

Silje Larsen
Pauline L. Skomedal

Integrating BIM and LCA for Efficient Structural Modelling and Environmental Impact Assessment using IFC Files

Enhancing Sustainable Decision-Making in Early Design Phases

Master's thesis in Civil and Environmental Engineering
Supervisor: Sverre Magnus Haakonsen
June 2023



Norwegian University of
Science and Technology

Silje Larsen
Pauline L. Skomedal

Integrating BIM and LCA for Efficient Structural Modelling and Environmental Impact Assessment using IFC Files

Enhancing Sustainable Decision-Making in Early
Design Phases

Master's thesis in Civil and Environmental Engineering
Supervisor: Sverre Magnus Haakonsen
June 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Structural Engineering





MASTER THESIS 2023

SUBJECT AREA: Conceptual Structural Design	DATE: 08.06.2023	NO. OF PAGES: vii + 58
---	---------------------	---------------------------

TITLE:

Enhancing Sustainable Decision-Making in Early Design Phases: Integrating BIM and LCA for Efficient Structural Modelling and Environmental Impact Assessment using IFC Files

Forsterket bærekraftig beslutningstaking i tidlige designfaser: Integrering av BIM og LCA for effektiv strukturell modellering og miljøvurdering ved bruk av IFC-filer

BY:

Silje Larsen

Pauline L. Skomedal



SUMMARY:

There is a consensus among experts that the construction sector is a significant contributor to global CO₂ emissions. As the focus shifts to the environmental impact of new buildings, the construction industry has started to implement tools to control and measure the environmental impact of projects. Examining CO₂ emissions is an efficient and measurable approach for evaluating different solutions.

In modern construction projects, Building Information Models (BIM) are widely employed, starting with limited information in the early stages and gradually increasing in detail. The emissions assessment often involves a Life Cycle Assessment (LCA), which analyzes the contribution of various building elements to emissions. Utilizing visual programming languages during the initial design phase may be an optimal approach for integrating BIM with LCA.

This thesis aims to develop a methodology that efficiently processes a BIM model as input, analyzes the structural framework, optimizes the cross sections, and ultimately identifies solutions with the lowest LCA footprint. Working with IFC files presents inherent challenges, particularly in relation to element misclassification, which often necessitates substantial clean-up efforts within the file. Furthermore, ensuring structural stability may be challenging when extracting lines from the IFC file. However, by employing modifications involving manipulating geometry and establishing connections within acceptable tolerances, it is possible to achieve the desired stability of the structure. The method is applied and tested on three different buildings of varying sizes.

The research shows that utilizing IFC files in structural modeling and LCA optimization is undoubtedly significant. Using the methodology presented, the results from the case studies show a reduction of over 60% of the LCA footprint compared to the original models. In this way, structural engineers may effectively analyze BIM models created by others during the early design phases. By making specific modifications, they can gain valuable insights into the environmental impact of these models, enabling informed decision-making throughout the project's life cycle.

RESPONSIBLE TEACHER: Sverre Magnus Haakonsen

SUPERVISOR(S): Sverre Magnus Haakonsen

CARRIED OUT AT: Department of Structural Engineering, Norwegian University of Science and Technology

Acknowledgements

This master's thesis was written and delivered in the spring of 2023. It is written on behalf of the Department of Structural Engineering at the Norwegian University of Science and Technology (NTNU) as the concluding part of our Master of Science.

We want to express our heartfelt gratitude to our supervisor, Sverre Magnus Haakonsen, for his invaluable guidance, support, and expertise throughout this project. His insightful feedback and dedication have been crucial in shaping our research. We are also deeply grateful to Associate Professor Marcin Luczkowski for his invaluable contribution. Furthermore, we extend our sincere appreciation to the department for their continuous support, encouragement and for providing an environment conducive to learning and growth throughout our five years at NTNU. We are also grateful for the delightful cups of coffee that have fueled our endeavors. We want to acknowledge our classmates and colleagues for their camaraderie and support throughout this journey. The exchange of ideas and shared experiences have created a stimulating and collaborative atmosphere. Also, we would like to express our sincere gratitude to COWI for providing the necessary reports and IFC files, which were instrumental in conducting this research. In addition, we would like to thank our family and friends for all their support throughout our years at NTNU.

Finally, we want to extend a special thank you to each other for our dedication, hard work, and effective collaboration. Our combined efforts and mutual support have been crucial to overcome challenges and achieving our research goals.

Silje Larsen and Pauline L. Skomedal

Abstract

There is a consensus among experts that the construction sector is a significant contributor to global CO₂ emissions. As the focus shifts to the environmental impact of new buildings, the construction industry has started to implement tools to control and measure the environmental impact of projects. Examining CO₂ emissions is an efficient and measurable approach for evaluating different solutions.

In modern construction projects, Building Information Models (BIM) are widely employed, starting with limited information in the early stages and gradually increasing in detail. The emissions assessment often involves a Life Cycle Assessment (LCA), which analyzes the contribution of various building elements to emissions. Utilizing visual programming languages during the initial design phase may be an optimal approach for integrating BIM with LCA.

This thesis aims to develop a methodology that efficiently processes a BIM model as input, analyzes the structural framework, optimizes the cross sections, and ultimately identifies solutions with the lowest LCA footprint. Working with IFC files presents inherent challenges, particularly in relation to element misclassification, which often necessitates substantial clean-up efforts within the file. Furthermore, ensuring structural stability may be challenging when extracting lines from the IFC file. However, by employing modifications involving manipulating geometry and establishing connections within acceptable tolerances, it is possible to achieve the desired stability of the structure. The method is applied and tested on three different buildings of varying sizes.

The research shows that utilizing IFC files in structural modeling and LCA optimization is undoubtedly significant. Using the methodology presented, the results from the case studies show a reduction of over 60% of the LCA footprint compared to the original models. In this way, structural engineers may effectively analyze BIM models created by others during the early design phases. By making specific modifications, they can gain valuable insights into the environmental impact of these models, enabling informed decision-making throughout the project's life cycle.

Sammendrag

Det er enighet blant eksperter om at byggesektoren er en betydelig bidragsyter til globale CO₂-utslipp. Ettersom fokuset skifter mot miljøpåvirkningen fra nye bygninger, har byggebransjen begynt å implementere verktøy for å kontrollere og måle miljøpåvirkningen fra prosjekter. Å undersøke CO₂-utslipp er en effektiv og målbar tilnærming for å vurdere ulike løsninger.

I moderne byggeprosjekter er Bygningsinformasjonsmodeller (BIM) mye brukt, der man starter med begrenset informasjon i de tidlige fasene og gradvis øker detaljnivået utover i prosjektet. Vurderingen av utslippene involverer ofte en Livssyklusanalyse (LCA), som analyserer bidraget fra ulike bygningselementer til utslippene. Bruk av visuelle programmeringsspråk i den tidlige designfasen kan være en optimal tilnærming for å integrere BIM med LCA.

Denne masteroppgaven har som mål å utvikle en metodikk som effektivt prosesserer en BIM-modell som inndata, analyserer den statiske modellen, optimaliserer tverrsnittene og identifiserer til slutt løsninger med lavest mulig LCA-fotavtrykk. Arbeidet med IFC-filer byr på utfordringer, spesielt med hensyn til feilklassifisering av elementer, noe som ofte krever betydelig opprydding i filen. Videre kan det være utfordrende å sikre strukturell stabilitet når man ekstraherer linjer fra IFC-filen. Imidlertid er det mulig å oppnå ønsket stabilitet i strukturen ved å bruke endringer som innebærer manipulering av geometri og etablering av forbindelser innenfor akseptable toleranser. Metoden blir anvendt og testet på tre forskjellige bygninger av varierende størrelse.

Oppgaven viser at bruk av IFC-filer i strukturell modellering og LCA-optimalisering er utvilsomt betydelig. Ved å bruke den presenterte metodikken viser resultatene fra case-studiene en reduksjon på over 60% av LCA-fotavtrykket sammenlignet med de opprinnelige modellene. På denne måten kan en ingeniør effektivt analysere BIM-modeller som er opprettet av andre i de tidlige designfasene. Ved å gjøre spesifikke endringer kan de få verdifulle innsikter i miljøpåvirkningen fra disse modellene, noe som muliggjør informert beslutningstaking gjennom hele prosjektets livssyklus.

Table of Contents

Acknowledgements	i
Abstract	ii
Sammendrag	iii
List of Figures	v
List of Tables	vi
1 Introduction	1
2 Theory	3
2.1 BIM-LCA implementation today	3
2.2 IFC - Industry Foundations Classes	4
2.3 LCA - life cycle assessment	5
2.4 EPD - Environmental Product Declaration	7
2.5 Materials	8
3 Software	10
3.1 Python - IfcOpenShell	10
3.2 Rhino	10
4 Method	13
4.1 IFC to lines	13
4.2 Creating the structural model	16
4.3 Optimization	21
4.4 Analysis and results	25
5 Cases	30
5.1 Case 1 - Tennebekk	30
5.2 Case 2 - Strandhagen	34

5.3 Case 3 - Buebygget	40
6 Discussion	46
7 Conclusion	50
8 Future work	51
Bibliography	52
Appendix	55
A Grasshopper files	55
B Python file	55
C Tennebekk	56
D Strandhagen	57

List of Figures

2.1 Steps in a Life Cycle Assessment.	5
2.2 Building life cycle stages.	6
4.1 Flowchart: Methodology.	13
4.2 Flowchart: IFC file to Grasshopper using Python.	14
4.3 Flowchart: IFC file to Grasshopper using GeometryGym.	15
4.4 Sorting elements in Grasshopper.	16
4.5 Flowchart: Structural model.	17
4.6 Adjusting geometry to common nodes	18
4.7 Illustration of original IFC elements	19
4.8 Illustration of elements with artificial beams	20
4.9 Flowchart: Optimization.	21
4.10 Wallacei X in Grasshopper	23

4.11	Illustration of the simple frame.	25
4.12	Simple Frame: Wallacei Settings.	26
4.13	Simple Frame: Wallacei Analytics.	27
4.14	Simple Frame: Wallacei Selection.	28
4.15	Simple Frame: Geometry	28
5.1	Illustration of Cases.	30
5.2	Case 1: Geometry to common nodes	31
5.3	Case 1: Geometry	33
5.4	Case 1: Color mapping	34
5.5	Case 2: Geometry to common nodes	36
5.6	Case 2: Rotating columns.	36
5.7	Case 2: Moment distribution of columns.	37
5.8	Case 2: Geometry	38
5.9	Case 2: Color mapping	40
5.10	Case 3: Removing the cellar.	42
5.11	Case 3: Artificial beams.	43
5.12	Case 3: Geometry to common nodes	43
5.13	Case 3: Color mapping.	45

List of Tables

2.1	Existing BIM to LCA software and their limitations	3
4.1	Loads for typical scenarios	22
4.2	The materials selected for optimization and their EPDs.	25
4.3	Simple Frame: Results from the analysis.	29
5.1	An overview of the case study.	30
5.2	Case 1: Key data used in the analysis.	32
5.3	Case 1: Results from the analysis.	33
5.4	Case 2: Key data used in the analysis.	38

5.5	Case 2: Results from the analysis.	39
5.6	Case 3: Results from the manual analysis.	44

1 Introduction

As the focus shifts to the environmental impact of new buildings, the construction industry has started to implement tools to control and measure the environmental impact of projects. Examining carbon dioxide (CO₂) emissions is an informal yet efficient and measurable approach for evaluating different solutions. There is a consensus among experts that the construction sector is a significant contributor to global CO₂ emissions. According to the Global Alliance for Buildings and Construction, buildings and construction account for nearly 40% of global energy-related CO₂ emissions (IEA, 2019). This includes the emissions from the construction phase, such as the production of building materials and construction activities, the operation of buildings, and the end-of-life phase, such as demolition and disposal of materials.

Today, the majority of larger construction projects have a Building Information Model (BIM) with different levels of information saved inside. In the early stages, the model includes less information, and as the project proceeds, the detail level increases. There are different ways to calculate the emissions; however, a widely used method is to look at the Life Cycle Assessment (LCA) and how the different elements of the building lead to emissions. The idea of integrating an LCA analysis in BIM is widely investigated. As BIM models may include quantities and possibly information on materials, the idea of extracting CO₂ quantities automatically is achievable. Therefore, using BIM in the assessment may increase the quality of the building (Soust-Verdaguer et al., 2020).

Tam et al., 2023 have investigated the best methods to integrate BIM and LCA in different scenarios. Their study is relevant to this thesis as it identifies the optimal approach for integrating BIM with LCA. It concludes with utilizing visual programming languages during the initial design phase. Due to the significant design decisions made during this stage, the integration of BIM and LCA becomes crucial (Budig et al., 2021).

Therefore, this thesis aims to develop a code and methodology to efficiently process a BIM model as input, analyze the structural framework, optimize the cross sections, and ultimately identify solutions with the lowest LCA footprint. The objective is to streamline decision-making processes regarding the environmental impacts of the building's structural framework during the initial design phase. This is achieved through the provision of visualizations and data, which serve to facilitate and enhance the decision-making process for an engineer.

The thesis continues with Section 2, which provides a comprehensive overview of the relevant theoretical foundations. This section offers an in-depth exploration of current knowledge and advancements in the field. Furthermore, it includes detailed explanations of key concepts such as IFC files, LCA, Environmental Product Declaration (EPD), and essential information regarding selected materials. Following the theory section, Section 3 presents a thorough examination of the software tools and plugins employed in the thesis. This section aims to provide a deeper understanding of each software's functionality and the potential contributions they offer to the research. Subsequently, in Section 4, the methodology is outlined, highlighting the common steps implemented across all cases. Flow charts are included to visually illustrate the process, facilitating comprehension and serving as a helpful guide through each step. Moving on to Section 5, three cases are presented, each following the established methodology. This consistent approach aims to ensure coherence and clarity in the explanations. Section 6 provides a comprehensive examination

of the methodology, case study, and the significance of the results. The thesis continues with concluding remarks in Section 7, addressing the findings and complexities encountered throughout the research. Finally, Section 8, briefly discusses potential avenues for future research, offering suggestions and possibilities to explore further in the field.

2 Theory

2.1 BIM-LCA implementation today

Tools to integrate BIM and LCA already exist, and they all have different advantages and disadvantages. This is discussed and systematized in the Danish article, *A BIM-based LCA tool for sustainable building design during the early design stage* (Kamari et al., 2022). Table 2.1 highlights the limitations of the various programs examined in the study, shedding light on the industry's existing gaps and areas that require improvement.

Table 2.1: Existing BIM to LCA software and their limitations (Kamari et al., 2022).

Program	Country	Limitations
One-Click LCA	Finland	Difficult to find information
TallyTM	USA	<ul style="list-style-type: none">- It is specific for Autodesk Revit software- The inventory data as the LCIA methods cannot be changed or updated by the user.
LCAbyg	Denmark	<ul style="list-style-type: none">- Under development- Focus only on the A1–A3, B4, B6 and C3, C4 life cycle stages- Danish user interface- Manual extraction of the BIM model quantities- Manual mapping of the building materials- Use of outdated normalization factors
Byggsektorns Miljöoberäkning-verktyg(BM)	Sweden	<ul style="list-style-type: none">- Environmental assessment of the production stage- Swedish material database- No clear calculation methods- No possibility of selecting a few materials at once while assigning the EPDs- The user is forced to buy a special template destined for VICO software- Lack of description of the different mapping methods- No possibility of extracting data from a Revit model

Tam et al., 2023 have investigated, in different scenarios, the optimal method to integrate BIM and LCA. They have considered "A) environmental impact assessment in the early design stage, B) environmental impact assessment in the detailed design stage, and C) environmental impact assessment in the construction design stage." By evaluating the different types of integration with different performance indicators through weighting results, the study gets numerical results and scores.

As their study is from 2023, it may be assumed that the methods accounted for is relevant, combined with the fact that no other methods besides the ones mentioned in the study of Tam et al., 2023 have appeared in the study of the literature. The discussed methods to obtain the environmental assessment results are T1: Exporting bill of quantities (BOQ) into Excel, T2; Exporting BOQ into a dedicated LCA tool, T3: Adopting LCA plugin for BIM-software, for example, tally, T4: to use visual programming software and T5: Including LCA information in BIM objects.

After judging the methods in the three different project stages, T4 gained the highest score in both stage A) and C), and in stage B) T3 and T4 obtained the same score.

The Type 4 method is defined as:

1) creating a BIM model for a concerned building, 2) extracting both geometrical and physical data of building components and data of embodied impact factors, and inputting into the visual programming languages' environment (such as in Dynamo), and 3) establishing the relationships between material data and corresponding embodied impact factors for conducting calculation on the embodied environmental impact of a building in visual programming languages' environment.

This aspect is particularly fascinating as it identifies the optimal approach to integrating BIM with LCA and draws a conclusion favoring utilizing visual programming languages during the early design stage.

2.2 IFC - Industry Foundations Classes

Industry Foundation Classes is a file format used in the architecture, engineering, and construction (AEC) industry for the exchange of Building Information Model (BIM) data (BuildingSmart, 2023). The IFC file format, maintained by the buildingSMART organization, is widely recognized as the predominant non-proprietary software for BIM. Non-proprietary file formats are vendor-neutral, allowing them to be accessed and modified by various software applications. These file formats are readable text files that encompass essential information for generating 3D models of buildings.

IFC is a file format that establishes international standards to import and export building objects, including geometry, materials, properties, and relationships between objects. It is used as a common digital representation throughout the design and construction of buildings (ISO, 2018). The ultimate objective of the IFC file format is to improve communication, productivity, delivery time, and quality throughout the life cycle of a building. It provides interoperability between different software applications (Grutters, 2022). IFC files may be exported from BIM software, such as Autodesk Revit, Graphisoft ArchiCAD, Tekla Structures, and Trimble SketchUp, and imported into other BIM software applications for further use. IFC is also compatible with some Computer-Aided Design (CAD) software applications, such as AutoCAD and Rhino/Grasshopper, through plugins.

There are several benefits to using the IFC file format; on the other hand, it is also associated with some challenges, such as:

- **Complexity:** IFC is a complex file format. It might be difficult to understand and implement, resulting in errors and interoperability issues.
- **Lack of standardization:** While IFC is an open standard maintained by buildingSMART, there is still a lack of standardization across software applications, which might result in errors.

- File size: IFC files may be quite large, which may make them difficult to work with and transfer between different software applications.
- Incomplete data: Depending on the level of detail included in the IFC file, some data may be missing or incomplete, leading to errors.
- Limited functionality: IFC files may not support all of the features and functionalities available in the original software application, which may limit the usefulness of the file. The file format does not intended to replace the software, it is typically used to exchange information from one party to another (Grutters, 2022).

