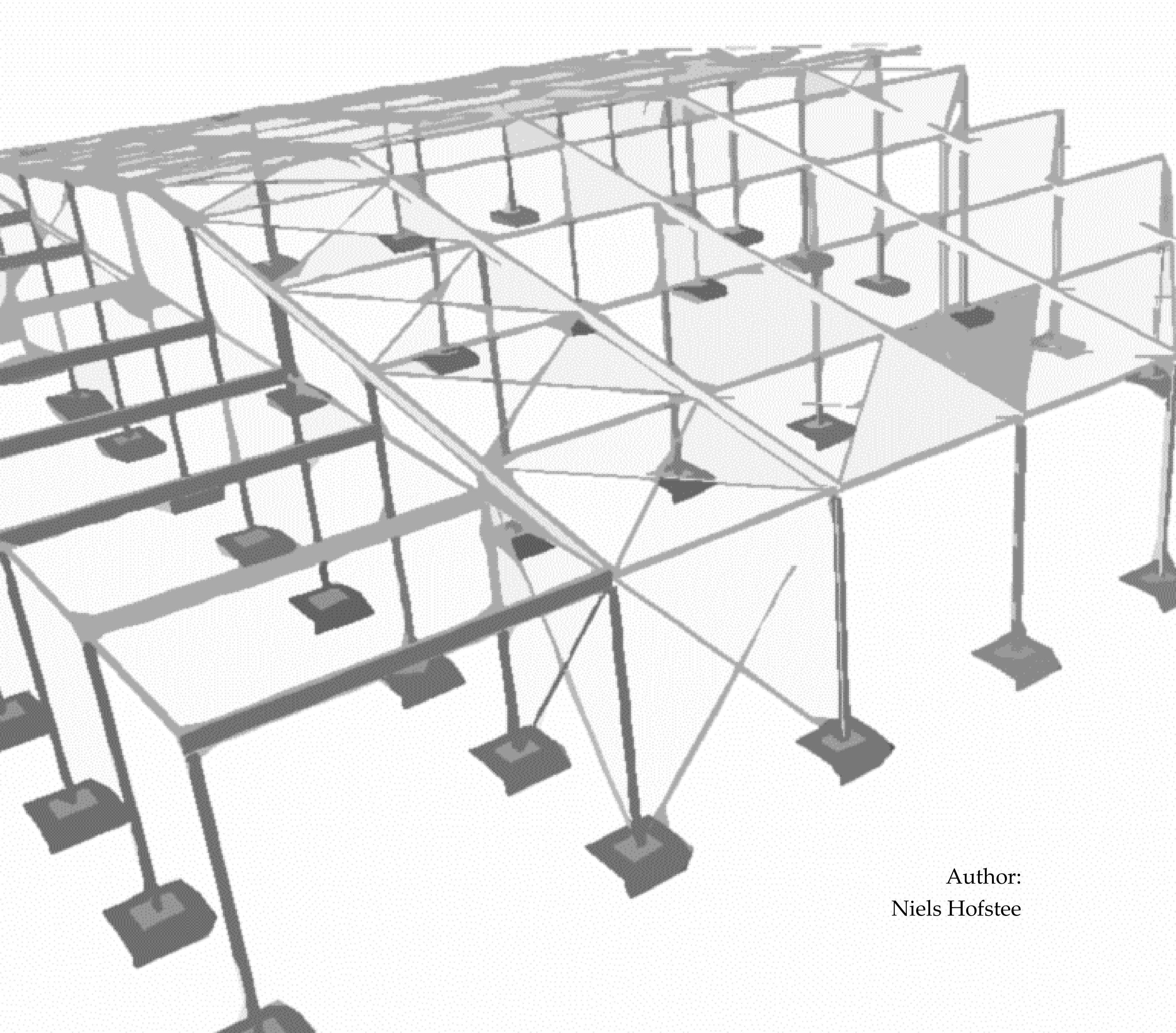
MSc Thesis report

DATA DRIVEN DESIGN IN SUSTAINABILITY

Development of software prototype to aid the conceptual design of industrial warehouses on sustainability



Author:
Niels Hofstee

DATA DRIVEN DESIGN IN SUSTAINABILITY

Development of software prototype to aid the conceptual design of industrial warehouses on sustainability

by

Niels Hofstee
Student number: 4087364

in partial fulfilment of the requirements for the degree of

Master of Science
in Building Engineering

at the Delft University of Technology and

in cooperation with White Lioness technologies,

to be defended publicly on the 16th of November 2018.

Thesis committee:	Prof. ir. R. Nijsse	TU Delft
	Dr. ir. H.R. Schipper	TU Delft
	Dr. H.M. Jonkers	TU Delft
	Dr. ir. J.L. Coenders	White Lioness technologies

PREFACE

Before you I present my Master's Thesis. It finishes my study Building Engineering at the faculty Civil Engineering at the TU Delft. I'm proud of this work which was the biggest challenge in my study career.

Looking back, the seed for this graduation work was planted at the course Special Structures. The lectures from Jeroen Coenders inspired the tech enthusiast in me. The parametric program Grasshopper immediately fascinated me. I discovered an entire new field in Civil Engineering. One connected with innovation, digital transformation, optimisation and flexible parametric design. It lead to an internship in this field and finally my graduation work.

I'm grateful for the amount of people that helped me to produce my final work. First, I would like to thank my entire graduation committee for the support during the graduation period. In particular I would like to thank Jeroen Coenders who introduced me to the computational design field and was the driving inspiration behind my work.

I thank the people at White Lioness technologies for input in this thesis. For the opportunity to do additional work besides my graduation. In particular I would like to thank Jorn van der Steen for developing the Grasshopper plugin for me.

Then I would like to thank Jan-Pieter den Hollander and Frank Maatje from Bouwen met Staal for the time they invested in finding data. Connected with this, I would like to thank Jelle Brantsma of Voortman steelgroup, Bob Soetekouw from GB steelgroup, and Mart de Klerk from ASK Romein for providing me with valuable data on industrial warehouses. Furthermore, I would like to thank Concretio for providing me with calculation reports on industrial warehouses.

Next to this, I would like to thank all people that provided feedback on my report and work. In particular Celine and Ina for proof reading parts of my report. I thank Michaël for brainstorming with me on my thesis and for the fun collaboration in creating Grasshopper tutorials that kept me motivated during the graduation time.

Lastly I would like to thank my parents and family for the unlimited support and trust. I wasn't here without them.

Niels Hofstee,
November 2018

ABSTRACT

The construction sector contributes to a significant portion of global energy consumption, CO_2 emissions and the use of material resources. As climate change escalates with rapidly increasing temperatures, more extreme weather patterns and rising sea levels, an urge to reduce the environmental impact is evident.

A discrepancy is visible between early structural design and the required information for performing an environmental assessment. The first phase of design is an excellent time to steer the design to be more sustainable. However, today's procedures, involving Life Cycle Assessment (LCA), only provides feedback at a later phase. This is due to the required highly detailed information for performing this complex and time consuming assessment. An assessment tool that supports the conceptual design would help to reduce the environmental impact.

The research project aims to show the feasibility of sustainability feedback using analytical methods on stored building data in early design. It provides LCA feedback in the conceptual design phase. A proof-of-concept is developed on industrial warehouses showcasing the functionality of the framework. Lacking the required building data and LCA scoring, a fully automated script is created with the parametric plugin Grasshopper, to generate the warehouse data. Due to time constrains and performance limitations, the scope was limited to the structural system of beams, columns, purlins and braces. The script randomly iterates over its parameter domain, automatically changing the geometry, performing a structural optimisation and calculating the LCA. This assessment is limited to a cradle-to-gate system boundary. Data is automatically written to a web-hosted database on the Packhunt.io platform of White Lioness technologies. The feedback is based on retrieving data from the database, applying a performance based algorithm, and finding the parameter that decreases the LCA score the most. Results are visualised to the user.

The result is the Interactive Design Assist (IDA) prototype that suggests parameters to the engineer in reducing the LCA score. This information provides guidance in the conceptual design phase on structural design of industrial warehouses. It demonstrates the large potential of reusing 'knowledge' inside the design process to improve building designs.

CONTENTS

1 INTRODUCTION

1.1 MOTIVE

In today's world it's hard to deny the change in climate. Increasing global temperatures are unmistakable. Extreme weather is a constant topic in the media and sea level rise threatens to consume many islands in the Pacific Ocean (Nace, 2017). One of the main cause of these effects is the emission of greenhouse gasses in our atmosphere, released by countless processes in today's world. Every country under the 2015 Paris Agreement has the responsibility of reducing their greenhouse gas emissions according to their 'Intended Nationally Determined Contribution' (Kim et al., 2017).

The European Parliament signed the 2010 legislation 'on the energy performance of buildings' which demands a decrease of carbon dioxide emission of 20% in 2020 with as reference 1990 (EU, 2010). Implementation by the Dutch government (Rijksoverheid, 2018) sets a performance level of being 'carbon free' in 2050. Furthermore, the building sector is set to construct only 'near energy neutral' buildings by 2020 and have all the available buildings in the Netherlands energy neutral in 2050.

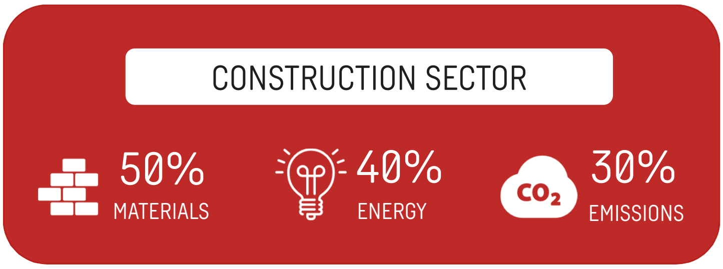


Figure 1.1: Total percent of environmental impact of construction sector related to world usage (Hollberg and Ruth, 2016)

Reasons for such a large focus on the construction sector, is the fact that it is one of the world's biggest contributors to global CO_2 emissions; about one third of the world's total emission output (Figure 1.1). Furthermore, about 50% of the world's raw materials are used in this sector (Hollberg and Ruth, 2016; EU, 2010). In the material processing, production, transportation, and the assembly at site, a large number of greenhouse gasses and pollutants are emitted (Kim et al., 2017).

> *Sustainability is explained in the dictionary as "...avoidance of the depletion of natural resources in order to maintain an ecological balance."*

Energy use in buildings count for about 40% of the world's primary energy (EU, 2010). During the last decades of the previous century, public awareness increased. This mainly triggered by increased cost of energy generation. A trend towards passive housing, zero emission and energy-positive buildings is clearly visible, reducing the operational energy use of a building (Lolli, Fufa and Inman, 2017). However, De Wolf et al. (2016) points out that while operational energy from heating, cooling, ventilation, hot water and lighting can decrease during a buildings lifetime, the embodied energy that is stored in materials with the production, transport and construction cannot. An average of more than half of this embodied energy is due to the structural system (Danatzko and Sezen, 2011). In the design phase of a building the team of architects and engineers could already steer on a low impact building. Assessment of sustainability is critical in this process.

1.2 SUSTAINABILITY

Life Cycle Assessment (LCA) is often used for assessing the environmental impact of a building. The procedure is labour intensive and complicated to perform due to large amount of components, a long building lifespan (introducing many uncertainties) and the often conflicting and lacking information on sustainability of building materials (Hollberg and Ruth, 2016).

> *Life Cycle Assessment (LCA) is a method to calculate the environmental impact of an object (this can be a product or building) during its life time. Each phase in the cycle is judged on their effects and collected in 10 impact categories. For comparison reasons these categories are translated to unit cost by weighing factors. The cost is proportional to the money needed to mitigate the environmental effects (Stichting Bouwkwaliteit, 2017).*

The procedure contains a set of phases (Figure 1.2) that distinguishes the building process. A cradle-to-gate implementation takes into account the process from resource extraction, transport, to final product until the gate of the factory. Cradle-to-grave looks at the entire life

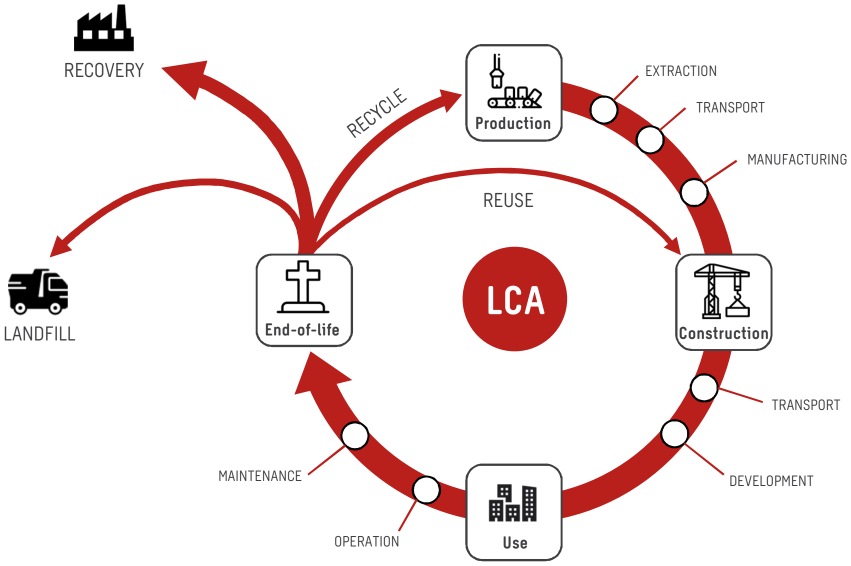


Figure 1.2: The life cycles that are included in the LCA procedure

cycle of the product, including demolition of the building with recycling of products and materials.

> *Conceptual design is the first design stage in which initial concepts are made. It is characterised by creativity and starting from a blank sheet of paper.*

This process requires detailed building information to give an accurate environmental representation. Only final design and construction are suited for this. Yet, this is in direct conflict with design process characteristics. When more information is collected and decisions are made, changes become less effective and more costly (Figure 1.3). Therefore it is essential to steer the sustainability early on. With more knowledge in conceptual design, the engineer is more aware of design choices influencing the sustainability. It can actively steer on a certain performance level, well regarded as the future of design (Turrin, Von Buelow and Stouffs, 2011). The need for accurate and fast assessment tools that support this process is evident.

1.3 DATA OF BUILDINGS

Latest development is the construction industry is the wide adoption of parametric design and the extensive use of Building Information Modelling (BIM) software. Parametric design allows for a flexible model that has the possibility of optimisation. BIM models contain a wide range of building information. Connect this with the rise of

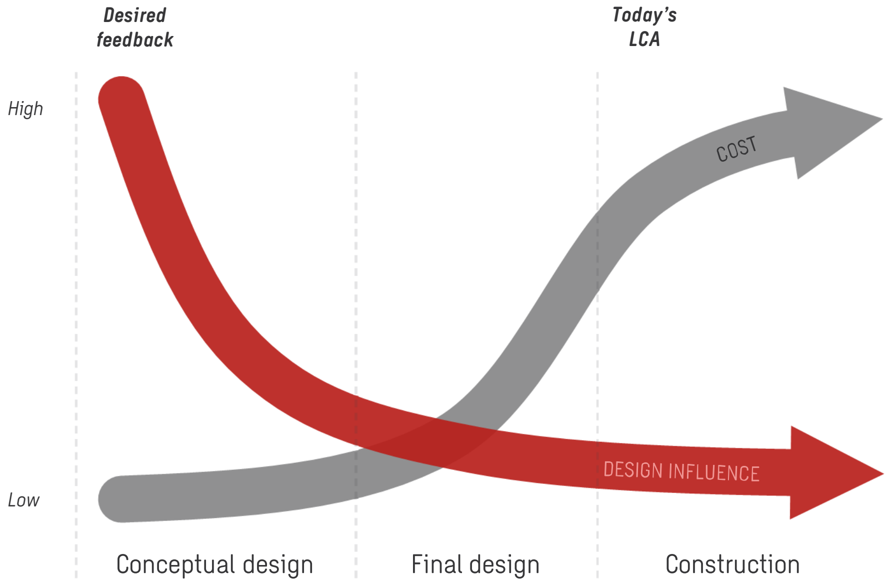


Figure 1.3: Influence of design decisions on cost with each stage of design (based on Bragança, Vieira and Andrade (2014))

big data, opportunities emerge to use this information in solving and estimating complex problems.

Examples are numerous in the medical industry (Jootoo et al., 2017) where Machine Learning (ML) algorithms are implemented to for example predict length-of-stay of patients. However, few implementations in structural design are seen despite large quantities of data in BIM applications. Problems are that this data is locked in programs, often unstructured, and lacking in quality and consistency (Solihin et al., 2017). Furthermore, during the starting phase of this research project, it was found that sustainability data on buildings is even harder to find as companies are highly protective of their information. Not one company could provide building data with some indication of sustainability.

> *Big data refers to massive and complex datasets, made up of a variety of data structures which are too big and complex for traditional data processing (Grolinger et al., 2013). It is an emerging field, fueled with new technologies to offer ways for extracting value out of information (Becker, Tilman and Curry, Edward and Jentzsch, Anja and Palmetshofer, Walter, 2016). Recently, the term refers to predictive analysis with help of Machine Learning (ML) algorithms. These are "...statistical models that can make predictions based on new instances of similar data." (Jootoo et al., 2017).*

1.4 VISION

The combination of advancements in BIM and the characteristics of big data, a solution with estimating the sustainability using already build structures can be envisioned. The 'knowledge' that is acquired and stored can be used to support the conceptual design phase. A knowledge database or platform can be envisioned. Recent research by Gkioka (2018) on a knowledge model for concrete viaducts, shows the benefit of this method. It significantly reducing design time, thereby increasing the amount of possible design iterations and allowing the architects and engineering to focus on the creative process. A knowledge platform allows the reuse of acquired knowledge in previous designs. Each completed project can enrich the database with new information. Three components are required for this platform: data of building, a storage method, and a way to extract useful feedback from this data (Figure 1.4).

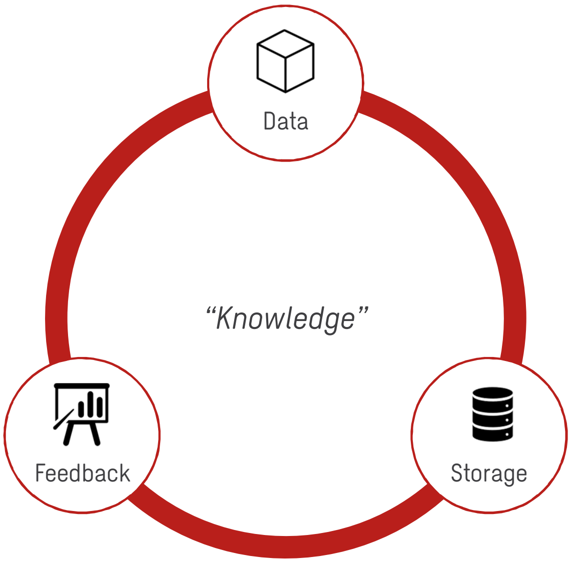


Figure 1.4: The three components data, storage and feedback to create a knowledge platform

1.4.1 *Data*

Data driven design is often limited by the underlying data. High quality building information is therefore required to provide accurate feedback. Information is often locked in difficult to access file formats (Solihin et al., 2017). Recent research by Eastman (2016) shows progression in this area with the transfer of BIM models and IFC files to databases. Problems with unstructured and inconsistent data however remains. The envisioned web based platform connects the data providers with the data users (Figure 1.5). Dataset on different types of buildings (industrial warehouses, offices, houses) or calculation knowledge in a certain field (e.g. strength of 3D printed concrete) can be envisioned. Key platform characteristics are scalability and flexibility. A wide range of functions can be added including cost analysis, sustainability feedback, and calculation support.

> *Industry Foundation Class (IFC) is a neutral and open file standard used to sharing information between project teams and commonly used software applications in design, construction, maintenance and operation (BuildingSMART, 2018). The data scheme is developed by BuildingSMART, a world-wide advocate for adoption of open standards.*

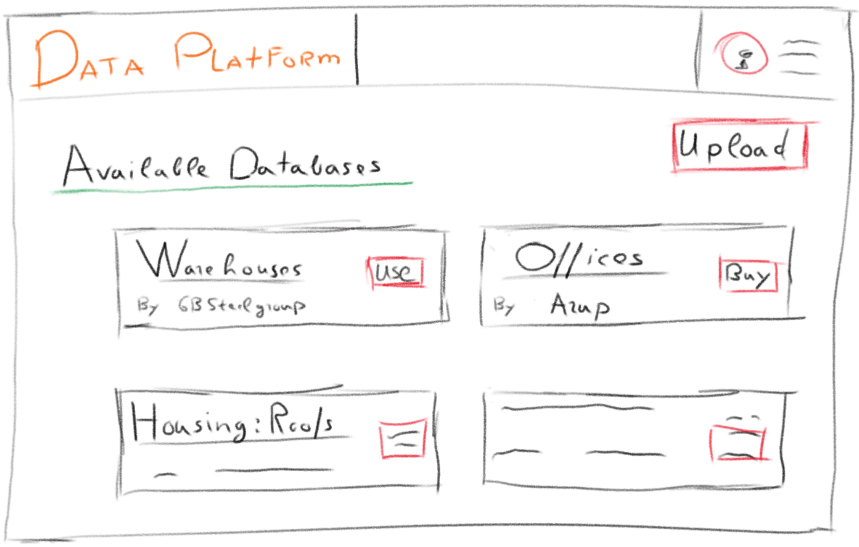


Figure 1.5: Sketch of a web-based data platform

1.4.2 *Storage*

The database containing building information requires the interaction between users and the database. A users can requests design feedback from parametric or BIM applications. This requires integration inside these programs. The database is queried and feedback send back to the model interface. Another data stream is the enrichment of the database by uploading BIM and IFC files. The envisioned cloud based storage is easy to scale with increasing users. At any point in time new data can be added. The storage model is robust to deal with incomplete and unstructured data.

1.4.3 *Feedback*

Feedback is generated from the database to steer the design process. A key characteristic is flexibility as a wide range of parameters and starting points are possible. Implementation is performed in a BIM or parametric application as shown in Figure 1.6. The algorithm learns and improves with each design iteration. Feedback is easy to interpret to allow for a quick continuation of the design process. It provides a comparison between design alternatives. Additionally, nearest data points are visualised in full geometry to give the user an quick impression of feedback result. Also possible is the addition of performance based design in which the user specifies the desired performance level. With each design iteration the algorithm provides new feedback and suggestions.

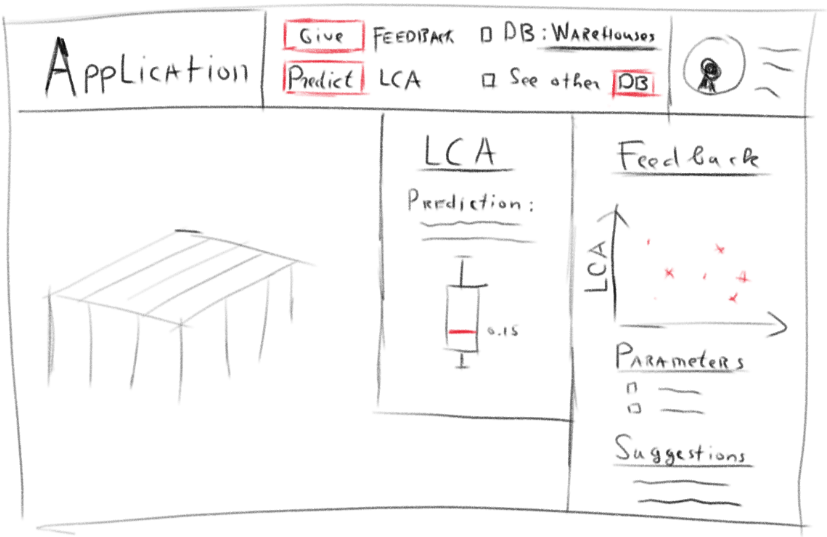


Figure 1.6: Sketch of feedback model build into a BIM application

2 OBJECTIVE

2.1 MAIN OBJECTIVE

It is essential for future building projects to take sustainability into account since the early stage of design. In conceptual design the influence of design decisions is large, unfortunately the uncertainties as well. Analytical technologies based on underlying data, can help to steer the design process in reducing the environmental impact. However, such a design support application is not yet present. Concluding, the main objective can be stated as follows:

"Research and develop a software prototype to aid the decision making process regarding the sustainability of building designs in the conceptual design phase"

2.2 RESEARCH QUESTIONS

From the main objective the following research questions can be stated:

1. *What research is done in providing feedback and aiding the decision making process in early design?*
   This question provides an overview of what kind of methods are available today, that provide the user with feedback. The advantages and limitations of these methods are presented and show opportunities for the software prototype.

2. *What software framework needs to be developed to aid the decision making process in the conceptual design phase?*
   This results in a framework based on the vision, available tools, and key characteristics of the conceptual design phase. This information is necessary for the feedback algorithm to function properly.

3. *How can the environmental impact of buildings be calculated?*
   Research is needed to what defines sustainability and what aspects are common and required. This results is sustainability scoring according to today's industry standards.

4. *What feedback algorithm can be implemented to steer the design in reducing the sustainability impact and which also complies with the conceptual design phase characteristics?*

A comparison of different algorithms is made to see the opportunities and limitations. New ideas are researched and the best option is worked out in a proof-of-concept. The result is a tool that can be used by designers.

2.3 SCOPE

Within the master's thesis time frame, some restrictions are necessary. The implementation is restricted to a proof-of-concept. This shows the potential and vision of the framework in reducing the environmental impact in building design. Along with this proof-of-concept implementation, it is decided to generated a dataset with help of parametric scripting due to lack of real data.

> *A proof-of-concept is a simplified representation of the actual fully working implementation. It is meant to demonstrate the potential of the method and serves as a starting point for future research.*

The building type is restricted to industrial warehouses due to their limited parameter complexity. This results in relatively easy scripts which significantly reduce computation time. Furthermore, only the structural load bearing system is modelled to even further simplify the process.

The sustainability calculation in the form of a LCA is a complex process. However, the most simple implementation, with a cradle-to-gate system boundary, is created. This included the resource extraction, transport and production of the material.

2.4 METHODOLOGY

This research project is divided in several parts. First, the proposed conceptual framework is further elaborated with the requirements and features of the prototype. The limited scope is further discussed. Next part is the data generation and storage. The parametric scripts are discussed in detail. This included the geometry creation, structural optimisation and sustainability calculation. Finally, generated data is transferred to a database for storage. Next step is the implementation of the feedback algorithm. The working of the analytical method is discussed in detail.

3

EARLY DESIGN FEEDBACK TOOL

Software development starts with a general understanding on the system with the respective requirements, features and development environment (Surendran et al., 2015). Stellman and Greene (2005) describes the software structure with a focus on the purpose of application, the overall description including functions, constrains and interfaces. This will be discussed in the following sections.

3.1 FRAMEWORK OVERVIEW

The proposed system is presented in Figure 3.1. A number of phases can be distinguished in this model. The 'design process' refers to the conceptual design phase of a building. Here the requirements, limitations and wishes are discussed within the design team. At this point design alternatives are explored resulting in different parameter combinations and values. Decision is made to choose certain parameters and values. At this point the proposed tool takes over.

