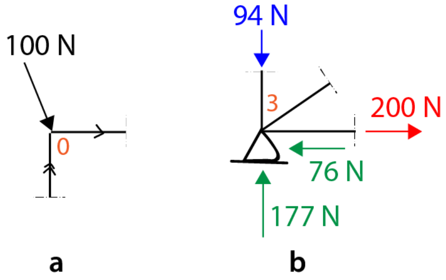


Figure 4.8: Close up of two joints of the truss from Figure 4.7: **a** Joint 0, containing one known external force and two unknown member forces, **b** Joint 3, containing two known support reactions, two known member forces and one yet unknown member force

member forces have already been solved by the adjacent joints, meaning that no additional solving actions are required. In the case of one or two unknowns, first the beginning of the joint force diagram is constructed by adding all known force vectors graphically, as can be seen in part a of figures 4.9 and 4.10. If the joint only has one unknown, a vector can be constructed from the endpoint of the sum of all known vectors to the startpoint of the sum of all known vectors. This process can be observed in part b of Figure 4.10 and creates a closed force polygon; the only requirement for joint equilibrium. The member force is proportional to the vector length, while the direction depicts whether it is in compression or tension. The slope of the vector should always be similar to that of the corresponding member in the force diagram.

In the case of a joint with two unknowns, some additional steps need to be taken to find all member forces. The first unknown line is drawn in the joint force diagram, with its starting point being the endpoint of the sum of all known force vectors, while the second unknown line is drawn starting at the startpoint of these vectors. An intersection point can be found between these infinite lines, which must be used to draw the final version of these previously unknown member forces, also closing the force polygon. This process can be seen in parts b and c of Figure 4.9.

These processes can be executed for all joints, making that at this stage all forces in the design are known. However, when one would combine all the constructed joint force diagrams into one, the results would not necessarily match, creating a very disorderly image. This is because the order of the force vectors in each diagram is inconsistent, and is split up into known and previously unknown forces. Therefore the force vectors in every joint force diagram must be reshuffled according to a set order in the form diagram, which in this case is chosen as clockwise. Additionally, all external forces and support reactions should be added last in the force diagram, making sure that it is not placed internally between two member forces, but always on the outside of the structure. A reshuffled joint force diagram containing the correct order is illustrated in part d of Figure 4.10.

For joints that contain external forces or support reactions an extra trick is necessary to make sure that the order of the vectors in the force diagram is correct. For a consistent and correct overall force diagram it is important that all external forces and support reactions on a joint are ordered in the joint force diagram as if they were applied from the external side of the node. Figure 4.11 shows this principle for the example of joint 0 of Figure 4.7 with F_1 rotated 180 degrees. By determining the placement of the external forces and support reactions in this way, the order is consistent in all joint force

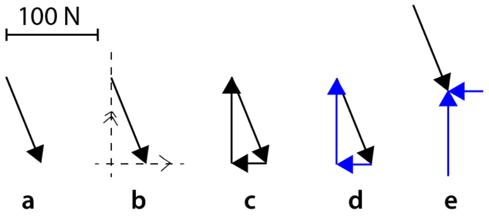


Figure 4.9: The solving process of the joint force diagram of joint 0, defined in Figure 4.8: **a** All known force vectors added up in clockwise order; in this case just one vector, **b** The first unknown force in clockwise order is drawn from the endpoint of the known vector, while the other unknown force is drawn from the startpoint of the same vector, **c** Using the intersection point of the two unknown force lines a closed force polygon can be created, **d** Both vectors are directed towards the joint meaning that they depict compression forces, which is indicated by the blue color. The vectors are already in clockwise order and do no not need to be reshuffled, **e** A visualization of how the forces act on the joint, referencing to the form diagram

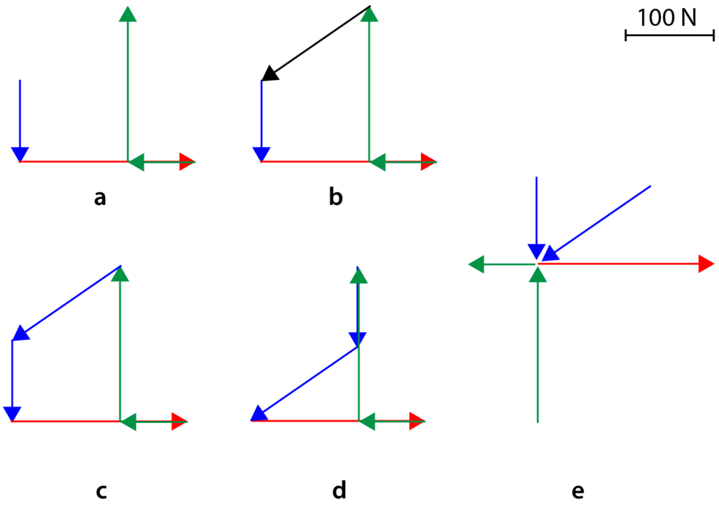


Figure 4.10: The solving process of the joint force diagram of joint 3, defined in Figure 4.8: **a** All known force vectors added up in clockwise order; in this case two support reactions (green), one compression force (blue) and one tension force (red), **b** Since there is only one unknown force, a vector can be drawn simply from the endpoint of the last known vector towards the startpoint of the first known vector, **c** The solved vector is directed towards the joint meaning a compression force visualized by a blue line, **d** The reshuffled diagram upholding the clockwise order, **e** A visualization of how the forces act on the joint, referencing to the form diagram

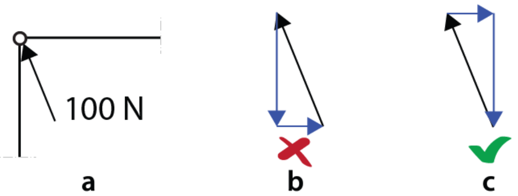


Figure 4.11: Illustration of the correct order of forces in the force diagram: **a** Joint 0 of Figure 4.7 with F_1 rotated 180 degrees, **b** The incorrectly constructed joint force diagram maintaining a simple clockwise order, **c** The correct joint force diagram maintaining still clockwise order, but regarding the external force as applied from the external side of the joint

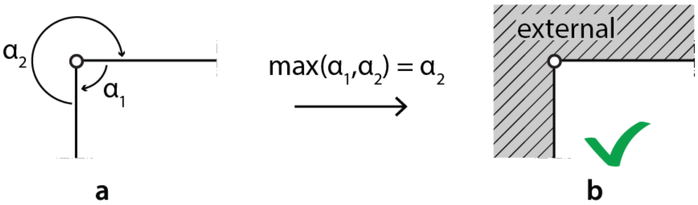


Figure 4.12: Definition of the external side of the joint, based on the angles between adjacent member: **a** Definition of the angles, **b** The correct definition of external space at the location of α_2

diagrams and matches perfectly together in the overall force diagram.

The only difficulty in this process is the definition of external and internal space of the structure. In the prototype this is implemented by measuring the angle between all adjacent member of a joint, where the location of the largest angle is assumed to be outside space. This is a method that is right in about 99% of the practical cases, but is however not always true. Figure 4.12 shows the method applied on the same joint as the previous example, while Figure 4.13 displays how this method can result in a faulty definition of external space. Another occurrence that is not supported by this method is the application of an external force on a fully internal joint. It is worth noting that if any of these events occur only the overall force diagram is displayed incorrectly, while the joint force diagrams, unified diagram and most importantly the analysis results itself are still correct.

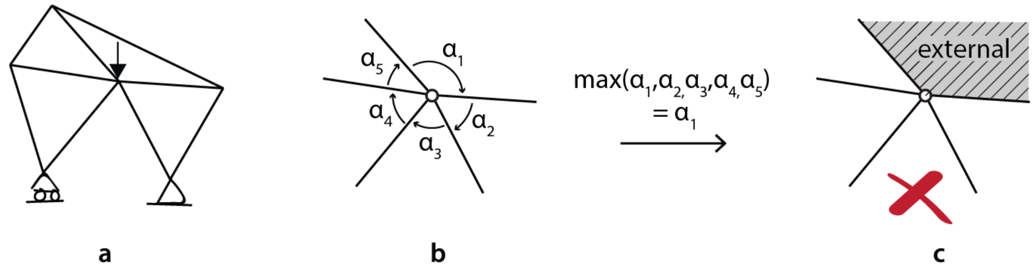


Figure 4.13: An incorrect definition of the external side of the joint for a certain design: **a** Definition of the design, **b** Definition of the joint on which the external force is applied, including all angles between adjacent members, **c** The faulty definition of external space, since the largest angle does not coincide with the outside space for this joint

The outputs of the component are mainly similar to those of the Solve Global Forces component, with the addition of the geometry and location of the joint force diagrams. The geometry of the resultant force is not included as an output.

Compute Fitness Function

The Compute Fitness Function component is a very simple component that processes the output of the Solve Member Forces component and computes a fitness function of the design. The inputs include the lines of the members in the form diagram, the force magnitudes and the color of the members, which all come directly from the Solve Member Forces component. Additional inputs include areas that must be kept free of structural elements (Curve), a penalty factor for tension and compression (Double) and a minimum length for buckling to occur in compression elements (Double). All these inputs are taken into account to quantify the performance of the design. The resulting fitness value is smaller for more efficient designs, making the minimization of this value a useful objective for optimization. More details on the theory, the underlying reasoning and the establishment of this fitness value will be discussed in Section 4.2.4.

4.2.3 Visualization

The visualization is a key factor of this design tool, since it is the main form of feedback towards the user. This section contains a description of the

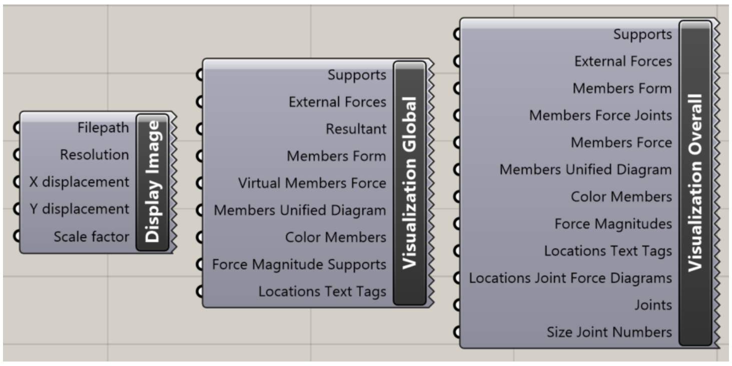


Figure 4.14: Overview of the three custom Grasshopper clusters of GSDesign

Grasshopper clusters that contains the visualization logic, which are compared against the theory provided in Section 1.3.3.

Besides the components described in Section 4.2.2 the tool also consists of several Grasshopper clusters that contain the visualization logic of the output of the components, of which an overview is provided in Figure 4.14. A cluster is simply a condensed representation of some logic defined in Grasshopper, making the overall code more structured and easier to read, and the defined logic easier to manage and share. Both solver components have their own unique visualization clusters, called Visualization Global and Visualization Overall. The inputs of these clusters include all the geometry that is exported by the components, as well as additional information regarding the visualization, like force magnitudes and locations of text tags.

Figure 4.15 contains a representation of how the results of the Solve Member Forces component are visualized as a unified form diagram and a force diagram, directly taken from the viewer of Rhinoceros. The following visualization rules can be identified:

- Compression elements are visualized in blue
- Tension elements are visualized in red
- External forces are visualized as vectors in black
- Support reactions are visualized as vectors in white
- In the unified diagram the compression and tension members are visualized as rectangles with a thickness linearly dependent on the associated member force.

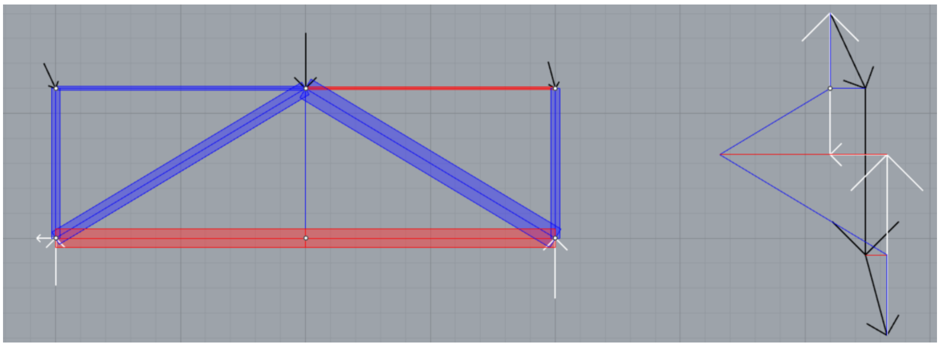


Figure 4.15: The solved unified diagram and force diagram of the structure of Figure 4.7 as visualized by GSDesign in the Rhinoceros viewer

Apart from the unified diagram and the overall force diagram, the user can also choose to display the form diagram and/or the joint force diagrams, which all follow the same coloring rules, but do not contain the rectangular elements of the unified diagram. As discussed extensively previously, the nature of Graphic Statics makes it so that the lengths of lines in the force diagrams are linearly related to the force that the members carry, which is an interesting and in theory easily quantifiable method of force visualization that could provide certain insights and a better understanding. A last means of communicating the analysis results to the user is by displaying the numerical values of the forces next to the corresponding lines and vectors in the form diagram. As opposed to the other visual methods this does not provide an immediate understanding of the structural behavior, but can be used when more precise information is required in order to objectively quantify the computed force in each member.

In Figure 4.16 a visualization of the Solve Global Forces component is displayed for the same design. This visualization follows the same rules regarding coloring as for the Solve Member Forces component, but additionally shows the resultant action line and force vector in gray. Also, instead of the actual members, here the virtual members of the global diagram are shown which is used as an intermediate step to solve the support reactions. For the Solve Global Forces visualization the user can choose between three display options, of which the first shows the unsolved actual members, the second shows the global form diagram, and the third shows the global unified diagram.

A third cluster bearing the name of Display Image allows the user to import a jpg, png or bmp image to the canvas, using the Grasshopper native Import Image component, which provides a mesh representation of the image. This cluster can be used to import a manually made sketch for the design

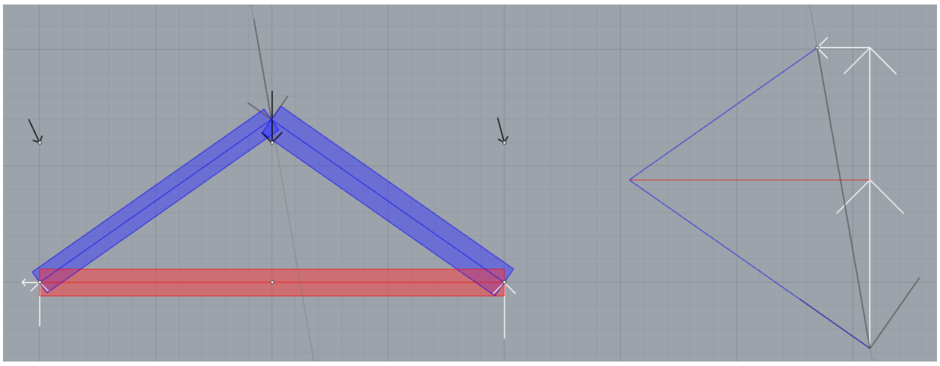


Figure 4.16: The solved global unified diagram and global force diagram of the structure of Figure 4.7 as visualized by GSDesign in the Rhinoceros viewer, where the gray vector represents the resultant force

problem at hand and place it on the right location at the desired scale. This cluster therefore effectively connects the actual design problem with a GSDesign model, and therefore actively supports the workflow depicted in Section 3.2.1.

4.2.4 Optimization

Based on the methods provided by Beghini et al. [2014] an optimization setup is built into the tool using the Compute Fitness Function component described in Section 4.2.2 combined with an external optimization tool like the Grasshopper native plugin Galapagos. To accommodate the objective of designing more efficient and sustainable structures a fitness function is formulated that aims to minimize the total volume of structural material needed. The formula in Equation 4.1 gives the relation between the minimal volume and the forces and lengths of the elements in the structure, stating that a minimum structural volume is obtained by minimizing the sum of the force in each member multiplied by the force it carries. This is a simplification of the minimum total volume that only holds if a constant state of stress is assumed, which is a reasonable approximation in this case [Stromberg et al., 2012]. The value that comes out of this summation is known as the total load path.

$$\min_{x} V = \min_{x} \frac{1}{\sigma} \sum |P_i| * L_i \tag{4.1}$$

In this equation the V stands for the total volume, σ depicts the stress and P_i and L_i respectively stand for the internal force and length of the

i^{th} member. In terms of Graphic Statics the total volume can therefore be minimized by multiplying the length of a line in the force diagram by the length of the corresponding member in the form diagram.

In order to provide the user with some handles to exercise more control on the optimization results, the option to favor either compression or tension-based structures is incorporated in the Compute Fitness Function component. A penalty factor can be imposed on either (or both) compression or tension elements, which results in a multiplication of all relevant $|P_i| * L_i$ computations with this factor. If the penalty factors are set to zero or one, the result will be similar to a result without a penalty factor. Additionally, the user can define certain curves wherein no structural elements are allowed. The Compute Fitness Function component will check if this requirement is met, and will multiply the fitness value with the arbitrary relatively large number of $1e^6$ to make sure that that design will be identified as inefficient by an optimization tool. Since the fitness value is manipulated by these penalty factors an extra output called Fitness Unweighted is added to the Compute Fitness Function component which contains purely the total load path $\sum |P_i| * L_i$.

The described penalty factors can be used to for example accommodate a favorability for a certain material or to achieve a different nature of the optimized structure. However, when dealing with compression it must be clear they do not account for buckling, which can be an important factor in the design of normal force-based structures. The formula for Euler buckling as given in Equation 4.2 is the simplest way to say something substantive about the buckling performance of a member and quantify the critical load P_{cr}. The equation does however include the Young's modulus (E) and the second moment of inertia (I), which are material and cross-sectional characteristics which are not yet defined during the conceptual design stage, meaning that the Euler buckling equation cannot directly be incorporated in this tool.

$$P_{cr} = \frac{\pi^2 EI}{L^2} \tag{4.2}$$

In the equation L stands for the buckling length, which is similar to the member length for pin-jointed elements. In order to incorporate buckling effects, without defining any kind of cross-sectional properties and material, only the inverse quadratic relationship of L with the critical load is incorporated. To quantify the effect that the member length has on the total volume of structural material required, a $L_{min,buc}$ must be defined as the minimum member length from where buckling starts to occur. If $L_i \geq L_{min,buc}$ is true for member i, the fitness value of that member is multiplied with factor

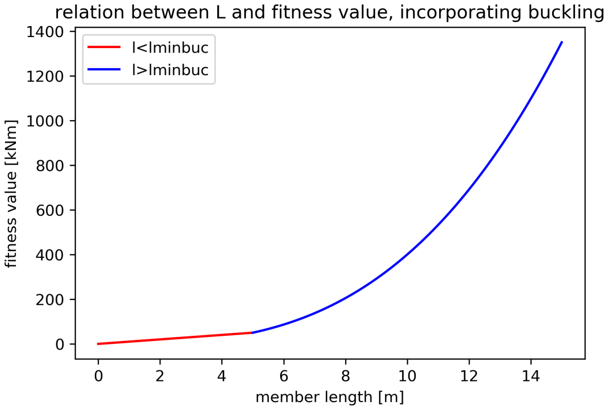


Figure 4.17: The relation between member length L_i and the fitness value, which is defined as $|P_i| * L_i$ for $0 \leq L_i < L_{min,buc}$ and $(|P_i| * L_i) * (\text{Ł}_i^2 / L_{min,buc}^2)$ for $L_i \geq L_{min,buc}$, with $L_{min,buc}$ set to $5m$

in Equation 4.3. If $L_{min,buc}$ is defined as 5m and a member load of 10kN has been calculated, the progression of the fitness value can be observed in Figure 4.17.

This described mechanism is included in the Compute Fitness Function component, where the value of $L_{min,buc}$ can be chosen by the user as one of the component inputs. A negative value in this input, as is the default, will disregard any buckling effects.

$$\frac{\text{Ł}_i^2}{L_{min,buc}^2} \tag{4.3}$$

Chapter 5

Validation

This chapter describes the validation of the design tool described in Chapter 4. For starters, the tool is tested against certain benchmarks cases, after which a more extensive case study is executed to showcase the tool in a practical use case and validate its workflow. Finally, a user experiment is included in order to say something substantive about the benefit that the tool could provide in a practical environment. An evaluation of the results of the whole chapter is included as the last section.

5.1 Benchmark cases

Benchmark cases are design cases where the outcome is either known because of previous research or can easily be verified by performing simple calculations and applying basic physics. Two benchmark cases have been included, namely a single panel truss and a topology optimized cantilever structure, which will be presented in this order.

5.1.1 Single panel truss

Setup The first benchmark case that will be used to validate the developed tool is the so-called single panel truss. It is a simple structure that consists of one vertically applied external force, two supports, and a total of three members and three joints, forming a triangle. The basic setup and geometry is displayed in Figure 5.1, which shows that the structure is symmetrical with the top joint, where the one external force is applied exactly in the horizontal middle. Additionally, Figure 5.2 shows a complete definition of the design problem, including the form and force diagram. Due to the symmetrical nature of the truss and the parallel relations of the lines in the form and

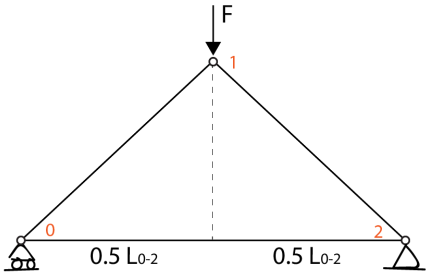


Figure 5.1: The basic geometry and setup of the single panel truss benchmark case

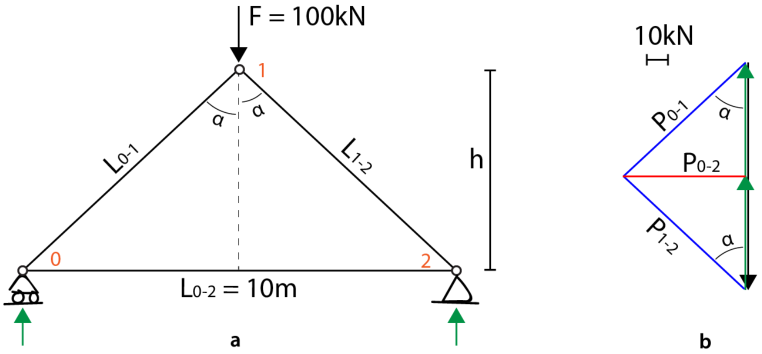


Figure 5.2: Full definition of the single panel truss benchmark case: **a** The form diagram, including a quantification of the F and L_{0-2} variables, **b** The force diagram

force diagram the angle α occurs four times as is displayed in the figure. Since the external force F and the length of the bottom member L_{0-2} are defined in this case as respectively $100kN$ and $10m$, the lengths of all lines in both diagrams only depend on the height of the truss h.

The next step is to create this design within the GSDesign tool, and both verify the basic outlook of the force diagram and perform a basic optimization process. Before the results of the GSDesign tool will be analyzed, first the expected optimization result will be calculated.

Optimization To find the desired optimization result the previously presented Equation 4.1 must be used. In the case of this single panel truss the definition of $\sum |P_i| * L_i$ is given in Equation 5.1, which provides an expression that is only dependent on h once all definitions given in Equation 5.2 are included. A plot of the total load path value of $\sum |P_i| * L_i$ against the truss height h is given in Figure 5.3, where the optimal minimum result is

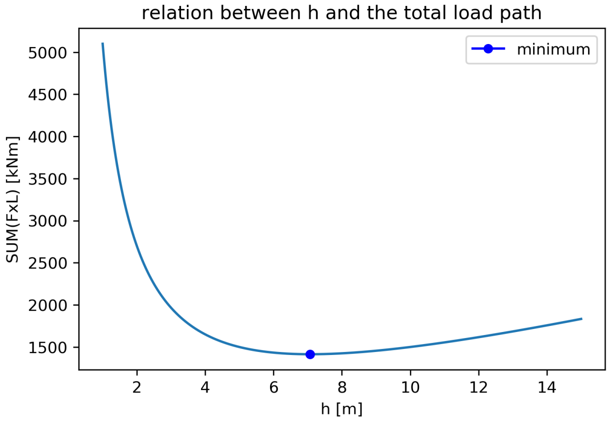


Figure 5.3: The total load path plotted against the truss height h, with $F = 100kN$ and $L_{0-2} = 10m$

marked with a blue dot. The minimum value seems to appear at $h \approx 7$, and further analysis of this minimum sets the optimal value of h at 7.07107... or $5\sqrt{2} = 0.5L_{0-2}\sqrt{2}$.

$$\sum |P_i| * L_i = P_{0-1} * L_{0-1} + P_{1-2} * L_{1-2} + P_{0-2} * L_{0-2} \tag{5.1}$$

$$\begin{aligned} F &= 100kN \\ L_{0-2} &= 10m \\ \alpha &= \arctan \frac{L_{0-2}}{2h} \\ L_{0-1} = L_{1-2} &= \frac{h}{\cos \alpha} \\ P_{0-1} = P_{1-2} &= \frac{F}{2\cos \alpha} \\ P_{0-2} &= \frac{Ftan\alpha}{2} \end{aligned} \tag{5.2}$$

Results The design of the single panel truss can easily be created in the GSDesign tool using only the Solve Member Forces component and the Visualization Overall cluster, which creates an image like the one in Figure 5.4. It is clear that the force diagram has the same appearance as the one from Figure 5.2, and the translation of these forces to the unified diagram also checks out, maintaining the right color scheme and displaying the relations of the forces properly by varying thicknesses of the lines.

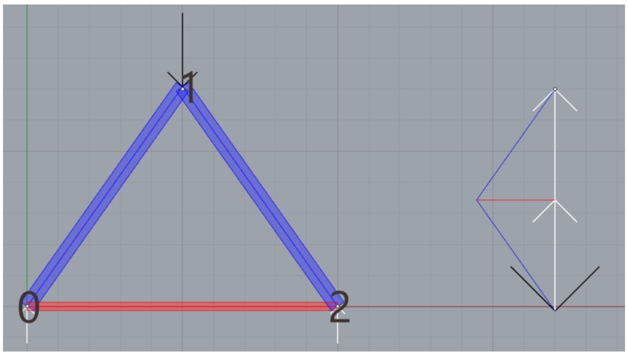


Figure 5.4: Visualization of the optimized design produced by the GSDesign tool applying the proposed workflow, taken directly from the Rhinoceros viewer

To set up the optimization the Compute Fitness Function component is included and connected with a Galapagos component that will perform the genetic optimization. The only genome for the optimization is the y-value of the top joint, which represents the h parameter introduced earlier. Figure 5.4 does not just display an arbitrary single panel truss design, but actually displays the optimized structure. Due to the 1.0 by 1.0 raster in the background it is clearly visible that the height of truss is around $7m$, and an parameterization of the height value with five decimal points gives a value of $y = 7.07107$ for the top joint, which perfectly corresponds with the expected value.

Figure 5.5 shows the optimized design when the compression penalty input of the Compute Fitness Function component is set to a value just slightly larger than 1.0, essentially flipping the optimal structure so that most of the structure is in tension. The exact value of the compression penalty is set to $1 + 1E^{-9}$, which is big enough to favor a tension-based structure, but small enough to not really influence the shape of the optimal design, which is proven by the found optimized value of $y = -7.07107$ for joint 1. Compression or tension penalties that are significantly smaller or larger than 1.0 will result in different shapes, deviating from the $0.5L_{0-2}\sqrt{2}$ value since compression and tension elements are no longer valued equally. Overall, it can be concluded that the GSDesign tool exhibits the desired behavior and the general functionality and optimization process have been validated for this benchmark case.

