

Czech Technical University in Prague)
Faculty of Civil Engineering

Department of Concrete and Masonry Structures

Field of study: Building Structures



**Enhancing Early-Stage Design Process of
Concrete Structure Using Parametric Modelling
and Multobjective Optimization**

MASTER THESIS

Author: Durdon Qurbonova

Supervisor: Ing. Martin Petřík, Ph.D.

May 2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Qurbonova Jméno: Durdona Osobní číslo: 476968
 Zadávající katedra: Katedra betonových a zděných konstrukcí
 Studijní program: Stavební inženýrství
 Studijní obor/specializace: Konstrukce pozemních staveb

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce: Vylepšení procesu návrhu betonové konstrukce v rané fázi pomocí parametrického modelování a víceúčelové optimalizace

Název diplomové práce anglicky: Enhancing Early-Stage Design Process of Concrete Structure Using Parametric Modelling and Multiobjective Optimization

Pokyny pro vypracování:

- Zpracování parametrického modelu analyzovaného objektu v prostředí Grasshopper
- Stanovení kritérií optimalizace
- Vyhodnocení a srovnání jednotlivých řešení
- Ověření dimenzovatelnosti

Seznam doporučené literatury:

- Karamba3D 2.2.0 manual
- <https://www.ladybug.tools/resource.html>
- Wallacei Primer 2.0
- Deb, Kalyan & Pratap, Amrit & Agarwal, Sameer & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation, IEEE Transactions on. 6. 182 - 197. 10.1109/4235.996017.

Jméno vedoucího diplomové práce: Ing. Martin Petřík, Ph.D.

Datum zadání diplomové práce: 28.02.2023 Termín odevzdání DP v IS KOS: 22.05.2023
 Údaj uveďte v souladu s datem v časovém plánu příslušného ak. roku

Podpis vedoucího práce

Podpis vedoucího katedry

III. PŘEVZETÍ ZADÁNÍ

Beru na vědomí, že jsem povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je nutné uvést v diplomové práci a při citování postupovat v souladu s metodickou příručkou ČVUT „Jak psát vysokoškolské závěrečné práce“ a metodickým pokynem ČVUT „O dodržování etických principů při přípravě vysokoškolských závěrečných prací“.

28.2.2023
 Datum převzetí zadání

Podpis studenta(ky)

Declaration

I, Durdona Qurbonova, hereby declare that I have written my Master's thesis on Enhancing Early-Stage Design Process of Concrete Structure Using Parametric Modelling and Multiobjective Optimization independently using only materials (literature, projects, software, etc.) listed in bibliography.

Prague, Czech Republic

May 2023

.....

Durdona Qurbonova

Acknowledgements

I would like to thank my supervisor Ing. Martin Petřík, Ph.D. for his professional guidance and invaluable advice. I would also like to thank my family, partner and friends for their support and encouragement during the course of this thesis.

Durdona Qurbonova

Title:

Enhancing Early-Stage Design Process of Concrete Structure Using Parametric Modelling and Multiobjective Optimization

Author: Durдона Qurbonova

Study program: Civil Engineering

Field of study: Building Structures

Type of work: Master thesis

Supervisor: Ing. Martin Petřík, Ph.D.
Department of Concrete and Masonry Structures

Abstract: The early-stage design process of concrete structures is a critical phase in construction projects, influencing the efficiency and quality of the final product. However, this phase is challenging due to the complex and variable nature of design parameters, as well as the need to balance multiple conflicting objectives, including visual aesthetics, structural performance, and user comfort. This diploma thesis explores the application of parametric modelling and multi-objective optimization techniques, specifically using evolutionary genetic algorithms, to enhance the early-stage design process of concrete structures. The research aims to provide practical insights into the effectiveness and feasibility of employing modern tools in this phase, ultimately contributing to the advancement of efficient and sustainable design practices. Through the use of parametric modelling and multi-objective optimization, this research offers a systematic approach to addressing the challenges inherent in early-stage design, enabling designers to efficiently explore design alternatives and make informed decisions. The findings of this study have implications for architectural and engineering practices, emphasizing the potential benefits of incorporating parametric modelling and multi-objective optimization in the early stages of the design process.

Keywords: Parametric modelling, visual programming, multi-objective optimization, Grasshopper, NSGA-II, early-stage design process, concrete structures

Contents

List of Tables	ix
List of Figures	x
Introduction	1
1 Parametric modelling	3
1.1 The History	3
2 Architecture	7
2.1 Pre-Design	8
2.1.1 Project Objectives	8
2.1.2 Inspiration	8
2.2 Petřiny Panorama House: Design Process	11
3 Visual Programming	17
3.1 Grasshopper	17
3.2 Design Process	18
3.2.1 Parametric Model of The House	18
3.2.2 Designing The Tree Column	18
3.2.3 Parametric Openings	20
3.2.4 Cantilever Length	21
3.2.5 The Awnings	21
4 Structural Analysis	23
4.1 The Importance of Structural Analysis in Early-Stage Design	23
4.2 An Integrated FEM tool within Grasshopper	23
4.2.1 Karamba 3D	24
4.3 The FEM Model	25
4.3.1 Slabs	25
4.3.2 Tree Column	26
4.3.3 Loads	26
4.4 Setting the Structural Criteria	27
5 Environmental Analysis for Indoor Comfort	30
5.1 Importance of Environmental Analysis for Indoor Comfort	30
5.1.1 Daylight Analysis	31
5.1.2 Assessing the View Out of Windows	31
5.1.3 Access to Sunlight	31
5.1.4 Prevention of Glare	31
5.2 Process of Integrated Environmental Analysis for Optimized Design	32
5.2.1 Integration of Comprehensive Analysis Tool	32
5.2.2 Ladybug - Functionalities and Benefits	32

5.2.3	Environmental Analysis Setup	33
5.3	Preparing Environmental Criteria for Optimization	36
6	Multi-objective Optimization	38
6.1	Multi-objective Optimization Methods	38
6.2	NSGA-II	40
6.2.1	NSGA-II: Working Principle and Benefits	41
7	Optimization Process	45
7.1	Multi-objective Optimization Tool	46
7.2	Setting Optimization Criteria	47
7.3	Optimization Setup	50
7.4	Analyzing Optimization Results	54
7.5	Decision Making Process	59
7.6	Exploring Different Optimization Results	60
7.6.1	The Chosen Solution	67
8	Cross-Sectional Analysis	72
8.1	Tree Column Branch and Slab Connection	73
9	Digital fabrication	75
	Conclusion	81
	Bibliography	83
	Appendices	85
A	Definition of the designed model in Grasshopper	85
B	Decoded genomes	86
C	Phenotypes of Pareto front solutions	90
D	Generated formwork	92

List of Tables

4.1	Tree column cross-sectional dimensions	26
4.2	Load conditions	27
4.3	Weights of each structural criteria for "Weighted Average" operation .	29
5.1	Weights of each environmental criteria for "Weighted Average" operation	36
7.1	Best and worst results of the structural criteria obtained from the "Data" output of "Wallacei X" component	62
7.2	Best and worst results of the environmental criteria obtained from the "Data" output of "Wallacei X" component	63
7.3	Fitness values of the chosen solution (normalized)	67
7.4	Real values of the component criteria of the chosen solution	67
7.5	The cross-sectional dimensions of the tree column of the chosen solution	68
7.6	Real values of the component criteria of the structural criterion for cantilever of a length of 9 m	69
8.1	Internal forces obtained from Karamba 3D for the chosen cross-section	72

List of Figures

1	MacLeamy Curve: the impact of the changes made in different stages of the design process [6]	1
1.1	Hooke's anagram of the hanging chain model. At the time, anagrams were a common way to claim the first publication of an idea before the results were ready to publish [23]	4
1.2	Gaudi's hanging chain model of Colònia Güell [10]	4
1.3	Frei Otto experimenting with soap bubbles [10]	5
2.1	"4x4" house by Tadao Ando	7
2.2	Map of Petriný	8
2.3	R128 by Werner Sobek [22]	9
2.4	House with One Wall by Christian Kerez [25]	9
2.5	House with One Wall Concept [25]	10
2.6	Crossed House by Clavel Arquitectos [27]	10
2.7	Crossed House plan by Clavel Arquitectos [27]	11
2.8	The concept of the designed house	12
2.9	The floor plans of the designed house	13
2.10	Illustration of tree columns of Sagrada Familia Cathedral designed by Gaudi (in right) and a tree (in left) [21]	14
2.11	Model of tree structure columns by Otto Frei [ottotree]	14
2.12	Tree Column by RWPA as inspiration for load-bearing structure [20] .	15
2.13	Sketch of the designed column	15
2.14	North elevation of the designed house	16
2.15	Visualisation of the designed house (result of the provided optimization)	16
3.1	Chosen software for parametric modelling [26]	17
3.2	Three parametric options of tree column	18
3.3	Modelling process of chosen tree column	19
3.4	"Evaluate Curve" component of Grasshopper	19
3.5	"Evaluate Surface" component of Grasshopper	20
3.6	Illustration of tree columns with different parameters	20
3.7	Illustration of 3rd floor openings with different parameters	21
3.8	Illustration of the awnings with different parameters	22
4.1	Karamba 3D - visualization of deflection of a model	24
4.2	"Mesh Breps" component of Karamba 3D [7]	25
4.3	N-M-M capacity diagrams of chosen croce-sectional dimensions	26
4.4	Normalization of stress value using Grasshopper components	28
5.1	Ladybug Tools for environmental analysis [8]	33
5.2	Ladybug Tools components used for environmental analysis	33

5.3	Position of the sun in summer	34
5.4	Position of the sun in winter	35
5.5	Position of the sun in period from the 1st of February to the 21st of March	35
5.6	Normalization of sunlight value using Grasshopper components	36
6.1	Basic components of genetic algorithm [5]	39
6.2	NSGA-II algorithm flowchart [14]	41
6.3	Ranking process in NSGA-II	42
6.4	Illustration of crowded distance in NSGA-II [2]	43
6.5	Schematic of NSGA-II procedure [2]	44
7.1	Illustration of optimization process [9]	45
7.2	Environment of Wallacei plugin [13]	47
7.3	"Weighted Average" operation definition for structural criteria in Grasshopper	48
7.4	Collision detection and normalization of the number of collisions in Grasshopper	49
7.5	"Weighted Average" operation definition for environmental criteria in Grasshopper	49
7.6	"Wallacei X" component's inputs illustration	51
7.7	Wallacei X control panel for illustrating "Random Seed" and "Mutation Probability" options	52
7.8	Illustration of the results of test optimizations with "Random Seed" option set to 1	52
7.9	Illustration of the results of test optimization with "Random Seed" option set to 0	53
7.10	Illustration of the results of test optimizations with different "Mutation Probability"	53
7.11	Illustration of optimization progress of the second optimization process	54
7.12	"Wallacei Analytics" tab for analyzing the provided optimization	55
7.13	Explanation of Standard Deviation chart in "Wallacei Analytics"	55
7.14	Fitness Values chart in "Wallacei Analytics"	56
7.15	Visualization of Diamond Fitness chart in "Wallacei Analytics"	56
7.16	Illustration of analysis of the first (structural) criteria with rank 0	57
7.17	Illustration of analysis of the third (environmental) criteria with rank 0	57
7.18	Illustration of analysis of the fourth (distance between the trees) criteria with rank 0	58
7.19	Visualization of clustering process in "Wallacei Selection" tab	58
7.20	Illustration of utilizing the "Diamond Fitness Chart" component in Grasshopper environment	59
7.21	Definition of sorting process in Grasshopper	60
7.22	Selected four best solutions - one from each optimization process	60
7.23	"Wallacei X" component's outputs illustrations	61
7.24	Illustration of decoded genomes using "Decode Genome" component	61
7.25	Illustration of the compared fitness values of the structural criterion obtained from the "Fitness Values" output of "Wallacei X" component	62

7.26	Illustration of the compared fitness values of the environmental criterion obtained from the "Fitness Values" output of "Wallacei X" component	62
7.27	Illustration of the deformed geometries of the compared structural fitness values	64
7.28	Illustration of the M_y 's of the compared structural fitness values . . .	64
7.29	Illustration of the M_z 's of the compared structural fitness values . . .	65
7.30	Illustration of the results of compared environmental fitness values (top view)	65
7.31	Illustration of the results of compared environmental fitness values (perspective view)	66
7.32	Illustration of the difference between trees and awning shapes of the compared environmental fitness values	66
7.33	Visualization of the structural analysis results of the chosen solution .	68
7.34	Cantilever 9m	69
7.35	Visualization of the environmental analysis results of the chosen solution	70
7.36	Illustration of the awning shape and tree positions for the chosen solution	70
7.37	Illustration of the results of the environmental analysis comparing the column of the chosen solution with a column of maximum thickness .	71
8.1	N-N-M capacity diagram from provided analysis for the chosen cross-section	73
8.2	Design of reinforcement for the chosen cross-section	73
8.3	Informative reinforcement scheme of the reinforcement design in intersection points of the tree column	74
8.4	3D illustration of the Schöck Isokorb® XT type B	74
8.5	2D scheme of the connection of the column branch and slab using Schöck Isokorb® XT type B	74
9.1	"Pack Objects" component in Grasshopper	76
9.2	OpenNest component in Grasshopper	76
9.3	Generated small pieces of formwork using OpenNest plugin	77
9.4	Generated big pieces of formwork using OpenNest plugin	77
9.5	Axonometric visualization of the tree column formwork	78
9.6	Exploded axonometric visualization of the tree column formwork . . .	79
9.7	Illustration of RWPA inspirational column (see Figure 2.12) formwork	79
9.8	Illustration of RWPA inspirational column (see Figure 2.12) construction process	80
9.9	Showcase of a column formwork with a finishing plastic layer on sheathing [18]	80
10	Definition of the designed model in Grasshopper	85
11	Decoded genomes of the first and last generations of the first optimization process	86
12	Decoded genomes of the first and last generations of the second optimization process	87
13	Decoded genomes of the first and last generations of the third optimization process	88

14	Decoded genomes of the first and last generations of the fourth optimization process	89
15	Pareto front solutions of the first provided optimization	90
16	Pareto front solutions of the second provided optimization	90
17	Pareto front solutions of the third provided optimization	91
18	Pareto front solutions of the fourth provided optimization	91
19	Generated formwork with big pieces	92
20	Generated formwork with small pieces	93

Introduction

In today's highly competitive construction market, the ability to deliver projects that meet clients' requirements, adhere to standards, and satisfy cost and time constraints is of utmost importance. Efficient management of the design process is crucial to ensure that these objectives are achieved before the commencement of construction. Any interference or disruptions in the construction process due to design changes or delayed information supply can result in significant costs and delays.[1]

Figure 1 represents the MacLeamy Curve, which illustrates the concept of making design decisions earlier in the project when the opportunity to influence positive outcomes is maximized and the cost of changes minimized, especially as regards the designer and design consultant roles. [6]

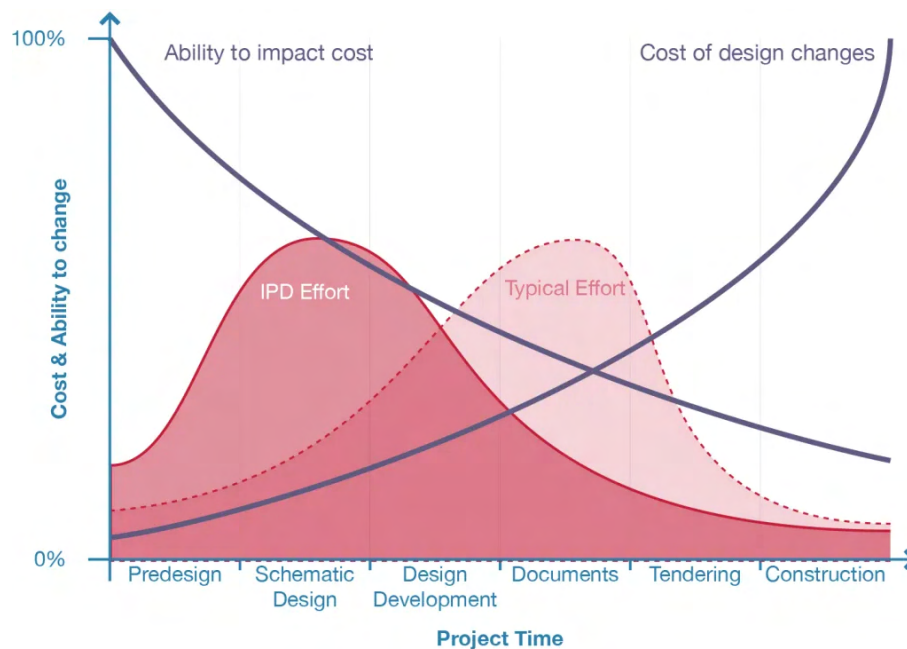


Figure 1: MacLeamy Curve: the impact of the changes made in different stages of the design process [6]

Design decisions made during the early stages of a project have a profound impact on the entire construction process. The choices made at this critical phase can influence the project's overall success, including its cost, timeline, and quality. In contrast, decisions made towards the end of the design process, when construction is

imminent, often have limited flexibility and can be costly to modify. Early-stage design decisions set the foundation for the project and establish the framework within which subsequent design iterations will unfold.

Traditionally, the early-stage design process of concrete structures has been associated with challenges. Ineffective design management can lead to prolonged design timescales and conflicting construction details, causing delays and complications during the construction phase. The cost implications of design changes or late information supply to the construction team can be substantial. These challenges necessitate a more efficient and streamlined approach to the early-stage design process.

Parametric modelling and multi-objective optimization offer promising solutions to enhance the early-stage design process of concrete structures. By leveraging parametric modelling, designers can create dynamic digital models that allow for flexible exploration and manipulation of design variables. This approach enables the generation of numerous design alternatives and facilitates quick iterations to evaluate different design possibilities.

Multi-objective optimization complements parametric modelling by integrating diverse design criteria and objectives. It allows designers to simultaneously optimize multiple aspects, such as structural performance, cost, energy efficiency, and constructability. By exploring the trade-offs between these objectives, designers can make informed decisions that balance competing factors, ultimately leading to more efficient and optimized design solutions.

The application of parametric modelling and multi-objective optimization in the early-stage design process offers several advantages over traditional approaches. Firstly, it enables designers to explore a broader design space and consider various design possibilities, leading to more creative and innovative solutions. Secondly, it makes it possible to systematically evaluate the design alternatives based on multiple criteria, enabling a comprehensive analysis of the design's performance and feasibility. This approach reduces the risk of unforeseen complications during the construction phase and improves the overall quality and efficiency of the design.

Furthermore, the integration of parametric modelling and multi-objective optimization enhances collaboration among project stakeholders. It promotes interdisciplinary communication, facilitates decision-making, and fosters a shared understanding of design objectives and constraints. This collaborative approach reduces the likelihood of conflicts and delays, improving overall project coordination and success.

Chapter 1

Parametric modelling

It is accurate to say that parametric modelling has revolutionized the way architects and engineers design structures. This modelling process allows for real-time modification of a model's geometry based on changes to dimensional values through computer programming code, typically in the form of a script that defines the dimensions and shapes of the model.

The resulting 3D representation of the model in drafting programs provides a realistic representation of the project's behaviour, and it is common for parametric models to use feature-based modelling tools to manipulate the model's attributes.

Furthermore, it is important to note that the design created using parametric modelling is called Parametric Design or Parametricism, a term coined by Patrik Schumacher. This design approach utilizes computational algorithms and parameters to create complex and intricate forms.

1.1 The History

Even though it seems to be something new in the Architectural Engineering and Construction sector, the term parametric has a long history in mathematics [23]. The earliest examples of parametric being used to describe a three-dimensional model come from the 19th century in the works of James Dana called "In the Drawing of Figures of Crystals", John Leslie called "Geometrical Analysis and Geometry of Curve Lines" and Samuel Earnshaw called "On the Nature of the Molecular Forces which Regulate the Constitution of the Luminiferous Ether" [12]. The first use of parametric modelling in architecture belongs to Antonio Gaudi. We can not say if Gaudi was familiar with the works of mathematicians mentioned above, Gaudi's work certainly employed models underpinned by parametric equations when designing architecture [23]. The use of parametric equations can be seen in many aspects of Gaudi's architecture but is perhaps best illustrated by his use of the hanging chain model. The hanging chain model originates from Robert Hooke's anagram "abcccd-deeeefggiiiiiiiillmmmmnnnnnooprssstttttuuuuuuuuu" shown in Figure 1.1, which unscrambled and translated from Latin reads "as hangs the flexible line, so but

computes the parametric outcomes. Rather than manually calculating the outputs from the catenary curve's parametric formula, Gaudí could automatically derive the shape of catenary curves through the force of gravity acting on strings. This method of analogue computing was enlarged by Frei Otto to include, amongst other things, minimal surfaces derived from soap films and minimal paths found through wool dipped in liquid. Otto calls designing with these models form finding, which is a phrase that foregrounds the exploratory nature of parametric modelling [23].

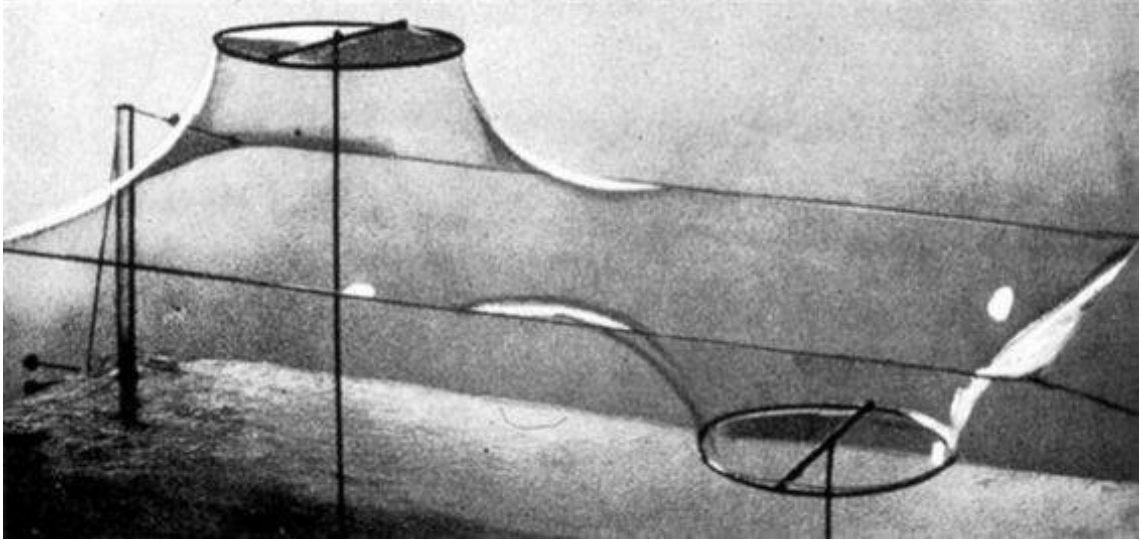


Figure 1.3: Frei Otto experimenting with soap bubbles [10]

In the late 20th century the term parametric architecture became a part of architectural discourse [12]. In 1960 Luigi Moretti presented versions of parametric stadiums as part of his parametric architecture. In the same decade, Sutherland created the first interactive computer-aided design program “Sketchpad”. His contemporaries were optimistic about the development of computer-aided design programs, but at that time computers were too expensive to make computer-aided design popular in the AEC sector. In 1982, when computers were becoming affordable enough for people to own a personal computer, AutoCAD was released and quickly rose to dominate the fledgling computer-aided design industry. In 1985, the former mathematics professor Samuel Geisberg founded Parametric Technology Corporation. They shipped what would become the first commercially successful parametric software, Pro/ENGINEER, in 1988 [23]. In 2004 was released another parametric program by Gehry Technologies called Digital Projects. While the majority of architecture firms may never use overly parametric software like Pro/ENGINEER or Digital Projects, most of them use parametric equations in some capacity to model their buildings using BIM software such as Revit, Archicad etc.

Parametric modelling has also made its way into the projects through the scripting interfaces of software packages. Scripting interfaces allow designers to write code to automate parts of the software. The developers of software like AutoCAD, even back in 1982 realised that including a scripting interface allowed them to “avoid lots of custom coding and application-specific stuff they would otherwise get asked for”. But textual scripting interface has not developed significantly from the early days of AutoCAD. Instead, the past decade has seen the emergence of a new type of scripting

interface, the visual interface. Visual programming represents the program using visual elements such as diagrams and flowcharts with a drag-and-drop interface. It gives the user the power of programming without any coding skills. The diagrams or flowcharts are created by the components connected with each other by wires. This way the code is more transparent, which helps when working in teams. [23]

Architects got their first visual-scripting language when Robert Aish, then working for Bentley Systems, started quietly beta testing Generative Components with select architecture firms in 2003. Robert McNeel 'I&' Associates, after trying unsuccessfully to license Generative Components, assigned developer David Rutten to make their own version. Released in 2007 as Explicit History, Rutten later dubbed his visual scripting interface Grasshopper. [23] While Grasshopper remains the most widely used visual scripting language in the AEC sector, there are other popular options such as Dynamo for Revit and Allplan Visual Scripting.

The advantages of visual programming over traditional programming include easier learning and usage, code readability, faster development, and simpler collaboration. However, there are some disadvantages as well, such as limitations on what can be created, integration with other software, and mainly slower performance for computationally intensive tasks.

Chapter 2

Architecture

During the initial consultation, my supervisor recommended the "4x4" house by Japanese architect Tadao Ando as a starting point for our project. The objective was to address the issue of expensive land for building houses and explore possible solutions. The "4x4" house is a compact four-story structure with dimensions of 4 by 4 meters, situated on the outskirts of Kobe, Japan, near the Hyogo coast. However, as someone with previous experience in architecture, I decided to create my unique design.



Figure 2.1: "4x4" house by Tadao Ando

2.1 Pre-Design

2.1.1 Project Objectives

The task at hand was to create an attractive design for a small house suitable for a couple without children or a vacation home, with specific dimensions of 8x6 m. For this project, I aimed to identify a distinctive location, and after careful consideration, I chose a parcel on Petřiny Hill in Malá Strana, Prague.

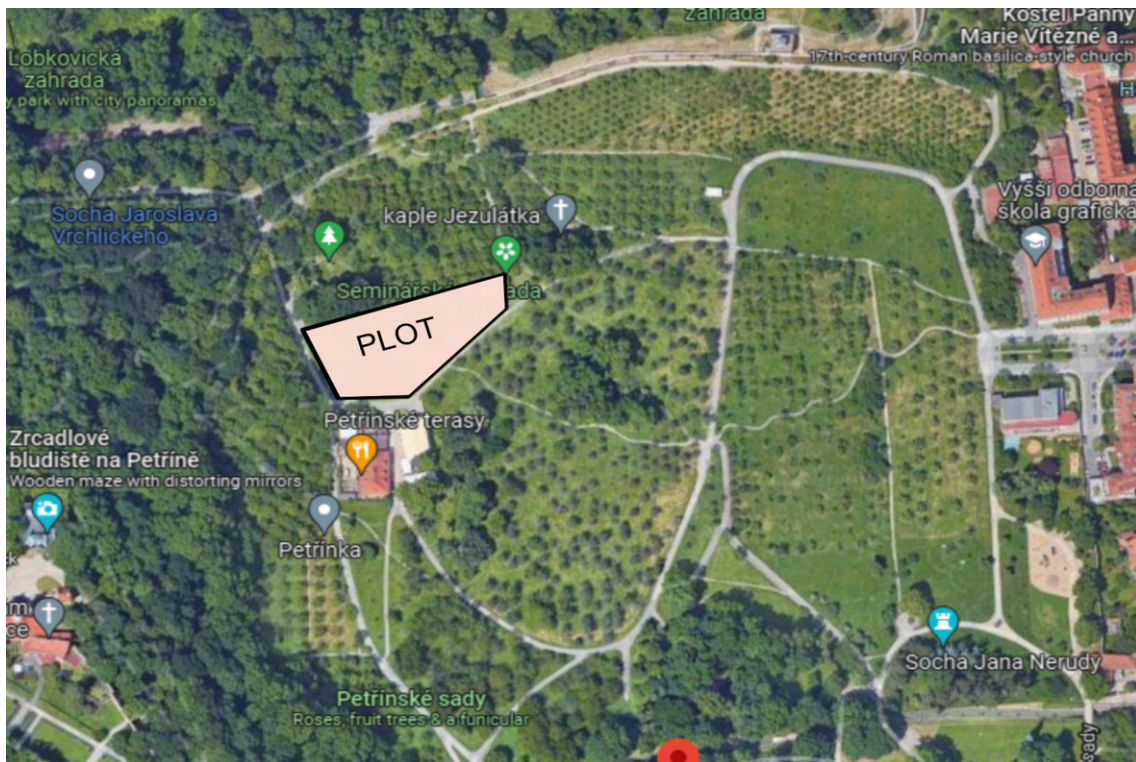


Figure 2.2: Map of Petřiny

2.1.2 Inspiration

The primary goal was to create an aesthetically pleasing structure incorporating architectural and structural elements. The chosen location inspired me to design a structure that complemented the surrounding environment and provided a sense of connection with nature. To begin the design process, I conducted research for inspiration and discovered several remarkable projects. One such project is R128 shown in Figure 2.3 by Werner Sobek, a German architect and structural engineer.



Figure 2.3: R128 by Werner Sobek [22]

R128 is a four-story building situated on a steep site at the edge of the Stuttgart basin. Its integration with the outdoors and the site's similarity to my chosen location provided valuable insights.

Another inspiring project is the House with One Wall by Swiss architect Christian Kerez shown in Figures 2.4 and 2.5.



Figure 2.4: House with One Wall by Christian Kerez [25]

Although relatively small, the clever design creates a spacious interior environment. This innovative approach to spatial organization resonated with the goals of my own design.

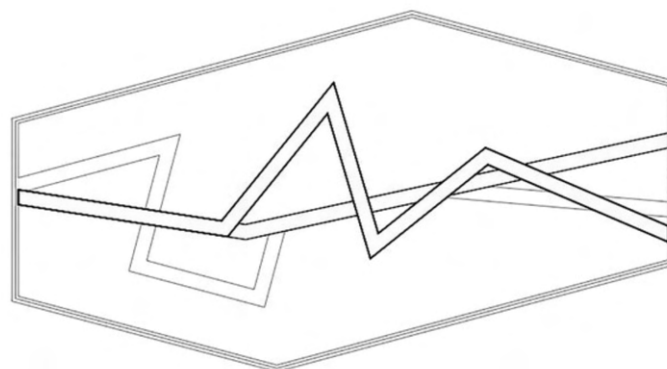


Figure 2.5: House with One Wall Concept [25]

Additionally, the Crossed House by Spanish firm Clavel Arquitectos shown in Figure 2.6 served as a source of massing inspiration.