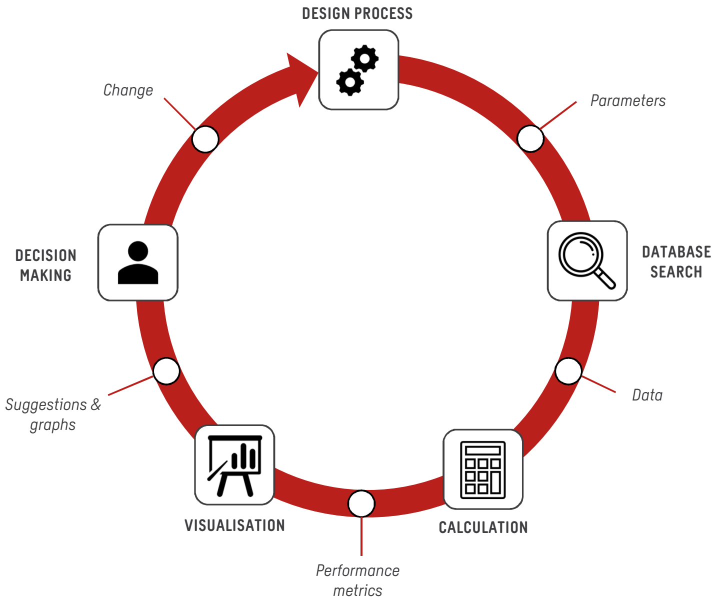


Figure 3.1: Framework overview of the software implementation

A database search is performed with the available parameters. The dataset contains design alternatives of industrial warehouses. A comparison is made between the available data and the received parameters, and a selection of similar designs is made. Data is extracted and filtered on better performing design in terms of LCA score. This is based on the mean LCA score. The performance improvement for each parameter and value is calculated. The best performing parameter is selected. This in visualised to the user in graphs. The suggested parameter is set out against the possible performance improvements. Quantitative information on the LCA scoring is added along side.

> *A framework is a central concept of programming providing a standard way to build and deploy applications (Riehle and R. Gross, 1998). It increases productivity and reduces development time. It essentially outlines the mechanism on which applications can be build.*

Based on the provided suggestion and extensive information in the graphs, the user is able to make a decision on the parameter value. Initiation of a new feedback loop occurs when the design is changed. The process is fast and occurs within a second. This means that the user is able to quickly investigate a wide range of starting points with correlating suggestions.

3.2 USE CASES

Exploring different scenarios help shape the feature set and requirements. The proof-of-concept implements sustainability prediction of the load bearing structure. Users are structural engineers and architects, participating in the conceptual design process of a building. A set of use cases are possible.

Parametric design applications are excellent in exploring a design space. Due to their nature of defining logic with components and associates, high flexibility is possible (Coenders, 2011). Integration with the proof-of-concept is possible and results in a dynamic and fast feedback loop. The user can adjust parameter values and iteratively receive feedback on the sustainability. This implementation requires a robust integration with the parametric software. Relevant is the presentation of detailed information and the interaction between input parameters of the tool and the parametric logic.

Today's buildings are increasingly modelled in BIM applications (Solihin et al., 2017). The information in these models can directly be integrated in the proof-of-concept tool. A connection between the tool and the BIM information is necessary for a good user experience. The function of the proof-of-concept is to provide feedback as the building model is being created.

In engineering firms, building designs are often started with pen and paper. Sustainability feedback is needed in a much simpler and quicker way. The proof-of-concept can aid in this stage by having a separate front-end. Parameter values can be assigned and feedback generated accordingly. Advantage is the easy and quick way feedback can be retrieved. A disadvantage is that the design space is explored in a limit fashion. This use-case requires a front-end to be build. It needs to be easy to use and quick to learn.

3.3 CONCEPTUAL DESIGN PHASE

The framework is implemented in the conceptual design phase. The characteristics of this phase are essential for the requirements and features. Conceptual design is defined by creativity and starting from a blank sheet of paper (Appendix A.1). Østergård, Jensen and Maagaard (2016) points out the key aspects in this design stage.

Conceptual design is characterised by lack of fixed starting point. Multiple ways of initialising the project are possible. Often requirements and wishes from the client and architect result in initial schematic drawings and sketches of system configurations (Bragança, Vieira and Andrade, 2014). The order of which requirements are set depends on many factors: the architectural firm, the client, the type of building, the way the project team is organised, etc. Starting points could be a desired shape by the client or a certain vision or idea by the architect. Other possibilities are environmental limitations or a required performance level. These aspects are subjected to rapid change in the conceptual design.

The information quality in conceptual design is often poor and low in quantity. Decisions are made with low confidence and high uncertainty. The discrepancy between available information and flexibility in structural design is the largest in this phase (Heidegger, Coenders and Rolvink, 2014). According to Østergård, Jensen and Maagaard (2016) predicting the consequences of decisions in the conceptual design phase is very difficult but crucial for meeting high performance goals. Simulation can be very powerful but is often evaluative instead of proactive in this phase. Even sophisticated software programs are mostly suitable for compliance, benchmarking and assessing multiple design variations and are often not usable in the conceptual design phase.

This early in the process, one has to make assumptions which naturally come with uncertainties. Simulations and calculations contain uncertainties by using simplifications and schematisation (Coenders, 2011). Sensitivity analysis can be performed to aid the decision process. This can answer "what-if" questions by calculating correlations.

This indicates the size and direction of the change in performance with changes in input (Østergård, Jensen and Maagaard, 2016). Next to this, the assumptions can change at any moment.

A vast design space is visible due to th possible parameters combinations. Exploration of this space is key to find solutions within the requirements. Without the use of computational techniques it is difficult and labour intensive to determine a local optimum. A single simulation only reveals one point in the global design space without providing guidance to improve the design.

3.4 REQUIREMENTS

Based on the conceptual design implementation and the possible use-cases, specific requirements are set for the proof-of-concept application. Its main function is to provided sustainability for aiding the decision making process. It needs to steer on lowering the LCA score by providing parameters and performance improvement indications. It should suggest a best parameter but still be flexible for the user to create their own path.

The prototype implementation needs to be flexible to deal with rapid changes of design, different starting points and multiple parameter combinations. The user should receive feedback on the sustainability in such a time that it will actually be used in the design process. Automation of this process is essential to limit disruptions in the design workflow. It requires quick input of parameters and limited interactions with the tool to receive feedback.

3.5 FEATURES

The feature set of the proof-of-concept needs to be clear from the beginning to allow for an efficient development. Important is the interaction between the user and the application. The front-end has input fields for parameters and their respective values. It consists of a graph with data values and a written suggestion of best parameter. Furthermore, the feedback contains both quantitative information with the LCA scoring and comparative information between design options. This supports the user to make decisions. Options are available to go back in the process and change the parameters. This results in new feedback and suggestions. At least one parameter needs to be chosen for the tool to work.

3.6 LIMITED SCOPE

Within the proof-of-concept implementation additional limitations further reduce the scope. Both the sustainability implementation and the structural design of the industrial warehouses are further detailed.

3.6.1 *Sustainability*

A cradle-to-gate system boundary is chosen for decreasing the complexity of the calculation. This assessment is based on the embodied energy in the materials. It takes into account the extraction of raw materials, transport to production facilities and the production of the material. These are all environmental impacts up to the 'gate' of the factory. The Netherlands is chosen as building location. Here, environmental data on materials, products and processes are provided by the Nationale Milieu Database (NMD) (NMD, 2017). In this research project, version 2.1 of November 2017 is used. The average values of the materials in this database are used. The environmental impact of the industrial warehouses is calculated in €/m^2/year. Weighing factors from the NMD are used to calculate the impact in Euro. This is the cost to mitigate the environmental effects. The area is related to the 'bebouwd-vloeroppervlak' (BVO), build floor area. For the service life of the industrial warehouse 50 years is taken according to the Eurocodes.

Additional aspects of the sustainability are provided in Appendix B. Here a thorough look on the influences of these choices is presented, as well as a more detailed look into the different system boundaries and phases of the LCA. Furthermore, the NMD as data source is discussed in detail.

3.6.2 *Structural design*

A scope reduction is applied to the structural design to simplify the process and increase performance of the automated data generation. Only the load bearing structure of the industrial warehouse is considered. It consists of the beams, columns, purlins and braces. For this proof-of-concept, the material steel (S235) is selected. This to simplify the automated calculation. Within the Netherlands most warehouses are constructed in steel.

A load transfer from roof to beam to column is implemented. Furthermore, a braced frame with hinged connections is taken as stability system. The structural optimisation only considers the load combination of permanent + wind in Serviceability Limit State (SLS) for the columns, and permanent + snow in SLS for the beams.

A thorough description is available in Appendix C. It discusses the applied loads, and the performance criteria on deformations and stresses. Finally it presents design calculations to validate the automated structural optimisation in Grasshopper.

4 DATA GENERATION

The data used in the proof-of-concept is generated with help of parametric scripting. The plugin Grasshopper (Rutten, 2017) for Rhinoceros (Robert McNeel & Associates, 2017) is chosen for this process. Reasons are numerous including the availability of additional plugins such as the structural analysis plugin Karamba (Preisinger, 2013), and the extensive knowledge of the researcher on this specific software. The scripted warehouses are designed according to the available Eurocode and National Annex.

> *Parametric design is "... a geometric representation of a design with components and attributes which are parametrized" (Turrin, Von Buelow and Stouffs, 2011). Within this design space, a logic can be build based on components and their relations. The combination with a graphical user interface results in an efficient way of assessing design alternatives.*

4.1 INDUSTRIAL WAREHOUSE IFC DATA

The 25 collected IFC files are from the companies Voortman Steelgroup, GB Steelgroup, and ASK Romein, provide guidance is shape and functionality of the warehouses. An overview of these files is found in Appendix E. The warehouses contain many differentiations in shape, span type, arrangement of columns and roof. Roofs are mostly flat but sometimes a pitch is seen. The spans are made with beams, intermediate supporting columns (mid columns), and trusses. The combination of trusses and mid columns is seen a lot as well. Multiple blocks of warehouses are present in the total arrangement. Often these are of a different span type and direction. Functionality wise, intermediate floors are created to serve as office space.

4.2 WORKFLOW

Parametric scripts with a steel load bearing structure of industrial warehouses are created. The workflow presented in Figure 4.1, generates a range of design alternatives. The script starts with the 'parameters' which are randomised to generate industrial warehouse designs. These parameters are linked to the 'geometry' on which a 'structural

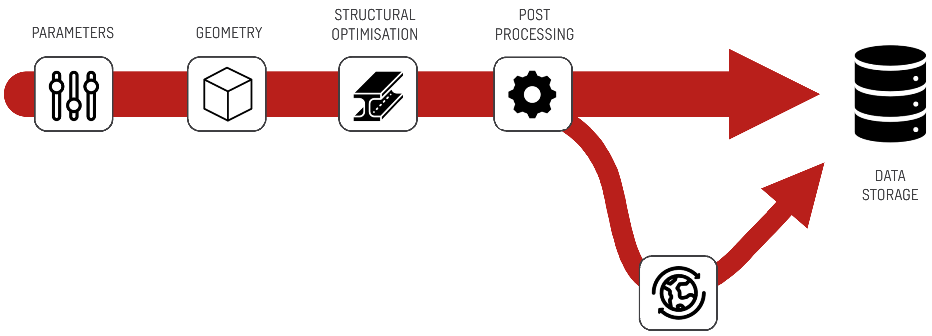


Figure 4.1: Grasshopper workflow enables the data generation of industrial warehouses

optimisation' is performed. Optimal sections are chosen considering the different loads and structural parameters. In the 'post processing', data on parameters, materials and sections are extracted from the model. This is both used in the 'LCA' procedure and stored in the 'database'. The structural analysis, LCA and the post processing phase are saved as Grasshopper clusters. Clustering allows for a group of Grasshopper components to be saved as a separate file. Reference to this file can be made from every Grasshopper script. Changes in the three main clusters are automatically propagated to files it is referenced to.

4.3 PARAMETERS

Parameters are the starting point of the model. Selection is performed based on a typical design process and available data of build industrial warehouses. The design booklet of 'Bouwen met Staal' (Bouwen met Staal, 2013), the steel association, on industrial warehouses is used as additional reference.

4.3.1 *Selection of parameters*

The following changeable parameters are integrated in the Grasshopper scripts. An visual overview is given in Figure 4.2.

- *Width:* This regulates the maximum width of the warehouse. The beams and trusses span this direction.
- *Count:* It controls the length of the warehouse and is directly related to the amount of portal frames. The frame distance times the count results in the length of the warehouse.

- *Height:* This shapes the warehouse in height, measured from ground floor to roof top. The free height is related to the distance from ground floor to bottom of the structural system.
- *Portal frame distance:* The portal frame distance manipulates the distance between the portal frames. A frame is a beam connected by two or more columns.
- *Purlin distance:* The beams are separated by a set distance. At this location purlins are placed.
- *Max span beam:* The maximum distance that a beam spans. At this location an additional supporting column is generated.
- *Additional height:* The additional height for generating a pitched roof. The total height of the pitched roof is the height + the additional height.

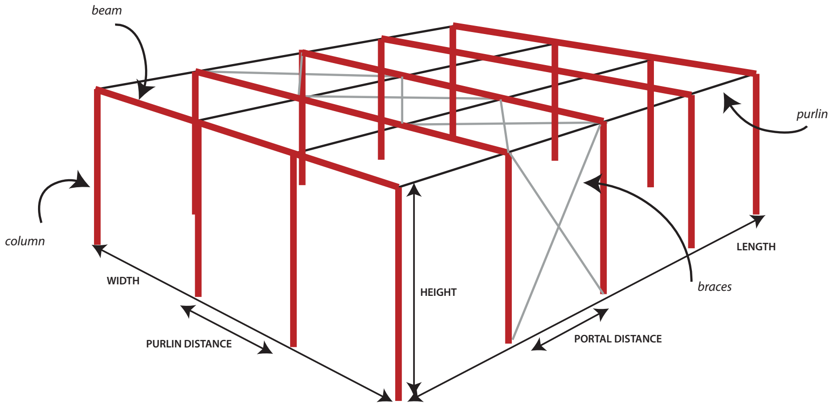


Figure 4.2: The specification of the parameters and elements in the industrial warehouse

4.3.2 *Parameter variation expectation*

The random variation of parameters result in different industrial warehouses. What is certain is the linear relation between structural mass and environmental scoring. This is simply due to the applied weighing factors from the NMD. Analysing the parameter 'height' of the warehouse two observations can be made. First, reducing the height results in shorter columns and thus less material. Second, lowering the height reduces the wind induced forces on the warehouse. This has large consequences for the sizing of the structural elements. It is expected that reducing the height is always better in reducing the impact.

PARAMETERS	B	T	M-C	B PITCH
Width (m)	8 - 20	20 - 60	25 - 100	8 - 20
Count	2 - 6	2 - 8	2 - 14	2 - 6
Height (m)	4 - 14	4 - 14	4 - 14	4 - 10
Portal frame distance (m)	4 - 8	4 - 8	4 - 8	4 - 8
Purlin distance (m)	4 - 6	4 - 6	4 - 6	4 - 6
Max span beams (m)	N/A	N/A	10 - 20	N/A
Add. height (m)	N/A	N/A	N/A	1 - 2

Table 4.1: Parameter constrains for the industrial warehouse designs (B = Beams, T = Truss, M-C = Mid-Column).

Considering the situation that a specific area is required. Two options are available: increasing the length and therefore decreasing the width, and the other way around. Increasing the length of the warehouse not only increases the amount of portals used, but due to wind forces it is also expected that the sections will increase. The other option is to increase the span of the portals. This is limited in range as sections cannot be bigger than a certain size. Changing the width will increase the sections rapidly. It is expected that increasing the length is better than increasing the width

A more difficult to assess scenario is with multiple portal distances. Given a fixed length and width, it is probably better to have less portals with larger sections than a short portal distance with more portals and smaller sections. The purlin distance could be relevant at larger lengths. The wind forces are transferred by compression and tension in respectively the purlins and roof bracing. It will probably be better to use fewer purlins in the warehouse (larger purlin distance), as seen with the beams.

4.3.3 *Parameter constrains*

The multi dimensional design space is restricted by two factors. First, certain combinations of parameters are structurally not feasible given the limited amount of available sections. Second, a wide range of warehouses can be built but are generally seen as unrealistic. This results in a restricted design space, leaving out potential interesting combinations (Figure 4.3). Reasons for including as many options as possible, is to enrich the dataset. Filtering out the unwanted warehouses is required. Implementation is performed at database level (Section 5.3). Table 4.1 and Table 4.2 provides the restrictions of the parameters.

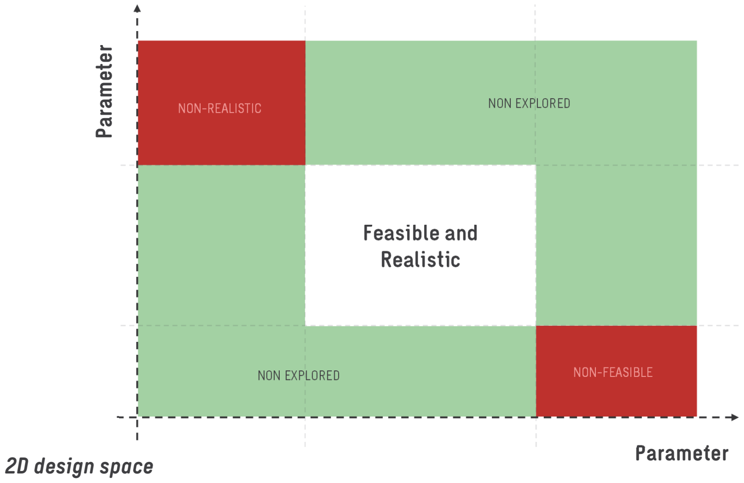


Figure 4.3: Visualisation of a simplified 2d design space (two parameters). Certain parameter combinations are not possible or unrealistic, resulting in a restricted design space.

4.3.4 *Randomise parameters*

Generating a wide range of designs, requires automation and randomisation of input parameters. The plugin 'Design Space Exploration' (DSE) is used to generate a random sample of the design space and iterate over the variables (MIT, 2018). A typical set-up is displayed in Figure 4.4. The 'Sampler' generates a design map according to the attached parameters, amount of samples, seed and sampling type. The Latin Hypercube method is used for this implementation. This statistical sampling technique divides the multidimensional design space in a grid and makes sure that only one sample is present in each row and column (Olsson, Sandberg and Dahlblom, 2003). Main advantages are that it gives a good representation of real variability and that it doesn't require more samples for more dimensions.

The 'Capture' component takes the design map, the related objectives (in this case the LCA score) and the attached parameters, and iterates over the design space.

4.4 GEOMETRY

The next phase is the creation of the geometry. A total of eight different warehouse designs are made. Figure 4.5 shows each model generated in Grasshopper. The relatively easy shapes and elements (beams, columns, purlins, wind braces) allow for a straightforward

PARAMETERS	M-C TRUSS	M-C PITCH	M-C FLOOR	M-C HALL
Width (m)	25 - 100	25 - 100	25 - 100	25 - 100
Count	2 - 14	2 - 14	2 - 14	2 - 14
Height (m)	4 - 14	4 - 12	4 - 14	4 - 14
Portal frame distance (m)	4 - 8	4 - 8	4 - 8	4 - 8
Purlin distance (m)	4 - 6	4 - 6	4 - 6	4 - 6
Max span beams (m)	20 - 35	10 - 20	10 - 20	10 - 20
Add. height (m)	N/A	1 - 3	N/A	N/A

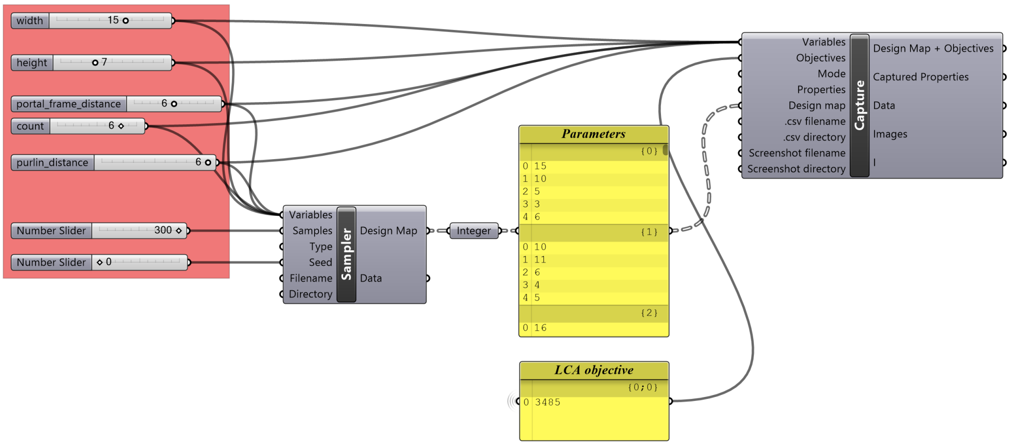
Table 4.2: Parameter constrains for the industrial warehouse designs (B = Beams, T = Truss, M-C = Mid-Column).

parametric script (points and lines). The scripts with beams ((a) and (b)) contain a flat and a pitched roof. The pitched roof is centred at halfway of the width.

The mid column scripts ((c), (d), (e), (f)) have beams that have intermediate column support. A flat roof variant and a pitched roof variant is created. Additionally, there is a variant with an intermediate floor, spanning between the first and second portal frame. It is located half way at the column height. This simulates an office area within the warehouse. The last variant has an additional attached warehouse. It spans with beams without the intermediate columns. The direction of the beams is changed compared to the main warehouse. A full connection between the both warehouses is made.

Trusses are used for the roof in the last two variants ((g) and (h)). One is supported with intermediate columns and one without. The trusses form a warren truss layout. This type is often seen in warehouses due to their relatively low weight and good strength properties. The truss height is governed by the span itself and the factor *l/20*.

For all variants, stability against wind is provided by placing wind braces between the beams and purlins. Multiple locations are possible depending on the length and width of the warehouse. Perpendicular to the wind, two positions are chosen. At the beginning and end of the roof. These are optimal for forces distribution to ground level. Parallel to the wind three positions are possible: begin, mid and end displayed in Figure 4.6). With increasing dimensions (>35 and >80

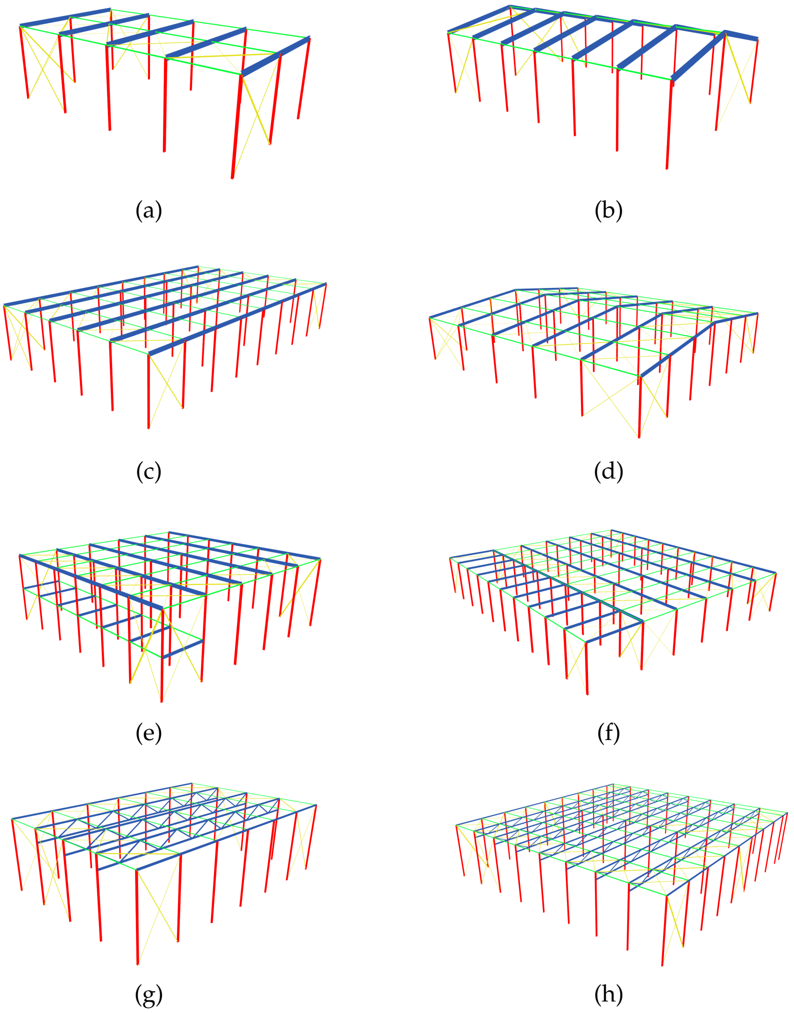


```
{0}
0 15
1 10
2 5
3 3
4 6
{1}
0 10
1 11
2 6
3 4
4 5
{2}
0 16
```

```
{0;0}
0 3485
```

Figure 4.4: Typical DSE set-up with the Sampler randomising the parameters and the Capture component iterating over the design space

meter) these braces are added. The braces in the façade are positioned at the location of the rood braces.



(a)

(b)

(c)

(d)

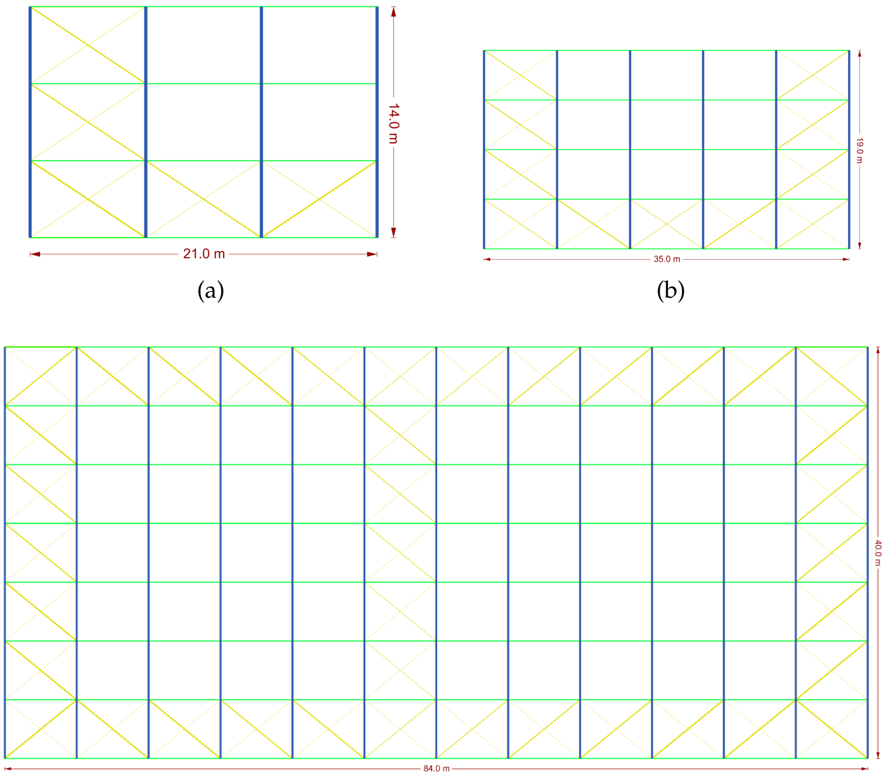
(e)

(f)

(g)

(h)

Figure 4.5: The following warehouses are linked to the tables E.1 and E.2. (a) Warehouse with beams. (b) Warehouse with beams and pitched roof. (c) Warehouse with mid columns. (d) Warehouse with mid columns and pitched roof. (e) Warehouse with mid columns and additional sub floor. (f) Warehouse with mid columns and additional hall. (g) Warehouse with trusses. (h) Warehouse with trusses and mid columns.



(a)

(b)

(c)

Figure 4.6: Wind braces are automatically added with increasing length and width of the warehouse. (a) and (b) are models with beams and (c) is a model with additional mid columns.

4.5 STRUCTURAL OPTIMISATION

The structural analysis and optimisation is performed with the Grasshopper plugin Karamba. This piece of software is chosen for two reasons: first, the plugin allows to keep working in the same environment. This simplifies the process as no data transfer is required. Secondly, Karamba contains the useful component 'optimise cross-section' which selects the appropriate sections for given dimensions and loading conditions. The distinguishable parts of the script are further elaborated. The cluster of Grasshopper objects that perform the optimisation is provided in Figure 4.7. A thorough analysis regarding the starting requirements and the loads is presented in Appendix C.