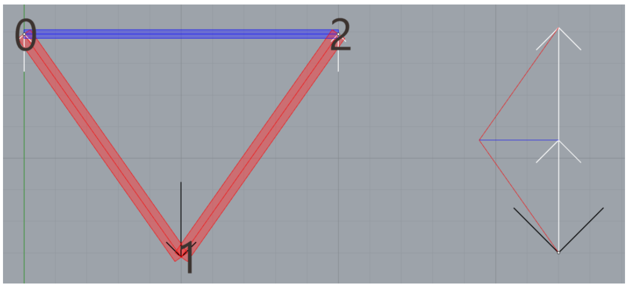


Figure 5.5: Visualization of the optimized design with a favorability for tension elements, produced by the GSDesign tool applying the proposed workflow, taken directly from the Rhinoceros viewer

5.1.2 Cantilever

Setup Unlike the former benchmark case, this case is based on previous academic research on the subject of optimization using Graphic Statics. Although the applied optimization methods do differ, an example case that is included in Beghini et al. [2014] can be used to validate the functionality and optimization process of the tool for more complex cases with numerous degrees of freedom. The case study by Beghini is on itself also based on an even older example introduced in Figure 15 in Rozvany et al. [1995], which concerns a topology optimization for a cantilever problem. Figure 5.6 shows the main result of the topology optimization, which is interpreted by Beghini into a certain connectivity of axial members. The initial form diagram based on this interpretation is shown in image a of Figure 5.10. The objective of Beghini was to optimize this initial design and show that the topology optimization in combination with Graphic Statics-based optimization is an efficient and effective design workflow. The objective of this thesis in relation to this case is to achieve similar optimization results to validate the optimization process of the developed tool.

To translate the design by Beghini to a design suitable to the GSDesign framework there is one important obstacle that needs to be overcome. The design by Beghini contains two pinned supports and is therefore externally statically indeterminate; a characteristic that is not supported in GSDesign. In order to solve this issue an extra member is added to the structure between the two supports, which ensures the internal stability in a design where also one of the pinned supports is changed into a roller. Additionally, the base of the roller support is rotated in such a way that the newly added member carries no force at all. This setup is shown in Figure 5.7 and essentially makes the structure and the support reactions the same as in Beghini's statically indeterminate design, ensuring also similar behavior. For clarity purposes

Figure 5.6: The topology optimization results of the example in Figure 15 in Rozvany et al. [1995], as generated by Beghini et al. [2014]

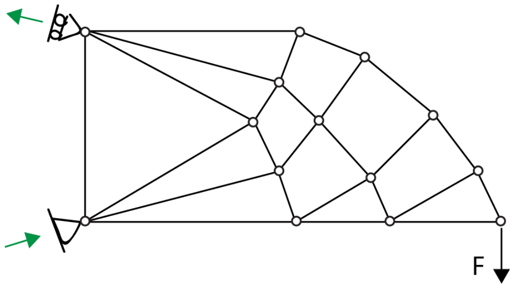


Figure 5.7: Statically determinate connectivity interpretation functioning as the initial form diagram, based on the interpretation by Beghini et al. [2014] which is shown in image a of Figure 5.10

the pinned support is also rotated so that GSDesign will show the support reaction as one vector, instead of it being split up into a perpendicular and parallel part.

Figure 5.8 shows the initial design modelled in GSDesign, clearly displaying that the member between the two supports bears no visible force. Slight discrepancies can be noted between the force diagram in this image and the one by Beghini in image b of Figure 5.10. A deeper analysis of the image by Beghini reveals that the force diagram does not exactly match the corresponding form diagram, which is most clearly illustrated when one compares the clearly mismatching slope of member 6-7 in the form and force diagrams; a trait that is not allowed in Graphic Statics. Further discrepancies could form due to slight variations when constructing the two designs, leaving absolutely no indication that the GSDesign tool is performing faulty here. The appearance that this image by Beghini in not entirely correct does not necessarily damage this benchmark analysis, since it is in fact only the optimized form and force diagram that are needed for validation. No matter how ex-

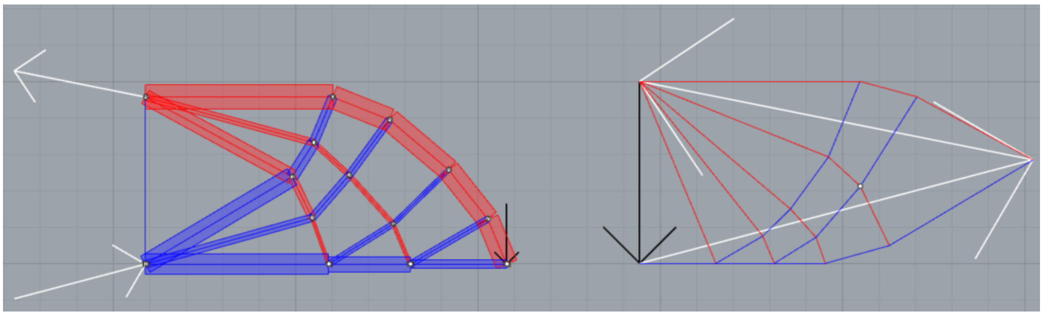


Figure 5.8: The form and force diagram of the initial design, created in the GSDesign tool

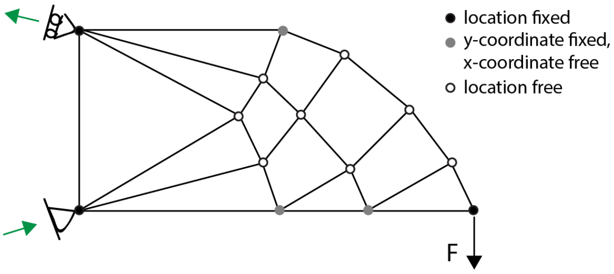


Figure 5.9: The setup of the optimization, depicting how the location of each joint is parameterized

actly the initial form or force diagram is defined, as long as the supports and external force locations and the general connectivity are consistent, an optimization should always converge to the same result.

Optimization The optimization results by Beghini are displayed in Figure 5.10. Out of these images and the explanation given by Beghini et al. [2014] it can be derived that an optimization setup like the one presented in Figure 5.9 was used. Beghini reports that the initial design is optimized using the same basic fitness function as is used in GSDesign, namely that of the total load path. The results of the optimization using the GSDesign tool should therefore be similar to those in images c and d of Figure 5.10.

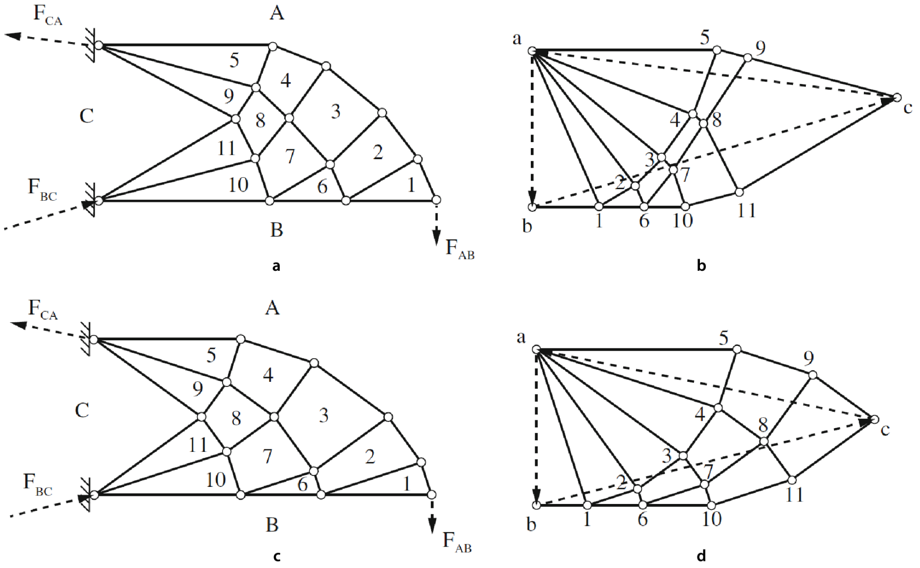


Figure 5.10: The optimization results by Beghini et al. [2014]: **a** The initial form diagram, **b** The corresponding initial force diagram, **c** The optimized form diagram, **d** The corresponding optimized force diagram. All illustrations courtesy of Beghini et al. [2014]

Results An unified diagram and form diagram of the in GSDesign optimized structure are displayed in Figure 5.11. The changes of the locations of the joints can be observed by comparing joints in the design against the original joints depicted as white dots, indicating that the design indeed has changed considerably.

In order to properly analyze the results Figure 5.12 provides an overlay of the force diagrams of both Beghini and GSDesign. From this the conclusions can be drawn that the overall shape of the diagrams are similar, but that they are although very close not a perfect match. The reason for these differences remains somewhat unclear due to a lack of precise performance information from the design by Beghini, but could be accredited to a different definition of the variables, a difference in the precision of the quantification of the variables and/or the adjustments that were necessary to make the design statically determinate. Another cause could be the use of a different optimization algorithm, finding different optima. An analysis with the GSDesign tool of Beghini's optimal design returns the information that the fitness functions for both designs are virtually equal. Therefore the results are deemed similar enough to indeed consider the results acceptable, validating the optimization process of GSDesign for more complex cases like this cantilever problem.

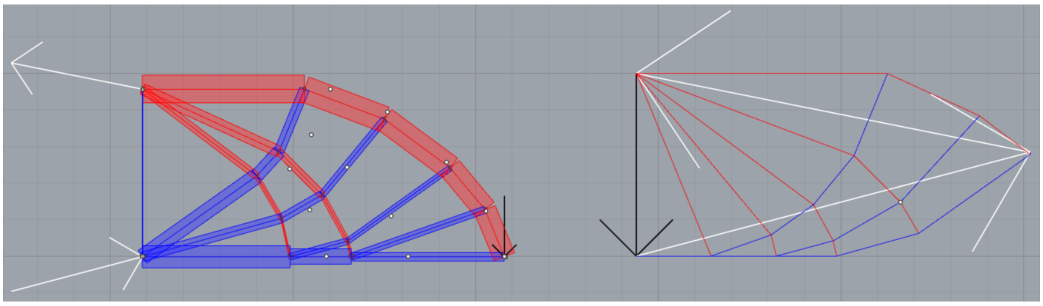


Figure 5.11: The optimization results obtained by GSDesign, containing the optimized unified diagram on the left and the corresponding force diagram on the right

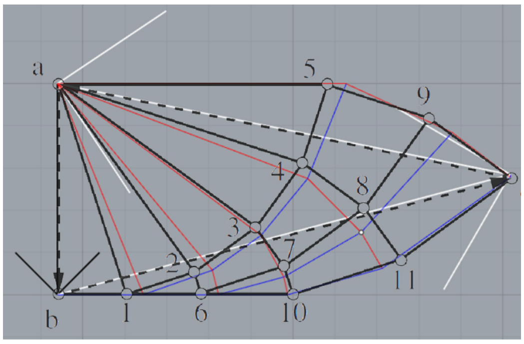


Figure 5.12: An overlay of the optimized force diagram found by Beghini (black) and the GSDesign tool (red and blue)

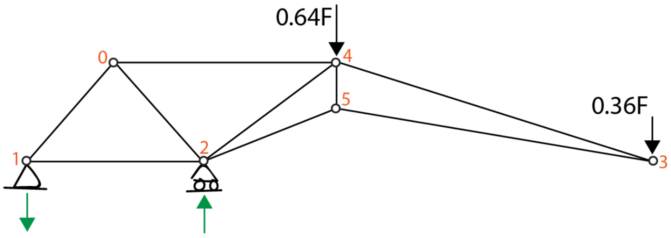


Figure 5.13: Initial form diagram of the TU Delft auditorium roof structure

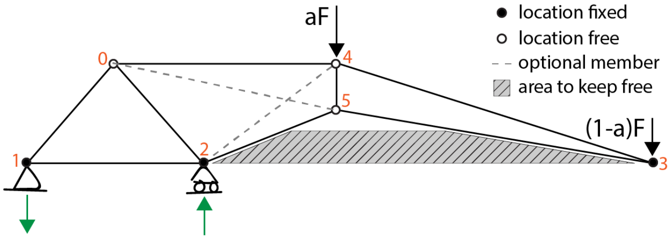


Figure 5.14: Optimization setup for the TU Delft auditorium roof structure

5.2 Case study

Now that the developed tool is validated for the benchmark cases a case study will be executed, showcasing the use of the tool in a practical scenario. The design case here will be similar to the one in Section 3.1.3, and therefore concerns the roof and floor structure of the auditorium on the campus of the Deflt University of Technology. Again the roof and floor structure are regarded as separate structures which will be treated separately.

5.2.1 Roof structure

Setup Compared to the proposed initial design setup for Graphic Statics in Figure 3.5 the initial form diagram is mostly similar. The only difference is that the distribution of the roof load on the two loaded joints is recomputed using the actual distance between the joints to compute which proportion of the total load is directed to either node. The resulting initial design setup is displayed in Figure 5.13 where the top joints gets 64% of the total force instead of the former estimation of 70%.

Optimization The optimization setup for this structure can be observed in Figure 5.14. Nodes 1 and 2 have a fixed location to ensure that the locations of the supports are constant in every design, while the location of node 3 is fixed to guarantee that the auditorium is fully covered and properly connected to the floor structure. The other three nodes have parameterized x and y-coordinates of a certain range, allowing them to move around. The most important limit in this range is that nodes 0 and 4 cannot move up more than in the initial design to impose a limit on the structural height; which could be a requirement of the architect. Also the external loads on joints 4 and 3 are automatically computed so that a shift of node 4 towards node 3 will result in a proportionally larger force on node 4, or the other way around, always making sure that the summation of the loads will remain equal.

Furthermore, the member that was between joints 2 and 4 has been parameterized into a member between either joints 2 and 4 or joints 0 and 5. This configuration shows the wide range of parameterization possibilities, extending the optimization opportunities, which is as exhibited not limited to the locations of the nodes, but can at the same time take into account the connectivity. A final input for the optimization is the marked area between nodes 2, 3 and 5 that needs to be kept free to ensure the internal height requirements of the auditorium.

Results The form diagram and the described optimization setup have been created in the GSDesign tool, resulting in the visualizations depicted in Figure 5.15. This image includes the force diagram and the unified diagram, including also the trapezoidal curve which represents the area that needs to remain free of structural elements. An optimization using the Compute Fitness Function component in combination with Galapagos and a parameterization of the inputs defined in Grasshopper as illustrated in Figure 5.14 returns the result displayed in Figure 5.16. As visible the whole available height is used in the optimized design, and a structure is created that distributes the forces in a more balanced way. Node 4 has shifted significantly to the right, which has resulted in a load distribution where it takes 71.5% of the total vertical external load. This resulted in a compacter force diagram and a force distribution that is more concentrated with smaller forces in the members connected to node 3. Analyzing the value for the total load path, it can be determined that the total load path of the optimized design has a value of 85.9% of the initial design. This might seem like a relatively small improvement, but is important to keep in mind that the optimization setup was very restrictive and that the design itself already was quite efficient, be-

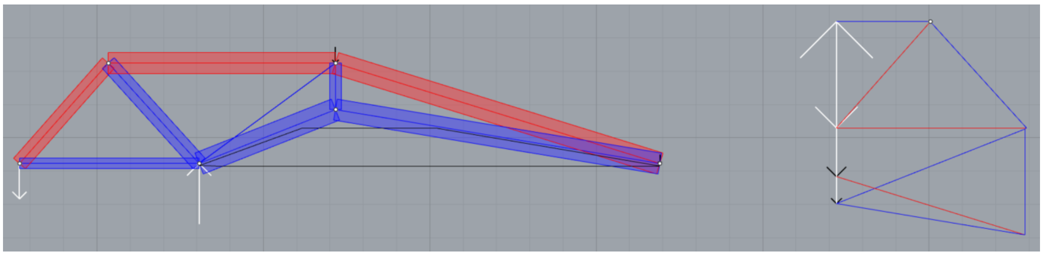


Figure 5.15: The unified and force diagram of the initial design of the auditorium roof structure in the GSDesign tool

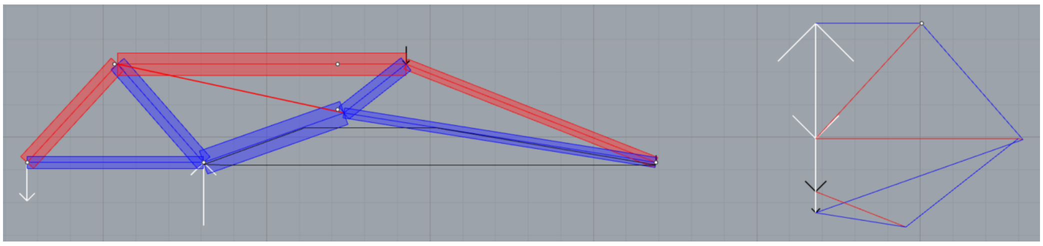


Figure 5.16: The unified and force diagram of the optimized design of the auditorium roof structure as computed and visualized by GSDesign

ing based on the actual design of the auditorium structure. Therefore an efficiency gain of 14% can be seen as a significant result, and would lead to substantial material saving considering the scale and necessary repetitions of this structural part.

5.2.2 Floor structure

Setup An analysis of the auditorium floor structural design initially proposed as a starting point for a Graphic Statics-based tool in Figure 3.5 returns the fact that the design is internally unstable by a factor of two, meaning that two members need to be added to make the truss structure valid. Using this knowledge the initially proposed design has been modified into the initial form diagram of Figure 5.17. The horizontal spacing between each node that carries an external force is exactly one fourth of the total horizontal length of the structure, so that the distribution of the total vertical floor load is indeed justified by the geometry. The initial design can be characterized as a truss structure, containing vertical elements and elements that more or less follow the shape of the auditorium floor. This design is relatively far away from the

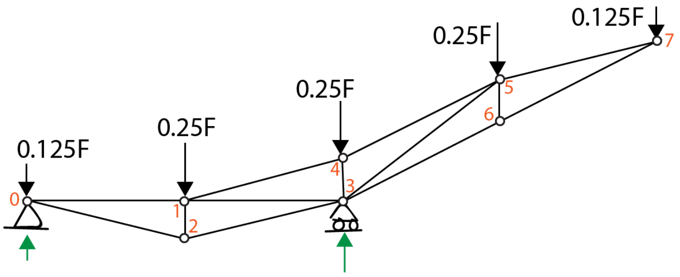


Figure 5.17: Initial form diagram of the TU Delft auditorium floor structure

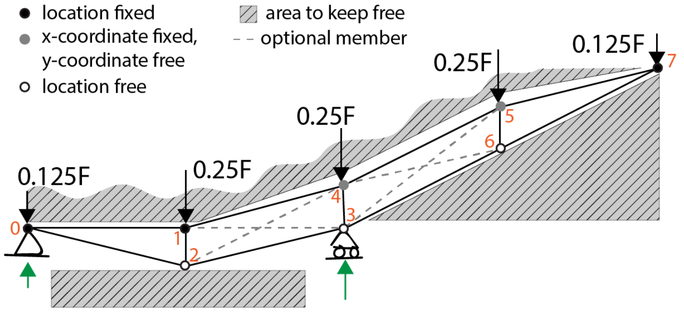


Figure 5.18: Optimization setup for the TU Delft auditorium floor structure

actual structure in place in the auditorium, which relies mostly on bending. This is not a problem, but it is important to keep in mind that a comparison of the optimized design with the initial design does not say anything about its performance compared to the existing floor structure.

Optimization The optimization setup is depicted in Figure 5.18. The nodes on which an external force is applied are all fixed in the x-direction to ensure that the same load distribution remains valid, while additionally nodes 0, 1 and 2 are also fixed in the y-direction to create a flat surface between nodes 0 and 1, and to ensure the desired connection to the roof structure at node 7. Apart from that, the joints are free to move, while be it in between the boundaries of the areas that need to be kept free. These areas are defined with the architectural requirements of the building in mind, and are also in place to maintain the slenderness of the initial design. In the quadrilateral space made up of nodes 1, 2, 3 and 4 a design can either contain a member between nodes 2 and 4 or 1 and 3. In the other quadrilateral space the same holds up for a member between nodes 3 and 5 or 4 and 6.

In order to show the parameterization of the locations of the nodes and the member connectivity, some small chunks of Grasshopper script are pre-

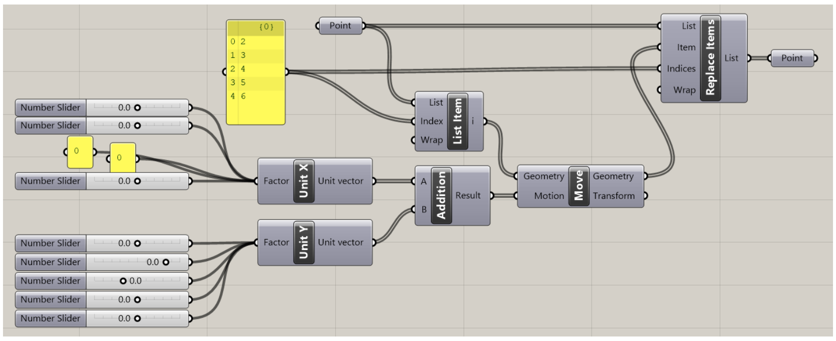


Figure 5.19: Definition of the joints of the auditorium floor structure in Grasshopper

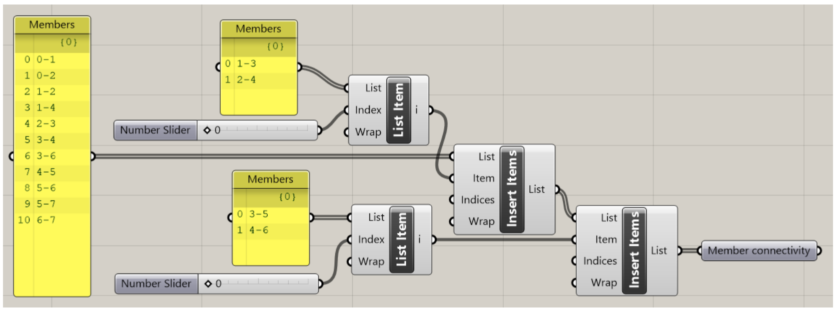


Figure 5.20: Definition of the connectivity of the auditorium floor structure in Grasshopper

sented Figure 5.19 displays the logic that takes the set of points defined in the Rhinoceros viewer, parameterizes the relevant nodes according to the described rules, and inserts them back into the point list replacing the original data. Figure 5.20 shows the definition of the connectivity, where each entry in the *'Members'* list depicts a member between two nodes. A slider with only the values 0 and 1 is used to choose which one of the additional members will be added to the list for each quadrilateral.

Results A direct visualization of the initial design processed by the GS-Design tool is shown in Figure 5.21. The unified and force diagrams display a very uneven load distribution with relatively high forces in four of the members, making it look like there is plenty of space for optimization. An optimization according to the described setup returns the design displayed in Figure 5.22. The force distribution is definitely more even and the emerged triangles look more logical and embody in essence a freeform 2D truss structure. A comparison of the total load path value reveals that the optimized

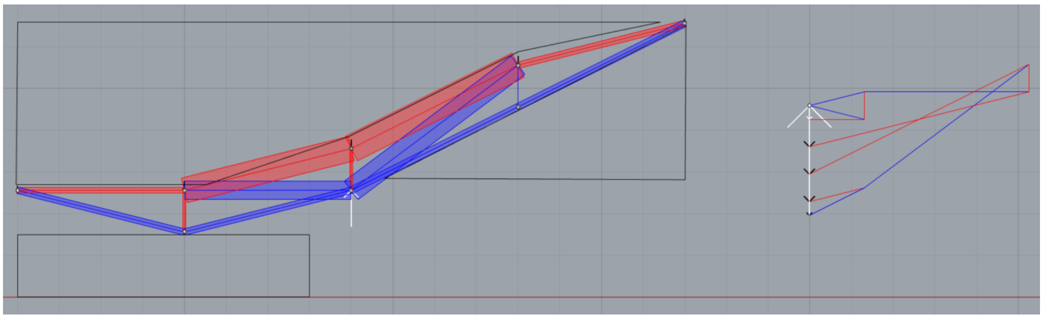


Figure 5.21: The unified and force diagram of the initial design of the auditorium floor structure in the GSDesign tool

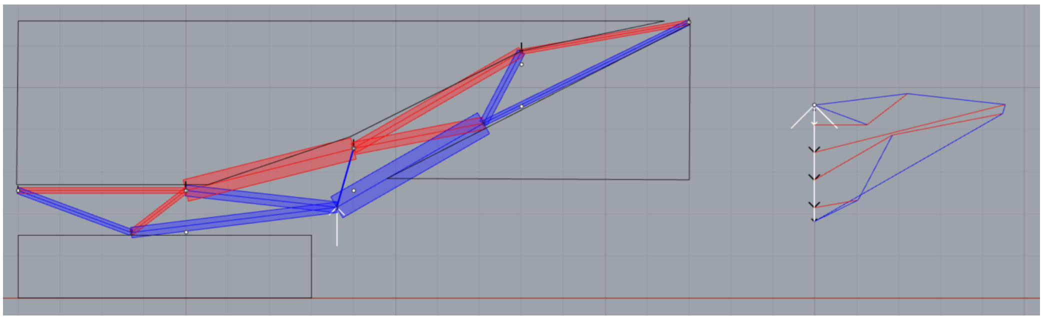


Figure 5.22: The unified and force diagram of the optimized design of the auditorium floor structure as computed and visualized by GSDesign

structure only requires 82.3% of the structural material compared to the initial design. If the triangular area on the right would be narrowed slightly as to allow some more width for the structure, the optimal design would look like the one in Figure 5.23, which has a total load path value of only 47.8% of the initial design. This massive drop in structural volume is impressive when regarding the small amount of extra space that was provided, indicating that the optimization process performs exponentially better when more freedom is given to the system.

5.3 User experiment

The presented benchmark cases and case study give insight in the basic functionality and the optimization process of the tool, but lack the capability to validate the practical value and user experience of GSDesign. In order

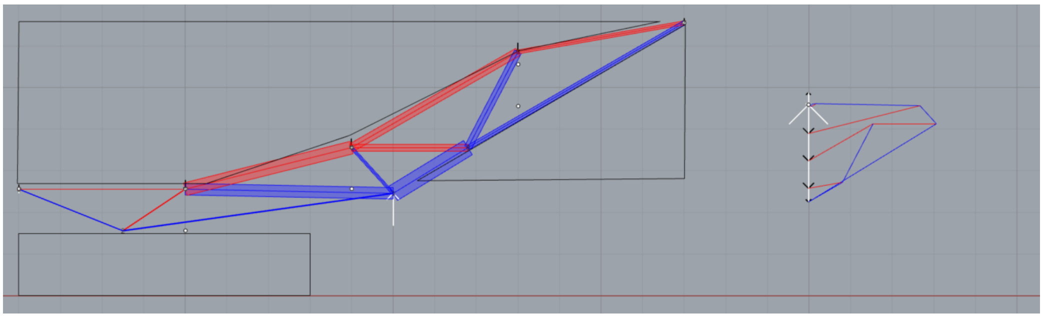


Figure 5.23: The unified and force diagram of the optimized design, similar to Figure 5.22, but with slightly more lenient definition of the triangular free area on the right

to say something substantive about this a user experiment is set up where the participants generate a conceptual structural design for a simple design case, walking through the complete design process according to the workflow described in Section 3.2.1. The objective of this experiment is to test the hypothesis that the tool helps structural designers to create and explore conceptual structural designs, providing users with new insights and guiding them towards structural efficiency structures in an actual design process. Additionally, other things like the user experience and intuitiveness of the tool are tested, although it must be noted that an optimal user experience was not one of the priorities during the development.