Figure 2.6: Crossed House by Clavel Arquitectos [27]

The concept behind this project was to orient the lower level of the house towards the intimate garden space while offering panoramic views from the upper level,

considering future developments and solar radiation. The use of cantilevers and the geometric rotation of two stacked elements created an intriguing design that addressed site conditions and maximized views, see Figure 2.7.

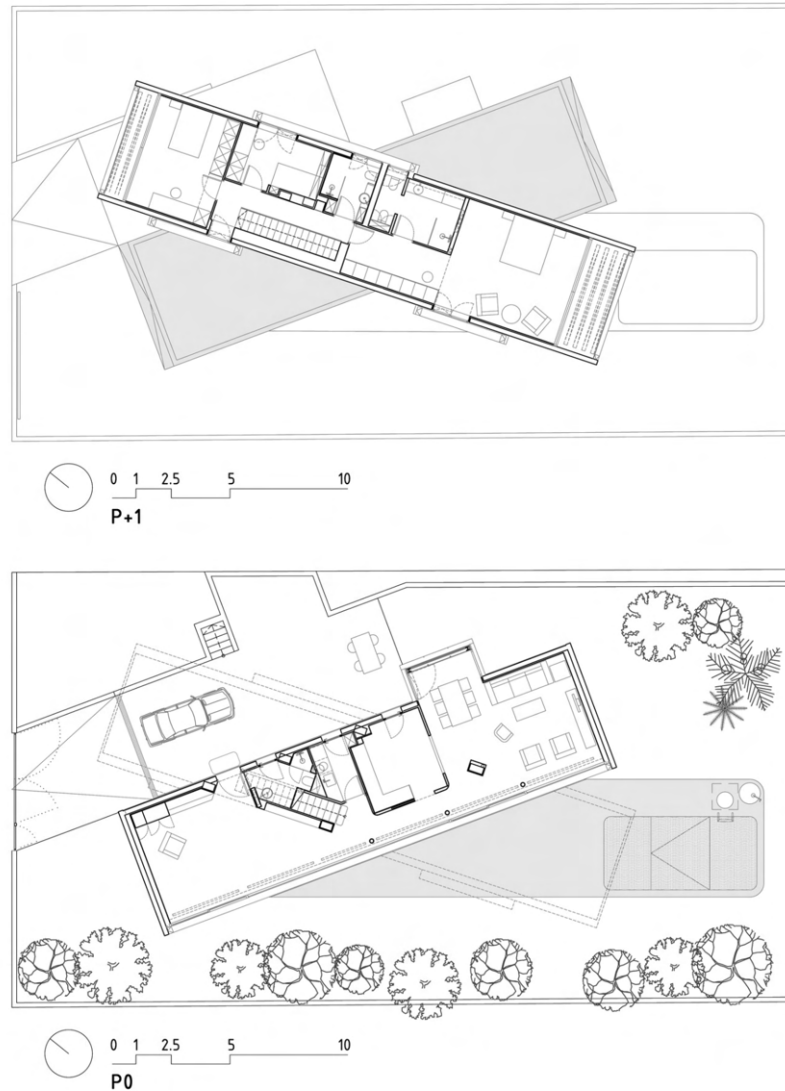


Figure 2.7: Crossed House plan by Clavel Arquitectos [27]

2.2 Petřiny Panorama House: Design Process

The chosen location for the house is situated within the "Seminářská zahrada" garden, and the boundaries of the garden served as a guide for the design of the house's mass. The house was designed to follow these boundaries and blend with the natural landscape, becoming an integral part of it. The ground floor and second floor share the same shape, while the third floor features a long cantilevered tube placed on top of the second floor as illustrated in Figure 2.8.

To fully appreciate the natural surroundings, the design aimed to provide an open layout that maximizes views and natural light. The use of minimal enclosures ensures

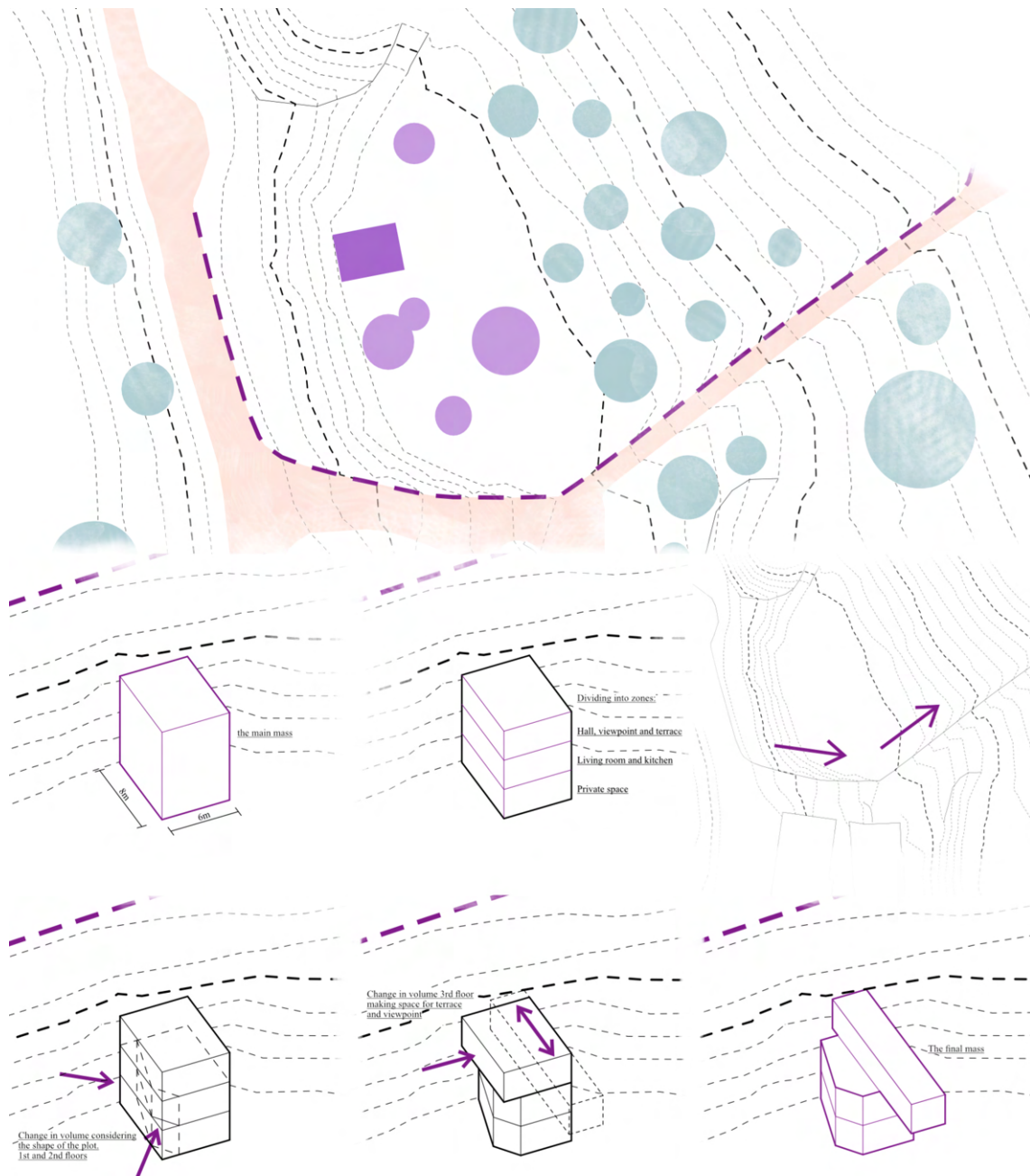


Figure 2.8: The concept of the designed house

that the landscape and outdoor life can be observed from any room in the house. However, to ensure privacy, rooms such as the bathroom and toilet were enclosed, while the kitchen was partially closed as illustrated in Figure 2.9.

The entrance to the house is on the third floor, where a toilet and access to the terrace are also located. The third floor features a long corridor that leads to the end of the cantilever, which serves as a viewpoint of Prague Castle and a relaxation area. The second floor comprises a kitchen and a living room, while the first floor houses a bedroom, bathroom and storage space under the stairs.

In terms of load-bearing structures, the house incorporates reinforced concrete walls

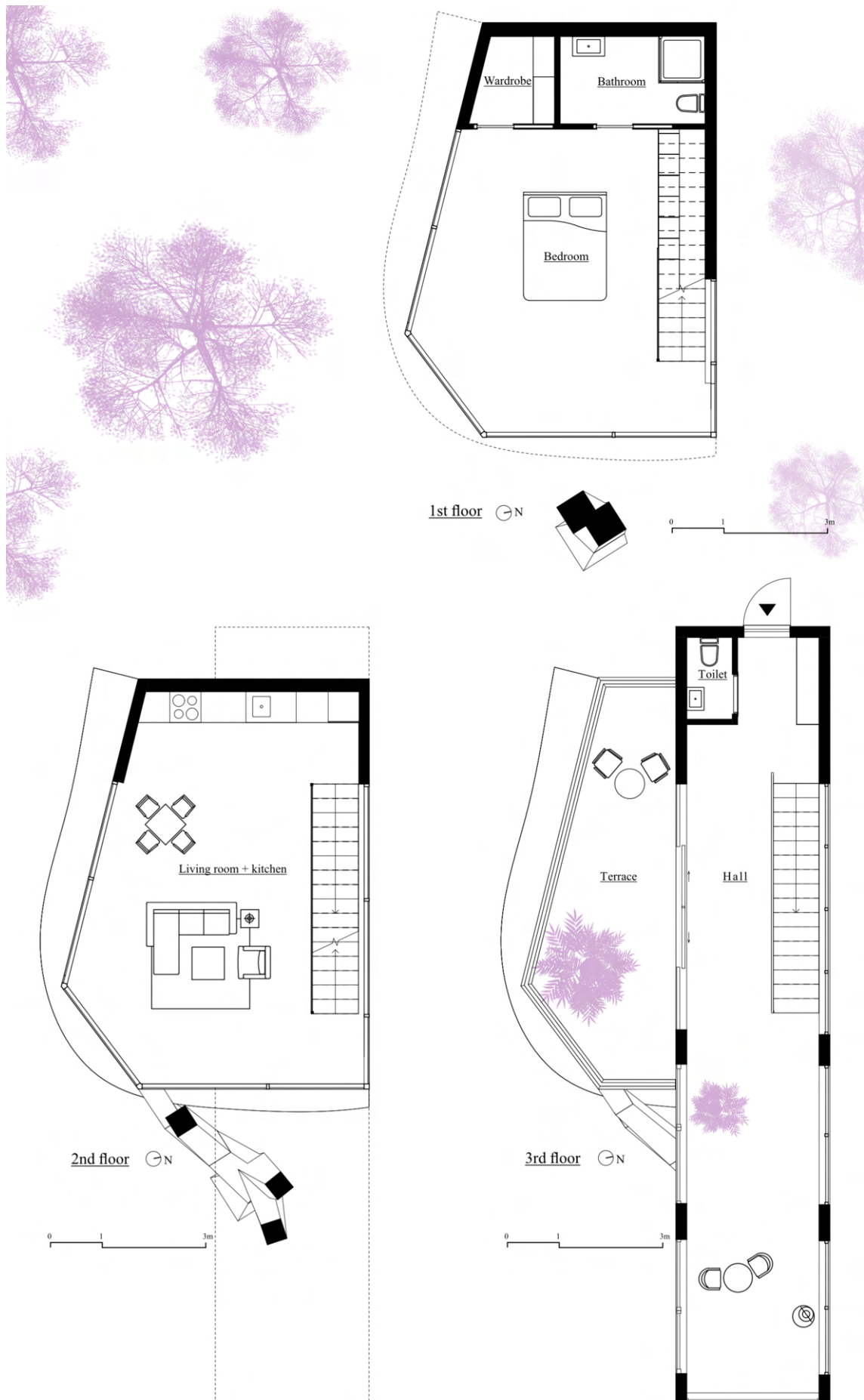


Figure 2.9: The floor plans of the designed house



Figure 2.10: Illustration of tree columns of Sagrada Familia Cathedral designed by Gaudí (in right) and a tree (in left) [21]



Figure 2.11: Model of tree structure columns by Otto Frei [ottotree]

and a unique tree column. To create a connection with nature and the house's surroundings, I drew inspiration from the works of renowned architects like Gaudí and Otto, who utilized tree-like structural elements in their projects, see Figures 2.10 and 2.11.

After exploring various ideas, I discovered inspiration in the N°006 - Gsteig project by the Rohrbach Wehrli Pellegrino Architectural Agency. In this project, a concrete tree column supports the existing and new parts of a school, serving as the central element in the new break hall, see Figure 2.12.

Building upon this concept, I incorporated an external exoskeleton column made of concrete, resembling a tree, to provide both structural support and a distinctive design feature, see Figures 2.13 and 2.14. This approach allowed for more usable indoor space while maintaining a close connection to nature. This tree column represents a key element of my thesis, and significant attention will be devoted to its design and implementation.

Additionally, to enhance comfort and interior well-being, external wooden awnings were integrated into the design, serving as shading devices for the first and second

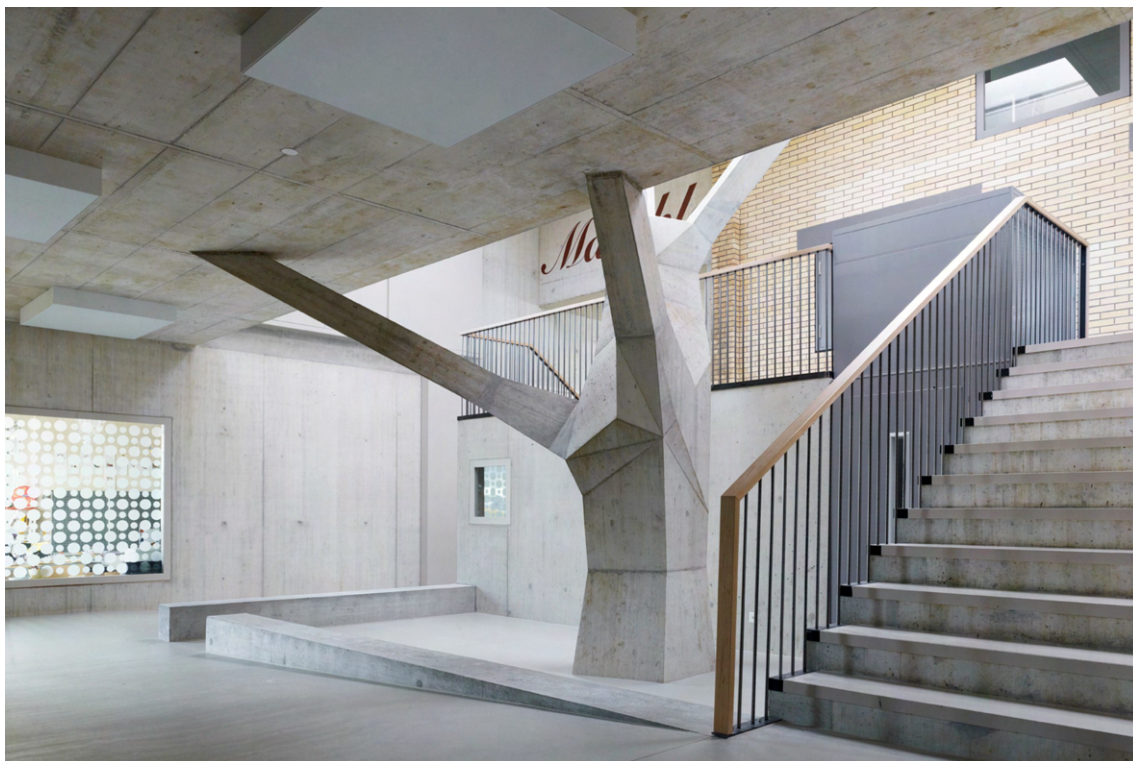


Figure 2.12: Tree Column by RWPA as inspiration for load-bearing structure [20]

floors. These wooden awnings bring a touch of warmth and softness to the overall design while providing the necessary protection from sunlight, see Figure 2.15.

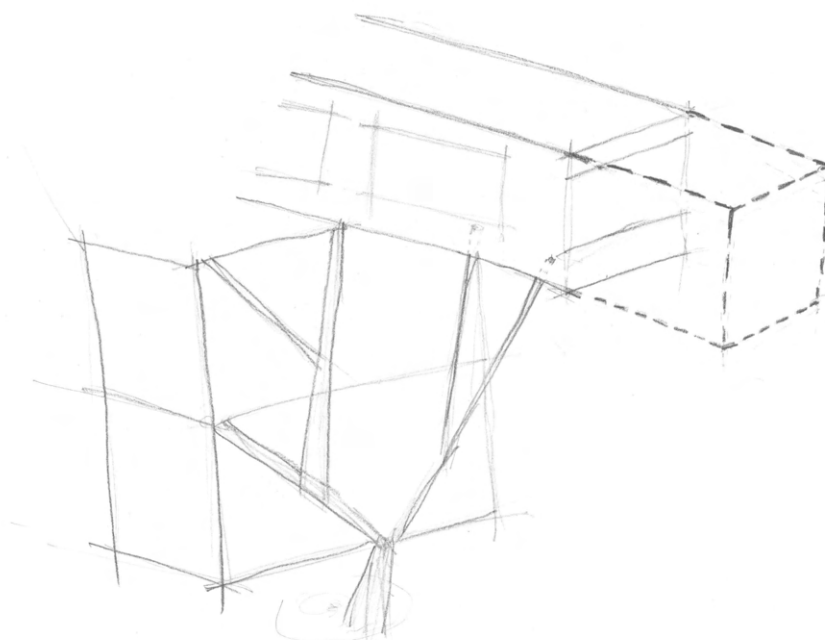


Figure 2.13: Sketch of the designed column

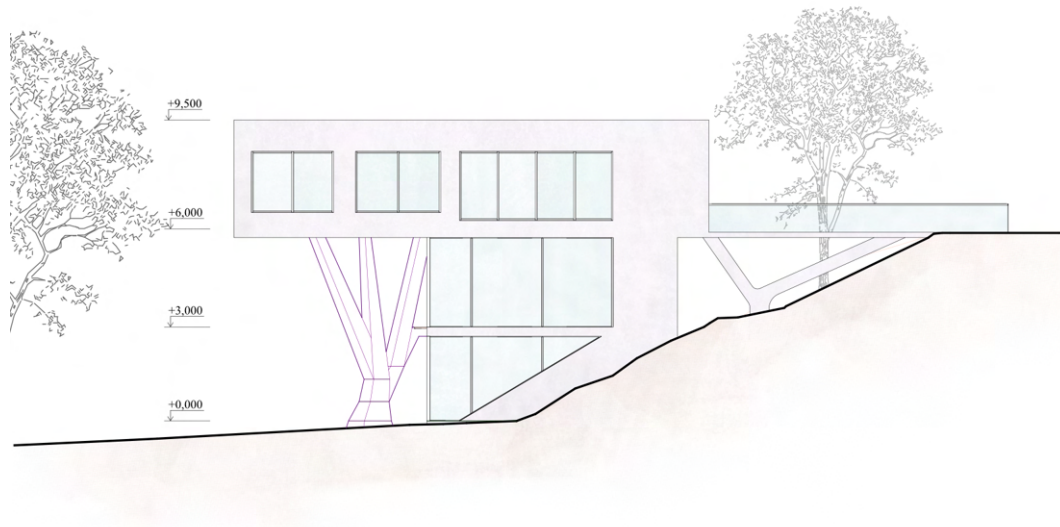


Figure 2.14: North elevation of the designed house

The incorporation of a glass envelope on the first and second floors, as well as the cantilevered structure of the third floor, blurs the boundaries between the interior and exterior spaces, offering panoramic views of the garden from all levels of the house. Notably, the second and third floors provide breathtaking views of Prague and Prague Castle, further enhancing the connection between the house and its prominent location.



Figure 2.15: Visualisation of the designed house (result of the provided optimization)

Chapter 3

Visual Programming

The selection of an appropriate modelling tool plays a crucial role in enhancing the design process. The emergence of visual programming tools has brought a paradigm shift to the field of architectural and structural design, offering flexible and efficient approaches to parametric modelling. This chapter delves into the reasons behind choosing Grasshopper as the preferred tool for this thesis. Through a comprehensive evaluation, this chapter aims to highlight the strengths and capabilities of Grasshopper that make it an ideal choice for improving the early-stage design process.

3.1 Grasshopper

Grasshopper is a robust visual programming language tightly integrated with Rhinoceros 3D, a widely utilized computer-aided design (CAD) software. Its interface is intuitive and user-friendly, making it accessible to architects, designers, and engineers. Additionally, Grasshopper boasts an extensive library of plugins and extensions, rendering it a popular choice. Its flexibility and efficiency empower users to create and manipulate parametric models, facilitating the exploration of various design possibilities and enabling rapid iterations through alternative designs.[26]



Figure 3.1: Chosen software for parametric modelling [26]

A significant advantage of Grasshopper lies in its thriving plugin ecosystem, which expands its core features. These plugins provide specialized tools and components tailored to diverse design domains, including structural and environmental analysis,

optimization algorithms, and fabrication workflows. This diverse range of plugins enhances Grasshopper's capabilities, making it a comprehensive and adaptable tool for achieving the research objectives of this thesis.[26]

Furthermore, the Grasshopper community is vibrant and extensive, encompassing architects, designers, and engineers who actively contribute to knowledge sharing and development. This active community fosters collaboration and provides access to an extensive repository of tutorials, forums, and user-generated components. Leveraging the collective expertise of the Grasshopper community empowers researchers to tap into a wealth of resources and best practices, facilitating the progression of the research and effectively addressing complex design challenges.

3.2 Design Process

Here I would like to describe the process of creating a script for Grasshopper to enhance the early-stage design process. The illustration of the complete definition of the script can be found in Appendix A.

3.2.1 Parametric Model of The House

To facilitate flexibility and accommodate design changes, the house was fully modelled parametrically. Rather than importing the geometry from Rhinoceros, I opted to create the entire house within Grasshopper, enabling dynamic modifications throughout the design process. This approach proved highly advantageous, particularly when some alterations were made to the design of the third-floor walls.

3.2.2 Designing The Tree Column

Three design options for the tree column were created once the main mass of the house was established, see Figure 3.2. Each option underwent testing and optimization to ensure compliance with architectural criteria, with an emphasis on the overall appearance. Following these optimizations, the second option was chosen for further analysis.

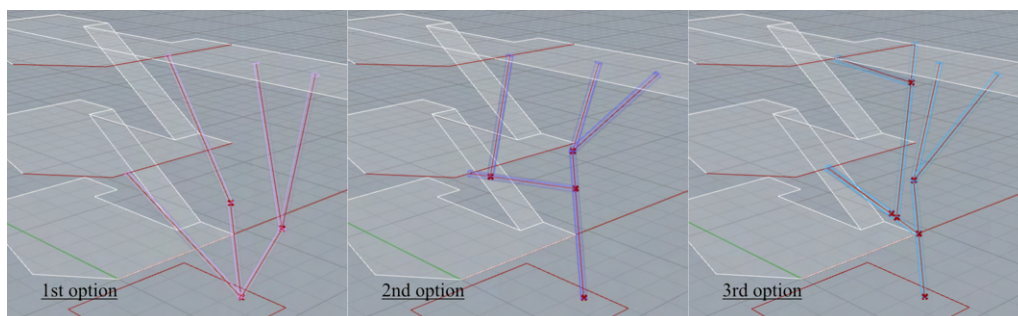


Figure 3.2: Three parametric options of tree column

- The tree column's starting point is located between the slabs, specifically between the first and second-floor slabs and the cantilever. This point's position is not fixed and can be adjusted within a 5.5m x 4m surface area, as illustrated in Figure 3.3. From this starting point, a line is created with a range of minimum length of 1.0m and a maximum length of 3.5 m.
- At some point along this line, a division occurs, leading to the creation of a new line directed towards the first-floor slab, which serves as the first branch of the tree column. The first part of the divided line thereby becomes a trunk of the column and the second part becomes the second branch as shown in Figure 3.3. It is important to note that all lengths within this design are parametric

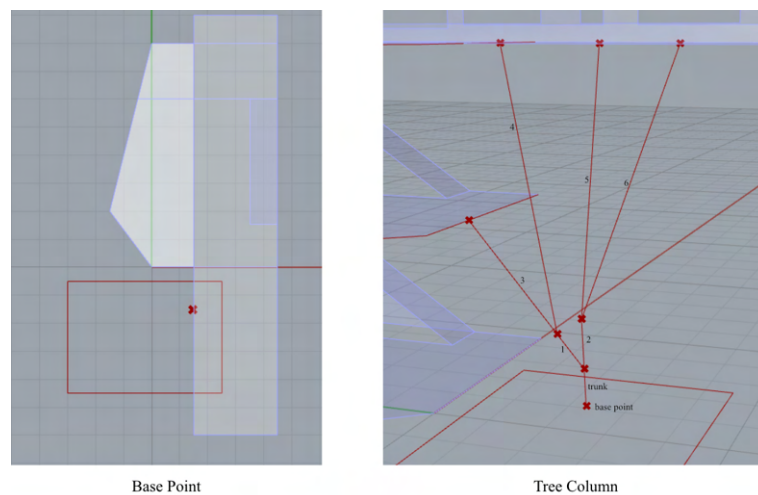


Figure 3.3: Modelling process of chosen tree column

and subject to change.

- The line directed towards the first slab is further divided at a specific point. The resulting segment, connected to the first ceiling, forms the third branch. From this division point, another line is drawn towards the second-floor ceiling, representing the fourth branch. These branches will eventually connect with the front lines of the ceilings, specifically the unsupported sections.

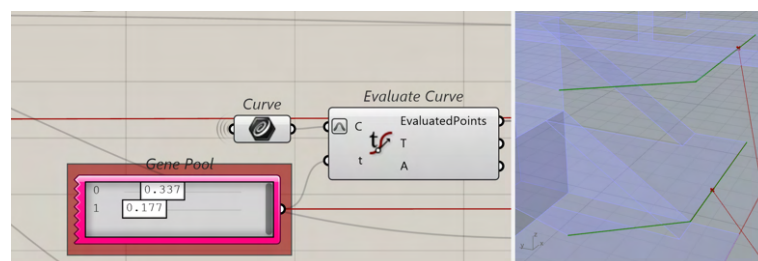


Figure 3.4: "Evaluate Curve" component of Grasshopper

- The supporting points, where the branches connect with the ceilings, are defined using the "Evaluate Curve" component, see Figure 3.4. This component reparametrizes the given curves to a length of 1, allowing for the determination of a point at parameter t on each curve. The position of these points can

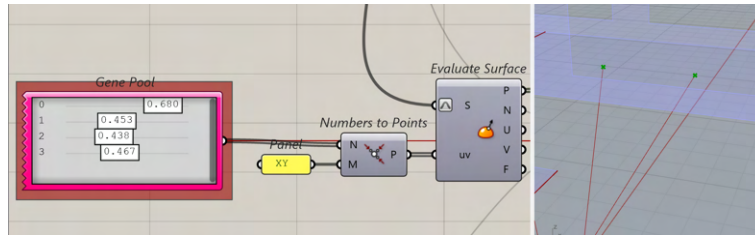


Figure 3.5: "Evaluate Surface" component of Grasshopper

be adjusted using sliders or a "Gene Pool" component. The number of sliders corresponds to the number of given curves.

- From the endpoint of the second branch, two new lines emerge and connect with the cantilever at certain points. The supporting points for the cantilever are determined using the "Evaluate Surface" component shown in Figure 3.5, which is similar to the "Evaluate Curve" component but requires uv coordinates instead.

In Figure 3.6 below I would like to show how the column was changed by changing the parameters.

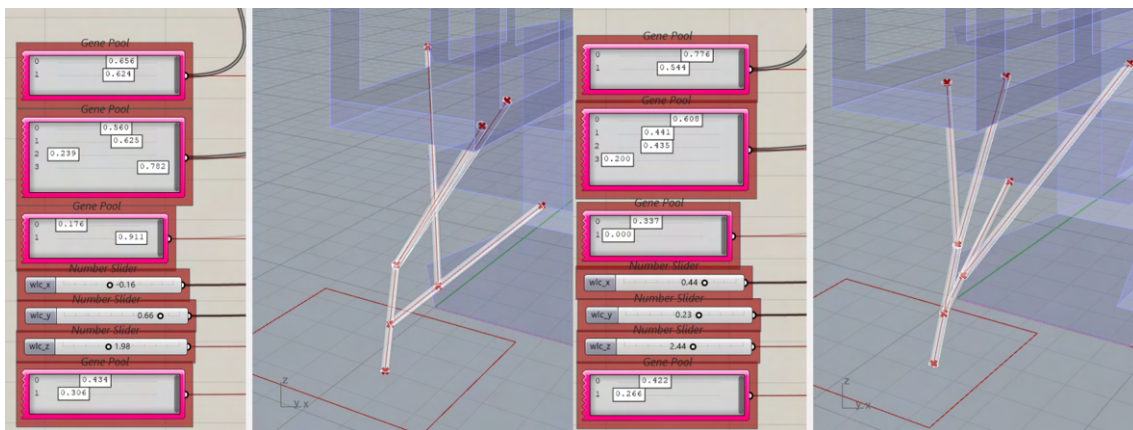


Figure 3.6: Illustration of tree columns with different parameters

3.2.3 Parametric Openings

The third-floor openings are also fully parametric. Both the width and height of these openings can be controlled using sliders, see Figure 3.7. While the width of all openings is adjustable, only the height of the front opening varies. The heads of all the openings align at the same level, ensuring visual coherence. The smaller window openings share a common sill height, while the patio door's sill is positioned lower.