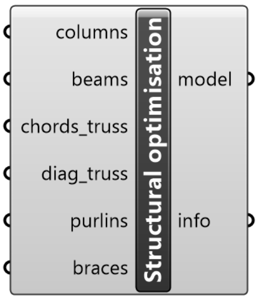


Figure 4.7: Grasshopper cluster that performs the structural optimisation

> *Karamba is a structural analysis program that seamlessly integrates in the parametric environment of Grasshopper. It allows for advance modelling and optimisation of the structural system. Developed by Clemens Preisinger in cooperation with the company Bollinger & Grohman (Preisinger, 2013).*

4.5.1 *Import data from Excel*

The industrial warehouses are generated in multiple scripts. To allow for easy modification of parameters across all the scripts, a central Excel sheet is used. Parameters that are supplied by the sheet are related to the naming of the elements and the structural parameters as load values, coefficients and deflection limits. The plugin Lunchbox (Provingground, 2017) provides the necessary tools to read the Excel file.

4.5.2 *Elements*

The structural elements provide the core of the structural analysis in Karamba. The component 'Line-to-Beam' is used to create beam ele-

PARAMETERS	COLUMN	BEAM	PURLIN	BRACING
Material	S235	S235	S235	S235
Section family	HEA	IPE	RHSC	CHSH
Connectivity	hinged	fixed	hinged	fixed
Bending stiffness	yes	yes	yes	no

Table 4.3: Elements in steel beam design with selected material, section, connectivity and bending stiffness assessment

ments of the columns, beams, purlins and braces. Generally speaking, this component offers the ability to create a beam element from given line input with an associate ID, colour and cross section. Additional manipulation of the element allows to neglect bending stiffness. The selection of section families is supported by data provided by the companies Voortman Steelgroup, GB Steelgroup, and ASK Romein (Appendix E).

A summary of the elements and their properties are provided in Table 4.3. Connectivity between the columns and beams (Figure 4.8) is taken as hinged (free rotation about y axis). Similarly with the purlin and beam connection (free rotation about y and z-axis). This underestimating the actual stiffness as some stiffness is present in the connections. The bending stiffness of the bracing is not taken into account. For all elements S235 is taken (further elaborated in Appendix C.

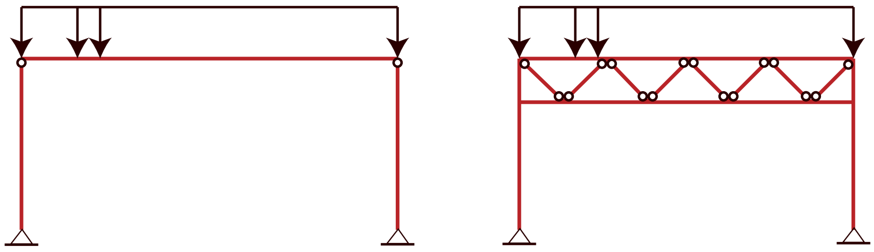


Figure 4.8: Left, a portal frame consisting of beam and columns. Connection between elements is hinged, support also hinged. Right, a portal with a truss. Connection of diagonals are hinged to the upper and lower chords of the truss.

4.5.3 *Supports*

The points at ground level of the columns are selected as support points. A pinned connection is applied (translation in x, y and z is prohibited with no rotational stiffness). This is on the save side as typical beam-column connections do have a certain degree of rotational stiffness.

4.5.4 *Loads*

Applied forces on the structural elements can be separated in three categories: permanent load, snow load and wind load. Within the permanent load, self-weight is not taken into account as a pre-bend is assumed to remove this deformation (Section C.5). The component 'Loads' in Karamba provides a range of loading types in the drop down menu. In this script the 'MeshLoad constant' is applied. Values are extracted from Section C.3.

The permanent roof load is applied with a constant mesh load on the roof beams. This component allows for a surface load to generate line and point loads on desired elements. The 'Mesh Brep' components triangulates the surface according to a set resolution. The mesh loading component works as described by the Karamba user manual (Preisinger, 2016). First the resultant load on each face is calculated. Next with help of additional nodes along the beam elements, the face load is distributed to the nearest nodes. Lastly the loads at each node are summed and divided by the beam length. In the designs only line loads are generated (Figure 4.9). This allows for easy validation with hand calculation. The snow load is generated in a similar fashion.

Wind loading is applied on the front façade as pressure and on the back façade as suction (Figure 4.10). At both locations the surface is selected, meshed with 'Brep mesh' and loading applied with mesh loading. The wind loading on the roof is uniformly taken over the surface (see Section C.3 on wind load). The wind pressure values are dynamical adjusted according to the height input. Linear interpolation is done between 3 and 15 meters. Only one direction is assessed to further reduce the complexity. Analysis of both directions are made and no significant difference was found (see Appendix D.1 for the analysis results).

A combination of permanent and snow load is made, as well as a combination of permanent and wind load. Both are calculated in SLS. Simple hand calculations proof that deformation is governing and that these load combinations are dominant (see Section C.5 for results).

4.5.5 *Cross section optimiser*

The selection of appropriate cross sections is automatically done with the component 'optimise cross-section'. It takes into account the limit

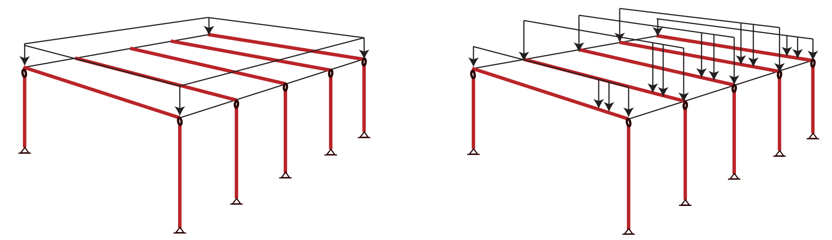


Figure 4.9: Roof loading. Left the total applied load on the roof. Right the translation to the calculation model on the beams.

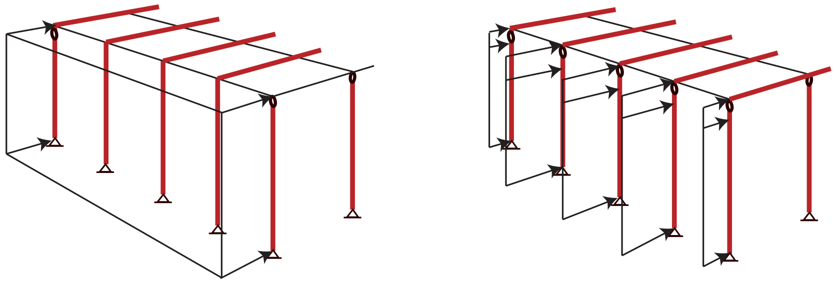


Figure 4.10: Façade loading. Left the total applied load on the façade. Right the translation to the calculation model on the façade columns.

stresses and deformation (Preisinger, 2016). The procedure is as follows: first cross sections are selected that stresses are below yield stress. This is done by analysing at three points in each element, the sectional forces and selecting the first sufficient cross section. This iterates over each elements and checks at each step if all the sections are still sufficient. When a limit deformation is provided, the component temporally lowers yield stress and iteratively selects appropriate cross sections until set limitation is reached. This procedure does not guarantee a solution.

The section optimiser also checks the cross section according to the Eurocode 1993-1-1 (NEN-EN 1993-1-1, 2006) for steel structures. Local buckling and lateral torsional buckling are checked according to the length and connectivity of the beam element. A realistic buckling length is approximated to search for nodes which connect to more than two elements. The procedure takes into account normal forces, bending, shear and torsion. Important to remember is that *global buckling* of the structure is not taken into account and needs to be checked manually. The following input parameters for the 'optimise cross section' component are defined:

- *GroupIDs:* The element groups column, beam and purlin are provided as input. This results in the same cross section for each element with the same ID. It allows to simplify the design and create more realistic warehouses. The braces are disregarded. It proved that unrealistic results were obtained when the braces are also given as input.
- *Cross sections:* All section families (IPE, HEA, CHSH) used in the elements are provided as input.
- *MaxUtil:* The maximum utilisation of the element is set to 1.0. As the structure is assessed in SLS the utilisation can be fully used.
- *MaxDisp:* The maximum displacement is provided as the maximum length of the span divided by 250 (see Appendix C). However, this will result in high local deflections at the columns due to wind loading. Further strengthening measures are needed if the structure is actually build. In the model this is not further detailed and accepted as imperfection. This corresponds with a proof-of-concept implementation.
- *Additional parameters:* Other input parameters are left untouched. In version 1.3 this includes ULSIter as 5, DispIter as 5, nSamples as 3, Elast set as True, and both gammaM0 and gammaM1 as 1.0.

4.5.6 *Mid column scripts*

The mid columns are created at each portal frame corresponding with the parameter max-span-beam. These run from ground level to the supporting beam. The connection is hinged. The pitch roof variant functions similarly as flat roof versions.

The additional floor in one of the mid column scripts, is generated at half the column height. The floor beams run from the 1st portal frame to the 2nd portal frame. A static section is taken as no large differences in span are expected. An additional column is supporting each floor beam end. The connecting is hinged.

The additional warehouse is generated to the left of the original warehouse. It shares a column row and is hinged connected to them. Loads are placed as with a normal warehouse, meaning façade load on the columns and permanent loads on the beams. These beams are added to the optimisation algorithm.

4.5.7 *Truss scripts*

The script with trusses is similar to the beam script. The beams are replaced with trusses in which the upper and lower chords are continues and the diagonals are hinged connected in between. Furthermore, the diagonals and chords are optimised separately on cross section. The height of the warehouse is separated in free height (height - height truss) and the height of the roof. Columns run from ground level to roof level. Connection with the truss lower chord is hinged.

4.6 post processing

The post processing phase extracts the necessary values for both the LCA and storage in database for providing feedback. Goal is an easy to use implementation that can be imported in multiple Grasshopper scripts (Figure 4.11).

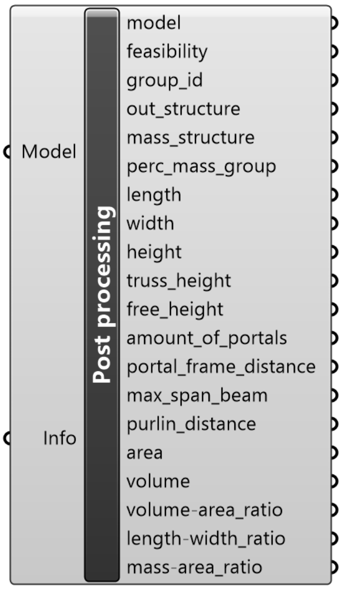


Figure 4.11: Grasshopper cluster that performs the post processing of the model data

The LCA procedure calculates the environmental impact of each element and element group. It gives insight in the different parts of the structural system. In this assessment the mass and material type of each element needs to be extracted.

Feedback requires the info of industrial warehouses to be transferred to a database. Parameters are selected taking into account the design process and engineering judgement. The user changeable variables

NAME	UNIT	COMMENTS
ID	string	Naming elements
Mass	kg	Mass elements
Material	string	Material elements
Count	int	Amount of each element
Section	string	Selected section elements
Feasibility	string	Convergence of Karamba
Area	m^2	Floor area
Volume	m^3	Volume of warehouse
Length	m	Length warehouse
Width	m	Width warehouse
Height	m	Height warehouse
Free height	float	Façade height minus truss height
Portal frame distance	m	Distance between portal frames
Purlin distance	m	Distance between purlins
Amount of portals	int	Amount of portal frames
Truss height	m	Height of truss used
Max span beam	m	Max span if mid columns are used
Mass structure	kg	Total mass of structure
Mass-area ratio	float	Total mass divided by area
Length-width ratio	float	Length divided by width
Volume-area ratio	float	Volume divided by area
Perc. mass group	float	Percentage of mass for each element group

Table 4.4: The extracted performance parameters

are extracted as well as the area and volume of the warehouse. For each ID group the name, section, count, material and mass is selected. Table 4.4 provides an overview on all parameters with respective unit and explanation.

The height and pitch of the roof are taken together in the parameter 'height'. The additional warehouse in one of the scripts is deconstructed in components and added to the the beams, purlins, columns and braces of the main warehouse. The width of such a warehouse is related to the width of the main hall and the span of the additional warehouse. The total area is related to the main warehouse and the additional warehouse combined.

For the script with a mid floor, the components are again added to the respective elements of the main warehouse. The location of the floor is not stored in the database. The additional area that is created with this floor, is added to the floor area of the ground floor.

4.7 ENVIRONMENTAL IMPACT CALCULATION

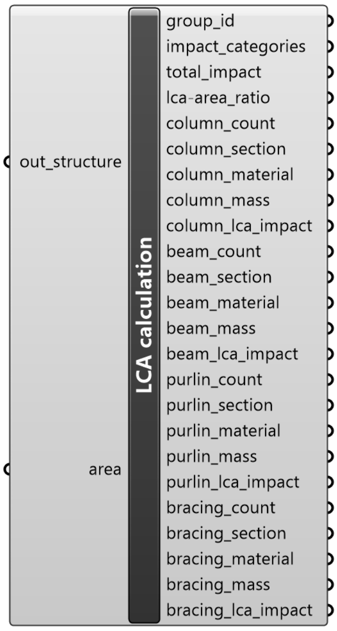


Figure 4.12: Grasshopper cluster that performs the LCA calculation

The LCA cluster (Figure 4.12) calculates the sustainability in both shadow price and 10 impact categories for each structural element. The background of this implementation is provided in Appendix B. The sustainability data of materials originate from the NMD of which version 2.1 (November 2017) is provided by the supervisors of this research project. For the steel elements a NMD average is used. This gives the most objective overview, free from any local high efficiencies in production processes. The calculation is performed inside Grasshopper and requires no data transfer.

The sustainability data from the NMD on steel is read from Excel and put in Grasshopper. The output is structured and the required materials are selected. Following calculation is performed:

$$impact = mass_{elements} * NMD_{data} \tag{4.1}$$

$$shadowprice = \frac{\sum impact * factors}{area * servicelife} \tag{4.2}$$

Here the shadowprice is calculated in €/m^2/year with a service life of 50 years which is according to the codes (Stichting Bouwkwaliteit, 2017).

Additional outputs allow for more insight in the results. All parameters are given in Table 4.5.

NAME	UNIT	EXPLAIN
ID	string	Naming elements
Mass	kg	Mass elements
Material	string	Material elements
Count	int	Amount of each element
Section	string	Selected section elements
LCA	€	LCA impact for each element
Impact categories	string	Assessed categories sustainability
Total impact	€	Total sustainability impact structure
LCA-area ratio	€/m^2/year	Impact over area ratio

Table 4.5: The calculated parameters concerning the sustainability

4.8 VALIDATION

The quality and accuracy is important for the feedback quality. Validation of results are assessed by looking at the calculations from Appendix C. Furthermore, data is compared with available IFC files described in Appendix E. The sustainability is checked against results from a simple LCA program.

4.8.1 *Industrial warehouse data*

The hand calculations and the design booklet by 'Bouwen met Staal' (Bouwen met Staal, 2013) provide insight in accuracy of generated data (Appendix C.5). The same design parameters are used in each validation step: l = 28m, b = 15m, h = 8m, portal frame distance = 7m. For the purlin distance 5 meter is taken. An overview is given in Table 4.6. It can be concluded that the Karamba results are identical to the hand calculations, and comparable to the design booklet. This strengthens the confidence in the generated data.

PARAMETERS	KARAMBA	HAND CALCULATION	BMS BOOKLET
Column section	HEA180	HEA180	HEA200
Beam section	IPE500	IPE500	IPE400
Deformation roof	47 mm	<60 mm	N/A
Deformation façade	53 mm	<53 mm	N/A

Table 4.6: Comparing the results of the Karamba calculation with the hand calculation and the 'Bouwen met Staal' design booklet

The design booklet also provides a rough estimation of total steel usage per element group. An example warehouse of 15.6x31.2 meter is provided. This is very comparable to the design example used in the calculations. Table 4.7 provides the detailed information. It can be concluded that the warehouse generated by Karamba is very similar to data in the booklet. The additional façade steel is not modelled and causes most of the differences. However, obtained results reinforces the quality of the generated data.

Additional validation of Karamba calculations is required to increase output confidence. SCIA Engineer (Nemetschek Group, 2018) is used as calculation software. The 3D model is exported as IFC with GeometryGym plugin 'KarambaExport' (Mirtschin, 2017). Supports and

PARAMETERS	KARAMBA (KG)	BMS BOOKLET (KG)
Columns	3900	4000
Beams	6900	6000
Purlins	900	1000
Braces	1300	650
Additional façade steel	0	1300

Table 4.7: Mass for each element group. It compares the outcome of the Karamba calculation with the data provided in the 'Bouwen met Staal' design booklet

loads are manually added in SCIA. Appendix D.3 provides a detailed overview of the model and results. To summarise, the SCIA calculation, in similar conditions (loads, geometry, supports and other boundary conditions), provides the same results as seen in Karamba. Both deformations and stresses are identical.

Industrial warehouses in IFC format are acquired from the companies Voortman Steelgroup, GB Steelgroup, and ASK Romein. These are used to validate the quality of the generated warehouses. Appendix E contains a thorough overview of all findings. Out of 25 designs a total of 6 designs where found to be comparable enough to validate the script with. This shows the wide variety in final designs compared to the conceptual design stage implementation. Comparing IFC data with generated data in section profiles, result in very comparable to identical sections for beams and columns. However, with increased width and height, the limitations of the parametric script become apparent. The sections found in the generated data are significantly bigger than IFC data shows. This is due to increased wind load and the insufficient modelling of the stability system.

4.8.2 *Sustainability results*

The generated industrial warehouses are assessed on their environmental impact (Section 4.7). This results in a score of €/m^2/year for each design. However, a limited scope is used in terms of elements and system boundary. The question remains to what extend the generated warehouses are comparable to actual build warehouses in structural mass and sustainability impact.

During this research project a number of sources and companies are contacted to acquire sustainability data of industrial warehouses. Via 'Bouwen met Staal', IFC data is acquired however no sustainability scoring. The Dutch Green Building Counsel (DGBC) is contacted but they revealed that all their certificate ratings and data are classified.

Another contact from TNO enclosed that the available data didn't include the necessary parameters for the validation. Only total floor area and material impact, in €/m^2/year, are recorded and are mandatory for the MPG calculation in the BREEAM-NL rating. This information, even if it is accessible, is not detailed enough to provide any validation of calculated LCA results. A last attempt is made by contacting Nibe, a certified LCA and sustainability consultant company. Again they couldn't provide an applicable dataset.

For a simple warehouse (Figure 4.13), a comparison between the LCA calculation performed in Grasshopper and a calculation with the software MPGcalc (DGMR, 2018) is made. The latter is a software based on building elements (from the NMD) and has a cradle to grave system boundary. It therefore includes recycling and demolition of the building product (see Appendix A.2.6). The relevant input parameters and LCA-area ratio scores are given in Table 4.8. It can be conclude that there is a factor 7 difference in the calculation made with Grasshopper and the software MPGcalc. Including the recycling and demolition of steel has a huge impact on the final LCA score. Generated sustainability data in Grasshopper is therefore not representative for actual qualitative answers. However, this is expected due to the applied cradle-to-gate system boundary.

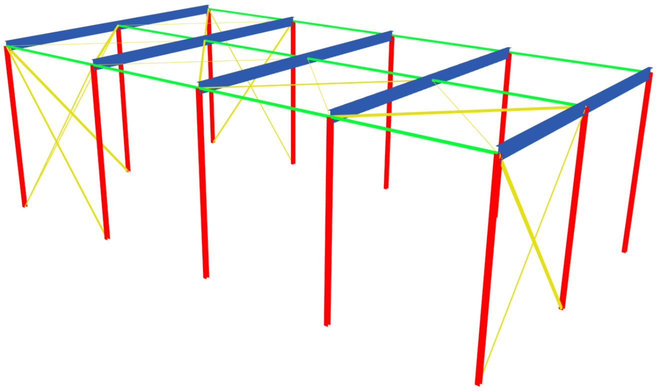


Figure 4.13: Simple industrial warehouse serving as example for validation

PARAMETERS	VALUES
length (m)	28
width (m)	15
height (m)	8
total column length (m)	112
total brace length (m)	267
total beam length (m)	90
total purlin length (m)	90
column section	HEA180
brace section	CHSH114
beam section	IPE500
purlin section	RCSH80
LCA-area ratio GH (€/m2/year)	0.14
LCA-area ratio MPGcalc (€/m2/year)	0.02

Table 4.8: The parameters and values that serve as input for Grasshopper and the software MPGcalc. The final LCA-area ratio for both calculations is presented

5 STORAGE

The generated data on industrial warehouses is stored in a database. Main function is to access the information at a later time. This chapter describes the type of database, the interaction with the data, the shape and characteristics of the dataset, and finally the ways to access and filter the information. In this research project the Python programming language (Python software foundation, 2018) is extensively used. Reasons for it are numerous including the wide adoption, its open source nature, and its extensive documentation.

5.1 PACKHUNT.IO RESOURCE

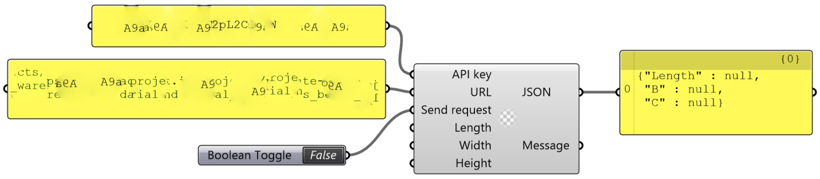
In collaboration with White Lioness technologies, a database is set-up within their Packhunt.io environment, called 'Resources'. This storage model is build on top of the REpresentational State Transfer (REST) framework, commonly used in web services, and popular due to its simplicity and use of well known architecture elements. It uses a key-value database structure to store data. This is a highly flexible format compared to standard database types (Grolinger et al., 2013). Communication between database and user is managed by the Application Program Interface (API) of Packhunt.io in the form of API keys.

> *A Resource is a web-hosted database on the Packhunt.io platform of White Lioness technologies.*

With help of White Lioness technologies, a Grasshopper plugin is developed to transfer generated data in Grasshopper to web-hosted database. Figure 5.1 gives a typical implementation of the plugin within the parametric script. The API key and URL specifies the connection and location of the database. The 'send request' toggles the writing to the database. Additional inputs can be created within the component.

5.2 DATA TABLES

Data on the generated industrial warehouses is subdivided into user changeable parameters and performance variables of the model (Table



```
{"Length" : null,
 "B" : null,
 "C" : null}
```

Figure 5.1: The developed Grasshopper plugin to connect with the resource database of Packhunt.io. The API key and URL are washed out for privacy reasons.

4.4 in Section 4.6). The database layout is determined beforehand (Table 5.1). The JSON (JavaScript Object Notation) format is used as layout and does not require adherence to a fixed schema (Grolinger et al., 2013). The ’slug’ is the unique identifier of the resource. Within the ’field-schema’ the data tables can be formed. Inputs is needed on ’name’, data ’type’ (float, chars, etc), if ’required’, and ’slug’. A small part of the code is inserted below:

```
{
        "name": "data industrial warehouses_4",
        "slug": "data_industrial_warehouses_4",
        "fields_schema": [
        {
                "name": "feasibility",
                "options": {},
                "type": "chars",
                "required": false,
                "slug": "feasibility"
        },
        {
                "name": "width",
                "options": {},
                "type": "float",
                "required": false,
                "slug": "width"
        },
```

5.3 CURATION

With the data driven approach, the quality of the feedback is directly related to the data quality in the database. It is therefore essential to manage this quality. Data curation can be done on many levels and with different techniques (Appendix A.3). Within the data generation phase it is chosen to loosen the parameter range of the industrial

NAME	TYPE	NAME	TYPE
name	chars	column_material	chars
width	float	column_mass	float
length	float	column_lca_impact	float
height	float	beam_count	float
portal_frame_distance	float	beam_section	chars
purlin_distance	float	beam_material	chars
max_span_beam	float	beam_mass	float
truss_height	float	beam_lca_impact	float
free_height	float	purlin_count	float
amount of portals	float	purlin_section	chars
area	float	purlin_material	chars
volume	float	purlin_mass	float
total_mass	float	purlin_lca_impact	float
length_width_ratio	float	bracing_count	float
volume_area_ratio	float	bracing_section	chars
mass_area_ratio	float	bracing_material	chars
lca_area_ratio	float	bracing_mass	float
column_count	float	bracing_lca_impact	float
column_section	chars		

Table 5.1: The created data tables in the Resource database

warehouses (Section 4.3.3). This results in unrealistic, non-buildable and duplicate solutions in the dataset. Duplicates are easily removed with a function written in Python. Non-buildable warehouses are characterised by the boolean 'False' in the column 'feasibility'. A function is written to remove them.

Unrealistic warehouses are characterised by a certain l/w-ratio, column section and free height. The gathered IFC data (Appendix E) is used as reference material to determine realistic levels. The l/w-ratio is limited between 0.5 and 5. A too large column section indicates structural problems. It means that a more efficient way of load transfer is required. The parameter limit of the Grasshopper script is reached. However, it is structural feasible but not seen as a realistic scenario. Column sections larger than HEA400's are removed from the dataset. Lastly, the free height is the usable height in the warehouse , the height of roof level minus the height of the structural elements. Especially the warehouses with trusses are susceptible to a reduced free height. A free height of less than 3 meter is considered unrealistic.

5.4 INTERACTION

Delivering feedback to the end-user requires a data requests to the database. A first option is to query the database with user defined parameters and values. Current implementation of the resource database is limited in this aspect. No value range can be queried. This means that only specific values can be retrieved. This is not a realistic scenario, especially with parameters as area. Querying at database level is fast and efficient. Only needed data is transferred between application and database. However, this implementation is limited, with big datasets, by the performance of the database and the user's internet connection. In cases when hundreds of people start to use the application, the amount of requests can limit the responsiveness of the feedback.

The second option is the pre-load or cache the entire database to the user's computer. With this implementation, all feedback is generated locally. This has relatively few interactions with the database. The dataset is synced at times to provide up to date information. The biggest limitation is local storage.

6 FEEDBACK

The framework presented in Chapter 3 can be further detailed. Figure 6.1 shows the distinguishable steps in the proposed framework. The process is divided in an application frond-end and a Python back-end. This chapter further elaborates the back-end and continues with a demonstration on a possible implemented front-end.

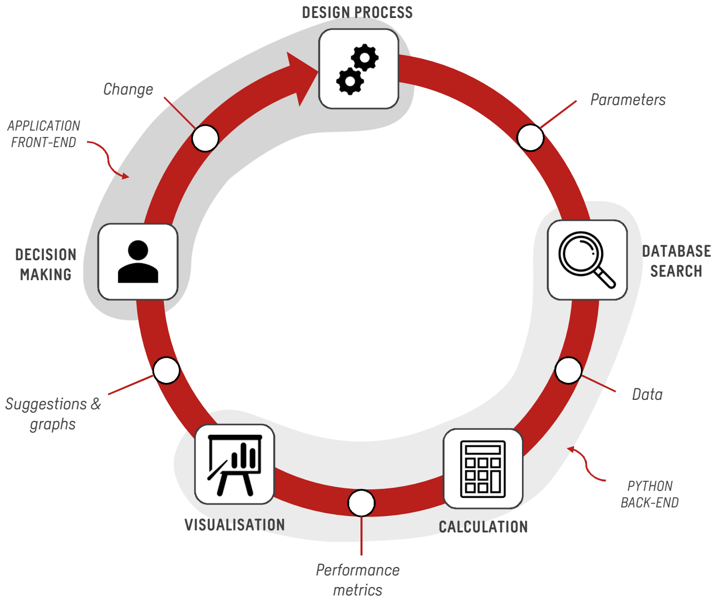


Figure 6.1: Proposed feedback workflow

The parameter input serve as starting point for the back-end processes. Based on the parameter input, a database query is performed. Data that agrees with the parameter values are retrieved. Data selection based on lowering the LCA score is done. Next, performance improvements are calculated for each non user determined parameter. Visualisation produces graphs and suggests the parameter that decreases the LCA score the most. The user can now make decisions.

Important to remember is the conceptual design implementation. As mentioned in Chapter 3, this design phase is characterised by rapid changes, large uncertainties and lack of information.