2.3 LCA - life cycle assessment

Life Cycle Assessment (LCA) is a widely used tool for identifying opportunities to reduce the environmental impacts of products and processes and supporting decision-making in sustainable product and process design. LCA is a method for assessing the environmental impacts of a product, service, or process throughout its entire life cycle, from raw material extraction to end-of-life disposal or recycling.

LCA typically includes the following stages (European Commission, 2023):

- Goal and Scope Definition: This stage defines the purpose and boundaries of the assessment, including the functional unit, system boundaries, and impact categories.
- Life Cycle Inventory: This stage involves collecting and quantifying data on the inputs, outputs, and environmental impacts of the product, service, or process.
- Life Cycle Impact Assessment: This stage involves evaluating the potential environmental impacts of the product, service, or process across a range of impact categories, such as climate change, water use, and human toxicity.
- Interpretation: This stage involves interpreting the results of the LCA and drawing conclusions about the environmental performance of the product, service, or process.

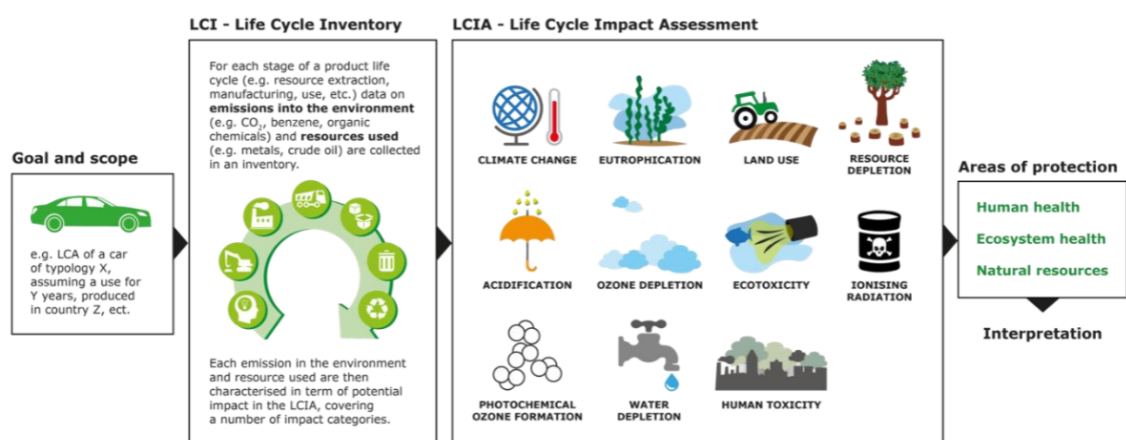


Figure 2.1: Life Cycle Assessment steps, with goal and scope definition, life cycle inventory, life cycle impact assessment and interpretation (Sala et al., 2016).

In the European markets, the building life cycle stages are defined by EN 15978 and EN 15804 standards. These standards provide a framework for defining the building life cycle stages of the analysis and are shown in Figure 2.2 taken from Shaun, 2023.

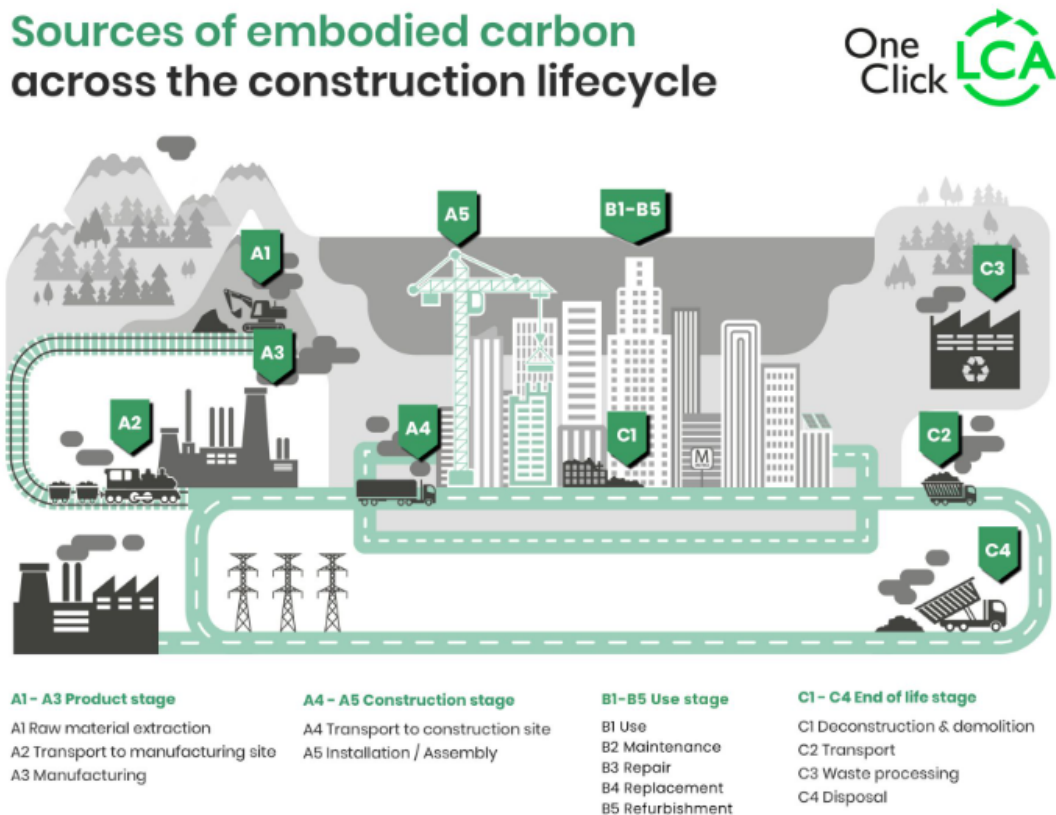


Figure 2.2: Building life cycle stages defined in EN standards, illustration taken from One Click LCA (Shaun, 2023).

LCA is often used in the product design and development phase. It may help inform decisions about material selection, manufacturing processes, and packaging design to minimize the environmental impact of a product. LCA should be used in the conceptual phase to make better decisions by changing the building design to decrease the environmental impacts early (Rezaei et al., 2019). Conducting an LCA during the early stages of product development may help ensure that environmental considerations are integrated into the design process, which may result in more sustainable products and processes.

An LCA provides valuable information about the environmental impact of a product or process that could be used to inform decision-making and improve environmental performance (Curran, 2006). Such as:

- Identification of environmental hotspots: An LCA may identify the stages of a product’s life cycle with the greatest environmental impact, known as *hotspots*. This information may be used to prioritize environmental improvements and guide product design decisions (Guinée et al., 2011).

-
- Comparison of alternatives: An LCA may be used to compare the environmental impact of different product designs, materials, and processes. This helps to identify sustainable options (Curran, 2006).
 - Quantification of environmental impact: An LCA provides a quantitative estimate of the environmental impact of a product, process, or service in terms of resource consumption, emissions, and other environmental factors (Curran, 2006).
 - An LCA may be used to evaluate the environmental performance of a product or process against industry standards or other products in the same category (Curran, 2006).

2.4 EPD - Environmental Product Declaration

EPD stands for Environmental Product Declaration. It is a third-party verified and registered document that provides transparent and verified information on the environmental impacts of a product throughout its entire life cycle (EPD-Norway). An EPD is based on the results of an LCA and therefore includes information on a product's environmental performance in a standardized format.

An EPD typically details the product's raw material extraction, manufacturing processes, packaging, transportation, use, and disposal or recycling. The boundaries for the analysis are defined from A1-C4, as illustrated in Figure 2.2. Additionally, an EPD includes environmental impact data on parameters within these boundaries, such as global warming potential (GWP), acidification potential of land and water (AP), eutrophication potential (EP), and more.

They may be used by consumers, architects, and designers to make informed decisions when selecting products based on their environmental impact. They may also be used by manufacturers to demonstrate their commitment to sustainable practices and differentiate their products in the marketplace.

The development of an EPD is guided by international standards such as ISO 14025 and EN 15804, which specify the requirements for the content, format, and verification of EPDs. Developing an EPD is typically conducted by a third-party organization. It involves data collection, analysis, and review to ensure the accuracy and reliability of the information presented in the EPD.

There are different types of EPDs, including product-specific EPDs and average EPDs (EPD-Norway, 2022).

- A product-specific EPD gives the environmental impact for one product from only one supplier. The EPD may give one or more representative scenarios for transportation, installation, use, and disposal.
- An average EPD for one product is made out of the average product when multiple manufacturers work together to declare the same product type. There are no requirements for variation between the manufacturers as long as the product is identical.

2.5 Materials

To narrow down the complexity of including all existing building materials, this thesis will focus on structural systems made of concrete, steel, and timber and various cross sections of these materials.

2.5.1 Concrete

Concrete is a construction material composed of cement, fine aggregates, and coarse aggregates mixed with water which hardens with time. It is the highest consumed material after water. The material is inexpensive and widely available around the globe if compared to other materials such as steel and other construction materials (Anand, 2016). It is used in a majority of all bridges, roads, dams, and constructions, although it also releases an extreme amount of CO₂ each year (Ramsden, 2020).

The cement sector holds the position of being the world's third-largest industrial energy consumer, accounting for 7% of industrial energy usage (IEA, 2018). Additionally, it ranks as the second-largest industrial emitter of CO₂, contributing approximately 7% of global emissions.

Concrete may withstand temperatures better than timber and steel (Anand, 2016). Some industrial wastes may be recycled as a substitute for cement or aggregate. This includes fly ash, waste glass, and even ground vehicle tires in concrete. Thus concrete production may reduce environmental impacts due to industrial waste. Concrete has the benefit of having a large compressive strength, although rather small tension strength. For construction matters, reinforced concrete by using steel is common, giving the material better tensile strength.

The concrete industry has demonstrated noteworthy sustainability advancements, particularly in reducing CO₂ emissions during cement production. Since 1990, the cement sector has reduced carbon emissions by 19% (Global Cement and Concrete Association, 2023). Furthermore, the industry has launched Climate Ambition, aiming to achieve carbon-neutral concrete by 2050 in alignment with global climate targets. This initiative allows concrete suppliers, designers, and clients to leverage concrete's potential to minimize carbon footprints through various measures. These include optimizing concrete mixes, efficiently utilizing materials, and recognizing the capacity of concrete to store CO₂ in structures permanently.

2.5.2 Structural steel

Steel is a popular choice of building material due to its strength, durability, and other advantages. The high strength-to-weight ratio of steel allows for efficient structural designs, providing flexibility in architectural and engineering possibilities (AISC, 2023). This ratio means that steel may offer significant strength while maintaining a relatively low weight, making it suitable for constructing large and complex buildings.

Additionally, steel exhibits excellent durability, offering long-term performance and structural integrity. It is highly resistant to fire, providing enhanced safety in the event of a fire, and is also

corrosion-resistant. This helps to maintain its structural properties over time. These characteristics contribute to the widespread use of steel in various construction applications.

Structural steel is a highly sustainable material that is 100% recyclable, making it circular for generations and contributing to its environmental sustainability (AISC, 2023). However, the production of steel has a significant environmental impact. The process of mining and extracting iron ore, as well as the manufacturing of steel, requires large amounts of energy and generates greenhouse gas emissions, including CO₂, which contribute to climate change (Dudka, 1997).

2.5.3 Timber

Timber, as a constructive material, has several special attributes compared to steel and concrete. It is lightweight and has a high strength-to-weight ratio, making it an ideal construction material. In terms of service life, timber structures may have a long lifespan if maintained properly. The service life of timber varies depending on the wood species, the environment it is exposed to, and the protective coating. According to a study published in the *Journal of Cleaner Production* (Chen and Taylor, 2020), timber structures may last over 100 years with proper maintenance.

Regarding fire resistance, timber has been historically associated with higher fire risk. However, modern building techniques and fire-retardant coatings have greatly improved the fire resistance of timber structures. Fire-retardant coatings significantly improve the fire resistance of timber structures (Moschas and Steirou, 2013).

From an environmental perspective, timber is a renewable resource that may be sustainably harvested. The use of timber in construction may help to reduce greenhouse gas emissions, as trees absorb CO₂ from the atmosphere as they grow. Additionally, timber may be recycled and repurposed at the end of its service life (Dodoo et al., 2009), reducing waste and further reducing its environmental impact.

There are several types of timber used in construction. This includes softwood, hardwood, and mass timber products such as glulam and cross-laminated timber (CLT). In larger building constructions, mass timber is used as an alternative to steel and concrete in the structural frame (Crawford and Cadorel, 2017). Glulam and CLT are engineered timber products becoming increasingly popular in construction due to their strength, low environmental impact, and versatility.

The calculation of GWP in LCA involving timber is a subject of ongoing debate and lacks consensus. A review conducted by Z. Duan et al. on LCA of mass timber construction examines various models and perspectives (Duan et al., 2022). Some argue that timber is a net carbon sink, leading to negative emissions. There are differing viewpoints regarding the carbon impact of timber in situations of decay or incineration. Some contend that such processes ultimately release carbon into the environment, potentially resulting in carbon neutrality or even carbon positivity. This consideration considers emissions associated with extraction, manufacturing, and transportation. In this thesis, the GWP of timber members is calculated as negative, aligning with the approach taken in Norwegian EPDs for mass timber (The Norwegian EPD Foundation, 2018). It is important to note that the calculation of GWP in relation to timber in LCA remains a complex and evolving topic, and different perspectives exist within the scientific community.

3 Software

3.1 Python - IfcOpenShell

The Industry Foundation Classes (IFC) file format, commonly used in the construction industry, may be effectively manipulated and processed using the IfcOpenShell library. IfcOpenShell is an open-source software library primarily designed for coding in C++, which aligns with the C++ origins of the IFC format. However, it also provides the option to utilize Python programming for working with IFC files.

By leveraging the capabilities of IfcOpenShell, developers may access a range of attributes and functionalities to read, edit, access, and even create IFC files using Python code. This integration allows for efficient and flexible handling of IFC files, enabling tasks such as extracting information, modifying geometry, accessing property sets, and performing other operations as needed. The Python interface provided by IfcOpenShell facilitates seamless interaction with IFC data, offering a powerful tool for working with the IFC file format in a programming environment.

3.2 Rhino

Grasshopper, developed by Robert McNeel & Associates, is a plugin for Rhino 7 that provides an algorithms-aided design (AAD) environment (McNeel et al., 2023). It works with various plugins, including GeometryGym for working with IFC files, Karamba3D for structural analysis, and Wallacei as an optimization tool.

Grasshopper allows engineers to program geometric elements using a visual programming interface. By establishing relationships between different components, such as points and lines, changes will automatically update related elements. This flexibility makes Grasshopper well-suited for the early design phase, where multiple design options are explored.

One of the key features of Grasshopper is the ability to set parameters as variable values using sliders. For instance, the height, width, and length of a building could be defined as variables, enabling engineers to explore numerous design options quickly. This parametric approach significantly speeds up the design process, as changes could be made without manually redrawing the entire model each time.

Grasshopper also benefits from a wide range of free plugins that extend its capabilities beyond basic 3D modeling. These plugins include tools for structural analysis (e.g., Karamba3D), optimization (e.g., Wallacei), and interoperability (e.g., GeometryGym). The availability of these plugins allows engineers to perform complex tasks within the Grasshopper environment, further enhancing its usefulness in engineering workflows.

3.2.1 GeometryGym

GeometryGym is a versatile software that offers a comprehensive set of interoperability tools for BIM and structural analysis. It provides utilities and plugins that seamlessly integrate with

various design and analysis programs such as Rhino3D, Grasshopper, Revit, Tekla, Navisworks, and several structural analysis software (GeometryGym, 2023).

By utilizing the GeometryGym plugins, engineers and designers may access and utilize the information embedded in IFC files. This includes extracting and manipulating geometry, properties, and structural elements from the BIM models. The plugins provide tools for seamlessly integrating BIM data into the design and analysis workflows of programs like Rhino3D and Grasshopper.

3.2.2 Karamba3D

Karamba3D is a powerful Finite Element Analysis (FEA) plugin tool within the Grasshopper environment. The program is developed for parametric design in the early stages of the design process. Karamba3D offers engineers real-time analysis capabilities for evaluating the structural performance of their designs (Preisinger, 2013).

By seamlessly integrating with Grasshopper, Karamba3D enables engineers to perform structural analysis directly within the parametric modeling environment. This integration eliminates the need for separate analysis software, streamlining the workflow and allowing for rapid design iterations.

With Karamba3D, engineers may simulate various loading conditions, such as gravity, wind, or seismic forces, and analyze how their designs respond to these factors. The plugin provides comprehensive visualization tools to help users understand the structural behavior and identify potential areas of concern. Engineers may assess factors such as deformations, stresses, and displacements to ensure the structural integrity and performance of their designs.

3.2.3 Speckle

Speckle is a powerful data interoperability platform for architecture, engineering, and construction professionals. It enables seamless exchange and collaboration of design data across different software applications and teams (Aec Systems Ltd., 2023).

With Speckle, users may easily transfer and share design information, including 3D models, geometry, and associated metadata, in an open and standardized format. The platform supports popular design software such as Rhino, Grasshopper, Revit, and more, making it compatible with various industry tools.

Speckle simplifies the sharing and synchronization of design data by leveraging a cloud-based infrastructure. Users may upload their design models to the Speckle server, where they become accessible to authorized team members. Any updates made to the model are automatically synchronized across all connected applications, ensuring everyone has access to up-to-date information. Speckle is also a plugin for Grasshopper, making it possible to retrieve and send geometry created in Grasshopper.

3.2.4 TT toolbox

TT Toolbox plugin for Rhino is a software extension that enables users to read and write Excel files directly within the Rhino and Grasshopper modeling environment (Howes, 2023). It enables users to import data from Excel spreadsheets for visualization or analysis. Likewise, it allows users to export data from Rhino and Grasshopper to Excel for further processing or sharing with others.

The purpose of this plugin in this thesis is to transfer data from Grasshopper to an Excel sheet, thereby providing a user-friendly representation of the results.

3.2.5 Wallacei

Wallacei is a computational design and optimization tool designed as a plugin for Rhino and Grasshopper (Makki et al., 2019). It integrates evolutionary algorithms and performance-based optimization techniques to facilitate the exploration and analysis of design alternatives in architectural and engineering domains. Through its parametric simulation capabilities, Wallacei enables designers and architects to generate and evaluate multiple design solutions while considering specific performance criteria. Employing evolutionary computation methods supports the exploration of design spaces and aids in identifying optimal or near-optimal solutions for complex design challenges.

The plugin is intended to give users efficient access to the data outputted by their evolutionary simulations and enable clear and efficient methods for analysis and selection (Makki et al., 2019). For detailed instructions on utilizing Wallacei and its components, please refer to the Wallacei primer available for download at this [link](#).