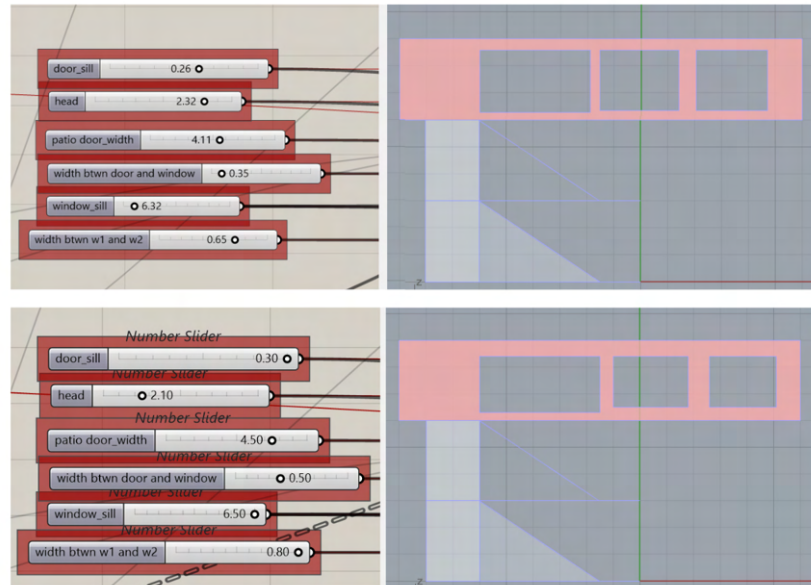


Figure 3.7: Illustration of 3rd floor openings with different parameters

3.2.4 Cantilever Length

Although the cantilever length is inherently parametric after conducting numerous testing optimizations and consulting with my supervisor, a fixed cantilever length of 6 m was chosen for further analysis.

3.2.5 The Awnings

The awnings for the first and second floors were created using the "Graph Mapper" component, utilizing the "Sine" graph type. The shape of the awnings is controlled by three sliders, allowing for easy adjustment and exploration of different configurations, see Figure 3.8. This parametric approach to designing the awnings provides flexibility in defining their form and will help in further analysis to achieve desired aesthetic effects and environmental results.

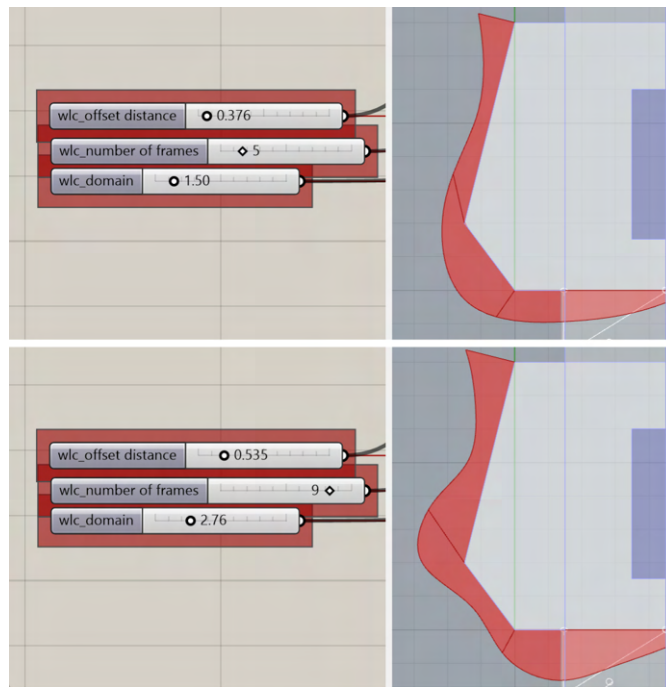


Figure 3.8: Illustration of the awnings with different parameters

Chapter 4

Structural Analysis

Structural analysis plays a crucial role in the design and evaluation of concrete structures. It is essential to ensure structural integrity, strength, and stability while considering architectural constraints and aesthetic requirements. However, the early-stage design process often lacks a comprehensive analysis framework to evaluate structural performance. This poses a challenge as designers require immediate feedback on their design decisions to ensure structural integrity, stability, and efficiency. Traditional methods of conducting structural analysis, such as standalone software, can be time-consuming and disruptive to the design process. Therefore, there is a need for seamless integration of structural analysis capabilities within the parametric modelling environment to facilitate real-time feedback and efficient design optimization.

4.1 The Importance of Structural Analysis in Early-Stage Design

In the context of this master thesis, the integration of structural analysis into the early-stage design process aims to enhance the overall performance and reliability of the concrete structure. By combining parametric modelling, multi-objective optimization, and structural analysis, a comprehensive and iterative design workflow can be established, resulting in an optimized and structurally sound architectural solution. The key challenge is to find fast and efficient FEM software that provides real-time analysis results within the Grasshopper environment, eliminating the need for switching between different software platforms and facilitating a seamless design workflow.

4.2 An Integrated FEM tool within Grasshopper

As the thesis focuses on the early-stage design process, fast and integrated FEM software is required to provide the necessary structural analysis. The chosen tool

must operate within Grasshopper to avoid the time, memory, and energy consumption associated with switching between software platforms. Additionally, the selected plugin should deliver real-time and fast results to support the multi-objective optimization process. Several FEM tools have been developed to provide structural analysis capabilities within the Grasshopper environment. These tools enable designers to perform various types of analyses, including linear static analysis, buckling analysis, form finding, and optimization. Alpaca, Kiwi3D, and Karamba are notable examples of FEM tools designed specifically for Grasshopper. Alpaca and Kiwi3D offer powerful structural analysis capabilities within the Grasshopper framework. They provide essential features such as geometry creation, meshing, material assignment, load application, and result visualization. These tools enable designers to perform structural analysis directly within Grasshopper, facilitating a more integrated design workflow. However, while Alpaca [19] and Kiwi3D [24] offer valuable analysis functionalities, they may lack certain advanced features required for more complex design scenarios.

4.2.1 Karamba 3D

Among the available FEM tools for Grasshopper, Karamba 3D emerged as the optimal solution for enhancing the early-stage design process of concrete structures. Karamba 3D [7] offers advanced structural analysis features that seamlessly integrate with Grasshopper. The plugin provides a wide range of analysis capabilities, including linear static analysis, buckling analysis, form finding, and optimization. Its direct manipulation of geometry and structural properties within Grasshopper ensures a streamlined design and analysis workflow.

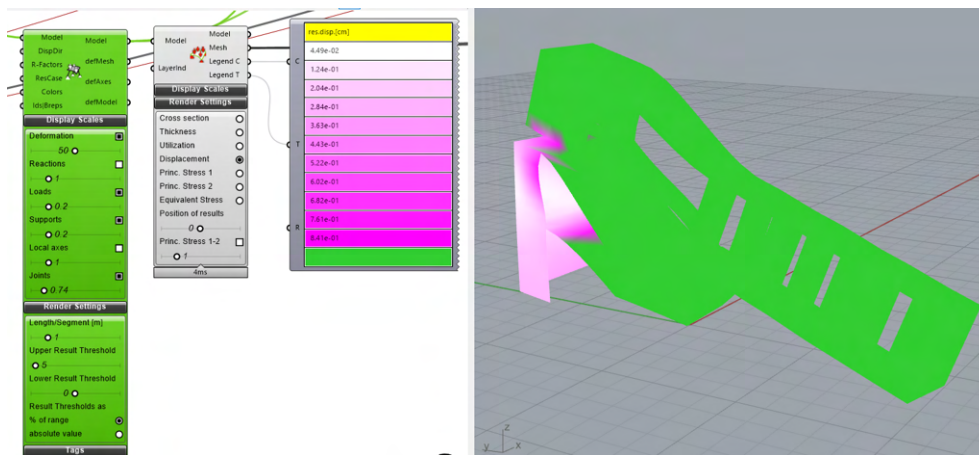


Figure 4.1: Karamba 3D - visualization of deflection of a model

The key advantage of the Karamba 3D plugin lies in its ability to perform real-time structural analysis, providing designers and engineers with instant feedback on design iterations. This capability enables the exploration of various design alternatives and quick evaluation of their structural performance. Furthermore, Karamba 3D offers comprehensive visualization tools for assessing structural behaviour, including stress distribution, deformation, and mode shapes. In Figure 4.1 is shown a deformed

model of the house without a supporting column, where the maximum deformation is set to 5 cm and the display scale to 50.

4.3 The FEM Model

For the structural analysis, the Grasshopper model containing lines and surfaces needed to be converted into a Karamba 3D model, representing beams and slabs. The chosen material for the structures is concrete C30/37.

4.3.1 Slabs

The thickness of the walls ranged from 120 mm to 250 mm, the floors ranged from 180 mm to 300 mm, and the stair slab ranged from 100 mm to 200 mm. It is important to note that the dimensions obtained after optimization need to be rounded, taking into consideration formworks in further design stages.

A crucial step in the analysis of slabs using Karamba 3D is the transformation of surfaces or breps (Boundary Representations) into meshes. To achieve this, the "Mesh Breps" component provided by Karamba 3D is utilized. This component ensures the connectedness of meshes generated from multiple breps and allows the definition of points on those breps where mesh vertices should be generated. For further description see Figure 4.2

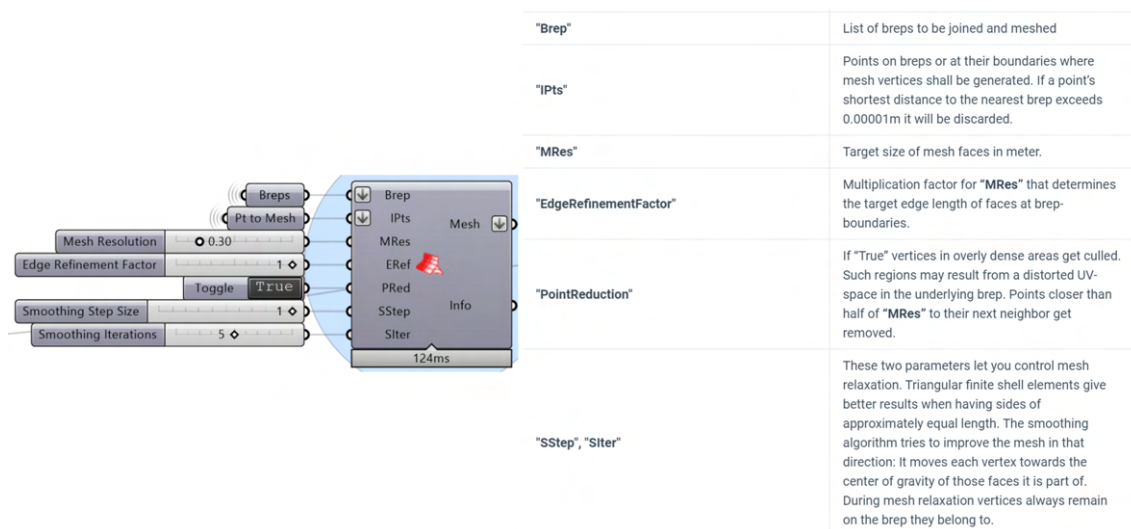


Figure 4.2: "Mesh Breps" component of Karamba 3D [7]

4.3.2 Tree Column

In the case of the tree column, each branch is divided into five segments. The first and last segments have specific cross-section dimension ranges, while the dimensions of the intermediate segments are evenly interpolated based on the branches' edge cross-sections. To determine the appropriate dimensions for further analysis, cross-sections with dimensions of the dimensions shown in Table 4.1 were analyzed using the IDEA Statica software. With a focus on safety, the degree of reinforcement for these cross-sections was chosen as $\rho = 0.015$. The moments obtained from the N-N-M Capacity Diagram, see Figure 4.3 were compared with the moments calculated by Karamba 3D.

Tree Column Cross-Sectional Dimensions	
Column Part	Dimensions (mm)
trunk bottom	800 to 1000
trunk top	600 to 800
branches 1 and 2 bottom	600 to 800
branches 1 and 2 top	400 to 600
branches 3,4,5 and 6 bottom	400 to 600
branches 3,4,5 and 6 top	250 to 400

Table 4.1: Tree column cross-sectional dimensions

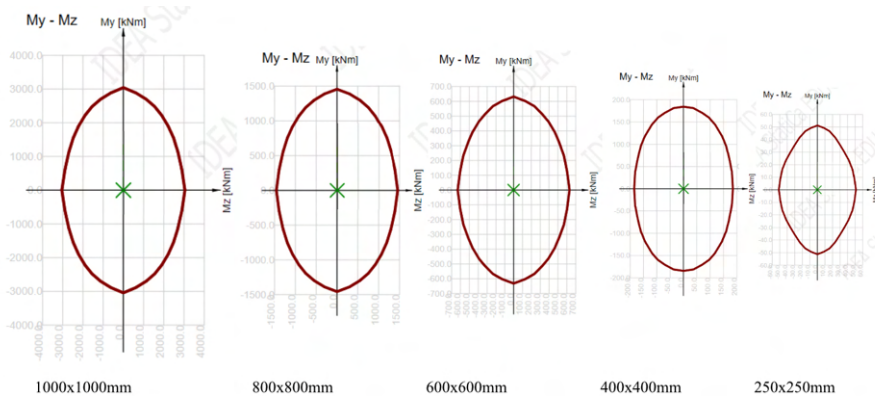


Figure 4.3: N-M-M capacity diagrams of chosen cross-sectional dimensions

4.3.3 Loads

For the purposes of this thesis, the structural analysis focused on three main load conditions: dead load, permanent loads, and live loads, see Table 4.2. The dead load, which refers to the weight of the structure itself and any fixed elements, was automatically calculated. As this analysis was conducted in the early stages of design

where specific horizontal compositions were not yet known, estimated values were used for the permanent loads. The live loads, which account for the variable loads imposed on the structure during its intended use, were determined according to the national annex to CSN EN 1991-1-1 [3].

Load Conditions		
LC1 - dead load	Calculated automatically	-
LC2 - permanent loads	floor composition	$g_{1,k} = 1.5 \text{ kN/m}^2$
	roof and terrace composition	$g_{2,k} = 1.35 \text{ kN/m}^2$
LC3 - live loads	floor slab (category A)	$q_{1,k} = 2 \text{ kN/m}^2$
	terrace slab (category A)	$q_{2,k} = 2.5 \text{ kN/m}^2$
	stair slab (category A)	$q_{3,k} = 2.0 \text{ kN/m}^2$
	roof slab (category H)	$q_{4,k} = 0.75 \text{ kN/m}^2$

Table 4.2: Load conditions

4.4 Setting the Structural Criteria

To guide the optimization process, six criteria were chosen to ensure structural integrity and stability:

1. Deflection of the column
2. Stress on the 3rd-floor walls
3. Deflection of the model at end-points and midpoints of elements
4. Deflection of the 1st and 2nd-floor slabs(without cantilever)
5. Weight of the column
6. Weight of the slabs

Considering these six criteria individually would be inconvenient and may result in the extraction of unfeasible solutions. Therefore, a weighted average approach was chosen. Before applying the weighting operation, it was necessary to normalize the criteria values. The Grasshopper component "Remap" was used for this purpose as shown in Figure 4.4. This component requires the value to be remapped, a source domain, and a target domain. For example, the second criterion, stress on the 3rd-floor walls, was remapped. The "Result Vectors on Shells" component was used to extract the tensile stress on the walls. Taking into account the properties of the concrete, only tensile stress was considered. As Karamba 3D provides linear static analysis, adjustments were made for this calculation. The mean tensile strength of

the concrete used is 2.9 MPa, the design yield strength of reinforcement is 435 MPa and for reinforcement was chosen a ratio of $\rho = 0.015$. From this expression $435 * \rho$ the maximum tensile strength of 6.525 MPa was determined. The source domain was set from 0 to 6.525 to accommodate the stress values, while the target domain for all criteria was set from 0 to 1 inside the component itself.

After the remapping process, the criteria values underwent further refinement using the "Graph Mapper" component with a "Bezier" graph type. This additional step served to enhance the quality of favourable solutions while amplifying the deficiencies of suboptimal solutions. This approach proved particularly advantageous during the extraction of solutions following the optimization phase. Figure 4.4 provides a visual representation of the changes in values achieved through the utilization of the "Graph Mapper" component. By observing the alterations depicted in the graph, the influence of the mapping process on the criteria values becomes evident, ultimately playing a significant role in facilitating the optimization of the analysis. The same

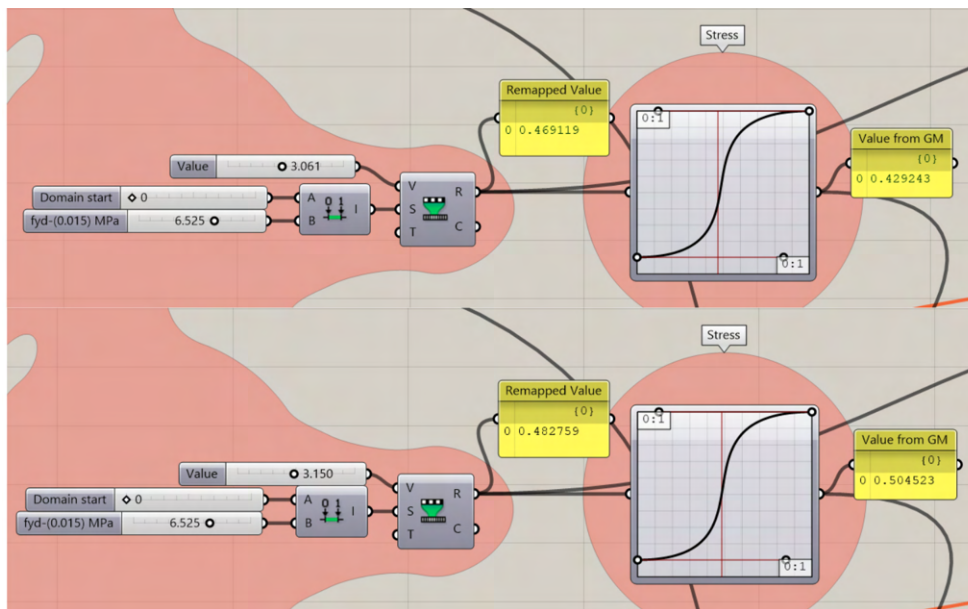


Figure 4.4: Normalization of stress value using Grasshopper components

process of normalization and mapping was repeated for all criteria, ensuring that each criterion was appropriately prepared for further analysis. Once the normalization was complete, a weighted average operation was applied to obtain a single value representing the overall performance of the structure, which would be used during the optimization phase. The weights assigned to the criteria were carefully determined as shown in Table 4.3:

After performing the weighted average operation, the resulting value represents a comprehensive evaluation of the structure's performance based on the selected criteria. This single value facilitates the comparison and ranking of different design alternatives during the optimization process.

Criteria	Weights
Deflection of the column	4
Stress on the 3rd-floor walls	2
Deflection of the model	5
Deflection of the 1st and 2nd-floor slabs	5
Weight of the column	3
Weight of the slabs	4

Table 4.3: Weights of each structural criteria for "Weighted Average" operation

Chapter 5

Environmental Analysis for Indoor Comfort

In the field of architectural design, the well-being and comfort of occupants are paramount considerations. Indoor comfort encompasses various factors, including temperature, air quality, acoustics, and lighting conditions. This chapter highlights the significance of conducting a comprehensive environmental analysis to ensure optimal indoor comfort. Specifically, it focuses on the assessment of daylight, the view out of windows, access to sunlight, and the prevention of glare. The analysis is conducted following the guidelines outlined in EN 17037, a European standard that provides a standardized approach to evaluating indoor environmental quality.

5.1 Importance of Environmental Analysis for Indoor Comfort

Environmental analysis plays a crucial role in architectural design for several compelling reasons. Firstly, it prioritizes the well-being and comfort of occupants by providing favourable indoor environmental conditions. By evaluating factors such as natural light, pleasing views, access to sunlight, and glare prevention, designers can create spaces that promote occupant satisfaction and productivity. Secondly, the environmental analysis contributes to energy efficiency by reducing the reliance on artificial lighting and minimizing thermal loads. Additionally, it fosters sustainability by optimizing the utilization of natural resources and reducing environmental impacts. By integrating environmental analysis into the design process, architects can create buildings that prioritize occupant comfort, energy efficiency, and environmental responsibility. The European standard EN 17037, also known as "Daylight of Buildings," has introduced new possibilities for assessment and serves as a valuable guideline for architects. This standard, implemented in the Czech Republic since August 2019 as ČSN EN 17037 "Daylighting of Buildings," not only addresses daylighting but also includes requirements for access to sunlight, the view from windows, and the prevention of glare. By following this standard, architects can adhere to best practices and ensure compliance with regulations, leading to improved indoor

environmental quality. [17]

5.1.1 Daylight Analysis

Daylight is a fundamental element of building design, significantly influencing visual comfort and the overall well-being of occupants. The assessment of daylight involves a comprehensive analysis of its availability, distribution, and quality within the building. Designers consider various factors, including window size, orientation, shading devices, and interior finishes. Through simulations and calculations, metrics such as the daylight factor are determined, representing the amount of natural light reaching specific points. By optimizing the design to maximize natural light penetration, architects can minimize reliance on artificial lighting, reduce energy consumption, and create visually comfortable and inviting indoor spaces. [11]

5.1.2 Assessing the View Out of Windows

A pleasing view out of windows has a positive impact on occupant satisfaction and well-being. It establishes a connection to the external environment, providing visual stimulation and a sense of openness. The assessment of the view out of windows involves evaluating the visual quality, including the presence of vegetation, landmarks, or natural elements. Window placement is carefully considered to optimize views while maintaining privacy. By incorporating visually appealing views that harmonize with the surroundings, architects can create spaces that enhance occupants' experiences and foster a deep connection with nature. [11]

5.1.3 Access to Sunlight

Access to sunlight is vital for both visual comfort and physiological well-being. Sunlight provides natural illumination, warmth, and essential vitamin D. The analysis of access to sunlight involves evaluating the availability of direct sunlight and potential obstructions such as neighbouring buildings or vegetation. Architects strategically locate windows and design building forms that allow for sufficient sunlight penetration, reducing reliance on artificial lighting and creating spaces that are naturally illuminated. This approach creates a more pleasant and inviting atmosphere while promoting energy efficiency and occupant well-being. [11]

5.1.4 Prevention of Glare

Glare occurs when there is excessive contrast between light and dark areas, leading to visual discomfort and decreased visual performance. The prevention of glare involves analyzing the building's design, including window placement, size, orientation, and the use of shading devices. By effectively controlling the amount and distribution of daylight entering the space, architects can minimize glare and ensure optimal

visual comfort for occupants. This analysis contributes to the creation of visually comfortable spaces that are free from distracting glare, promoting productivity and overall well-being. [11]

5.2 Process of Integrated Environmental Analysis for Optimized Design

For the purpose of this thesis, three environmental criteria were selected for analysis: solar radiation for summer and winter, view out, and sunlight hours. To perform this analysis, a powerful and comprehensive tool was required—one that could integrate seamlessly with Grasshopper and efficiently provide the necessary information for the optimization process.

5.2.1 Integration of Comprehensive Analysis Tool

After careful consideration, the Ladybug Tools platform was chosen for this task. Originally developed as a plugin for Grasshopper, Ladybug Tools shown in Figure 5.1 offers a range of capabilities that perfectly align with the analysis requirements of this thesis. The tool is built on top of several validated simulation engines, including Radiance, EnergyPlus/OpenStudio, Therm/Window, and OpenFOAM [8]. What sets Ladybug Tools apart is its unique position as the only open-source interface that unites all of these underlying open-source engines. Moreover, Ladybug Tools continuously evolves through the consensus of an open community of experts, ensuring its accuracy and relevance.

5.2.2 Ladybug - Functionalities and Benefits

Ladybug Tools is supported by a passionate and diverse community from around the world, with an active online forum where discussions shape its future development. The tool operates seamlessly within 3D modelling software, allowing for streamlined data transfer between its simulation engines. With Ladybug Tools, all aspects of geometry creation, simulation, and visualization can be performed within a single interface, eliminating the need for multiple software platforms and enhancing efficiency. [8]

One of the key advantages of Ladybug Tools is its modular component-based structure. This flexibility enables its usage across different stages of design and caters to a variety of research questions. By harnessing the capabilities of CAD interfaces, Ladybug Tools empowers designers to explore design spaces, automate tasks, and generate interactive 3D graphics, animations, and data visualizations. This integration of parametric visual scripting interfaces with Ladybug Tools not only enhances the analysis process but also provides a dynamic and interactive environment for evaluating and optimizing design solutions. [8]

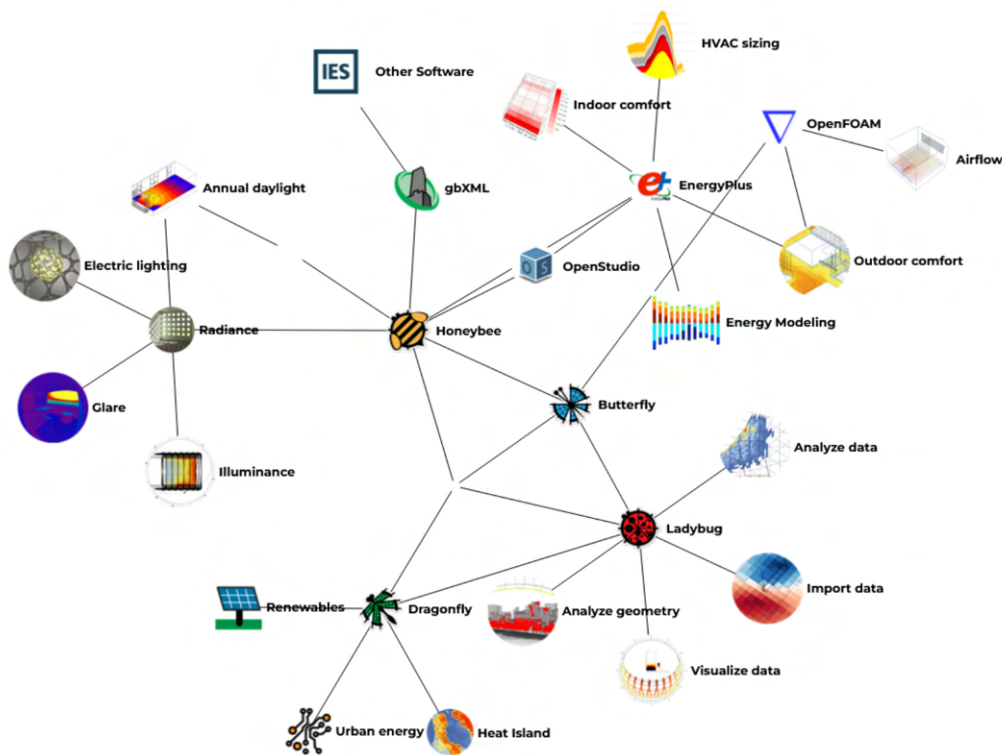


Figure 5.1: Ladybug Tools for environmental analysis [8]

5.2.3 Environmental Analysis Setup

To conduct the environmental analysis for the analyzed house located in Prague as was mentioned in 2.1.1, the weather data specific to this location was obtained using the Ladybug component "EPWmap." This component proved to be invaluable as it eliminates the need to search for weather data from external sources, streamlining the analysis process.

The environmental analysis was performed for all floors of the house, taking into consideration the surrounding natural elements, including trees. Although some trees may need to be removed during the construction phase, they will be replaced or reintroduced strategically to maximize their benefits for the users. The design of trees was facilitated using the Lands Design plugin, considering seasonal variations separately for summer and winter radiation. Additionally, the presence of trees can help with shading, enhancing the overall comfort of the indoor spaces.

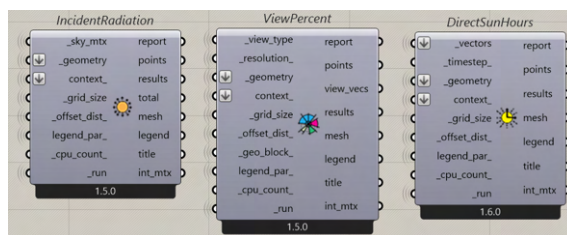


Figure 5.2: Ladybug Tools components used for environmental analysis

Solar Radiation Analysis

Solar radiation analysis was conducted for both summer and winter seasons, as the desired outcomes differ for each period, see Figures 5.3 and 5.4. In summer, the aim is to reduce solar radiation indoors, while in winter, the objective is to increase solar radiation for warmth. The calculations were performed using the "Incident Radiance" component, see Figure 5.2. It is worth mentioning the "_grid size" node of this component, which determines the size of grid cells used to subdivide the geometry for incident radiation analysis. Optimal grid size selection is crucial, as it affects the resolution and computation time. A grid size of 1.2 was set for this analysis, as results for smaller grid sizes did not significantly differ, but the larger grid size expedited the calculation process.

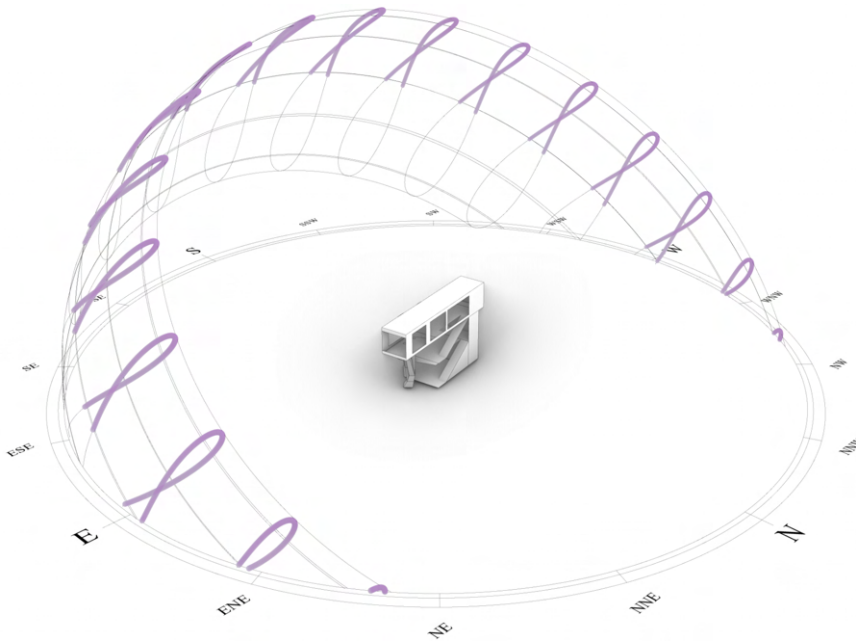


Figure 5.3: Position of the sun in summer

View-Out Analysis

The assessment of the view out of windows was conducted using the "View Percent" component, shown in Figure 5.2, with the view type 1 - "Horizontal30DegreeOffset." This view type represents the percentage of the 360-degree horizontal view band, bounded on top and bottom by a 30-degree offset from the horizontal plane. The grid size for this analysis was set to 0.5, as larger grid sizes yielded less favourable results, compromising the overall view quality.