6.1 ALGORITHM CONCEPT

The proposed feedback mechanism is based on a maximum performance improvement regarding the LCA score. This method is a combination of elements seen in literature study (Appendix A.5). It results in suggesting parameters and values which offer the most reduction in LCA scoring. A Python implementation is performed with the following steps:

1. Copy data and query database with user input parameters
2. Extract mean LCA and filter on data lower than mean
3. Calculate performance improvements for each parameter by comparing new mean with previous mean
4. Select maximum performance & parameter
5. Visualise parameter and performance
6. Return graphs and parameter suggestions to user

These steps are further detailed in the next sections and presented in Figure 6.2. The Python libraries Numpy, Pandas and Matplotlib (scipy, 2018) are used extensively.

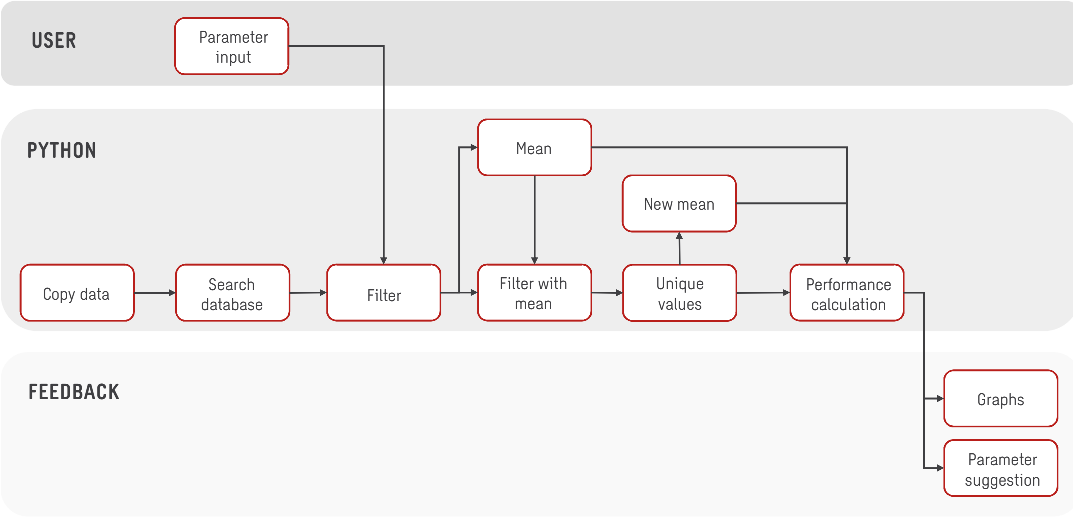


Figure 6.2: Workflow diagram on how the feedback is generated

6.2 DATABASE SEARCH

Searching or querying the database is performed with the user defined parameters and values. Current limitations on database side requires the dataset to be loaded locally (Section 5.4). All data is copied and filtered with user defined parameters. The pseudo code block below details how data is retrieved from the database.

```
input = {parameters, values}

create copy of cached data

for each parameter:
        if value = string:
                search database on: parameter == value
        else:
                lower = 10% below value
                higher = 10% above value
                search database with parameter between: [lower <
                    value < higher]

return filtered data
```

Before filtering a copy of the data is made to preserve the original dataset. Within the for-loop the parameters are each selected and used to filter the copied data. Certain parameters have strings as values (e.g. "name" with span type, or materials). Filtering between strings does not work. Direct search on these parameters is implemented. To cope with sparse data, a higher and lower boundary for the values of 10% is chosen. This is however easily changeable if desired. Filtering is done between the higher and lower bound of the data.

6.3 Calculation

The performance metrics for each parameter is calculated. This determines the best parameter to set next and presents this as suggestion to the user. The pseudo code below shows the procedure of the algorithm.

```
input = filtered data

parameter selection
defined parameters filtered out

orginal mean calculated
data filtered on < mean

for each parameter:
        find unique values
        for each unique value:
                find new_mean
                performance impr. = mean / new_mean

if len(performance_list) <= 8:
        assess each individual value
```

```
else:
        take mean of performance_list

return parameter, performance, mean
```

At the start of the design process, the user defines a set of parameters. Next, data is filtered and selection is made of the parameters which are offered as suggestion. This limits the amount of possibilities and only displays the user desired parameters. Already defined parameters are filtered out of this list as they don't need to be suggested. The parameters that the algorithm provides as suggestion are: length, width, free height, purlin distance, portal frame distance, maximum span beam, and name (span type). But this is all changeable by the user if desired. The performance is calculated with the mean LCA reduction of each data point, compared to the original mean LCA. To steer the design towards reducing the LCA, data points lower than the original mean are selected. For each parameter the unique values in the list are found. This groups the values in one parameter value with one LCA score. The result is easier to interpret graphs. Next, the maximum performance improvement is determined across all parameters. For lists smaller then 8 values, each individual LCA score is assessed. This reduces the risks of missing out on interesting outliers. For longer lists an average is calculated. Selection is made of the best performing parameter.

6.4 VISUALISATION

The visualisation of the results is made in graphs and bar plots. Below the code block is presented. An example of a result is displayed in Figure 6.3 and in Figure 6.4.

```
input = parameter, performance, mean

matplotlib scatter plot (performance, parameter, mean)
seaborn barplot (parameters, mean)

show plot

return graph
```

A positive performance improvement is chosen. The data points are coloured according to their LCA score. This makes it easy to understand and interpret. Also, thin black lines are implemented around the circles to improve visibility. Suggestion in text is of the following: *"Maximum performance improvement can be reached with parameter 'portal frame distance' with an average of 28%"*. This steers the user in a clear direction.

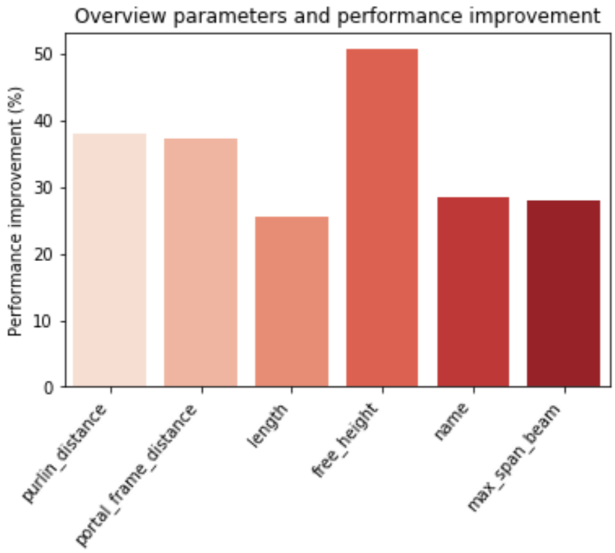


Figure 6.3: Bar plot of all parameters and their performance improvements. This gives the user an good overview on which parameters are important.

6.5 ADVANTAGES

The performance based algorithm excels in speed compared to techniques that perform simulations in the background. Querying is limited by database speed and internet connection, both of which are only limited at very large datasets. Given the current dataset size, feedback is generated instantly. Methods with real time simulation and calculation take seconds if not minutes, depending greatly on the topic of simulation. Structural calculations performed with Karamba are relative efficient. Even here, with a large industrial warehouse, at least 5 seconds are needed to complete the calculation.

Current algorithm implementation allows for quick investigation of the solution space. It has flexibility in a fully interactive way. During the design process values can be changed, new parameters generated, and the solution space interactively explored. It steers the design process but still allows the user to influence this. An additional point is the robustness and independents of the algorithm. No matter what the underlying data characteristics are, the algorithm provides feedback. The strength of relying on actual building data increases confidence. Methods based on ML always contain prediction errors.

6.6 LIMITATIONS

The current feedback algorithm does have limitations. Firstly, if the dataset is sparse and limited in parameter combinations, the feedback

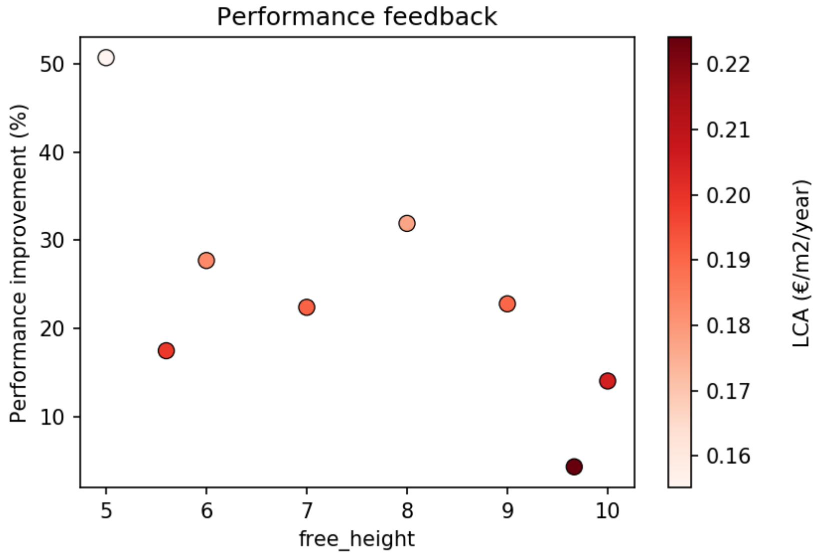


Figure 6.4: Example of graph feedback to the user. In this case a free height is suggested as best parameter to set next. The best performance improvement can be reached by choosing 5 meter. On the right additional information of the LCA value in €/m^2/year is shown.

will be limited. The suggestion algorithm can only actively steer on the data available in the database. This means that no 'new' designs can be suggested. In contrary to prediction algorithms which can provide answers with each parameter combination.

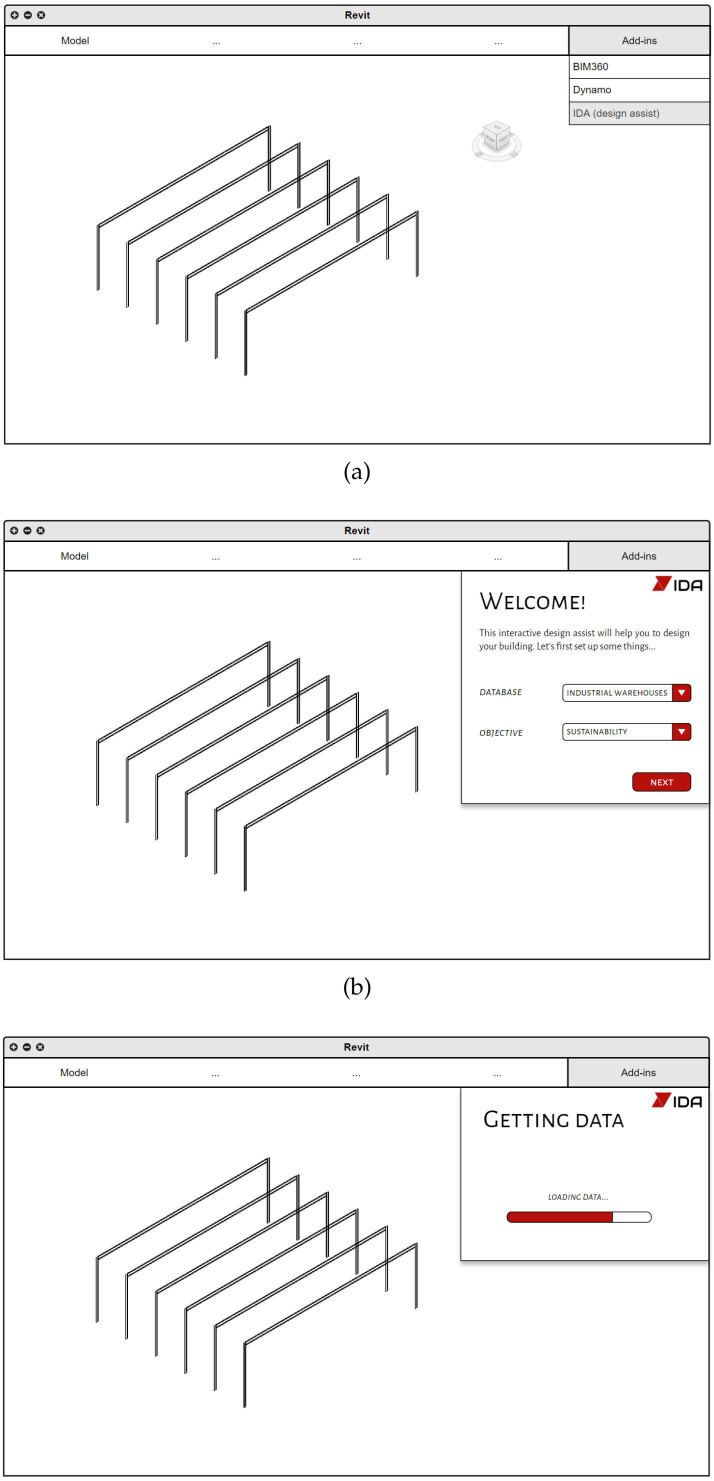
Secondly, current implementation loads the database locally. This is high in performance but can be a bottleneck with large datasets. This is however quickly resolved with additional development on the database side.

The algorithm takes the mean value of certain parameter performance improvements to simplify the output. Possible interesting outliers are lost in this process. This is the consequence of simplifying the output.

6.7 DEMONSTRATION

The feedback is used in the design process of the industrial warehouse. Some mock-ups of a possible integration as an extension or application are made. This shows a possible workflow from start to feedback generation and implementation. Important to note, the mock ups are only sketches and are not actually coded. What however is worked out, is the core feedback algorithm with visualisation of data. To keep in mind is that the demonstration is just one workflow. The algorithm is flexible in providing feedback for many possible workflows.

6.7.1 *Complete workflow*



(a)

Welcome!

This interactive design assist will help you to design your building. Let's first set up some things...

DATABASE: INDUSTRIAL WAREHOUSES

OBJECTIVE: SUSTAINABILITY

(b)

Getting data

LOADING DATA...

(c)

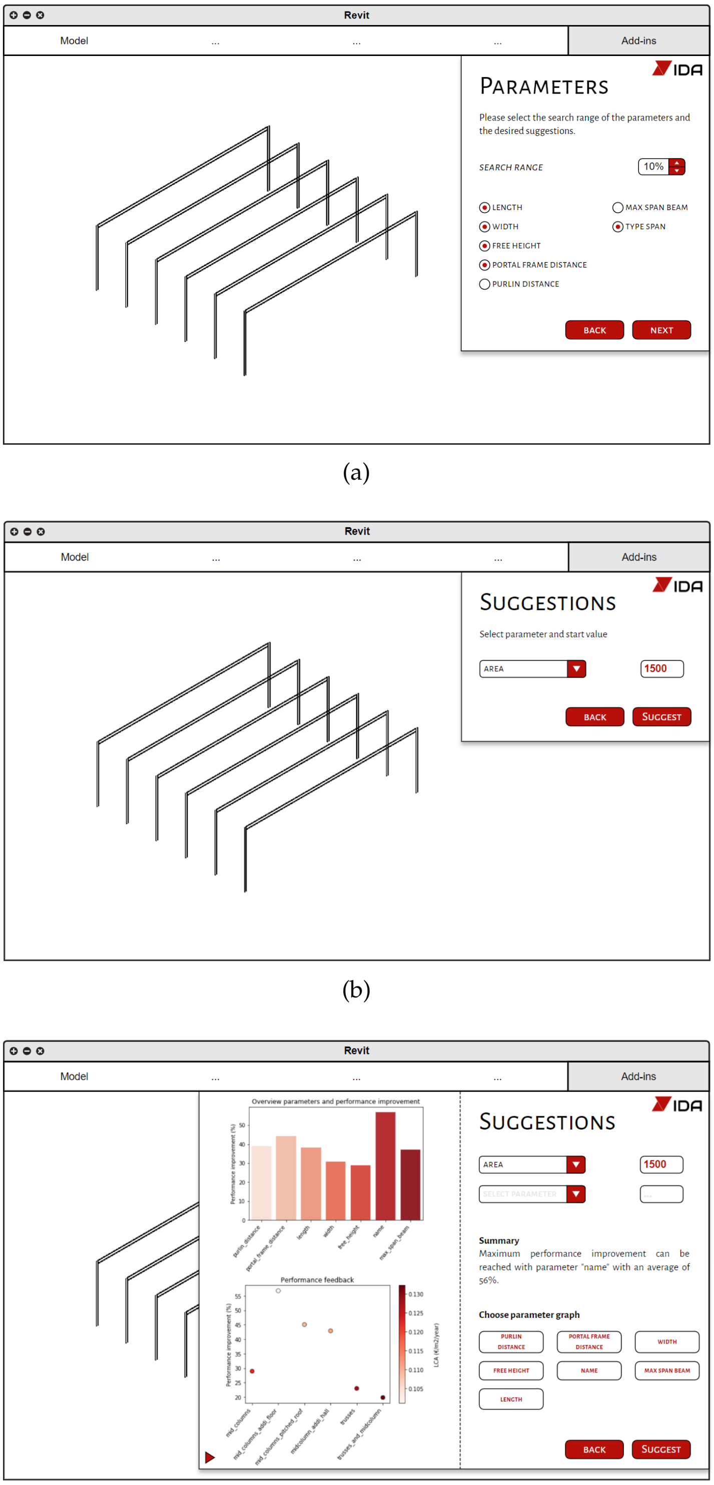
Figure 6.5: IDA is integrated in the application Revit. Selection of database and objective is performed and data is loaded.

First, the application Revit is opened and a start of the structural model is made. The user is interested in what possible parameters to change in order to improve the sustainability of the warehouse. Within the add-ins the Interactive Design Assist (IDA) is installed (Figure 6.5). IDA starts in a window inside Revit. Options appear to select a database and objective for the assistant.

Pressing 'next' loads the selected database and presents the user with the parameters selection page (Figure 6.6 (a)). Here a search range can be selected (how strict the algorithm selects data) and below the parameters available for suggestions are selected. The following screen gives the user the option to choose a parameter and value. Area as starting point is selected of 1500 m^2.

Suggestions are provided in the next screen (Figure 6.6 (c)). A bar plot shows the available parameters and their average performance increase. The scatter plot shows the best parameter with the best performance improvement. The user is able to change the scatter plot data with buttons below. This allows the user to choose his own preferred parameter or let it be guided by the application. Quantitative LCA score results are presented by colours. The user can request detailed information by clicking on the graph (Figure 6.7 (a)). The figure opens in a larger window and reveals additional settings such as showing the LCA values and the pop-up info. Hovering over the data points in the graph reveals extra information.

Next, the user can make its choice on the parameter value. The warehouse type with mid columns and a pitched roof is preferred. IDA provides the next suggestion now with two parameters fixed. This time the length is the most important one (Figure 6.7 (c)).



PARAMETERS

Please select the search range of the parameters and the desired suggestions.

SEARCH RANGE 10%

● LENGTH
● WIDTH
● FREE HEIGHT
● PORTAL FRAME DISTANCE
○ PURLIN DISTANCE
○ MAX SPAN BEAM
● TYPE SPAN

(a)

SUGGESTIONS

Select parameter and start value

AREA 1500

(b)

SUGGESTIONS

AREA 1500

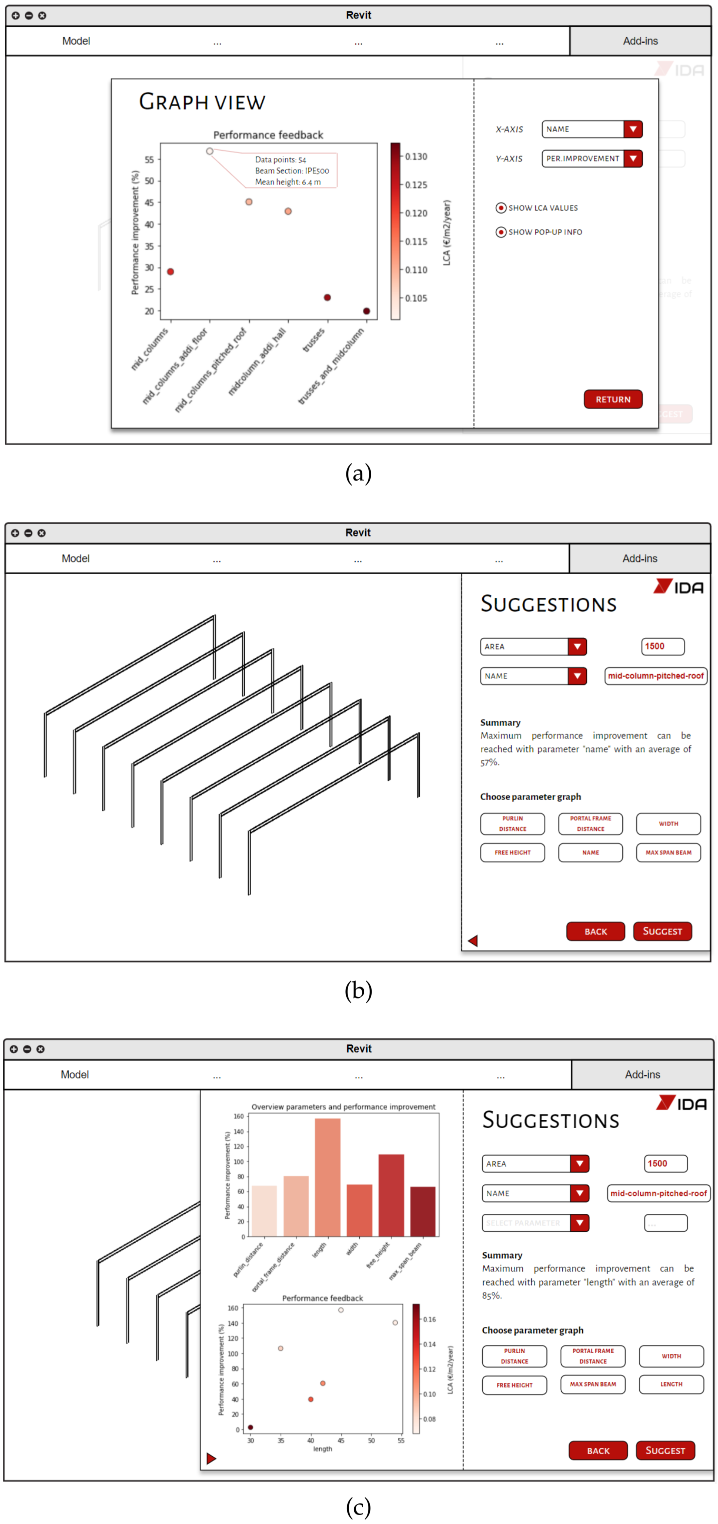
SELECT PARAMETER ...

Summary
Maximum performance improvement can be reached with parameter "name" with an average of 56%.

Choose parameter graph

(c)

Figure 6.6: Within IDA the parameter selection is made and the user provides the given starting point. Feedback is generated and presented to the user



(a)

Summary
Maximum performance improvement can be reached with parameter "name" with an average of 57%.

(b)

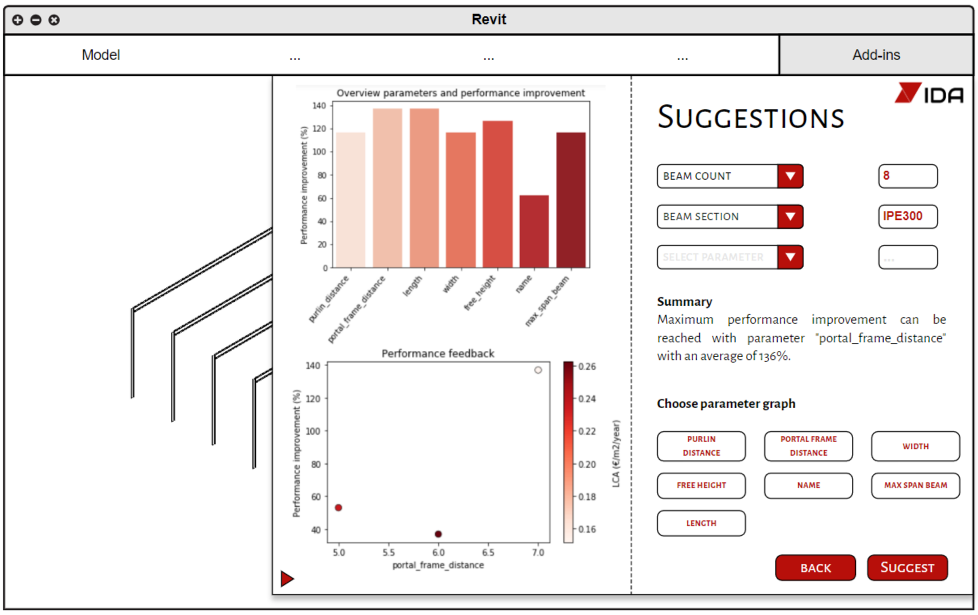
Summary
Maximum performance improvement can be reached with parameter "length" with an average of 85%.

(c)

Figure 6.7: IDA provides more detailed information when necessary. The next parameter is given and suggestion generated.

6.7.2 *Selection beams*

Another workflow is if the user is interested in what is possible with a certain amount of beams. Figure 6.8 provides the feedback with a beam count of 8 and a IPE300 section. The parameter feedback can be explored by the user to see what options are available.



BEAM COUNT: 8

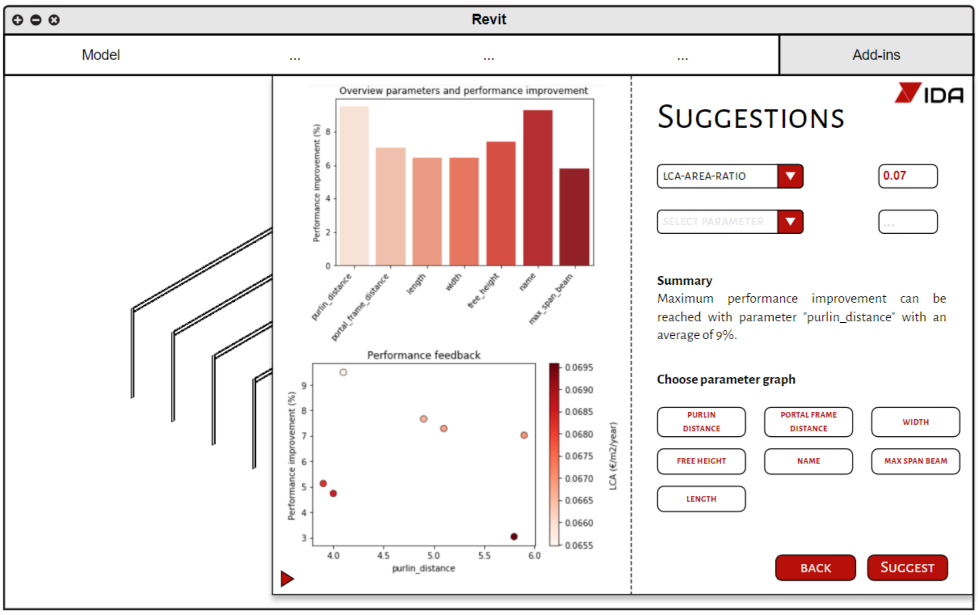
BEAM SECTION: IPE300

Summary
Maximum performance improvement can be reached with parameter "portal_frame_distance" with an average of 136%.

Figure 6.8: Feedback generated with providing beam count and section.

6.7.3 *Lowest LCA score*

A third workflow is based on setting a desired performance level from the start. The LCA score is set to 0.07 €/m^2/year (Figure 6.9). Again, the parameter feedback can be explored to find desired results.



SUGGESTIONS

LCA-AREA-RATIO ▼ 0.07

SELECT PARAMETER ▼ ...

Summary
Maximum performance improvement can be reached with parameter "purlin_distance" with an average of 9%.

Choose parameter graph

Figure 6.9: Feedback generated with providing a desired LCA score.

7 DISCUSSION

In this chapter the obtained results and limitations are discussed. The proposed framework, the use of generated data, and the feedback algorithm are elaborated. Furthermore, the main research question and the vision are addressed.