4 Method

As highlighted in the introduction in Section 1, incorporating BIM and LCA during the early design phase is particularly challenging yet crucial, as it involves making significant decisions that are difficult to modify without incurring time and cost expenses in later stages. In the initial stages, simplified BIM models are typically created to provide project visualizations and facilitate communication. Although these early-stage models contain less integrated information, they possess valuable data such as basic quantities like lengths and element representations.

The objective is to utilize an IFC file containing a building's representation. This file already provides information through the geometric properties of its elements. By developing a code that takes the IFC file as input, it processes the geometry, establishes connections between the elements, and creates a structurally stable system. Furthermore, the aim is to conduct a simplified structural analysis. The code enables modifications on cross sections of the elements, allowing optimization of the building design. The optimization process seeks to identify material and cross section combinations that are both structurally valid and result in the lowest possible LCA footprint.

Figure 4.1 presents a flow chart that simplifies and enhances the comprehension of the code setup and procedures. This flow chart will be used as a guide throughout the section to provide a clear explanation. The code is also divided into the same setup as the flowchart. This means that it consists of a total of three files that encompass the entire code. The first file contains the code for converting IFC to lines, the second file is where you create the structural model, and the last file brings together the optimization process before running the analysis in the same file. Speckle is used to send the necessary data between these files. Each step is explained in its own section.

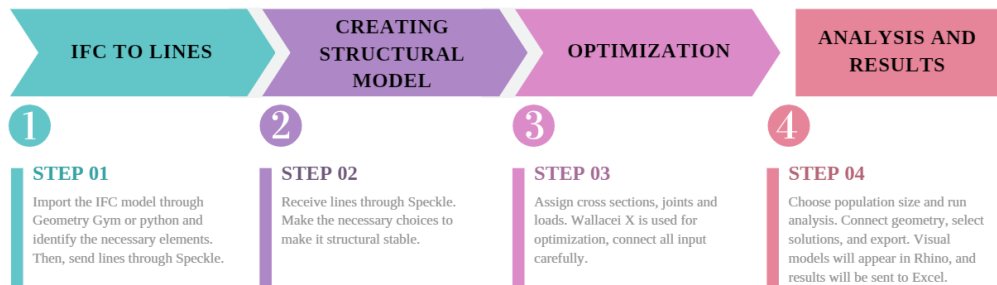


Figure 4.1: Flowchart of methodology.

4.1 IFC to lines

By default, numerous IFC files contain extensive information not essential for creating structural models. For example, architects may include fire insulation and wall claddings. Throughout this study, there are two primary approaches for extracting structural geometry from the IFC file: GeometryGym in Grasshopper and IfcOpenShell in Python. These methods provide alternative ways to process the IFC file and obtain the desired geometry information.

The simplest approach for extracting structural elements involves utilizing the GeometryGym method. However, it may not always be feasible to apply this method. When an element is initially designed as a boundary representation (brep), it lacks a defined edge that may be represented as a line in Grasshopper. Then, GeometryGym cannot be utilized, and the Python method must be tried.

Python

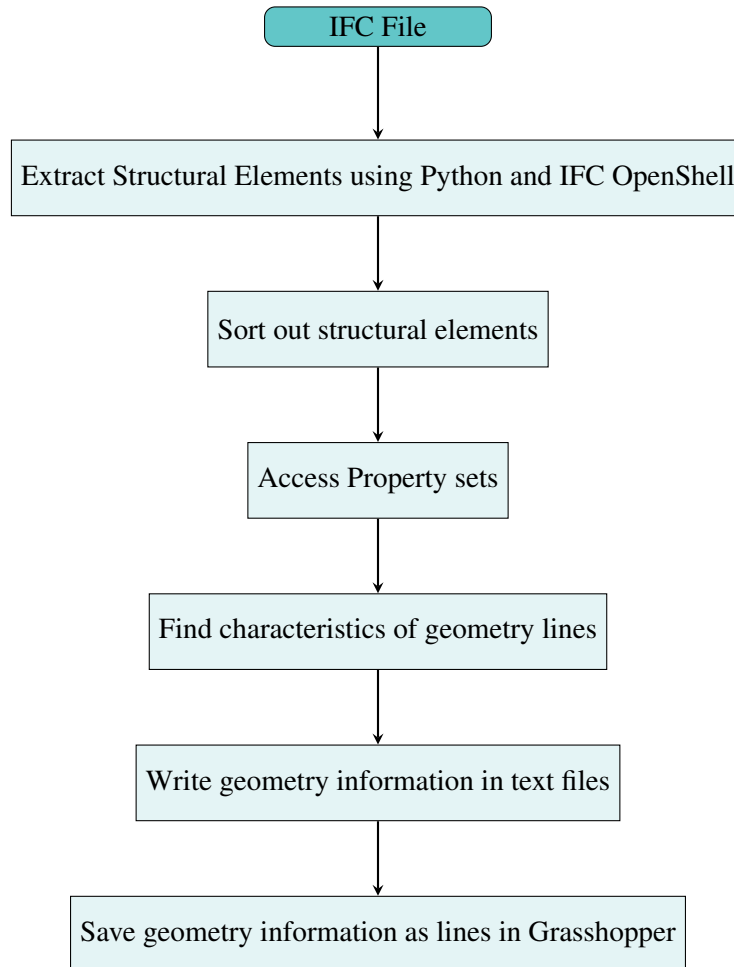


Figure 4.2: Flowchart illustrating IFC file to Grasshopper using Python.

By using Python, the geometry of the IFC elements may be saved as new text files. To extract structural elements from the IFC file, Python is utilized to identify critical elements and sort them along with important information. The Python library *IfcOpenShell* is employed to extract and process the elements that form the structural system in a project. Figure 4.2 shows the method of finding the important geometry lines from the IFC files using Python.

In an IFC file, elements are categorized based on their types, such as *IfcColumn* or *IfcBeam*. By utilizing *IfcOpenShell*, these structural elements, along with their relevant information, may be extracted from the IFC text file. A list is created for each element type, populated with all the instances found in the model. The information includes a designated name. To identify the appropriate and structurally significant elements, keywords like *steel* and *concrete* filter out elements

with these characteristics. This helps exclude elements such as fire isolation layers around beams, which may be labelled *IfcBeam*, though they are not structurally essential.

After filtering out non-structural elements, the geometry undergoes processing. IFC elements typically include property sets (psets), and the specific information stored in an IFC file may vary depending on the model and software used. For instance, models may contain psets with quantities such as lengths. They also include local placement, which provides coordinates for an endpoint and a local coordinate system. To extract the global coordinates, length, and local coordinate system, *IfcOpenShell* in Python is utilized, and the data is saved as text strings that may be translated into Grasshopper's coordinates, lengths, and endpoints. This enables access to the geometry of elements like beams and columns.

The same approach may be applied to walls and slabs in the model by extracting specified properties such as *Length*, *Width*, and *Height*, along with global coordinates and the local axis system. If the required properties cannot be found in the IFC file, the underlying geometry of lines may also be accessed using *IfcOpenShell*'s code: *element.Representation.Representations*. Lengths, coordinates, and other geometric details may be found by searching through the matrices of the representations and accessing *IfcCartesianPoint* start- and endpoints. This method is more complex than predefined quantities. At the same time, it is useful when the required quantities are not readily accessible.

GeometryGym

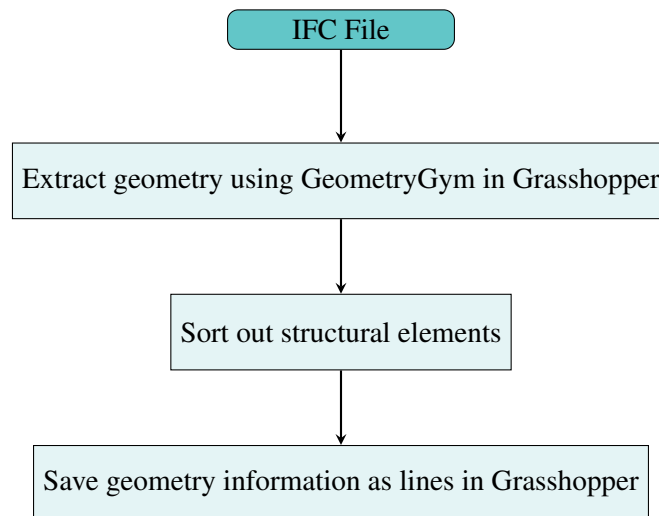


Figure 4.3: Flowchart illustrating IFC file to Grasshopper using GeometryGym.

As seen in Figure 4.3, the geometry of elements may also be obtained through GeometryGym in Grasshopper. This method is similar to manual Python sorting, where element types such as *IfcBeam* and *IfcColumn* are filtered out. Next, non-structural elements, such as the fire insulation around beams, must be sorted out manually. In GeometryGym, this is done by reading the name and index of elements, processing only the structural elements further. Figure 4.4 provides an example of this sorting process, as well as shedding light on the need to exclude invalid elements without proper geometry defined.

The elements may then be decomposed, and by accessing the *axis*, Grasshopper interprets it as a line with a corresponding geometric length and placement as the centerline of the original IFC element. Then, the geometry is scaled and pushed to Speckle as lines, ready to be made into beams, walls, and other elements in Karamba3D.

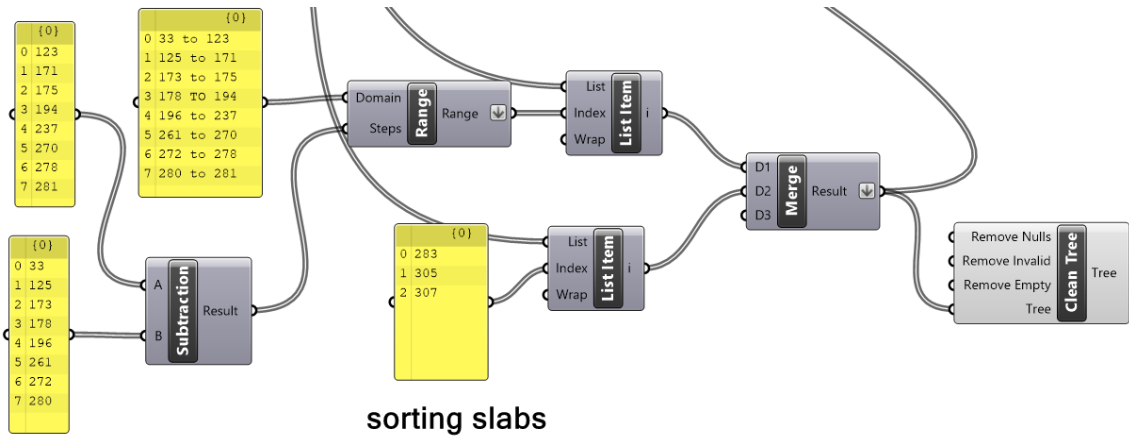


Figure 4.4: Illustration showing manual sorting of elements in Grasshopper.

4.2 Creating the structural model

Ensuring structural stability in the model becomes crucial after extracting the lines in Section 4.1. While IFC files are primarily intended for construction purposes, their direct use in structural analysis may pose challenges. This is because elements in IFC files are constructed based on geometric representations rather than being optimized for structural connections.

This means that the element center lines obtained in Section 4.1 do not intersect by default. They have a distance coming from the cross section dimensions in the original IFC file. For instance, if a 40 cm slab is positioned on top of a column in the IFC file, its center lines would have a distance of 20 cm between them.

In order to address this issue, specific Grasshopper scripts have been developed as part of this thesis. These codes aim to fix the structural model by implementing various techniques. The flowchart in Figure 4.5 illustrates the sequence of steps involved in this process.

Key concepts employed in the method include identifying intersections, snapping elements to common nodes, and creating artificial beams to establish structural connectivity. These techniques bridge the gap between the IFC file and the requirement for intersecting nodes in the center lines of elements in Karamba3D.

It is important to emphasize that the flowchart presented serves as a methodology rather than a comprehensive solution. Due to the inherent complexity and variability of IFC files, different scenarios may require specific adaptations or additional techniques. Nonetheless, the flowchart provides a systematic framework successfully applied in this thesis to address structural stability concerns and resolve issues step-by-step.

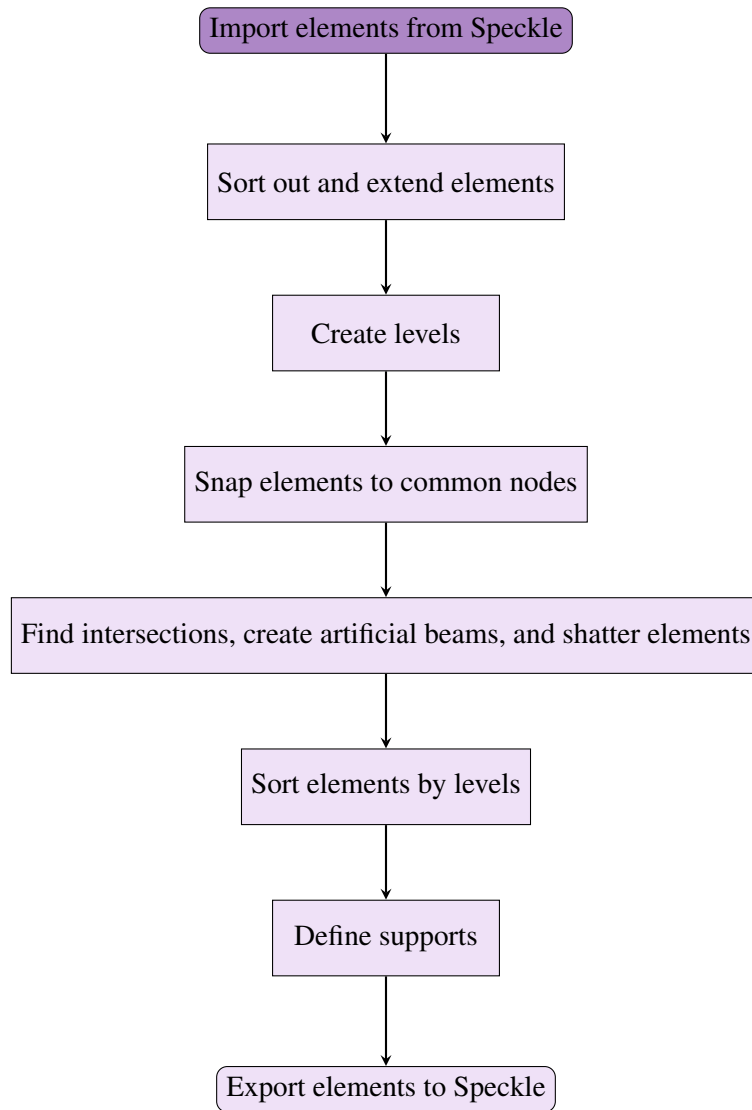


Figure 4.5: Flowchart illustrating the creation of structural models.

4.2.1 Creating levels

In large structures, force distribution varies across levels. For example, lower levels typically experience higher forces compared to upper levels. This variation presents an opportunity to optimize the structural design by tailoring it to the characteristics of each level. This includes optimizing cross sections based on the specific force distribution at each level, resulting in computational efficiency and less time use by reducing the need to search for elements on different levels. By considering the unique characteristics of each level, the structural design may be optimized effectively. However, individually testing and optimizing each element's cross section is impractical due to data capacity and construction challenges.

To establish levels in the structure, element types such as lines representing slabs should be utilized. These are properly sectioned on different floors, and the z-coordinate of the centroid of each geometry element results in a list that represents distinct vertical positions within the structure. The list is then sorted in ascending or descending order to establish a hierarchical arrangement.

If there are closely spaced values, they may be merged at this stage to streamline the structure's representation. Utilizing the sorted and merged z-values, planes are generated at each level. These planes serve as horizontal reference levels, demarcating the position of individual levels within the structure.

4.2.2 Common nodes

After establishing the levels, the elements undergo manipulation to ensure the presence of nodes at these designated levels intersecting with neighboring elements. In conventional building structures, columns and walls transfer forces to the ground. Consequently, the intersections between planes and column and wall edges are designated as common nodes. In cases where multiple nodes are in close proximity within a specified tolerance, they are replaced by a merged common node positioned at the center of these points.

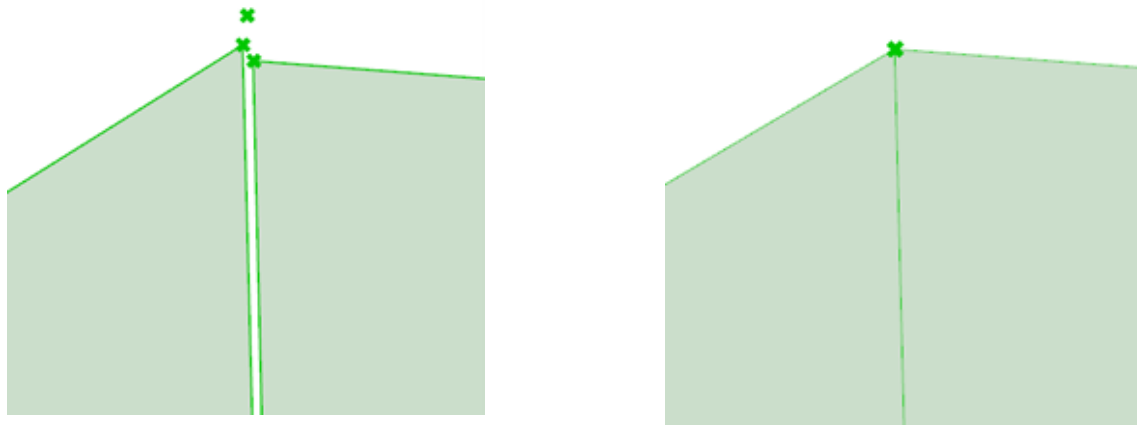


Figure 4.6: Before and after moving geometry to common node.

As depicted in Figure 4.6, the original wall geometry extracted from the IFC file does not exhibit a satisfactory intersection characterized for FEA software. Therefore, a common node is created at the midpoint between the two corner points of the wall, aligned at a specified level height. Subsequently, the corners of the walls are adjusted to align with this newly created common node. Consequently, the walls intersect and may also intersect with nearby beams, hollow core elements, or other connecting elements. By allowing these intersections, FEA software considers the elements as connected. This means that the structural analysis calculations consider the interaction and mutual influence between these intersecting elements.

To facilitate geometry adjustment, the endpoints of each line in the model are examined to align with the specified nodes. If the distance between an endpoint and a node falls below a predetermined tolerance threshold, the node will become the new endpoint, and the line will be updated accordingly. It is crucial to carefully balance the tolerance value to include all closely positioned elements while excluding endpoints that are not in proximity to the starting node. This becomes

particularly critical in larger structures, where precise tolerances are necessary. In cases where finding an optimal balance is challenging, the creation of artificial beams may be necessary.

By employing common nodes, the geometry becomes interconnected. It may be modified, optimized, and adapted to meet specific design requirements and ensure the desired performance and stability of the structure.