5.3.1 Experiment setup

The user experiment is carried out by a target group of structural engineers and designers that go through the design process of a conceptual structure for a simple design case, using the developed prototype GSDesign. This design case must be relatively simple and conceptual, in order to obtain meaningful results without getting lost in an abundance of variables. Furthermore, there should not be one obvious solution to the design problem, which instead must allow for different design directions. Naturally, due to the nature of the prototype, the design problem must be in 2D.

The design case displayed in Figure 5.24 is created specifically for this experiment to support these requirements. It concerns the supporting structure of a horizontal platform along a cliff. The platform is 4m wide and transfers a load of $25kN/m$ onto the structure in the depicted 2D plane. Five nodes are given as a starting point that directly support this platform with a spacing of $1m$, so that the middle three take $25kN$ and the other two $12.5kN$. The node on the left-hand side is also the first support; a roller at an angle of 30 degrees relative to the vertical axis. Another restriction is the given outline of the cliff, which cannot be crossed by any structural elements, and on which a second pinned support must be placed. This inclusion of the environmental surroundings is a good way to bring some asymmetry into the design problem, making the solutions space less obvious and more diverse. The nodes are supported in the out-of-plane direction, but buckling of the individual members between the nodes is possible and must be taken into consideration in the design process. The main objective for the participants is to design a conceptual structure that transfers the load to the supports as efficiently as possible, meaning to minimize the structural volume.

Structural engineers that were familiar with using Rhinoceros and Grasshopper were asked to take part in the experiment and had to work through a manual individually. This manual is presented in full in Appendix C, and contained a download link for all required files, installation instructions, an explanation of the functionality accompanied with a tutorial, and the instructions for the experiment. In total the experiment counted eleven participants.

The experiment is divided into four steps, namely:

1. Manual design
2. GSDesign
3. Submit results
4. Questionnaire

During the manual design step, participants are asked to generate an initial design for the design case using pen and paper or digital sketching software, optionally accompanied by any analogue or digital tool of their choice, only restricted by a time limit of fifteen minutes. A template pdf file of the design problem has been prepared for this step, displaying the starting situation.

When one or more initial manual designs are defined and the time limit has been reached, the developed tool GSDesign comes into play in step two. The participants are given a Rhinoceros and Grasshopper file containing a basic setup of the design problem, where they can easily recreate their own initial designs. The participants are asked to go through the Grasshopper file to make sure they understand it, after which they can start to import their own initial drawings and create these designs with the tool, in order to

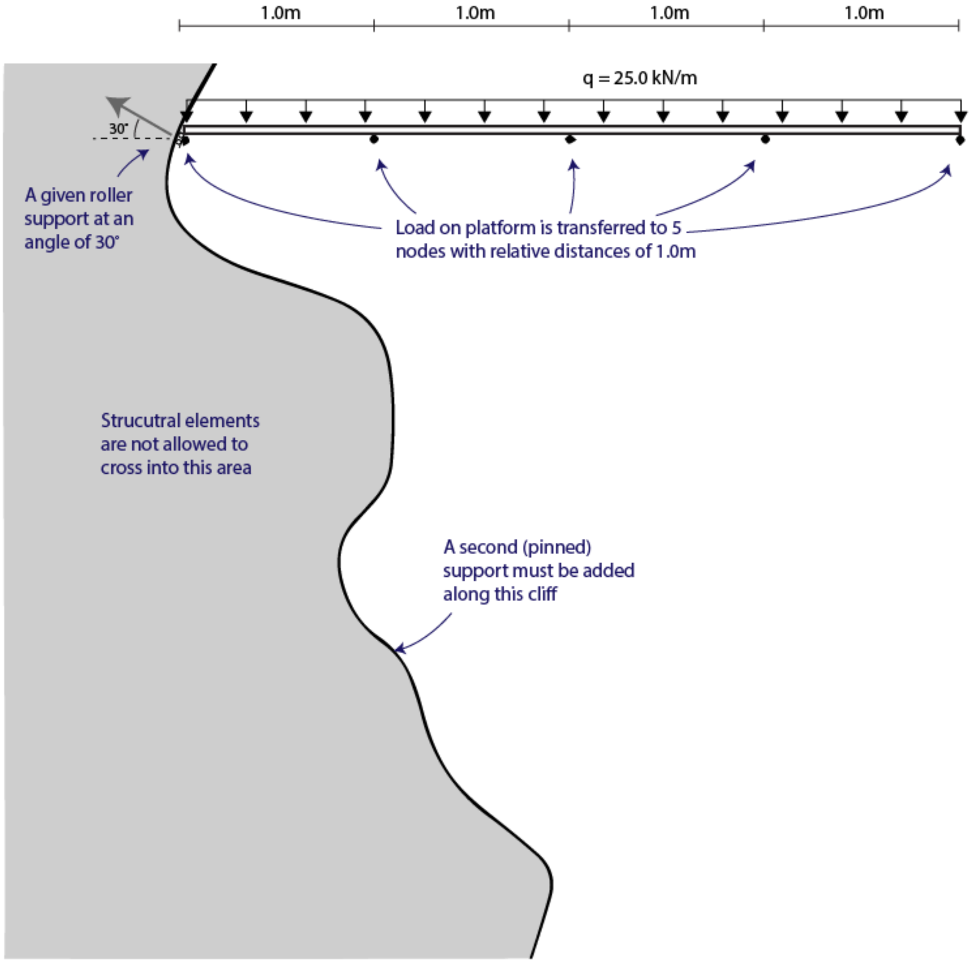


Figure 5.24: Setup of the design case of the user experiment

explore them further and optimize them as desired. For this step a time limit of thirty minutes has been defined, to make sure that the process supports a swift design process that is characteristic for the conceptual design phase.

The third step is the transmission of all relevant created design files to the author, which is essential for a proper analysis of the design data. Finally the participants are requested to fill in a questionnaire to create an understanding of how they experienced the design process and the results of the experiment, and how they value the developed tool in general. This information is then combined with the data generated in the design exercise, which concerns both the improvement in structural efficiency facilitated by the tool, as well as the change in structural geometry and topology between the initial and improved designs. For the multiple choice questions of the questionnaire a scale was used ranging from 1 to 6, which is deliberately chosen as an even number in order to avoid a neutral middle option, forcing the respondents to lean towards either one of the extremities. The only exception to this are the four questions where the respondents are asked to rate their own designs on efficiency and practicality; here a more familiar ten-point grading scale is used to allow more precision.

5.3.2 Experiment results

A complete overview of all results of the user experiment is presented in Appendix D, including images displaying the progression of conceptual structural designs accompanied with the fitness data, as well as an overview of the responses to the questionnaire. For the experiment sixteen different manual conceptual structural designs were created and thereafter optimized with GSDesign. Logically, for all cases use of GSDesign led to a design with a better fitness function, which ranged from 41% to 97% of the initial value, with an average of 69%.

Two of these design progressions are displayed in Figure 5.25 and Figure 5.26, which show how the workflow can work out for two different design ideas. The initial design of Figure 5.25 shows a very logical and systematic design, using even a grid of 1m by 1m. GSDesign takes this connectivity and optimizes the locations of the nodes into a structural design that looks more natural and organic, and utilizes more of the available structural height, suggesting a signficantly different structural topology and geometry. Figure 5.26 on the other hand shows a far simpler initial design concept, consisting only of one extra node in addition to the given ones. In this case the x and y-values of the extra node and the location of the second support along the cliff are the only three variables in the conducted optimization process, which is one of the reasons that the topology of the optimized structure is very similar to that of the initial design. However, despite these differences for both these design progressions the reduction in structural material needs was found at around 20%, showing that the design process with GSDesign is effective in suggesting new topologies as well as optimizing in a fixed topology.

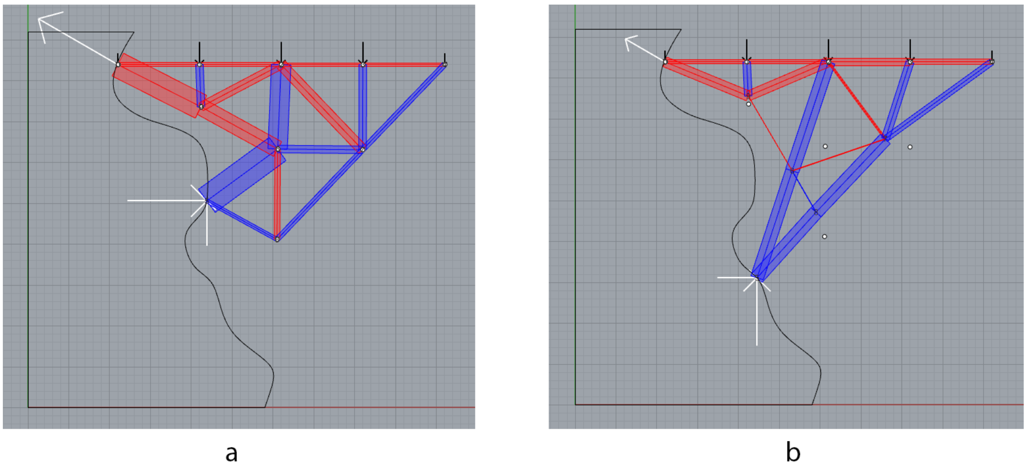


Figure 5.25: One of the design progressions created in the user experiment (design 1 of participant 2): **a** The initial manual design, **b** The optimized design

The questions in the questionnaire can be divided into three categories, namely regarding the first step of the experiment, the second step of the experiment, and the general opinion on GSDesign. In the first step of the experiment, where the initial manual designs are defined, respondents were asked about their opinion on these designs and how they experienced the given time limit. On average the participants were content on the set time limit of fifteen minutes, which turned out to provide them with enough time to make some designs, but was restrictive enough to ensure a high-paced design process and limit the use of other tools. In terms of structural efficiency the participants rated their structural designs in retrospect on average at 5.4 out of 10, while the practical feasibility of these designs was self-graded at a slightly higher average of 6.2 out of 10.

The thirty minutes given to the participants to try to improve these initial designs with the use of GSDesign was experienced quite balanced but just slightly on the restrictive side, at an average of 3.2 out of 6 where 1 meant "too restrictive" and 6 meant "more than enough time", indicating that

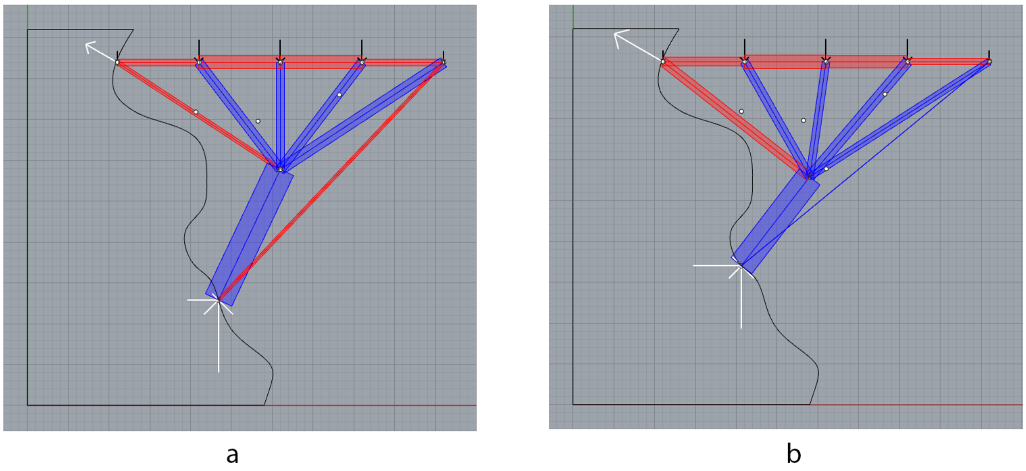


Figure 5.26: One of the design progressions created in the user experiment (design 2 of participant 4): **a** The initial manual design, **b** The optimized design

such a period of time is representative for most users when making a quick design for a structure of this scale with GSDesign. On the question “Did the use of GSDesign provide you with new insights regarding this design case?” the response was very positive, averaging a value of 4.8 out of 6, with the lowest response being a 3. When asked to rate the designs that they optimized with GSDesign, the average scores given regarding structural efficiency and practical feasibility were respectively 7.6 and 7.2, which is a significant increase in structural efficiency of 41%, while also improving the practical feasibility score of the designs with 16%. A comparison of the ratings of each initial and optimized design pair are displayed in Figure 5.27 for the structural efficiency and in Figure 5.28 for the practical feasibility, showing the design progression as perceived by the participants.

In the third section of the questionnaire on the general aspects of GSDesign, the participants rated the user-friendliness of the tool on average at 3.8 out of 6, which is fairly neutral but on the positive side. When asked about the practical value of the tool, the response was positive, averaging at 4.5 out of 6, where 1 meant ”no value at all” and 6 meant ”very valuable”. The educational value of the tool was rated even higher at an average of 5.3 out of 6 on the same scale.

Additionally, in the open questions, participants were asked about the strengths, weaknesses and missing features of GSDesign. Tables 5.1, 5.2, and 5.3 contain an overview of the responses to these questions, showing the reformulated and categorized main answers as well as how many times they were mentioned by different respondents.

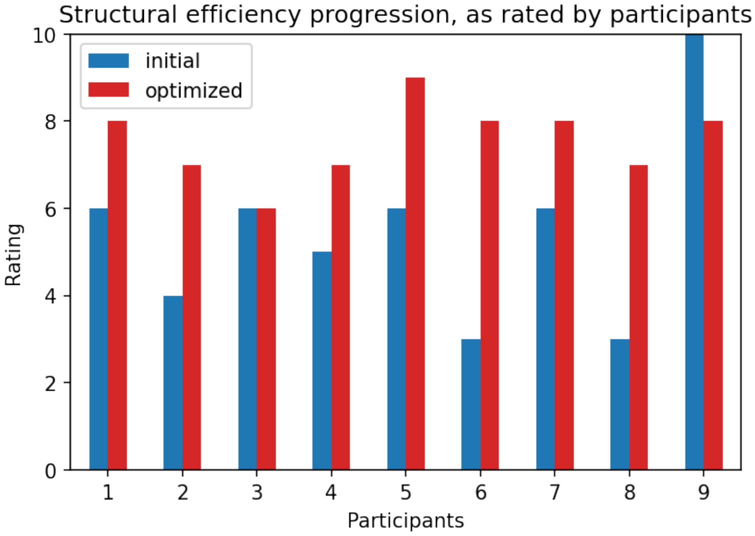


Figure 5.27: Progression of structural efficiency rating of the designs, as valued by the participants

Table 5.1: Analysis of strengths of GSDesign according to respondents

Category	Strength	Mentions
General	Accessible 2D Graphic Statics tool	1
	Direct link to structural mechanics through Graphic Statics	1
Design process	Human design input combined with optimization and computational efficiency	5
	Helpful insights towards efficiency	5
	Flexibility in optimization process	2
	Simple but effective fitness function	1
	Incorporation of buckling	1
	Fast, intuitive and easy to use	1
Visuals	Strong visuals of force flow	4
	Real time force diagrams (for learning Graphic Statics)	1

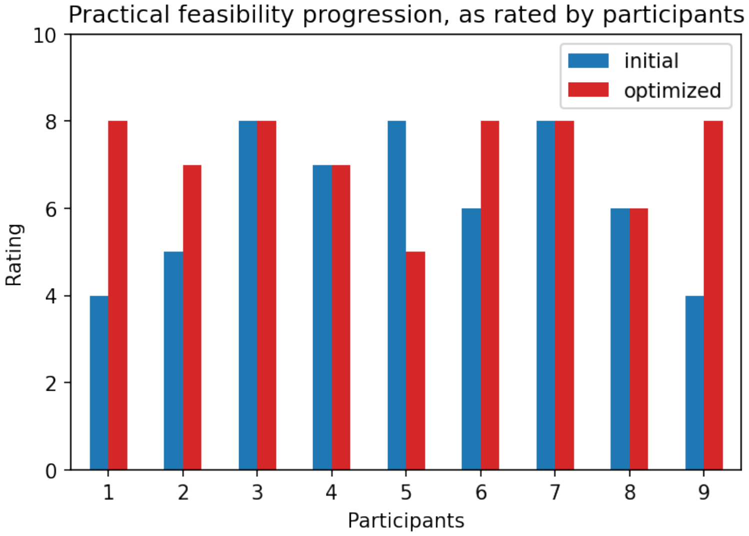


Figure 5.28: Progression of practical feasibility rating of the designs, as valued by the participants

Table 5.2: Analysis of weaknesses of GSDesign according to respondents

Category	Weakness	Mentions
User experience	Current layout of the tool with many inputs and outputs can be confusing for new users	2
	Users base is limited to people that know Rhino and Grasshopper	2
	The fitness function is very abstract, so performance is difficult to grasp	2
	Loading in a new design is a manual and slow process	1
	The process of the buckling calculation is unclear	1
Functionality	Optimization space is limited and not automated, with a fixed amount of members and joints per optimization sequence	2
	Limitation to statically determinate designs	2
	Little consideration of practical requirements and constructability	2
	Results are dependent on (possibly sub-optimal) human design input	1
	Limitation to 2D	1

Table 5.3: Analysis of missing features of GSDesign according to respondents

Category	Feature	Mentions
User experience	Improvement of member input methodology (draw instead of type)	3
	Automatic loading of a design based on sketch	1
	Include Bow's notation	1
	Include a human UI window to improve user-friendliness	1
Functionality	More automation in optimization, adjust solution topology after certain stage	2
	Handling of statical indeterminacy	1
	Include comparison of pre- vs post-optimization	1
	Allow moving loads and multiple load cases	1
	Compute and visualize deflection of frame	1
	More buckling options	1

5.3.3 Conclusions

The user experiment was created to test if GSDesign actually helps structural designers to create and explore conceptual structural designs, providing new insights and guiding users towards efficient structures. The design experiment showed that the optimized designs looked overall logical, balanced, and natural, and the accompanying data of the total load path shows that the optimized designs on average require only 69% of the structural material; a significant increase in efficiency. This progression in structural efficiency is acknowledged by the participants, who on average rated their optimized designs more than 40% higher regarding structural efficiency than their initial designs. When asked about the practical feasibility of their optimized designs the participants rated them on average at 7.2 out of 10, which surprisingly is 16% higher than the scores they gave to their self-created initial designs. Also the respondents were very positive about the new insights the tool provided, altogether making the overall positive effect of using GSDesign in a simple design process undeniable, strongly validating the hypothesis of the experiment. This statement becomes even more evident by the participants rating the practical value of the tool for structural design at 4.5 out of 6.

Additionally, more interesting conclusions can be obtained from the experiment. The general educational value of the tool was rated even higher than the practical value at 5.3 out of 6, which supports the claim that GSDesign offers clear visualizations of the Graphics Statics methodology and provides a good overview of the structural behavior. Furthermore the participants liked that the computational process starts out with human design input, which indeed is a distinctive characteristic of the tool compared to other optimization tools resulting ultimately in more logical and effective designs. Although this is a great strength of the tool, it can also be a pitfall for designers, since the quality of their initial design determines how optimal the designs are that an optimization with GSDesign can find. The tool gives responsibility to the user to lead the design process and come up with certain structural directions, not allowing them to be lazy and let the tool do the hard work of generating a design.

The "flexibility in the optimization process" was also noted by two participants, acknowledging the incorporated design freedom. On the other hand, further automation of the optimization space by suggesting adjustments to the solution topology of the structure during optimization was noted as a possible next step, which indeed could be an interesting topic for further development to improve the functionality and extend the guidance capacities.

The intuitive visualizations of the force flow were noted by four people as a strength, validating the feedback mechanism and the clear structural overview that the visualizations provide. The user friendliness was rated just positive of neutral, but when asked about the weaknesses of the tool the participants did mention that current layout of the tool can be confusing and that the process of creating a new design is rather manual and slow, clearly leaving some room for improvement. Also the definition of the connectivity was found tedious, where people preferred a more intuitive method than a text definition of the indices of the to be connected points. Also the fitness function led to divided opinions; where one participant valued it as simple and effective, two others found it confusing, in particular in combination with the buckling calculation, making the performance of a design hard to grasp. It would therefore be a good idea to make the fitness function more transparent, since it is important that users understand this value. Additionally, although the value of embedding the tool in Grasshopper was noted, also the weakness was mentioned that in order to use the tool people would have to have access to, and already be familiar with Rhinoceros and Grasshopper, limiting the accessibility.

5.4 Evaluation

5.4.1 Case study

While it is safe to say that the benchmarks cases of the single panel truss and the topology optimization-inspired cantilever prove that the GSDesign tool functions and handles optimization problems properly, the results of the case study of the auditorium structure require more context to correctly interpret. As stated previously the Delft University auditorium has a load bearing structure that acts in three dimensions, making it impossible to fully model its behavior in GSDesign. The presented 2D case study, as introduced in Section 3.1.3, is a reasonably valid simplification with explanatory and educational value, but its results cannot be directly applied on the actual building. The true added value of this case study is that of the journey from the in 2D depicted initial design towards an optimized design, using all the freedom that the GSDesign tool provides, while adhering to predetermined practical requirements.

The case study mainly focuses on the optimization process, which returned promising results and noticeable gains in efficiency, with designs that appeared logical and exhibited a natural aesthetic. Another notable trait of GSDesign is that multiple optimization parameters can be combined with some resourceful Grasshopper scripting, optimizing not only the locations of the nodes, but also how these nodes are connected.

5.4.2 User experiment

The design experiment has produced positive results and validates the concepts incorporated in the prototype GSDesign. Although the experiment was rather small-scale with only eleven participants, this does not undermine the validity of the experiment. Nielsen [2000] even argues that small use tests with only five participants are ideal to effectively determine the usability of a digital tool, and that a testing pool of fifteen users will results in finding 99% of the relevant issues and usage data, meaning that with this user pool of eleven the results should be rather complete and definitively representative.

In the first step of the experiment there were clear differences noticeable in the approach of creating the initial design between the participants. Where some would just sketch something acceptable from the cuff leaving obvious room for improvement, others created more informed initial designs using a layout generator tool called LayOpt based on the ground structure method [Fairclough et al., 2020], and the live physics engine Kangaroo [Piker, 2017] for initial form-finding. It was interesting to see that even these very informed initial designs could be optimized with GSDesign, although the process of recreating the initial design to fit the Graphic Statics framework could at times be described as cumbersome. The main difficulty in creating these structural design in GSDesign is that the generated designs were in these cases not statically determinate, meaning that members had to be added, removed or shuffled around to make it so, deviating from their initially optimal but often impractical concepts.

Another interesting shortcoming that came to light during the experiment is an oversight in the solving methodology of the member forces. As described in Section 4.2.2 in the Solve Member Forces component every yet unsolved node is analyzed and solved if it contains two or less unknowns, finding the forces in the members that make up the joint. If a joint counts three or more unknown members it is skipped because there is an inifinite amount of possible equilibrium states, coming back to it later when more member forces are known. This method works perfectly in most cases, but as Figure 5.29 shows a situation can occur where the solving process gets stuck because all yet unsolved nodes (2, 5, 6, 7, 8, and 9) have three or more unknowns, even though the design as whole is statically determinate. This unfortunate event occurred three times in the experiment, and led to the situation that two participants could not properly complete the experiment, therefore his or her responses to the experiment-related questions of the questionnaire were not included in this report. A fix for these situations will not be in the scope of this research project, but could possibly be achieved by assuming one of the unknown forces, followed by an iterative process towards the one and only possible correct equilibrium state.

A final interesting discussion topic is the relatively high educational value that GSDesign was awarded in the questionnaire. The main objective of this research project was to develop a design tool, but seeing the tool in an educational context is also very interesting and promising. Future development on this tool could potentially be focused more on this aspect, defining and developing certain functions that are important when using the tool for an educational purpose.

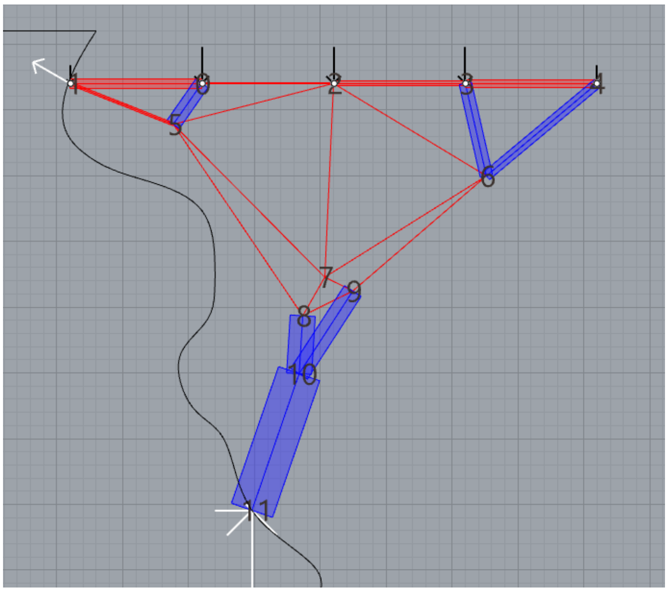


Figure 5.29: A design produced in the user experiment for which the current solving method is inadequate, being unable to deal with the situation where all remaining unsolved nodes have three or more unknowns

Chapter 6

Discussion

In this chapter the meaning, importance and relevance of the results are discussed. After an initial reflection on the objective and research questions, the limitations of the acquired results are mentioned, concerning both the developed prototype and the fundamentals of the concept. The chapter is concluded with a discussion of the effect of the results on sustainability related issues.

6.1 Objectives

The main objective of this research project has been defined in the second chapter as:

> *To develop a prototype of a conceptual force flow design tool supporting the principles of feedback, guidance, design freedom and structural overview.*

This objective has been achieved by the development of a design tool that uses the principles of Graphic Statics, and is built on the software platform Grasshopper, a plugin of Rhinoceros. The necessary research that underpins these development choices are structured by three research questions, which will now be reflected on.

> *1. What is an appropriate method to model the force flow?*

There are multiple methods in use in engineering that can be applied to model a force flow or load path. No guidebook exists on the kind of force flow modelling that this research project requires, so an exploration of multiple methods was carried out, coming eventually to the definition of three fundamentally different methods that could potentially be used as the basis of the computational tool. This initial exploration was performed by applying the basics of structural engineering and cross-referencing possible methods with the requirements of the design tool, and was further aided by multiple discussions with professors and structural engineers. The three methods that came out of this exploration (Graphic Statics, drawing vectors, and system of differential equations) were elaborated and compared in a practical sense using a predetermined design case. The use of the design case brought to light the different characteristics of each method, making clear also its implications on the design process and the system architecture. Using these insights, the strong and weak aspects of each method were presented, and valued against the defined requirements, resulting in the selection of Graphic Statics as the most appropriate option.

> *2. How can the computational tool take shape to support conceptual force flow design and meet the defined requirements?*

To answer this question, the choice of Graphic Statics as force flow modelling method was used as a starting point. Adhering to the requirements of this method a design workflow for the tool was worked out, proposing in a rather detailed way how the tool must be used. During the conception of this workflow the four essential principles to implement in new computational design tools to properly support the engineer in the conceptual structural design phase, as defined in the introduction, were ensured to be incorporated. Additionally the statical determinacy requirements of Graphic Statics were defined, since it is important to incorporate this information during development.

> *3. What is a suitable framework for the development of the conceptual design tool?*

To define the meaning of suitable in this context, the available literature was studied, especially looking at the previous StructuralComponents research projects. In relation to this project, five main requirements were defined for the development process, namely good accessibility, the incorporation of real-time modelling, geometric flexibility in design, extensibility, and presentation independence. Using these requirements and personal preferences of the author the choice for the software to develop the tool on fell on Grasshopper. After some initial explorations in development using individual Python components, it was decided to develop the tool predominantly as custom components coded in C# in Visual Studio, allowing more control and structure in this process. Additionally, the visualizations of the results were incorporated separately as Grasshopper clusters, using native components to shorten development time.