7.1 OBJECTIVE AND VISION

The aim of this research project is to develop a method to aid the decision making process in the conceptual design on sustainability. A software based prototype (IDA) is developed to present the user with LCA feedback at crucial points in the design phase. Such a solution is fast and capable of finding correlations between parameters and objective. Changes in underlying data (building and LCA) are immediately propagated throughout the application, saving time.

A conceptual design implementation is required. The basic design concept is developed during this phase. Including shape of building, positioning, structural load bearing concepts, placement of windows. These are fundamentally difficult to change in later stages. It emphasises the importance of this design phase and the need to make well balanced decisions. The biggest risk in this stage is the large uncertainty of made assumptions and the lack of information. IDA provides design help in a flexible way. When concepts or assumptions change, new feedback can be generated in a quick and easy way.

The vision consists of a web-based data platform, a cloud hosted database and an intelligent feedback algorithm. At the core this vision is based on data that is provided by companies and thus available for other people to use. This research project shows that information is not richly available and that companies are protective of their intellectual property. These are continuous challenges for the future. However, it is expected that at a certain point companies will acknowledge the enormous benefit of sharing knowledge within the industry. Especially if this could be attached to new business model, reduction of risks, and better performing projects.

7.2 FRAMEWORK

Developed framework is based on a data driven approach with a conceptual design implementation. It steers the design process, provides feedback to the user and is flexible in starting point and parameters. The main goal is to support the decision making process of the end-user at any point. The framework provides the user with the most important parameters and which performance improvements can be achieved. With easy to interpret feedback, users are less likely to oversee the optimal solution. This also increases the confidence in the feedback. Visualisation is one of the key elements in this process and is thoroughly researched. Performance of the feedback is near instant, which supports the exploration of the design space. This directly integrates with the characteristics of the conceptual design phase and the required flexibility.

The feedback is generated based on user input. Simple database queries are performed to retrieve building and LCA data. Filtering is applied to disregard data points below the LCA mean. This means that the feedback always shows an improvement in LCA score compared to current decided parameters. The performance improvement is calculated by comparing the previous mean LCA with the current LCA score. This implementation is limited in finding complex correlations between parameters in the database. However, the simplicity allows for better comprehension of the results, unlike ML prediction algorithms which are often a black-box and contain prediction errors.

However, the implementation does have a risk in becoming a black-box. At this point, the user has no clear understanding of the processes inside the application. It reduces the confidence in the feedback and it negatively effects the decision making process. The user is unsure whether the program is reliable and provides correct feedback. To reduce black-box behaviour, it is essential to provide good documentation and have inside in the code.

The framework's dependency on an underlying database is something to keep in mind. It means that the quality of the feedback is directly correlated with the quality of the building and LCA data. Careful management is needed with strict control on what data is allowed in the dataset. Data curation is therefore an important step in the process.

Important to realise is that this framework is very flexible and can easily be extended with additional parameters, objectives, building types, etc. The framework concept will remain the same.

7.3 generated data vs real-life data

At the first stages of this research project, it is decided to generate the dataset due to lack of real world data. The most important difference between actual data and generated data is the quality. It is expected that real world building data is of low quality, unstructured and that large differences in types and parameters are found. These differences lead to a large 'emptiness' in the high dimensional design space. Finding data points correlated with the user's input is difficult. Therefore it is a challenge to extract feedback. However, in these circumstances with a large enough search range the prototype application is capable of providing feedback.

The generated data does not fully represent the actual build warehouses due to limitations in shape, elements and parameters. This however, makes it easier to store in a database. With more complex building types, the translation from drawing, BIM model or IFC file to a simple (semi-)structured database is complex and challenging. A shape which is non-rectangular is rather hard to capture in a simple set of parameters (e.g. length, width, height). Similarly with a certain selection of materials. An ordinary building has dozen of materials all related to specific elements. When selection is required in the feedback algorithm, a choice has to be made what material is dominate: "is it a steel building or a concrete building?". These are challenges that need to be further researched.

Sustainability of buildings greatly depends on the chosen system boundary. With real data there is a high chance of differences in these boundaries. The data can therefore not be easily compared. Although the LCA procedure is a standardised method, there is plenty of room for interpretation. Additionally, the underlying material and element database is often country specific and results are difficult to compare on a global scale.

7.4 reflection tool requirements

Chapter 3 discusses the requirements and features of the tool. A key aspect is to steer the design process on lowering the LCA score of the warehouse. This objective is achieved by removing data points which are lower than the mean LCA. In this way, the user is only presented with data that improves the LCA score. The presentation of data is visually provided in a scatter plot. It shows all parameter value possibilities with respective performance improvement and quantitative LCA score. This way the user has a good overview of the data. Flexibility is implemented by providing feedback from ever possible starting point, only limited by the data in the database. The performance and interactivity of the application is high which helps the integration in the current conceptual design process.

The current implementation is lacking a user front-end and integration in a BIM application. The feedback is at this moment limited by data in the database. It can only steer on designs which are actually present in the dataset. Furthermore, the current implementation is limited by the query capabilities of the database. For very large datasets memory size could pose a limitation.

8 CONCLUSION & RECOMMENDATIONS

8.1 CONCLUSION

The stated main objective in Chapter 2 is as follows:

"Research and develop a software prototype to aid the decision making process regarding the sustainability of building designs in the conceptual design phase"

From the developed framework and prototype application it can be concluded that a data driven approach based on an underlying database, is the best way of supporting the conceptual design process. The framework steers towards lowering the LCA score with help of building and LCA data. The data is generated with parametric scripts in Grasshopper and focusses on industrial warehouses. Feedback is generated by comparing parameter input with underlying data. Performance improvements are calculated for each parameter and visualised to the user.

Answers on the stated research questions are now further elaborated.

What research is done in providing feedback and aiding the decision making process in early design?

From literature review it can be concluded that optimisation algorithms, predictive algorithms and a data driven approach are used to support the easy design process. Optimisation algorithms present the user with the optimal solution considering the given boundaries. However, this method does not balance the user's needs with the best solution. Current algorithms are often integrated in parametric environments. This requires constant simulation of objectives which significantly slow down the design process. Often predictive algorithms are applied which function as a black-box and require extensive parameter tuning to achieve high performance. A data driven approach retrieves its feedback from already present data.

What software framework needs to be developed to aid the decision making process in the conceptual design phase?

From a combination of literature study and implementation it can be concluded that a framework based on a data driven approach is the optimal solution. 'Knowledge' on industrial warehouses is stored in a database and from this feedback is provided. The framework contains a front-end for parameter input and a back-end, closely linked to the database, the feedback algorithm and the visualisation. The advantage is that this framework is easy to steer in a desired direction. It achieves a method to bring back previously acquired knowledge to the early design phase of a new building.

How can the environmental impact of buildings be calculated?

From literature review on sustainability calculation is can be concluded that a LCA method is commonly performed as a standardised way of calculating the environmental impact across an entire building life. The chosen system boundary significantly influences the results. Documentation of this is essential when comparisons are made.

What feedback algorithm can be implemented to steer the design in reducing the sustainability impact and which also complies with the conceptual design phase characteristics?

From literature review it can be concluded that the conceptual design is characterised by creativity, iterative nature, exploration of the design space, flexible in starting points, large uncertainty and lack of information. Furthermore, it can be concluded that the engineer is interested in which parameters are most important in lowering the LCA score. The Interactive Design Assist (IDA) prototype application supports this process by providing LCA score improvements related to individual parameters from the industrial warehouse. Feedback is visualised in scatter plots with the freedom to select parameter and desired value. IDA steers the design process but keeps the user in control.

8.2 RECOMMENDATIONS

The following recommendations apply to this research project.

- The generated data is a simplified version in terms of shape, elements and materials, of actual built industrial warehouses. The proposed framework and algorithm needs to be tested and applied on real data to show its capabilities. The challenge is to find or create such a dataset of large enough size and quality. Possibilities are to create a system to transfer BIM models and IFC files to a (semi-) structured database. Additionally, the dataset can be enriched with more complicated shapes, materials, different building types and new objectives (e.g. cost).
- The dataset has a high risk of being very sparse in the high dimensional design space. This means that with many parameters and few data, low quality feedback is generated. Techniques need to be explored to increase this quality. Possibilities are in generating correlations between existing data with help of ML

algorithms. It results in receiving feedback on parameter combinations that are not stored in the dataset. However, this algorithm should also support the flexible conceptual design process. Many of today's ML algorithms require a fix parameter set to train on. This directly goes against to philosophy of the conceptual design phase.

- The feedback algorithm can be further improved and extended. Ideas are to give the user control on which set of data to select. Current implementation only takes data which is below the LCA mean. Information as cross sections, beam count, LCA scores for each element group are not used right now but can be implemented as additional feedback. Furthermore, a plot of the environmental scoring during the design process can be made. This gives the user insight into the consequences of his design choices. Possibilities are to use this with predicting the best order of choosing parameters. A pattern recognition or rule learning algorithm can be implemented.

- The proof-of-concept is very limited in implementation. First, a usable front-end for the user could be developed. Second, integration in a BIM or parametric application could be researched. The final goal is to integrate the modelling and design feedback in one coherent package.

- The LCA scores needs to be validated against a real-world dataset to show its limitations. Furthermore, the cradle-to-gate system boundary is very limited looking at the entire life-cycle. Especially if additional materials are added to the database, it becomes important to assess the end-of-life scenarios. Recycling and re-use have large consequence for the final score. Therefore by making a fair comparison a more extensive assessment of the life cycle is needed. A possible solution is to use the NMD more extensively and select building elements as pre-produced beams and columns. These elements include the end-of-life phases mentioned before.

- A topic of research is how to transfer BIM models and IFC files to a (semi-) structured database. This would help in creating the desired database of actually build building.

BIBLIOGRAPHY

Adams, William M et al. (2006). 'The future of sustainability: Rethinking environment and development in the twenty-first century'. In: *Report of the IUCN renowned thinkers meeting*. Vol. 29, p. 31.

Amasyali, Kadir and Nora M. El-Gohary (2018). *A review of data-driven building energy consumption prediction studies*. DOI: 10.1016/j.rser.2017.04.095.

Autodesk Inc. (2017). *Autodesk*. URL: www.autodesk.com (visited on 26/11/2017).

Becker, Tilman and Curry, Edward and Jentzsch, Anja and Palmetshofer, Walter (2016). *New horizons for a data-driven economy: Roadmaps and action plans for technology, businesses, policy, and society*. ISBN: 9783319215693. DOI: 10.1007/978-3-319-21569-3_16.

Bouwen met Staal (2013). *Hallenboek*.

Braendstrup, C (2017). 'Conceptual design of a demountable , reusable composite flooring system'. Graduation Thesis. TU Delft.

Bragança, Luís, Susana M Vieira and Joana B Andrade (2014). 'Early stage design decisions: the way to achieve sustainable buildings at lower costs.' In: *TheScientificWorldJournal* 2014, p. 365364. ISSN: 1537-744X. DOI: 10.1155/2014/365364.

BuildingSMART (2018). *Building Smart open standards*. URL: {www.buildingsmart.org/standards/technical-vision/open-standards}.

Coenders, J.L. (2011). 'NetworkedDesign: next generation infrastructure for computational design'. PhD thesis. TU Delft, pp. 39–46.

Coenders, J.L. (2013). 'Sustainability-open: why every building will be sustainable in the future'. In: *Design Modelling Symposium Berlin 2013*, pp. 1–8.

Curry, Edward (2015). *The Big Data Value Chain: Definitions, Concepts, and Theoretical Approaches*.

Danatzko, Joseph M and Halil Sezen (2011). 'Sustainable Structural Design Methodologies'. In: *Practice Periodical on Structural Design and Construction* 16.NOVEMBER, pp. 186–190. ISSN: 1084-0680. DOI: 10.1061/(ASCE)SC.1943-5576.0000095..

De Wolf, Catherine, Frances Yang, Duncan Cox, Andrea Charlson, Amy Seif Hattan and John Ochsendorf (2016). 'Material quantities and embodied carbon dioxide in structures'. In: *Proceedings of the Institution of Civil Engineers - Engineering Sustainability* 169.4, pp. 150–161. ISSN: 1478-4629.

Delen, Dursun, Glenn Walker and Amit Kadam (2005). 'Predicting breast cancer survivability: A comparison of three data mining methods'. In: *Artificial Intelligence in Medicine* 34.2, pp. 113–127. ISSN: 09333657. DOI: 10.1016/j.artmed.2004.07.002.

Devanathan, Srikanth, Devarajan Ramanujan, William Z. Bernstein, Fu Zhao and Karthik Ramani (2010). 'Integration of Sustainability Into Early Design Through the Function Impact Matrix'. In: *Journal of Mechanical Design* 132.8, p. 081004. ISSN: 10500472. DOI: 10.1115/1.4001890.

DGMR (2018). *MPGCalc software version 1.0*. URL: https://mpgcalc.nl/.

Dixit, Manish Kumar, José L Fernández-solís, Sarel Lavy and Charles H Culp (2010). 'Identification of parameters for embodied energy measurement : A literature review'. In: *Energy and Buildings* 42, pp. 1238–1247. DOI: 10.1016/j.enbuild.2010.02.016.

Eastman, Charles M (2016). 'Cloud-based BIM Data Transmission: Current Status and Challenges'. In: *33rd International Symposium on Automation and Robotics in Construction (ISARC 2016)* September. DOI: 10.13140/RG.2.2.30454.60482.

Eleftheriadis, S., D. Mumovic, P. Greening and A. Chronis (2016). 'BIM enabled optimisation framework for environmentally responsible and structurally efficient design systems'. In: *32nd International Symposium on Automation and Robotics in Construction and Mining: Connected to the Future, Proceedings*.

EU (2010). *DIRECTIVE 2010/31/EU on the energy performance of buildings*. (Visited on 13/10/2017).

Felkner, J, E Chatzi and T. Kotnik (2013). 'Architectural feedback in the structural optimization process'. In: *Structures and Architecture: Concepts, Applications and Challenges*, pp. 676–684.

FigureEight (2018). *Figure Eight*. URL: https://www.figure-eight.com/ (visited on 02/08/2018).

Foraboschi, Paolo, Mattia Mercanzin and Dario Trabucco (2014). 'Sustainable structural design of tall buildings based on embodied energy'. In: *Energy and Buildings*. ISSN: 03787788. DOI: 10.1016/j.enbuild.2013.09.003.

Gkioka, Maria (2018). 'Knowledge Model for the Parametric Design of Concrete Viaducts'. TU Delft.

Grolinger, Katarina, Wilson A Higashino, Abhinav Tiwari and Miriam AM Capretz (2013). 'Data management in cloud environments: NoSQL and NewSQL data stores'. In: *Journal of Cloud Computing: Advances, Systems and Applications* 2.1, p. 22. ISSN: 2192-113X. DOI: 10.1186/2192-113X-2-22.

Hachesu, Peyman Rezaei, Maryam Ahmadi, Somayyeh Alizadeh and Farahnaz Sadoughi (2013). 'Use of data mining techniques to determine and predict length of stay of cardiac patients'. In: *Healthcare Informatics Research* 19.2, pp. 121–129. ISSN: 20933681. DOI: 10.4258/hir.2013.19.2.121.

Haerder, Theo and Andreas Reuter (1983). 'Principles of transaction-oriented database recovery'. In: *ACM Computing Surveys (CSUR)* 15.4, pp. 287–317.

Heidegger, Veronika, Jeroen Coenders and Anke Rolvink (2014). 'EnergyFacade - operational energy optimisation for conceptual facade design'. In: *EnergyFacade - operational energy optimisation for conceptual facade design*. IASS-SLTE 2014 Symposium.

Hollberg, Alexander and Jürgen Ruth (2016). 'LCA in architectural design—a parametric approach'. In: *International Journal of Life Cycle Assessment* 21.7, pp. 943–960. ISSN: 16147502. DOI: 10.1007/s11367-016-1065-1.

IGG (2018). *IGG kostenadviseur*. URL: https://www.igg.nl/ (visited on 30/05/2018).

Jootoo, Achyuthan, S M Asce, David Lattanzi and M Asce (2017). 'Bridge Type Classification: Supervised Learning on a Modified NBI Data Set'. In: 31.6, pp. 1–11. DOI: 10.1061/(ASCE)CP.1943-5487.0000712.

Kaethner, S. C. and J. A. Burridge (2012). 'Embodied CO2 of structural frames'. In: *Structural Engineer* 90.5, pp. 33–40. ISSN: 14665123.

Kellenberger, Daniel and Hans Jörg Althaus (2009). 'Relevance of simplifications in LCA of building components'. In: *Building and Environment* 44.4, pp. 818–825. ISSN: 03601323. DOI: 10.1016/j.buildenv.2008.06.002.

Kim, Joyce, Yuxun Zhou, Stefano Schiavon, Paul Raftery and Gail Brager (2018). 'Personal comfort models: Predicting individuals' thermal preference using occupant heating and cooling behavior and machine learning'. In: *Building and Environment* 129.October 2017, pp. 96–106. ISSN: 03601323. DOI: 10.1016/j.buildenv.2017.12.011. URL: https://doi.org/10.1016/j.buildenv.2017.12.011.

Kim, Kyong Ju, Won Gun Yun, Namho Cho and Jikwang Ha (2017). 'Life cycle assessment based environmental impact estimation model for pre-stressed concrete beam bridge in the early design phase'. In: *Environmental Impact Assessment Review* 64, pp. 47–56. ISSN: 01959255.

Kingspan (2018). *X-DEK large span composite roof panel*. URL: https://www.kingspan.com (visited on 25/04/2018).

Kourou, Konstantina, Themis P. Exarchos, Konstantinos P. Exarchos, Michalis V. Karamouzis and Dimitrios I. Fotiadis (2015). 'Machine learning applications in cancer prognosis and prediction'. In: *Computational and Structural Biotechnology Journal* 13, pp. 8–17. ISSN: 20010370. DOI: 10.1016/j.csbj.2014.11.005. arXiv: 9781591404590. URL: http://dx.doi.org/10.1016/j.csbj.2014.11.005.

Kurilla, Lukáš, Henri Achten and Miloš Florián (2013). 'Scripting Design Supported by Feedback Loop from Structural Analysis'. In: *Computation and Performance* September, pp. 51–60.

Laney, Doug (2001). '3D data management: Controlling data volume, velocity and variety'. In: *META Group Research Note* 6.70.

Lankhorst, G.J. (2018). 'Sustainable Structural Design, Life-cycle assessment of main load bearing structures of high-rise buildings in the Netherlands'. TU Delft.

Lee, Ghang, Jiyong Jeong, Jongsung Won, Chiyon Cho, Seok-joon You, Sungil Ham and Hoonsig Kang (2014). 'Query Performance of the IFC Model Server Using an Object-Relational Database Approach and a Traditional Relational Database Approach'. In: *Journal of Computing in Civil Engineering* 28.2, pp. 210–222. ISSN: 0887-3801. DOI: 10.1061/(ASCE)CP.1943-5487.0000256.

Li, Hang, Hua Liu, Yong Liu and Yuan Wang (2016). 'An object-relational IFC storage model based on oracle database'. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* 41.July, pp. 625–631. DOI: 10.5194/isprsarchives-XLI-B2-625-2016.

Liu, Yuezhong, YiChun Huang, Singapore Rudi Stouffs and Dr Rudi Stouffs (2015). 'Using a Data-Driven Approach To Support the Design of Energy-Efficient Buildings'. In: *Journal of Information Technology in Construction* 20.October 2014, pp. 80–96. ISSN: 14006529. URL: http://www.itcon.org/2015/6.

Lolli, Nicola, Selamawit Mamo Fufa and Marianne Inman (2017). 'A Parametric Tool for the Assessment of Operational Energy Use, Embodied Energy and Embodied Material Emissions in Building'. In: *Energy Procedia* 111.1876, pp. 21–30. ISSN: 18766102. DOI: 10.1016/j.egypro.2017.03.004.

MacMillan, Sebastian, John Steele, Simon Austin, Paul Kirby and Robin Spence (2001). 'Development and verification of a generic framework for conceptual design'. In: *Design Studies* 22.2, pp. 169–191. ISSN: 0142694X. DOI: 10.1016/S0142-694X(00)00025-9.

Mangalathu, Sujith and Jong-su Jeon (2018). 'Classification of failure mode and prediction of shear strength for reinforced concrete beam-column joints using machine learning techniques'. In: *Engineering Structures* 160.January, pp. 85–94. ISSN: 01410296. DOI: 10.1016/j.engstruct.2018.01.008. URL: http://linkinghub.elsevier.com/retrieve/pii/S0141029617329401.

Marjaba, G. E. and S. E. Chidiac (2016). 'Sustainability and resiliency metrics for buildings - Critical review'. In: *Building and Environment* 101, pp. 116–125. ISSN: 03601323. DOI: 10.1016/j.buildenv.2016.03.002. URL: http://dx.doi.org/10.1016/j.buildenv.2016.03.002.

Markelj, Jernej, Manja Kitek Kuzman, Petra Grošelj and Martina Zbašnik-Senegačnik (2014). 'A simplified method for evaluating building sustainability in the early design phase for architects'. In: *Sustainability (Switzerland)* 6.12, pp. 8775–8795. ISSN: 20711050. DOI: 10.3390/su6128775.

Microsoft Azure (2017). *Algorithm choice.* (Visited on 04/03/2018).

Mirtschin, J (2017). *GeometryGym*. URL: http://geometrygym.com/ (visited on 26/11/2017).

MIT (2018). *Digital Structures GH plugin, version 1.3*. URL: http://digitalstructures.mit.edu/ (visited on 14/06/2018).

Nace, Trevor (2017). *Islands are swalloded by sea level rises*. URL: https://goo.gl/E9XzP4 (visited on 01/02/2018).

Nemetschek Group (2018). *SCIA Engineer, version 17.01*. URL: https://www.scia.net/en (visited on 09/02/2018).

NEN-EN 15804 (2013). *Sustainability of construction works - Environmental product declarations - Core rules for the product category of construction products*. Tech. rep. Delft: Nederlands Normalisatie-instituut.

NEN-EN 1990 NB (2011). *Eurocode: Basis of structural design*. Standard. Delft: Nederlands Normalisatie-instituut.

NEN-EN 1991-1-1 NB (2011). *Eurocode 1: Actions on structures - Part 1-1: General actions - Densities, self-weight, imposed loads for buildings*. Standard. Delft: Nederlands Normalisatie-instituut.

NEN-EN 1991-1-3 NB (2011). *Eurocode 1: Actions on structures - Part 1-3: General actions - Snow loads*. Standard. Delft: Nederlands Normalisatie-instituut.

NEN-EN 1991-1-4 NB (2011). *Eurocode 1: Actions on structures - Part 1-4: General actions - Wind actions*. Standard. Delft: Nederlands Normalisatie-instituut.

NEN-EN 1991-1-4 (2011). *Eurocode 1: Actions on structures - Part 1-4: General actions - Wind actions*. Tech. rep.

NEN-EN 1993-1-1 (2006). *Eurocode 3: Design of steel structures - Part 1-1: General rules and rules for buildings*. Tech. rep.

NEN-EN-iso 14044 (2006). *Environmental management - Life cycle assessment - Requirements and guidelines*. Tech. rep. Delft: Nederlands Normalisatie-instituut.

NMD (2017). *Nationale milieudatabase*. URL: https://www.milieudatabase.nl (visited on 06/11/2017).

O'Donnell (2018). *Metal roof deck*. URL: www.odonnellmetaldeck.com/c/roof-deck/ (visited on 08/05/2018).

Olsson, A., G. Sandberg and O. Dahlblom (2003). 'On Latin hypercube sampling for structural reliability analysis'. In: *Structural Safety* 25.1, pp. 47–68. ISSN: 0167-4730. DOI: 10.1016/S0167-4730(02)00039-5. URL: https://www.sciencedirect.com/science/article/pii/S0167473002000395.

Østergård, Torben, Rasmus L. Jensen and Steffen E. Maagaard (2016). 'Building simulations supporting decision making in early design - A review'. In: *Renewable and Sustainable Energy Reviews* 61, pp. 187–

201. ISSN: 18790690. DOI: 10.1016/j.rser.2016.03.045. URL: http://dx.doi.org/10.1016/j.rser.2016.03.045.

Oti, Akponanabofa Henry and Walid Tizani (2015). 'BIM extension for the sustainability appraisal of conceptual steel design'. In: *Advanced Engineering Informatics* 29.1, pp. 28–46. ISSN: 14740346.

Preisinger, Clemens (2013). 'Karamba, Linking structure and parametric geometry, version 1.3'. In: *Architectural Design* 83.2, pp. 110–113.

Preisinger, Clemens (2016). *Karamba User Manual*. Tech. rep., pp. 1–104.

Provingground (2017). *Lunchbox*. URL: https://provingground.io/ (visited on 29/11/2017).

Python software foundation (2018). *Python software foundation, version 3.6*. URL: https://www.python.org/.

Rai, Deepak, Behzad Sodagar, Rosi Fieldson and Xiao Hu (2011). 'Assessment of CO2 emissions reduction in a distribution warehouse'. In: *Energy* 36.4, pp. 2271–2277. ISSN: 03605442. DOI: 10.1016/j.energy.2010.05.006. URL: http://dx.doi.org/10.1016/j.energy.2010.05.006.

Riehle, Dirk and Thomas R. Gross (1998). *Role Model Based Framework Design and Integration*.

Rijksoverheid (2018). *Rijksoverheid stimuleert energiebesparing*. URL: https://goo.gl/mzz5e3 (visited on 02/02/2018).

Robert McNeel & Associates (2017). *Rhinoceros, version 5.0*. URL: https://www.rhino3d.com/ (visited on 26/11/2017).

Rutten, David (2017). *Grasshopper3D, version 0.9.0076*. URL: http://www.grasshopper3d.com (visited on 10/10/2017).

scipy (2018). *Scipy libraries*. URL: https://scipy.org/.

Solihin, Wawan and Charles M. Eastman (2016). 'A Simplified BIM Model Server on a Big Data Platform'. In: 33rd CIB W78 Conference.

Solihin, Wawan, Charles Eastman, Yong Cheol Lee and Dong Hoon Yang (2017). 'A simplified relational database schema for transformation of BIM data into a query-efficient and spatially enabled database'. In: *Automation in Construction* 84.August, pp. 367–383. ISSN: 09265805. DOI: 10.1016/j.autcon.2017.10.002.

Stellman, Andrew and Jennifer Greene (2005). *Applied software project management*. " O'Reilly Media, Inc."

Stichting Bouwkwaliteit (2017). *Bepalingsmethode Milieuprestatie Gebouwen en GGW werken*. Stichting Bouwkwaliteit.

Surendran, Sridevi Koonath, Anke Rolvink, Jeroen Coenders and Hans Welleman (2015). 'Embodied Energy Optimization Tool'. In: IASS 2015 Amsterdam.

Turrin, Michela, Peter Von Buelow and Rudi Stouffs (2011). 'Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms'. In: *Advanced*

Engineering Informatics 25.4, pp. 656–675. ISSN: 14740346. DOI: 10.1016/j.aei.2011.07.009.

Wackernagel, Mathis, Chad Monfreda, Niels B Schulz, Karl-Heinz Erb, Helmut Haberl and Fridolin Krausmann (2004). 'Calculating national and global ecological footprint time series: resolving conceptual challenges'. In: *Land use policy* 21.3, pp. 271–278.

Wikipedia (2018). *Wikipedia*. URL: www.wikipedia.com (visited on 02/08/2018).

Witten, Ian H, Eibe Frank, Mark A Hall and Christopher J Pal (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Yang, Ding, Yimin Sun, Michela Turrin, Peter Von Buelow and Joop Paul (2015). 'Multi-objective and multidisciplinary design optimization of large sports building envelopes : a case study .' In: IASS 2015 Amsterdam.

Zhang, Yu, Henry V. Burton, Han Sun and Mehrdad Shokrabadi (2018). 'A machine learning framework for assessing post-earthquake structural safety'. In: *Structural Safety* 72, pp. 1–16. ISSN: 01674730. DOI: 10.1016/j.strusafe.2017.12.001. URL: https://doi.org/10.1016/j.strusafe.2017.12.001.