Sunlight Analysis

For the sunlight analysis, the "Direct Sun Hours" component, shown in Figure 5.2 was employed. The analysis was performed within the period specified by the national

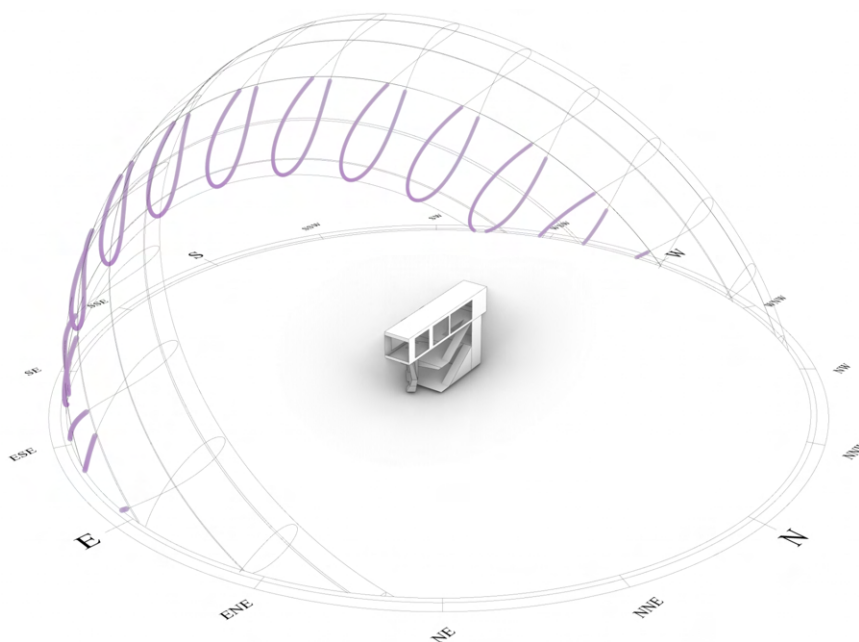


Figure 5.4: Position of the sun in winter

standard ČSN EN 17037, ranging from the 1st of February to the 21st of March [17], shown in Figure 5.5. To obtain meaningful results, a grid size of 0.8 was chosen for this component.

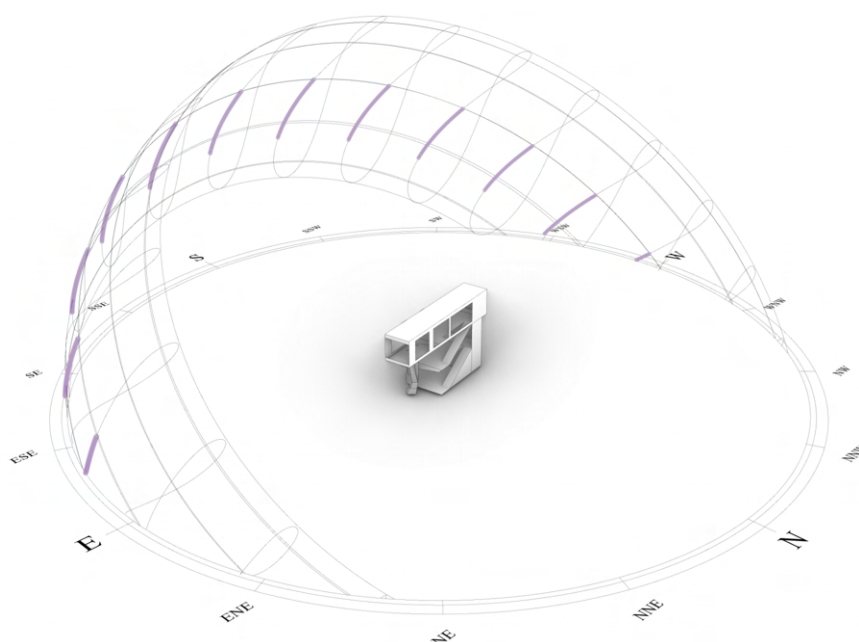


Figure 5.5: Position of the sun in period from the 1st of February to the 21st of March

5.3 Preparing Environmental Criteria for Optimization

In order to prepare the selected environmental criteria for further analysis, a similar approach was applied to the structural criteria. The first step involved extracting average values from the aforementioned components mentioned above. These values were then normalized using the "Remap" component and the "Graph Mapper" component with a parabolic graph type.

The choice of using a parabolic graph type was driven by the objective of extracting optimal values for environmental optimization that lie between the extremes of minimum and maximum values. For instance, considering the recommended sun exposure hours in EN 17037 [11], the minimum value is 1.5 hours, while the maximum value is 4 hours. By utilizing a parabolic graph, the analysis yields values within this inclusive range, representing desirable outcomes., see Figure 5.6.

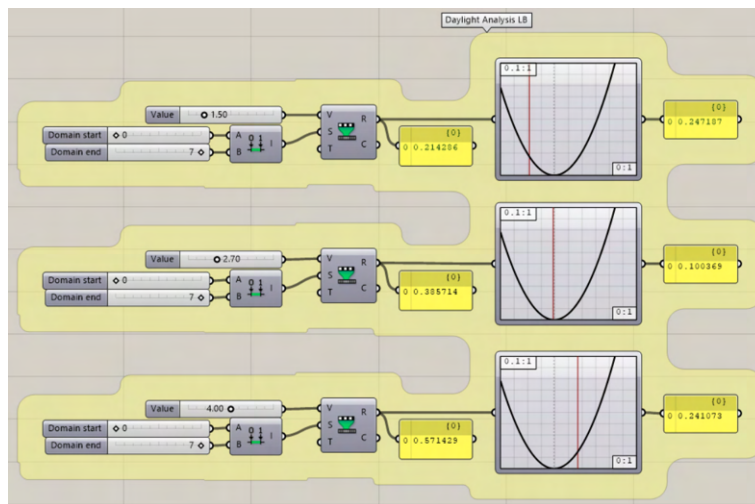


Figure 5.6: Normalization of sunlight value using Grasshopper components

Subsequently, a weighted operation was conducted to combine the criteria effectively. The weights assigned to the respective criteria are shown in Table 5.1.

Criteria	Weights
Summer Radiation	3
Winter Radiations	5
View-Out	5
Sunlight	4

Table 5.1: Weights of each environmental criteria for "Weighted Average" operation

These weights reflect the relative importance assigned to each criterion and will be utilized in the optimization process to achieve a balanced and optimal environmental

design.

Chapter 6

Multi-objective Optimization

In the realm of architectural design, optimizing the performance of buildings is a complex task that requires the analysis and optimization of multiple criteria. Buildings are multifaceted structures where various aspects need to be considered and improved to achieve an optimal outcome. In addition to architectural aspects, such as form, aesthetics, and spatial layout, other key criteria for building optimization include structural integrity, environmental sustainability, and economic viability.

Architectural aspects encompass the visual appeal, functionality, and usability of a building, ensuring that it meets the design goals and satisfies the occupants' needs. Structural optimization focuses on enhancing the structural integrity, strength, and safety of the building, while also minimizing material usage and construction costs. Environmental optimization aims to create energy-efficient, sustainable, and environmentally friendly buildings that reduce resource consumption and minimize negative impacts on the environment. Finally, economic optimization seeks to achieve cost-effectiveness and financial viability by optimizing construction, maintenance, and operational expenses throughout the building's lifecycle.

Considering the interdependencies and trade-offs among these diverse criteria, it becomes evident that a comprehensive optimization approach is necessary to address the complexity and multi-dimensionality of building design. Traditional single-objective optimization methods are limited in their ability to account for multiple conflicting objectives simultaneously. Therefore, the need arises for multi-objective optimization techniques that can handle the inherent complexity and trade-offs associated with building performance.

6.1 Multi-objective Optimization Methods

Multi-objective optimization refers to the process of optimizing multiple conflicting objectives simultaneously, with the goal of finding a set of solutions that represents a trade-off between these objectives (i.e., solutions in which one objective cannot be improved without worsening another one). Such solutions constitute the Pareto optimal set, and the image of this set (i.e., the corresponding objective function

values) forms the so-called Pareto front. [15] Traditional mathematical programming techniques for multi-objective optimization have limitations, including sensitivity to the shape and continuity of the Pareto front and the generation of only one element of the Pareto optimal set per algorithmic execution. Additionally, some mathematical programming techniques require that the objective functions and the constraints are provided in algebraic form and in many real-world problems we can only obtain such values from a simulator. These limitations have motivated the use of alternative approaches, from which metaheuristics have been a very popular choice, mainly because of their flexibility (i.e., they require little domain-specific information) and their ease of use. From the many metaheuristics currently available, evolutionary algorithms have certainly been the most popular in the last few years in this area, giving rise to a field now known as evolutionary multiobjective optimization (EMO). [15]

Evolutionary optimization algorithms imitate the process of natural selection and evolution, where a population of potential solutions evolves over generations, gradually improving and adapting to meet the requirements of a problem. These algorithms utilize genetic operators, including selection, crossover, and mutation, to create offspring solutions from the existing population. The newly generated solutions are then evaluated based on their fitness, determining their ability to survive and thrive within the search space. [4]

In genetic algorithms, a population consists of chromosomes, also known as individuals, which represent potential solutions to the problem at hand. A chromosome or individual is typically represented as a string of characters or numbers. Figure 6.1 illustrates a simple chromosome structure composed of binary genes with values of '0' or '1'. Within a population, certain chromosomes will perform better than others based on the specific problem or environment. Hence, each chromosome is assigned a fitness score that assesses its performance relative to the problem being solved. [5]

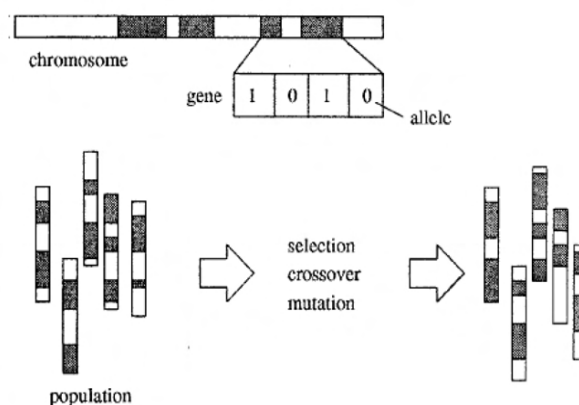


Figure 6.1: Basic components of genetic algorithm [5]

Additionally, Figure 6.1 depicts the reproduction process of a genetic algorithm, which involves selection, crossover, and mutation operators. In this reproduction process, highly fit chromosomes or individuals from the population are selected (selection), combined through recombination (crossover), and subjected to partial manipulation (mutation) to generate an evolved population of potential solutions.

The process of creating a new population from the original population is known as a generation, and it is repeated iteratively. As the number of generations increases, the evolving population will progressively incorporate a higher proportion of characteristics possessed by well-performing individuals from previous generations. Throughout the evolution, individuals with poor evaluations are gradually phased out. The algorithm focuses on exploring the most promising areas of the search space by favouring the mating of more fit individuals. A simple genetic algorithm can be summarized as follows:

1. Random initialization of the population
2. Calculation of fitness for each individual
3. Selection of individuals for reproduction
4. Execution of genetic operators (recombination/mutation) to generate offspring

Early approaches in EMO, such as the Vector Evaluated Genetic Algorithm (VEGA), subdivided the population into subpopulations based on the number of objectives and selected solutions based on their performance on individual objectives. Linear aggregating functions and lexicographic ordering were also common approaches but had limitations in dealing with non-convex Pareto fronts. [15]

Pareto ranking, introduced by Goldberg, addressed the issue of diversity in MOEAs by ranking solutions based on Pareto optimality. This led to the development of the Multi-Objective Genetic Algorithm (MOGA) by Fonseca and Fleming and the Nondominated Sorting Genetic Algorithm (NSGA) by Srinivas and Deb. NSGA used a more computationally expensive approach of creating several layers of solutions through re-ranking the population. [15]

To maintain diversity, density estimators were incorporated, such as fitness sharing used in MOGA and NSGA. Other density estimators, including clustering, adaptive grids, crowding, entropy, and parallel coordinates, have been proposed. Elitism, which retains the best solutions, was introduced to prevent the dilution of selection pressure in MOEAs. The Strength Pareto Evolutionary Algorithm (SPEA) by Zitzler incorporated an external archive to store nondominated solutions and pruned it once a limit was reached using clustering. [15]

Among the Pareto-based MOEAs, the Nondominated Sorting Genetic Algorithm-II (NSGA-II) has gained significant popularity. NSGA-II improved upon the ranking scheme of NSGA, adopted a more efficient nondominated sorting approach, and introduced a crowded comparison operator as its density estimator. [15]

6.2 NSGA-II

Comparing existing evolutionary optimization methods, NSGA-II stands out as a robust and efficient algorithm. It utilizes non-dominated sorting and crowding dis-

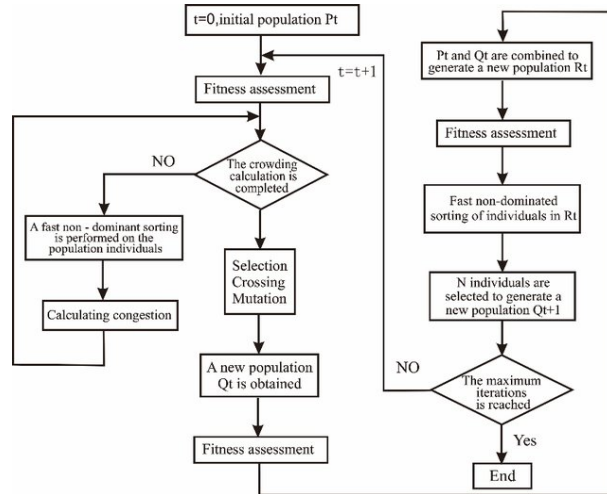


Figure 6.2: NSGA-II algorithm flowchart [14]

tance assignment mechanisms to maintain diversity and convergence in the population. NSGA-II applies a fast sorting technique to identify non-dominated solutions, ensuring that the generated Pareto front captures a diverse range of optimal solutions. Additionally, the crowding distance assignment enables NSGA-II to maintain a good balance between exploration and exploitation of the search space. [2]

6.2.1 NSGA-II: Working Principle and Benefits

NSGA-II is a well-established algorithm that follows a multi-step process to effectively solve multi-objective optimization problems. The algorithm begins by initializing a population of potential solutions randomly. These solutions represent individuals in the search space and serve as the starting point for the optimization process.

Each solution in the population is then evaluated based on its fitness, which is determined by the objective functions and constraints associated with the problem. The fitness evaluation provides a quantitative measure of how well a solution performs with respect to the objectives.[2]

Next, the non-dominated sorting technique is applied to classify the solutions into multiple fronts based on their dominance relationships. Dominance is a key concept in multi-objective optimization, where one solution is said to dominate another if it is at least as good as the other in all objectives and strictly better in at least one objective. By using non-dominated sorting, the solutions are grouped into different fronts based on their dominance relationships, with the first front containing the non-dominated solutions.[2]

To illustrate this process in Figure, let's consider a two-objective problem with objectives f_1 and f_2 . Suppose we initialize a population $P(t)$, referred to as the parent population, with a size of $N = 8$. Random solutions are generated within the search space, and their fitness values f_1 and f_2 are calculated. Each solution, denoted as p , is then assigned a rank based on its dominance depth, see Figure 6.3.

The dominance depth calculation involves two key entities: the domination count n_p , which represents the number of solutions that dominate solution p , and the set S_p , which consists of solutions dominated by p . Solutions for which $n_p = 0$ are considered non-dominated and are assigned a rank of 1. These non-dominated solutions form the first front, representing the best trade-off solutions in the objective space. [2]

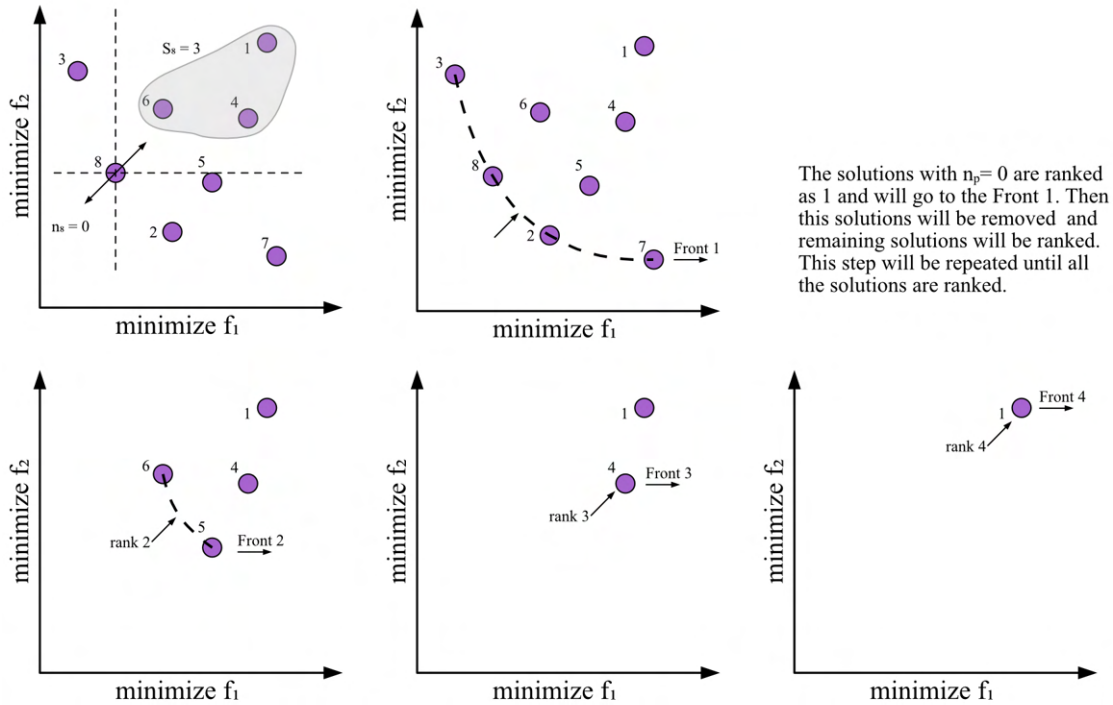


Figure 6.3: Ranking process in NSGA-II

By applying non-dominated sorting and assigning ranks to the solutions, NSGA-II effectively identifies the Pareto optimal front, which consists of solutions that cannot be improved in one objective without worsening another.

After the sorting process, NSGA-II calculates the crowding distance for each solution, see Figure 6.4, which represents its diversity and distribution within the population. Crowding distance measures the crowdedness of a solution with respect to its neighbours lying on the same front. The crowding distance guides the selection of individuals for the next generation, favouring solutions with greater diversity to maintain a well-distributed Pareto front.

In NSGA-II, the generation of offspring solutions is carried out through selection, crossover, and mutation operations. Selection is performed using the Crowded Binary Tournament Selection Operator. Each solution in the population is characterized by two attributes: non-domination rank and crowding distance. In a tournament selection process, a pair of randomly chosen solutions, i and j , are compared based on the following rules:

1. If the ranks of solutions i and j are different, the solution with the better rank is selected.

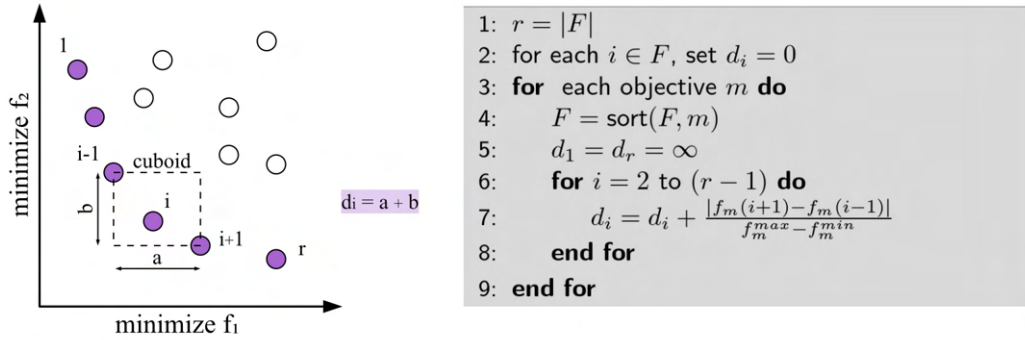


Figure 6.4: Illustration of crowded distance in NSGA-II [2]

2. If the ranks of solutions i and j are the same, the solution with the greater crowding distance is selected.
3. If both the rank and crowding distance of solutions i and j are identical, one of them is randomly selected.

[2]

After the selection process, crossover and mutation operations are applied to the selected solutions, generating offspring. These offspring solutions are then combined with the parent population to form a new population denoted as $R(t)$, with a size of $2N$, where N is the population size. [2]

Next, the population $R(t)$ is classified into different non-domination classes or fronts. The classification is based on the solutions' non-domination ranks. The new population $R(t)$ is then filled with solutions from different non-domination fronts, one front at a time. The filling process begins with the first non-domination front (class one), followed by the second non-domination front, and so on. [2]

Since the size of $R(t)$ is limited to $2N$, it may not be able to accommodate all the fronts. In such cases, the fronts that cannot be accommodated are deleted. When processing the last allowed front, there might be more solutions in the front than the remaining slots in the new population. This situation is illustrated in Figure 6.5. Instead of arbitrarily discarding some members from the last front, a selection criterion based on diversity is employed. The solutions that contribute the most to the diversity of the selected solutions are chosen. [2] To achieve this, the solutions of the last front that cannot be fully accommodated are sorted based on their crowding distance values in descending order. The solutions with higher crowding distance values are given priority. The non-dominated sorting process is then performed again on the combined population to identify new fronts and calculate the crowding distances for the updated population. [2]

The process of creating offspring, evaluating fitness, sorting the population, and selecting solutions iteratively continues until a termination condition is met, such as reaching a maximum number of generations or convergence of the population. [2]

By following this process, NSGA-II ensures that the new population maintains a di-

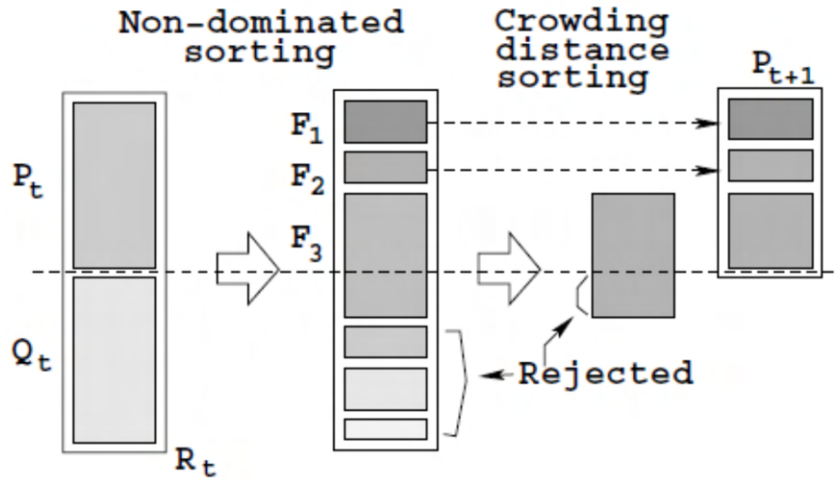


Figure 6.5: Schematic of NSGA-II procedure [2]

verse set of solutions across the non-domination fronts, promoting a well-distributed Pareto front and preserving the quality of the solutions. [2]

NSGA-II offers several benefits for solving multi-objective optimization problems in the context of building design. Firstly, it provides a diverse set of Pareto-optimal solutions, which allows decision-makers to explore trade-offs and make informed choices based on their preferences. The algorithm strikes a balance between convergence (finding optimal solutions) and diversity (capturing a wide range of solutions) through non-dominated sorting and crowding distance assignment.

Moreover, NSGA-II is computationally efficient, making it suitable for complex problems with multiple objectives and constraints. The fast non-dominated sorting technique and the use of elitism ensure that the algorithm maintains a high-quality population, leading to faster convergence and better solution quality. Additionally, the integration of genetic operators, such as crossover and mutation, allows for the exploration of the search space, enhancing the algorithm's ability to discover diverse and novel solutions.

Multi-objective optimization is crucial in building design to improve performance across various criteria. Evolutionary algorithms, such as NSGA-II, offer an effective approach to solving multi-objective problems by simulating the process of natural selection and evolution. NSGA-II's ability to generate a diverse set of Pareto-optimal solutions, maintain convergence and diversity, and handle constraints makes it a suitable choice for addressing the complexity of building optimization.

in Figure 7.1. Converting a multiobjective optimization problem into a simplistic single-objective problem puts decision-making before optimization, that is before alternatives are known. As explained above, articulating preferences without a good knowledge of alternatives is difficult, and thus the resulting optimum may not correspond to the solution the user would have selected from the set of Pareto optimal solutions. Treating the problem as a true multiobjective problem means putting the preference articulation stage after optimization, or interlacing optimization and preference articulation. This will help the user gain a much better understanding of the problem and the available alternatives, thus leading to a more conscious and better choice. [4]

Furthermore, the resulting multiple Pareto optimal solutions can be analyzed to learn about interdependencies among decision variables, objectives, and constraints. Such knowledge about the interactions can be used to redefine the model of the optimization problem to get solutions that, on the one hand, correspond better to reality, and, on the other hand, satisfy better the DM's preferences. [4]

In the field of multiobjective optimization, there has been a traditional approach of combining mathematical programming techniques and decision-making in a mutually dependent manner. The primary objective has always been to assist the decision maker (DM) in identifying the solution that aligns best with their preferences. This is achieved through an interactive procedure characterized by alternating stages of decision-making and optimization.[4]

The DM actively participates throughout this procedure, particularly during the decision-making stage. When considering the alternatives generated by the optimization process, decision-making requires a model that captures the DM's preferences, either explicitly or implicitly. This model serves as a guide to determine the most preferred solution among the currently considered alternatives or to provide insights for finding better solutions in subsequent optimization stages.[4]

7.1 Multi-objective Optimization Tool

In order to optimize the objectives of this thesis, the decision was made to use an existing tool instead of developing a custom implementation of the NSGA-II algorithm. Since the entire problem was formulated and solved within the Grasshopper environment, it was preferred to choose a tool that seamlessly integrates with Grasshopper. Therefore, the Wallacei plugin was selected for the purpose of multi-objective optimization.

Wallacei shown in Figure 7.2 is an evolutionary engine designed specifically for running evolutionary simulations in Grasshopper 3D. It provides users with highly detailed analytic tools and a range of comprehensive selection methods, allowing them to gain a deeper understanding of their evolutionary runs and make informed decisions at every stage of the simulation. This includes setting up the design problem, analyzing the results, and selecting the desired solution(s) for the final output. Additionally, Wallacei enables users to selectively reconstruct and extract any phe-

notype from the population after completing the simulation. [13]

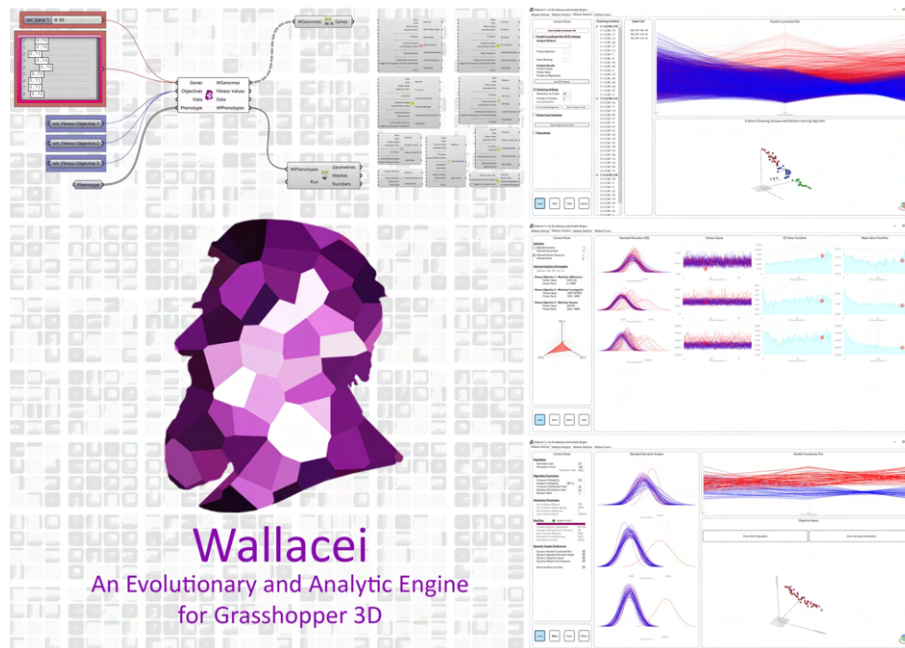


Figure 7.2: Environment of Wallacei plugin [13]

The Wallacei plugin is designed to provide users with efficient access to the data generated by their evolutionary simulations. It offers clear and effective methods for analysis and selection, with the goal of helping users, regardless of their expertise level, better understand their evolutionary simulations, thoroughly grasp the outputted numeric values, and seamlessly extract the optimized data all within a single user interface.[13]

Wallacei X employs the NSGA-II algorithm [2] as its primary evolutionary algorithm. It also utilizes the K-means method as the clustering algorithm, which aids in organizing and categorizing the solutions generated by the optimization process. [13]

By utilizing the Wallacei plugin, which is specifically tailored for evolutionary simulations in Grasshopper, I was able to take advantage of its advanced features, seamless integration with the existing workflow, and user-friendly interface. This choice provided a streamlined and efficient approach to accessing and analyzing the output data, ultimately facilitating the optimization process and supporting decision-making.

7.2 Setting Optimization Criteria

The optimization process in multi-objective optimization involves three phases: model building, optimization process, and decision-making. A properly built model with well-defined problem criteria is crucial for achieving effective optimization. In the context of enhancing the early-stage design process of concrete structures, the

first criterion focused on the structural aspects. After conducting several testing optimizations, six criteria were identified as follows (as described in Chapter 4 in Section 4.4):

1. Deflection of the column
2. Stress on the 3rd-floor walls
3. Deflection of the model at end-points and midpoints of elements
4. Deflection of the 1st and 2nd-floor slabs
5. Weight of the column
6. Weight of the slabs

For optimization purposes, these criteria were normalized, and a weighted average procedure, as illustrated in Figure 7.3, was applied to obtain a single objective or criterion.