6.2 Limitations

Here the limitations of the developed prototype and the underlying theory are discussed, as well as possible next steps for future development.

Functionality of the developed prototype

- Only designs with exactly two supports are allowed now. A next step would be to include more freedom in the support definition allowing also designs with three support points. These supports must all be rollers for the global structure to be statically determinate.
- When an external force is applied on an internal node, the developed tool does not know how to order the forces in that joint, resulting more often than not in a disorderly and non-closing overall force diagram. This is inconvenient, although not disastrous since in these cases the solved forces and the joint force diagrams are still completely correct. However, it would be nice to investigate this and incorporate a solution in the next iteration.
- In the prototype most functionality of GSDesign is available as custom Grasshopper components stored in a plugin, and some as Grasshopper clusters. This separation can be inconvenient and confusing for the user. The next step towards a more professional tool would be to redefine the GSDesign Grasshopper clusters as custom components, thereby making sure that all GSDesign functionality is available in one plugin.
- The tool is developed for Grasshopper, which has many advantages but has the downside that the user base is limited to people that have access to Rhinoceros and are familiar with Grasshopper.
- The current version of GSDesign does not incorporate the capability to properly compare two or more designs like for example the initial design and the optimized design. This is a difficult feature to achieve in Grasshopper, which in general does not support the existence of multiple design variants very well. However, some mechanisms and

workarounds are possible to compare the data of multiple design variants in an insightful way, which could be interesting to investigate and incorporate in the future.

Fundamentals

- Dead loads of the structure are in general not considered in Graphic Statics and are difficult to incorporate since it would require assumptions on cross-sectional and material properties. For this reason, dead loads are also not included in the developed design tool, while their effect can be very significant for certain designs. Incorporation of a feature that includes dead weight would be very informative, but does possibly infringe the fundamental and conceptual nature of the design tool.

- The developed tool now has the strict requirement that a design must be statically determinate externally and internally, which can be very limiting in the design process. Internal statical determinacy will always be a requirement if one wishes to use Graphic Statics to analyze a structure, but external statical determinacy is not always necessary for the methodology to work. Furthermore, there are also statically determinate structures which cannot be solved with the developed tool, due to a circumstantial absence of joints with two or one unknowns. A next step could be to investigate this and possibly expand the solving process of the tool using the acquired information.

- Buckling is an important factor in the design of normal force-based structures, but can be difficult to apply in early design. The incorporated buckling calculation in GSDesign seems reasonably representative but has not been properly validated. The main objection that one could have against the applied method is that it only uses the length of the member as an input, and not the area and second moment of inertia, which are the other geometric aspects important for buckling according to Euler's formula. The reason for not taking these variables into account is that it would require many assumptions which would be difficult to substantiate and likely too specific for the conceptual design stage where this tool is intended for. However, the next step to incorporate buckling effects in a more correct way could be to try to link the computed member force to the mentioned cross-sectional geometric properties, quantifying the Euler's critical buckling load that way, while taking care that the tool upholds its intuitive and flexible character.

- The tool is only developed to support 2D problems. Some structures can indeed be properly approximated as two-dimensional, but the reality is that structures exist in three-dimensional space, and therefore must be designed in 3D to achieve the best results. A next step would be to try to incorporate the same principles in three dimensions.

6.3 Sustainability

In Section 1.2.3 three main areas of improvement were defined related to sustainable development where this research project could contribute. The fist and most obvious of these three is reducing construction-related carbon emissions, which mainly refers to the embodied carbon in structural materials like steel and concrete. A reduction of the total structural material volume in a building would linearly reduce these emissions, which is exactly what the developed design tool aims to facilitate. The benchmark cases have shown that the functionality of the tool works and the case study illustrates that a realistic initial design with realistic boundary conditions can be optimized into a design that requires just 50% to 80% of the structural material, depending on how strict the boundaries are defined. The user experiment shows that actual structural engineers can optimize their manually made initial designs into a structure that require on average just 69% of the material for a simple cantilever design case.

Although these numbers seem promising it is worth noting that these design cases are still fairly simple and conceptual, and that more development is necessary for the tool to be able to solve practical design problems on a larger scale. As discussed in the previous sections of this chapter, the tool still has some limitations that impede it from reaching its full potential. Most notably, an extension into 3D would make the tool a lot more applicable as an actual design tool for different kinds of structures. Also, it must be noted that although the depicted material efficiency gain is a good step, it is on itself just a tool to make the industry a little less polluting. The only way to make the building industry closer to carbon-neutral or even carbon-negative is to combine design research like this with other research projects into topics like sustainable material and construction methods. In the coming decades massive innovation on every front in the AEC industry is required to effectively face the threats that pollution, resource scarcity, and material waste pose on our society.

The second defined inefficiency lies in the generation and documentation of complete design data, which is vital to properly assess certain sustainability based performances. This current issue is only partly solved by the developed tool, which on the one hand generates a lot of useful data regarding the conceptual structural design, obtaining information about the efficiency of a geometry and topology. This is an important step since there is often little information available of the conceptual design phase.

Proper documentation of the design process on the other hand is not directly incorporated in the prototype, and is currently the full responsibility of the user. Grasshopper, the platform on which the prototype was developed, does not really support the existence and direct comparison of multiple design variants in one file, and is not effective in documenting the progression of a design. However, initiatives such as Inkbeagle [Loos, 2020], BHoM [Fisher, 2020], and ideas discussed by Coenders [2011], can be applied to incorporate this functionality better in Grasshopper or a similar framework, meaning that the process of continuously storing and comparing design information will become easier.

The final point made was that improving the communication and clarity in the design process will counteract the inefficiency and chaos of building design, and therefore improves the ability to achieve (sustainability based) design goals. The developed design tool achieves this by providing clear conceptual structural designs, that give a great overview of the structural behavior and are easy to understand by anyone involved in the design. Additionally, the early structural integration that this tool provides makes theoretically for a more structured and efficient design process with less iterations and last-minute changes. These statements are substantiated by the results of the user experiment and the case study, but can only be truly validated by how the tool is used and perceived in an actual multidisciplinary design team.

Chapter 7

Conclusions and recommendations

This chapter finalizes the research project by drawing a conclusion regarding the main objective, as well as addressing the defined research questions. The chapter concludes with the formulation of a set of recommendations for future efforts on this research topic.

7.1 Conclusions

The main objective was to develop a prototype of a conceptual force flow design tool supporting the principles of feedback, guidance, design freedom and structural overview. This prototype has been developed as a plugin for the Grasshopper environment under the name GSDesign. For the analysis GSDesign uses the principles of Graphic Statics to solve and visualize the results, making them not only easy to interpret but also fast to generate. It facilitates the design of truss-like structures consisting of elements that only bear normal forces, which can be interpreted as force flow designs. The tool supports a fast and flexible generation of designs, real-time analysis, and requires no material or cross-sectional properties as input, making the tool extremely suitable for conceptual design.

The feedback system is automated with a standardized but customizable visualization sequence relying on the graphic principles of Graphic Statics upgraded with a color scheme to differentiate between tensile, compression, external, and support forces.

The principle of guidance is in the first place incorporated intrinsically in the method of Graphic Statics, which helps to steer a designer in the right direction for two reasons. First, this method only uses pure normal force action which is, if used correctly, always more efficient in material use than relying on bending. Second, the force diagram and unified diagram visualizations help the designer in understanding the structure more intuitively. Additionally to these principles, GSDesign also offers the possibility of optimization, by combining GSDesign with an exisiting (genetic) optimization component. The functionality of this optimization process has been validated and offers high speed, a good workflow, and most importantly a great amount of freedom in the selection of design parameters.

The principle of design freedom is present in the flexibility that the design tool offers. Users can set up their design by their own preferences, and design variations can be scripted using Grasshopper. The optimization process suggests certain optimal designs, but does not enforce these decisions and leaves the setup of the optimization and the choice for certain design directions to the user. The user must logically adhere to the basic Grafic Statics rules, which can feel limiting but on the other hand encourage a certain simplicity and logic in designs, while in the end also leaving infinite possibilities to explore different designs. The main limitation is that the designed structures must be statically determinate.

A basic structural overview is obtained by the visualizations incorporated in the tool, which are facilitated by the different viewer options available to provide the desired display for any situation. The unified diagram particularly makes the force flow of a structure clear at one glance for any internal or external party, while the force diagram adds valuable insight during the force flow design process.

To aid the process of fulfilling the main objective three research questions have been formulated. Conclusions regarding these research questions are provided here.

What is an appropriate method to model the force flow?

Graphic Statics is deemed the most appropriate method because of its fundamental intuitive principles, the fast setup and solving process, and the possibilities for optimization, making it very suitable for a computational tool with reliable high quality feedback and excellent guidance capacities. Also the intrinsic qualities of Graphic Statics are a good match with the efficiency objectives tied to this research project. Considered alternatives were the drawings vectors and system of differential equations method. For more information on these methods the reader is referred to Section 3.1.1.

How can the computational tool take shape to support conceptual force flow design and meet the defined requirements?

The choice of Graphic Statics as the base for the computational tool imposes a certain setup. For the solving process to initiate, a complete design must be in place, containing joints, members, external forces and supports, forming a statically determinate structure. In order to develop a viable computational tool that incorporates Graphic Statics-specific requirements as well as the earlier defined general requirements, a workflow has been defined. The workflow starts with the user identifying a design problem where a 2D Graphic Statics tool could contribute, meaning that a statically determinate conceptual structural design can be made using only normal force and discrete members connected with pin joints. The user then manually makes an initial design sketch, assisted by any preferred available analog or digital tool that fits the design process, identifying also all the boundary conditions, supports, and governing loads. The next step involves a translation from this manual design into the computational tool. Here the design is immediately solved when correctly defined, after which the user can start exploring this design manually or try to optimize it.

What is a suitable framework for the development of the conceptual design tool?

Five requirements have been defined that are vital for the development process, namely accessibility, real-time modelling, geometric flexibility, extensibility, and presentation independence. Grasshopper, a plugin of Rhinoceros, has been identified as a platform on which the tool can be developed that could meet all of these criteria, making it the logical software base. The tool has been developed mainly as a Grasshopper plugin, with different components written in the C# programming language, making use of the .NET framework and the RhinoCommon API. Additionally, since Grasshopper contains very useful visualization components, the visualization parts of the design tool are available as Grasshopper clusters instead of custom components, which does impair the manageability of the tool but significantly reduces development time.

Finally, a user experiment has been carried out by structural engineers who were asked to make a design for a simple two-dimensional cantilever design case, first by hand and subsequently with the developed prototype. The progressions of the designs by eleven structural engineers were analyzed qualitatively and quantitatively, and combined with the data of a questionnaire with questions about their perceptions regarding the created designs, and the value, practicality and limitations of the tool.

The results of the experiment were primarily positive, with the fitness value of the optimized designs averaging at 69% of the initial manual designs, meaning that tool facilitated a structural material volume reduction of 31%. The participants endorsed this outcome of the efficiency improvement, rating their own optimized designs on average 41% higher than their initial manual designs on structural efficiency, without compromising on practical feasibility. Furthermore the designs that were optimized with GSDesign showed a natural, balanced and logical structural aesthetic, in many cases leading to new structural topologies and, according to the questionnaire, new insights among the users. Furthermore the respondents strongly acknowledged the practical value that GSDesign can have for the structural design process and assessed the educational value of the tool even higher. The workflow of combining human design input with optimization came out as a key strength of the tool, as well as the insight it provides towards efficiency and the strong indicative visuals, which definitively validate the guidance capacity and structural overview that the tool provides. The key limitations as experienced by the respondents are the fixed optimization space, the limitation to statically determinate designs and the manual time-consuming design input methodology.

7.2 Recommendations

GSDesign is a functional prototype of the envisioned conceptual force flow design tool, but still has much potential to perform better and incorporate extended functionality. Furthermore, there are multiple other research directions that were kept out of the scope of this research which would be fascinating to investigate. To present the recommendations for future research orderly, a division has been made into two categories, namely additions and extensions on the current GSDesign prototype, and research contributions on a larger scale that not necessarily fit in the presented framework.

Extensions on GSDesign

- The tool generates very valuable information for the conceptual structural design, but does not offer a good way to store the design data of different variants to present the full design story as envisioned in Section 1.2.2. This is partly due to the nature of Grasshopper, but nevertheless this would be an interesting topic to investigate and could maybe entail a combination of this research project with Structural-Components 4 [Bovenberg, 2015].

- The prototype currently cannot properly handle external forces on internal nodes. Investigating and incorporating this functionality would make GSDesign more complete.
- The option to define multiple load cases and possibly even moving loads could be interesting to incorporate. This addition also poses interesting challenges regarding the visualizations and comparison of force flows for these different loads.
- A new or complimentary solving process could be included that can solve any internally statically determinate structure, even if it has more than three support reactions or if it has no joints with two or less unknowns.
- In terms of user experience, improvements can be made by simplifying the inputs and outputs of the components of the tool, maybe splitting the components into multiple ones or adding a human user interface window. Also a faster and more intuitive member definition, and a more automated process of creating manually made designs in the tool based on an image would greatly improve the user-friendliness.
- The option to display Bow's notation [Bow, 1873] could be included, which is a labelling convention that links lines in the force and form diagrams to each other.

Large-scale attributions

- Buckling effects are currently incorporated as good as possible without making assumption on cross-sectional and material properties, providing results that are approximately correct when the user provides the right parameter, based on the forces in the structural elements, and the intended material and design application. It would be an interesting project to incorporate Euler buckling in a more correct way, making assumptions that are representative for most cases without compromising the fast workflow and conceptual nature of the tool. Since the inclusion of dead weight deals with the same issues of the necessity of material and cross-sectional properties, it could be incorporated at the same time.

- The optimization process as described in this report and as used in the user experiment is rather basic. Further research could be done in a more automated process where the solution topology is updated after a certain stage, for example when nodes are colliding or if certain members are not carrying any force. Also presets could be developed to allow multiple possible optimization goals next to the total load path, like minimizing the support reactions or even distribution of forces.

- A 3D version of the tool would greatly enlarge the applicability of GSDesign on real-life design problems. It must be investigated if the vector-based method used currently is also suitable for three dimensional space, looking at the solving process, but also at visualization and design aspects. Another approach would be to use polyhedrons to represent the equilibrium of a node, as described by Lee [2018], possibly applying a similar design-based approach to this concept.

Bibliography

N. Aders. TU Delft: Aula Congrescentrum, 2012. URL `https://commons.wikimedia.org/wiki/Category:Aula,_TU_Delft#/media/File:TU-Delft-Aula.jpg`.

S. Adriaenssens. Numerical Form Finding, 2020.

R. Aish. Introduction to GenerativeComponents: A parametric and associative design system for architecture, building engineering and digital fabrication, 2005.

Autodesk. Revit, 2020a. URL `https://www.autodesk.nl/products/revit/overview`.

Autodesk. Robot Structural Analysis Professional, 2020b. URL `https://www.autodesk.com/products/robot-structural-analysis/overview`.

L. Beghini, J. Carrion, A. Beghini, A. Mazurek, and W. Baker. Structural optimization using graphic statics. *Structural and Multidisciplinary Optimization*, 49:351–366, Mar. 2014. doi: 10.1007/s00158-013-1002-x.

P. Block. Digital Master Builders - Disruptive construction technologies, Nov. 2018. URL `https://www.youtube.com/watch?v=PPqKmILmz_o`.

A. C. Bovenberg. StructuralComponents 4: Conceptual building models with structural design justification. Master's thesis, Delft University of Technology, 2015. URL `https://repository.tudelft.nl/islandora/object/uuid%3A663ca19b-ddac-4d46-8e04-1c84386d3a0c`.

R. H. Bow. *Economics of construction in relation to framed structures*. London: E. & F. N. Spon 48 Charing Cross, New York: 446 Broome Street, Edinburgh: Adam & Charles Black, 1873. URL `http://archive.org/details/cu31924004138123`.

R. Bregman. *De meeste mensen deugen*. De Correspondent, 2019. ISBN 978-90-829421-8-7.

J. Breider. Structural Components - Development of parametric associative design tools for the structural design of high-rise buildings. Master's thesis, Delft University of Technology, Delft, 2008.

K. Brown and K. Rasmussen. The Sustainable Development Goals in 2019: People, Planet, Prosperity in Focus, July 2019. URL `https://unfoundation.org/blog/post/the-sustainable-development-goals-in-2019-people-planet-prosperity-in-focus/`.

G. H. Brundtland. Our Common Future—Call for Action. *Environmental Conservation*, 14(4):291–294, 1987. ISSN 1469-4387, 0376-8929. doi: 10.1017/S0376892900016805. URL `http://www.cambridge.org/core/journals/environmental-conservation/article/our-common-futurecall-for-action/65808D6676E07552EF891DF31C3DF7A1`.

P. Coad and E. Yourdon. *Object-oriented analysis*. Yourdon Press, USA, 1990. ISBN 978-0-13-629122-0.

J. Coenders and L. Wagemans. Structural Design Tools: The next step in modelling for structural design. *report (iabse symp.)*, 92(12):57–64, 2006. ISSN 2221-3783. doi: 10.2749/222137806796184860. URL `http://www.ingentaconnect.com/content/10.2749/222137806796184860`.

J. L. Coenders. *NetworkedDesign: Next generation infrastructure for computational design*. Doctoral Thesis, Delft University of Technology, 2011. URL `https://repository.tudelft.nl/islandora/object/uuid%3Abf930678-d676-4c4f-85c1-c7188b6718f7`.

I. Computers and Structures. ETABS, 2019. URL `https://www.csiamerica.com/products/etabs`.

I. Computers and Structures. Structural Software for Analysis and Design | SAP2000, 2020. URL `https://www.csiamerica.com/products/sap2000`.

K. Culmann. *Die Graphische Statik*. Verlag Meyer und Zeller, Zürich, 1st ed. edition, 1866.

J.-P. Den Hollander. Voorspanning in auditorium. *Bouwwereld*, 22:44–46, 1964. URL `http://www.jpdh.info/pdf/Bouwwereld100_AulaTUDelft.PDF`.

L. Dierker Viik. StructuralComponents 6: An early-stage design tool for flexible topologies of mid-rise concrete buildings. Master's thesis, Delft University of Technology, 2019. URL `https://repository.tudelft.nl/islandora/object/uuid%3Aa582d714-bd9c-48e5-9a74-36de7ec4e3c9`.

C. Drucker. Load Paths! The Most Common Source of Engineering Errors, Apr. 2014. URL `https://www.aisc.org/globalassets/continuing-education/quiz-handouts/load-path-common-source-of-engeneering-errors.pdf`.

J. Elkington. *Cannibals with forks : the triple bottom line of 21st century business.* Capstone Pub, Oxford, 1997. ISBN 978-1-900961-27-1.

H. Fairclough, T. Pritchard, L. He, and M. Gilbert. LayOpt: A truss layout optimization web application, 2020. URL `https://www.layopt.com`.

A. Fisher. Buildings and Habitats object Model, Aug. 2020. URL `https://github.com/BHoM/documentation/wiki`.

S. Greenwold, E. Allen, and W. Zalewski. Active Statics, 2003. URL `https://acg.media.mit.edu/people/simong/statics/data/`.

C. Hartsuijker. *Toegepaste Mechanica Deel 1: Evenwicht.* Academic Service, Den Haag, 1 edition, 2012. ISBN 978-90-395-0593-9. URL `https://www.libris.nl/boek/?authortitle=c-hartsuijker%2ftoegepaste-mechanica-1-evenwicht-leerlingenboek--9789039505939`.

B. Hohrath. StructuralComponents 5.0: Super element based tool for early design collaboration applied to mid-rise buildings. Master's thesis, Delft University of Technology, 2018. URL `https://repository.tudelft.nl/islandora/object/uuid%3A125bab83-8209-4459-8538-7d79b7976b9b`.

International Energy Agency. 2018 Global Status Report - towards a zero-emmission, efficient and resilient buildings and construction sector. Technical report, United Nations Environment Programme, 2018. URL `https://wedocs.unep.org/bitstream/handle/20.500.11822/27140/Global_Status_2018.pdf?sequence=1&isAllowed=y`.

J. D. Jansen, A. Rolvink, J. L. Coenders, and M. Schevenels. Interactive distributed optimisation for multidisciplinary design. *Proceedings of the IASS-SLTE 2014 Symposium "Shells, Membranes and Spatial Structures: Footprints", Brasilia, Brazil, 15-19 September 2014*, 2014. URL `https://repository.tudelft.nl/islandora/object/uuid%3A1a4968fc-5aac-4999-98ca-5826b201a9cd`.

S. C. Joyce. Web Based Data Visualisation Applied to Creative Decision Making in Parametric Structural Design. In *Proceedings of IASS Annual Symposia*, page 13, Amsterdam, 2015.

L. Kappert. Programming principles, Nov. 2020. URL `https://github.com/webpro/programming-principles`. original-date: 2014-10-24T21:20:48Z.

A. Kilian and J. Ochsendorf. Particle-Spring System for Structural Form Finding. *Journal of the International Association for Shell and Spatial Structures*, 46(147):8, 2005.

J. Lee. *Computational Design Framework for 3D Graphic Statics*. Doctoral Thesis, ETH Zurich, 2018. URL `https://www.research-collection.ethz.ch/handle/20.500.11850/331210`.

Y. C. Liu, A. Chakrabarti, and T. Bligh. Towards an 'ideal' approach for concept generation. *Design Studies*, 24(4):341–355, July 2003. ISSN 0142-694X. doi: 10.1016/S0142-694X(03)00003-6. URL `http://www.sciencedirect.com/science/article/pii/S0142694X03000036`.

L. Loos. inkbeagle.io, 2020. URL `https://inkbeagle.io/`.

J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transaction On Graphics (Tog)*, 5(2):110–141, 1986.

P. MacLeamy. The future of the building industry (3/5): The Effort Curve, 2010. URL `https://www.youtube.com/watch?v=9bUlBYc_Gl4`.

K. Martini. A New Kind of Software for Teaching Structural Behavior and Design. In *Proceedings of the 2006 Building Technology Educators Conference*, College park, Maryland, Aug. 2006.

K. Martini. Arcade: Interactive Non-linear Structural Analysis and Animation, 2009.

J. C. Maxwell. On reciprocal figures and diagrams of forces. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 27(182):250–261, Apr. 1864. ISSN 1941-5982. doi: 10.1080/14786446408643663. URL `https://doi.org/10.1080/14786446408643663`.

McNeel. Rhinoceros, 2020a. URL `https://www.rhino3d.com`.

R. McNeel. RhinoCommon API, Nov. 2020b. URL `https://developer.rhino3d.com/api/RhinoCommon/html/R_Project_RhinoCommon.htm`.

J. Mirtschin. Geometry Gym, 2020. URL `https://geometrygym.wordpress.com/`.

C. T. Mueller. *Computational exploration of the structural design space*. Thesis, Massachusetts Institute of Technology, 2014. URL `https://dspace.mit.edu/handle/1721.1/91293`.

J. Nielsen. Why You Only Need to Test with 5 Users, Mar. 2000. URL `https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/`.

D. Niño Romero. StructuralComponents7.0: A study of rigid frames behaviour and a computational tool prototype for the early design of mid-rise buildings. Master's thesis, Delft University of Technology, 2019. URL `https://repository.tudelft.nl/islandora/object/uuid%3A2fa3056b-1d8f-40f6-8298-ccbd67cdec05`.

E. Piermarini and A. Declercq. Introduction to the BHoM, Jan. 2019. URL `https://www.youtube.com/watch?v=FOclwkfdm9U`.

D. Piker. Kangaroo Physics, Oct. 2017. URL `https://www.food4rhino.com/app/kangaroo-physics`.

C. Preisinger. Karamba3D, 2020. URL `https://www.karamba3d.com/`.

M. Rippmann, L. Lachauer, and P. Block. Interactive Vault Design. *International Journal of Space Structures*, 27(4):219–230, Dec. 2012. ISSN 0266-3511, 2059-8033. doi: 10.1260/0266-3511.27.4.219. URL `http://journals.sagepub.com/doi/10.1260/0266-3511.27.4.219`.

A. Rolvink. StructuralComponents2.0 - a parametric and associative toolbox for conceptual design of tall building structures. Master's thesis, Delft University of Technology, Delft, 2010.

G. I. N. Rozvany, M. P. Bendsoe, and U. Kirsch. Layout optimization of structures. *Appl Mech Rev*, 48(2):41–119, 1995.

M. S. M. S. Shearer. Analyzing and creating forms : rapid generation of graphic statics solutions through RhinoScript. Master's thesis, Massachusetts Institute of Technology, 2010. URL `https://dspace.mit.edu/handle/1721.1/60779`.

R. D. J. M. Steenbergen. *Super elements in high-rise buildings under stochastic wind load.* Doctoral Thesis, Delft University of Technology, 2007. URL `https://repository.tudelft.nl/islandora/object/uuid%3Abc62e51b-5ed4-4101-968b-3d2be3dbca92`. Publisher: Eburon.

S. Stevin. *De Beghinselen des Waterwichts.* Raphelengius, Franciscus, 1586.

L. L. Stromberg, A. Beghini, W. F. Baker, and G. H. Paulino. Topology optimization for brace frames: combining continuum and beam/column elements. *Eng Struct*, 37:106–124, 2012.

H. Tamai, K. Takahashi, B. Izumi, M. Shigematsu, and K. Ando. Catalogue of equilibrium forms of structures. In *Proceedings of IASS Annual Symposia*, pages 2638–2645, Barcelona, Spain, 2019. International Centre for Numerical Methods in Engineering. ISBN 978-84-12-11010-4.

K. C. Terwel. *Structural safety: Study into critical factors in the design and construction process.* Doctoral Thesis, Delft University of Technology, 2014. URL `https://repository.tudelft.nl/islandora/object/uuid%3Ac8a11e06-6ea8-49e7-b18c-273ac4c8a137`.

E. R. Tufte. *Visual and statistical thinking: Displays of evidence for making decisions*, volume 12. Graphics Press, Cheshire, CT, 1997. URL `http://courses.ischool.utexas.edu/Cherian_Antony/INF327E/restrict/visual&statistical_thinking.pdf`.

B. M. Van de Weerd. StructuralComponents: A client-server software architecture for FEM-based structural design exploration. Master's thesis, Delft University of Technology, 2013. URL `https://repository.tudelft.nl/islandora/object/uuid%3A6d95dd31-e1a7-41c9-abc9-49a8fcb80926`.

T. Van Mele, P. Block, C. Ernst, and L. Ballo. eQUILIBRIUM, 2009. URL `http://block.arch.ethz.ch/equilibrium/`.

T. Van Mele, A. Liew, T. Méndez Echenagucia, and M. Rippmann. COMPAS: A framework for computational research in architecture and structures, 2017. URL `https://compas-dev.github.io/`.

P. Varignon. *Nouvelle mécanique ou statique: ouverage posthume.* C. Jombert, Paris, 1725.

K. Verbeeck, L. Loos, L. Muller, and L. De Leat. How to re-open the black box in the structural design of complex geometries. In *Structures and architecture beyond their limits*, pages 316–323, Guimaraes, Portugal,

2016. Taylor and Francis Group. URL `http://www.crcnetbase.com/doi/book/10.1201/b20891`.

R. Vierlinger. Octopus, 2018. URL `https://www.food4rhino.com/app/octopus`.

R. Wiltjer, P. Peter, R. Stark, and P. Korthagen. *De kunst van het construeren*. Bouwen met Staal, 2018. ISBN 978-90-72830-97-5.