ACRONYMS

GWP	Global Warming Potential
LCA	Life Cycle Assessment
BIM	Building Information Modelling
LCC	Life Cycle Costing
LCI	Life Cycle Inventory
NMD	Nationale Milieu Database
EPD	Environmental Product Declarations
CAD	Computer Aided Design
IFC	Industry Foundation Class
ML	Machine Learning
SLS	Serviceability Limit State
ULS	Ultimate Limit State

A

APPENDIX: LITERATURE BACKGROUND

The literature background discusses the topics structural design, sustainability in the building industry, data usage and storage, predictive technologies and decision support. Sections are build up to provide background information on the topic and concludes with recent research.

A.1 STRUCTURAL DESIGN

Structural design is a complex iterative, highly non-linear and at times chaotic, cyclic or evolutionary process with the aim to find a solution (Coenders, 2011) (Figure A.1). The designer starts with a blank sheet with the essential requirements from the architect and client. This is the conceptual design phase; creativity is crucial. More information is added during the preliminary phase. It requires analysis with models and calculations. Detailed design requires knowledge on detailing (connections) and in depth drawings. The execution phase is characterised by creating schedules and building order.

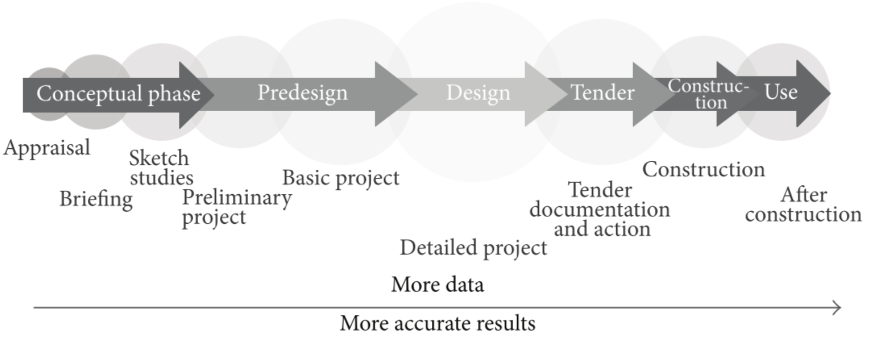


Figure A.1: The design phases of structural design (Bragança, Vieira and Andrade, 2014)

Østergård, Jensen and Maagaard (2016) mentioned challenges in every stage of the structural design: contradicting and stricter requirements, interoperability, limited reuse of knowledge, discrepancy between simulation and real-life measurements, and lack of simulation guidance.

The conceptual design phase is characterised by strong collaboration between the different parties (Heidegger, Coenders and Rolvink, 2014). MacMillan et al. (2001) highlights two important aspects: the success of the project depends highly on the shared understanding between the different parties. And crucial is the free interaction between the disciplines to achieve optimal solutions and reducing compromises at a later stage. Coenders (2011) mentioned that each design, each project and each building is unique due to the specific combination of people in the collaboration resulting in unique processes.

A.2 SUSTAINABILITY IN THE BUILDING INDUSTRY

Sustainability is referred in the dictionary in two ways: "the ability to maintain a certain rate or level" and "avoidance of the depletion of natural resources in order to maintain an ecological balance". The latter statement is often intended when people talk about sustainability. During the World Summit of 2005 three areas were distinguished in sustainable development: environment, economic and social (Adams, 2006). The following subsections describe the performance indicators, the challenges in sustainability assessment, the 5 methods to implement sustainable structural design, and a thorough overview in the LCA procedure.

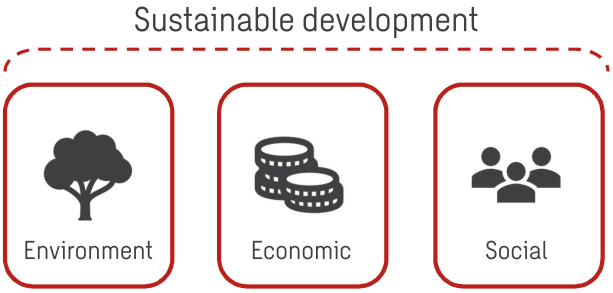


Figure A.2: Three areas for sustainable development

A.2.1 *Sustainable performance indicators*

Bragança, Vieira and Andrade (2014) presents four main indicators to express the sustainability of a building. Firstly the environmental indicators. Both carbon footprint and ecological footprint are included. Oti and Tizani (2015) defines carbon footprint as "...a measure of the exclusive total amount of carbon dioxide emissions that is directly and indirectly caused by an activity or is accumulated over the life stages of a product." For an apple-to-apple comparison for different gasses, the Global Warming Potential (GWP) is used expressed in equivalent kg CO_2 emission. Wackernagel et al. (2004) developed the ecological footprint concept which can be explained by how much natural capital is necessary for the demands in resources.

The energy impact indicators are divided in operational energy and embodied energy. The operational energy, measured in $kWh/m^2/yr$, describes the energy consumption of a building in the use phase (lighting, heating, cooling, ventilation, etc). The share of operational energy compared to embodied energy decreased over the years from around 85% to about 70%, induced by EU regulations and recent energy efficiency measures (Bragança, Vieira and Andrade, 2014). The embodied energy refers to the energy that is used in the production, construction, replacement and end-of-life stage of the building measured in kWh/m^2 . The share on the total energy resource has increased due to two reasons: One, energy saving measures reducing the overall operational energy increasing the relative contribution of embodied energy. Two, these application have increased the absolute value of the energy as well. Surendran et al. (2015) concluded that within the embodied energy the structural load bearing system counts for roughly 82% of the energy in the system.

Economic indicators are in the for of Life Cycle Costing (LCC). It describes the cost of the structure during its entire life time (from cradle-to-grave). It depends on the construction cost, operational cost and the end-of-life cost, all measured in $€/m^2$.

A.2.2 *Challenges*

Inconsistency in the calculated sustainability impact is found to be large in literature (Dixit et al., 2010; Eleftheriadis et al., 2016). Main reason is the use of different system boundaries especially with decided depth of analysis. This causes exclusion of upstream processes and creates large differences in scoring. Additionally, the geographic location and data source have a large influence. Each country has its own raw material production, its factory processes, energy generation, transport methods and distances, making it hard to compare embodied energy on different locations in the world. Different regions have different environmental databases. Varying climates, building methods and indoor conditions produces widely different energy consumption results. Next to this, the results are also time dependent. Factory processes are upgraded each time, transportation modes renewed and made more sustainable, etc.

A.2.3 *Methods for sustainable structural design*

Designing for sustainability is not an easy task for the structural engineer. The design process is cyclic and a chaotic behaviour is recognised. Furthermore, the amount of information at key design stages (conceptual and preliminary design) is low and often lacking in quality. Danatzko and Sezen (2011) points out five design methodologies to minimize the impact of the project and describes the main assumption in sustainable structural design as: "... a structural system that meets the needs of the owner and user while minimizing the environmental impact and conserving resources where possible."

- *Minimize material use:* The objective is to reduce raw materials usage in the structural system. Solutions are to create a more efficient structural system in terms of layout and shape. Optimisation of structural members and shape is another option. Many software tools are available to aid this process.

- *Minimize material production energy:* The production of structural materials such as steel and concrete, requires energy and natural resources. The overall goal is to minimize these aspects in the production chain, making the materials more sustainable. Also, engineers are entitled to choose materials with low environmental impact (e.g. materials which have an energy-efficient production chain).

- *Minimize embodied energy:* An extension of minimizing the material production energy is to look at the entire design life of a building. The embodied energy is a measure for the energy within the structural elements. This methodology gives the engineers incentive to optimize with a balance of both operational and embodied energy. A key aspect is an effective use of natural resources to reduce the consumption in the use phase.

- *Maximise structural reuse:* Reuse of the structural system is a way of containing the embodied energy instead of using extra energy for recycling. It forces the engineer to create structural systems which allows of easy disassembly. Furthermore, it gives incentive to assess material types and elements which can be reused easily. Standardisation of connections and structural elements will further enhance this methodology in th future.

- *Life Cycle Assessment (LCA):* A common tool for sustainability assessment is the employment of LCA. It determines the sustainability properties in various areas of the structural system. This allows for optimisation and balancing of the numerous sustainable aspects. Section A.2.4 will describe this methodology in detail.

A.2.4 *Life Cycle Assessment (LCA)*

Life Cycle Assessment (LCA) is a widely used and applied technique to assess the sustainability of the building during its entire life. It evaluates resources to calculate the impact of a building on a product or whole building level (Lolli, Fufa and Inman, 2017). The procedure of performing an LCA is fully documented in the Eurocode and further described in the Dutch National Annex (Stichting Bouwkwaliteit, 2017).

Assessment on building level is made in four steps (Figure A.3): defining the scope and goal, collecting the data of building products and building elements, environmental scoring in the impact categories and shadowprice ('schaduwprijs'), and finally interpretation of the results.

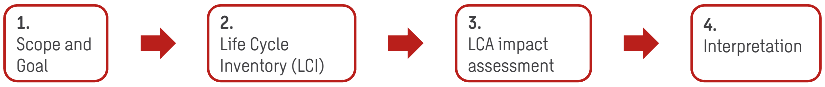


Figure A.3: Procedure assessment on building level

SCOPE The functional unit can be expressed in terms of a description, specification and the amount of the product. The supplier is responsible for the relevant information on quality and durability (service life). An important aspects in the procedure is the choice of system boundary. An overview of the available boundaries are given in Figure A.4. The process starts with the production of the material. Extraction, transport to factory and manufacturing are included in this phase. The materials are delivered to the site and used in the construction of the building. The operation or use phase is next with regular maintenance and replacement. In this phase the building uses most energy for lighting, heating and cooling (operational energy). The end of life concludes the cycle with demolition of the building elements. Re-use, recycle and recovery is possible respectively in the form of implementation in other buildings, recycle into new products, and burning to receive energy. Products that are not usable end up in a landfill.

LIFE CYCLE INVENTORY (LCI) The selection of data is described in the EN 15804 (NEN-EN 15804, 2013, p.26) with the rule of using specific data for particular product processes whenever possible. Especially in the manufacturing phase (A3) specific data of the manu-

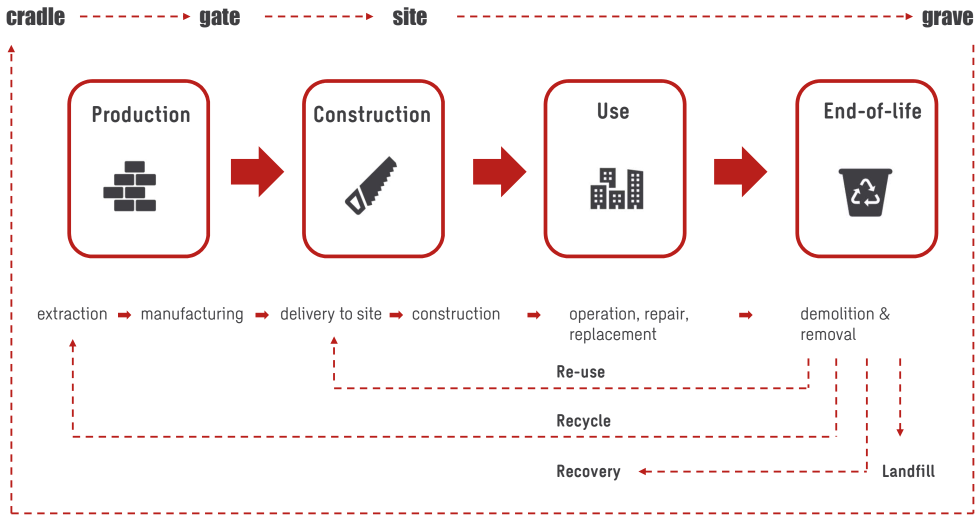


Figure A.4: System boundaries (based on De Wolf et al. (2016)

facture need to be used or an representative average. In the extraction of resources (A1) and the phases A4-A5, B and C, generic data can be used. The Stichting Bouwkwaliteit (2017) gives extra guidance in assessing these types of data for the Dutch market. The quality of the data is essential, must be as current as possible and assessed with a data quality system.

The data of each component within the system boundary is collected and used to assess, per unit process, its input and output. Sources are carefully mentioned with descriptions of each process. Data classification can be done in raw material inputs, energy inputs, products, waste, release to water, air, and ground (NEN-EN-iso 14044, 2006). Calculation follows the same strategy with clear sources and descriptions. Validation of the results is essential with providing a calculation of the mass and energy balance. Refinements on the chosen system boundaries is carefully explored.

LCA IMPACT ASSESSMENT The impact assessment is carried out in categories mentioned in the EN 15804. Extra categories have been added by the Dutch annex (Stichting Bouwkwaliteit, 2017, p.26-27). The final assessment is performed by taking the calculated data from the LCI, applying it for every mentioned category above, multiply it with the characteristic factor presented in the NMD and adding each process in the category. This results in an environmental profile in 10 categories. The weighing factors can be used to generate a scoring in Euro. This helps to clearly see which category is predominant. Adding these values to one score results in the shadowcost. Important is that the factors are different for each region depending on the environmental conditions and resources.

INTERPRETATION Interpretation of the results is required after performing the LCA. A sensitivity analysis on assumptions and choices is performed (Stichting Bouwkwaliteit, 2017, p.27-28). A set of scenarios need to be tested in which the outcome may not exceed a 20% difference with the original outcome. Additionally the EN 15804 (NEN-EN 15804, 2013, p.43) requires a critical assessment on data quality.

DRAWBACKS Hollberg and Ruth (2016) stated that today's LCA procedure has drawback related to complexity and time efficiency. A building consists of many materials and components, making the quantity take off analysis a demanding and time consuming job. Material databases are available but lack in quality. Lolli, Fufa and Inman (2017) further stated that there are databases available however, they rely on generic data and require a developed platform for LCA calculations. Within the long life span (50 to 100 years), function may change and renovations are certain. Within the calculated LCA score this posses great uncertainty. At the end-of-life demolition and recycling is often difficult due to interconnected products. The question is to what extend the building is being recycled. This if difficult to determine at the start of the building's life cycle.

SOFTWARE LCA software is used to assess the scoring in an easy way. Different uses are distinguished by Oti and Tizani (2015): for product comparison, as a decision support tool and assessing the sustainability of the whole building. According to Hollberg and Ruth (2016) a division between tools can be made in the following categories:

- *Generic LCA tools:* Typical tools are Gabi or OpenLCA of which the later is open source. Usage requires extensive knowledge on the LCA procedure. The interaction with the software is not ideal looking at the design process. Often models need to be recreated and detailed information need to systematically provided. The design process on the other hand is dynamic and quickly changing.
- *Spreadsheet based tools:* In combination with a material database and material quantities, the environmental impact is calculated. Manually input during the entire process makes it error prone and time consuming, resulting in underusing the optimisation potential.
- *Component catalogues:* Online component databases are used to perform an LCA. This database can be constantly updated in

the background. However, it lacks the ability for an integrated approach, leading to a lot of manual work.

- *BIM integrated:* Recent developments in BIM have produced a number of environmental plugins (Tally, Impact) which are integrated in the environment. Quantities are automatically taken for the 3D model and the LCA calculated in combination with environmental data. This integration is a step in the right direction. However, limitations are present the often complex modelling making it less suited for smaller projects.
- *Parametric and associate design integrated:* Flexible parametric and associate models are used in the conceptual design phase of the building. The integration with a sustainability assessment calculation creates a powerful system to steer on better performing buildings in terms of environmental impact. However, applications are rare due to the many often conflicting parameters (Heidegger, Coenders and Rolvink, 2014; Hollberg and Ruth, 2016).

A.2.5 *Service life of building*

Assessing the service life is essential for calculating the impact score with direct impact on the use stage of a building. EN 15804 (NEN-EN 15804, 2013) presents an estimation of the service life based upon "... empirical, probabilistic, statistical, deemed to satisfy or research (scientific) data...". It later stated that the Estimated Service Life (ESL) is depending on the service life of the load bearing structure as it is non replaceable. The national annex (Stichting Bouwkwaliteit, 2017, p.35) provides the service life upfront: 75 years for residential buildings, 50 years for buildings with functions as offices, schools, shops, etc, and combinations of residential and the latter functions with 75 years.

A.2.6 *Nationale Milieudatabase (NMD)*

A central role in the LCA assessment is the source of data on materials and products. In the Netherlands the Nationale Milieu Database (NMD) is used for this purpose. It contains three sets of data: the process database, the basic profiles and the product cards (Environmental Product Declarations (EPD)). The process database is concerned with the input and output of relevant processes as material production, transport, manufacturing and waste treatment. This information is provided by the data owners (suppliers). A basic profile for a building element is created by using the process database. Product cards for building products and building elements are produced using the

CATEGORY	UNIT	WEIGHT FACTOR(€)
Depletion of abiotic resources (fossil + elements)	Sb eq	0.16
Global warming	CO_2 eq	0.05
Ozone depletion	CFK-11 eq	30
Photochemical ozone creation	C_2H_4 eq	2
Acidification of soil and water	SO_2 eq	4
Eutrophication	CFK-11 PO_4	9
Human toxicity	1,4-DCB eq	0.09
Fresh water ecotoxicity	1,4-DCB eq	0.03
Marine aquatic ecotoxicity	1,4-DCB eq	0.0001
Terrestrial ecotoxicity	1,4-DCB eq	0.06

Table A.1: Impact categories with complimentary units and weighing factors (source:NMD)

needed basic profiles. It's mandatory to include at least the production, recycling and demolition of the product (Stichting Bouwkwaliteit, 2017, p.11). Verification of data is done by independent parties.

The data in the NMD provides environmental information in 10 impact categories (Figure A.1). Each product and material is assessed in these categories. Result is a environmental profile in respectively per element or unit mass of material. Furthermore, weighing factors per impact category are provided to express the impact categories in unit cost ('schaduwprijs'). This is related to the amount of money needed to mitigate the environmental effects. Moreover, this also enables addition to one score for each element which is useful in comparing alternatives.

A.2.7 *Certificates*

The Building Research Establishment Environmental Assessment Method (BREEAM) rating system, besides Leaderschip in Energy and Environmental Design (LEED), is a popular sustainability scoring system for buildings. It advantage compared to LEED is that it can be altered for local applications and codes (Marjaba and Chidiac, 2016). The BREEAM-NL certification is for example used in the Netherlands. A total of 9 categories containing 73 factors are assessed in this procedure. Health, management, energy, transport, water, materials, waste, land-use and pollution are part of the categories. Assessment based on materials (in BREEAM Mat-1) is just 1 of these 73 factors. In the Netherlands this calculation is mandatory for offices and houses, and is called MPG (MilieuPrestatie Gebouwen) or MKI (Milie-Kosten-Indicator) (Stichting Bouwkwaliteit, 2017).

A.2.8 *Implementations*

Lolli, Fufa and Inman (2017) proposes a spreadsheet based parametric tool in order to evaluate the wall and windows components in a building, optimise it in minimal operational energy use, embodied CO_2 emissions and embodied energy. Markelj et al. (2014) proposes a spreadsheet based Simplified Method for Evaluating Building Sustainability (SMEBS) which determines the fulfilment of the sustainability demands in an early phase. The used weighing factors for the 33 parameters are calibrated by interviews done on experts in the sustainability field. No external assessors is needed and the scores are quickly obtained. However, the implementation is location specific. Local differences in the priority of the weighing factor have large influences on the outcome.

The Function Impact Matrix design strategy is used for specific product design but the essentials can be implemented in the building industry (Devanathan et al., 2010). It states that every new design is a novel combination of existing concepts, which can be broken down into knowledge and assessed individually. The methodology can be described as assessing an LCA on numerous products, or in this case buildings, in which afterwards the environmental impact is distributed over the functions to establish function impact correlations.

A range of parametric tools are nowadays available to assess and optimise parameters as optimal structure, optimal daylight and comfort. However, not many tools are developed that both look at operational and embodied energy (Lolli, Fufa and Inman, 2017). Hollberg and Ruth (2016) proposes a simplified parametric LCA procedure for architects in which the primary energy demand, embodied energy and life cycle impact is calculated in real time. Assumptions on energy standard, material and heating system are made based on typical solutions. A software prototype is implemented in Grasshopper (Rutten, 2017). Oti and Tizani (2015) proposes a modelling framework combining the indicators life cycle costing, ecological footprint and carbon footprint, in a sustainability assessment for comparing design alternatives. These steel designs are provided with a multi criteria analysis. A prototype is developed and connected with the BIM modelling software

Revit (Autodesk Inc., 2017) through its API interface. Coenders (2013) proposes an open-source sustainability framework to offer a clear and insightful way of calculating and assessing the sustainability performance. The open-source nature allows for anyone to contribute to the project with additional components. Implementation is performed in the parametric environment of Grasshopper. Currently it includes 'designers', 'analysis methods' and 'assessment methods', which can respectively encode knowledge of engineers, perform thermal analysis, and assess the total energy of the building.

A.3 DATA AND DATABASE TECHNIQUES

The future of Building Information Modelling (BIM) and the structural field in general focusses on two visible trends (Solihin and Eastman, 2016). Firstly, multiple models are connected to enhance the usability for building managers in maintenance, operation and end-of-life scenarios. Creating large city-models opens up the concept of city analysis. The second innovation is related to increasing use of laser scan point cloud data for existing buildings. Both mentioned aspects are in need for efficient large scale data storage. As of this moment, data in models is locked in their proprietary formats or in the open source cross platform IFC standard (Solihin et al., 2017). The latter is a good starting point but the data is not structured in a way that allows for easy searching. This aspect is more and more important due to the increasingly amount of data captured in BIM models (Solihin et al., 2017). Data can only be as useful as the information it can generate to the end-user.

Large scale acquisition of data from BIM fits within the area of so called 'Big Data'. The term refers to massive and complex datasets made up of a variety of data structures including structured, semi-structured, and unstructured data (Grolinger et al., 2013). Laney (2001) describes it as follows: "Big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization". The three V's are further elaborated by Becker, Tilman and Curry, Edward and Jentzsch, Anja and Palmetshofer, Walter (2016):

- *Volume:* Dealing with large amount of data within data processing
- *Velocity:* Dealing with high frequency of incoming real time data (e.g. sensor data)
- *Variety:* A large range of data sources and types need to be dealt with

A Value Chain can be described as a continues value adding activities for getting a better understanding of the process (Becker, Tilman and Curry, Edward and Jentzsch, Anja and Palmetshofer, Walter, 2016). Curry (2015) describes this chain for big data characterised by the high level activities acquisition, analysis, curation, storage and usage.

A.3.1 *Acquisition*

In the building sector a few ways of acquiring data can be sketched. Data transfer from BIM models and IFC files to a database is under interest in the last years (Solihin et al., 2017; Li et al., 2016; Eastman, 2016). Main focus is the mapping from IFC format to a structure that is efficient, easy to query and fits within current database structures. Datasets can be generated through scripts, functions and models. However, limited interrelations can be created if there is a need for a large dataset. This means that deep predictive algorithms cannot be applied.

A.3.2 *Analysis*

The main task in this phase is to add structure to the raw data (often in many forms, dimensions and file formats), and prepares for the decision making process. With the 3 V's in mind, new techniques developed around placing scalability at the centre of development. This ensures high volume throughput with large scale reasoning, data mining and ML (Becker, Tilman and Curry, Edward and Jentzsch, Anja and Palmetshofer, Walter, 2016, p.64). Large scale reasoning is defined by principles as soundness and completeness, a far reach from reality. Data is often contradictory and incomplete. Literature shows (Delen, Walker and Kadam, 2005) that around 80% of the time is put in refining the data in such away that it can be used for decision making.

A.3.3 *Curation*

Data quality is the main principle to ensure high quality analytics. Problems emerge in the increase of data sources (volume) and in the complexity and variety that needs be coped with. Becker, Tilman and Curry, Edward and Jentzsch, Anja and Palmetshofer, Walter (2016) describes data curation as "... the methodological and technological data management support to address data quality issues maximizing the usability of the data". A challenge is to include unstructured, less frequently used data together with structured data providing a more comprehensive model.

Solutions in improving data quality are numerous. Special data management systems are able to remove duplicates and standardize data syntax, rating systems are implemented with algorithms that categories the data, and crowdsourcing is used to speak to the 'wisdom of the crowds'. The latter is applied in for example 'Wikipedia' (Wikipedia, 2018) where checking and data addition is done by volunteers. Other examples are platforms as 'FigureEight' (FigureEight, 2018) which divides data in small tasks such as simple classifications of images.

A.3.4 *Storage*

A database is a way to store data and information in a structured, organised and viewable way. The core principles of a database transactions are described with the ACID keys proposed by Haerder and Reuter (1983): atomicity, consistency, isolation and durability. With atomicity the data transaction is described as all or nothing. This ensures that the database is not filled with partly faulty data. Consistency requires the written data to comply with set rules and constrains. Isolation describes that each transaction to a database must not influence other transactions. Once this transaction is written to the database, the durability principle states that it will be stored permanently even with a loss of power or server crash.

A.3.5 *Usage*

Core usage of data is to support the user in the decision making process. This consists of reporting, exploration of the data and finding correlations, comparisons and what-if scenarios through searching (Becker, Tilman and Curry, Edward and Jentzsch, Anja and Palmetshofer, Walter, 2016). The discovery of new relations and dependencies can lead to new economic opportunities and to higher efficiency. Deeper connections in the data provide better understanding of all the dependencies and makes the system more transparent.

Exposing the data and results through visualisation is an important aspect in the decision making process. Datasets are large and to make the results manageable and effective, a well visualised representation is needed. If not, the decision making process can have slowdowns and lack of confidence (Becker, Tilman and Curry, Edward and Jentzsch, Anja and Palmetshofer, Walter, 2016).

Application in the building sector is on its way with Industry 4.0, the new evolution of the manufacturing industry using many IT technology based on interaction with large datasets. Examples are point cloud data, interaction with real time sensor data and using predictive technologies in design and maintenance.

A.3.6 *Implementations*

Lee et al. (2014) developed an object-relational IFC (OR-IFC) server using the object-relational database ORDB approach to improve query performance by simplifying the mapping process from IFC format to OR-IFC. It combines the advantages of the object based OODB and the stability of the RDB Benchmarking showed a clear performance improvement over a standard RDB implementation.

Similarly (Li et al., 2016) implements a novel object-relational storage model that uses Oracle database to store IFC data. It demonstrates the potential usage of high level queries in building design, with requesting all elements on the second floor or creating a construction schedule from presented data.

Solihin et al. (2017) proposes an easy, efficient, and fast query-able building model with build-in spatial support using the data warehouse star-like schema. It maps the BIM data to an Oracle Relation Database. This method allows for searches and insight in BIM data with sophisticated queries such as properties and spatial location. This however keep the RDB in place due stability of the platform and maturity of the query language. Researched conducted by Solihin and Eastman (2016) implements a proof-of-concept to transform BIM data into a NoSQL based graph database.

A.4 PREDICTIVE TECHNOLOGIES

Prediction can be used to aid the decision making process. Looking closely at this procedures three distinct levels of aid, with increasing complexity, can be described (Becker, Tilman and Curry, Edward and Jentzsch, Anja and Palmetshofer, Walter, 2016):

A.4.1 *Lookup and Learning*

Lookup can be described as retrieval of known items. Examples can be searching for information on the building design just to verify that you taken the right value (e.g. a profile, material, dimension). Additional functionality can be implemented to navigate through the datasets. This could be to have an overview of what building design are present and increase confidence.

Learning is more directed to retrieving information of unknown items. This supports simple searches for information (data acquisition) in for example what building type is correlated with floor area range. It enables real comprehension on the underlying data. Comparison, aggregation and integration of the data is visible. An example could be to search for a floor area range of which the algorithm outputs the sustainability impact with most correlated parameters in an aggregated and integrated way. Furthermore, a range of data points could be selected to allow for comparison and increase the user's confidence of the outcome.