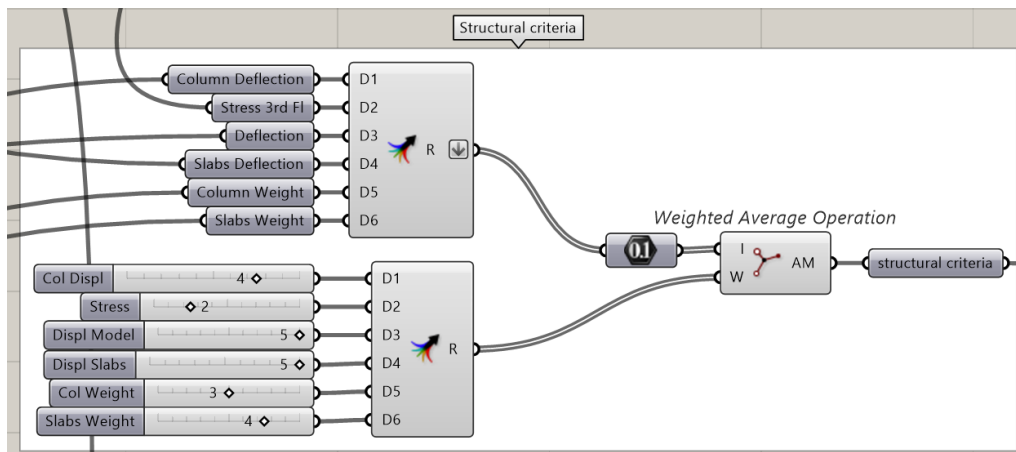


Figure 7.3: "Weighted Average" operation definition for structural criteria in Grasshopper

Providing feedback to the optimization algorithm is crucial for achieving desirable optimization results. Feedback can be given through tools such as the "Graph Mapper" component mentioned in Chapter 4 in Section 4.4. Another approach is penalization, which involves intentionally worsening the value of a criterion to discourage the reproduction of certain individuals in the optimization process. By penalizing undesirable solutions, their influence on subsequent generations is reduced, helping to eliminate unfavourable traits.

In the case of the designed building, it was undesirable for the branches of the tree column located on the exterior to intersect with the glass wall and be present in the interior. To address this architectural criterion, the penalization process was employed. Collisions between the column and the house were detected using the "Brep|Brep" component, and the resulting values were used as a form of penalization. The "Graph Mapper" component, incorporating the "Remap" component described

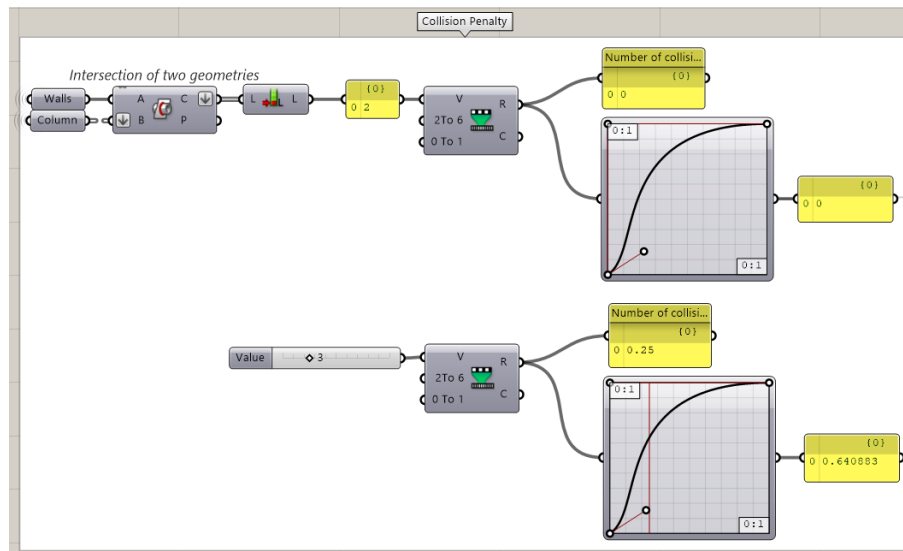


Figure 7.4: Collision detection and normalization of the number of collisions in Grasshopper

in Chapter 4 in Section 4.4, was utilized to ensure proper penalization. A "Bezier" graph type, depicted in Figure 7.4, was employed for collision penalization.

Next, environmental criteria were defined to assess the building's performance in terms of occupant comfort, as explained in Chapter 5 in Section 5.3. Four criteria were established for this analysis:

1. Solar radiation analysis for summer
2. Solar radiation analysis for winter
3. View Out analysis
4. Sunlight analysis

Similar to the structural criteria, these environmental criteria were also normalized, and a weighted average operation, depicted in Figure 7.5, was applied to obtain a single objective or criterion.

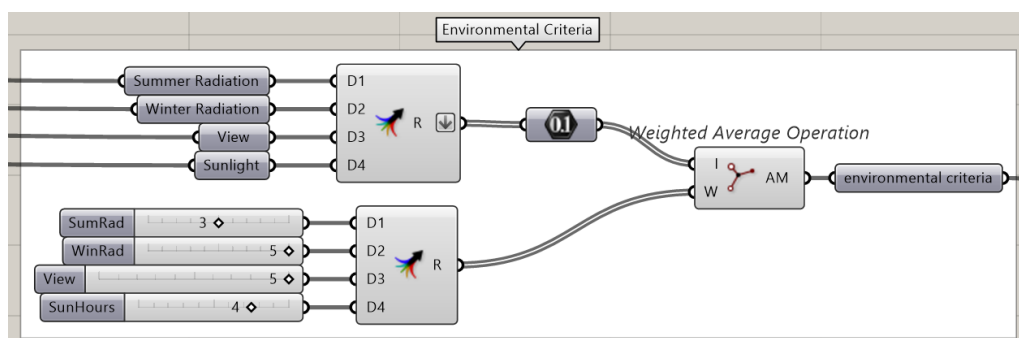


Figure 7.5: "Weighted Average" operation definition for environmental criteria in Grasshopper

As mentioned in Chapter 5 in Subsection 5.2.3, for the purposes of the optimization trees were designed using the Lands Design plugin. To prevent collisions between these trees, information about their positions was provided to the optimization algorithm. The "Point Intersection" component from the Kangaroo plugin, a built-in component for Grasshopper, facilitated the connection of pairs of points representing the tree positions and enabled the calculation of distances between them.

In summary, before commencing the optimization process, the following criteria were established:

1. Structural criteria to:
 - (a) Minimize the deflection of the column
 - (b) Minimize the stress on the 3rd-floor walls
 - (c) Minimize the deflection of the model at end-points and midpoints of elements
 - (d) Minimize the deflection of the 1st and 2nd-floor slabs
 - (e) Minimize the weight of the column
 - (f) Minimize the weight of the slabs
2. Architectural criterion to:
 - (a) Minimize collision between the tree column and the house
3. Environmental criteria to:
 - (a) Minimize solar gain in summer
 - (b) Maximize solar gain in winter
 - (c) Maximize View Out
 - (d) Normalize sunlight hours
4. Collision criterion for trees:
 - (a) Maximize the distances between trees to avoid collisions

When maximizing certain criteria in the optimization process, it is necessary to divide 1 by the value that needs to be maximized. This approach ensures that a larger number results in a smaller normalized value, aligning with the objective of maximizing the criterion.

7.3 Optimization Setup

The optimization setup began once all the criteria were defined and the model was prepared. The "Wallacei X" component of the Wallacei plugin shown in Figure 7.6 was used for this purpose. The "Genes" node was connected to sliders and gene pools, while the "Objectives" node was connected to the criteria established in the

previous section. The "Data" node of the "Wallacei X" component allowed the connection of various data types such as numbers, texts, and geometries to be saved for each solution in the population. However, it is recommended to use numerical data only to prevent CPU overload and potential crashes in Grasshopper. In the specific

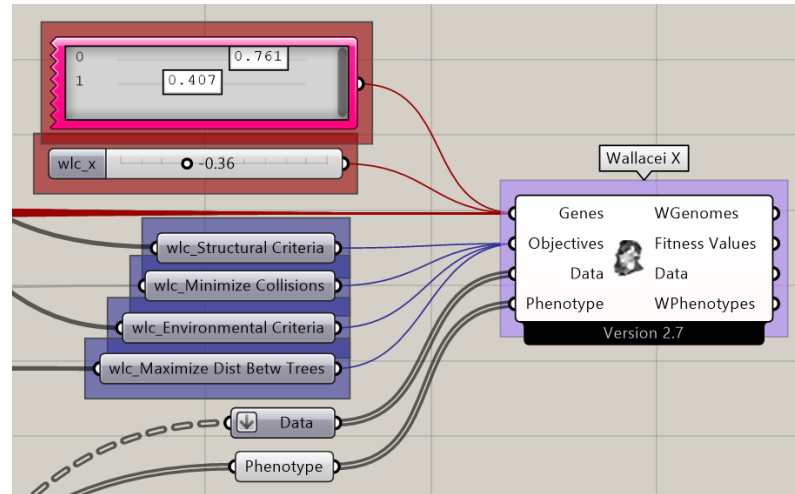


Figure 7.6: "Wallacei X" component's inputs illustration

optimization process, all the real values (non-normalized) of the structural and environmental criteria, as well as the dimensions of the tree column cross-sections, were connected to the "Data" node. This step facilitated the decision-making process by filtering the best solutions. The "Phenotype" node could also be connected to any data type, but it was advised to do so after the optimization process to avoid software crashes. In this case, the geometry of the building analyzed by the Karamba plugin, visual results from the Ladybug component, awning geometry, and the trees were connected before exporting the results. Within the "Wallacei X" component, there were five tabs accessible by double-clicking on the component, see Figure 7.11. For the optimization and decision-making processes, the first three tabs, namely "Wallacei Settings," "Wallacei Analytics," and "Wallacei Selection," are needed.

After consulting with my supervisor, it was decided to create a population of size 2500 with 50 individuals and 50 generations. However, due to the heavy Grasshopper definition and insufficient engine capacity, the software crashed, and the optimization simulation could not be completed. Following further consultation, it was decided to perform the optimization with a population size of 900, comprising 30 individuals and 30 generations. This population size had been used successfully throughout the project during testing optimizations, ensuring its feasibility. To compensate for the reduced population size and regain the benefits of a larger population, such as greater diversity and improved results, the "Mutation Probability" and "Random Seed" options shown in Figure 7.7 were utilized.

Firstly, I would like to mention the significance of the "Random Seed" option. As mentioned in Chapter 6.1, evolutionary algorithms possess inherent randomness. This means that running the same optimization repeatedly will not yield identical results. Consequently, if changes are made to the optimization problem definition, tracking the impact of these changes becomes challenging since the variations in optimization outcomes primarily result from the algorithm's randomized nature.

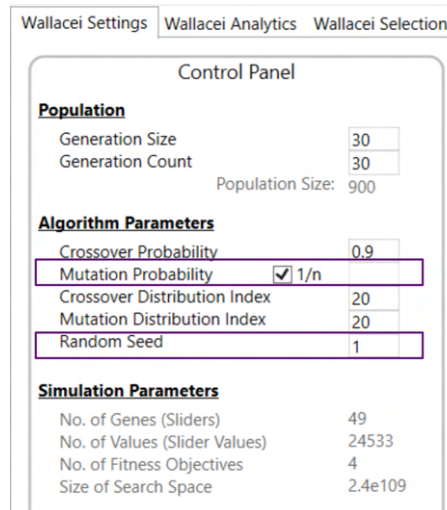


Figure 7.7: Wallacei X control panel for illustrating "Random Seed" and "Mutation Probability" options

In Wallacei, the "Random Seed" option provides control over this randomness. By default, it is set to 1, meaning that running multiple optimizations without changing the "Random Seed" will produce the same results.

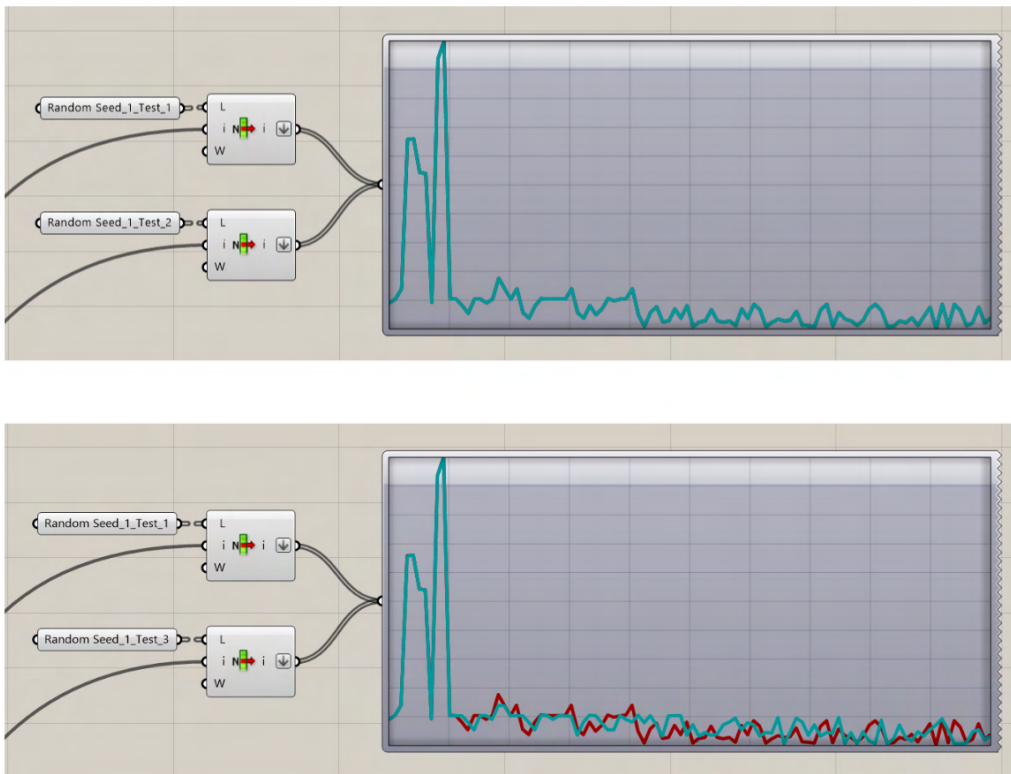


Figure 7.8: Illustration of the results of test optimizations with "Random Seed" option set to 1

To illustrate this process, several tests were conducted with a population size of 100 and the "Random Seed" set to 1. *Test1* and *Test2* maintained an unchanged problem definition, while *Test3* modified the number format of the sliders controlling

the dimensions of the column's trunk cross-section from integers to floating-point numbers. Figure 7.8 visualizes the differences between these tests using the "Quick Graph" component in Grasshopper.

Test1 and *Test2*, both with "Random Seed" set to 1 and no changes in the definition, exhibit no difference. However, *Test1* and *Test3* (and similarly, *Test2* and *Test3*) show distinctions, clearly indicating that changes to the problem definition caused the observed differences. On the other hand, setting the "Random Seed" to 0 allows the algorithm's inherent randomness to prevail, resulting in different optimizations that make tracking changes impractical. Two tests with "Random Seed" set to 0 were also conducted to demonstrate this behaviour, and the disparities between them are illustrated in Figure 7.9.

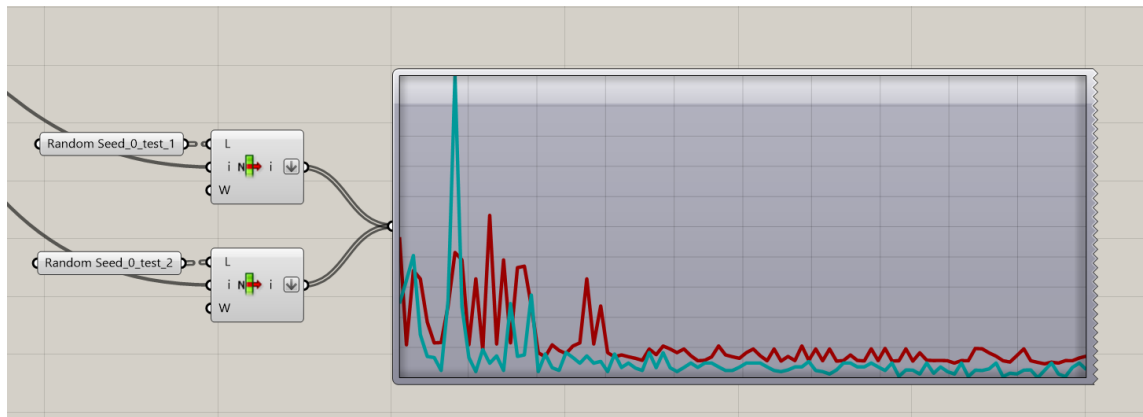


Figure 7.9: Illustration of the results of test optimization with "Random Seed" option set to 0

The "Mutation Probability" determines the percentage of mutations occurring within a generation. Deb et al. recommends setting the mutation probability to $1/n$, where n represents the number of variables (sliders) in the design problem. To introduce more diversity in the results, I chose to adjust this number to 0.1. Figure 7.10 illustrates the distinctions between two optimization scenarios that share the same problem definitions but differ in the setting of the "Mutation Probability." In the first scenario (*Test3*), the "Mutation Probability" is set to $1/n$, while in the second scenario (*Test4*), it is set to 0.1.

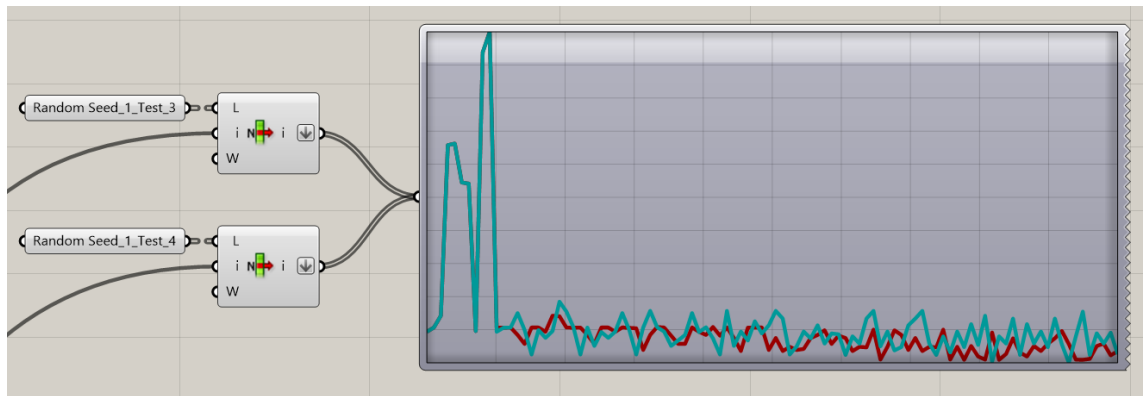


Figure 7.10: Illustration of the results of test optimizations with different "Mutation Probability"

These manipulations of the "Mutation Probability" and "Random Seed" options contribute to increased diversity among the optimization solutions, allowing for the selection of the best outcomes.

In the context of this thesis, four optimizations were performed:

1. "Mutation Probability" set to $1/n$ and "Random Seed" set to 1
2. "Mutation Probability" set to 0.1 and "Random Seed" set to 1
3. "Mutation Probability" set to $1/n$ and "Random Seed" set to 2
4. "Mutation Probability" set to 0.1 and "Random Seed" set to 2

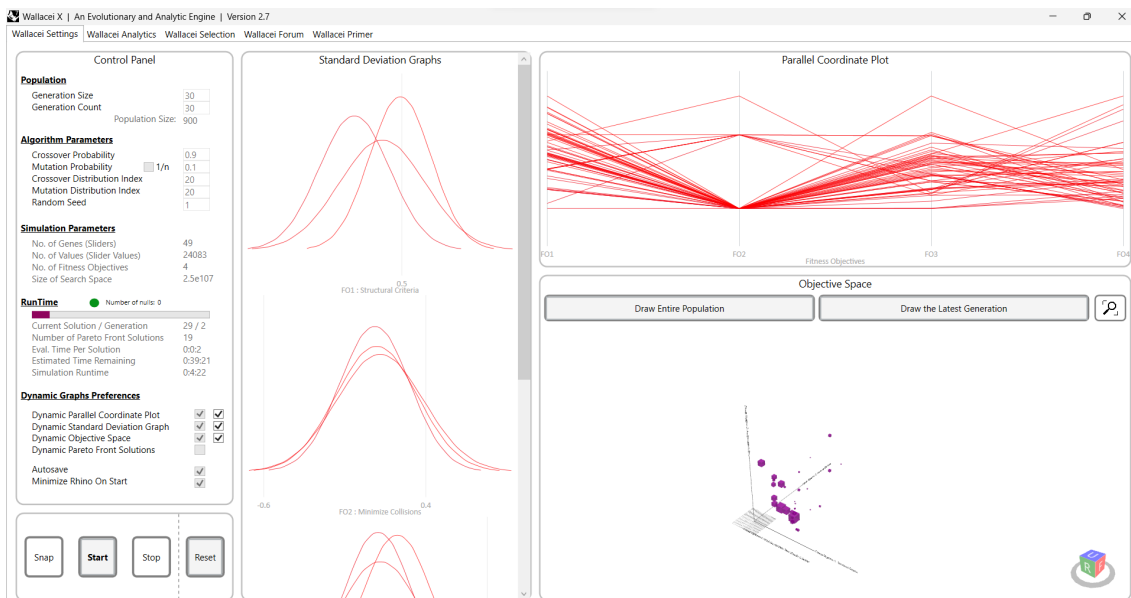


Figure 7.11: Illustration of optimization progress of the second optimization process

In Figure 7.11 the progress of the second optimization process with "Mutation Probability" set to 0.1 and "Random Seed" to 1 is illustrated. Within this tab, there are also additional windows displaying Standard Deviation graphs (SD), Parallel Coordinate Plot (PSP), and Objective Space (OS). These visualizations serve as valuable tools for monitoring the simulation's progress in real-time.

7.4 Analyzing Optimization Results

The "Wallacei Analytics" tab of "Wallacei X" serves as the main tool for analyzing the results of the simulation and cross-referencing the different fitness objectives. There are two methods to select a solution. The first method involves inputting the generation and solution number, while the second method requires inputting the fitness objective and the solution's fitness rank. The selected solution will then be displayed in the graphs as illustrated in figure 7.12.

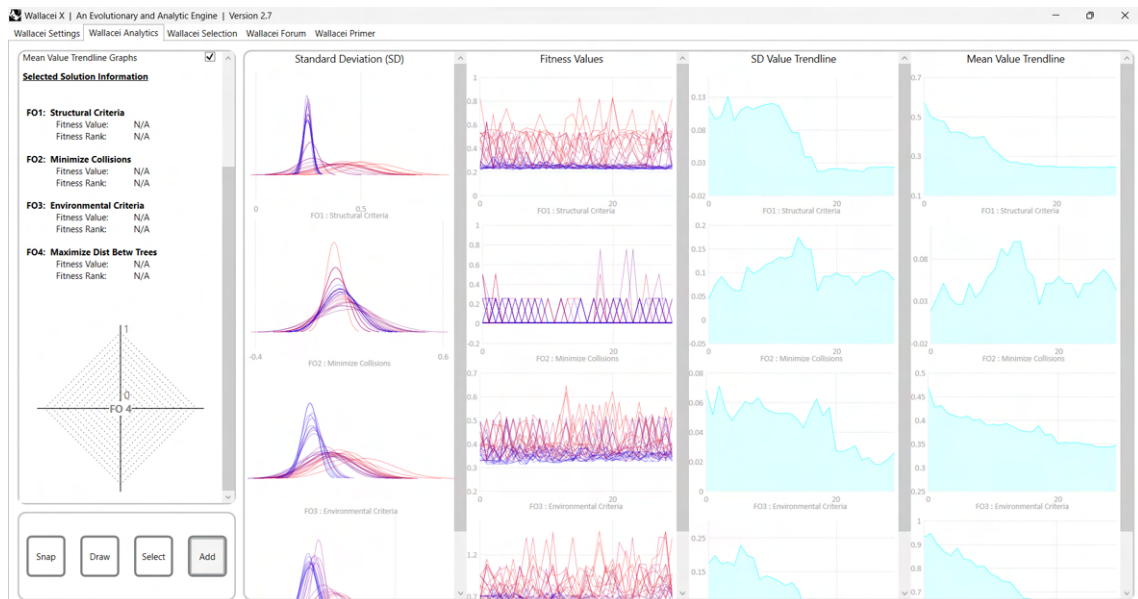


Figure 7.12: "Wallacei Analytics" tab for analyzing the provided optimization

As shown in Figure 7.12 this tab contains of Standard Deviation chart, Fitness Values graph, SD Value Trendline and Mean Value Trendline. The curves in SD and FV graphs illustrate the generations from first - as red to the last - as blue.

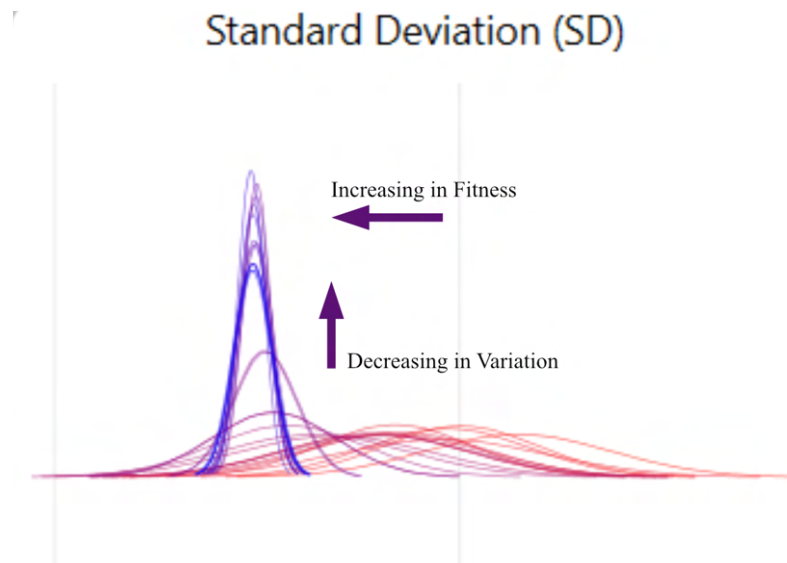


Figure 7.13: Explanation of Standard Deviation chart in "Wallacei Analytics"

The curves in SD Chart that move to the left are getting better in mean value, if they move to the right it means that they are getting worse. Also if the cruves get higher, it means that they are decreasing in variation and vice versa, see Figure 7.13.

In Figure 7.14 the FV chart is illustrated. Here we can see that the last generations have better fitness values as they are closer to 0.

Although all the graphs are very informative and user-friendly, I found the "Diamond Fitness Chart" the most useful. This chart can have multiple axes, depending on the

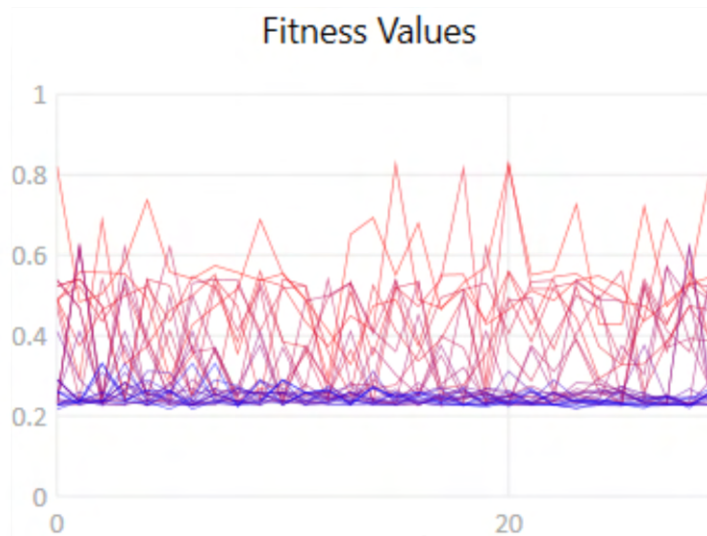


Figure 7.14: Fitness Values chart in "Wallacei Analytics"

number of fitness objectives or criteria, see Figure 7.15. Each axis on the diamond chart represents a fitness objective, and the proximity of a point to the centre of the chart indicates the fitness level of the corresponding solution.

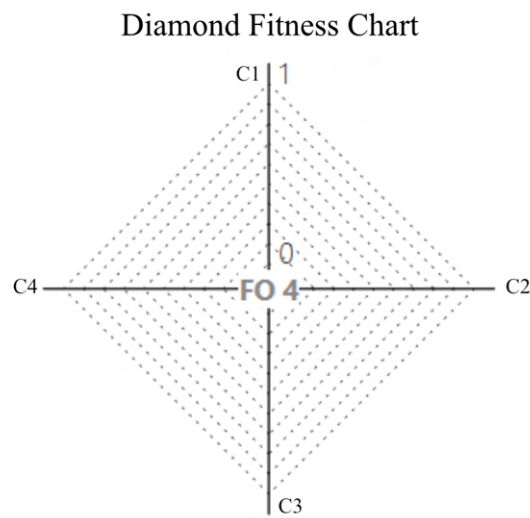


Figure 7.15: Visualization of Diamond Fitness chart in "Wallacei Analytics"

After the completion of the simulation optimization, the "Wallacei Analytics" tab was utilized for the decision-making process. In this tab, the solution with rank 0 (the best solution) was reviewed for all criteria except for the second criterion, as illustrated in Figures 7.16, 7.17 and 7.18, and selected for export with respect to the second criterion (see Section 7.2).

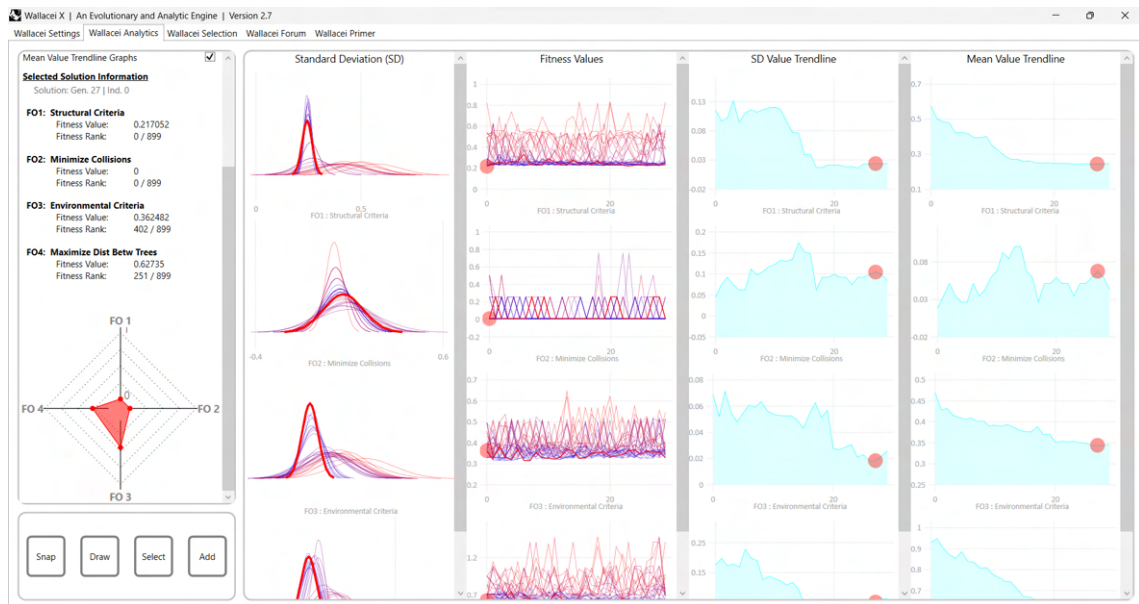


Figure 7.16: Illustration of analysis of the first (structural) criteria with rank 0

As illustrated in Figure 7.17 the fitness value of the second criterion for the best solution of the third criterion is not equal to 0, which means that there are unwanted collisions between the tree column branches and building walls. Thus this solution was not selected for export.