Appendices

Appendix A

Design tools analysis

A.1 Real-time numerical analysis

CSI Model Alive Computers and Structures Inc. (CSI) has found a way to implement the functionality of real-time feedback on structural designs in their software programs SAP2000 [Computers and Structures, 2020] and ETABS [Computers and Structures, 2019]. Both these products are structural analysis and design tools, with the latter taking a broader approach by including drafting functionality and addressing more general building aspects.The model alive function allows for instant structural feedback when changes in geometry, member properties or load conditions are applied. This can make the structural behavior insightful and can definitely fasten the design process, but its usability is limited because it can only be applied on small to medium-sized structures [Hohrath, 2018].

Arcade The freely available computer program Arcade [Martini, 2009] takes an entirely different approach on the simulation and visualization of structures, using computation methods and interaction models that stem from computer games. Feedback is provided instantly and the model can be interacted with while the analysis is in progress, merging effectively the synthesis, analysis and interpretation phase. The computation method, which is commonly referred to as *physics engine* or *particle system*, allows for greater visual realism and makes it possible to execute a more realistic non-linear analysis. The main disadvantages of this program are that it is limited to 2D problems, and that it is computationally heavy. This makes that only small problems can be analyzed. The tool is developed for educational purposes, for which it seems valuable, but is still far from a commercial application due to the given limitations and lack of recent developments [Martini, 2006].

A.2 Integrated numerical analysis

Geometry Gym Geometry Gym [Mirtschin, 2020] can be seen as a series of tools that deliver interoperability solutions. It mainly functions as a plugin in programs like Rhino, Grasshopper, Revit, Tekla and structural analysis software. It for example allows the transfer of a parametric geometric model in Grasshopper to structural software like Oasys GSA, even including the option to completely manage the analysis and result interpretation process in the Rhino-Grasshopper environment. This process of data transfer is useful but rather time intensive to set up and not very flexible with design changes. Therefore it does not match with conceptual design and is in this form only applicable at later design stages.

BHoM The BHoM or Building and Habitats object Model can best be described as a common language. Its objectives are somewhat similar to BIM (Building Information Modelling) but aims to define all the information in a shared common language, which is open source and not proprietary. This systems allows for standardized and simpler links between different architecture and engineering software, and therefore in a way achieves what Geometry Gym does without having to define a new plugin for every combination of software programs [Piermarini and Declercq, 2019]. Although the BHoM could ensure a better, more controlled and freer version of the current workflow, it is still embedded in a sequential design process, making it on itself not a solution for the current drawbacks of computational structural design tools.

Karamba3D Embedded as a plugin in the PAD environment of Grasshopper, Karmaba3D [Preisinger, 2020] is a parametric tool that can analyze the performance of a structure in real-time. It uses regular FEM-based computation, and allows for analysis of many kinds of structures. The fact that the tool is developed specifically for the Grasshopper environment has both benefits and drawbacks. Grasshopper is a powerful, fast-growing and relatively easy-to-use platform, and the existence of many other plugins like Kangaroo and Galapagos make it relatively easy to respectively incorporate form-finding and evolutionary optimization, extending the functionality. However, the use of Karmaba is inherently limited to this specific environment, making it unreachable for designers working in different (similar) environments or new emerging platforms. Also, Karamba remains a finite element solver and therefore still acts like a black-box in many design cases, although real-time feedback and proper visualizations can help in this respect.

Robot Robot Structural Analysis [Autodesk, 2020b] is a structural analysis software program by Autodesk. It can be coupled with Revit, which since 2018 does not anymore contain structural analysis capabilities in itself. Like other analysis tools it is based on FEM and is just like Revit mainly applicable on later design stages, and not really on conceptual design.

compas_fea COMPAS [Van Mele et al., 2017] is an open source Python-based framework for computational research in architecture and structures. Different packages exist that are built on the COMPAS core, of which one is compas_fea. This tool "aims to aid the user [...] in creating and analyzing a suitable finite element model for their problem." The unique quality of the COMPAS framework is that it is not limited to a certain software environment, making that its functionalities can in theory be applied to any PAD, CAD or even FEM software.

A.3 Form-finding

CADenary Using particle-spring systems, CADenary [Kilian and Ochsendorf, 2005] is a form-finding tool for structures that are tension-only or compression-only. This method has the advantage that the user can real-time change forces and geometry while the analysis is running, making it an interactive tool. Disadvantages are that the procedures are computationally expensive and that the method can't make a distinction between tension and compression element, meaning for example that a cable element can be computed to bear a compression force if no measures are taken to prevent this.

RhinoVAULT RhinoVAULT [Rippmann et al., 2012] performs structural form-finding using the Thrust Network Analysis (TNA) to explore compression-only structures. This method has the same fundamentals as Graphic Statics, and uses the reciprocal relation between force and form diagrams. The tool is developed as a functional Grasshopper plugin and effectively extends 2D Graphic Statics to '2.5D' using "planar projections of a discretized shell geometry to construct interactive form and force diagrams," quoting Lee [2018].

Dynamic relaxation Where designers in the past would use hanging chains or soap films to find optimal shapes, does dynamic relaxation provide a numerical method to simulate these models, which can be used to find optimal geometries for cable, membrane and shell structures. The process is iterative and simulates the motion of a structure under applied loading over time to an equilibrium solution. Princeton's form finding lab has developed a web-based tool for form-finding for shell structures using this method [Adriaenssens, 2020]. Additionally, dynamic relaxation is incorporated in some commercial structural analysis programs.

Appendix B

Overview StructuralComponents

B.1 Structural design tools for tall buildings

StructuralComponents 1 The first implementation of the SDT concepts was developed by Breider [2008] and concerns a tool for the structural design of tall buildings in the PAD environment of GenerativeComponent [Aish, 2005]. It is based on a Lego-block plus dashboard approach, and performs structural analysis using a variant of the super element method [Steenbergen, 2007]. The thesis is limited to 2D high-rise buildings consisting of cores, columns and outriggers, whose configurations can be explored in a flexible way, and can be evaluated interactively [Breider, 2008].

StructuralComponents 2 Rolvink [2010] continues in the same field of tall buildings but focuses on several issues to improve the first version of StructuralComponents. A big emphasis of the thesis is put on the development a new framework architecture for the toolbox, including a more modular setup, independency of (PAD) software and an improved workflow. Another focus point is the addition of new features to the toolbox, allowing the stability systems to include a wider variety of structural elements like perforated cores and shear wall structures. Other improvements are the development of a specific interface for Grasshopper, and more load modelling capabilities [Rolvink, 2010].

B.2 Focus on synthesis and a client-server software architecture

StructuralComponents 3 This rather versatile but thorough thesis by Van de Weerd [2013] contains three major objectives. Firstly its goal is to bring StructuralComponents more in harmony with the design cycle by an increased emphasis on the synthesis phase. This goal can also be explained as introducing the functionality of guidance to a thus far feedback-based tool, and is achieved by a system of abstraction that combines automatic generation with interactive visualization and exploration. Another main goal was to remove software dependencies and reimplement the tool with the use of a client-server software architecture, providing full data structure openness, transparency, flexibility and scalability. Lastly, a FEM analysis engine was developed and implemented in the tool [Van de Weerd, 2013].

B.3 Interactive distributed optimization for multidisciplinary design

Although formally not a StructuralComponents project, the research by Jansen et al. [2014] can be regarded as under its umbrella. A system is described that has the main aim to provide "designers and engineers with an intuitive tool to define, evaluate and optimize the performance of large multidisciplinary model," which is accomplished with the development of 2 major components, namely a multidisciplinary optimization framework and a distributed cloud-based analysis framework. The optimization framework uses an interactive optimization search strategy, giving insight into the performance of the problem and the strategy. The analysis framework provides a flexible infrastructure that allows for quick evaluation of design variants [Jansen et al., 2014].

B.4 Conceptual building models with design justification

StructuralComponents 4 Bovenberg [2015] steps away from the specialized high-rise toolbox and approaches the concepts of SDT in a completely different way. The objective is to model structural design concepts in a more complete way, and make the design story of a structure explicit by modelling design justifications, which can consist of models, simplifications, reasoning, alternatives and scenarios. The proposed tool consists of a conceptual building model, a user interface and a computational (parametric) engine. Seven high level characteristics of (structural) design have been defined that the tool must support, resulting in six central components of the conceptual building model. Some identified benefits of the tool are an improvement in the communication of reasoning, and a more effective use of optimization and analysis by the concepts of scenarios and parallel alternatives [Bovenberg, 2015].

B.5 Design tools for mid-rise buildings

StructuralComponents 5 A new implementation of StructuralComponents by Hohrath [2018] applies the super element method to develop a conceptual structural design tool for concrete mid-rise buildings. The structural model essentially consists of combinable, stackable and parametrically adaptable building blocks, that are for this project composed only of shear walls and cores. The super element method allows faster analysis than conventional FEM and provides near real-time validation [Hohrath, 2018].

StructuralComponents 6 Dierker Viik [2019] takes a different approach for concrete mid-rise buildings, allowing more flexible configurations of structural elements in the horizontal plane. This also results in the necessity of a different calculation method, which is based on a basic system of three differential equations. With the horizontal flexibility that is gained, the vertical flexibility to stack building blocks is lost, meaning that only 2D structures can be evaluated [Dierker Viik, 2019].

StructuralComponents 7 The most recent implementation of StructuralComponents by Niño Romero [2019] studies the rigid frame behavior in mid-rise buildings, and delivers a prototype early design tool for such structures. Different models of rigid frames under lateral loads are discussed, including the shear beam and a timoshenko beam based method. The tool allows for structural variations over the height, and adopts the principles of parametric adaptability, instant visualization of results and stacking connections between components. The tool also includes an indication of the applicability range of different methods, which can predict the accuracy of any specific analysis case [Niño Romero, 2019].

Appendix C

GSDesign manual

The following pages include the manual as it was provided for all participants of the user experiment as a Google Docs document. In this thesis report all links that were present in the manual are disabled.

GSDesign manual

12/12/2020

This document is a manual for the use of GSDesign, a **2D graphic statics based design tool** developed as part of the TU Delft Building Engineering masters' thesis 'StructuralComponents 8' by Floris Bruinsma.

GSDesign is a design tool where you have to provide your own initial structural sketch as a starting point. It enforces structural engineers to think by themselves to come up with a self-defined, logical, topology. Only after this initial definition can the tool come into play to explore the design and optimize it, creating a workflow that combines the engineers gut-feeling with the power of computation. Although optimization is involved in this process, it must be clear that the developed tool does not perform a typical topology optimization.

If you are taking part in the user experiment you are kindly requested to follow all three sections. If you are just interested in trying out GSDesign, only sections 1 and 2 are relevant. Participants of the user experiment are expected to be comfortable and handy in the use of Grasshopper, since the developed tool is based in this environment. Furthermore, during the tutorial and the experiment use will be made of Galapagos; an evolutionary optimization component embedded in Grasshopper. Galapagos is very user-friendly and works by connecting the Genome input to sliders whose value influence the design, and the Fitness input to a fitness value, as illustrated in the tutorial file of section 2.

Content

1. Installation (5 minutes)

First of all, please download the GSDesign tool, tutorial files, and the accompanying files for the user experiment via the following link. After opening the link, right-click the GSDesign.zip file and choose '*Download*'.

GSDesign-zip

The GSDesign tool consists of 2 parts: the main functionality is developed into a grasshopper plugin (**GSDesign-0.1.gha**), while some extra visualization logic is defined as Grasshopper clusters which are stored in a separate file (**GSDesignClusters.gh**). The cluster file can be opened in Grasshopper and used directly, while the plugin needs to be installed. Please follow these instructions:

1. Open Rhino 6 (Rhino 7 should also work, but has not been tested yet)
2. Open Grasshopper by typing `Grasshopper` into the Rhino command line, or by clicking the icon.
3. In Grasshopper, go to *File > Special Folders > Components Folder*
4. Copy the **GSDesign-0.1.gha** file into this folder. For windows: make sure that the file is unblocked by right clicking the file, selecting *properties*, ticking '*unblock*', and selecting *ok*.
5. Restart Rhino
6. GSDesign should now be added as an extra plugin, and should show up like displayed in figure 1. If you receive an error message similar to the one in figure 2, you should update Rhino. Version 6.31.20315.17001 or higher is required.

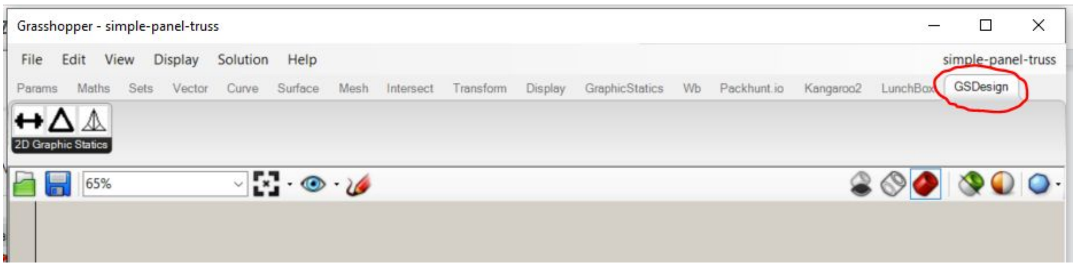
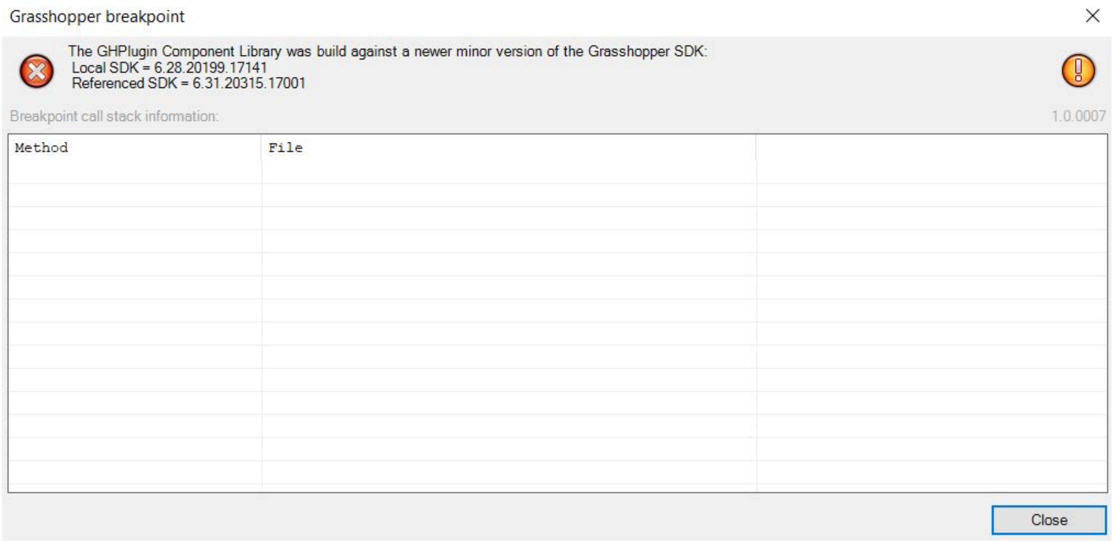


Figure 1: Grasshopper interface with GSDesign plugin properly installed

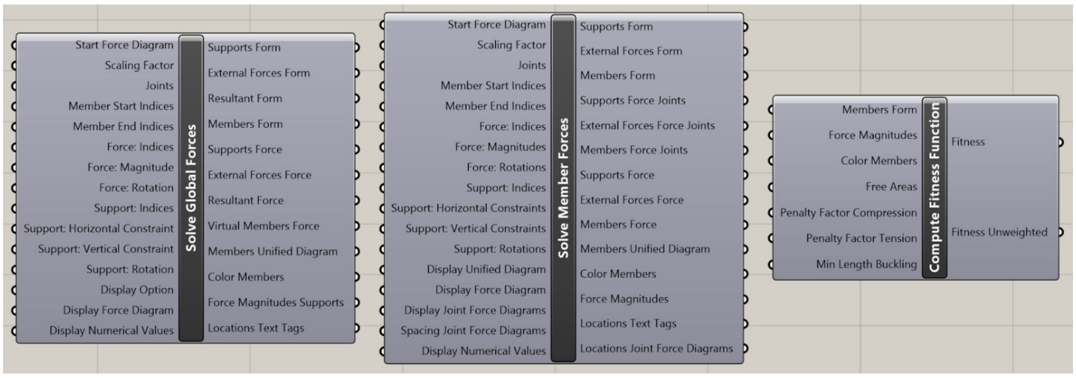


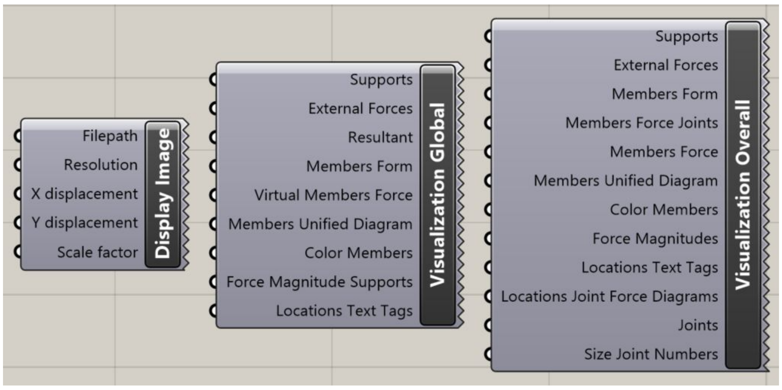
The GHPlugin Component Library was build against a newer minor version of the Grasshopper SDK:
Local SDK = 6.28.20199.17141
Referenced SDK = 6.31.20315.17001

Figure 2: Error message indicating that Rhino should be updated

2. Functionality (15 minutes)

As mentioned, GSDesign consists of components and Grasshopper clusters. The two images below show all the components (top), and clusters (bottom) available within the tool. As mentioned before, the clusters can be obtained by opening the **GSDesignClusters.gh** file.





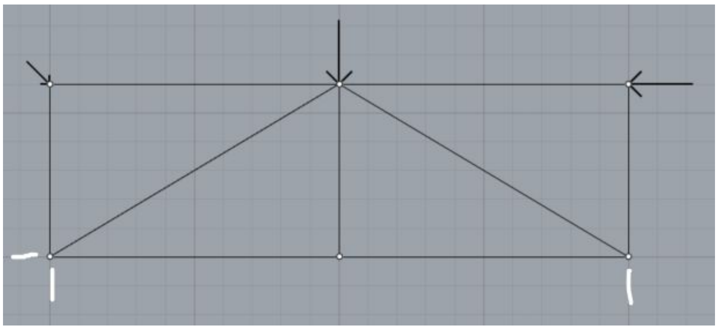
The functionality of each element is as follows:

- **Solve Global Forces** (component): Finds the resultant of all external forces, creates and solves a global form and force diagram, and thereby finds the support reactions.
- **Visualization Global** (cluster): visualizes the results of **Solve Global Forces**.
- **Solve Member Forces** (component): Incorporates the same functionality of **Solve Global Forces**, but then proceeds to solve all individual joints in the actual structure, finding all member forces. Output is focussed on the overall results and does not include the global diagram or the force resultant.
- **Visualization Overall** (cluster): Visualizes the results of the **Solve Member Forces**.
- **Compute Fitness Function** (component): Computes a fitness function, based on the total load path SUM(F_i * L_i), incorporating also other design input like areas that need to remain free, a penalty factor for either tension or compression elements, and a factor that can include buckling behaviour. The total load path calculated by SUM(F_i * L_i) is a

basic representation of the necessary structural material volume, assuming a constant stress distribution. SUM(F_i * L_i) multiplies the length of each member *i* by the force it carries and adds this to the total summation, resulting in one value that indicates the structural material use, therefore quantifying the efficiency of the design.

- **Display Image** (cluster): imports a jpg or png image and lets the user scale and move it to a preferred location on the canvas. Can be used to import an initial design sketch, which can then be manually traced in Rhino.

SimpleHoweTruss-tutorial.gh contains a tutorial file showcasing how these components can be used to obtain valuable design information. The design case that is used here is that of a simple Howe Truss, defined as in the image below, with black arrows depicting external forces, and white lines depicting the yet unknown support reactions. The tutorial file must be opened in combination with the Rhino file **SimpleHoweTruss-tutorial.3dm**



The Grasshopper tutorial file contains multiple groups showing the functionality of GSDesign step-by-step. The content of these groups will be explained here:

Solve Global Forces

This group shows how the **Solve Global Forces** component can be used in combination with the **Visualize Global** cluster. The reader is encouraged to play around, change certain inputs and observe how the design and visualization changes. To analyze which index a certain point has in a list (which must be known in order to define the members), the Point List component can be used. The grey line and arrow in the visualization represent the resultant force of the three external forces.

Solve Member Forces

This group shows how the **Solve Member Forces** component can be used in combination with the **Visualize Overall** cluster. As can be seen the inputs are mostly similar to those of the

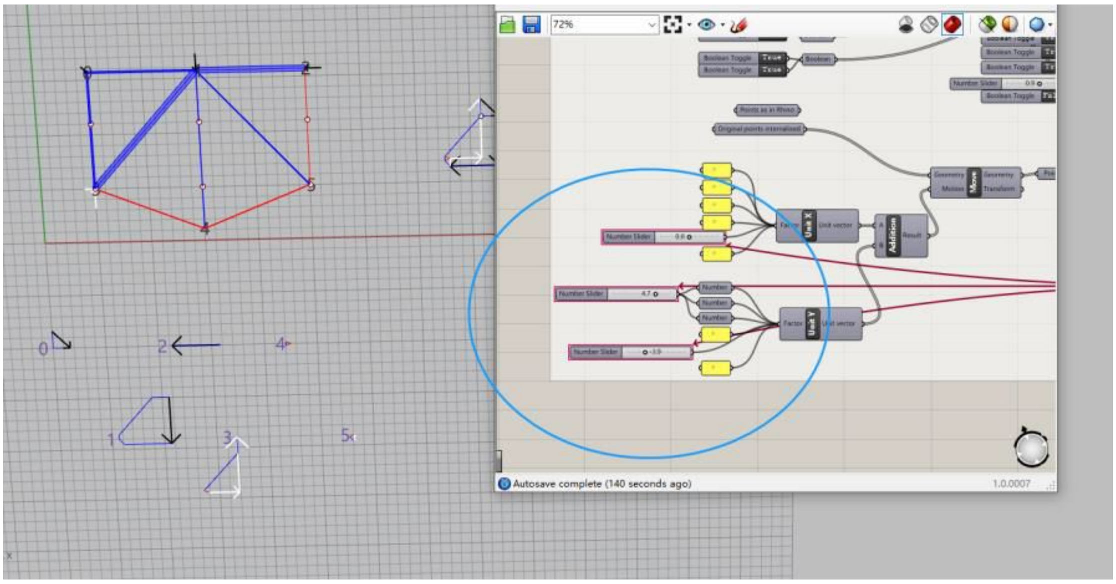
Solve Global Forces component, and are therefore linked from the original input definition in order to not repeat ourselves.

In order to enable the display of the Visualisation Overall cluster in the Rhino viewer, right click the cluster and select *preview*. With the same method the Visualization Global display can be hidden.

Again the reader is encouraged to play around with all the inputs, in order to understand the functionality of this component + cluster.

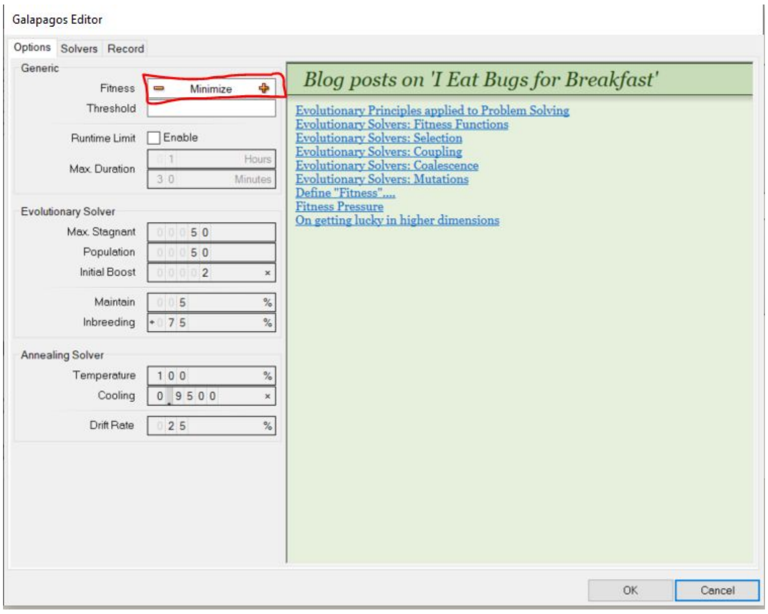
Optimization 1

This group shows a simple optimization setup where three parameters are defined that manipulate the locations of the joints that make up the structure. These parameters are shown in the figure below. The top slider manipulates the x-value of joint 4, the middle slider manipulates the y-value of joints 0, 1 and 2, and the bottom slider manipulates the y-value of joint 4. Apart from this, the locations of the joints are fixed.



The **Compute Fitness Function** is introduced here and uses three inputs of the Solve Member Forces component to compute the ''fitness" of the design. No additional inputs have been defined, so the component only computes the pure load path value.

Galapagos is used to optimize the structure based on the fitness value, using the defined parameters. By double-clicking the Galapagos component, one can access the Galapagos optimization interface, and start an optimization sequence by going to the *solvers* tab and clicking *Start Solver*. For both this optimization and the next one, the advised Galapagos settings are displayed in the figure below. Most important is that the Fitness value must be minimized instead of maximized.

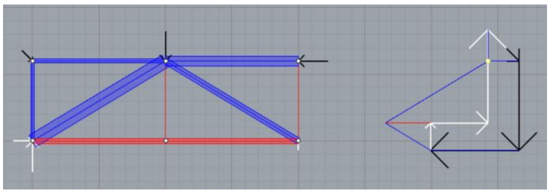


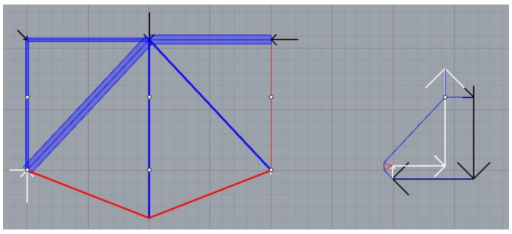
Generic
Fitness: Minimize
Threshold:
Runtime Limit: ☐ Enable
Max. Duration: 1 Hours, 30 Minutes

Evolutionary Solver
Max. Stagnant: 50
Population: 50
Initial Boost: 2 ×
Maintain: 5 %
Inbreeding: 75 %

Annealing Solver
Temperature: 100 %
Cooling: 0.9500 ×
Drift Rate: 25 %

The results should be as in the images below. With the start situation on the left, and the (for these specific parameters) optimized design on the right.



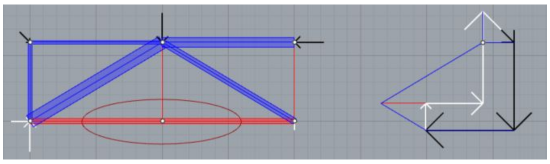


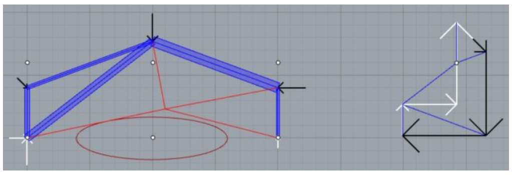
Optimization 2

For this optimization sequence, two extra things have been incorporated:

- A curve has been added to the Free Areas input of the Compute Fitness Function component, which contains an oval shape (right-click the curve component and select *preview*, if you don't see it), wherein no structural elements are allowed.
- The parameterization of the locations of certain points is now accompanied with a parameterization of certain members. Two number sliders can be set to either one or zero, which results in a different connectivity of the joints.