4.2.3 Artificial beams

In situations with no wall edges or columns available to define common nodes, or it is not feasible to snap elements to common nodes within the given tolerance, additional connections between elements need to be established. To address this, artificial beams are introduced into the structural model. These artificial beams serve the purpose of creating the necessary connections and ensuring structural integrity within the model.

Artificial beams are small, robust elements that serve as joints between elements that are at a certain distance from each other. Artificial beams act as connectors, ensuring structural connectivity and integrity even though they do not exist physically in the construction.

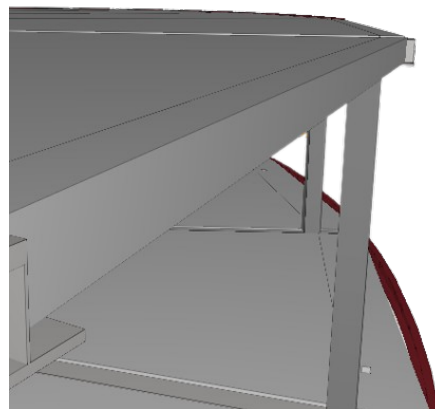


Figure 4.7: Illustration of IFC elements in Solibri BIM model viewer.

Figure 4.7 shows how elements are placed next to each other in a BIM model; however, their centerlines do not intersect. In FEA software, the beam would not intersect with the slab and fall to the ground. It is necessary to connect the beam with the slab using artificial beams. This is illustrated in Figure 4.8, showing artificial beams between a slab and a beam. The beam is also visibly split at the point where the artificial beam intersects, which means that the model will connect properly.

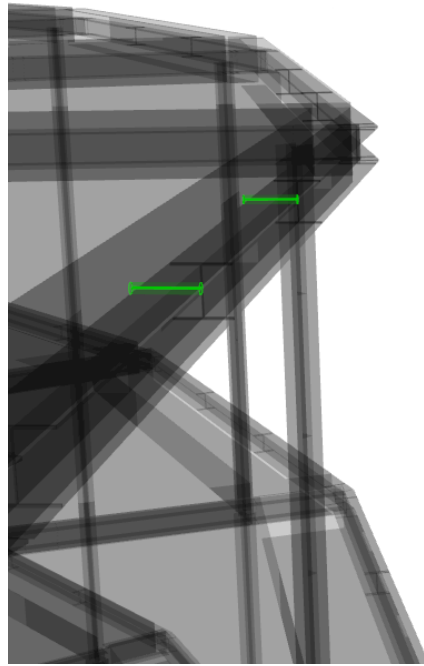


Figure 4.8: Illustration of elements edited and viewed in Grasshopper, with artificial beams, in green, as a connection.

4.2.4 Shatter elements

When an element intersects with another element at a point other than the endpoint, it is necessary to divide the element at that intersection to establish a connection. This process involves shattering the element at the intersection point, creating a node that may be used to connect with other elements. The nodes also split the element into several elements, with rigid connections at the nodes.

To achieve this, intersecting points are identified within a specified distance. Once these points are identified, the element is divided at those locations, resulting in the creation of nodes. These nodes serve as connection points for the element with other intersecting elements, facilitating the overall structural connectivity.

4.2.5 Sorting elements to levels

Organizing different element types into their respective levels involves a sorting procedure based on proximity. A tolerance equal to half the average height of a level is employed to determine the threshold for proximity. By measuring the distance from each element to all the reference planes representing the levels, the elements are assigned to the closest plane.

This sorting mechanism results in a structured arrangement where elements of the same level are grouped in a branch within a data tree. By repeating this procedure for all elements, the entire geometry is effectively sorted into their corresponding levels.

This approach ensures that each element is associated with its appropriate level based on its proximity to the reference planes. Utilizing a data tree allows for organized storage and easy retrieval of geometry specific to each floor or level within the structure.

4.2.6 Defining support points

Once the levels in a model are established, identifying the supporting points becomes straightforward. If the geometry intersects with the lowest level, it is considered a support point. In the context of buildings, this is typically the case for columns and walls. These elements provide essential structural support and bear the weight of the overall structure. By recognizing the intersections of these elements with the lowest level, their role as supports may be accurately described and accounted for in the structural analysis.

4.3 Optimization

The optimization process is primarily concentrated on beam elements, which include both beams and columns. Meshes are not a part of the analysis and are assigned a default material. This is due to simplifying and minimizing the optimization run time. The workflow for the optimization process is illustrated in Figure 4.9.

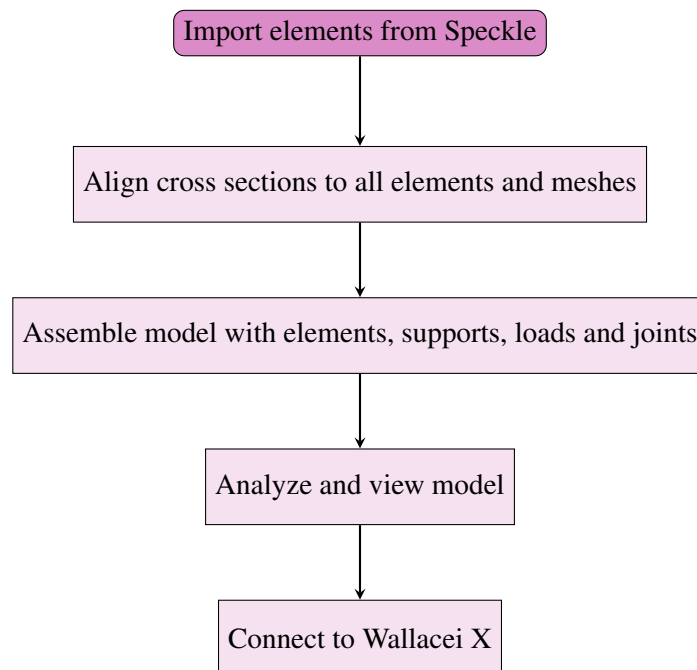


Figure 4.9: Flowchart illustrating the optimization.

The geometry required for the optimization process is imported into the Grasshopper file using Speckle. The required geometry includes lines representing columns, beams, walls, slabs and artificial beams, as well as points for supports and inclusion points for meshes.

The levels of the building are divided into one or multiple groups, allowing for the alignment of

different cross sections to specific groups for the same structural element type. This approach enables the optimization of larger cross sections at the bottom of the building and smaller cross sections on the upper levels, which reflects a more realistic design. Each group is associated with a dedicated slider, which interacts with both the cross section table and the table with the corresponding GWP value. The approach to how these tables are made is explained in Section 4.3.1 and 4.3.2. The slider aligns cross sections to the structural elements. At this stage, curve lengths and their corresponding GWP values are exported, as they are necessary for calculating the building's LCA footprint.

The next step involves assembling the model by connecting all the elements, supports, loads, and joints using Karamba3D. In the *Assemble* component, all input data structures are flattened to ensure one single model as the final result.

When incorporating joints, the components utilized are either the *line-joint* or *beam-joint* components from Karamba3D. By employing these components, the interconnected elements are not rigidly fixed, thereby influencing the distribution of forces among them. This method effectively reduces the occurrence of moment forces within the connections.

To determine the loads, the self-weight of all cross sections and line load on slabs are considered. The line load refers to a uniformly distributed load. Table 4.1 display the loads for common situations, sourced from Karamba3D (Karamba3D 2.2.0, 2021). This table may be utilized when specific load information is not provided.

Table 4.1: Loads for typical scenarios (Karamba3D 2.2.0, 2021).

Type	kN/m ²
live load in dwellings	3.0
live load in offices	4.0
snow on horizontal plane	1.0

Once the model is obtained, it may be analyzed. It is possible to identify the highest values of the elements' utilization and examine the beam and shell views. The Wallacei X component utilizes the largest utilization as an objective, and the render beam and shell meshes as a visual representation.

Wallacei X serves as the optimization component in this study. Figure 4.10 illustrates the component itself and its corresponding connections. The genes, represented by red lines in the figure, act as sliders that iterate through both the cross section table and the table containing GWP values. These adjustable values play a crucial role in finding the optimal solution. The two fitness objectives are here denoted in blue. The first objective involves identifying the element with the highest utilization, while the second objective focuses on calculating the total LCA footprint of all elements. A domain is established to guide the algorithm towards a value of utilization within the range of 0.51 and 0.99. This range ensures that the results are safe, preventing element failure. It also makes it efficient, avoiding wastage of materials due to under-utilization. The domain restricts all input within the given range, resulting in a minimum value of 0.48. Any values below or above this range will yield larger values, making 0.48 the lowest achievable value for this fitness objective in the simulation.

The optimization process aims to achieve the utilization mentioned and minimize the LCA footprint. The highest utilization achieved in each iteration is captured in the input *data*. This makes it simple to extract the precise utilization value when selecting solutions. The last input parameter is the phenotype, which represents the visual representation of the exported solutions. The render beams and shell meshes are used as the phenotype, allowing visual differentiation between various selected solutions.

It should be noted that the number of iterations for each slider should match the number of cross sections in the cross section table attached to the elements with the corresponding slider. Additionally, it is crucial to exercise caution when connecting to the Wallacei X component. This carefulness is necessary to ensure accurate solutions and prevent the simulation from crashing. Careless connection and disconnection of sliders may result in inconsistencies in the analysis when retrieving slider values for the analysis results.

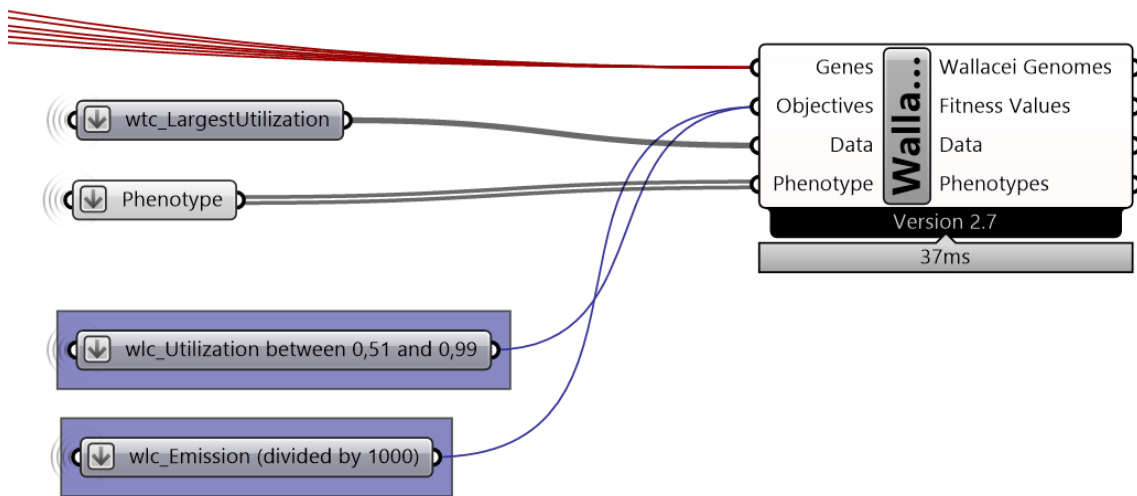


Figure 4.10: Wallacei X component with connections from Grasshopper.

4.3.1 Generation of cross section tables

To optimize the cross sections, tables containing various cross sections are required. These cross sections may be iterated through by using the sliders. To accomplish this, two distinct cross section tables were created: one for beam members and another for meshes. These lists were compiled by incorporating selected default materials and cross sections from Karamba3D.

When referring to default cross sections, it implies that the design standards employed for each cross section may vary. The created tables include cross sections with EU-, UK- and Chinese standards.

A diverse range of cross sections and materials are utilized when creating the cross section table for beam elements. Three materials, namely steel, concrete, and glulam timber, were selected for inclusion. Within each material category, various dimensions for cross sections were selected. For steel, dimensions such as IPE and HEA were included, while for concrete, dimensions like C30, C35, and C45 were chosen. In the case of glulam timber, the GL28c quality was selected. The

cross sections from each family and material were combined before being merged into a single consolidated table. The tables are a part of the Grasshopper file regarding optimization and are shown in Appendix A.

For cross sections used in meshes, concrete is employed as the material with two different strengths: C30 and C45. The heights of these shell elements range from 20 cm to 50 cm. The specific shell element utilized is, *ShellRC Std Const*, which enables the specification of a reinforced concrete cross section that remains constant throughout the shell ('3.3.2: Shell Cross Sections - Karamba3D 2.2.0', n.d.).

The cross section table for mesh primarily serves as a reference list for selecting default materials and allows for easy switching between cross sections if necessary. However, it is important to note that this table is not utilized during the optimization process when calculating the LCA footprint, as explained earlier.

4.3.2 Calculation of LCA footprint

To determine the LCA footprint and quantify the environmental impact of the buildings, a specific quantity of materials is chosen during the optimization process. For each selected material, a product-specific EPD is utilized to gather the necessary information required for the calculation. The EPDs used are taken from the website of EPD-Norway and are linked in Table 4.2 with all the selected materials.

The EPDs offer the essential data, specifically the GWP value expressed in kilograms of CO₂-equivalent per unit. The analysis encompasses scope A1-A4, which includes the entire life cycle of the materials, starting from raw material extraction through transportation to the customer on-site.

To facilitate the calculation of the LCA footprint, the GWP values are manually listed alongside the cross section table in the grasshopper file. This allows for calculating the corresponding footprint for each selected cross section used on the elements. By linking the GWP values directly to the cross section table, the LCA footprint associated with each cross section choice are determined by multiplying the value with the corresponding length of each element. In the end, all values are added together to get the total footprint. This value of the total LCA footprint of the building is the second objective in the Wallacei X component.

Each product-specific EPD includes the calculation of the GWP value within scope A1-A4. As part of this analysis, transportation from the manufacturing facility to the construction site is considered, with the distance specified for each material provided in Table 4.2. It is important to note that different distances of transportation are employed for different materials. It is included in the analysis to raise awareness about the transportation aspect.

Table 4.2: The materials selected for optimization and their EPDs.

Material	GWP [kg CO ₂ - eq]	Unit	Distance [km]	Link
C30	217.935	m ³	8	Link
C35	203.57	m ³	28	Link
C45	249.95	m ³	20	Link
Steel beams I, H, U, L and T, S355	1.0878	kg	300	Link
Cold-formed hollow profiles, S355	2.5548	kg	300	Link
Standard glulam, GL30c	-597	m ³	200	Link

4.4 Analysis and results

When running the analysis, it is necessary to control that all needed inputs are correct and connected to the Wallacei X component as described above in Section 4.3. If it is not done, there is a high chance that the simulation will crash. When opening the Wallacei X component, the first page you come to is Wallacei Settings, where it is necessary to choose population size. The size of the population depends on the problem you are optimizing. Under simulation parameters, it is also possible to verify that all your sliders and fitness objectives are properly connected. The number of fitness objectives should be two: the utilization value and the calculated LCA footprint. The number of sliders depends on how you divide the elements into groups to align with cross sections. After choosing the population size and controlling the inputs, you start the analysis. For our project, the default values for *Algorithm parameters* are chosen.

As an example throughout this section, a simple frame was created. The simple frame is designed as a single-story structure comprising only six columns and seven beams. The optimization process only utilizes two sliders. One slider is assigned for controlling the columns, while the other slider is dedicated to the beams. The fitness objectives are as mentioned in the section regarding optimization 4.3, the value of the largest utilized element, and the LCA footprint corresponding to selected cross sections.

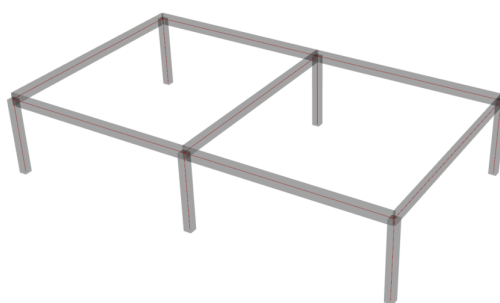


Figure 4.11: Illustration of the simple frame.

The optimization process for the simple frame involved running an analysis with a population size of 2000. The generation size was set to 40, indicating the number of individuals in each generation. In contrast, the generation count was set to 50, representing the total number of generations

generated during the optimization process. An illustration of Wallacei Settings after running the simulation is shown in Figure 4.12.

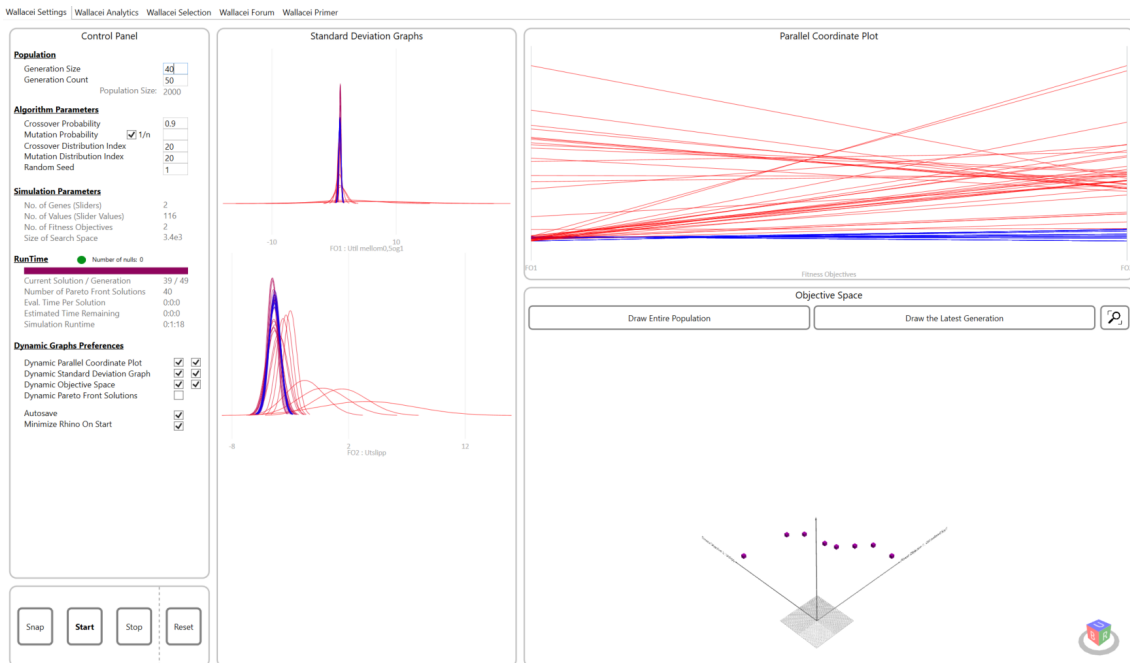


Figure 4.12: Illustration of the Wallacei settings with results from the simple frame example.