A.4.2 *Supervised Machine Learning (ML)*

Jootoo et al. (2017) describes supervised machine learning as "encompasses algorithms that use data, usually in large amounts, to develop statistical models that can make predictions based on new instances of similar data." Supervised learning, in contrary to unsupervised learning, uses a labelled set of training data to estimate the input data to the desired output data (Kourou et al., 2015). A standard procedure of ML implementation is described by Witten et al. (2016, p.29). Three core steps are distinguished:

- *Data preparation:* In the data preparation phase the dataset is preprocessed in order for the algorithm to find a model. Improving the quality of data is essential for a representative outcome. In some occasions modification of data is needed to create a better fitting in the ML method. Possible technique are dimensional reduction, feature selection and feature extraction. With lower dimensions (fewer variables) ML algorithms perform better. Extra benefits are the exclusion of irrelevant features, less noise and better prediction results (Kourou et al., 2015).
- *Modelling:* Modelling and data preparation are a combined activity. The modelled data provides new feedback on the preparation of the raw data.
- *Evaluation:* A crucial part is the evaluation of the obtained results. Performance on a set of data doesn't guarantee good performance in real-life scenarios (with new data). Methods to check and validate performance are the Holdout Method, Random Sampling, Cross-Validation and Bootstrap (Kourou et al., 2015). In the Holdout Method the data is split into a training and test set. A model is generated based on the training set and performance is tested on the test set. Random Sampling is based on the same concept although here the test and training sets are chosen multiple times randomly. Cross validation uses the data only once for testing and multiple times as training. In Boot-

strap the samples are separated in training and test sets but are again put back into the dataset.

Jootoo et al. (2017) points out that the confidence the user has on the prediction by the ML algorithm, is directly correlated with the depth the decision making process is explained to the user. This aspect is fundamental looking at possible implementations in the civil engineering design process.

A.4.3 *Algorithm choice*

The choice of ML algorithm is depending on many factors, making it a rather difficult procedure (Microsoft Azure, 2017). A key aspect in this choice is the type of data and how it is structured. Is it linear or is data clustered in groups? What accuracy is needed and what speed is required for the implementation?

A.4.4 *Limitations*

Amasyali and El-Gohary (2018) points out two main limitations when applying ML algorithms. First, it's difficult to assess if the prediction model performs well outside the training set, i.e. with real data. It strongly depends on the amount of training data and the general bias which is present in the data. Essential is to acquire representative training data although this can be costly and difficult to get. A solution is to set specific boundaries for the usage of the application. Secondly, predictive models often function as black-box, in other words it's difficult to understand the logic inside the prediction. Some ML algorithms are more easy to comprehend, such as decision trees. Possible solutions are so called grey-boxed approaches, which apply physical with predictive assessment procedures.

A.4.5 *Implementations*

Few implementations are made using supervised ML as aid in structural design (Jootoo et al., 2017). However in other structural engineering related applications, building envelop and energy design, and in the medical industry, examples are numerous. Jootoo et al. (2017) applied predictive algorithms in aiding the structural designer in an early stage on choosing the statistically optimal bridge type to increase the likelihood of optimised design, design standardisation, and reduced maintenance costs. Data of more than 600,000 bridges from the National Bridge Inventory (NBI) database were analysed and assessed on key attributes with feature selection techniques. Decision tree, Bayes network, and support vector machines where implemented to make the prediction. Resampling of the data was needed to reduce the bias to more common bridge types. More general application of ML in structural engineering is seen in for example component level analysis (Jootoo et al., 2017). Also, optimisation, a field close to ML, is under great interest the recent years (Jootoo et al., 2017). Recent research on assessing post-earthquake safety, integrates ML algorithms to map response and damage patterns to the structural safety state (safe or unsafe to occupy) of the building based on an acceptable threshold of residual collapse capacity (Zhang et al., 2018). Classification, regression tree and random forests are applied in the framework. Mangalathu and Jeon (2018) proposes the classification of failure mode and the prediction of associated shear strength of beam-column joints under seismic loading to be made with help of ML techniques. Extensive experimental data is gathered and used for training and testing. A wide range of ML algorithms are tested of which Lasso regression performed the best.

Energy consumption prediction has been researched a lot in recent years. A thorough review of available literature in this area is presented by Amasyali and El-Gohary (2018). It concluded that "There is no one-size-fits-all model that can be utilized under all conditions" and therefore each instance requires extensive research in model data and prediction algorithm. Research by Kim et al. (2018) showed the use of acquiring feedback of occupants' heating and cooling behaviour based on a personal comfort system, for the development of personal comfort models to predict individuals' thermal preference. Six different ML algorithms were deployed of which some had high accuracy but more computational cost.

A wide range of examples and applications have been successful in the medical field. Delen, Walker and Kadam (2005) implemented supervised learning to predict the survivability of cancer patients. A large dataset (more than 200.000 cases) are used to build the prediction model. The data mining algorithms decision trees and neural networks were used. Research by Hachesu et al. (2013) implemented supervised learning in prediction the length of stay of patients in a hospital bed. About 5000 records were used and implemented with decision tree, support vector machines (SVM), and artificial neural network (ANN) as techniques.

A.5 DECISION SUPPORT IN CONCEPTUAL DESIGN

Decision support is essential in conceptual design. Research is performed to find the current state of knowledge.

A.5.1 *Multi Objective Optimisation (MOO)*

Multi Objective Optimisation (MOO) gives designers the opportunity to optimise for a set of conflicting objectives and specify the trade-off between them (Yang et al., 2015). The set of best solutions are forming the Pareto frontier (Figure A.5). Instead of using a single objective optimisation for each objective in isolation, MOO allows multiple disciplines to find best solutions. This allows the design team to explore a larger area of the design space.

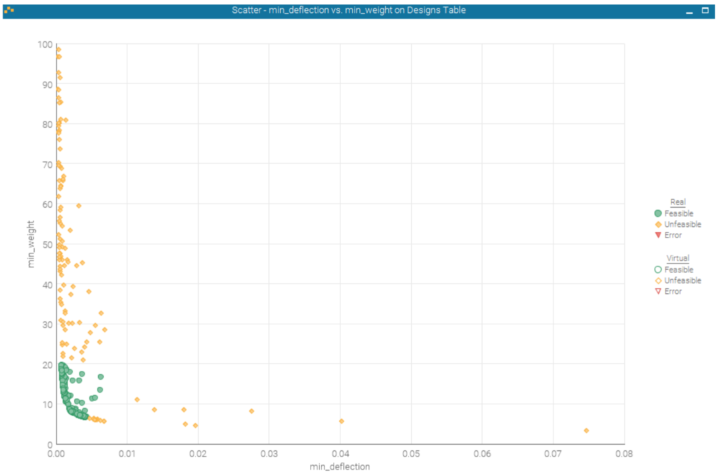


Figure A.5: Visible Pareto frontier with objectives minimum weight and deflection (from modefrontier)

Yang et al. (2015) points out current limitations of applying MOO in the building sector. It shows that within the building sector the dependencies of discipline are not strongly connected. This makes the application of MOO less useful. Furthermore, little focus is put on the visualisation and interpretation of the optimisation results. Often complex problems result in difficult to comprehend results. Interpretation required expert knowledge. MOO is also often limited in analysing complex geometry. Guiding the optimisation in for example an attractive shape is difficult as aesthetics are not easily quantifiable (Felkner, Chatzi and Kotnik, 2013).

A.5.2 *Optimisation with guidance*

Optimisation tools, when used uncritically and uninformed, offer architectural designs which seem random and unpleasant (Felkner, Chatzi and Kotnik, 2013). Aesthetics are difficult to put in such a form that an optimisation algorithm can take it into account. Felkner, Chatzi and Kotnik (2013) proposes an interactive numerical framework that supports the conceptual design process by given the user control over the direction of the optimisation. Its objective is to generate solutions in which the user can easily change constrains or make decisions based on aesthetics (Figure A.6).

Important in this framework is the low technical difficulty for the user. Furthermore, the user needs to compare design alternatives and the framework should provide support for produced results. All these aspects increase the flexibility of the tool.

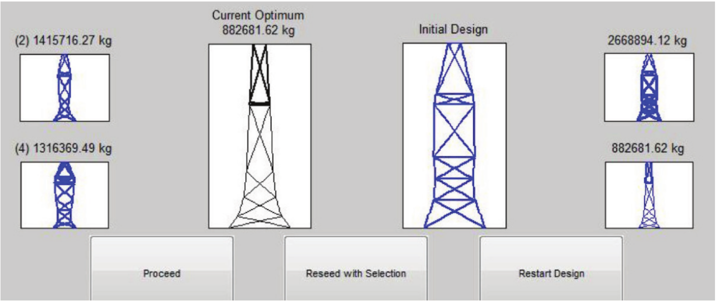


Figure A.6: Visible Pareto frontier with objectives minimum weight and deflection (Felkner, Chatzi and Kotnik, 2013)

A.5.3 *Parametric scripting environment*

The advancement of parametric software tools as Grasshopper and Dynamo, opens up the ability to explore a larger part of the design space. However, guidance and feedback is required in the conceptual design phase to find better solutions. Kurilla, Achten and Florián (2013) proposes a framework and application to aid the decision making process of the user regarding the structural options. No optimisation algorithm is implemented as these offer solutions which are often blindly chosen by inexperience users. It implements a decision support mechanism based on the comparison of alternatives, stored in the field of solutions. This has as the advantage that the user is more aware of the trade-off between the parameters. Implemented decision support mechanisms are comparison of current design against last design, and the comparison with best design found so far. Furthermore, filtering and visualisation of stored design alternatives is done.

A.5.4 *Data driven design*

Today's design process contains a lot of performance data. It is often difficult to process these in such away that they can be applied in design. Liu et al. (2015) proposes a data driven approach to support this design process. It incooperates information reading from a BIM model, analysis of energy consumption and data mining techniques as clustering and rule learning. When data is generated by the analysis tools, the user receives feedback on which parameters to choose. Is is based on a clustering-density method of which a large area indicates a high priority, and is offered as suggestion. Association rule learning is implemented to find possible correlations between parameters and their values, and is also used as feedback method. This data drive workflow is repeated until final design.

B

APPENDIX: SUSTAINABILITY IMPLEMENTATION

The sustainability impact is essential in this research project. This Appendix discusses the proof-of-concept implementation.

B.1 GOAL

The LCA procedure starts with defining a clear goal. This research project is interested in comparing generated structural designs on environmental impact. Although a qualitative answer is found, this is not required. Each design will have the same system error at each calculation. The difference between the generated design is of interest. The scope is kept small with the assessment of only structural members (beams, columns, purlins, braces), excluding the foundation and floor, constructed in steel of industrial warehouses. The Dutch market is used as location. The final sustainability score is based on the functional unit €/m^2/year, according to information provided by Stichting Bouwkwaliteit (2017). The cost is related to the amount of money needed to mitigate the environmental effects. The area is assessed as 'bebouwd-vloeroppervlak' (BVO), build-area. The per year relates to the total impact divided by the service life of the building. In case of warehouses 50 years is taken (Section C.2).

B.2 INFLUENCE OF CHOICES

The scope is limited to allow for a workable proof-of-concept. This has large influences on the final sustainability score. The main points are further elaborated.

- *Dutch market:* Choice of the Netherlands has consequences for the sustainability calculation. The Dutch implementation of EN-15804 is followed which includes additional requirements and interpretations (Stichting Bouwkwaliteit, 2017). One large consequence is the mandatory use of the Nationale Milieu Database (NMD), further explained in Section B.4.
- *Limited elements:* Only four structural member families are taken into account in the structural generation and sustainability calculation. This represents the main load bearing structure. It excludes the additional steel for mounting the façade panels and

extra stability elements in roof and façade. Bouwen met Staal (2013) mentioned that this can be up to 10% of the total steel in the warehouse. Additionally, no extra steel for connections are taken into account (e.g. bolts, end plates). Sources at IGG (2018) provided estimations between 7% and 10% of the total amount of steel is due to connections. The actual sustainability score of build warehouses will therefore be larger.

- *Floor:* The ground floor of the warehouse is not taken into account. Literature points out that floors can take up between 30% and 75% of the total environmental impact depending on the type and function of the floor (Foraboschi, Mercanzin and Trabucco, 2014; Lankhorst, 2018). This is significant however, the floor of an industrial warehouse is most often specifically designed for a function. A large variation is expected and in the data generation stage not enough information is available. Including it would mean a fixed increase for each design as for each design the same thickness would be assumed (connected to a specific use case). The above mentioned factors contributed to not taking the floor system into account.
- *Foundation:* The floor and foundation are strongly correlated. Decisions in for example pile distances directly effect the thickness of the floor. The main steel load bearing structure is of underlying importance due to the low force transfer compared to the expected floor loads. Neglecting the foundation causes heavier structures to be more favourable compared to lighter structures. However, in the explained scope high variations are not expected. Furthermore, foundation is also very location specific due to soil conditions. This result in high uncertainties of the applied foundation and is therefore disregarded.

B.3 SYSTEM BOUNDARY

A well defined system boundary is essential to interpret the results and open up the possibility of replication. Additionally, future entries to the database should in-cooperate the same boundaries. In this research project a *cradle-to-gate* boundary is implemented (Figure B.1). Main reason is to reduce complexity and allow for automation in the 'data generation' workflow (Chapter 4). The assessment takes into account the sustainability impact of the extraction of raw materials (A1), transport to the production facilities (A2) and the production of the materials (A3). Influence of these assumptions are elaborated.

- *Construction (A4 and A5):* The contribution of transport to the total sustainability depends on two aspects (Kellenberger and Althaus, 2009; Braendstrup, 2017; Kaethner and Burridge, 2012):

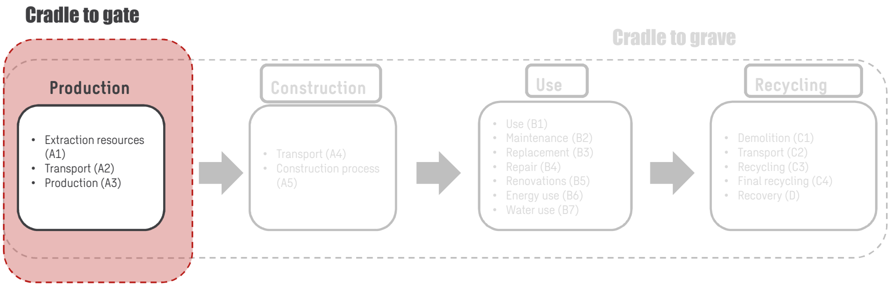


Figure B.1: Chosen system boundary, a cradle to gate implementation

first, what kind of material, size, shape and mass the object has. Second, the distance between the factories and the specific construction site. It is closely linked with the construction process on site. Indications range from 1% to 20% but are overall uncertain and greatly dependant on the specific project and location. Furthermore, reliable and quality data on the construction processes are hard to find (Hollberg and Ruth, 2016). In the generation of the industrial warehouses automation is key. The site location is unknown and building method unsure. Combine this with the relative low contribution to the total impact, it is decided to neglect this phase in the sustainability calculation. This has a negative effect on design combinations which are easy to construct, need less heavy machinery and are fast to build.

- *Use phase (B1-7):* The main load bearing structure is made for its entire life. Replacements, maintenance, repairs and renovation are of little importance and can be considered negligible. The energy and water use are lower in priority considering the warehouse function. Rai et al. (2011) mentioned that the ratio of embodied energy and energy use of a warehouse during a 25 year lifespan is roughly 1 to 2. The embodied energy becomes very relevant when warehouses are build for a short lifespan. As with the construction phase, the specific information is not available. Neglecting this phase results in negatively impacting the designs which are more efficient in energy use (e.g. due to shape).

- *Recycling (C1-4):* Design with reuse in mind reduces the impact significantly (Section A.2.3). This is however very difficult to assess to what degree, especially considering the limited information present in the data generation phase. Similarly with recycling. The data on the impact of steel is widespread due to the varying amount of scrap metal put in (Kaethner and Burridge, 2012). Section B.4 provides more detail on this topic. Steel is relatively easy to recycle. However, due to high uncertainties this phase is neglected in the environmental calculations.

B.4 DATA SOURCE

The core of the LCA procedure is the available sustainability data on materials, products and processes. In the Netherlands the Nationale Milieu Database (NMD) is used to store this information, resulting in an identical 'calculation core' as described in Stichting Bouwkwaliteit (2017, p. 5). Background information can be found in Section A.2.6.

This research project uses NMD version 2.1 of November 2017 for the environmental data. This is provided by a supervisor of this project. Certain points on the data need to be discussed.

- *Raw material vs building elements:* Both the raw material steel in impact per kg material, as steel building elements are provided in the NMD. The raw material includes a cradle-to-gate system boundary and is easy to implement in the data generation phase. The building elements (e.g. IPE beams, HEA columns) include a cradle-to-grave boundary and therefore include recycling. This difference is important to consider.

- *'Bouwen met Staal':* The steel branch organization of the Netherlands, Bouwen met Staal, also provides information to the NMD and on their website. A large difference is visible between the average of steel impact in the NMD and this new data. Consulting experts in the field revealed that Bouwen met Staal based their data on the production facility Tata Steel in IJmuiden instead of an European average. Questions can be asked on this decision. Especially since steel is a global product and shipped all over the world. Percentage of scrap metal is the dominate factor for the sustainability impact. High uncertainties and unclear information exists around this topic. It is often difficult or impossible to find the exact background information of the data. The NMD also doesn't show assumptions made in the calculation. The decision is made just to follow the latest information in the NMD.

- *Weighing factors:* The final score in € is determined by the weighing factors or 'schaduwfactoren'. These are also included in the NMD. The influence of these factors on the final score is high and Stichting Bouwkwaliteit (2017) mentioned that consensus on these values is of discussion. Future changes are inevitable.

C

APPENDIX: STRUCTURAL DESIGN OF INDUSTRIAL WAREHOUSE

C.1 GEOMETRY

Industrial warehouses are often constructed in a simple rectangular layout. Portal frames span between the two supports consisting of beams or trusses and columns. Lateral stability of the portal frame is provided by purlins and wind braces. However, several variations in industrial warehouses are observed (Figure C.1).

- *Purlins:* Warehouses can be constructed with or without purlins. The first is visible with increased frame distance. The load is first transferred to the purlin and then to the beam. No purlins are applicable with smaller frame distances in which the purlin is not loaded and only used as lateral torsional buckling support for the beam or truss.
- *Connections:* The connections between column and beam or truss can be made as completely hinged or with some rotational stiffness. Decision is made based on the trade-off between cost of the connection and the favourable moment distribution.
- *Stability system:* Stability can be provided by a braced frame with wind braces or an unbraced frame with stiff connections. A three-hinged frame is a typical solution for an unbraced industrial warehouse. Stability perpendicular to the frame can be made with wind braces or with the roof. From a structural point of view, the braces are more favourable in force distribution when they are at the edge of the roof structure. This prevents compression running all the way through the beams.

ROOF & FAÇADE The roof is a water retaining layer with insulation and acoustic capabilities. Typically it consists of a bitumen layer, insulation material and a load bearing structure. Thermal performance, acoustic performance and fire safety are important variables to consider in the roof design. Furthermore the span and loading conditions have consequences for the thickness and type of package. Three common types of roof can be distinguished (Figure C.2). A sandwich panel is an integrated package of steel load bearing structure with insulation in between. Limitations in span are often present. Due to a low self-weight and compact package, the panels are easy to transport

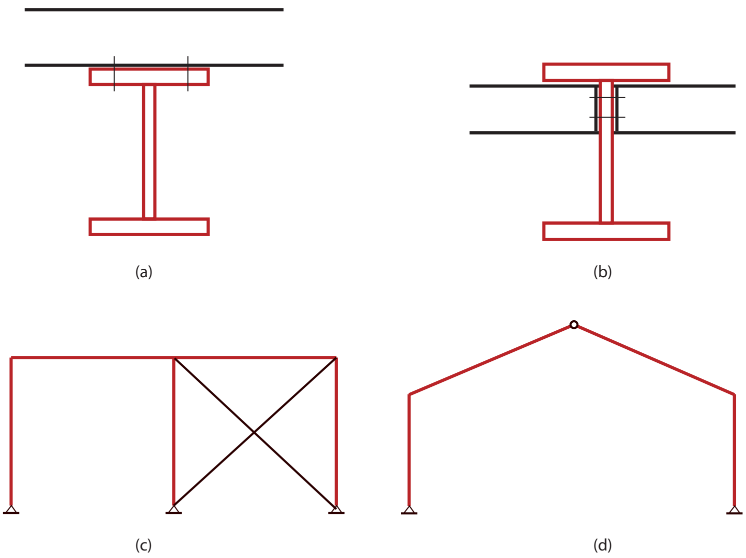


Figure C.1: (a) Typical connection of purlin and beam, load is transferred from roof to purlin to beam. (b) Connection of purlin only used as lateral torsional buckling support. (c) Braced stability system. (d) Three-hinged frame as alternative for stability in frame direction.

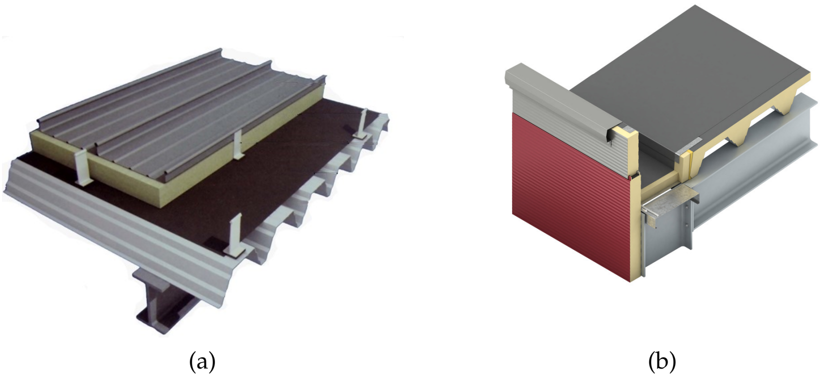


Figure C.2: (a) A standard roof with structural elements below, on top insulation material and steel or bitumen elements as water repellent layer (O'Donnell, 2018). (b) Sandwich panel as roof element, integrated insulation and structural system in one (Kingspan, 2018).

and install on site. The insulated roof plate is a corrugated steel sheet with on top an insulation and water proof bitumen layer. The separate structural layer allows for more flexibility in spans and higher loading conditions. The cold roof plate consist of corrugated steel panels. No insulation is added as these warehouses do not require heating.

PARAMETER	COMMENTS
Location: Netherlands	Dutch National Annex applies
Use class E	Area susceptible for accumulating of goods (NEN-EN 1991-1-1 NB, 2011, p.5)
Service Life class 3	50 years (NEN-EN 1990 NB, 2011, table NB.1-2.1)
Consequence class CC2	Moderate consequences of human life and high consequences in economic sense (NEN-EN 1990 NB, 2011, table NB.20–B1)

Table C.1: General parameters structural design of industrial warehouse

A roof slope should always be added to avoid water accumulation. NEN-EN 1990 NB (2011, p.15) specifies a minimum of 1.6 % slope assuming fixed connections. In situation with hinged connections the deformation will be larger and additional slope is needed. Additionally, instalment of emergency water drainage is required.

Very similar to the roof is the façade. It can also be constructed with sandwich panels, insulated façade plates or a cold façade plates. Additional steel is required between the columns to secure the panels.

C.2 ASSUMPTIONS

Structural design starts by assessment of assumptions. The made assumptions in the structural design is further elaborated.

C.2.1 *General*

The following list of codes are used in the structural design:

- NEN-EN 1990: Basis of structural design
- NEN-EN 1991: Actions on structures
- NEN-EN 1993: Design of steel structures

They provide as location the Netherlands with applicable national annex. Use class E is considered as warehouses are prone to accumulating of goods. This has consequences for the imposed floor loads. Service class 3 with a service life of 50 years is taken. Additionally consequence class 2 is assumed which is appropriate for a warehouse. Table C.1 provides a summary.

C.2.2 *Material*

For the structural elements steel quality S235 is considered. This is according to IFC data (Section E) and calculations (Section C.5) in which deformation proofed to be governing. Durability of the material is not further take into account.

C.2.3 *Connections*

It is assumed that the connectivity of the beams to the columns is hinged (in x, y and z direction), resulting in no rotational stiffness. This is easy to model, calculate and results in the most conservative outcome.

C.2.4 *Performance criteria*

The industrial warehouse is assessed in strength and stiffness. Both vertical and horizontal displacements are taken into account. According to NEN-EN 1990 NB (2011, Appendix A) the maximum allowed vertical roof deformation is *l/250*. This is due to long term permanent loading (w2) and imposed loads (w3). However, the Eurocode is not clear on the maximum allowable deformation with as reference point the support points. The old Dutch code NEN6702 provides a value of *0.004 * l* which is further used in this design. An overview is found in Figure C.3.

Global horizontal deformation needs to be below *h/150* (NEN-EN 1990 NB, 2011, Appendix A) in which *h* is the height of the warehouse. For local horizontal deformation by columns the same value is taken into account.

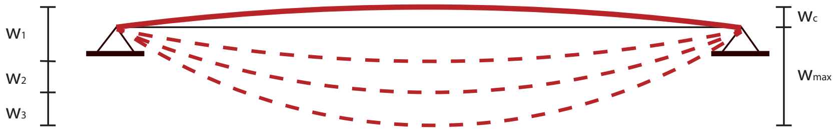


Figure C.3: Deformation diagram based on the Eurocode 1990 Appendix A. w1 provides the initial deformation due to self-weight and permanent load, w2 the deformation due to long term effects of permanent loading, w3 the deformation of imposed loads. w_c is the precamber of the element and w_{max} the maximum allowed deformation.

NAME	LOAD (kN/m^2)	COMMENTS
Bituminous layer	0.1	Separate put on top
Insulation	0.15	Accumulation on roof
Steel structure	0.2	Steel corrugated roof elements
Additional loads	0.1	Wires, lamps, pipes

Table C.2: Weight of roof with insulated steel roof, based on information from Kingspan (2018)

NAME	LOAD (kN/m^2)	COMMENTS
Sandwich panel	0.25	Corrugated steel elements with in between insulation
Additional loads	0.1	Wires, lamps, pipes

Table C.3: Weight of roof with insulated sandwich panels, based on information from Kingspan (2018)

Obtained stresses should be below the yield stress of the material. Not taken into account are specific criteria for the fire safety of the warehouse and the deformation due to non-uniform settlements of the foundation.

C.3 LOADS

Loading on the industrial warehouse can be divided in permanent loading (e.g. self-weight) and imposed loading (snow and wind load). Not taken into account is load by rain water, as subsequent measures are required in design, and loads by for example cranes.

C.3.1 *Permanent load*

The self-weight of the non-structural elements (roof and façade) is taken into account and applied as permanent load on the structure. As described in the geometry, two types of roof systems are used in practise. Both are assessed in self-weight with help of data by manufactures (Kingspan, 2018). An overview is given in Table C.2 and Table C.3.

Limitations in spans are present by using sandwich panels. Some manufactures are found that have large span panels available. Despite this, a range of portal frame distances can be used. This limitation is taken into account as heavier and thicker panels are needed with increased frame distances.

An average between the both roof systems is made resulting in a load of 0.45 kN/m^2. This is validated with the 'Bouwen met Staal' booklet on structural design of industrial warehouses (Bouwen met Staal, 2013) and references projects by the company Concretio. The self-weight of the façade is taken as 0.25 kN/m^2, considering a façade build-up with sandwich panels.

C.3.2 *Imposed load*

Imposed loads are divided in snow loads and wind loads.

SNOW Vertical snow load is elaborated in NEN-EN 1991-1-3 NB (2011). The following equation is used to calculate the snow load:

$$s = \mu_i * C_e * C_t * s_k \tag{C.1}$$

Where:

$$\begin{aligned} \mu_i &= \text{snow load shape coefficient} \\ C_e &= \text{exposure coefficient} \\ C_t &= \text{thermal coefficient} \\ s_k &= \text{characteristic value of snow load} \end{aligned}$$

For all regions in the Netherlands the characteristic snow load s_k is 0.7 kN/m^2, the exposure coefficient C_e 1.0, and the thermal coefficient C_t as well 1.0. The shape coefficient μ_i for the industrial warehouse can be found in applicable Eurocode for monopitch roofs. A roof slope of 0 % is taken as the actual slope is small in comparison with available options in the graph. A value of μ_i = 0.8 is found. This results in a snow load of 0.56 kN/m^2. This is independent of the parameters of the warehouse.