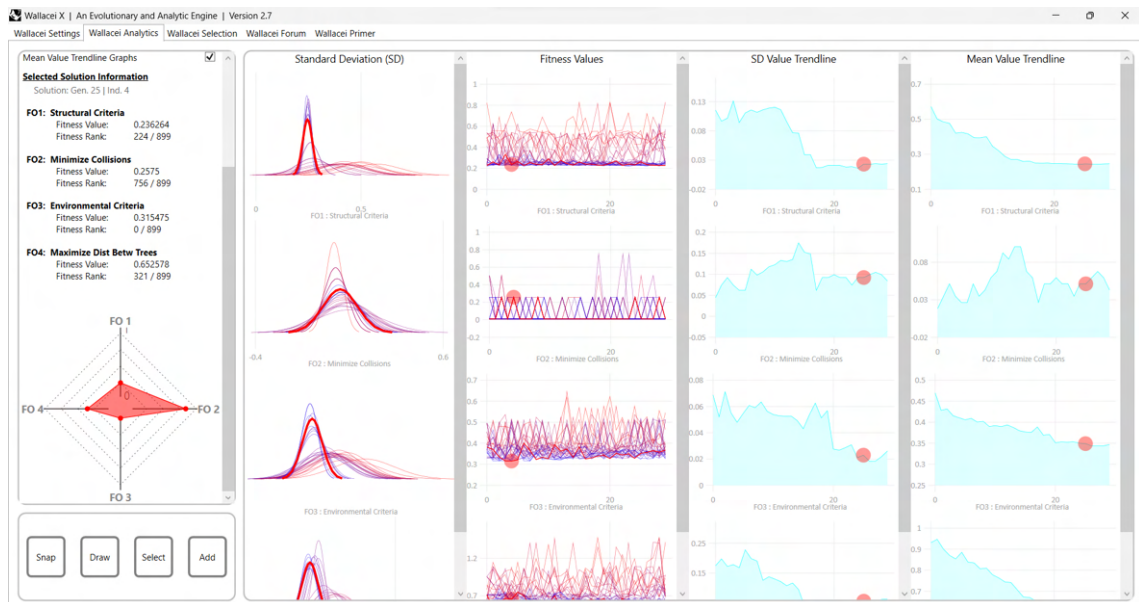


Figure 7.17: Illustration of analysis of the third (environmental) criteria with rank 0

Subsequently, the "Wallacei Selection" tab was accessed, where the K-Means Clustering Algorithm proved to be useful. This algorithm partitions the generation into a specified number of groups. The targeted generation for clustering was chosen, and the desired number of clusters was specified. The resulting clusters are by default displayed in an objective space (OS) and could also be visualized using Parallel Coordinates Plot (PCP), as shown in Figure 7.19. Dashed lines illustrate the centres

of the clusters.



Figure 7.18: Illustration of analysis of the fourth (distance between the trees) criteria with rank 0

To streamline the analysis, the last five generations were clustered, as it is generally more efficient to focus on the most recent generations that tend to contain the best solutions. Once the clustering process was complete, the solutions selected for export included those from the Pareto front that had not yet been included in the export list. These selected solutions were then exported for further analysis.

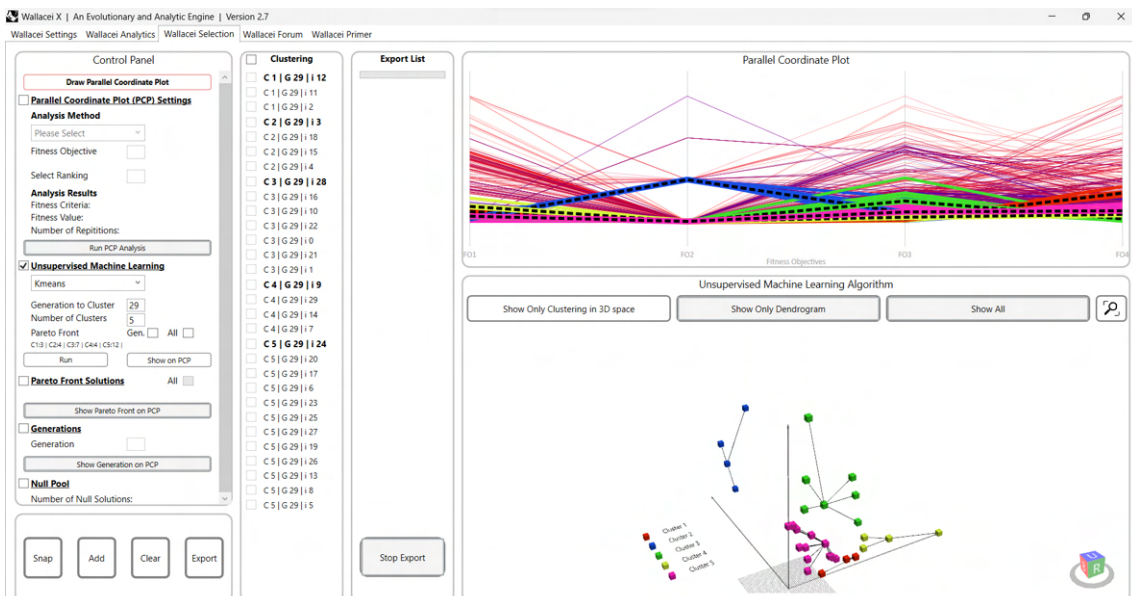


Figure 7.19: Visualization of clustering process in "Wallacei Selection" tab

7.5 Decision Making Process

In the decision-making process, the second criterion, which was assigned the highest weight as mentioned earlier, played a crucial role. After exporting the solutions, it was necessary to establish an effective method for determining the best optimal solution. Two steps were implemented to achieve this.

The first step involved utilizing the "Wallacei Analytics" tab and selecting the "Diamond Fitness Chart" component illustrated in Figure 7.20.

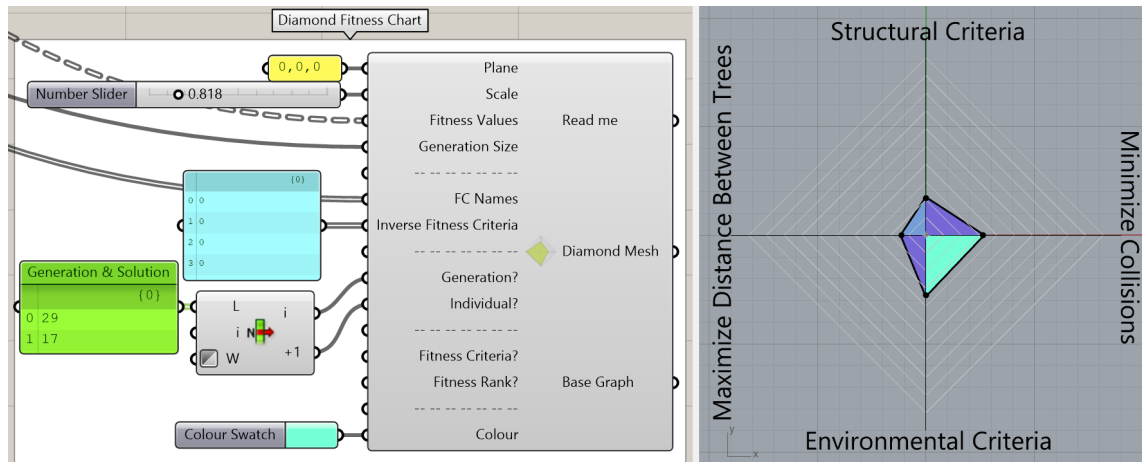


Figure 7.20: Illustration of utilizing the "Diamond Fitness Chart" component in Grasshopper environment

This component serves the same purpose as the "Diamond Fitness Chart" within the "Wallacei X" component, but having it available outside the "Wallacei X" component allowed for manipulation within the Grasshopper environment. The "Diamond Fitness Chart" component facilitated the analysis and filtering of the exported solutions based on their second fitness value, which needed to be equal to 0.

The next step involved utilizing the "Data" that had been inserted into the "Wallacei X" component to be saved for each solution. The solutions could be chosen based on the components of the structural and environmental criteria. Since criteria in multi-objective optimization often conflict with each other, preferences had to be assigned to find the optimal solution. For the structural criteria, preference was given to the deflection of the model, the deflection of the 1st and 2nd floor slabs, and the deflection of the tree column. Regarding the environmental criteria, the preference was given to solar gain in winter and the percentage of view-out.

Ultimately, the final decision regarding solution selection was based on personal preference. As the architect, I had the authority to determine which solution best aligned with my preferences and design intentions.

The selection process involved applying preferences step by step. Firstly, solutions were filtered based on the deflection of the model, followed by the deflection of the 1st and 2nd floor slabs, and finally the deflection of the tree column. In each step, solutions with the smallest values for the respective criteria were extracted.

The "Wallacei X" component consists of four outputs as illustrated in Figure 7.23.

The first output of the "Wallacei X" component consists of the genomes of all solutions. To gain insights into the genetic mechanisms behind these optimizations, the "Genomic Analysis" feature of the Wallacei plugin was employed. This anal-

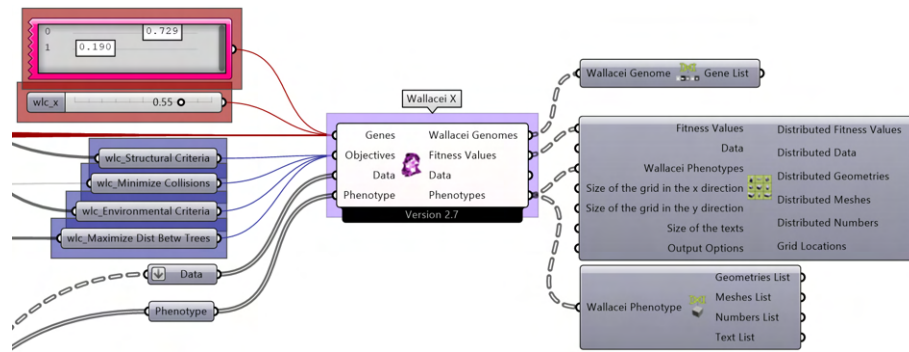


Figure 7.23: "Wallacei X" component's outputs illustrations

ysis utilizes the "Decode Genome" component, allowing for a detailed exploration and decoding of the genomes of the solutions. Appendix B showcases the complete decoded genomes of the first and last generations for each provided optimization.

Figure 7.24 provides an overview of the "Genomic Analysis" for the last generation of the second optimization.

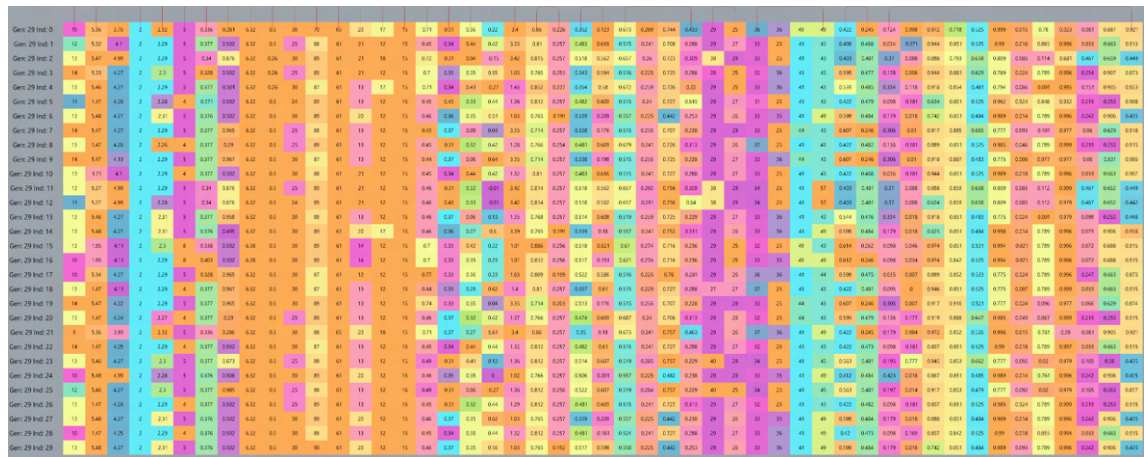


Figure 7.24: Illustration of decoded genomes using "Decode Genome" component

The second output of the "Wallacei X" component contains the fitness values of all solutions. These fitness values were crucial in the sorting process discussed in the previous section. In this section, the solutions with the best and worst structural fitness values across all provided optimizations were compared, see Figure 7.25.

Additionally, the best and worst fitness values for the environmental criteria were compared using the results obtained from the "Fitness Values" output, see Figure 7.26. This comparison allowed for an assessment of the performance of the solutions in terms of their environmental impact.

Seed_1_mut_0.1		Seed_2	
{1;29}		{29;1}	
0	0.829397	0	0.215035
1	0.2575	1	0
2	0.545571	2	0.536414
3	0.723073	3	0.622336

Figure 7.25: Illustration of the compared fitness values of the structural criterion obtained from the "Fitness Values" output of "Wallacei X" component

Seed_1_mut_0.1		Seed_2	
{25;4}		{0;10}	
0	0.236264	0	0.566101
1	0.2575	1	0
2	0.315475	2	0.755863
3	0.652578	3	0.797881

Figure 7.26: Illustration of the compared fitness values of the environmental criterion obtained from the "Fitness Values" output of "Wallacei X" component

The "Data" output of the named component, as mentioned in Section 7.5, was already described and utilized in the sorting process.

Criteria	Best Results	Worst Results
Deflection - column (<i>mm</i>)	0.542334	7.210784
Stress (<i>MPa</i>)	0.26869	0.454334
Deflection - model (<i>mm</i>)	1.301986	8.22241
Deflection - slabs (<i>mm</i>)	12.52259	117.02069
Weight - column (<i>kg</i>)	7705.963273	13112.246154
Weight - slabs (<i>kg</i>)	188788.94905	170480.103126

Table 7.1: Best and worst results of the structural criteria obtained from the "Data" output of "Wallacei X" component

To further illustrate the comparison of the best and worst fitness values mentioned earlier, Tables 7.1 and 7.2 present the actual values associated with these fitness values. Table 7.1 displays the real values corresponding to the best and worst fitness values achieved for the structural criteria and Table 7.2 presents the real values

corresponding to the best and worst fitness values obtained for the environmental criteria.

The mentioned tables offer a comprehensive perspective on the solutions' performance concerning the specific criteria under consideration. Analyzing the actual values presented in these tables facilitates the interpretation and comparison of the top-performing and lowest-performing solutions. This process helps to acquire valuable insights into the distinctive design attributes and identify potential areas for improvement.

As we can see, Table 7.2 highlights that among the environmental fitness values, the only criterion where the solution with the best fitness value outperforms the solution with the worst fitness value is the "View Out" criterion. The difference in the fitness values can be attributed to the higher weight assigned to the "View Out" criterion during the optimization process. This indicates that prioritizing the "View Out" criterion led to a better performance in this specific aspect, while other environmental criteria may have been compromised to some extent.

Criteria	Best Results	Worst Results
Summer Radiation (kWh/m^2)	208.452283	112.80662
Winter Radiation (kWh/m^2)	65.481066	67.501277
View Out (%)	31.872898	27.66527
Sunlight (<i>hours</i>)	3.273973	3.273973

Table 7.2: Best and worst results of the environmental criteria obtained from the "Data" output of "Wallacei X" component

Lastly, the "Phenotype" output of the "Wallacei X" component contains the geometries of the exported solutions. These geometries provide visual representations of the optimized solutions and serve as a basis for further analysis and evaluation.

In Figures 7.27, 7.28, 7.29, 7.30, 7.31 and 7.32 are shown the visual representations of the compared fitness values of the structural and environmental criteria, see Figures 7.25 and 7.26.

To enhance visualization and facilitate analysis, the results for the structural criteria have been scaled by a factor of 500. This scaling allows for a clearer representation of the deflection of the model. It is important to note that the structural analysis provided in this study is based on a linear static analysis, as discussed in Chapter 4.

Additionally, Appendix C illustrates the Pareto front *phenotypes* obtained from the provided optimizations.

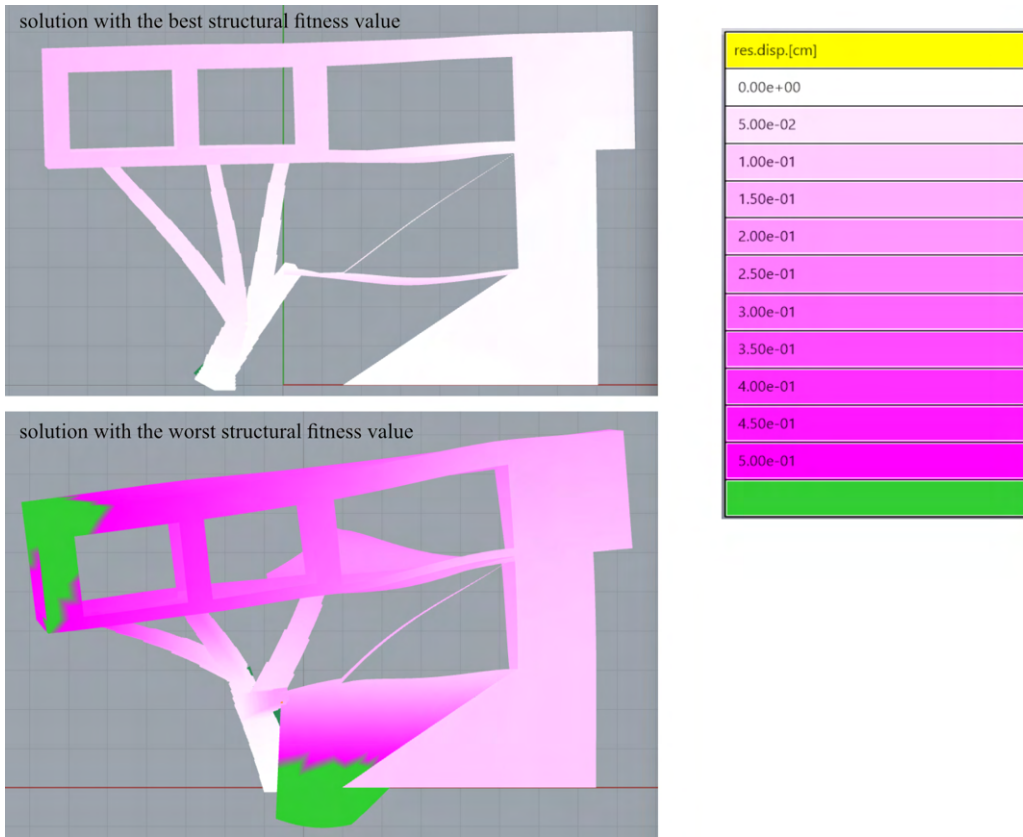


Figure 7.27: Illustration of the deformed geometries of the compared structural fitness values

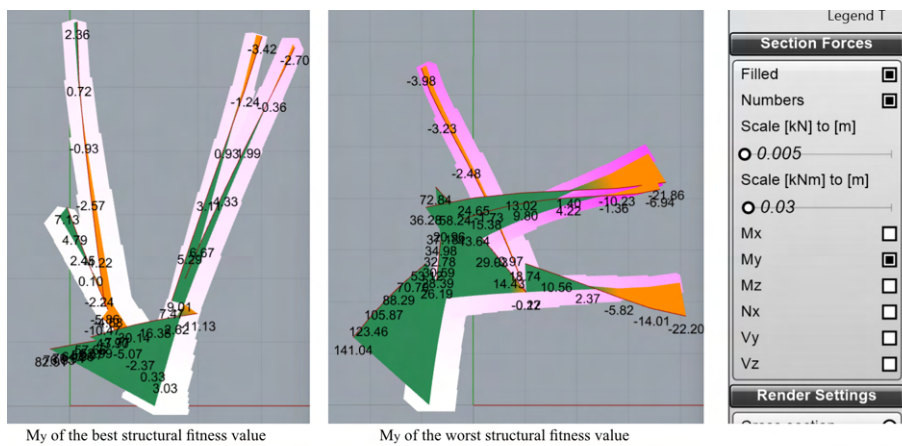


Figure 7.28: Illustration of the M_y 's of the compared structural fitness values

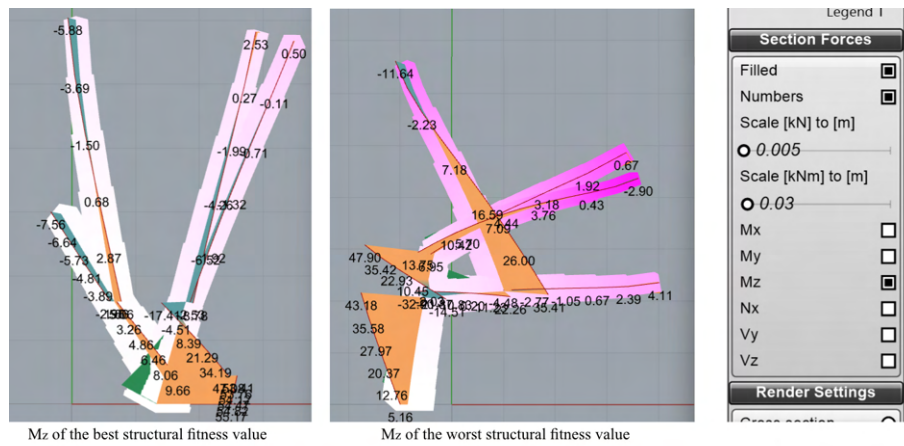


Figure 7.29: Illustration of the M_z 's of the compared structural fitness values

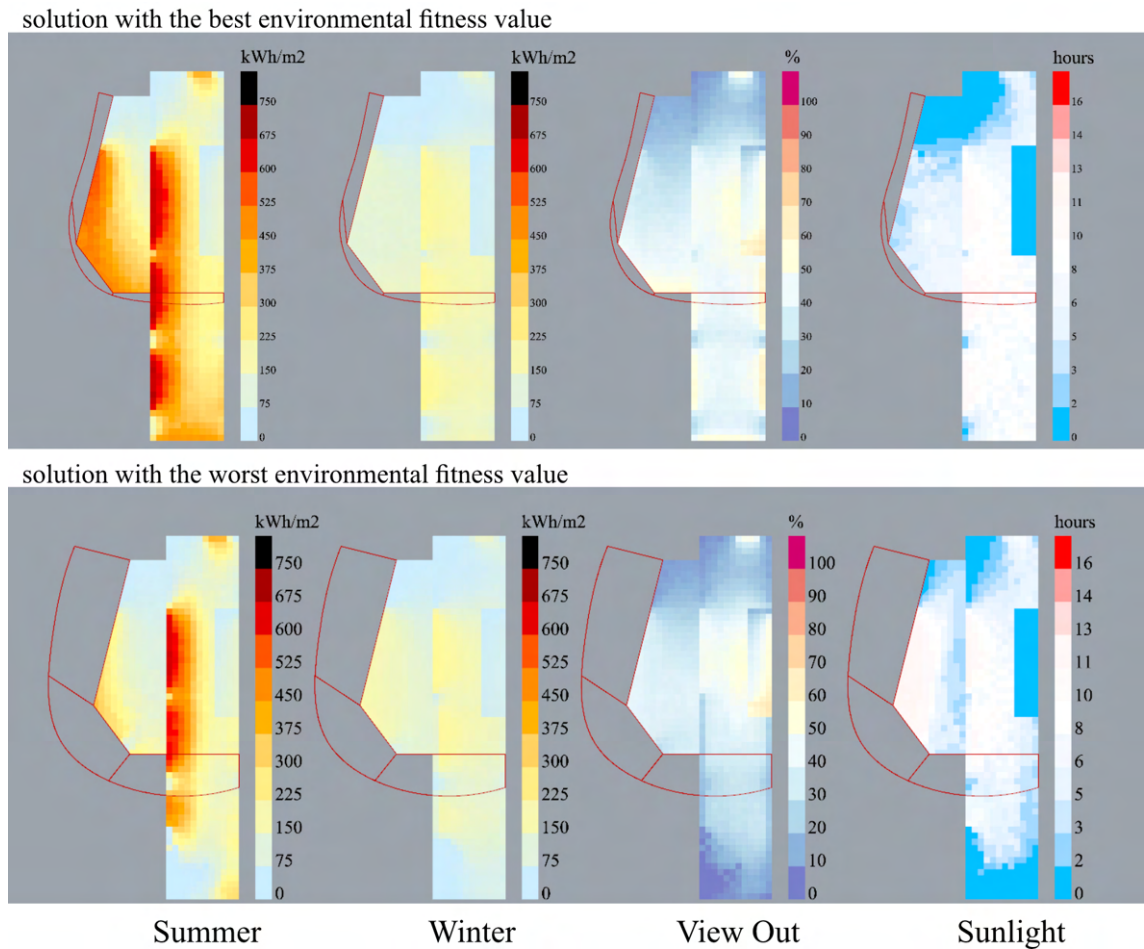


Figure 7.30: Illustration of the results of compared environmental fitness values (top view)

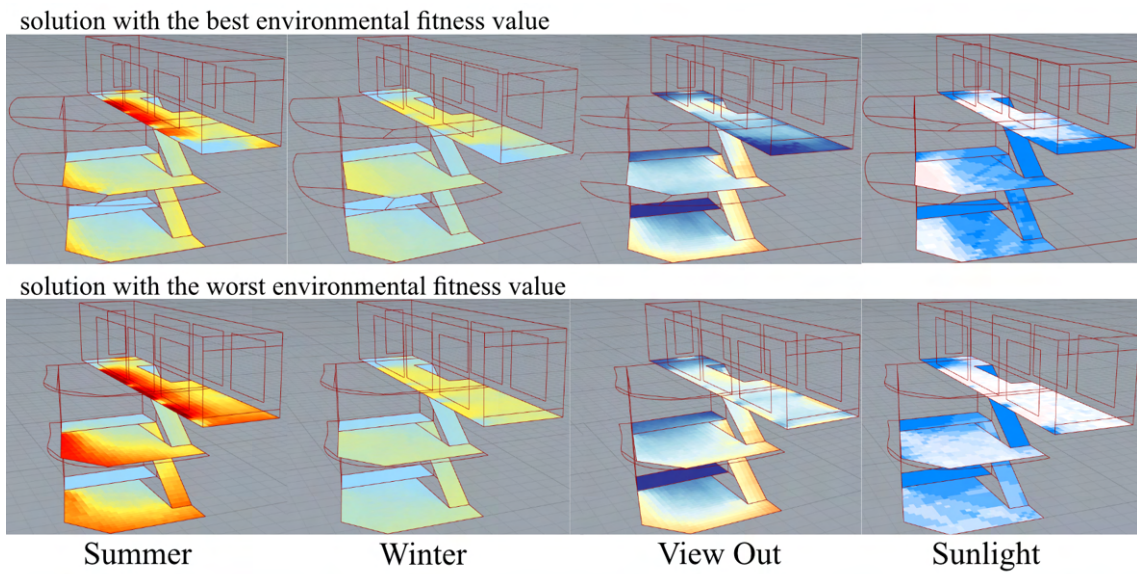


Figure 7.31: Illustration of the results of compared environmental fitness values (perspective view)

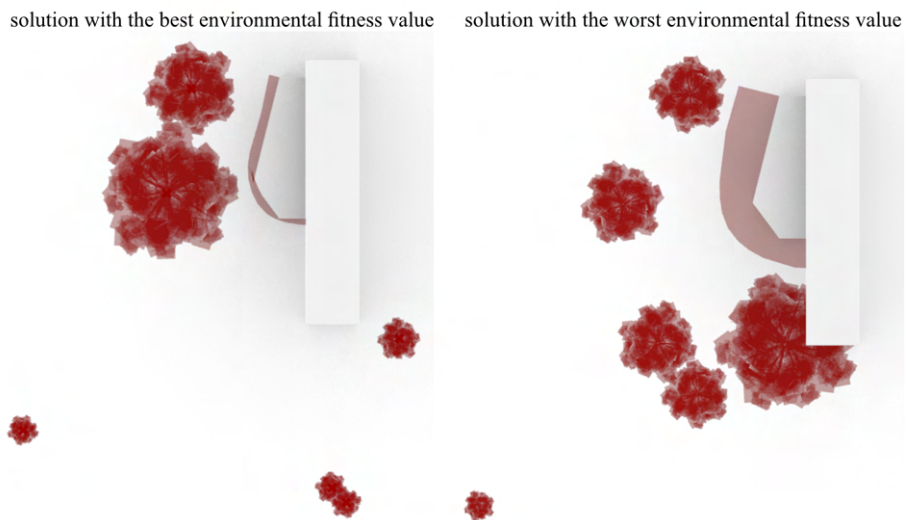


Figure 7.32: Illustration of the difference between trees and awning shapes of the compared environmental fitness values

7.6.1 The Chosen Solution

Here, I would like to showcase the chosen solution, which was determined through the sorting process described in the decision-making section, see 7.5. This particular solution was extracted from the second optimization discussed in the decision-making section.

Table 7.3 displays the fitness values of this chosen solution, while Table 7.4 provides the real values of the component criteria for both the structural and environmental aspects.

Criteria	Results
Structural criteria	0.232612
Collision (column x walls)	0
Environmental criteria	0.35024
Distance between trees	0.579295

Table 7.3: Fitness values of the chosen solution (normalized)

Criteria	Results
Deflection - column (<i>mm</i>)	1.708458
Stress (<i>MPa</i>)	0.229094
Deflection - model (<i>mm</i>)	0.9488
Deflection - slabs (<i>mm</i>)	7.05737
Weight - column (<i>kg</i>)	7855.504377
Weight - slabs (<i>kg</i>)	239255.172725
Summer Radiation (<i>kWh/m²</i>)	216.347877
Winter Radiation (<i>kWh/m²</i>)	82.600051
View Out (%)	34.531133
Sunlight (<i>hours</i>)	4.219406

Table 7.4: Real values of the component criteria of the chosen solution

Table 7.5 presents the cross-sectional dimensions of the column for the chosen solution. These dimensions provide insights into the physical characteristics of the column, which contribute to its structural performance and overall design.