Upon completion of the simulation, it becomes possible to examine and choose solutions. This is achieved by utilizing the Wallacei Analytic tool, located in the tab immediately following Wallacei Settings. To initiate the analysis, the *Draw* button is pressed, resulting in the display of various charts. A visual representation of this tab may be observed in Figure 4.13. To obtain the smallest value for each objective, the *Selected fitness objective* and *Selected rank* options are utilized. The smallest values may easily be detected by selecting fitness objective 1 with rank 0 or fitness objective 2 with rank 0. The rank is set to 0 to indicate the smallest value. To choose and add the desired solution to the export list, the *Select* and *Add* buttons are pressed, respectively.



Figure 4.13: Illustration of the Wallacei Analytics with results from the simple frame example.

The next tab, Wallacei Selection, is also used to retrieve solutions. The first step here is to press *Draw parallel coordinate plot*. Figure 4.14 shows this tab. Using the analysis method *Solutions with repeated fitness values* it is possible to export all valid solutions. Valid solutions are here defined as those solutions with a utilization value between 0.51 and 0.99. If the simulation has found valid solutions then these may be obtained by choosing fitness objective 1 and rank 0, these will then have a value equal to 0.48, as explained in Section 4.3 optimization. All valid solutions will then appear. Pressing *Add* on the solutions will add them all to the export list.

Further refinement is performed through manual sorting within the Grasshopper environment to identify unique solutions from the aforementioned set, before exporting the top eight optimal solutions to Excel.

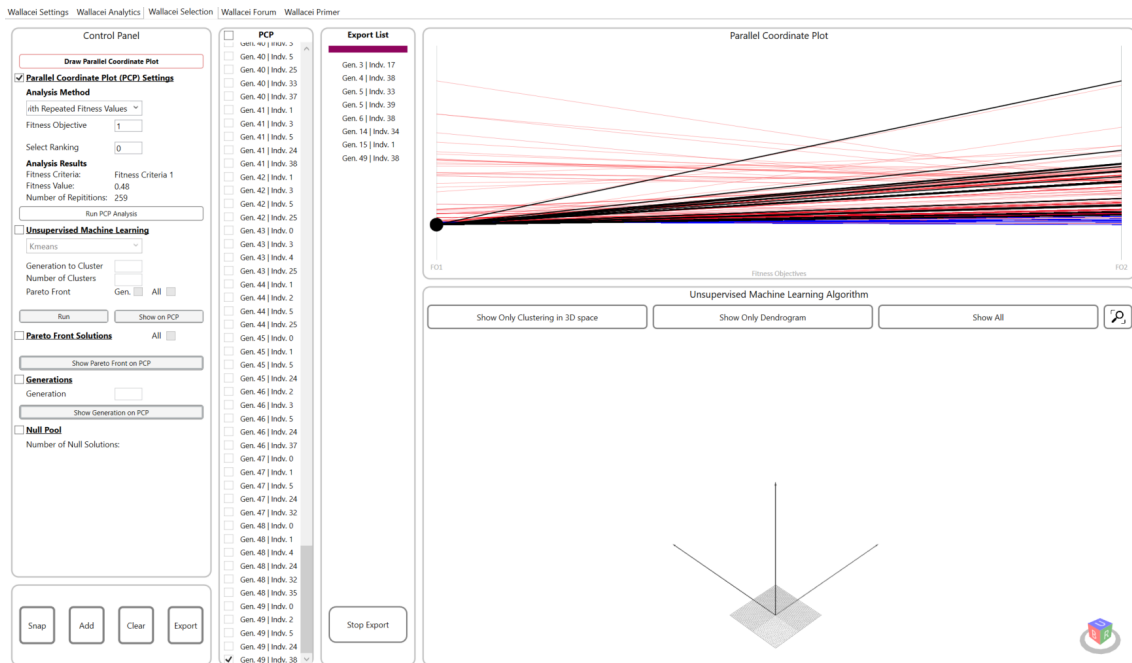


Figure 4.14: Illustration of the Wallacei Selection with results from the simple frame example.

Once all the relevant solutions have been thoroughly examined and added to the export list, it is crucial to ensure that the geometry meshes are correctly linked to the phenotype input of the Wallacei X component. Clicking the *export* button will then generate a visual representation of the geometry for the selected solutions. Additionally, a small code has been developed to export the solutions to an Excel file, specifically for a subset of eight solutions.

The visual representation of the geometry for the simple frame building may be observed in Figure 4.15, the optimal solution is marked with a black arrow.

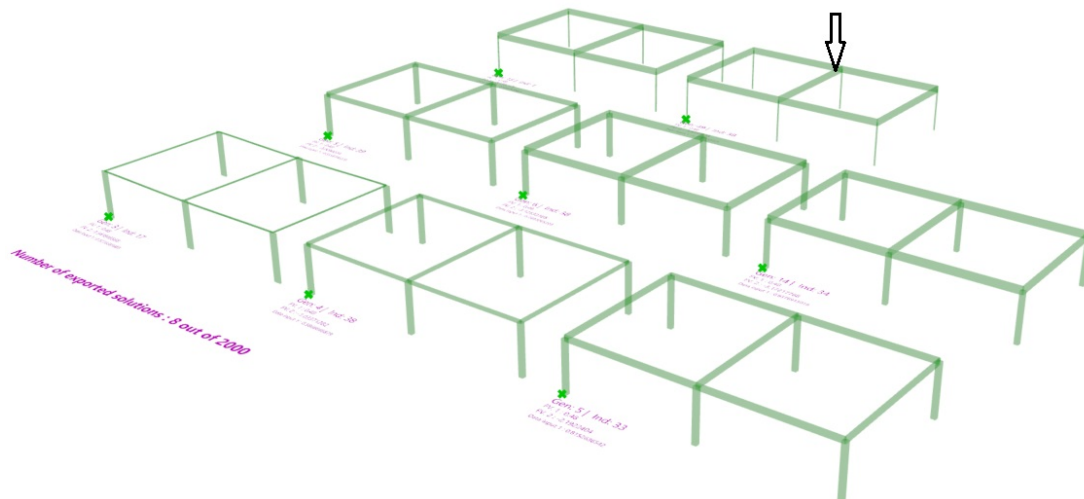


Figure 4.15: Visual representation of selected geometries of the Simple Frame. The black arrow indicates the optimal solutions with lowest LCA footprint.

Table 4.3 presents the exported solutions from the Excel file, including the LCA footprint, largest utilization, and selected cross sections. The cross sections are identified by the element name, either column or beam, along with an assigned number corresponding to their respective groups. In the simple frame, all columns and beams are each assigned to one group. However, for larger buildings with elements divided into multiple groups, the group containing the lowest floors is designated Column 1 or Beam 1, while subsequent groups are assigned higher numbers.

From the table, it is evident that solution {49;38}, corresponding to generation 49 and index 38, exhibits the smallest LCA footprint and a utilization value of 0.78.

When examining the analysis results, it is valuable to compare them to the original cross section choices in the model. In the original cases, all models had undergone analysis and were assigned finite cross sections. These cross sections may be found by viewing the IFC files in software like Solibri. Since the models have already been analyzed by external stakeholders, which are outside the scope of this thesis, no new analysis is conducted or included. Consequently, there is no maximum utilization number available for the original analysis in the context of this thesis. Consequently, while assessing the utilization may not be highly relevant, comparing the LCA footprints is meaningful. The calculation of the LCA footprint follows a similar procedure described in Section 4.3, although with a manual approach of aligning the original cross sections to the elements.

Table 4.3: Results from the analysis of the simple frame.

Solution	LCA footprint [kg CO₂-eq]	Largest util.	Column 1	Beam 1
{49;38}	-3636	0,78	IPE 120	Glulam 14x63
{15;1}	-3570	0,52	IPE 140	Glulam 14x63
{14;34}	-3172	0,84	C45 35x35	Glulam 14x63
{6;38}	-3125	0,75	C35 40x40	Glulam 14x63
{5;33}	-2192	0,82	C35 35x35	Glulam 14x45
{5;39}	-2009	0,56	C35 40x40	Glulam 14x45
{4;38}	-1034	0,59	C30 35x35	Glulam 14x27
{3;17}	1542	0,53	Glulam 14x63	SHS 100X5

5 Cases

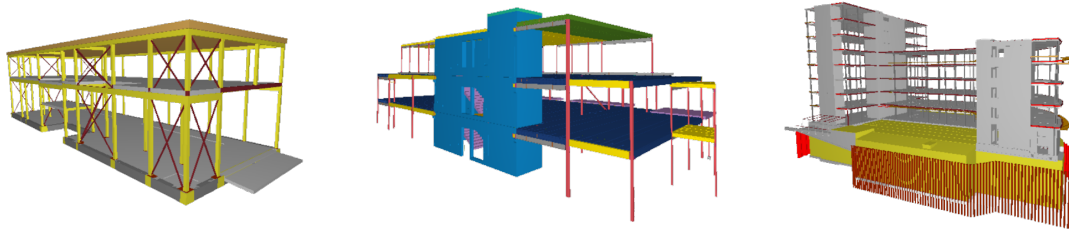


Figure 5.1: Case 1, 2, and 3. From the left Tennebekk, Strandhagen, and Buebygget.

This case study consists of three buildings of varying sizes. Certain modifications were required to get results. The modifications are explained in the same structure as the overall method in Section 4. Illustrations of the cases are presented in Figure 5.1. Table 5.1 presents an overview of the differences between the cases. The listed areas are approximate.

Table 5.1: An overview of the case study.

Case study	Number of floors	Area [m ²]	Total: columns and beams
Tennebekk	2	740	153
Strandhagen	3	1110	266
Buebygget	8 + cellar	14 000	3310

5.1 Case 1 - Tennebekk

Tennebekk is a two-floor building, located in Bergen, Norway. The structure is based on beams, columns, and slabs, and it is the smallest case study for this thesis. The model was created in Revit before being exported to an IFC file.

5.1.1 IFC to lines

In Case 1, the process of extracting structural elements from the IFC file involves the utilization of GeometryGym in Grasshopper, as explained in Section 4.1.

During the extraction process, beams and columns are exported without requiring any element sorting. However, the slabs undergo processing, excluding elements such as isolation and footings that are unnecessary for the creation of the structural system. In this particular case, only 4 out of the total 72 slab elements are exported. The walls are considered non-essential and are therefore not included in the export.

Following the sorting, the elements are scaled to meters and seamlessly transmitted from the initial Grasshopper file, labeled as *Tennebekk 1) Geometry* in Appendix A, to the subsequent stage using Speckle. This streamlined transfer mechanism ensures the smooth importation of the sorted elements into the next Grasshopper file.

5.1.2 Creating the structural model

To generate a structurally stable model from the IFC file, the method outlined in Section 4.2 and depicted in Figure 4.5 is employed. The Tennebekk model's unique characteristics are described below, and the geometric transformation is illustrated in Figure 5.2.

When creating the structural model, the first step is importing columns, beams, and slabs from Speckle. Following this, small elements are sorted out.

Next, levels are created considering the slabs and beams, resulting in the establishment of four levels. Elements are then snapped to common nodes, which are generated at the intersections between columns and planes. In the case of Tennebekk, all elements are close to these common nodes. As a result, the beams and slabs snap to the common nodes without additional artificial beams. This simplifies the structural configuration and eliminates the requirement for extra elements to facilitate connectivity.

The process continues by identifying intersections and shattering the elements. This includes finding inclusion points for the mesh of the slab, as well as addressing intersections between beams, columns, and slabs. Beams and columns are subsequently shattered to ensure proper connectivity.

The elements are then sorted based on their respective levels. Supports are defined by selecting only the column endpoints from the lowest floor. The structurally stable model is illustrated in Figure 5.2, representing the geometry before and after snapping to common nodes.

Finally, the resulting model is exported back to Speckle. This sets the stage for further analysis and optimization of the design.

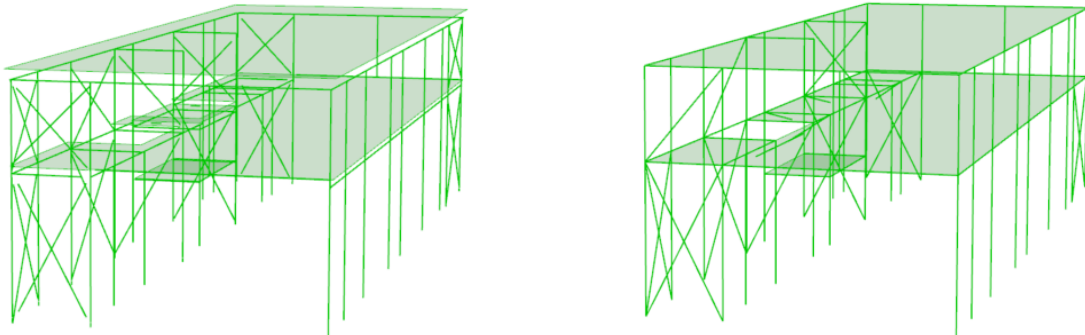


Figure 5.2: Before and after operating on the geometry of Tennebekk.

5.1.3 Optimization

The necessary elements are imported from Speckle and connected to their respective paths, indicated in pink, in the final Grasshopper file labeled *Tennebekk 3) Optimization* in Appendix A. In addition, the beams are separated into two categories: bracing elements and regular beams. The bracing elements are unique to this case and have their own code to distinguish them from the

straight beams. This separation allows the bracing elements to be free from bending and facilitates the connection of a different cross section table.

A separate cross section table is created specifically for bracing elements, which includes only steel cross sections. This decision addresses challenges that may arise when concrete elements are subjected to significant tensile forces. As a result, the iteration of the slider, which controls the selection and adjustment of cross sections, is limited to steel members within this category. In other words, the optimization process for bracing elements focuses solely on steel options while excluding other materials like concrete and timber from consideration.

The slab elements are divided into two groups, each assigned a cross section. The analysis utilizes a C30 cross section with a height of 50cm for the roof slab, while the remaining slabs are assigned a C35 cross section with a height of 45cm. It should be noted that slabs are not included in the analysis, and therefore, their cross section is not factored into the calculation of the LCA footprint.

The applied loads for the building are obtained from an existing report from COWI, with a load of -2.6 kN/m² applied to the roof and -9.4 kN/m² for other slabs.

In this case, the *Line-Joint* component is utilized for all slabs.

For the optimization process, six sliders are therefore used. The sliders are divided into two sliders for columns, two for beams, and two for bracing elements. The objectives remain the same as explained in Section 4.3.

The connection to the Wallacei X component is established following the steps outlined in Section 4.3.

5.1.4 Analysis and results

For the analysis, a generation size of 40 and a generation count of 50 resulted in a population size of 2000. An initial analysis was conducted with a population size of 1000. However, no solutions were found with a utilization value below 1. Consequently, the population size was increased to expand the search space.

Table 5.2: Key data utilized in the analysis and the optimization of Tennebekk.

	Number of columns	Number of beams	Generation size	Generation count	Population size	Sliders	Objectives
Case 1	60	93	40	50	2000	6	2

During the solution selection process, the primary objective was to identify valid solutions satisfying our two fitness objectives. Valid solutions were defined as those with a utilization value between 0.51 and 0.99. Using the method described in Section 4.4 resulted in 66 valid solutions.

Further refinement was performed through manual sorting within the Grasshopper environment to identify unique solutions from the aforementioned set. As a result, eight distinct solutions were identified and sent to Excel. The geometries of the solutions are shown in Figure 5.3, where a black arrow indicates the optimal solution. Table 5.3 shows the results from lowest to highest

LCA footprint. Based on the table, the optimal solution has an LCA footprint equal to 20 116 kg CO₂-eq. and maximum utilization of 0,98. It also shows the LCA footprint from the original IFC file and its cross sections.

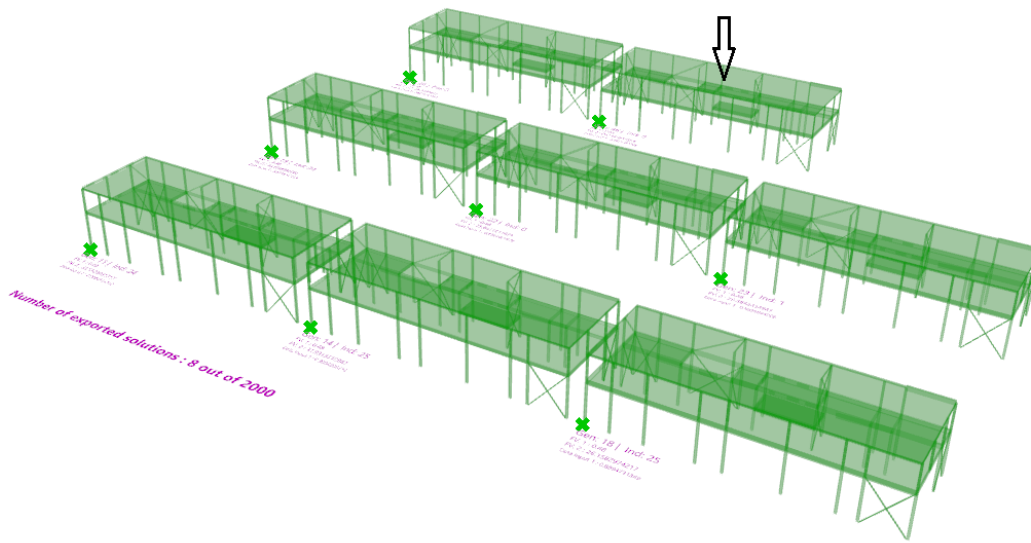


Figure 5.3: Visual representation of selected geometries of Tennebekk. The black arrow indicates the optimal solutions with lowest LCA footprint.

Table 5.3: Results from the analysis of Tennebekk. The original LCA footprint and cross sections are provided at the bottom for reference. The model has already been analyzed by external stakeholders, and no new analysis on utilization has been conducted or included.

Solution	LCA footprint [kg CO ₂ -eq]	foot-util. [kg]	Largest util.	Column 1	Column 2	Beam 1	Beam 2	Bracing 1	Bracing 2
{49;2}	20 116	0,98	HEA 300	SHS 180X8	Glulam 14x63	IPE 100	SHS 120X5	IPE 240	
{27;1}	21 195	0,89	HEA 300	SHS 200x8	Glulam 14x63	IPE 100	SHS 120X5	IPE 240	
{23;1}	25 584	0,89	HEA 300	SHS 250X8	Glulam 14x63	HEA 140	SHS 120X5	IPE 240	
{22;0}	25 841	0,89	HEA 300	SHS 250X8	Glulam 14x63	HEA 140	SHS 120X5	IPE 300	
{18;25}	26 158	0,89	HEA 300	SHS 250X8	Glulam 14x63	HEA 160	SHS 120X5	HEA 160	
{19;34}	26 165	0,89	HEA 300	SHS 250X8	Glulam 14x63	HEA 160	SHS 120X5	IPE 240	
{14;25}	31 931	0,90	HEA 300	SHS 250X8	Glulam 14x45	HEA 140	SHS 120X10	HEA 160	
{11;24}	32 512	0,90	HEA 300	SHS 250X8	Glulam 14x45	HEA 160	SHS 120X10	HEA 160	
Original	50 559	-	SHS 180x10	SHS 180x10	IPE 400	IPE 270	SHS 80x5	SHS 80x5	

In order to compare the LCA footprint of the optimized solution with the original model obtained from the IFC file, an approximate calculation was performed to assess the LCA footprint of the original building. The cross sectional data of beams and columns were extracted from the IFC file. All cross sections used for the analysis are shown at the bottom of Table 5.3. The calculation led to an estimated LCA footprint of 50,559 kg CO₂-eq.