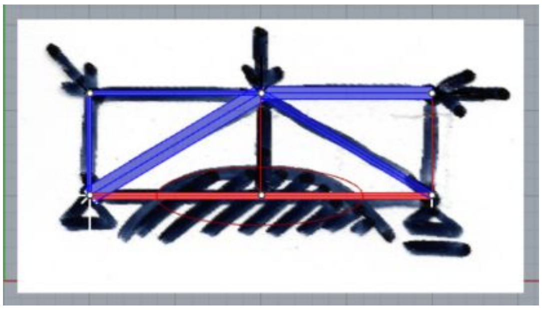
The reader is encouraged to try out these new parameters and check the optimization setup. When all is clear, an optimization can be carried out, which should (with this specific setup) converge towards a design similar to the image below on the right. It is worth noting that a heuristic optimization algorithm is used, which means that your outcome could slightly differ from the one shown here.





Display Image

The display image group shows how the **Display Image** cluster can be used to import a jpg or png image of a design sketch to the Rhino canvas, to then trace that design digitally. This would therefore normally be done at the beginning of a design process in GSDesign. To see the image, you must define the file path entry so that it matches the location of the **SimpleHoweTruss-sketch.jpg** on your computer, and be sure to set the preview of the cluster on. To define the file path, right-click the File Path component and choose '*select one existing file*', and select the **SimpleHoweTruss-sketch.jpg** file.



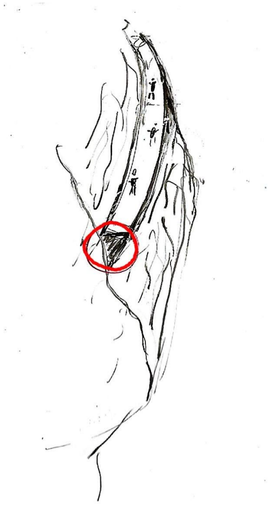
3. User Experiment (75 minutes)

First of all thank you very much for participating in this user experiment!!

The experiment concerns a design exercise, and consists of 2 parts. In the first part a ‘conventional’ design process is followed where the participants are asked to make a design manually, using their own expertise and insight. In the second part they attempt to construct an improved or new design in the GSDesign tool, using the manual design of the first step as a starting point.

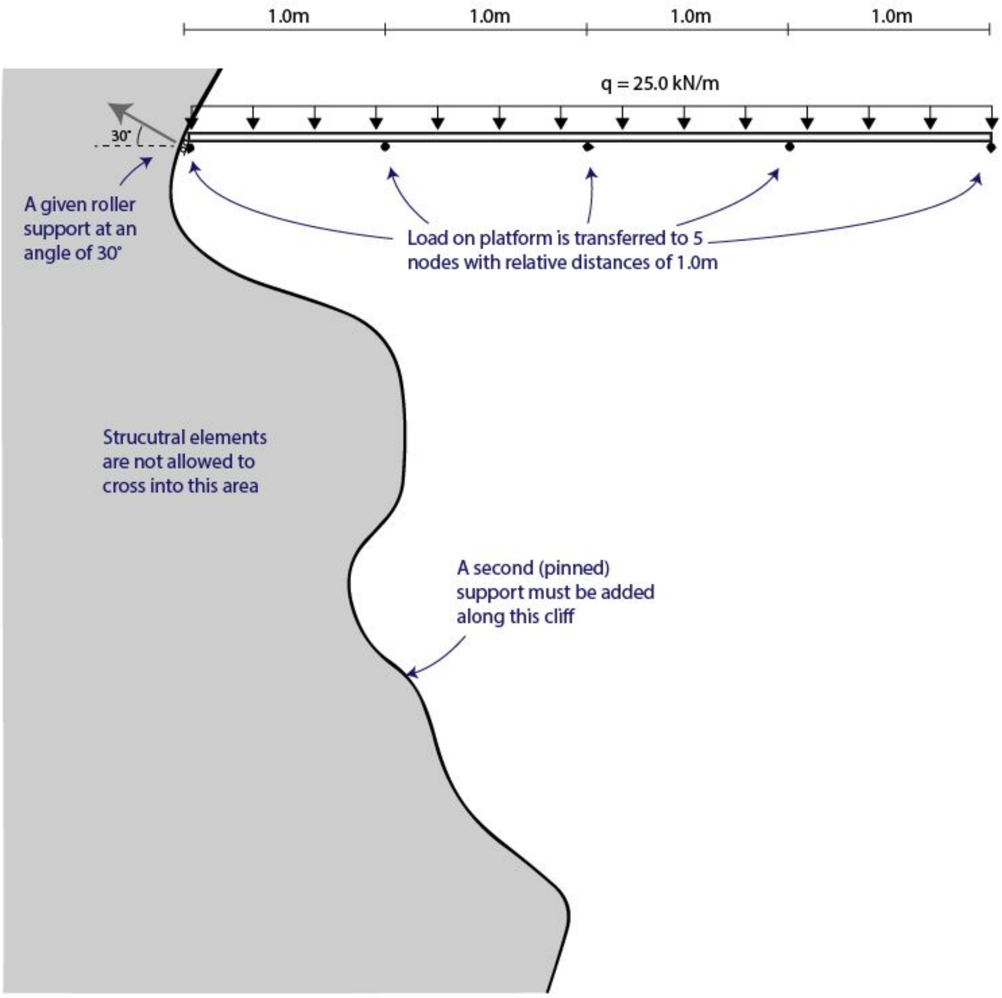
Setup of design case

The design case concerns a structure that supports a platform along a cliff.
The figures below show a picture of the construction of such a structure (left), and a sketch of the design situation in 3D (right).



A 2D structure must be created, which is assumed to be supported out of plane, however buckling of members between nodes must be taken into consideration.

The image on the next page shows the whole setup of the design problem. It concerns a 4m wide platform, bearing a distributed load. 5 nodes have been defined on which the load on the platform is transferred. The node on the left-hand side is also the first support, which is a roller at an angle of 30 degrees. The second (pin) support must be placed on the outline of the cliff, which is also given. An additional requirement is that no structural elements are allowed to go through the rock.



In order to allow proper comparison and to be able to digitalize the designs in the GSDesign tool, the following requirements apply:

- The design must be a truss-system, meaning that it consists only of **straight normal-force bearing elements**, connected by **pin joints**.
- The design must **statically determinate**, which for 2D trusses means that: **m + r - 2j = 0** (m = number of members, r = number of support reactions (3), j = number of joints)

Apart from the mentioned requirements, participants are free in their designs and can add as many nodes as they want. The main design objective of the experiment is:

To design a structure that transfers the loads to the supports as efficiently as possible.

Experiment instructions

Step 1: manual design

1. Print **UserExperiment-StartingPoint.pdf** on paper multiple times (5 minutes), or - if you prefer to make your sketch digitally - open the file in a suitable program.
2. Make one or multiple designs on the printed paper or digital environment. The following rules apply:
    a. You have **15 minutes** to complete this step, please time yourself!
    b. The use of any additional analog or digital tools besides (digital) pen and paper is allowed.
3. When time is up or when you feel that you are done, please document the final design(s) by scanning or taking a photo. Save the designs as jpg or png on your computer. (5 minutes)

Step 2: GSDesign

Now the goal is to take the designs during step 1 as a starting point to further explore possible solutions. The idea is to digitalize the design(s) of step 1 and optimize them using the previously described method. During this design process, you may be inspired to try out different designs in GSDesign, please do so if time permits.

1. Open **UserExperiment-CliffsideCantilever.3dm** and, after you started Grasshopper, **UserExperiment-CliffsideCantilever.gh**. These files contain a starting point for this specific case from where you can start designing. Please read the instructions in the file and make sure that you understand the workflow to define a new design, depicted by the numbered steps. (10 minutes)
2. Start designing! The following rules apply:
    a. You have **30 minutes** to complete this step, please time yourself!
    b. Please **save every meaningful newly created design** as a new (logically named) file. (also Rhino file if relevant)
3. When the time limit is up, please pick your final design and other relevant designs and save those Rhino and Grasshopper files in a handy spot on your computer.

Step 3: Submit results (5 minutes)

Take all the relevant files from steps 1 and 2, place them in one folder and send that folder via wetransfer to florisbruinsma@white-lioness.com.

Step 4: Questionnaire (5 minutes)

As a final step, please fill in this questionnaire. The questions mainly concern your experience in the experiment, and have the objective to get a view on the practical value and user friendliness of the tool.

Thank you!

Appendix D

Results user experiment

This appendix contains the complete results of the user experiment. The first section contains two images and a table per created design, showing the progression of the unified and force diagram from initial design to optimized design, as well as the accompanying fitness data. The second section contains an overview of the responses to the questionnaire, with the multiple-choice answers visualized in bar charts and the answers to the open questions provided in full.

D.1 Created structural designs

Participant 1

Design 1 of 1:

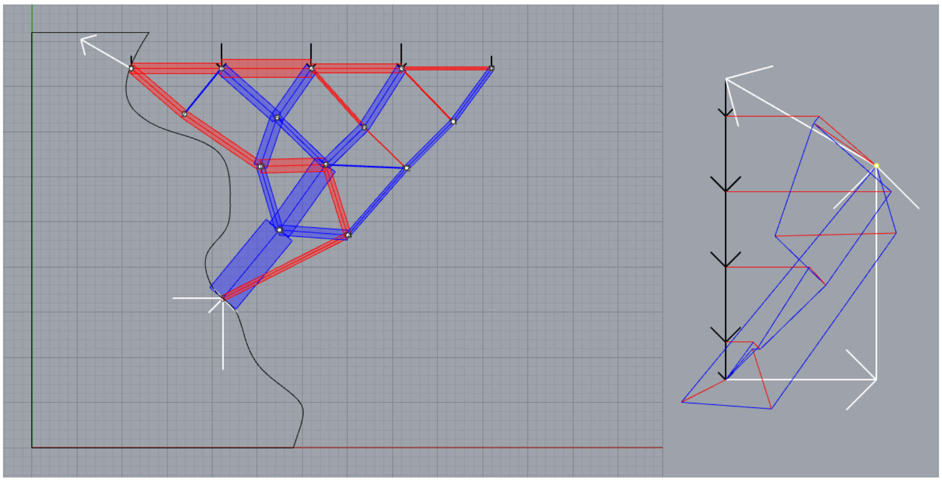


Figure D.1: The initial manual design

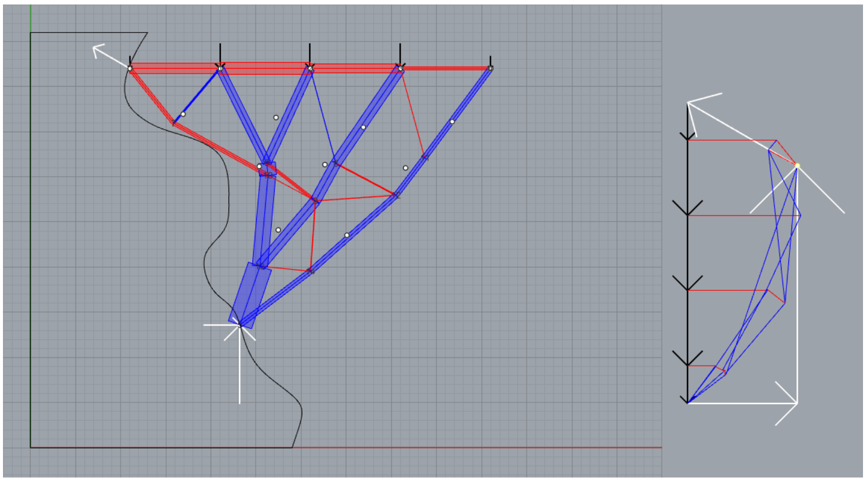


Figure D.2: The improved optimized design

Table D.1: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
6306	4500	71%

Participant 2

Design 1 of 1:

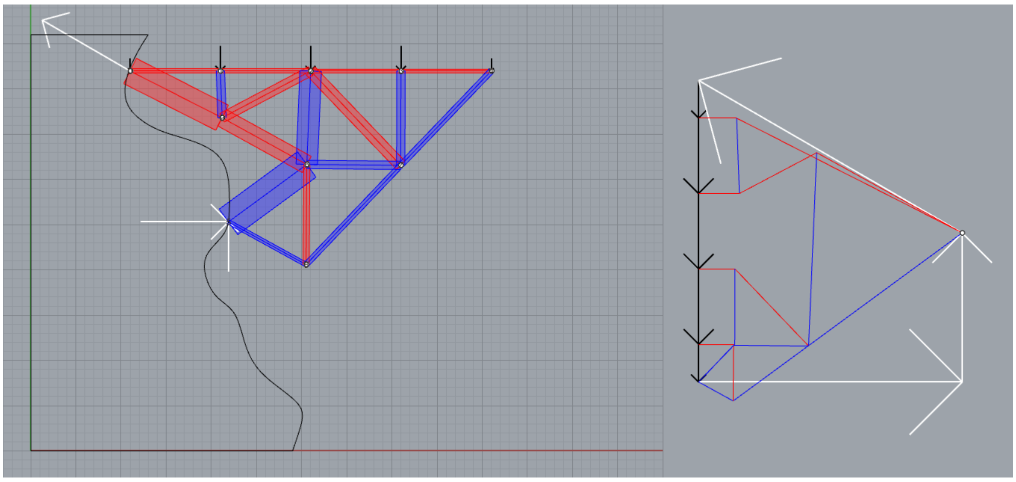


Figure D.3: The initial manual design

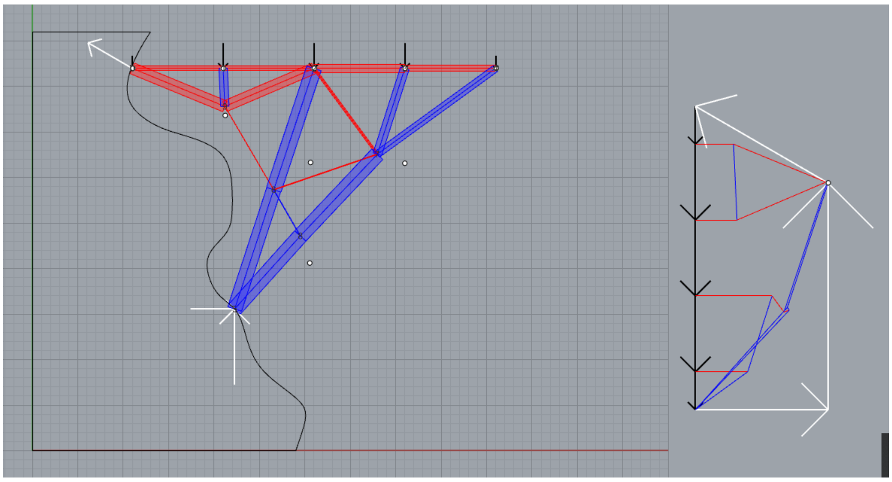


Figure D.4: The improved optimized design (first iteration)

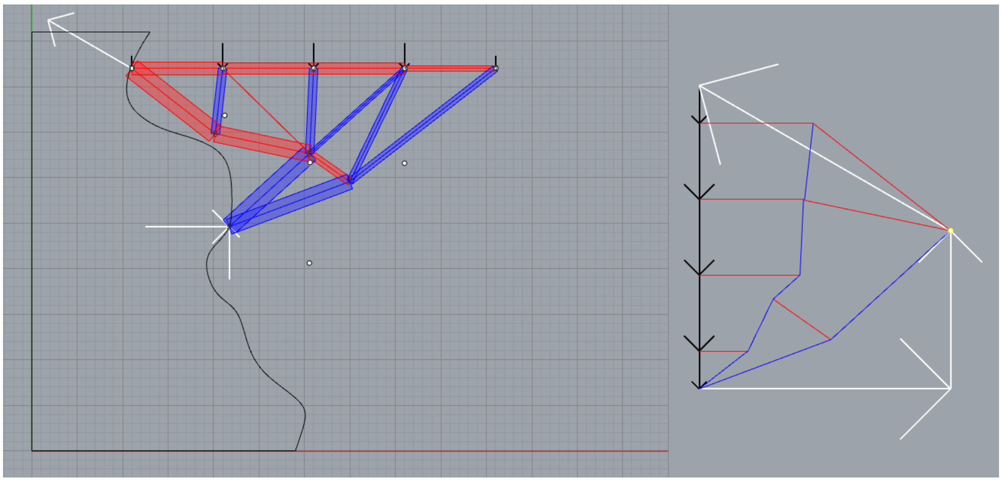


Figure D.5: The improved optimized design (second iteration, after removing a member)

Table D.2: Fitness values $[kNm]$ of the initial manual design and the improved optimized designs

Initial design	Optimized design (1)	Optimized design (2)	Percentage
6259	5024	6053	respectively 80% and 97%

Participant 3
Design 1 of 1:

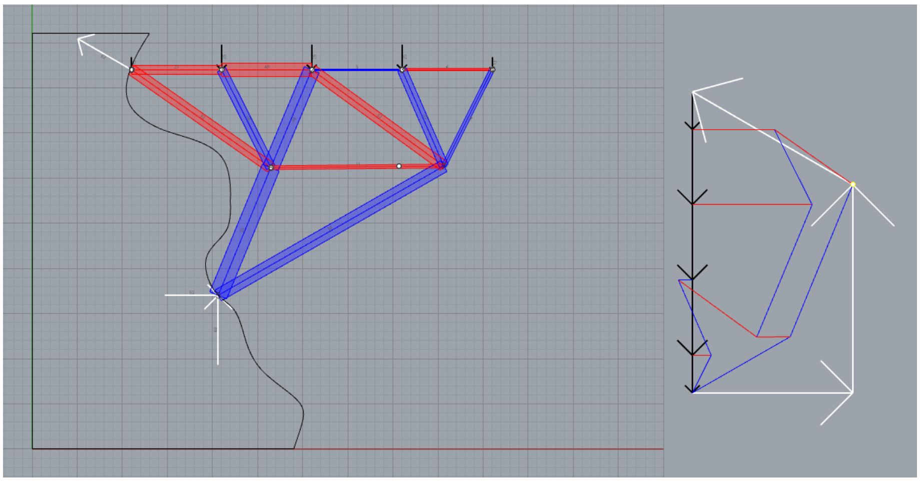


Figure D.6: The initial manual design

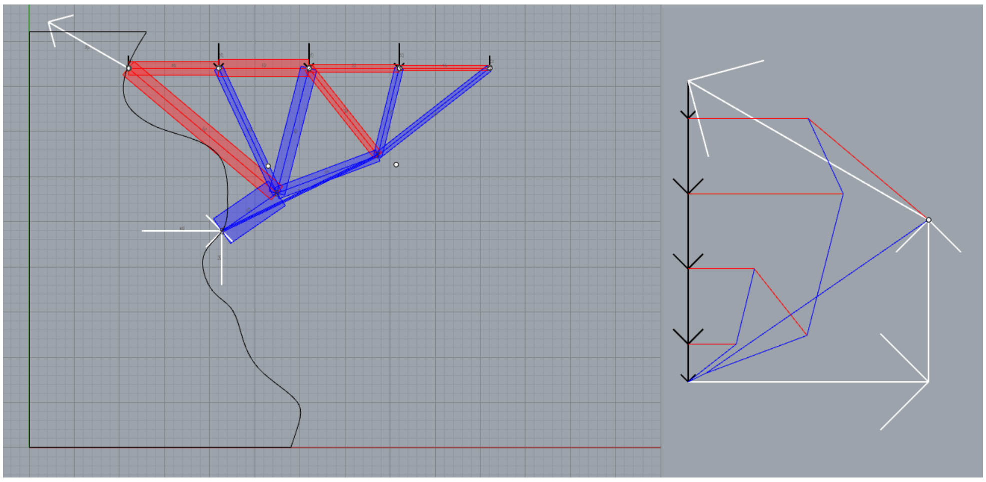


Figure D.7: The improved optimized design

Table D.3: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
10569	6450	61%

Participant 4

Design 1 of 3:

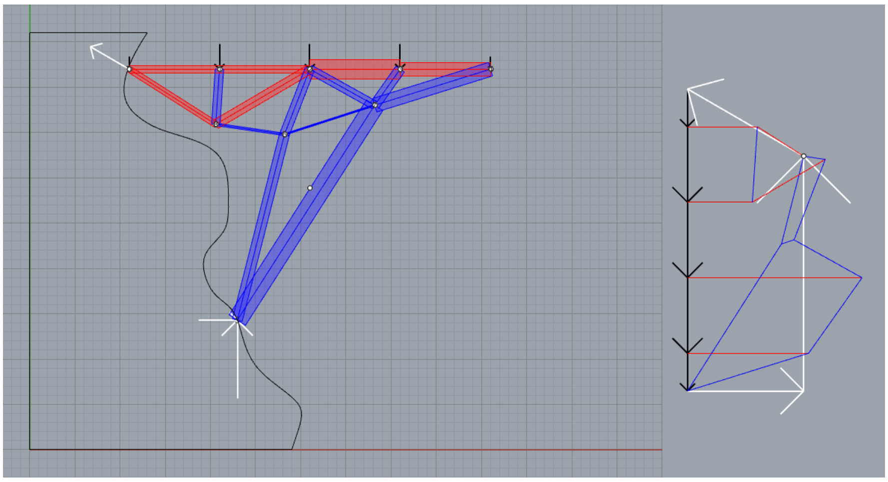


Figure D.8: The initial manual design

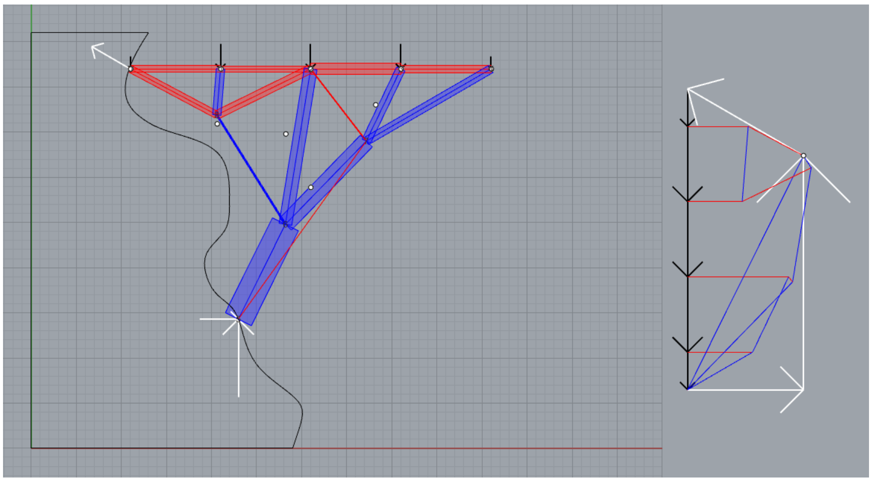


Figure D.9: The improved optimized design

Table D.4: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
13551	5603	41%

Design 2 of 3:

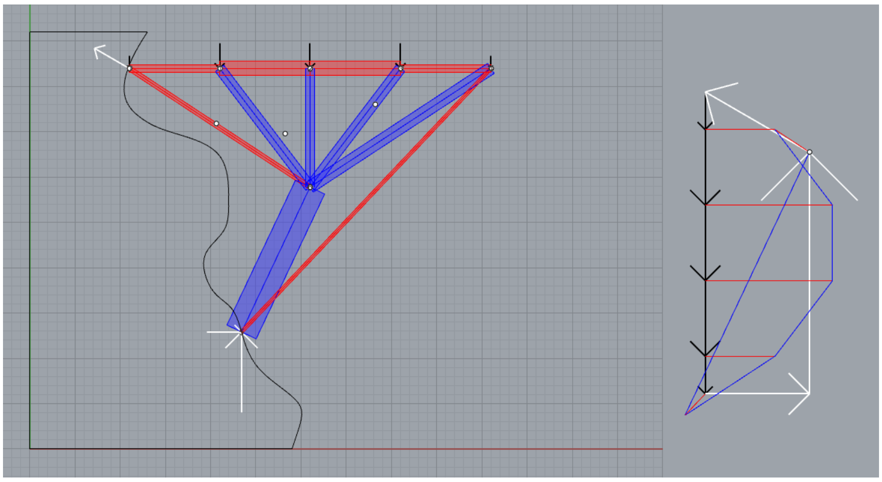


Figure D.10: The initial manual design

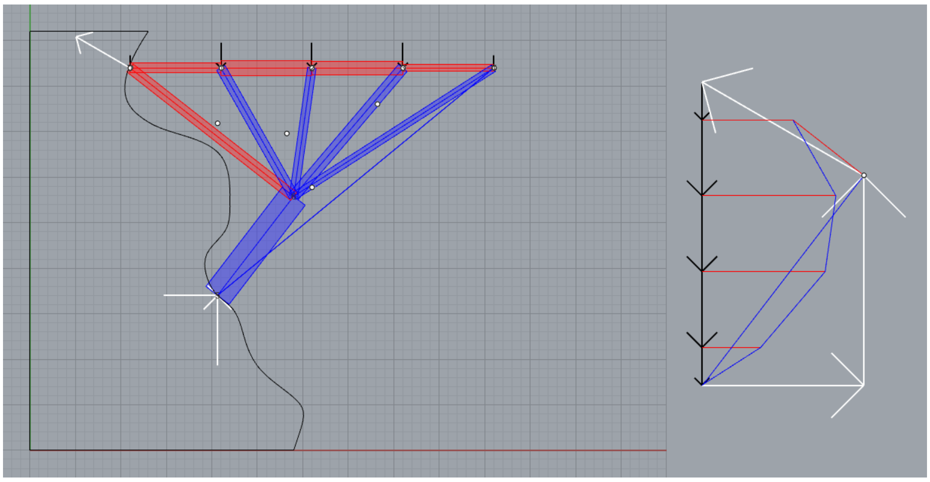


Figure D.11: The improved optimized design

Table D.5: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
10734	8852	82%

Design 3 of 3:

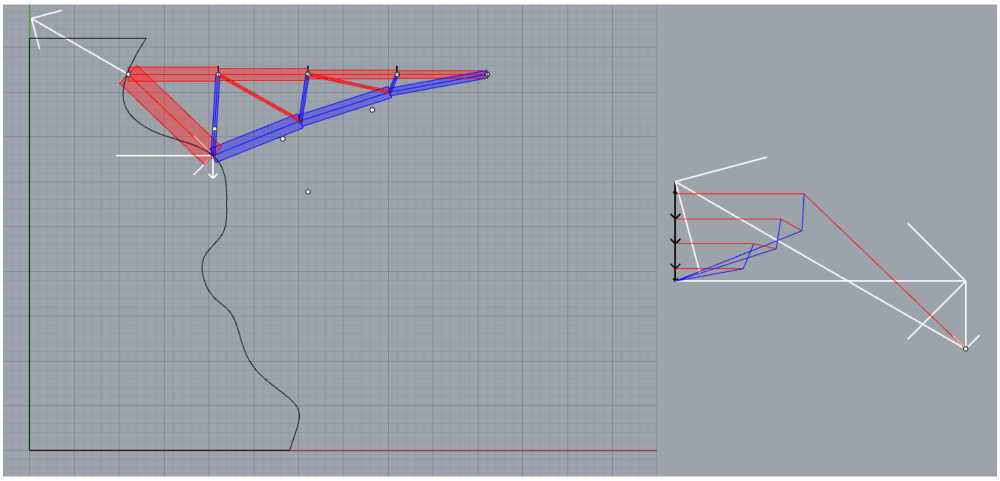


Figure D.12: The initial manual design (all forces scaled by $\frac{1}{3}$ to fit image on page)