Figure 7.33 represents the structural aspects of this solution, with the analyzed model scaled by a factor of 500 for improved visualization.

Column Part	Dimensions from bottom to top (mm)
trunk	870 to 610
branch 1	610 to 440
branch 2	610 to 410
branch 3	440 to 360
branch 4	440 to 360
branch 5	410 to 260
branch 6	410 to 290

Table 7.5: The cross-sectional dimensions of the tree column of the chosen solution

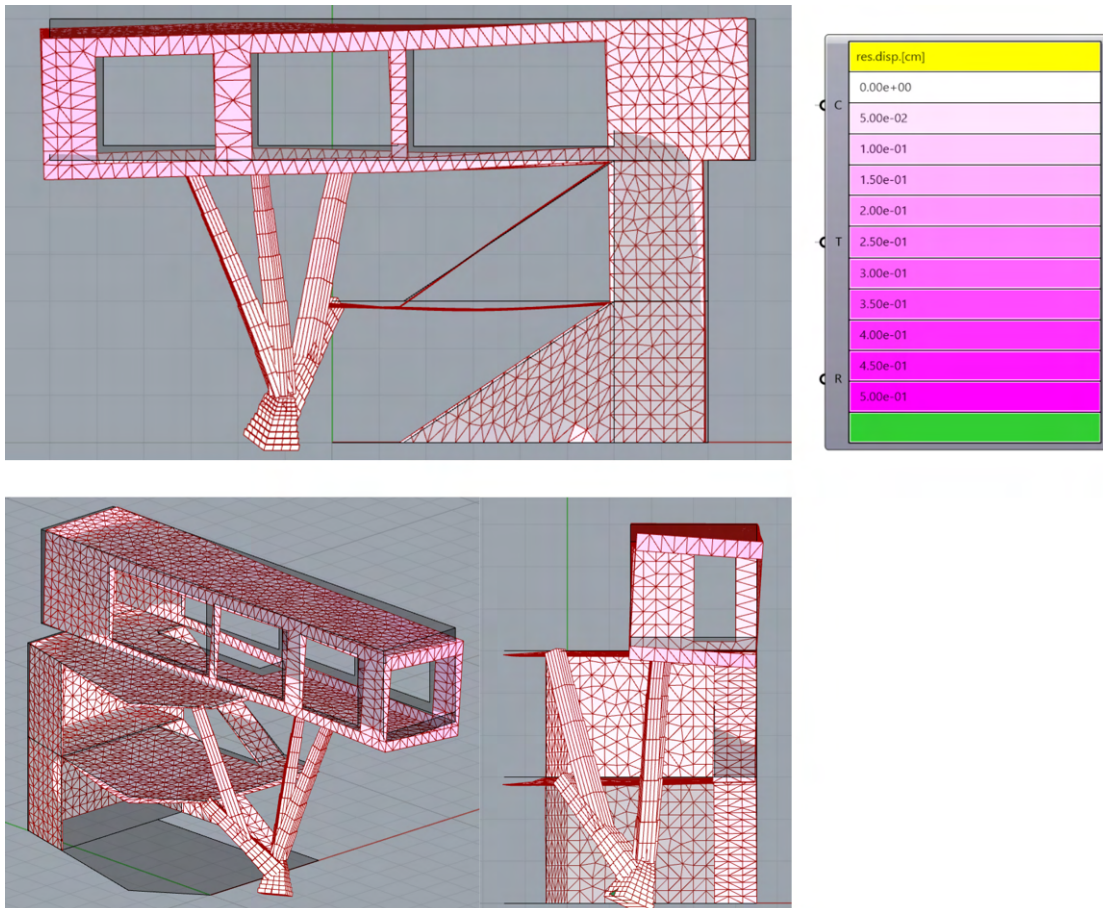


Figure 7.33: Visualization of the structural analysis results of the chosen solution

The tree column corresponding to this solution was also examined in terms of its performance as a cantilever with a length of 9m. The results of these investigations are presented in Table 7.6 and Figure 7.34.

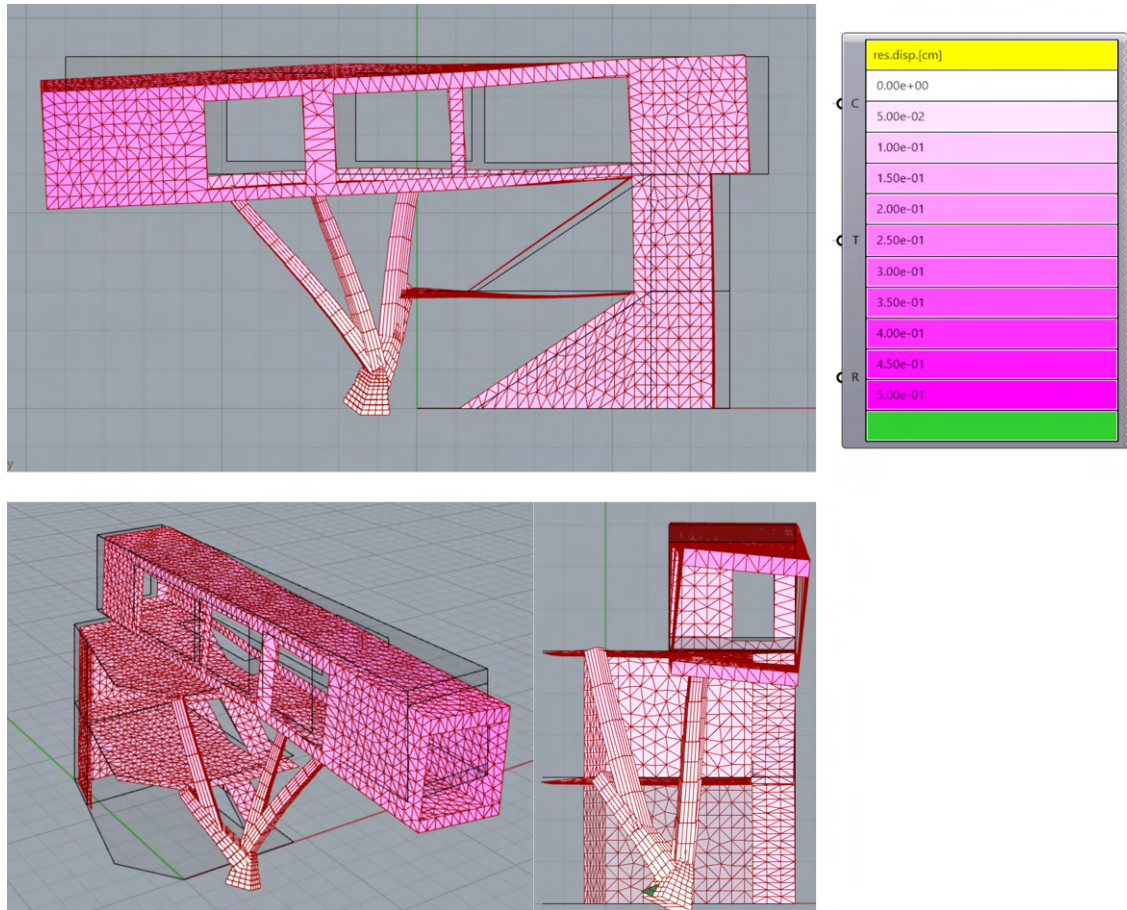


Figure 7.34: Cantilever 9m

Criteria	Results
Deflection - column (<i>mm</i>)	5.42682
Stress (<i>MPa</i>)	0.263511
Deflection - model (<i>mm</i>)	2.227728
Deflection - slabs (<i>mm</i>)	21.33972
Weight - column (<i>kg</i>)	8090.292579
Weight - slabs (<i>kg</i>)	273005.172725

Table 7.6: Real values of the component criteria of the structural criterion for cantilever of a length of 9 m

Figures 7.35 and 7.36 represent the environmental outcomes of the chosen solution.

Furthermore, the influence of column thickness on the environmental analysis was further explored, illustrated in Figure 7.37, by leveraging the data output from the "Wallacei X" component. This analysis aimed to examine how different column thicknesses impact the environmental criteria. To conduct this analysis, the column shape from the selected solution was retained, while the cross-section dimensions

were altered to observe the resulting effects.

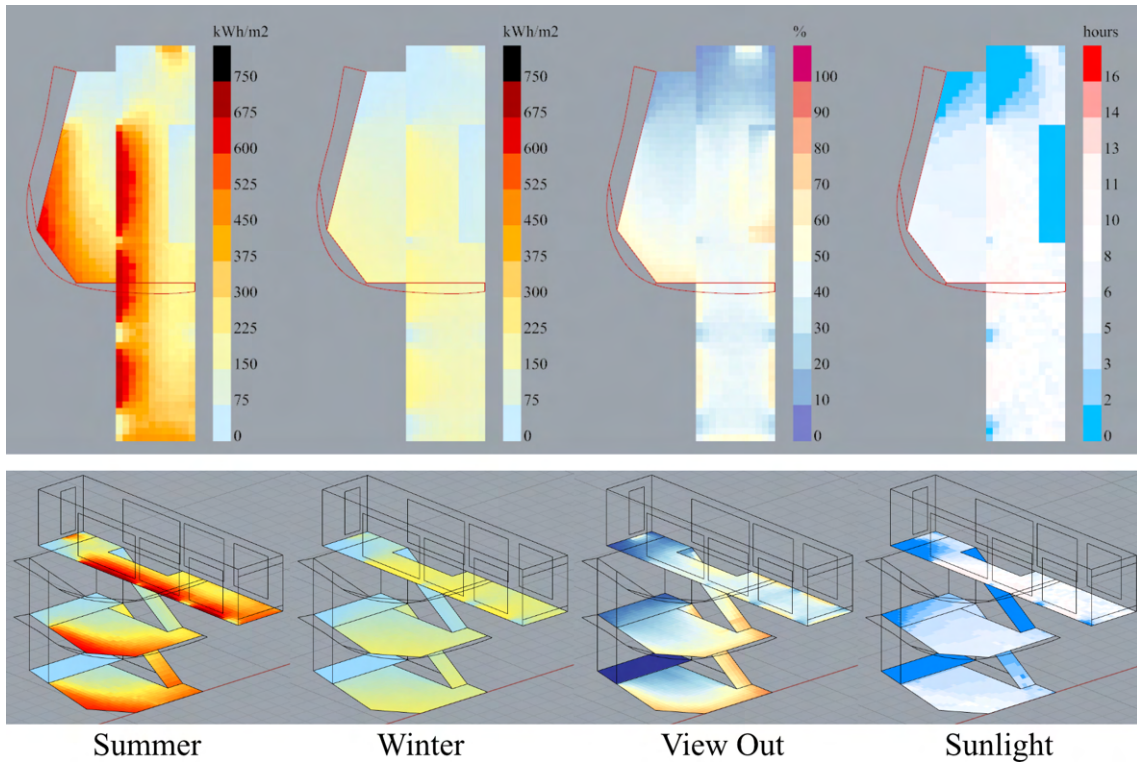


Figure 7.35: Visualization of the environmental analysis results of the chosen solution

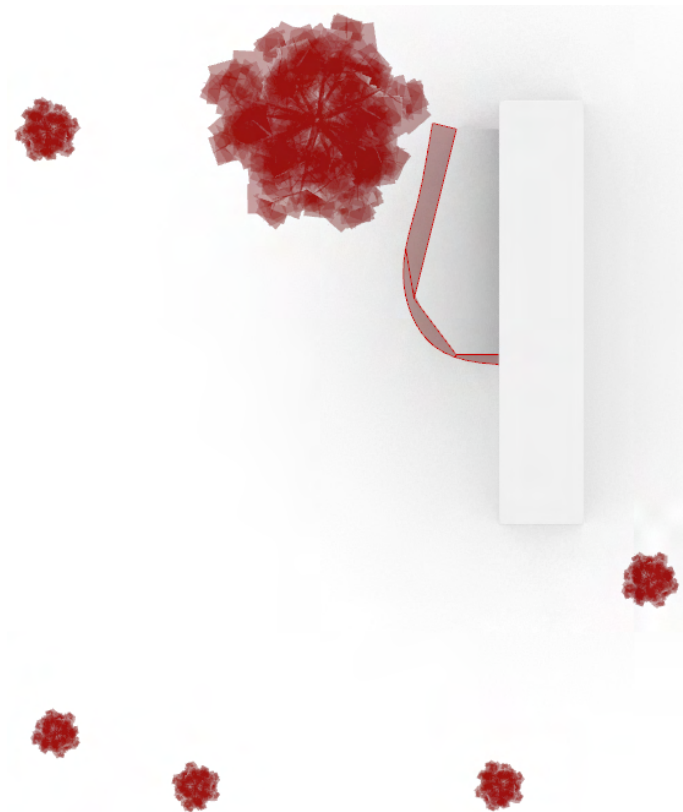


Figure 7.36: Illustration of the awning shape and tree positions for the chosen solution

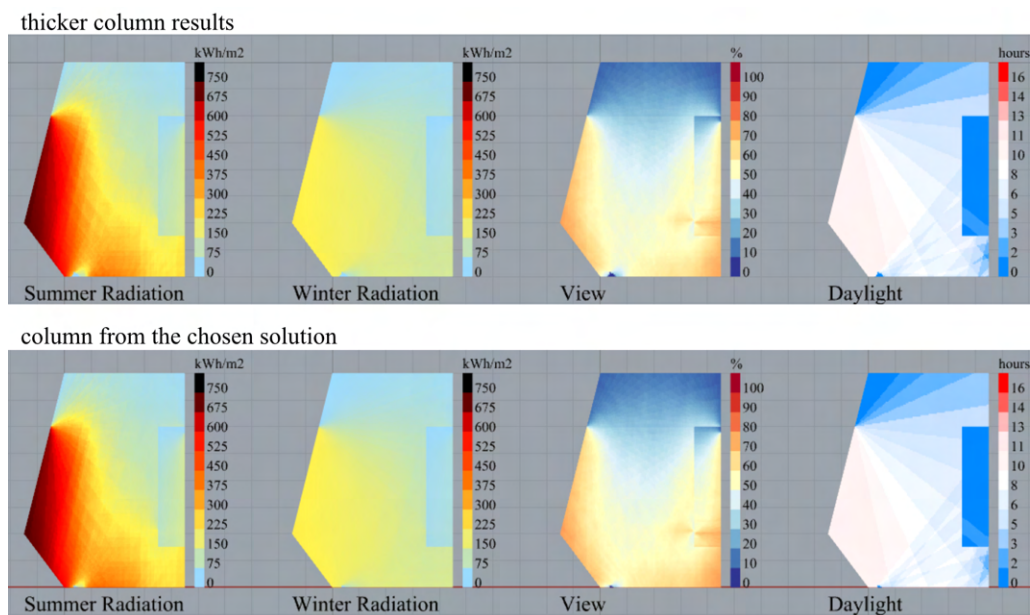


Figure 7.37: Illustration of the results of the environmental analysis comparing the column of the chosen solution with a column of maximum thickness

Chapter 8

Cross-Sectional Analysis

The chosen solution's column underwent a comprehensive cross-sectional analysis using the already mentioned software IDEA Statica, which is widely recognized in the industry for its advanced capabilities in structural analysis. The analysis focused specifically on the last cross-section of the first branch within the column tree. The cross-section dimensions were measured as 440x440, rounded to a practical size of 450x450.

Table 8.1 presents the internal forces determined for this specific cross-section by the Karamba plugin, providing essential data that elucidate the distribution and magnitude of forces acting within the column. These values play a crucial role in evaluating the structural response and performance of the column.

N (kN)	V_y (kN)	V_z (kN)	M_x (kN/m)	M_y (kN/m)	M_z (kN/m)
-61.2	10.1	-7.1	2.7	-2.8	4.7

Table 8.1: Internal forces obtained from Karamba 3D for the chosen cross-section

To further assess the column's performance, Figure 8.1 showcases the N-N-M Capacity Diagram for the aforementioned cross-section. This diagram provides a visual representation of the column's load-carrying capacity under various axial forces and bending moments, aiding in the assessment of its structural integrity.

Moreover, Figure 8.2 provides a visual representation of the designed reinforcement strategy for the calculation. This illustration offers insights into the arrangement and distribution of reinforcement elements within the cross-section. By strategically placing reinforcement bars, the column's strength and durability will be enhanced, ensuring optimal load transfer and structural integrity.

Figure 8.3 illustrates an informative reinforcement scheme at the connection points between different segments of the column.

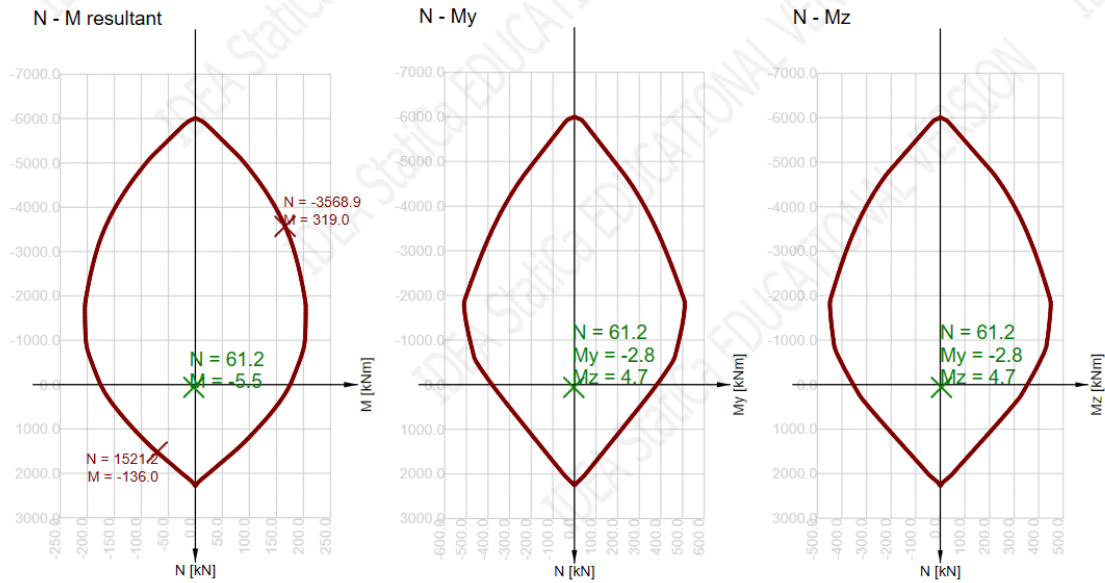


Figure 8.1: N-N-M capacity diagram from provided analysis for the chosen cross-section

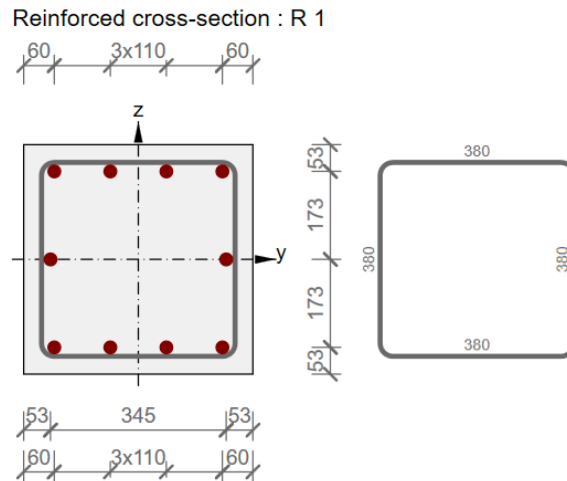


Figure 8.2: Design of reinforcement for the chosen cross-section

8.1 Tree Column Branch and Slab Connection

In this section is described how the tree column branches, responsible for supporting the first and second floor slabs, are connected to the slabs using the Schöck Isokorb® XT type B shown in Figure 8.4. This innovative connection system not only ensures structural integrity but also serves as a thermal break element, effectively minimizing thermal bridging. Figure 8.5 provides a visual representation of this connection configuration, highlighting the utilization of the Schöck Isokorb® XT type B in the connection process.

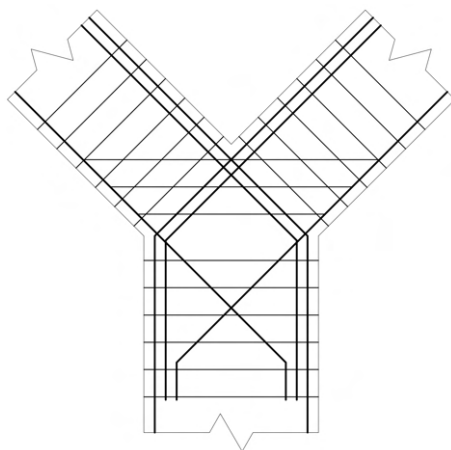


Figure 8.3: Informative reinforcement scheme of the reinforcement design in intersection points of the tree column

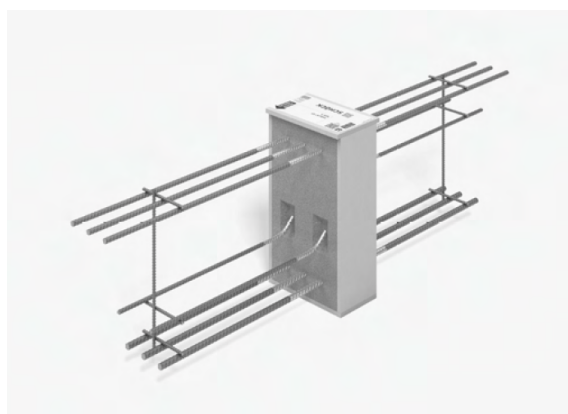


Figure 8.4: 3D illustration of the Schöck Isokorb® XT type B

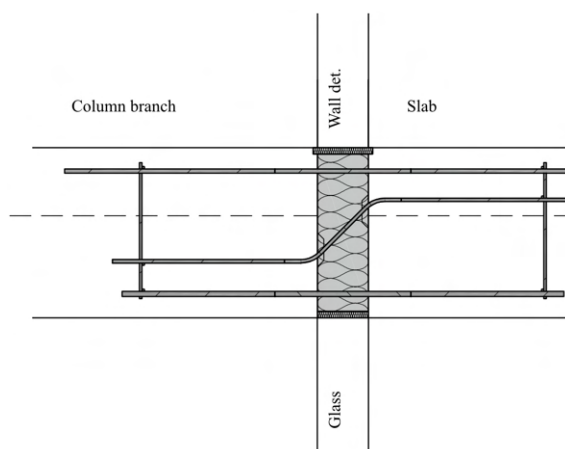


Figure 8.5: 2D scheme of the connection of the column branch and slab using Schöck Isokorb® XT type B

Chapter 9

Digital fabrication

The unique design of the column requires custom-made formworks specifically tailored to its shape. Using digital fabrication techniques is the most efficient approach to minimize material waste in the production of these formworks.

The construction industry has witnessed numerous benefits from the integration of technology, including reduced time and cost, improved quality, increased labour productivity, and enhanced worker safety. The success of a construction project relies on cost-effective methods that deliver the project within schedule, ensuring acceptable quality and safety standards.

Concrete formwork can account for up to 60 percent of the costs associated with a structure or building. Employing value-engineering solutions such as CNC machines is expected to significantly reduce project costs. In traditional construction practices, formwork carpenters construct formworks by cutting wooden panels. However, studies have shown that these carpenters often suffer from work-related musculoskeletal disorders in various areas of their bodies. CNC machines not only reduce concrete package costs but also promote the long-term health of workers, improve waste management, increase production rates, enhance formwork quality, and ensure worker safety. [16]

CNC machines are considered more environmentally sustainable compared to traditional saw-based approaches, as they cut wooden panels precisely, minimizing wood waste.[16]

To prepare the file for the CNC machine, a tool that enables fast and efficient file preparation was required. The OpenNest plugin, specifically the "OpenNest" component and the "Pack Objects" component, proved to be suitable for this purpose. OpenNest is a 2D polyline packing tool designed for fabrication processes such as laser or CNC cutting.

The fabrication file for the CNC machine was prepared for the chosen solution's column. Initially, the column was reconstructed since working with the "mesh" resulting from the optimization was not efficient. The cross-sectional dimensions of the column were rounded to 50 mm. Once the column was prepared, the packaging process could be started.

As mentioned earlier, two components of the OpenNest plugin, namely the "Pack Objects" component and the "OpenNest" component, were utilized. The "Pack Objects" component shown in Figure 9.1 converted the objects from the 3D environment into the 2D environment, arranging them in a single row. These objects were then

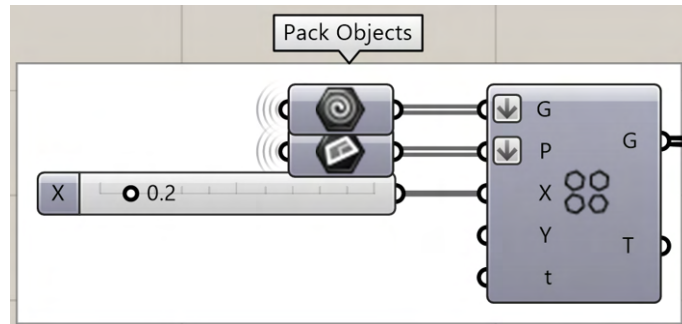


Figure 9.1: "Pack Objects" component in Grasshopper

connected to the "OpenNest" component, specifically the "Geo" (geometry) node. The "Sheets" node received a closed curve within which the geometry was packed. In this case, a rectangular curve with dimensions 6x2m was inputted. The component automatically optimized the packing by iteratively finding the solution. To initiate the iteration process, the iteration count was set to 0, and a "Boolean Toggle" component was connected, see Figure 9.2. For the formworks of the tree column, the object placements within the sheets remained unchanged from solution 1200, indicating that the most optimal packing had been achieved. Once the objects were

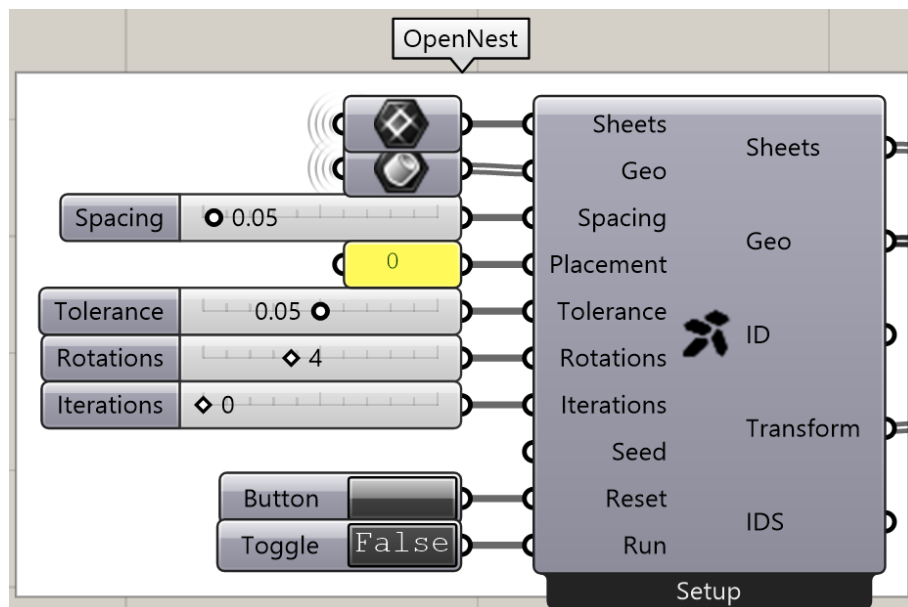


Figure 9.2: OpenNest component in Grasshopper

packed, they could be "baked" into the Rhino environment (baking refers to transferring Grasshopper objects to Rhinoceros 3D) and exported as a CAD file for the CNC machine.

The formwork pieces for the column were designed with varying heights ranging from 0.5m to 1.2m. To enhance the convenience of the construction process, packed geometries of the complete formwork pieces were created, incorporating both trimmed

and untrimmed sections. The trimmed pieces within the height range of 0.5m to 1.2m were individually numbered, along with the untrimmed pieces, as depicted in Figures 9.3 and 9.4. This numbering system serves as a valuable reference during construction, allowing workers to easily identify and assign each formwork piece to its corresponding section of the column. For a comprehensive overview of the generated formwork designs, please refer to Appendix D.

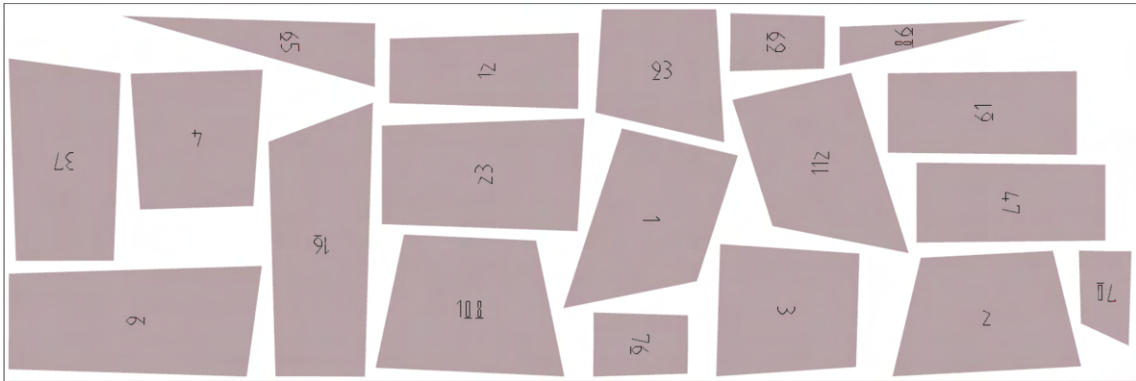


Figure 9.3: Generated small pieces of formwork using OpenNest plugin

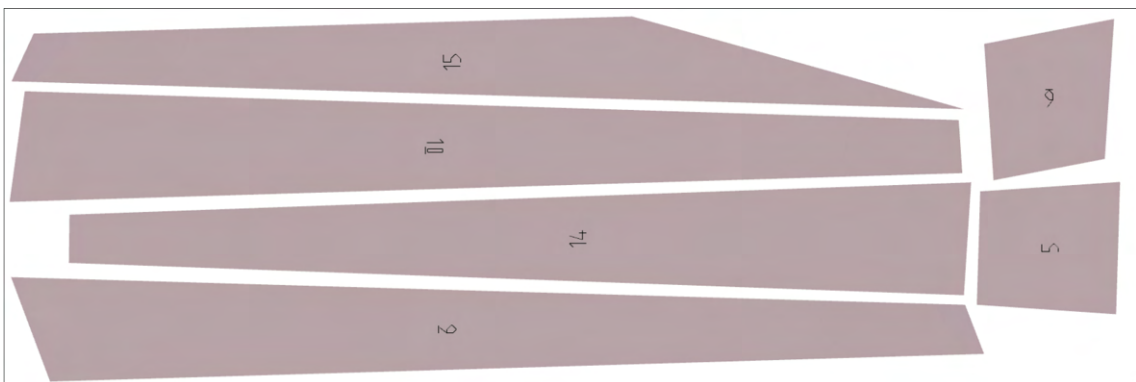


Figure 9.4: Generated big pieces of formwork using OpenNest plugin

Additionally, the axonometric view shown in Figure 9.5 and exploded axonometric views shown in Figure 9.6 were generated to further simplify the formwork construction process. These visual representations showcase the numbered whole formwork pieces, providing workers with a clear understanding of which piece belongs to each specific part of the column. This aids in maintaining accuracy and efficiency during the construction process.