The optimized solution achieved a notable improvement, demonstrating a 60% reduction in the LCA footprint.

Figure 5.4 presents a color mapping scheme for elements in the optimal solution {49;2}. In this representation, elements with the lowest contribution to the overall LCA footprint are visually distinguished by a green color. As the elements' contributions increase, they are progressively depicted in shades of yellow and red, indicating a higher level of impact. From the color mapping, it is evident that the columns contribute significantly to the overall LCA footprint in this model.

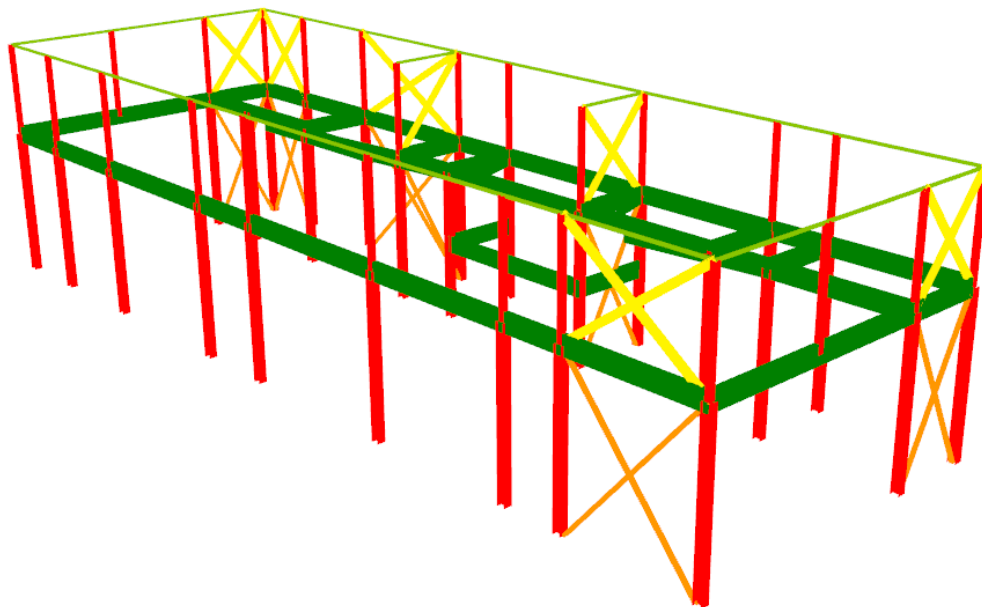


Figure 5.4: Color mapping of all elements showcasing their LCA footprint in Tennebekk.

5.2 Case 2 - Strandhagen

The second case, Strandhagen, is a prefabricated framework in steel and hollow core elements, originally built in the software Tekla. The project consists of two buildings with three floors each. It is a simple structure with beams, columns, walls, and hollow core elements.

5.2.1 IFC to lines

In this case, the GeometryGym method did not extract elements as needed, hence the representation of hollow core elements in the IFC files as breps. Therefore it is utilized the method of Python, IfcOpenShell, and Tekla Quantities as explained in Section 4.1.

The detailed code, presented in Appendix B, illustrates the systematic process of searching and extracting the required elements. The identified elements are categorized, and their relevant properties, such as lengths, coordinates, and local coordinate systems, are stored in separate files. These generated files are then integrated into the Grasshopper file labelled *Strandhagen 1) Geometry* in Appendix A.

The project encompasses two buildings; however, for practical purposes, it is convenient to analyze only one building. Due to the Python sorting process not allowing for identifying elements belonging to specific buildings, a bounding box is created in Rhino around the desired building. Only elements in this box are exported through Speckle and transmitted to the next stage.

5.2.2 Creating the structural model

To create the structural model in Strandhagen, several of the concepts described in Section 4.2 are employed. The steps from geometry from the IFC file to a structurally stable model are shown in Figure 4.5.

This process begins by importing elements from Speckle, which serves as the input. The elements are then sorted and extended as needed. Some non-structural elements, or elements that may not be connected properly, are included in the Speckle files. These elements are shorter than others and are therefore sorted out because of their length. The same is done for the hollow core slabs and the walls, sorted out from the size of the area. In addition, the hollow cores are extended in length to get an intersection with beams and walls later.

Next, planes are generated to establish the geometric reference for the structure. These planes are created by utilizing the endpoints of the hollow cores, with the starting point of the lowest wall serving as the reference for the bottom level. This approach results in Case 2 having a total of four levels.

Following that, the elements undergo a snapping process to align them to common nodes, ensuring connectivity and proper alignment. In the case of walls, an additional tall segment that complicates the geometry is removed by the highest plane. The nodes are generated according to the procedure outlined in Section 4.2, and the elements are then snapped to these nodes. As a result, a model is formed with intersecting elements, enabling further analysis and manipulation. A visualization of the geometric change is shown in Figure 5.5.

The next step involves identifying the intersections between elements and shattering them at the corresponding nodes. This allows for a proper connection between the elements. Subsequently, the elements are sorted into their respective levels, enabling the optimization of each cross section based on the forces within that level.

Once the elements have been sorted by levels, the supports are defined. This is achieved by retrieving the columns and walls on the lowest level and collecting their bottom points and lines. Finally, the processed elements are exported back to Speckle, completing the workflow.

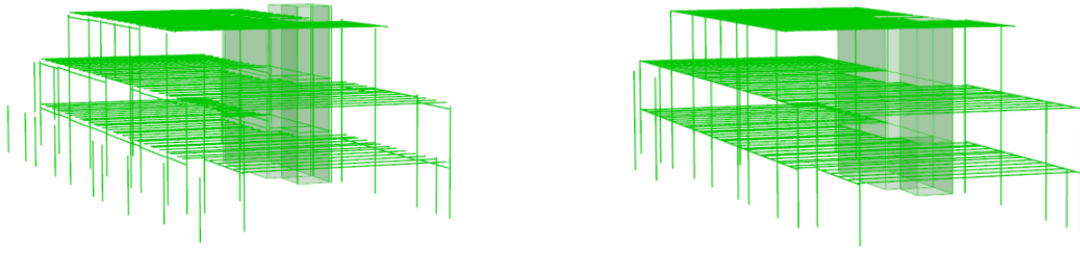


Figure 5.5: Before and after operating on the geometry of Strandhagen.

5.2.3 Optimization

The essential elements are imported from Speckle and connected to their corresponding paths, which are highlighted in pink in the final Grasshopper file titled "Strandhagen 3) Optimization" in Appendix [Grasshopper]. Columns and beams are segregated into three groups, each comprising elements from a single level. Notably, the first level contains no beams, while the last level lacks columns. As a result, three groups are formed, each consisting of one level. The dividing into groups results in six sliders, and the objectives remain the same as explained in Section 4.3.

The columns follow the global coordinate system, and it is necessary to align the local axis obtained from Speckle with the z-orientation of the columns by rotating the columns to their local XY-plane. Figure 5.6 compares the columns before and after rotating local axes in the XY-plane. Figure 5.7 illustrates the moment distribution along the column before and after the axis correction. It is evident that before the axis correction, the cross section experiences more pronounced moments in the y- and z-directions. However, after the correction, the moments are primarily observed in the y-direction, and the forces are flipped accordingly.

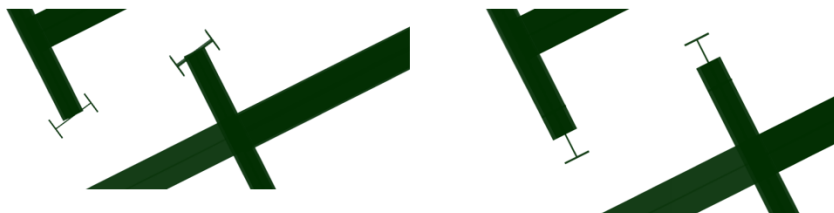


Figure 5.6: Illustration showing columns before and after rotating local axes in the XY-plane in Strandhagen.

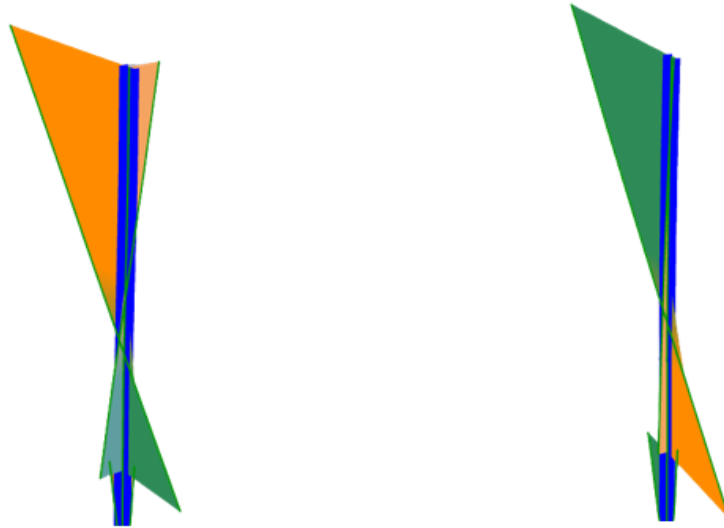


Figure 5.7: Moment distribution before and after rotating local axes in the XY-plane in Strandhagen.

Similar to the approach taken for the bracing elements in Case 1, the cross section table is modified to include only steel elements specifically for the beams. This means the slider will only assign cross sections consisting of steel to beams. This is because the analysis crashes when including both concrete and timber materials for beams in Case 2.

The wall elements are assembled in one group and assigned their respective cross section. The analysis utilizes a concrete quality of C45 with a width of 35cm for all walls. The hollow core elements are also grouped into one and assigned the steel cross section HEA240. Karamba3D does not provide specific functionality for modeling hollow core elements. It is therefore utilized steel, as employing ordinary concrete across the span would not provide a realistic representation.

It should be noted that walls and hollow core elements are not included in the analysis, and therefore, their cross section is not factored into the calculation of the LCA footprint.

The building is subjected to line loads applied on all hollow core elements with a factor of -5 kN/m. The factor is taken from Table 4.1, *live load in office*, multiplied with the approximate width on the hollow core elements extracted using Python as seen in Appendix B, on 120 cm. However, in an ideal analysis, it should have been hollow core elements with an approximate width of 120 cm each.

The *Beam-Joint* component is utilized. The joints are connected to all hollow core elements and all connections between beams and walls.

The connection to the Wallacei X component is established following the guidelines outlined in Section 4.3.

5.2.4 Analysis and results

For the analysis, a generation size of 40 and a generation count of 50 are utilized, resulting in a population size of 2000, the same as in Case 1. In contrast, Case 2 is larger and involves a greater number of beams, which presented challenges during the analysis. However, aligning only steel members on beams instead of all materials, and reducing the CPU usage for Rhino during the analysis, resolved these difficulties during the research.

Table 5.4: Key data utilized in the analysis and the optimization in Strandhagen.

	Number of columns	Number of beams	Generation size	Generation count	Population size	Sliders	Objectives
Case 2	53	213	40	50	2000	6	2

During the selection process, the primary objective was to identify solutions satisfying our two fitness objectives, utilization and LCA footprint. The selection is made as described in Section 4.4.

A total of 77 solutions were obtained when extracting all valid solutions. Further refinement was carried out to identify 11 unique solutions. From these, only the 8 solutions with the smallest LCA footprint were selected and exported to Excel for further analysis. The geometry of the solutions is shown in Figure 5.8, where a black arrow indicates the optimal solution. Table 5.5 shows the results from Excel sorted from lowest to highest LCA footprint, where the best solution is observed with an LCA footprint equal to 7013 kg CO₂-eq. and maximum utilization of 0,98. It also shows the LCA footprint from the original cross sections from the IFC file.

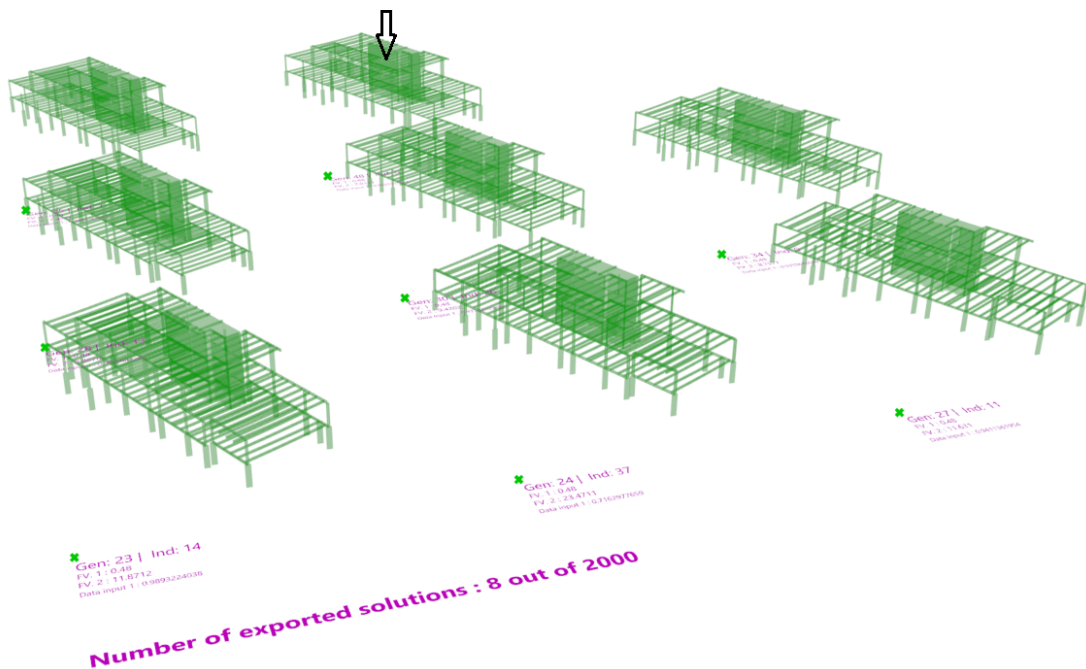


Figure 5.8: Visual representation of selected geometries of Strandhagen. The black arrow indicates the optimal solutions with lowest LCA footprint.

Table 5.5: Results from the analysis in Case 2. The original LCA footprint and cross sections are provided at the bottom for reference. The model has already been analyzed by external stakeholders, and no new analysis on utilization has been conducted or included.

Solution	LCA footprint [kg CO ₂ -eq]	Largest util.	Column 1	Column 2	Column 3	Beam 1	Beam 2	Beam 3
{48;0}	7013	0,98	Glulam 14x63	HEA 240	Glulam 14x63	IPE 220	IPE 300	HEA 240
{35;1}	7457	0,98	Glulam 14x54	HEA 240	Glulam 14x63	IPE 220	IPE 300	HEA 240
{34;0}	8758	0,94	Glulam 14x54	HEA 240	Glulam 14x63	HEA 200	IPE 300	HEA 240
{30;26}	9420	0,94	Glulam 14x54	HEA 240	Glulam 14x63	HEA 220	IPE 300	HEA 240
{28;13}	9868	0,99	Glulam 14x63	HEA 300	Glulam 14x54	HEA 180	IPE 300	HEA 240
{27;11}	11 631	0,94	Glulam 14x63	HEA 300	Glulam 14x63	HEA 240	HEA 200	HEA 240
{23;14}	11 871	0,99	Glulam 14x63	HEA 300	Glulam 14x54	HEA 240	IPE 300	HEA240
{24;37}	23 471	0,72	Glulam 14x63	HEA 300	SHS 180X8	IPE 300	SHS 250X8	HEA 240
Original	22 842	-	RHS 200x120x8	RHS 200x120x8	SHS 80x6	IPE 360	IPE 360	IPE 200

In order to compare the LCA footprint of the optimized solution with the original model obtained from the IFC file, an approximate calculation was performed to assess the LCA footprint of the original building. In the original IFC file, several elements were custom-created on-site, which posed a challenge for analysis. To address this, the average volume per meter of the beams was used as a reference. The goal was to find alternative geometries with a similar volume per meter and, consequently, a comparable structure and LCA footprint. Based on this approach, IPE360 beams were selected for the lower levels, while IPE200 beams were chosen for the upper levels. The columns are of the cross-section SHS and obtained from the cross section library in Karamba3D. All cross sections used for the analysis may be seen at the bottom of Table 5.5.

As a result, the LCA footprint for the beams and columns of the original model was 22,842 kg CO₂. Compared to the optimized cross sections, which achieved an LCA footprint of 7,013 kg CO₂-eq, it is reduced by over 70 %.

Figure 5.9 presents a color mapping scheme applied to all elements in the optimal solution {48;0}. In this representation, elements with the lowest contribution to the overall LCA footprint are visually distinguished by a green color. As the elements' contributions increase, they are progressively depicted in shades of yellow and red, indicating a higher level of impact. Based on the color mapping, it is evident that the columns on floor 2 and the beams on the top floors have the highest impact on the LCA footprint.

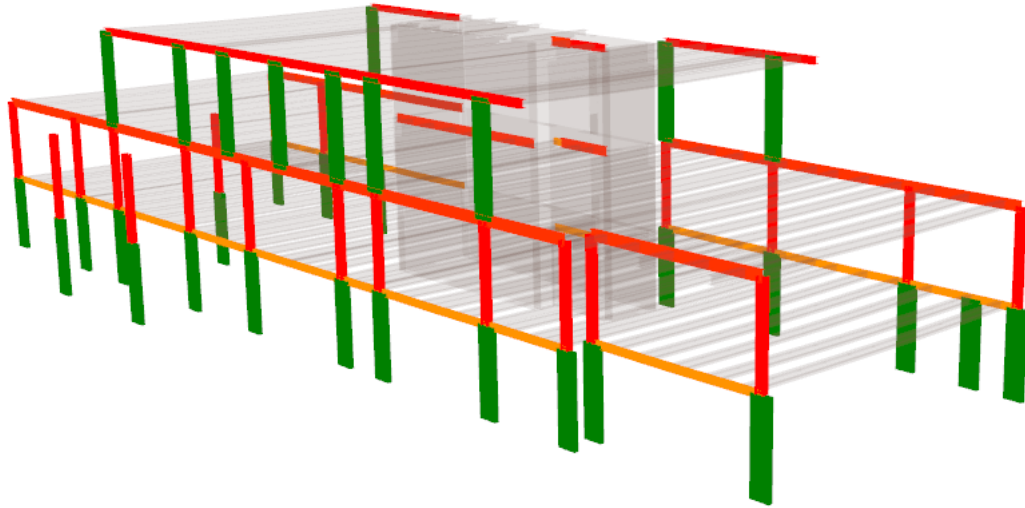


Figure 5.9: Color mapping of elements showcasing their LCA footprint in Strandhagen.