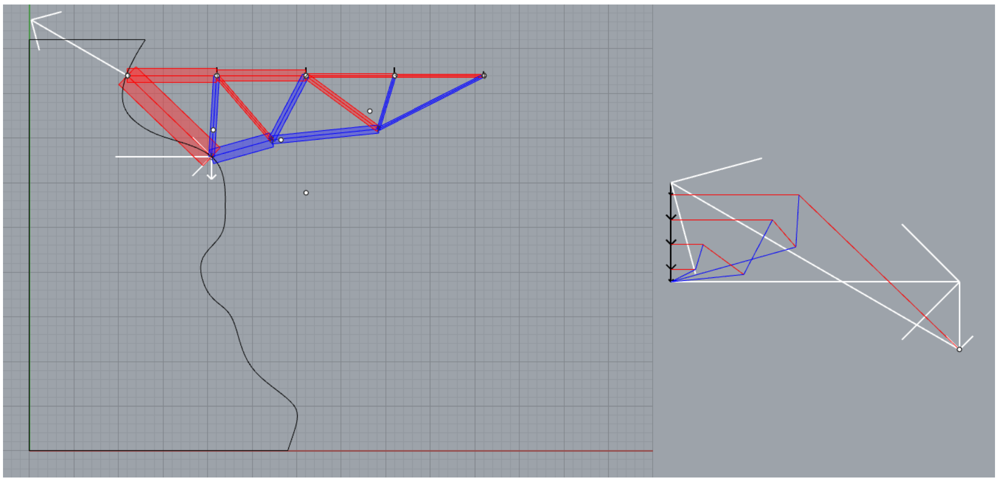


Figure D.13: The improved optimized design (all forces scaled by $\frac{1}{3}$ to fit image on page)

Table D.6: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
11104	10058	91%

Participant 5

Design 1 of 1:

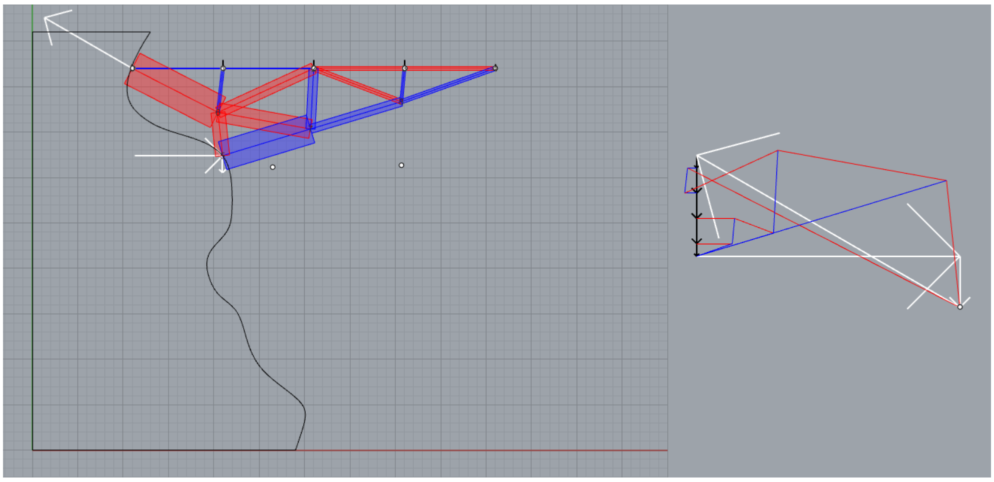


Figure D.14: The initial manual design (all forces scaled by $\frac{1}{3}$ to fit image on page)

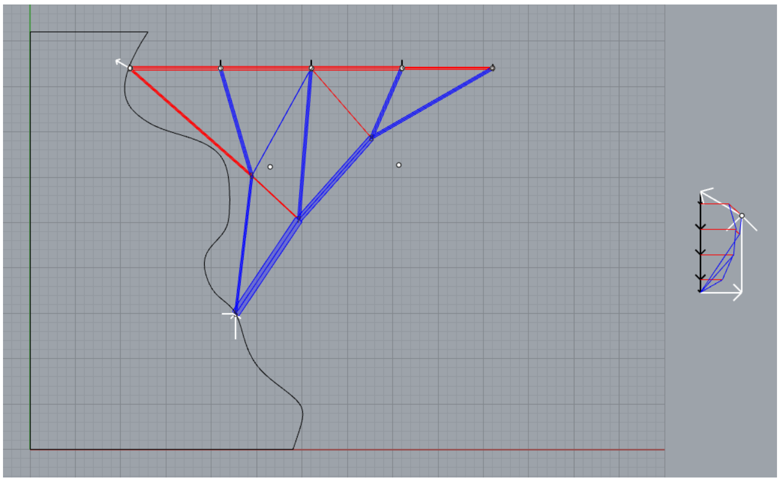


Figure D.15: The improved optimized design (all forces scaled by $\frac{1}{3}$ to fit image on page)

Table D.7: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
12756	5380	42%

Participant 6

Design 1 of 4:

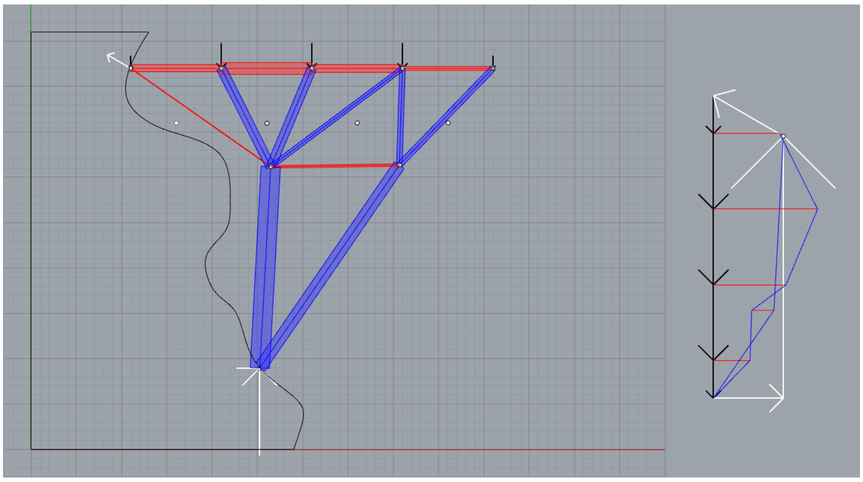


Figure D.16: The initial manual design

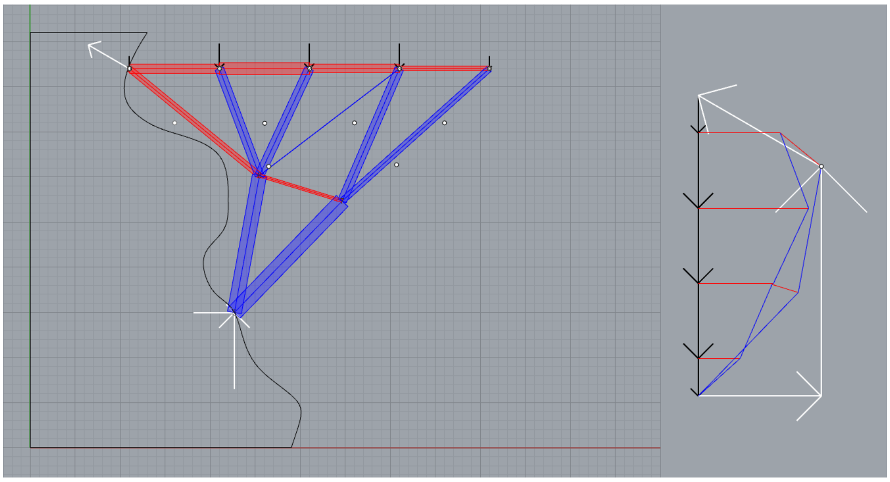


Figure D.17: The improved optimized design

Table D.8: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
11374	6759	59%

Design 2 of 4:

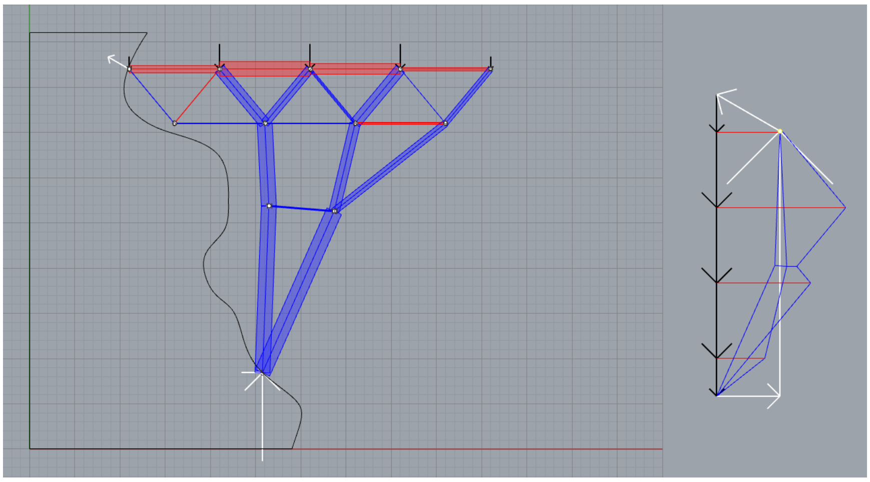


Figure D.18: The initial manual design

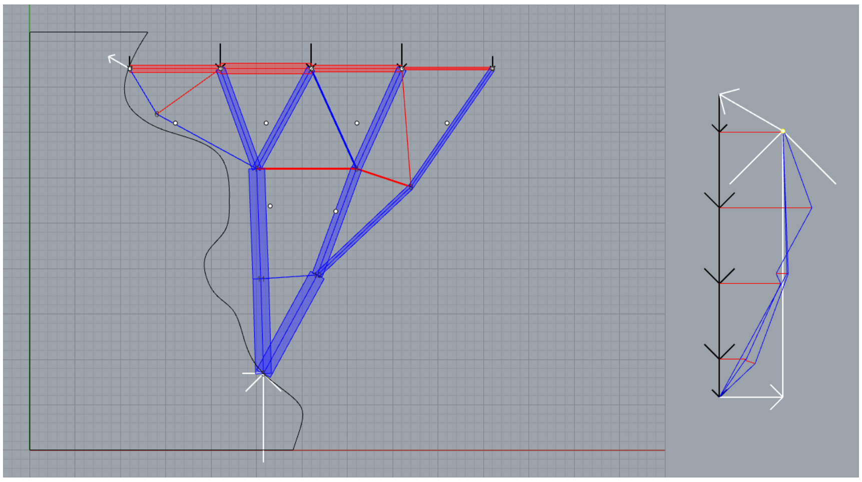


Figure D.19: The improved optimized design

Table D.9: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
7361	4608	63%

Design 3 of 4:

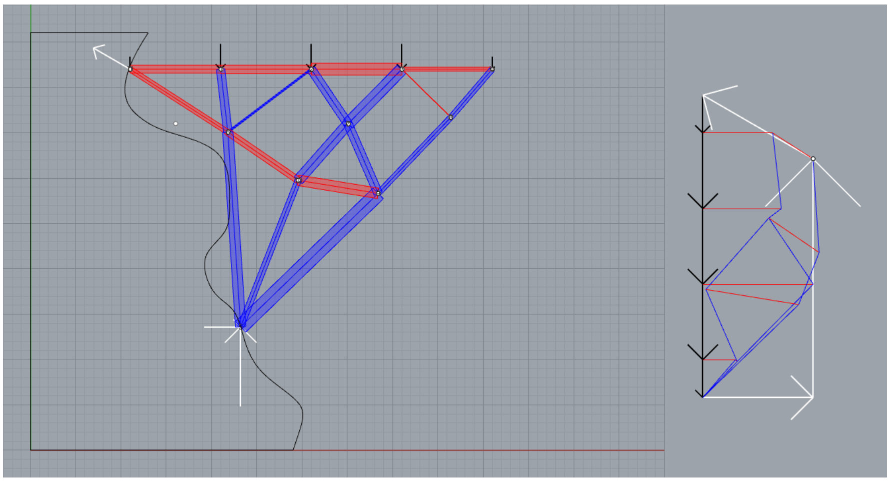


Figure D.20: The initial manual design

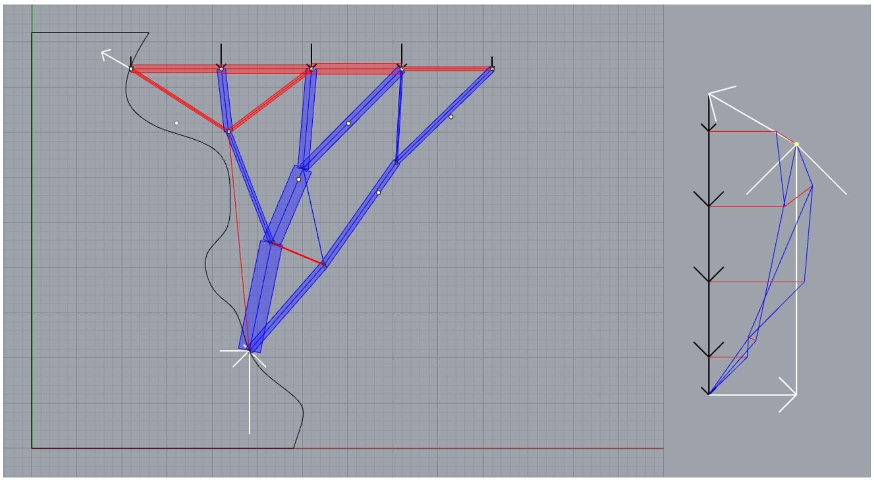


Figure D.21: The improved optimized design

Table D.10: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
8428	5005	59%

Design 4 of 4:

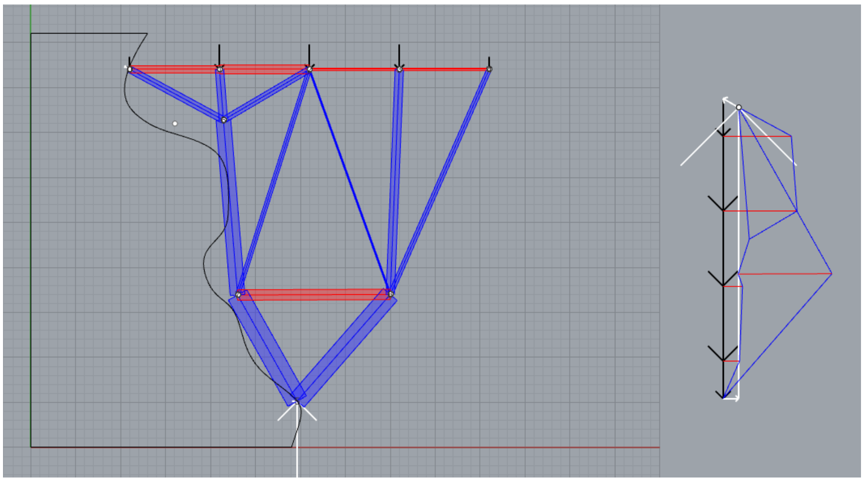


Figure D.22: The initial manual design

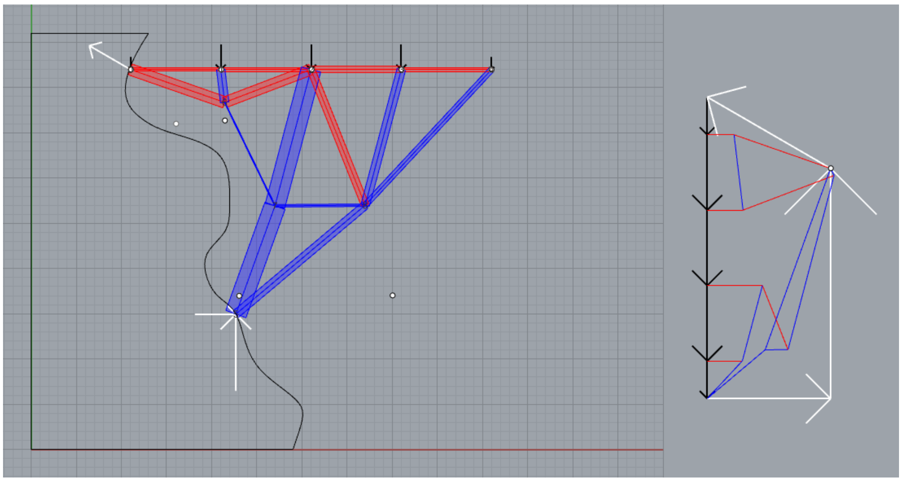


Figure D.23: The improved optimized design

Table D.11: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
12406	6697	54%

Participant 7

Design 1 of 1:

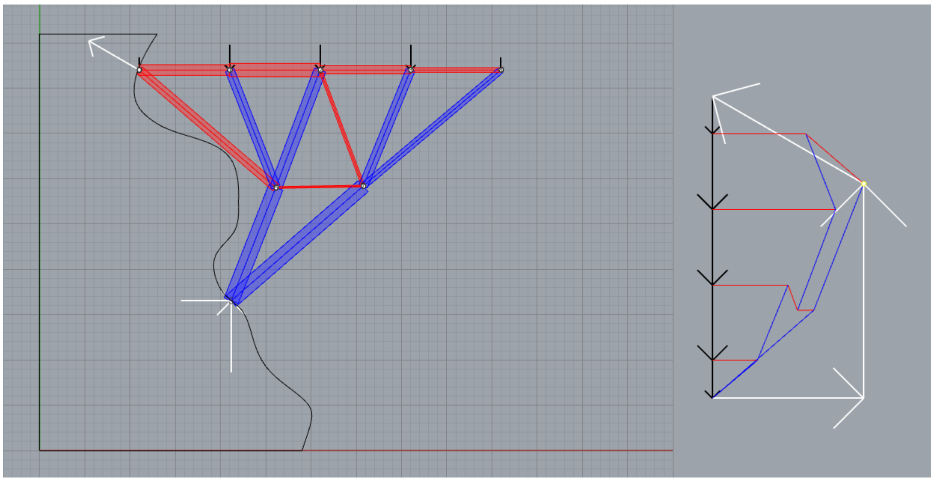


Figure D.24: The initial manual design

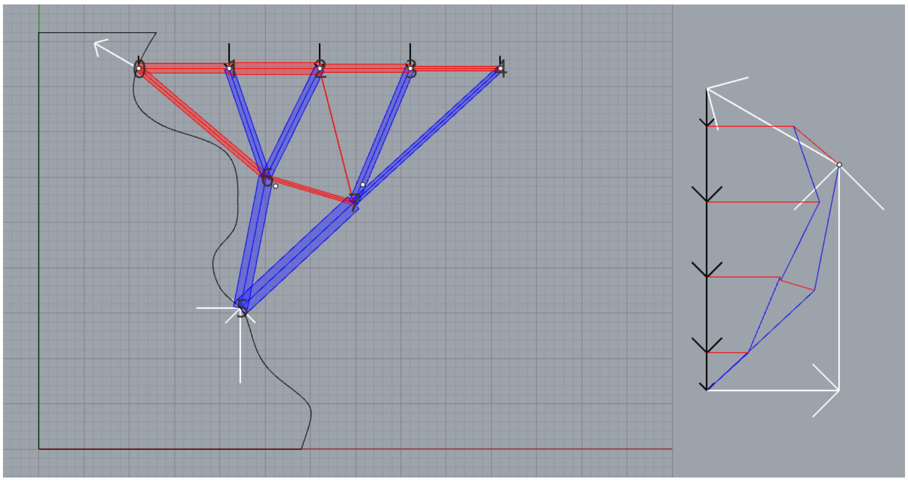


Figure D.25: The improved optimized design

Table D.12: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
7059	6720	95%

Participant 8

Design 1 of 1:

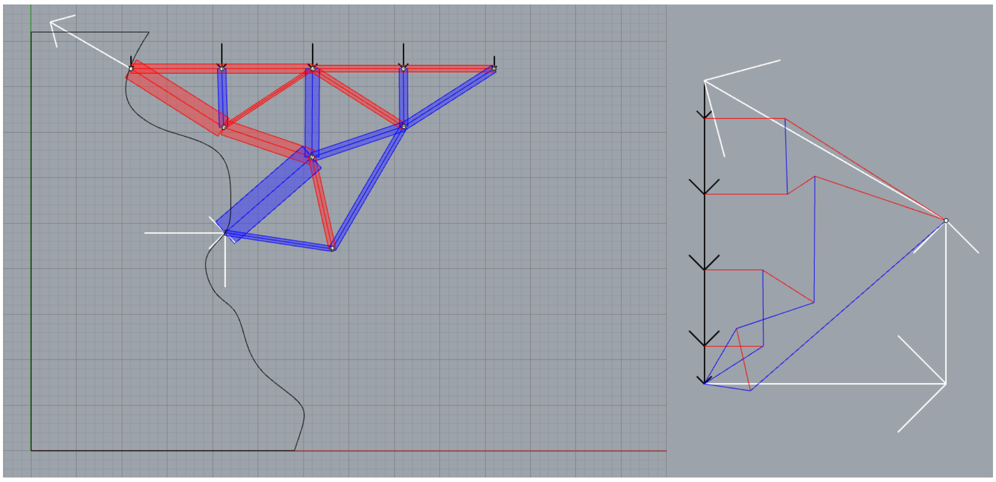


Figure D.26: The initial manual design

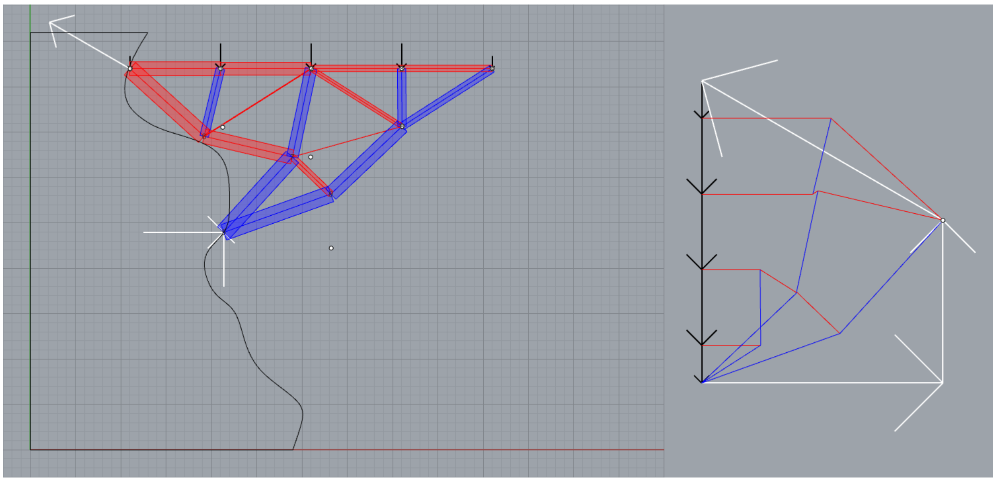


Figure D.27: The improved optimized design

Table D.13: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
5842	5110	87%

Participant 9

Design 1 of 1:

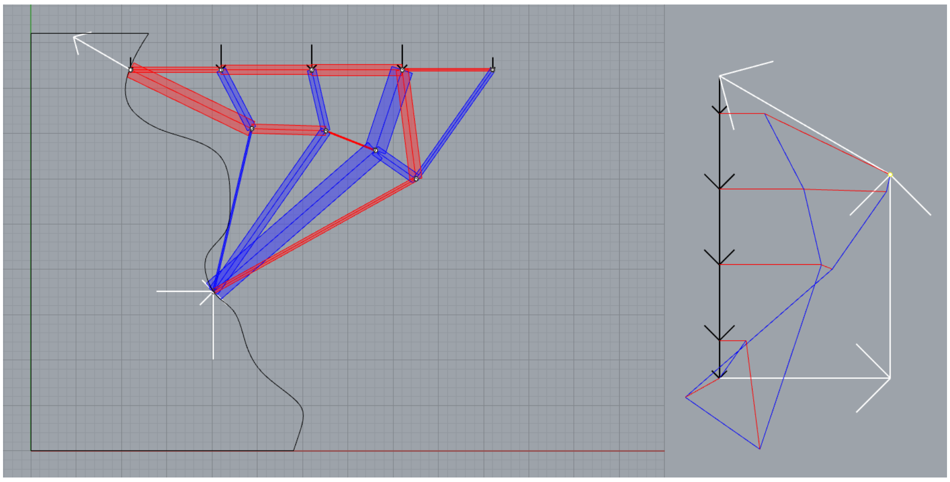


Figure D.28: The initial manual design

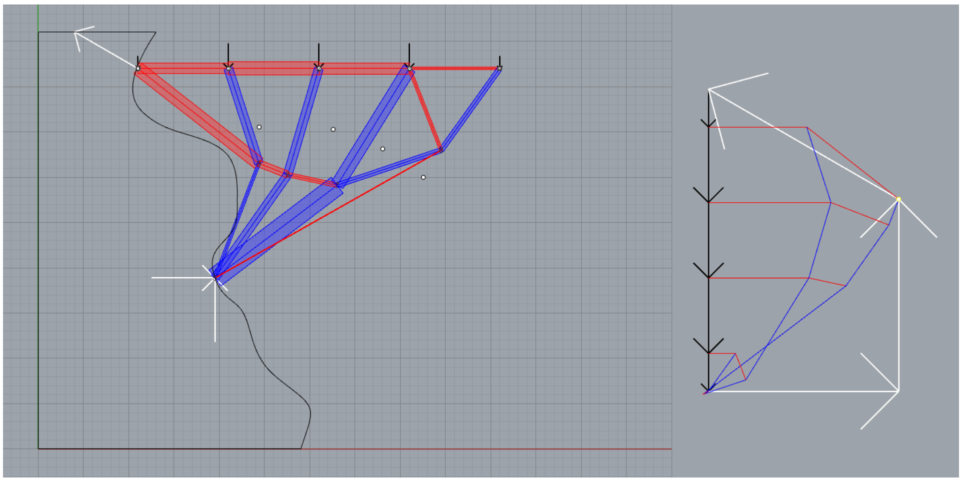


Figure D.29: The improved optimized design

Table D.14: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
11756	6376	54%

Participant 10

Design 1 of 1:

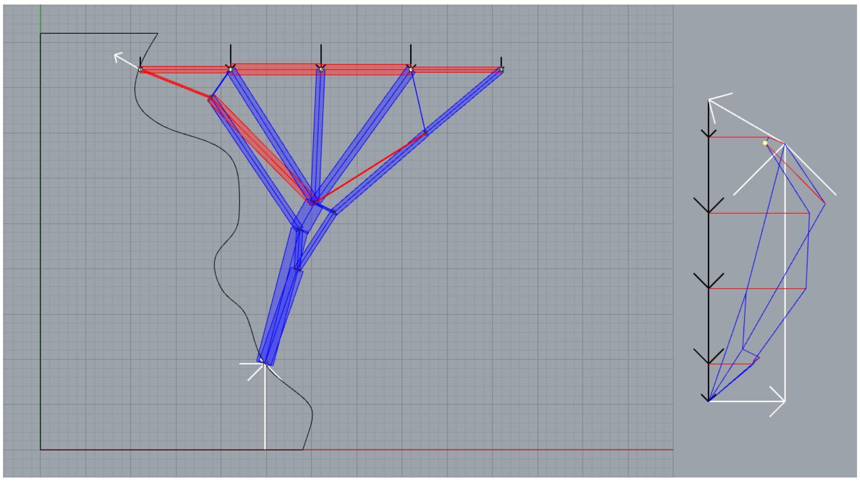


Figure D.30: The initial manual design

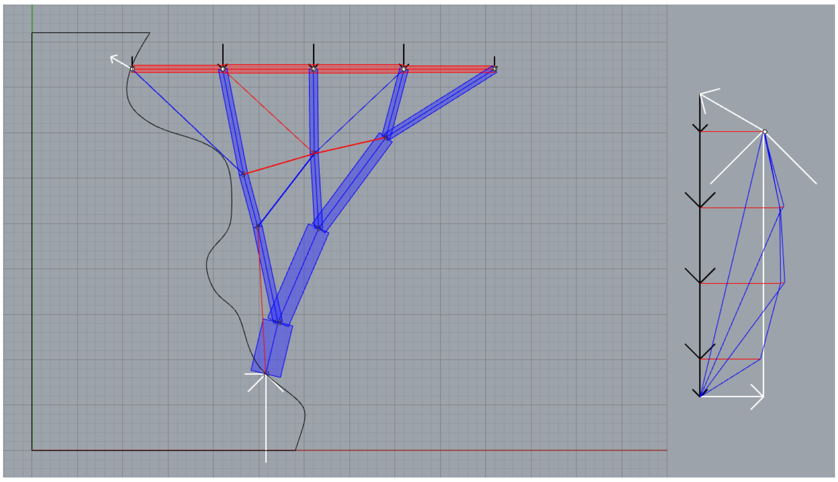


Figure D.31: The improved optimized design

Table D.15: Fitness values $[kNm]$ of the initial manual design and the improved optimized design