Figures 9.7 and 9.8 illustrate the construction process of a column described in Section 2.2, which served as inspiration for the designed column in this thesis. The construction process of the designed column is expected to follow a similar sequence.

Also, in addressing the lateral pressure exerted by concrete in its plastic state on formwork sheathing, the implementation of supporting systems such as yokes and clamps should be discussed with a qualified contractor. To facilitate the sheathing process, a suggested approach involves the use of a finishing plastic layer on sheathing, as shown in Figure 9.9. This technique eliminates the need for concrete surface re-finishing, resulting in a more efficient and streamlined construction process.

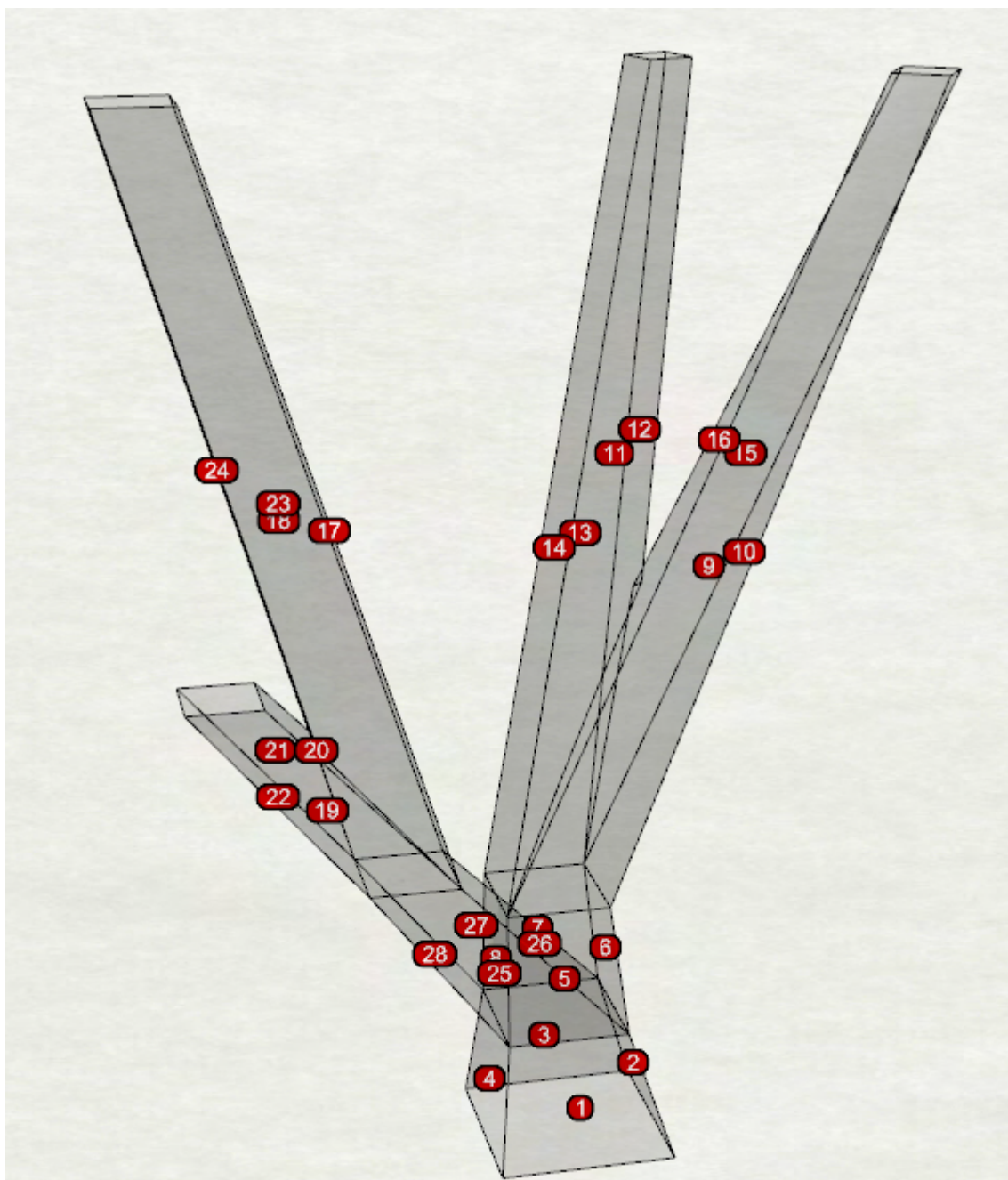


Figure 9.5: Axonometric visualization of the tree column formwork

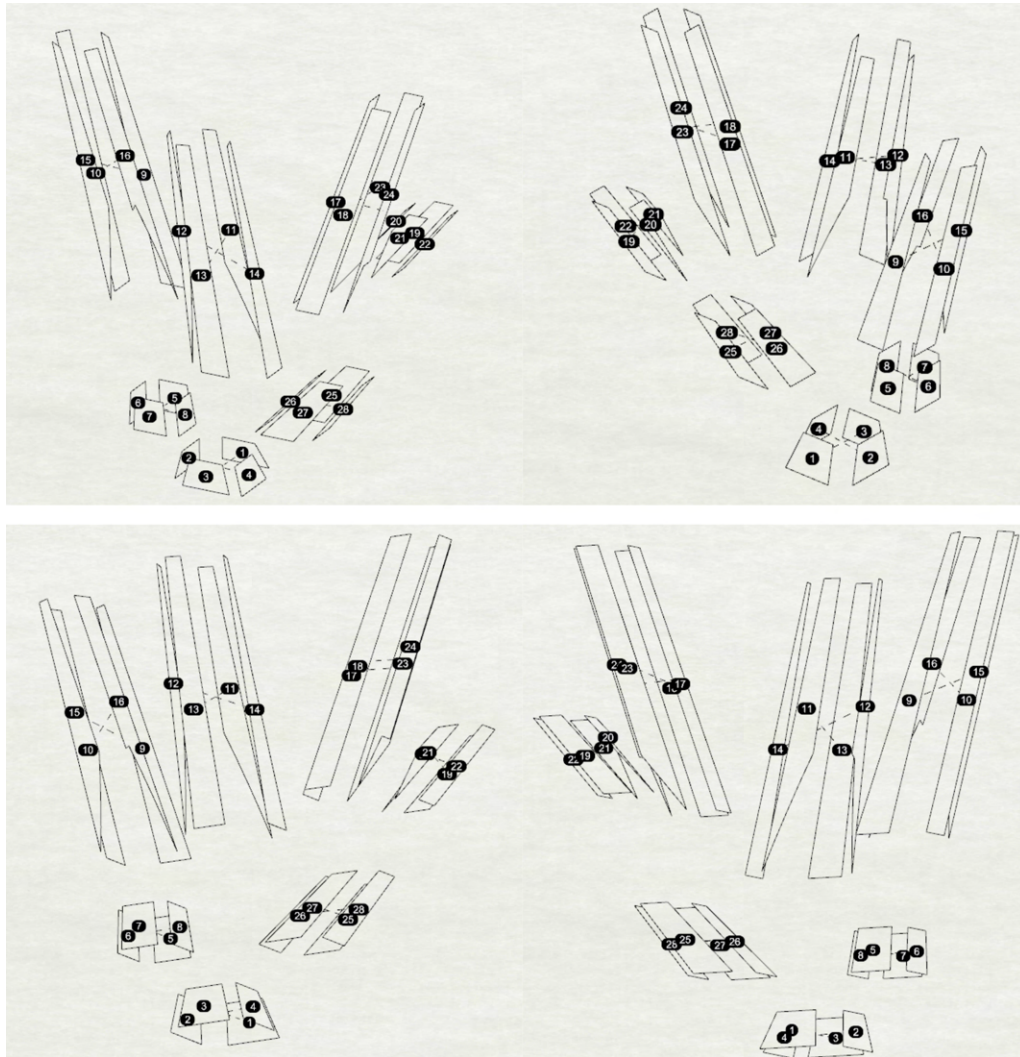


Figure 9.6: Exploded axonometric visualization of the tree column formwork



Figure 9.7: Illustration of RWPA inspirational column (see Figure 2.12) formwork



Figure 9.8: Illustration of RWPA inspirational column (see Figure 2.12) construction process



Figure 9.9: Showcase of a column formwork with a finishing plastic layer on sheathing [18]

Conclusion

This thesis aimed to explore the realm of modern tools, namely parametric modelling via visual programming and multiobjective optimization, to enhance the early-stage design process of concrete structures. The traditional challenges encountered in this process, including prolonged design timescales, conflicting construction details, and subsequent delays and complications during construction, have necessitated a more efficient and streamlined approach.

The integration of parametric modelling into the workflow has proven to be instrumental in generating and manipulating design options and solutions with greater effectiveness. This approach raises creativity, enables comprehensive exploration of the design space, ultimately saves time, increases productivity, and stimulates innovation in building engineering.

On the other hand, the implementation of multiobjective optimization addresses the complex nature of design criteria and enables deeper dives into the design option space. By formulating conflicting objectives as optimization criteria, designers can quantitatively evaluate trade-offs and identify optimal solutions that strike a balance between performance, cost, sustainability, and other crucial factors. This helps to improve decision-making and the development of more sustainable and resilient concrete structures, while also promoting interdisciplinary collaboration among designers from various fields.

Through the study conducted in this thesis, the practicality and effectiveness of these tools have been demonstrated. Parametric modelling has effectively simplified model manipulation and preparation for subsequent optimization, while plugins available within the modelling tool have provided a means to address additional aspects of building design such as structural integrity and environmental efficiency. Multiobjective optimization helped to delve deeper into the design space, enabling thorough analysis and informed selection of design options that match individual preferences.

Although this approach offers numerous advantages, such as the ability to accommodate client-driven design changes efficiently, it is not without its limitations. Computational intensity is a challenge as it can significantly slow down the performance of visual programming software and subsequently affect the integrated optimisation process. However, these challenges can be overcome through ongoing advances in computational technology.

As the construction industry continues to evolve and face new challenges, it is im-

perative to embrace technological advancements and optimized design practices. This thesis serves as a shining example of progress towards a more efficient future in building engineering. By wholeheartedly embracing this streamlined design process, stakeholders in the construction industry can effectively mitigate project risks, minimize delays, and optimize costs. Ultimately, the seamless integration of these innovative methods contributes to improved project outcomes, heightened client satisfaction, and the overall advancement of the building engineering field.

Bibliography

- [1] A.N. Baldwin et al. “Modelling information flow during the conceptual and schematic stages of building design, ” in: *Construction Management & Economics* 17.2 (1999), pp. 155–167. DOI: <https://doi.org/10.1080/014461999371655>.
- [2] Kalyan Deb et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *Evolutionary Computation, IEEE Transactions on* 6 (May 2002), pp. 182–197. DOI: 10.1109/4235.996017.
- [3] ČSN EN 1991-1-1. *Eurokód 1: Zatížení konstrukcí - Část 1-1: Obecná zatížení - Objemové tíhy, vlastní tíha a užitná zatížení pozemních staveb*. Třídící znak: 730035, 2004.
- [4] Jürgen Branke et al. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. 1st ed. Heidelberg: Springer Berlin, 2008. ISBN: 978-3-540-88908-3.
- [5] Hans-Christian Zohmann. “Genetic algorithms to solve multi-objective optimization problems: adopting NSGA-II to a periodic vehicle routing problem”. PhD thesis. Wien, 2011. URL: <https://theses.univie.ac.at/detail/14724>.
- [6] Daniel Davis. “Modelled on Software Engineering: Flexible Parametric Models in the Practice of Architecture.” In: *PhD dissertation, RMIT University* 243 (2013), p. 33.
- [7] Clemens Preisinger. “Linking Structure and Parametric Geometry”. In: *Architectural Design* 83 (2013), pp. 110–113. DOI: <https://doi.org/10.1002/ad.1564>.
- [8] Mostapha Sadeghipour Roudsari and Michelle Pak. “Ladybug: a parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design”. In: *Proceedings of the 13th International IBPSA Conference Held in Lyon, France Aug 25–30th* (2013). URL: http://www.ibpsa.org/proceedings/BS2013/p_2499.pdf.
- [9] Szymon Piasecki, Robert Szmurło, and Marek Jasinski. “Design of AC-DC Grid Connected Converter using Multi-Objective Optimization”. In: *Electrical, Control and Communication Engineering* 5 (May 2014). DOI: 10.2478/ecce-2014-0002.
- [10] Sun Zexin and Hongyuan Mei. “Robotic Form-Finding and Construction Based on the Architectural Projection Logic”. In: *IOP Conference Series: Materials Science and Engineering* 216 (June 2017), p. 012058. DOI: 10.1088/1757-899X/216/1/012058.

- [11] *CEN European Daylight Standard (EN 17037)*. 2018. URL: <https://velcdn.azureedge.net/~/media/marketing/ee/professional/28mai2019%5C%20seminar/veluxen17037tallinn28052019.pdf>.
- [12] Reza Assasi. “Parametric Design, A Historical and Theoretical Overview”. In: *Department of Architectural Science of Ryerson University (ICETAD2019)* (2019). URL: https://www.academia.edu/41325381/Parametric_Design_A_Historical_and_Theoretical_Overview.
- [13] Mohammed Makki, Milad Showkatbakhsh, and Yutao Song. *Wallacei Primer 2.0*. 2019. URL: <https://www.wallacei.com/>.
- [14] Xujing Zhang, Lichuan Wang, and Yan Chen. “Carbon Emission Reduction of Apparel Material Distribution Based on Multi-Objective Genetic Algorithm (NSGA-II)”. In: *Sustainability* 11 (May 2019), p. 2571. DOI: 10.3390/su11092571.
- [15] Coello Coello Carlos A. et al. “Evolutionary multiobjective optimization: open research areas and some challenges lying ahead.” In: *Complex & Intelligent Systems* 6 (2020), pp. 221–236. DOI: <https://doi.org/10.1007/s40747-019-0113-4>.
- [16] Sadra Fardhosseini et al. “The Cost-Effectiveness of Integrating Digital Fabrication for Concrete Formworks”. In: (Mar. 2020). DOI: 10.1061/9780784482865.114.
- [17] doc. Ing. Jan Kaňka Ph.D. “Hodnocení proslunění budov podle ČSN EN 17037”. In: (Mar. 2021). URL: <https://stavba.tzb-info.cz/denni-osvetleni-a-osluneni/21986-hodnoceni-prosluneni-budov-podle-csn-en-17037>.
- [18] *Administrativní budova Green Point, Praha Smíchov*. URL: <https://www.peri.cz/projekty/bytove-a-obcanske-stavby/administrativni-budova-green-point-praha.html>.
- [19] *Alpaca4d*. URL: <https://alpaca4d.github.io/about/index.html>.
- [20] Roland Bernath. *N°006 - Gsteig by RWPA Architecture Agency*. URL: <https://www.rwpa.ch/gsteig>.
- [21] *Biomimetic Architecture: Sagrada Familia*. URL: <https://steemit.com/architecture/@snaves/biomimetic-architecture-sagrada-familia>.
- [22] Zoey Braun and Roland Halbe. *R128 by Werner Sobek*. URL: <https://www.wernersobek.com/projects/r128/>.
- [23] Daniel Davis. *A History of Parametric*. URL: <https://www.danieldavis.com/a-history-of-parametric/>. (accessed: 15.04.2023).
- [24] *Kiwi!3D*. URL: <https://www.kiwi3d.com/>.
- [25] Walter Mair. *House with One Wall by Christian Kerez*. URL: <https://www.archdaily.com/604014/house-with-one-wall-christian-kerez>.
- [26] *Rhino in Architecture, Engineering, and Construction*. URL: <https://www.rhino3d.com/en/for/architecture/>.
- [27] David Frutos Ruiz. *Crossed House by Clavel Arquitectos*. URL: https://www.archdaily.com/617586/crossed-house-clavel-arquitectos?ad_source=myad_bookmarks&ad_medium=bookmark-open.

Appendices

A Definition of the designed model in Grasshopper

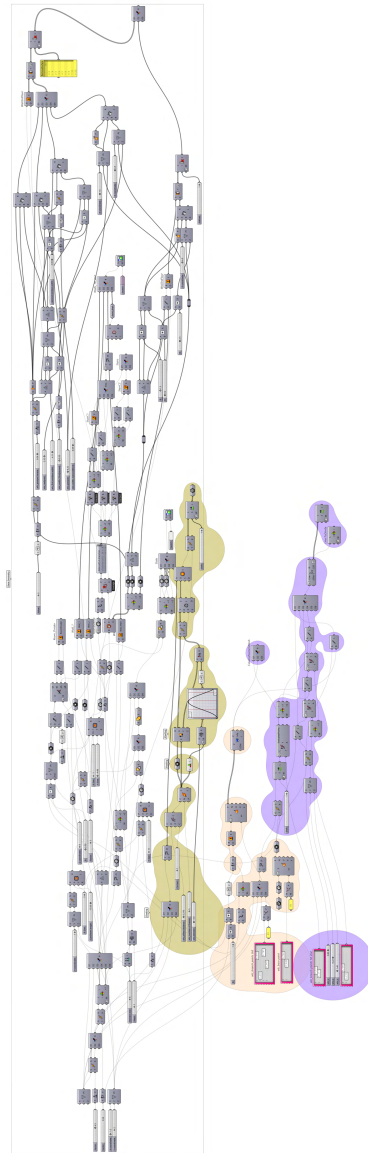


Figure 10: Definition of the designed model in Grasshopper

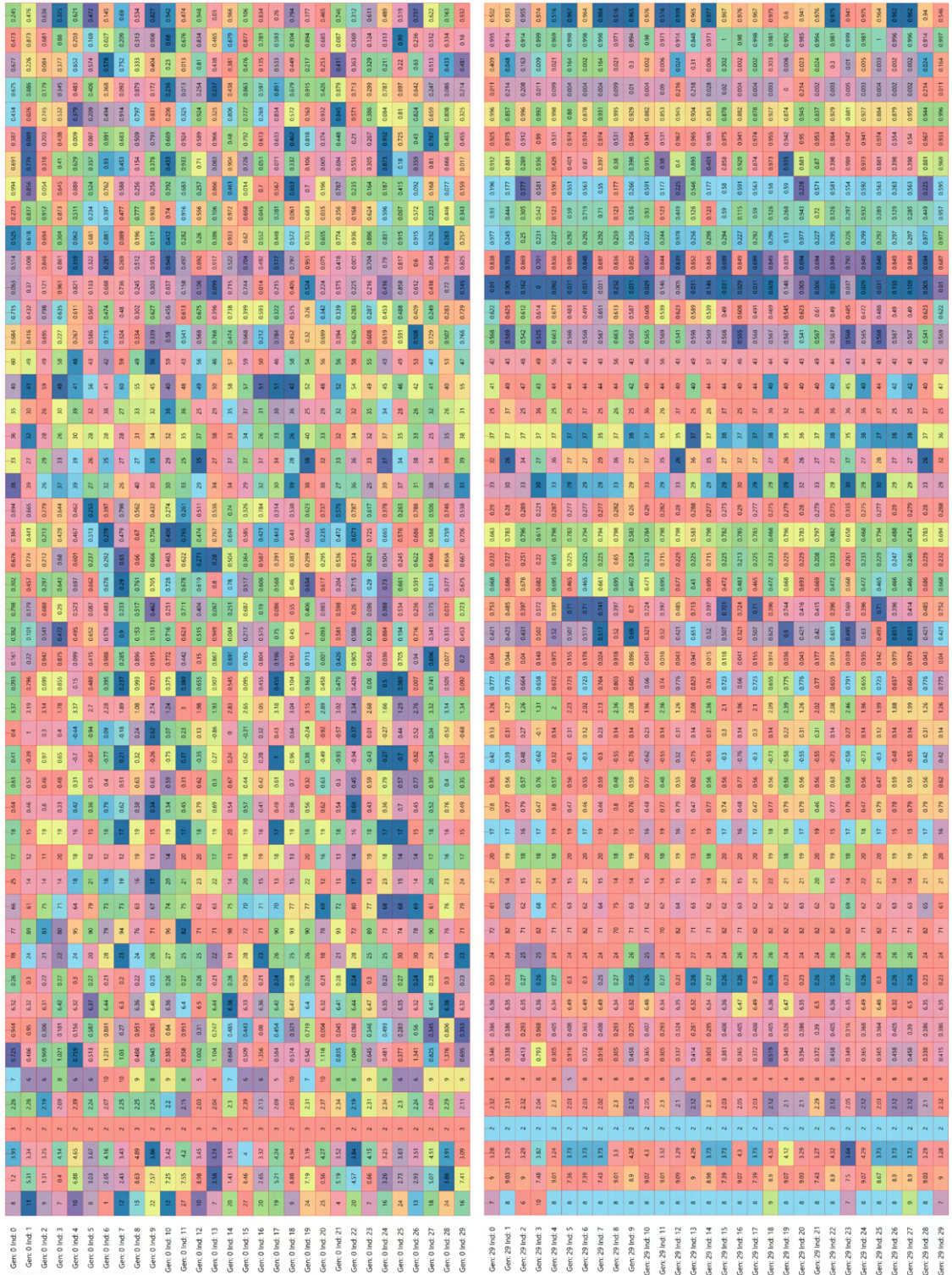


Figure 14: Decoded genomes of the first and last generations of the fourth optimization process

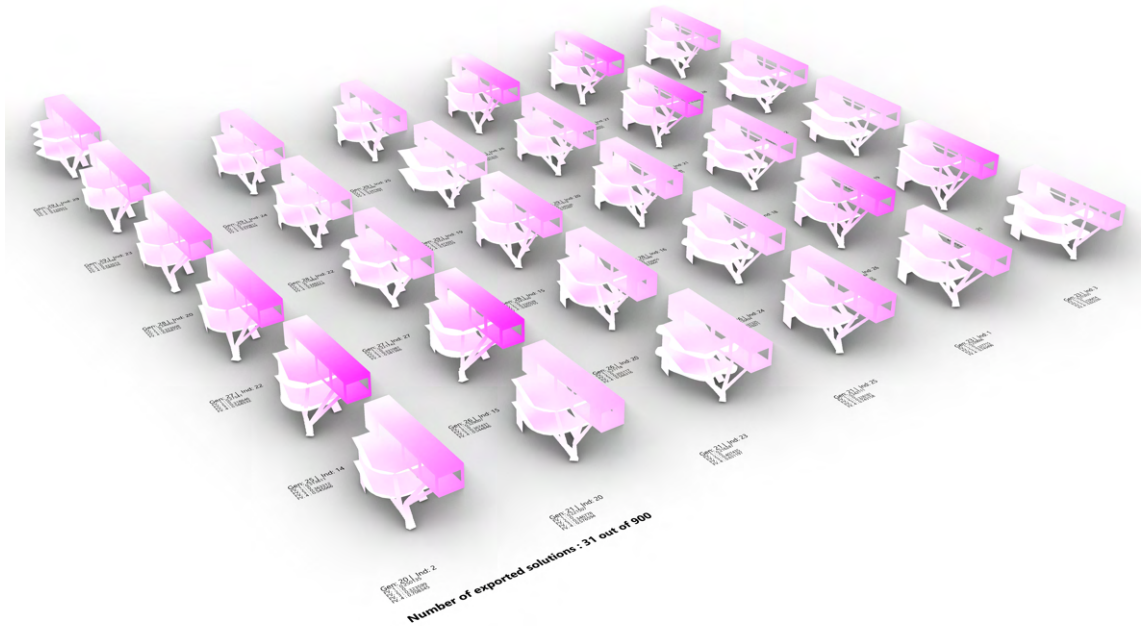


Figure 17: Pareto front solutions of the third provided optimization

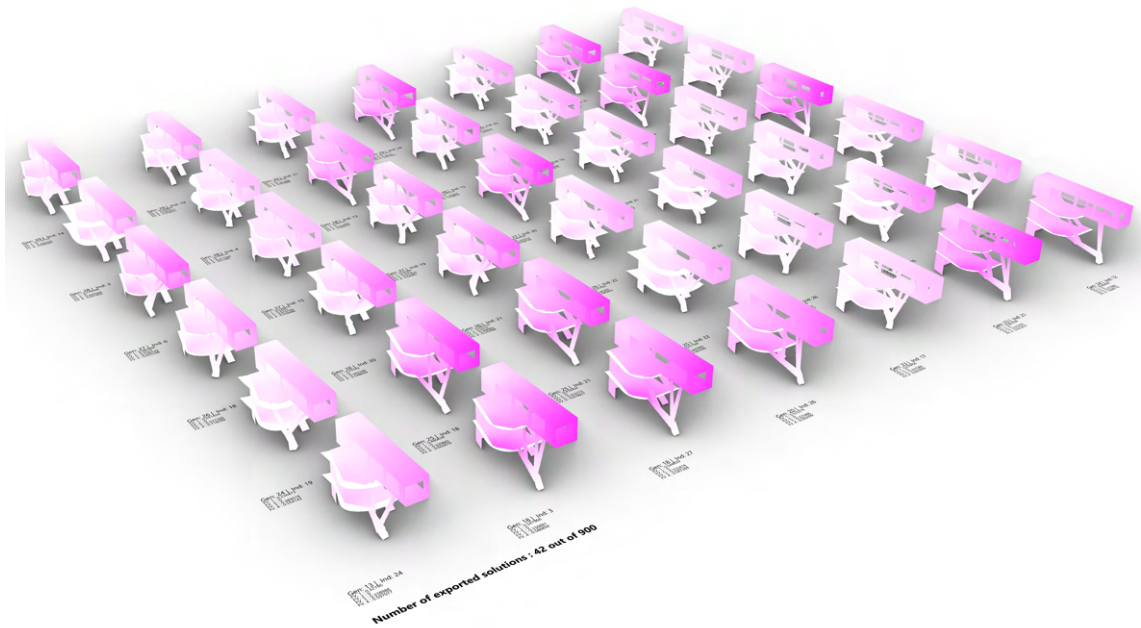


Figure 18: Pareto front solutions of the fourth provided optimization

D Generated formwork

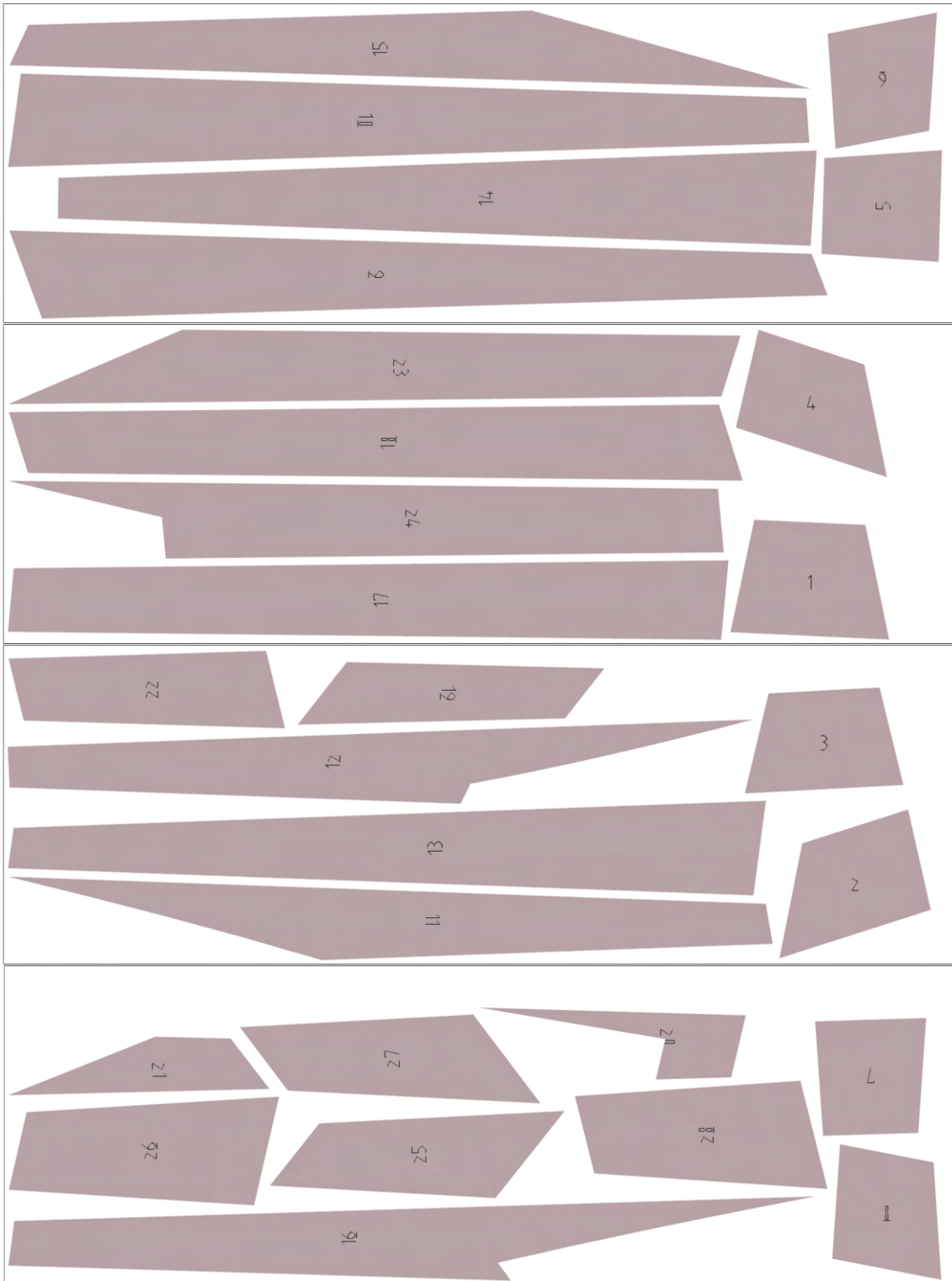


Figure 19: Generated formwork with big pieces



Figure 20: Generated formwork with small pieces