5.3 Case 3 - Buebygget

Buebygget, located in Bergen, Norway, is the most complex structure in this thesis. It encompasses eight floors and a three-story cellar, making it the largest building in the project. The original model was designed in Revit and consists of various elements such as beams, columns, walls, and slabs.

5.3.1 IFC to lines

In Case 3, the process of extracting structural elements from the IFC file is the same as in Case 1. It involves the utilization of GeometryGym in Grasshopper, as explained in Section 4.1.

In this particular case, the extraction process is more complex due to the large number of elements involved. It requires carefully sorting the required structural elements across all categories, here beams, columns, slabs, and walls. This sorting procedure is to identify and extract the specific elements needed for further analysis and design. The total amount of elements was reduced from 3179 elements to 1451 elements after this procedure.

During Case 3, it was observed that certain elements lacked a height parameter, indicated by a z-vector value of zero. To ensure accurate analysis, these elements had to be identified and excluded to avoid any potential errors in the next stages.

Following the sorting of elements, they are scaled and sent from the initial Grasshopper file, labeled as *Buebygget 1) Geometry* in Appendix A, to the subsequent stage, using Speckle.

5.3.2 Creating structural model

Creating the structural model for Buebygget presents greater complexity than in earlier cases. The building consists of a combination of small- and large slabs, columns, beams, and walls. Additionally, the structural configuration varies between the lower- and higher levels, and the level divisions do not apply uniformly to all elements. Furthermore, the large file size complicates the task. Given these challenges, there is no straightforward and generalized approach to connect the entire building seamlessly. A more tailored strategy is required to address the specific intricacies of the structural model.

When there are a large number of elements in a structural model, it becomes difficult to generalize a single approach that efficiently handles all of them. The complexity arises from the diverse characteristics, configurations, and interconnections of these elements. Each element may have unique properties, such as varying sizes, shapes, materials, and load-bearing capacities. Additionally, the relationships and interactions between different elements may vary significantly. Some elements may directly intersect or connect with each other, while others may have indirect dependencies.

Moreover, the sheer size of the model may result in computational challenges and increased processing time. It becomes impractical to apply a uniform strategy across the entire model, as it may not efficiently address the specific requirements and constraints of individual elements.

The structural model for Buebygget involves using columns, beams, slabs, and walls obtained from the Speckle platform. The slabs are defined by their boundary edges, while the walls are represented as surfaces.

To create the structural model, several steps are taken. Firstly, small elements are sorted out. Then, some elements are converted from curves to lines by exploding and transforming the curves into linear parts to meet the requirements of Grasshopper components that demand lines instead of curves. The slabs are linearized by utilizing their edges. Additionally, Case 3 includes ramps with non-horizontal surfaces, which are sorted out.

Next, planes are created to establish the levels within the model. This is done by generating planes from the slabs with a tolerance of 40 cm. If the height of a level is within 40 cm of another level, they are considered a single level. This results in the creation of 12 levels.

To streamline the model, the cellar levels, which mainly consist of walls and slabs that are not extensively analyzed in the thesis, are removed. Elements located below the fourth level are therefore removed. All elements are checked to determine if their center is above or below the chosen level, and if they are below, within a specified tolerance, they are removed from the model. The model is shown before and after the removal of the cellar in Figure 5.10.

After the cellar is removed, the walls in the model are trimmed at the top and bottom to align with the level height. This adjustment is essential because the initial walls may have different heights at their top and bottom compared to the other elements. By trimming the walls, they may properly snap to the correct common nodes established earlier. However, it is important to note that the trimming of walls has minimal impact on the overall structural analysis and optimization process since the walls primarily serve as support for the beams and remain independent entities.

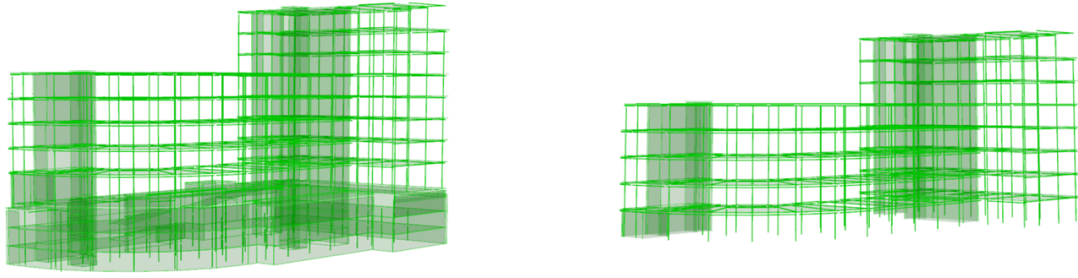


Figure 5.10: Before and after removing the cellar in Buebygget.

The walls have different intersections at each floor, so the common nodes created from the first floor's wall edges and columns are duplicated throughout the entire structure. Only columns and walls are snapped to these nodes. Beams and slabs have their connections.

For the beams, a series of steps are followed. Firstly, the beams are sorted into their respective levels using the method described in Section 4.2.1. Then, all endpoints of the beams are moved to the closest level, ensuring that all beams lie at the correct level. Beam nodes are created by placing them at the average point between two adjacent points. The beams are adjusted to these nodes if they fall within a specified tolerance, similar to how columns and walls are adjusted to common nodes. Finally, the beams are sorted again by levels, as the subsequent file processing will involve iterating through each floor with the beams.

Regarding the slabs, they are connected to the beams on their sides, replicating their typical arrangement in real buildings. The slabs retain their original form from the IFC file and are adjusted to align with the closest level with the height of other elements in the model.

To create artificial beams between slabs and beams, the following steps are performed using levels and meshing techniques:

Slab edges are divided based on the desired mesh size, resulting in the creation of points along the edges. These points are projected onto the sides of the beams on each floor. This projection process involves mapping each slab point onto the corresponding beam, resulting in a significant amount of data. By leveraging the use of leveled elements, the output data size is significantly reduced. Without leveling, the output data would be approximately six times larger, resulting in slower processing and around two million points. However, by iterating through each floor, the data size is reduced to approximately 300,000 points for a mesh size of 2 meters. To determine the closest point on a beam for each slab node, the slab points are connected to their nearest neighbors within a specified distance in the cloud of points on the beams. Next, two neighboring points from the slab nodes are identified for the designated beam points. Lines are then created to connect these slab points with the corresponding slabs on both sides of the beam. In Figure 5.11, the artificial beams connecting the slabs with the beams in Case 3 are illustrated.

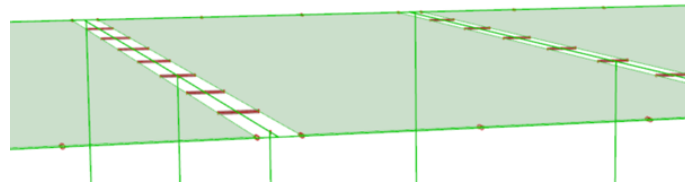


Figure 5.11: Artificial beams, marked in red, between slabs and beams in Buebygget.

To establish connections between beams and columns, artificial beams are created between the beams and the common nodes formed when columns and walls are aligned. This process ensures that the structural system effectively transfers forces from the slabs to the beams and the columns, ultimately distributing them to the ground.

In certain instances, slabs and beams depend exclusively on support from walls. Therefore, it becomes necessary to incorporate artificial beams to establish connections between walls and slabs, as well as between walls and beams. Subsequently, the beams and columns undergo a shattering process at their intersections with other elements or artificial nodes. Intersection points in walls and slabs are saved and included in the meshes created to form a shell structure.

Once all elements are connected, and the building achieves stability, the methods described in Section 4.2 are utilized to sort elements into levels and define their supports. Finally, the columns, beams, slabs, walls, wall inclusion points, slab inclusion points, support points, and support lines are sent to Speckle, where they are prepared for optimization. The results of the geometry editing process to create a structural model for Case 3 may be observed in Figure 5.12.

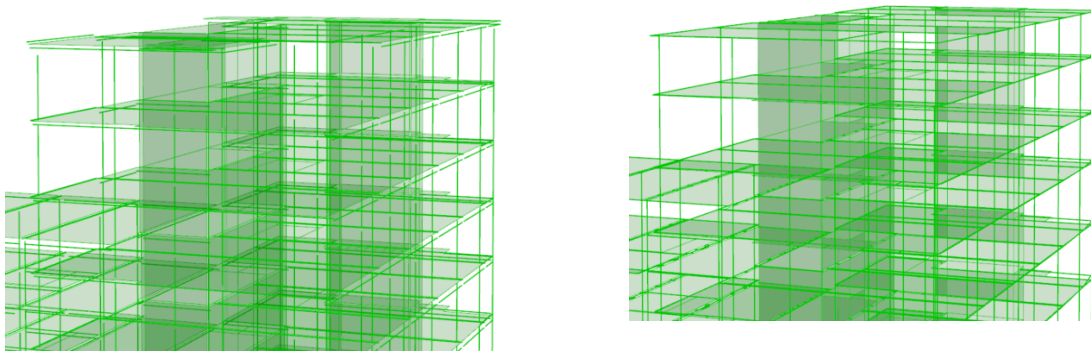


Figure 5.12: Before and after operating on the geometry of Buebygget.

5.3.3 Optimization

The necessary elements are imported from Speckle and connected to their respective paths, indicated in pink, in the final Grasshopper file named *Buebygget 3) Optimization* in Appendix A.

Columns and beams are both divided into three groups, where each group has elements from three consecutive levels. When optimizing, each set of three consecutive floors will be aligned with the same cross section.

Walls and slabs are each assigned a single cross section. The walls and slabs use a concrete cross section of C45, with walls of a width of 40 cm and slabs of a height of 40 cm. However, it is important to note that the slabs and walls are not included in the analysis, and their cross sections are not included in calculating the LCA footprint.

The applied loads for the building are obtained from Table 4.1, with a load of -4 kN/m^2 distributed on all slabs in the building.

In this case, the *Beam-Joint* component is utilized on all artificial beams.

5.3.4 Analysis and results

In the case of Buebygget, a large-scale model, the limitations of Wallacei X analysis become evident, necessitating manual optimization methods. To address the increased demand for stronger elements, extra cross sections are introduced into the cross section table. Manual optimization provides an alternative method to achieve desirable results without relying on Wallacei X. As the script enables real-time visualization of utilization and the LCA footprint, adjustments may be made to achieve an acceptable solution. This process is applied to Buebygget, and the results are presented in Table 5.6. The optimization efforts reduce the LCA footprint of columns and slabs to 215 388 $\text{CO}_2\text{-eq}$, with a maximum utilization of 1.09. Additionally, two other elements exhibit utilization values above 1, while the remaining 3000+ elements are below 1. A color mapping of the LCA footprint for the elements is illustrated in Figure 5.13.

Table 5.6: Results from the manual optimization and analysis in Buebygget. The original LCA footprint and cross sections are provided at the bottom for reference. The model has already been analyzed by external stakeholders, and no new analysis on utilization has been conducted or included.

Solution	LCA footprint [kg $\text{CO}_2\text{-eq}$]	Largest util.	Column 1	Column 2	Column 3	Beam 1	Beam 2	Beam 3
Manual	215 388	1,09	Glulam 28x63	Glulam 28x63	Glulam 14x63	HEA 340	HEA 340	HEA 340
Original	596 530	-	SHS 200x12,5	SHS 200x8	SHS 200x8	HEA 500	HEA 500	HEA 360

In the original IFC file, a significant number of columns and beams demonstrate similarities across all floors. To simplify the analysis, the average cross section of beams and columns from the bottom and top floors is utilized. This simplification leads to an estimated LCA footprint of 596 530 $\text{kg CO}_2\text{-eq}$ for the beams and columns. The results are documented in Table 5.6, following the manual optimization phase. Once again, the optimization efforts showcase a remarkable reduction of over 60 % of the LCA footprint, as observed in Case 1 and Case 2.

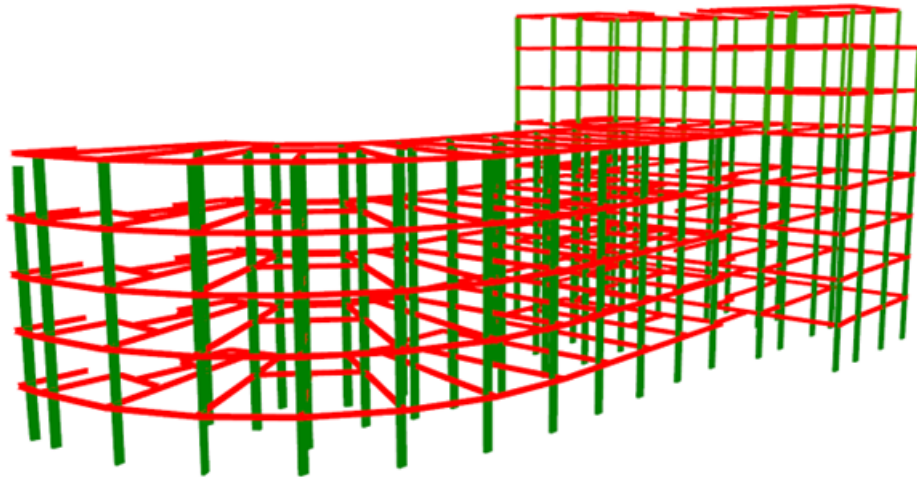


Figure 5.13: Color mapping of elements showcasing their LCA footprint in Buebygget.

6 Discussion

In all three cases, the results demonstrate a significant reduction in the LCA footprint, exceeding 60% compared to the original IFC file. Furthermore, the methodology employed throughout the research demonstrates effective performance in Case 1 and Case 2 without requiring significant adjustments. However, in the case of Case 3, optimization using Wallacei X proved to be unfeasible, though manual optimization within the Grasshopper environment yielded satisfactory results. These findings suggest that the methodology employed in this research is particularly effective for smaller-sized buildings like Case 1 and Case 2. For larger-sized buildings such as Case 3, a more manual approach is required. To accommodate the complexities involved, the procedure includes the creation of artificial beams within the structural model. Additionally, manual optimization guided by the engineer's expertise and knowledge is carried out. Although buildings of a greater size have a manual approach, designers may effectively optimize utilization and reduce the environmental impact of their models.

As explained throughout the thesis, working with IFC files presents several challenges, particularly in structural engineering. One of the primary challenges when working with IFC files is the misclassification of structural elements. This results in a significant amount of cleanup work within the file. This problem arises from human errors, such as incorrect naming and a limitation of the IFC format, which is not specifically designed for structural calculations. Consequently, considerable effort is required to sort and identify only structural elements when creating lines for the structural model. For instance, there may be instances where non-structural elements, like fire isolation around steel beams, are incorrectly classified as load-bearing. Therefore, manual sorting of IFC elements becomes necessary to include only the relevant structural elements.

The thesis uses IFC files obtained from existing buildings, which accurately capture the final and complete state of these structures. The IFC files contain a high level of detail, encompassing various elements representative of the constructed buildings. This presents certain challenges, as the complexity and size of the IFC files may affect the efficiency and generalization of the structural modeling and optimization process. Using IFC files from conceptual stages provided by architects could result in more simplified and rational models. These early-phase IFC files typically contain structural elements that are easier to define and comprehend, resulting in smaller file sizes and smoother computational workflows. Additionally, the simplified nature of early-phase IFC files allows for the development of generalized scripts and procedures that may be applied to a wider range of projects.

The initial objective was to extract specific components from the IFC file and generate a smaller, more manageable IFC file that focuses on the required elements. However, this type of methodology encountered challenges due to the interconnected nature of the elements within the IFC structure. Some models include quantities found in Python, while others have representations easily accessed through GeometryGym. Using GeometryGym proved straightforward for extracting lines in Cases 1 and 3, while Case 2 presented additional challenges that necessitated Python scripting. It depends on how elements are defined when exporting to an IFC format and what information is included.

It is important to note that the IFC files utilized in this thesis were not specifically adapted for this

project. Optimizing IFC files for specific purposes, such as including quantity information during export from FEA software, may enhance the process. Overall, working with IFC files in structural engineering presents challenges, requiring careful considerations and manual interventions to ensure accurate representation and stability within the structural model.

The size of the IFC file itself poses another challenge. It may contain millions of text strings, hence its ability to visualize large and complex buildings. This may result in slower processing times, especially on standard computers when creating structural models. In Case 3, the IFC file contained over one million text strings, which resulted in slow processing when working with the structural model in the file *Buebygget 2) Karamba* in appendix A.

IFC files present difficulties achieving structural stability within FEA software; hence the lines are not connected when extracted from the file. Although it is possible to manipulate the geometry and establish connections within tolerances, the process outlined in this thesis involves a manual approach. Each model required an engineer to determine appropriate tolerances and which elements should be connected based on their understanding of structural behavior and force transfer. Despite these challenges, the methodology presented in this thesis demonstrates that structural engineers may achieve structural stability within the IFC framework.

The LCA calculations employed in this thesis have certain limitations that must be acknowledged. The results obtained from the LCA calculations provide only a partial view of the overall environmental impact. Specifically, this analysis focuses solely on the embodied carbon emissions from A1-A4 of the LCA process. Factors such as maintenance, recycling, and reuse of building elements are not included in this assessment.

In the original models, the IFC files considered various requirements, such as fire safety and secondary loads. As a result, the original cross sections may have elevated standards compared to the stability checks and utilization analysis conducted during the early phase of the method. This means that the structural analysis performed in the cases may not fully capture all the detailed demands that could be present in the actual structure. It is also noticeable that the utilization of glulam timber, which has a negative footprint, proves to be superior to other materials in the cross section table. This superiority contributes significantly to the overall reduction in the LCA footprint across all cases. However, they are all based on real EPDs and are therefore valid in an early-phase analysis.

The approach of limiting the LCA scope to A1-A4 was adopted due to practical reasons, as predicting future outcomes beyond the construction stage is challenging. However, it does neglect important considerations, such as the environmental benefits of using recycled steel. Nevertheless, it is possible to enhance the model presented in this thesis by incorporating additional elements that have a positive environmental impact, such as low-carbon steel or low-carbon concrete. Expanding the model to include these elements would improve its effectiveness and provide a better understanding of the environmental impact of building structures. Furthermore, it would enable the comparison of environmentally-friendly alternatives with traditional materials, offering insights into the advantages of sustainable products. By expanding the analysis to encompass more environmentally significant factors, a more comprehensive assessment could be achieved. This may contribute to a more holistic evaluation of the environmental impact of building structures and facilitate informed decision-making in favor of sustainable construction practices.