Initial design	Optimized design	Percentage
7578	4677	62%

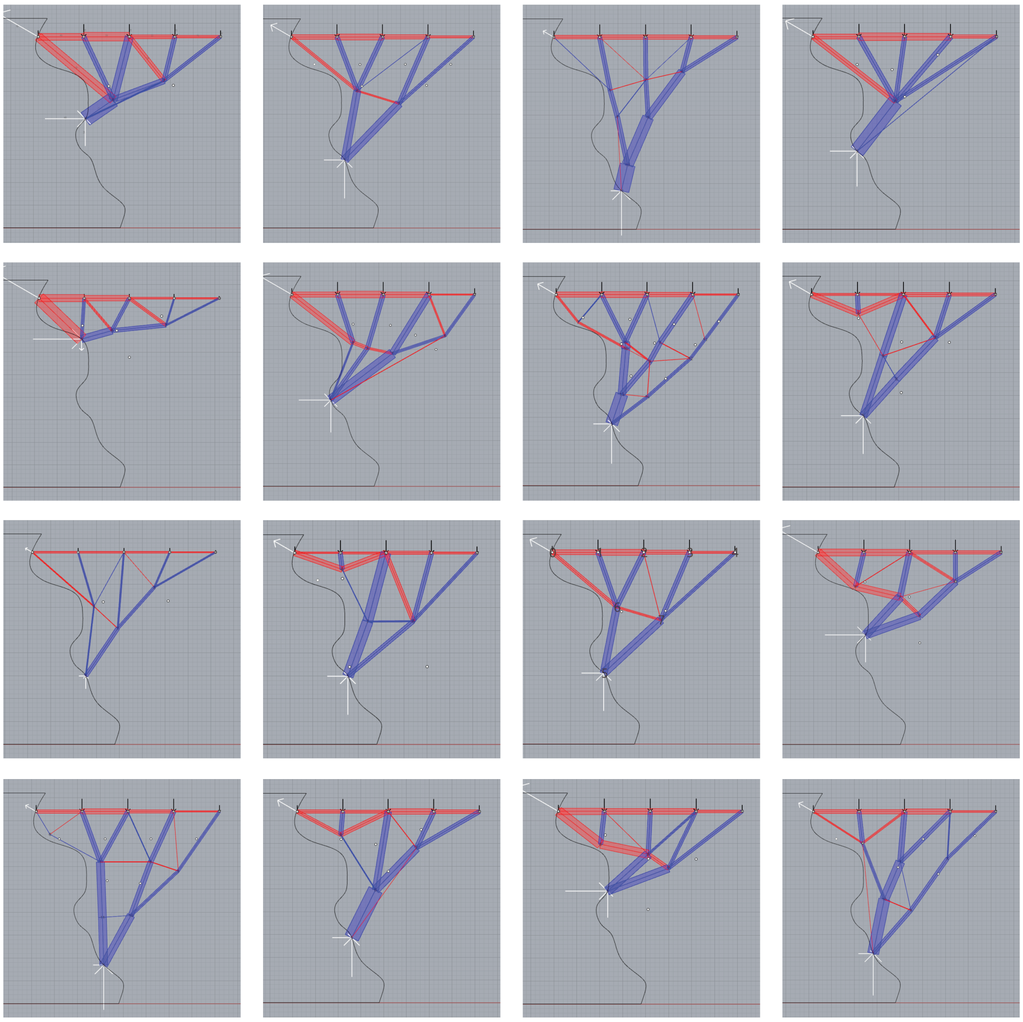


Figure D.32: An overview in random order of the unified diagrams of every optimized design created with GSDesgin for the experiment

D.2 Questionnaire

D.2.1 Multiple-choice questions

The experiment: Step 1 - manual design:

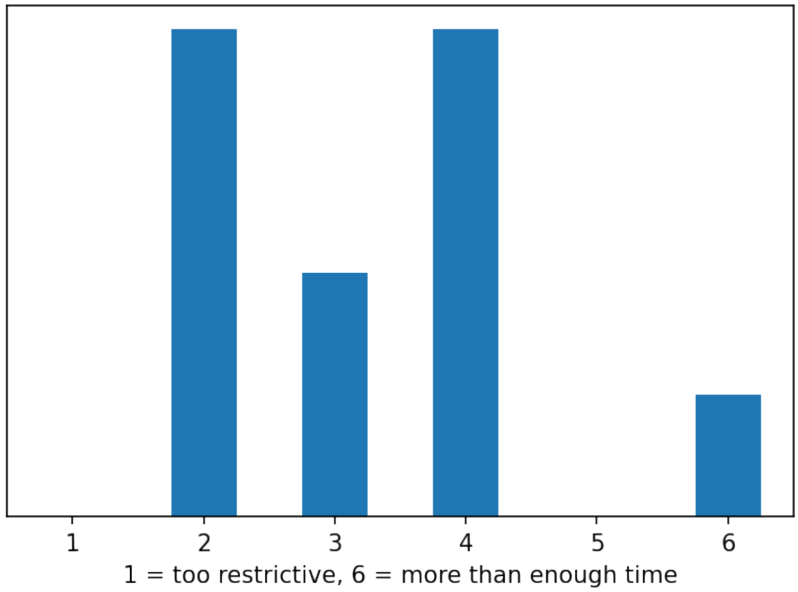


Figure D.33: How did you experience the 15 minutes time limit?

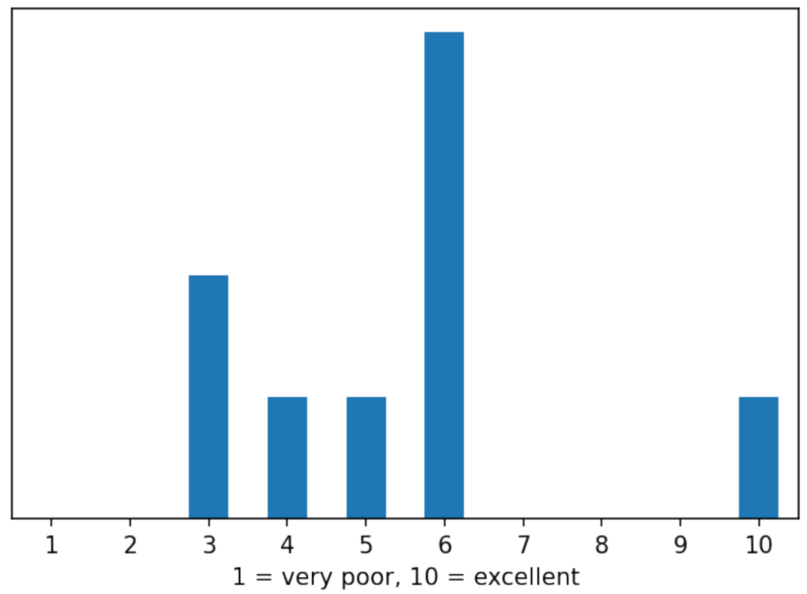


Figure D.34: In terms of structural efficiency, how would you rate your design(s) produced in step 1?

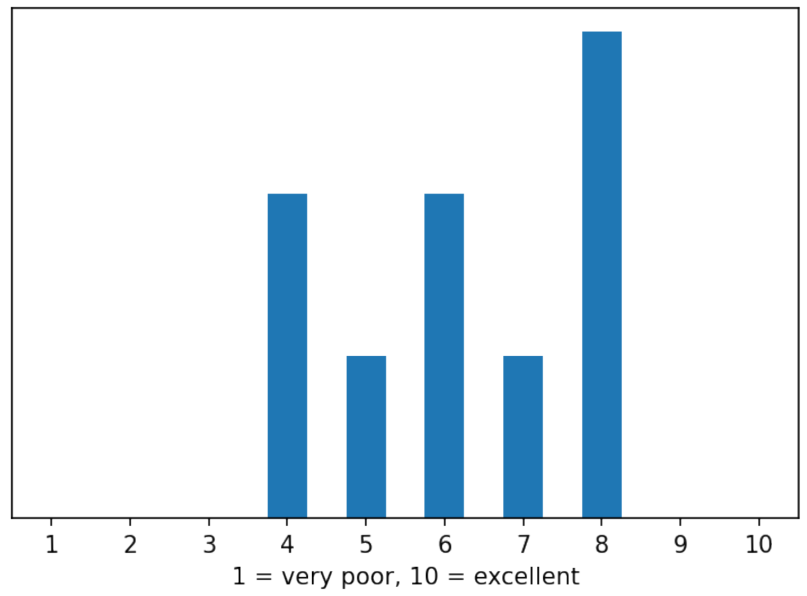


Figure D.35: In terms of practical feasibility, how would you rate your design(s) produced in step 1?

The experiment: Step 2 - GSDesign:

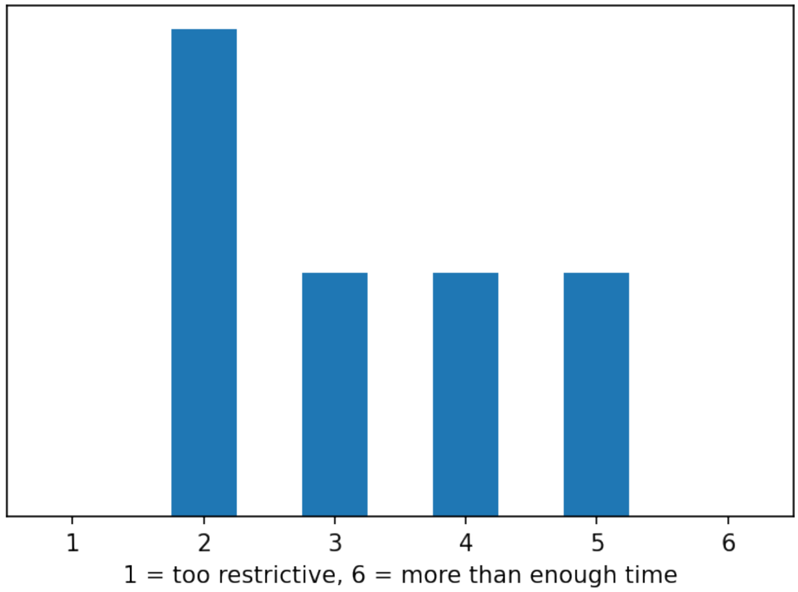


Figure D.36: How did you experience the 30 minutes time limit?

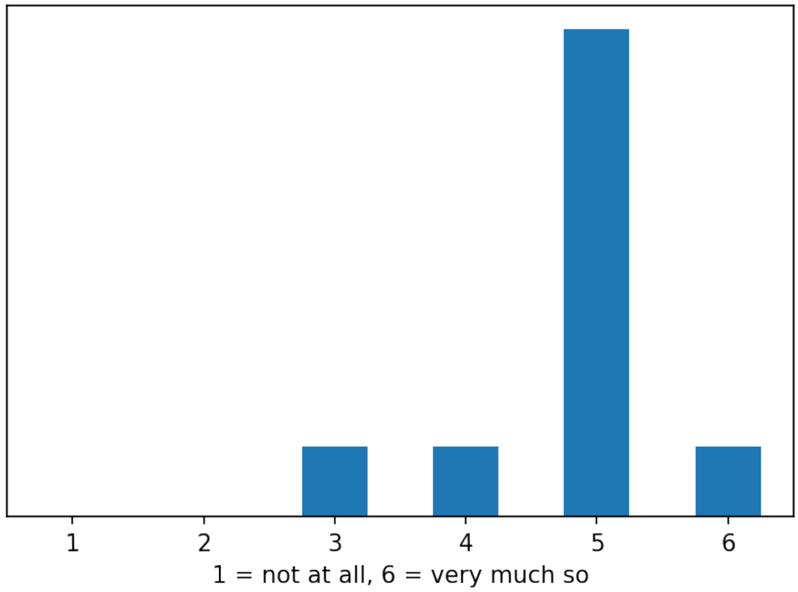


Figure D.37: Did the use of GSDesign provide you with new insights regarding this design case?

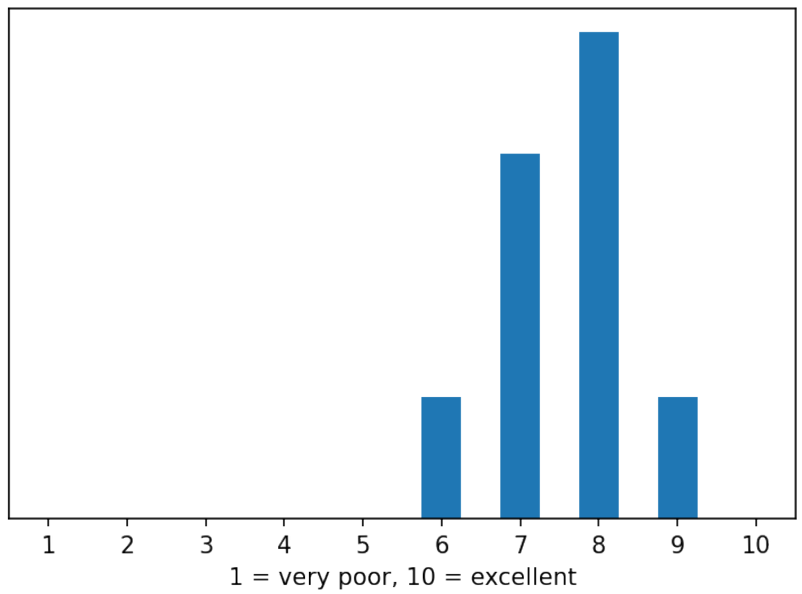


Figure D.38: In terms of structural efficiency, how would you rate your design(s) produced in step 2?

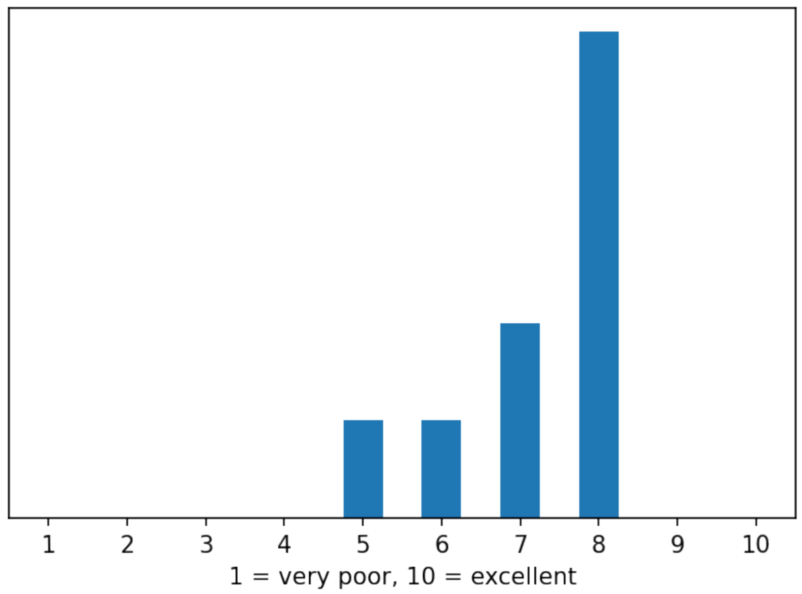


Figure D.39: In terms of practical feasibility, how would you rate your design(s) produced in step 2?

General opinion on GSDesign:

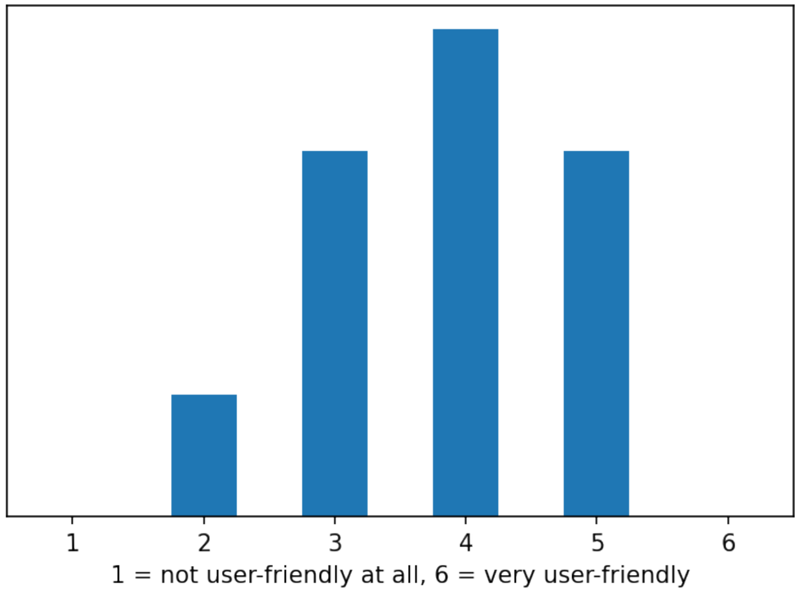


Figure D.40: How would you rate the user-friendliness of GSDesign?

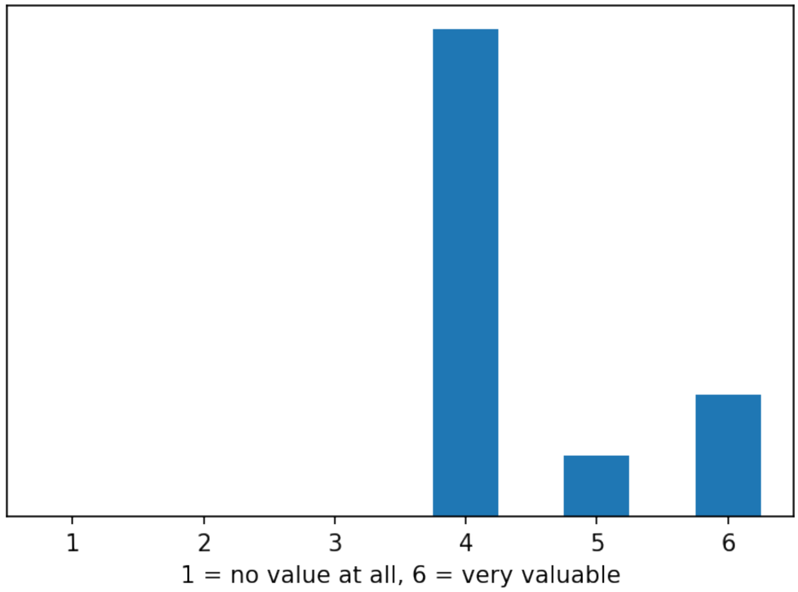


Figure D.41: How would you rate the practical value of current version of GSDesign for structural designers?

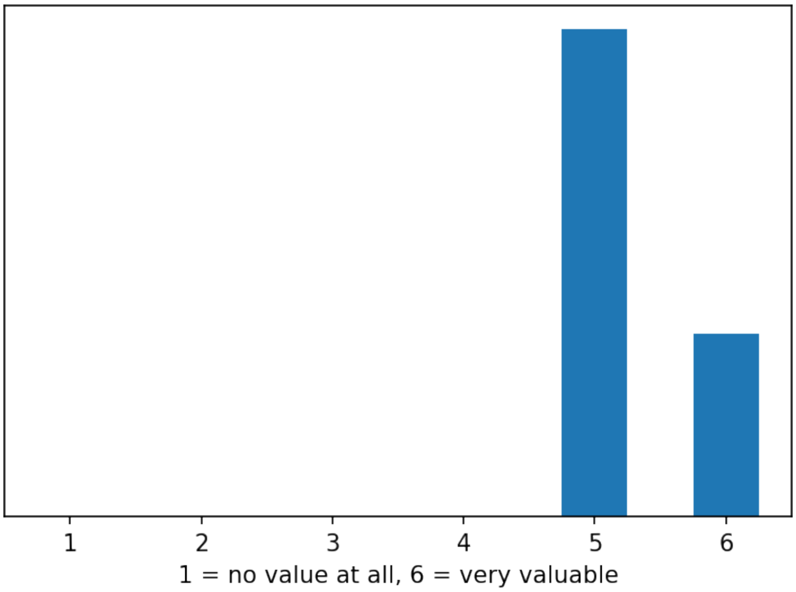


Figure D.42: How would you rate the educational value the current version of GSDesign could have for students?

D.2.2 Open questions

Table D.16: Next to pen and paper, what tools did you make use of during step 1?

Just drew in Rhino due to lack of scanner availability
Calculator, sketching software to make the truss sketch
Just using pen and paper
Nothing - I just had a guess!
Adobe Illustrator
Only digital Figma (no paper)
Bluebeam/pdf editor

Table D.17: In your opinion, what are the strengths of GSDesign?

Computational efficiency, combination of design input and optimisation as a design aid.
Being able to see the force flow and the unified strength diagrams was interesting and helpful, because I could see easily how the forces were moving through the truss. I also liked the flexibility in the optimization process; I had a lot of freedom to choose what elements I wanted to optimize, such as the location of the nodes or the location of the members. This allowed me to try out many different iterations. The optimization also made me realize that I had added an unnecessary node, which I deleted in later iterations and made my structure more efficient.
Giving insights for engineers to design an efficiency structure.
The real-time force diagrams are great - would be a brilliant resource for anyone learning graphic statics. The fitness function is a simple but pretty effective proxy for tonnage which I think is appropriate for initial concept studies. The graphic display of the member forces, loads and reactions is very clear. I like the workflow of starting from a 'human-created' concept, before tweaking the geometry to improve the efficiency. In many cases I would guess that this workflow is likely to yield more practical solutions that some other optimization methods e.g. discretised layout optimization or topology optimization (although these other methods may produce a lower theoretical tonnage).
The visual aspect is very appealing, it gives quick insight on how the design is made optimal. This insight has a positive impact on the design process.
Quick insights in current design, including improvement of that design. Including the direct visualization of forces (when inputting the design, as well running the optimization)
Very fast, intuitive tool. Very easy to use.
Direct link to structural mechanics via graphic statics information. Optimisation possibilities. Use of colours in the output. Nice option that it is possible to follow the curve of the mountain slope for the support point. Also the supporting text in GH to clarify what input/actions were needed was very helpful.

Table D.17: (continued)

Compared to the lay-out optimization tool I used for the first design, you directly incorporate practicality of the designed structure, i.e.: how many joints, how many members (This is possible in the lay-out tool, but is a bit more a black box). In case of not having a lay-out optimization tool it would definitely be an aid for designers to quickly find structurally efficient solutions, structures that are not straight-forward but help to save material. I haven't used the buckling or penalty factors, but it seems - in case of correct functioning - useful as well.
Parametric 2DGS (other tools are not straightforward to install or limited to 3DGS)
The optimization with Galapagos

Table D.18: In your opinion, what are the weaknesses of GSDesign?

Loading in a design is a rather manual process. Limitation to statically determined designs.
It might be confusing to put together a solution if there is not already an example file ready as you provided for this experiment, because there are so many inputs and outputs to each component. It might be easier for the user to understand if some inputs/outputs could be simplified, for example to have a single "visualisation" output that goes out of "Solve member forces" into the Visualisation component (or a few, depending on what would be possible to implement).
Less consideration on the practical requirements and constructability.
Would be good to further clarify in the user documentation how the buckling factor is calculated. Requirement for statically determinate structures is a limitation.
Performance of the design is not displayed in an easy to understand way. Fitness is very abstract. Expressing the variant in total weight where every members is selected based on buckling criteria could give more insight about the performance differences between the hand sketch and GSDesign. Second the optimization space is limited by the number of nodes the user defines. I am wondering if the suggested design by GSDesign could improve if an extra node was included or removed.

Table D.18: (continued)

Optimizing on fixed amount of members and joints. I am not totally sure what happens, but the script could run an optimization if you did not connect all the sliders to the "joints" collector and/or the Genome input. So in that case it is a bit error prone, especially for first time users. However your comments are there, but if you do multiple designs in a rush, people could forget.
Limited to 2D truss structures.
Requires a bit of start-up training (thanks for your help). Although I have been using Rhino-GH in the past, I'm not a daily user. I assume without your support I would have struggled a bit with the user-interface and expected actions.
With regards to the lay-out optimization you need to think of a truss structure yourself. So here potential optimal solutions could get lost when using a non-optimal start design. You need to know how to drive Grasshopper (and you need a license for Rhino). So maybe it can be placed on a web interface without the need for Rh+Gh (like the lay-out tool). It was a bit unclear to me what we are optimizing for? Total member forces, strain energy? So what is the outcome; a structure with least volume, largest stiffness? Assigning the relations between the nodes (drawing the members) is something to figure out and could be improved in the future. A warning needs to be made to the user to avoid overlapping members for the 'free nodes'. The movable nodes can become tangled as such that overlapping members can occur. This could result in undesirable outcomes. Small note: make sure that the user depreviews the mesh edges (ctrl+M), otherwise the loaded image is not well visible.
Too many unnecessary (topological) inputs. Should be handled internally.
That it doesn't seem very user-friendly, esspecially for people that don't know about rhino and grasshopper

Table D.19: What kind of features are you missing right now, and would you like to see implemented?

Adjustment of solution topology after a certain stage of optimisation; Auto-collapsing of nodes, removal of unstressed members, etc.
It would be nice to see the deflection of the frame under the applied forces.
None

Table D.19: (continued)

For application to larger structures, it might be useful to update the member input methodology - maybe could be based on lines drawn in Rhino rather than typed values?
It is quite some work to copy the sketch correctly position the nodes and define all member relationships. It would be more satisfying if this would be executed automatically based on the sketch.
Some small ones: Giving feedback if you have enough sliders as extra nodes. Giving some more attention to the result of the optimization (include a step 5, with some explanation of the result). Personally I only checked my results after I finished the experiment. And then I also discovered some mistakes that I made in the set-up. Probably due to the rush of the 30 minutes. Some more advanced features are dragging of the members instead of inputting the connection numbers.
More options with buckling, variable buckling lengths perhaps.
Some comparison of the pre- and post-optimalisation structures could be interesting, to see more clear which nodes have moved and what the effect of that was. Also redundant members that are not really necessary remain included in the model; you could add suggestions to remove these. In construction games such as Bridge Constructor, it is visualised which members are not strong enough. This is a powerful clue as to where to improve the structure. Moving loads / multiple load cases could be a useful additional feature.
The possibility to optimize for support reactions, e.g. optimize (minimise) horizontal/vertical support reactions. Direct export to analysis packages; members, support and loading conditions). Question: could it potentially assign cross-sections automatically?
Handling static indeterminacy? Maybe include Bow's notation?
Maybe a human UI window to improve the user-friendliness, and make it more accessible for people that don't known rhino and grasshopper.

Table D.20: Do you have any additional remarks? (only content-related answers are shown

It's a really cool tool! I enjoyed going through the process and optimizing my structure. It was really interesting to see all the different things I could change and the effects they had on the appearance of the structure.
Nice work! keep going on. For usability tips check; eight golden rules by Ben Shneiderman
There were stability issues when I tried a geometry with more nodes.