When defining cross sections in the Grasshopper file, only default concrete types such as C30 and C45 were used. These concrete types were not specifically adapted for each case to include the appropriate amount of reinforcement. This approach was chosen because the primary goal was to develop a methodology rather than create a fully refined model. Consequently, optimizing the reinforcement for various cross sections was not given priority.

To achieve valid solutions, defined in this research as solutions with a utilization value below 1, an additional cross section table was created with steel cross sections. This choice was motivated by their strong alignment with the utilization component and higher strength than other cross sections listed in the original cross section table. Using concrete and timber materials could result in infinite utilization values, leading to a crash in the optimization process. The examination of case studies unveiled notable shear forces in specific beam elements within the building, particularly in Case 2 and Case 3. Regarding the concrete cross sections, the reinforcement was set to default values. Modifying the reinforcement could potentially have decreased the utilization of concrete elements. However, to improve the accuracy of the utilization, the decision was made to employ steel exclusively where needed.

The current model organizes elements by type, such as beams and columns, while sorting them by their respective floors. However, there are alternative approaches to sorting the elements in a more detailed manner. One possibility is to create zones within each floor, allowing a more focused analysis. For instance, specific attention may be given to middle columns, which typically experience higher forces and may require tailored solutions. Additionally, sorting beams based on their length could be considered, providing an opportunity to optimize the cross sections of the longest beams individually. By implementing such targeted sorting strategies, the optimization process becomes more practical. It allows for the efficient utilization of resources while still addressing critical elements that require specialized design considerations. This approach strikes a balance between optimizing the model and accommodating the logistical constraints of the project, leading to more effective and manageable outcomes.

Utilizing parametric design tools such as Grasshopper and Karamba3D in the methodology offers several advantages. These tools provide users with the ability to easily adjust and modify parameters, allowing for rapid exploration of different design options. They also contribute to automating repetitive tasks and calculations; no hand calculations are necessary when calculating the LCA footprint as described in Section 4.3. Parametric design streamlines the process from an IFC file to its LCA footprint by enabling the exchange of information and parameters between different software components, such as Grasshopper and Speckle. Furthermore, these tools offer powerful visualization capabilities. By utilizing software like Grasshopper or other similar tools, designers may create visual representations of the structure, utilization of elements, and LCA footprint. This visual feedback provides valuable insights and facilitates informed decision-making throughout the design process.

In the context of Case 3, the utilization of Wallacei X for optimization led to system crashes. However, the subsequent adoption of a manual optimization approach proved effective. This outcome suggests that leveraging the expertise of civil engineers and employing parametric models, as described in this research, facilitates the comparison of solutions and ultimately leads to improved outcomes. The manual optimization process affords a more comprehensive examination

and evaluation of diverse design possibilities. Furthermore, the parametric approach simplifies incorporating additional factors, such as pricing considerations. Consequently, this enhanced capability empowers engineers to make informed decisions and attain superior results.

By incorporating this methodology in the conceptual stage, structural engineers may streamline the process of creating FEA models and selecting cross sections that meet utilization requirements. The time saved could then be redirected towards optimizing the project regarding the LCA footprint using the approaches outlined in this thesis. This optimization may be achieved through automated tools like Wallacei X or by manually exploring different cross sections and assessing their impact on the LCA footprint.

Looking to the future, engineers must deliver structurally sound constructions that are also environmentally sustainable. Engineers who prioritize minimizing the LCA footprint while ensuring safety may gain a competitive edge in the market. The ability to balance structural integrity and environmental considerations may be a crucial aspect of engineering practice, influencing the perception and success of professionals in the field.

7 Conclusion

The potential of utilizing IFC files in structural modelling and LCA optimization is undoubtedly significant. With a model such as the one presented in this thesis, structural engineers may efficiently analyze BIM models created by others during early phases and gain insights into their environmental impact. When structural engineers identify and propose alternatives for environmentally impactful elements, it integrates sustainability into the design process and promotes environmentally conscious practices.

The utilization of parametric design tools in the optimization of BIM models offers numerous benefits, including flexibility, automation, integration, and visualization. By embracing these tools, designers may effectively optimize utilization and reduce the environmental impact of their models, contributing to more sustainable building practices.

The results of the case studies demonstrate a significant reduction in the LCA footprint. The methodologies employed in Case 1 and Case 2 proved effective and optimal. However, when dealing with a large-scale building in Case 3, the Wallacei X analysis did not yield satisfactory results. Instead, a manual approach proved to be more successful in this particular scenario, also resulting in a reduction of the LCA footprint.

With a consistent reduction of over 60 % across all three cases, the method's effectiveness becomes apparent, providing valuable insights into environmentally conscious building design. The results demonstrate the potential benefits of parametric design in optimizing the cross sections and the LCA footprint. By utilizing parametric design techniques, engineers may easily explore various cross-section configurations and assess their impact on factors such as LCA footprints and structural utilization.

The methodology presented offers a practical approach for structural engineers to integrate sustainability into their design processes. By optimizing the LCA footprint and considering the environmental impact of structural choices, engineers may contribute to a more sustainable built environment. The potential for time savings and the competitive advantage gained by prioritizing environmental considerations further highlight the significance and relevance of this approach in the field of structural engineering.

8 Future work

Although this thesis provides a methodology for optimizing IFC files using LCA results, some areas require further development and refinement. The following suggestions present potential areas for future research:

The current model focuses primarily on optimizing linear elements such as beams and columns. It should include shell elements like walls and slabs in the optimization process as well.

The cross section table in the existing solution is not thorough, relying on default Karamba3D materials and selected sizes. Future work could involve expanding the cross section table and ensuring their properties align with Norwegian standards. Optimized levels of reinforcement for concrete elements could also be considered, with their properties aligned accordingly. It would also be beneficial to incorporate products like hollow core slabs, obtaining their material properties directly from the producers.

While the current model focuses on environmental impact, it could expand its scope to include other parameters such as economy and time considerations. Different design solutions may have varying perspectives regarding cost, construction duration, and spatial optimization.

The model relies on using components from Karamba3D when calculating element utilization, specifically designed for steel elements. It should be extended to address using other materials, such as concrete and timber. As an example, the Grasshopper plug-in *Beaver* may be utilized on timber elements.

Finally, the LCA footprint is obtained manually. This process should be automated by extracting GWP values directly from EPDs when adding new cross sections to the cross section table.

Bibliography

- 3.3.2: Shell Cross Sections - Karamba3D 2.2.0. (n.d.). <https://manual.karamba3d.com/3-in-depth-component-reference/3.3-cross-section/3.3.2-shell-cross-sections>
- Aec Systems Ltd., S. (2023). *Speckle*. Retrieved 25th May 2023, from <https://speckle.systems/about/>
- AISC. (2023). Structural Steel Sustainability | American Institute of Steel Construction. <https://www.aisc.org/why-steel/sustainability/>
- Anand, P. (2016). 10 Advantages of Concrete as Construction Material | Why Concrete - CivilDigital -. <https://civildigital.com/10-advantages-concrete-construction-material-concrete/>
- Budig, M., Heckmann, O., Hudert, M., Ng, A. Q. B., Xuereb Conti, Z., & Lork, C. J. H. (2021). Computational screening-LCA tools for early design stages. *International Journal of Architectural Computing*, 19(1), 6–22. <https://doi.org/10.1177/1478077120947996>
- BuildingSmart. (2023). Industry Foundation Classes (IFC) - buildingSMART International. <https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/>
- Chen, Q., & Taylor, D. (2020). Economic development and pollution emissions in Singapore: Evidence in support of the Environmental Kuznets Curve hypothesis and its implications for regional sustainability. *Journal of Cleaner Production*, 243, 118637. <https://doi.org/10.1016/J.JCLEPRO.2019.118637>
- Crawford, R. H., & Cadorel, X. (2017). A Framework for Assessing the Environmental Benefits of Mass Timber Construction. *Procedia Engineering*, 196, 838–846. <https://doi.org/10.1016/J.PROENG.2017.08.015>
- Curran, M. A. (2006). US EPA Life Cycle Assessment: Principles and Practice. <https://nepis.epa.gov/Exe/ZyNET.exe/P1000L86.txt?ZyActionD=ZyDocument&Client=EPA&Index=2006%20Thru%202010&Docs=&Query=&Time=&EndTime=&SearchMethod=1&TocRestrict=n&Toc=&TocEntry=&QField=&QFieldYear=&QFieldMonth=&QFieldDay=&UseQField=&IntQFieldOp=0&ExtQFieldOp=0&XmlQuery=&File=D%3A%5CZYFILES%5CINDEX%20DATA%5C06THRU10%5CTXT%5C00000002%5CP1000L86.txt&User=ANONYMOUS&Password=anonymous&SortMethod=h%7C-&MaximumDocuments=1&FuzzyDegree=0&ImageQuality=r75g8/r75g8/x150y150g16/i425&Display=hpfr&DefSeekPage=x&SearchBack=ZyActionL&Back=ZyActionS&BackDesc=Results%20page&MaximumPages=1&ZyEntry=11&slide>
- Dodoo, A., Gustavsson, L., & Sathre, R. (2009). Carbon implications of end-of-life management of building materials. 53, 276–286. <https://doi.org/10.1016/j.resconrec.2008.12.007>
- Duan, Z., Huang, Q., & Zhang, Q. (2022). Life cycle assessment of mass timber construction: A review. *Building and Environment*, 221, 109320. <https://doi.org/10.1016/j.buildenv.2022.109320>
- Dudka, S. (1997). Environmental Impacts of Metal Ore Mining and Processing: A Review; Environmental Impacts of Metal Ore Mining and Processing: A Review. <https://doi.org/10.2134/jeq1997.00472425002600030003x>
- EPD-Norway. (n.d.). What is an EPD? <https://www.epd-norge.no/hva-er-en-epd/>
- EPD-Norway. (2022). *What types of EPDs exist and how are they different?* (Tech. rep.). [https://www.epd-norge.no/getfile.php/1324556-1656447960/Dokumenter/Hvilke%20typer%](https://www.epd-norge.no/getfile.php/1324556-1656447960/Dokumenter/Hvilke%20typer%20)

-
- 20EPDer % 20finnes % 20og % 20hvordan % 20er % 20de % 20forskjellige % 20engelsk %
20ver%20mai%202022.pdf
- European Commission. (2023). Life Cycle Assessment (LCA). <https://eplca.jrc.ec.europa.eu/lifecycleassessment.html>
- GeometryGym. (2023). *Geometrygym*. Retrieved 26th May 2023, from <https://manual.karamba3d.com/3-in-depth-component-reference/3.6-results/3.6.6-utilization-of-elements>
- Global Cement and Concrete Association. (2023). Sustainability Benefits of Concrete : GCCA. <https://gccassociation.org/sustainability-benefits-of-concrete/>
- Grutters, B. (2022). Industry Foundation Classes (IFC) - buildingSMART Technical. <https://technical.buildingsmart.org/standards/ifc>
- Guinée, J. B., Heijungs, R., Huppes, G., Zamagni, A., Masoni, P., Buonamici, R., Ekvall, T., & Rydberg, T. (2011). Life cycle assessment: Past, present, and future. *Environmental Science and Technology*, 45(1), 90–96. <https://doi.org/10.1021/ES101316V>
- Howes, B. (2023). TT Toolbox for Grasshopper | Thornton Tomasetti. <https://www.thorntontomasetti.com/capability/tt-toolbox-grasshopper>
- IEA. (2018). Cement technology roadmap plots path to cutting CO2 emissions 24% by 2050 - News - IEA. <https://www.iea.org/news/cement-technology-roadmap-plots-path-to-cutting-co2-emissions-24-by-2050>
- IEA. (2019). Global Status Report for Buildings and Construction 2019 – Analysis - IEA. <https://www.iea.org/reports/global-status-report-for-buildings-and-construction-2019>
- ISO. (2018). ISO - ISO 16739-1:2018 - Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema. <https://www.iso.org/standard/70303.html>
- Kamari, A., Kotula, B., & Schultz, C. (2022). A BIM-based LCA tool for sustainable building design during the early design stage. *Smart and Sustainable Built Environment*, 11(2), 217–244. <https://doi.org/10.1108/SASBE-09-2021-0157>
- Karamba3D 2.2.0. (2021). A.2.3: Tips for Designing Statically Feasible Structures - Karamba3D 2.2.0. <https://manual.karamba3d.com/appendix/a.4-background-information/a.4.3-tips-for-designing-statically-feasible-structures#table-a.2.3.2-loads-for-typical-scenarios>
- Makki, M., Showkatbakhsh, M., & Song, Y. (2019). Wallacei Primer 2.0. <https://www.wallacei.com/>
- McNeel, R., et al. (2023). Rhinoceros 3d, version 7.0. *Robert McNeel & Associates, Seattle, WA*.
- Moschas, F., & Steirou, E. (2013). Statistical estimation of changes in the dominant frequencies of structures in long noisy series of monitoring data. *Mathematical Problems in Engineering*, 2013, 1–11. <https://doi.org/10.1155/2013/216860>
- Preisinger, C. (2013). Linking structure and parametric geometry. *Architectural Design*, 83(2), 110–113.
- Ramsden, K. (2020). Cement and Concrete: The Environmental Impact — PSCI. <https://psci.princeton.edu/tips/2020/11/3/cement-and-concrete-the-environmental-impact>
- Rezaei, F., Bulle, C., & Lesage, P. (2019). Integrating building information modeling and life cycle assessment in the early and detailed building design stages. *Building and Environment*, 153, 158–167. <https://doi.org/10.1016/J.BUILDENV.2019.01.034>
-

-
- Sala, S., Reale, F., Cristóbal-García, J., Marelli, L., & Rana, P. (2016). Life cycle assessment for the impact assessment of policies. Life thinking and assessment in the European policies and for evaluating policy options. *Joint Research Centre*, 28380, 53. <https://doi.org/10.2788/318544>
- Shaun. (2023). Life Cycle Stages – One Click LCA Help Centre. <https://oneclicklca.zendesk.com/hc/en-us/articles/360015064999-Life-Cycle-Stages>
- Soust-Verdaguer, B., Llatas, C., & Moya, L. (2020). Comparative BIM-based Life Cycle Assessment of Uruguayan timber and concrete-masonry single-family houses in design stage. *Journal of Cleaner Production*, 277, 121958. <https://doi.org/10.1016/J.JCLEPRO.2020.121958>
- Tam, V. W., Zhou, Y., Shen, L., & Le, K. N. (2023). Optimal BIM and LCA integration approach for embodied environmental impact assessment. *Journal of Cleaner Production*, 385, 135605. <https://doi.org/10.1016/J.JCLEPRO.2022.135605>
- The Norwegian EPD Foundation. (2018). ENVIRONMENTAL PRODUCT DECLARATION. www.epd-norge.no %20<https://www.epd-norge.no/epder/bygg/heltreprodukter/prosjektlimtre>

Appendix

A Grasshopper files

Case 1

Tennebekk 1) Geometry.gh

Tennebekk 2) Karamba.gh

Tennebekk 3) Optimization.gh

Case 2

Strandhagen 1) Geometry.gh

Strandhagen 2) Karamba.gh

Strandhagen 3) Optimization.gh

Case 3

Buebygget 1) Geometry.gh

Buebygget 2) Karamba.gh

Buebygget 3) Optimization.gh

B Python file

Strandhagen.ipnyb

C Tennebekk

Gen: 27 | Ind: 1
FV. 1 : 0.48
FV. 2 : 21.1951338022

Gen: 49 | Ind: 2
FV. 1 : 0.48
FV. 2 : 20.1159873559

Gen: 19 | Ind: 34
FV. 1 : 0.48
FV. 2 : 26.1649985783

Gen: 22 | Ind: 0
FV. 1 : 0.48
FV. 2 : 25.8411411671

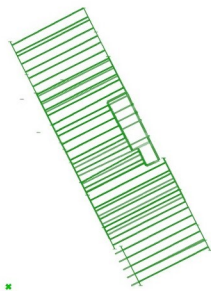
Gen: 23 | Ind: 1
FV. 1 : 0.48
FV. 2 : 25.5842634947

Gen: 11 | Ind: 24
FV. 1 : 0.48
FV. 2 : 32.5120661717

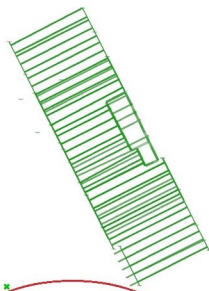
Gen: 14 | Ind: 25
FV. 1 : 0.48
FV. 2 : 31.9313310881

Gen: 18 | Ind: 25
FV. 1 : 0.48
FV. 2 : 26.1582974217

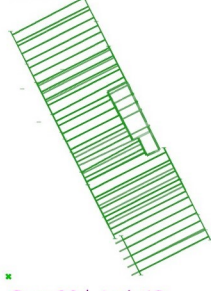
D Strandhagen



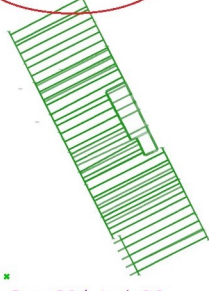
Gen: 35 | Ind: 1
FV. 1: 0.48
FV. 2: 7.4571



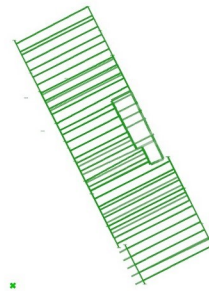
Gen: 48 | Ind: 0
FV. 1: 0.48
FV. 2: 7.0131



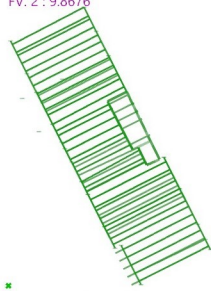
Gen: 28 | Ind: 13
FV. 1: 0.48
FV. 2: 9.8676



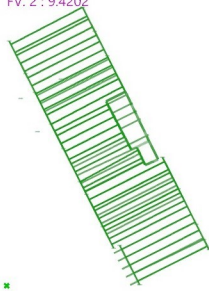
Gen: 30 | Ind: 26
FV. 1: 0.48
FV. 2: 9.4202



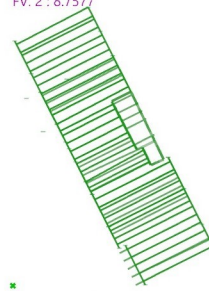
Gen: 34 | Ind: 0
FV. 1: 0.48
FV. 2: 8.7577



Gen: 23 | Ind: 14
FV. 1: 0.48
FV. 2: 11.8712



Gen: 24 | Ind: 37
FV. 1: 0.48
FV. 2: 23.4711



Gen: 27 | Ind: 11
FV. 1: 0.48
FV. 2: 11.631



 **NTNU**

Norwegian University of
Science and Technology