WIND The horizontal wind load (overview in Table C.4) is described in the Eurocode (NEN-EN 1991-1-4, 2011, p. 5.5) with the following equation:

$$F_w = c_s c_d * c_p * q_p(z_e) * A_{ref} \tag{C.2}$$

Where:

$$\begin{aligned} c_s c_d &= \text{structural factor} \\ c_p &= \text{pressure coefficient} \\ q_p(z_e) &= \text{peak velocity pressure at reference height } z_e \\ A_{ref} &= \text{reference area of structural element} \end{aligned}$$

The structural factor $c_s c_d$ takes into account the non-simultaneous occurrence of peak wind pressures on a surface (c_s) combined with effects of turbulence on the structure (c_d). The Eurocode provides a value of 1.0 for the combined factor $c_s c_d$ if the height h is smaller than 15 m (NEN-EN 1991-1-4, 2011, p.30). This is assumed in the structural design of the industrial warehouses.

The peak velocity pressure q_p depends on the reference height, the location of the building, and the terrain roughness. For this design it is save to assume that it will be build in an urban area. Furthermore, wind area II is chosen as location in the Netherlands. This is an intermediate zone between the coast and further inland, and covers large parts of the Netherlands. Lastly, the height z_e will be approximately between 3 and 15 meters. Given the above information, the National Annex provides a table to find the wind pressure (NEN-EN 1991-1-4 NB, 2011, Table NB.5): q_p between 0.58 kN/m^{2}and 0.80 kN/m^2. Linear interpolation is used between these values.

The pressure coefficient c_p can be described by an internal and external component for respectively the inter and outer parts of the structure. The factors have different values for locally loaded areas of 1 m^2 and larger areas of up to 10 m^2. This structural design will focus on the latter one. The value also depends on the ratio of *h/d* and the specific location in which it is assessed (Figure C.4). The external coefficients for façades are presented in Table NB.6-7.1 of the national annex (NEN-EN 1991-1-4 NB, 2011). It is assumed that the ratio of height and depth will always be lower than 1. This results for zone D in +0.8 and zone E in -0.5.

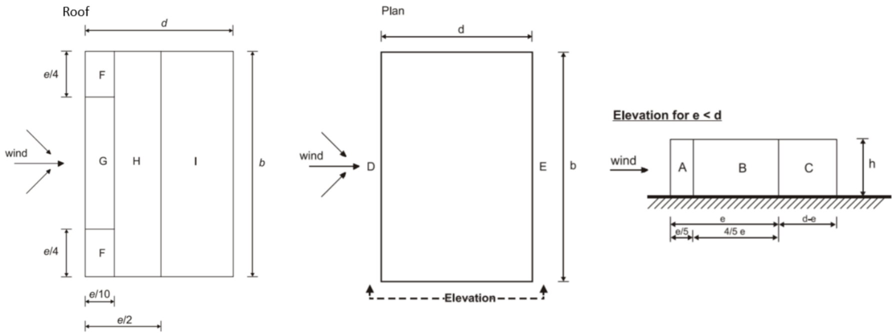


Figure C.4: Zone of wind loading different parts building (NEN-EN 1991-1-4 NB, 2011, p.38)

The external coefficients for the roof are spread in many zones. To simplify the process zone H is taken as average, considering a redistribution of forces and a sharp angled roof. The coefficient becomes -0.7. For a pitched roof these coefficients are lower and therefore are not further elaborated. The pitched roof is loaded the same as a flat roof.

The internal coefficient c_{pi} depends on the size and distribution of openings in the façade. An industrial warehouse has very few openings and with no specific design, it is difficult to assess the area of openings. The national annex prescribes that limit cases of c_{pi} need to be assessed resulting in values of -0.3 and 0.2. Load combinations will further indicate which is governing.

Friction forces due to wind are required to be taken into account as the surface area of industrial warehouses can become large. The following equation is used:

$$F_w = c_{fr} * q_p(z_e) * A_{ref} \tag{C.3}$$

Where the c_{fr} is the friction coefficient depending on the applied façade and roof material. It is assumed to be a smooth surface (steel of smooth concrete) resulting in a value of 0.01.

NAME	VALUE	COMMENTS
$c_s c_d$	1.0	h < 15 m
q_p	0.58 - 0.80 kN/m^2	Wind area II, urban area, 3 m < h < 15 m
c_{pe} façade	0.8 and -0.5	external, assumed h/b < 1
c_{pe} roof	-0.7	external, sharp angled roof, average on entire roof
c_{pi}	-0.3 and 0.2	internal, limit cases taken
c_{fr}	0.01	smooth surface (steel or concrete)

Table C.4: A summary of the coefficients regarding the wind forces

C.3.3 *Summary loads*

Table C.5 provides a summary of the permanent, snow and wind load on the industrial warehouse.

NAME	LOAD	COMMENTS
Permanent roof load	0.45 kN/m^2	Consisting of a sandwich panel
Snow load	0.56 kN/m^2	Accumulation on roof
Wind load	0.58 - 0.80 kN/m^2	Depending on height

Table C.5: Summary of loads

C.4 COMBINATIONS

The 'Bouwen met Staal' publication on industrial warehouses describes the required loading combination to be assessed (Bouwen met Staal, 2013). These can be split in SLS and Ultimate Limit State (ULS) situations. Both are retrieved according to the Eurocode (NEN-EN 1990 NB, 2011, Table NB.4-A1.2(B)). Combination factors ψ_0 for both snow and wind are 0, meaning they do not combine. An overview of the four possible loads are displayed in Figure C.5. Combinations of these are made in SLS and in ULS.

C.4.1 *SLS combinations*

The SLS is to check deformation of the roof and façade. The following combinations are made:

- $1.0 * G + 1.0 * Q_s$
  Combination of permanent roof load and snow load: (a).
- $1.0 * G + 1.0 * Q_w$
  Combination of permanent and wind load with under pressure: (a) + (b) + (c).

C.4.2 *ULS combinations*

The ULS is to check the stresses in the elements. The following combinations are made for the industrial warehouse:

- $1.2 * G + 1.5 * Q_s$
  Combines permanent and snow load: (a).
- $1.2 * G + 1.5 * Q_w$
  Combination of permanent and wind load with under pressure: (a) + (b) + (c).

- $1.35 * G$
  Only permanent load is assessed: (a).
- $0.9 * G + 1.5 * Q_w$
  This combines permanent roof load with upward wind suction and over pressure in the warehouse: (a) + (d).

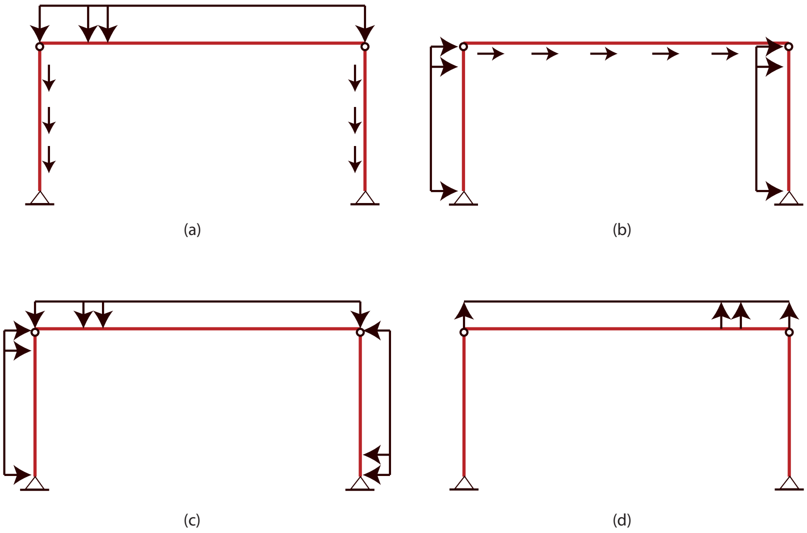


Figure C.5: The assessed load combinations in the structural design. (a) Permanent load of roof and façade elements combined with snow load. (b) Horizontal wind load with friction forces on roof. (c) Horizontal wind with under pressure in warehouse. (d) Wind with over pressure in warehouse.

C.5 CALCULATIONS

Preliminary calculations provide guidance and validation on the created Grasshopper model described in Chapter 4. Validation is done with hand calculations and the design booklet by 'Bouwen met Staal'.

To serve as example, a warehouse with length, width and height of respectively 28, 15, and 8 meters is assessed. It is build-up with beams and columns with a 7 meter portal frame distance, located in wind zone II, and with hinged connections between beam and column.

C.5.1 *'Bouwen met Staal' design booklet*

The design booklet on industrial warehouses provided by 'Bouwen met Staal', generates a quick overview on required sections of beams and columns. Assumptions are made on the assessed load combinations. No self-weight and permanent roof loading is applied in the SLS load combinations as it's assumed that the precamber of the beams will take this into account. Furthermore, in the ULS conditions a load factor of 1.3 is used for the imposed loads instead of the now required 1.5. This shows that age of the document (2007) is relevant.

The design takes into account the wind area, height, type of connection, portal frame distance and the span. Outcomes are presented in a design graph with on the x-axis the span and on the y-axis the weight in kg/m^2 (Figure C.6).

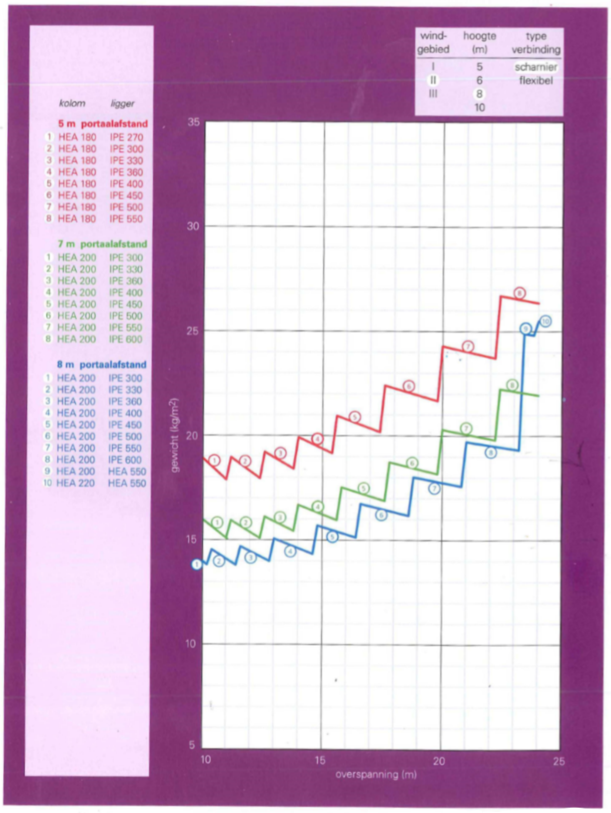


Figure C.6: Graph of the design booklet on industrial warehouses

Results for the typical example are an HEA200 as column and an IPE400 as beam. This would serve as a starting point for the conceptual design.

C.5.2 *Hand calculation*

A quick hand calculation is performed with the given load combinations in Section C.4. Appendix D provides an extensive look on the calculations. Additional assumptions are that initial deformation by self-weight are resolved by a precamber of the member. This means that in the simple calculations the self-weight is not taken into account.

Outcome of the example are an HEA180 as column and an IPE500 as beam. It can be concluded that the SLS combinations of permanent and snow for the beam and permanent and wind for the column is governing. Furthermore, permanent load by façade panels is not relevant for the choice of sections. The friction force on the structure is negligible small (1 %) and can be discarded in following calculations. A quick buckling check on the column is performed. It reveals enough overcapacity (factor of 8) to deal with imperfections, additional loads (horizontal and moments) and second order effects.

Comparing results with the above Section on the design booklet of 'Bouwen met Staal', reveals a much larger beam and a smaller column. In the hand calculation it's chosen to include the permanent roof load in contrary of what the booklet does. Data on build warehouses (Appendix E) showed governing deformation in which this is also simulated. Larger column sections in the booklet has to do with additional capacity regarding imperfections.

D

APPENDIX: CALCULATIONS

D.1 ASSESSMENT OF WIND IN X AND Y DIRECTION

Wind load on the warehouse is an important factor for the selection of columns in the structural optimisation. In automation of the data generation, it is key to only assess the most critical wind direction. Validation is needed. Figure D.1 shows the wind in x and y direction on a warehouse with length 21 m, width 60 m and height 10 m. This is an extreme case in which width is much larger than length.

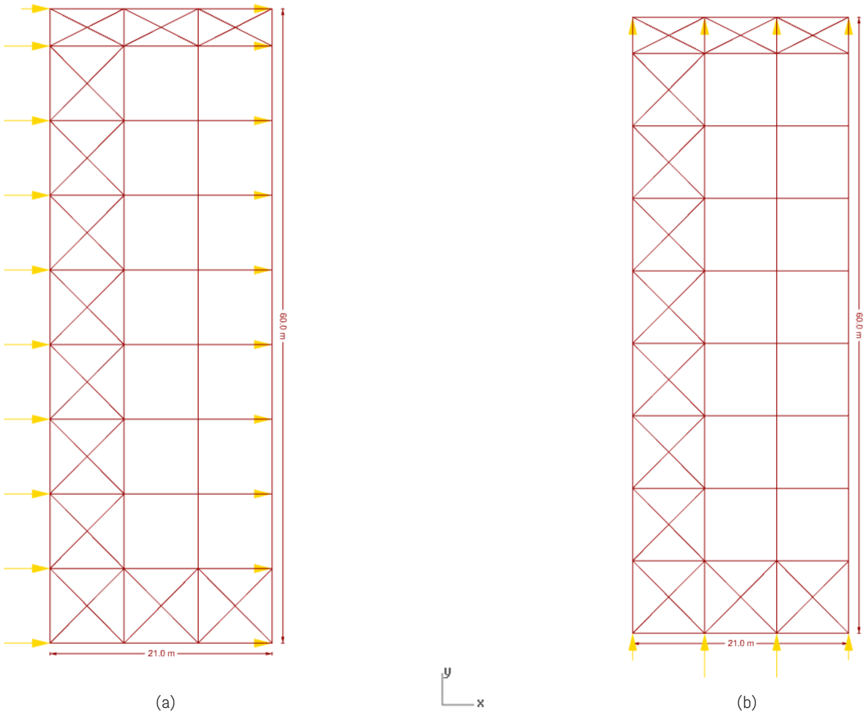


Figure D.1: (a) wind in x direction on the warehouse. (b) Wind in y direction on the warehouse

Table D.1 provides the results of both wind in x and wind of y direction. Both beams and purlins are found to be identical. Columns are slightly larger in the x direction which is expected due to a larger surface area. However, differences are small and considering the small length/width ratio it can be concluded that assessment of only wind in y direction is adequate.

PARAMETERS	WIND X	WIND Y
Columns	HEA280	HEA260
Beams	RHCS260x180x16	RHCS260x180x16
Purlins	RHSC150x100x10	RHSC150x100x10
Braces	CHSC114x6	CHSC89x5

Table D.1: Section selection due to wind in x and y direction

D.2 HAND CALCULATION SHEET

Hand calculation portal frame

May 14, 2018

1 Hand calculations

Calculations for industrial warehouse based on assumptions provided in chapter 3. To assess and verify the Grasshopper scripting and structural analysis with Karamba.

1.1 Parameters to input

```
In [1]: l = 15 # m
        b = 28 # m
        h = 8 # m
        portal_frame_distance = 7 # m
        fy = 235 # N/mm2
        E = 210*10**6 # kN/m2
        w_max = l/250 # max vertical deformation in m
        u_max = h/150 # max horizontal deformation in m

        print("Maximum vertical deformation can be: {} m".format(w_max))
        print("Maximum horizontal deformation can be: {} m".format(u_max))
```

```
Maximum vertical deformation can be: 0.06 m
Maximum horizontal deformation can be: 0.05333333333333334 m
```

```
In [2]: # loads
        q_perm_roof = 0.45 # kN/m2
        q_perm_facade = 0.25 # kN/m2
        q_structure = 0.20 # kN/m2

        q_snow = 0.56 # kN/m2

        q_wind_max = 0.80         # 15 meter
        q_wind_min = 0.58         # 3 meter

        C_e_facade_pressure = 0.8
        C_e_facade_suction = -0.5
        C_e_roof = -0.7
        C_i_max = 0.2
        C_i_min = -0.3

        # calculate wind load
        q_wind = ((q_wind_max - q_wind_min)/(15-3))*(h-3) + q_wind_min
        print("The wind load is {} kN/m2".format(q_wind))

        # friction
        c_fr = 0.01
        q_fr = q_wind * c_fr
```

```
The wind load is 0.6716666666666666 kN/m2
```

1.2 Data of steel sections

```
In [3]: import pandas as pd

        data = pd.read_csv(r'C:\Users\Niels\OneDrive\TU Delft\Graduation\Grasshopper\Structural
```

```
In [4]: data
```

```
Out[4]:       name    Iy (cm4)  Wy (cm3)
        0    HEA100     349.20     72.76
        1    HEA120     606.20    106.30
        2    HEA140    1033.00    155.40
        3    HEA160    1673.00    220.10
        4    HEA180    2510.00    293.60
        5    HEA200    3692.00    388.60
        6    HEA220    5410.00    515.20
        7    HEA240    7763.00    675.10
        8    HEA260   10450.00    836.40
        9    HEA280   13670.00   1013.00
        10   HEA300   18260.00   1260.00
        11   HEA320   22930.00   1479.00
        12   HEA340   27690.00   1678.00
        13   HEA360   33090.00   1891.00
        14   HEA400   45070.00   2311.00
        15   HEA450   63720.00   2896.00
        16   HEA500   86970.00   3550.00
        17   HEA550  111900.00   4146.00
        18   HEA600  141200.00   4787.00
        19   HEA650  175200.00   5474.00
        20   HEA700  215300.00   6241.00
        21   HEA800  303400.00   7682.00
        22   HEA900  422100.00   9485.00
        23  HEA1000  553800.00  11190.00
        24      NaN        NaN       NaN
        25    IPE80      80.14     20.03
        26   IPE100     171.00     34.20
```

```
27   IPE120    317.80    52.96
28   IPE140    541.20    77.32
29   IPE160    869.30   108.70
30   IPE180   1317.00   146.30
31   IPE200   1943.00   194.30
32   IPE220   2772.00   252.00
33   IPE240   3892.00   324.30
34   IPE270   5790.00   428.90
35   IPE300   8356.00   557.10
36   IPE330  11770.00   713.10
37   IPE360  16270.00   903.60
38   IPE400  23130.00  1156.00
39   IPE450  33740.00  1500.00
40   IPE500  48200.00  1928.00
41   IPE550  67120.00  2441.00
42   IPE600  92080.00  3069.00
```

1.3 Calculations main beam and column

1.3.1 SLS calculation

```
In [5]: # SLS snow + permanent

        q = (q_perm_roof + q_snow) * portal_frame_distance

        I_beam_max = 5*q*l**4/(384*w_max*E)*10**8

        print ("load q={} kN/m2".format(q))
        print("beam stiffness requirement I: {} cm4".format(I_beam_max))

        # IPE500 needs to be used
```

```
load q=7.07 kN/m2
beam stiffness requirement I: 36987.3046875 cm4
```

Note: Wind creates upward wind pressure on roof, however not taken into account as it creates less loading on the main beam

```
In [6]: # SLS permanent + wind horizontal and under pressure

        q = (q_perm_roof + q_wind * (C_i_max)) * portal_frame_distance

        I = 5*q*l**4/(384*w_max*E)*10**8

        print ("load on beam q = {} kN/m2".format(q))
        print("beam stiffness requirement I: {} cm4".format(I))
        # IPE330 needs to be used



        # columns
        q_facade = (q_wind * (C_e_facade_pressure + C_i_max)) * portal_frame_distance

        I_col = 5*q_facade*h**4/(384*u_max*E)*10**8

        percentage_friction = q_fr * portal_frame_distance/q_facade

        print ("load facade q={} kN/m2".format(q_facade))
        print("column stiffness I: {} cm4".format(I_col))
        print("percentage of friction of total load: {} %".format(percentage_friction*100))

        # HEA180 needs to be used
```

```
load on beam q = 4.090333333333334 kN/m2
beam stiffness requirement I: 21398.92578125 cm4
load facade q=4.701666666666666 kN/m2
column stiffness I: 2238.888888888889 cm4
percentage of friction of total load: 1.0 %
```

1.3.2 ULS calculations

```
In [7]: # ULS snow + permanent

        q_d = (1.2 * (q_perm_roof) + 1.5 * q_snow) * portal_frame_distance
        M_d = q_d*l**2/8
        W_d = M_d*10**6/(fy*1*10**3)

        print ("Design load q={} kN/m2 gives a moment {} kNm".format(q_d, M_d))
        print("beam strength requirement W: {} cm3".format(W_d))

        # IPE400 needs to be used


        # column
        load_facade = q_perm_facade * h * portal_frame_distance
        load_pb = q_d*l/2
        percentage_facade = load_facade/load_pb

        N_facade_max = load_facade + load_pb
        A = N_facade_max/fy*1000
        print("area column: {} mm2".format(A))
        print("percentage of facade load of total load: {} %".format(percentage_facade*100))
        # HEA100
```

```
Design load q=9.66 kN/m2 gives a moment 271.6875 kNm
beam strength requirement W: 1156.1170212765958 cm3
area column: 367.8723404255319 mm2
percentage of facade load of total load: 19.32367149758454 %
```

```
In [8]: # ULS permanent + wind

        q_d = ((1.2 * (q_perm_roof) + 1.5 * q_fr + 1.5 * q_wind * (C_i_max))
               * portal_frame_distance)
        M_d = q_d*l**2/8
        W_d = M_d*10**6/(fy*1*10**3)

        print ("Design load q={} kN/m2 gives a moment {} kNm".format(q_d, M_d))
        print("beam strength requirement W: {} cm3".format(W_d))

        # IPE330 needs to be used


        # columns
        q_facade = (1.5 * q_wind * (C_e_facade_pressure + C_i_max)) * portal_frame_distance

        M_col = q_facade*h**2/8
        W_col_max = M_col*10**6/(fy*1*10**3)

        print("horizontal load q on column is: {} cm3".format(q_facade))
        print("column strength W: {} cm3".format(W_col_max))

        # HEA180 needs to be used
```

```
Design load q=5.261025000000001 kN/m2 gives a moment 147.96632812500002 kNm
beam strength requirement W: 629.6439494680852 cm3
horizontal load q on column is: 7.052499999999998 cm3
column strength W: 240.08510638297867 cm3
```

```
In [9]: # ULS permanent only

        q_d = (1.35 * (q_perm_roof)) * portal_frame_distance
        M_d = q_d*l**2/8
        W_d = M_d*10**6/(fy*1*10**3)

        print ("Design load q={} kN/m2 gives a moment {} kNm".format(q_d, M_d))
        print("beam strength requirement W: {} cm3".format(W_d))

        # IPE300 needs to be used
```

```
         # column
         N_facade = q_perm_facade * h * portal_frame_distance + q_d*l/2
         A = N_facade/fy*1000
         print("area column: {} mm2".format(A))

         # HEA100 needs to be used
```

```
Design load q=4.2525 kN/m2 gives a moment 119.60156250000001 kNm
beam strength requirement W: 508.9428191489362 cm3
area column: 195.2925531914894 mm2
```

```
In [10]: # ULS permanent + wind upwards (over pressure)

         q_d = ((0.90 * (q_perm_roof ) + 1.5 * q_wind * (C_e_roof + C_i_min))
                * portal_frame_distance)
         M_d = q_d*l**2/8
         W_d = M_d*10**6/(fy*1*10**3)

         print ("Design load q={} kN/m2 gives a moment {} kNm".format(q_d, M_d))
         print("beam strength requirement W: {} cm3".format(-W_d))

         # IPE300 needs to be used


         # column
         N_facade = q_perm_facade * h * portal_frame_distance + q_d*l/2
         A = N_facade/fy*1000
         print("area column: {} mm2".format(-A))

         # HEA100
```

```
Design load q=-4.2174999999999985 kN/m2 gives a moment -118.61718749999996 kNm
beam strength requirement W: 504.75398936170194 cm3
area column: 75.02659574468079 mm2
```

1.4 Buckling check

```
In [11]: import numpy as np
```

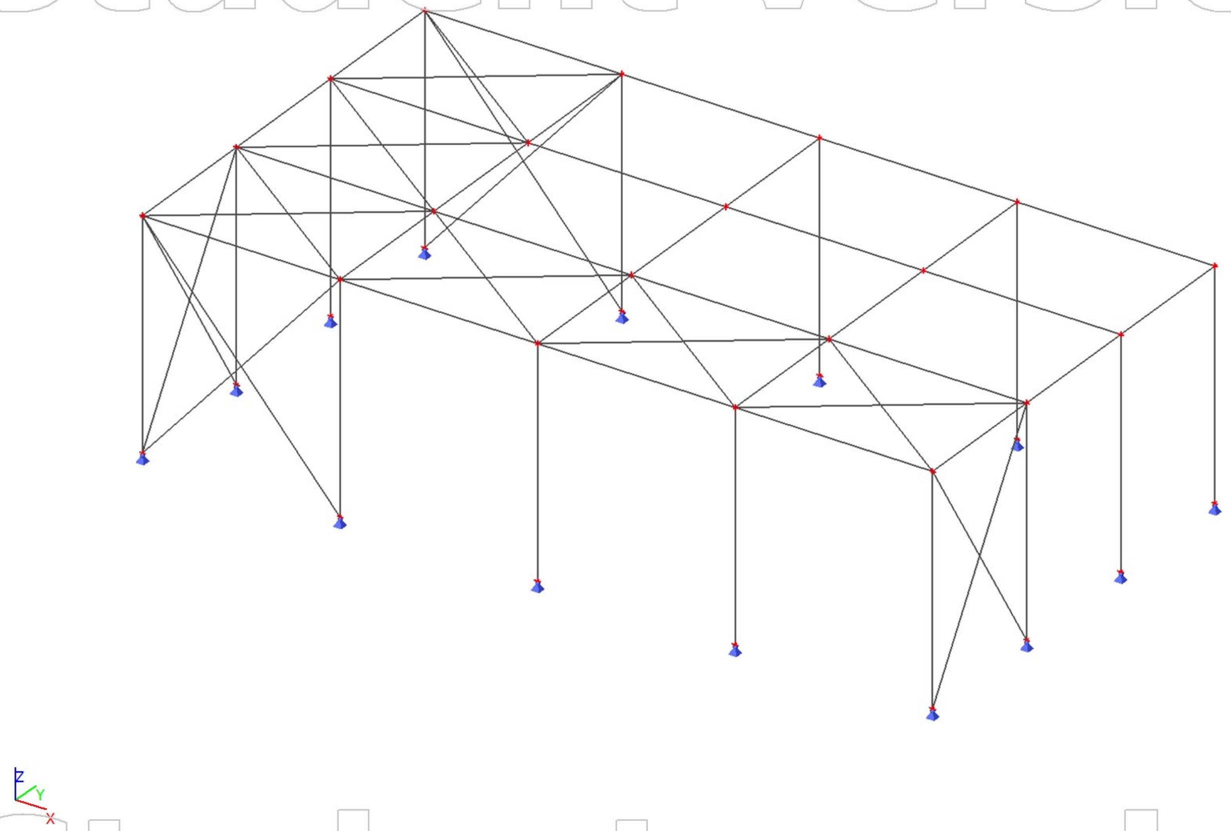
```
In [12]: print (W_col_max)
         # HEA180

         I = 2510*10**-8
         F_k = np.pi**2*E*I/(1*h)**2
         print("Euler max buckling force is: {} kN".format(F_k))
         print("Max found normal force: {} kN".format(N_facade_max))
```

```
240.08510638297867
Euler max buckling force is: 812.8544499709689 kN
Max found normal force: 86.45 kN
```

Note: The factor of buckling of 10 is oke for this first calculation. Imperfections and combination of loads will reduce the buckling capacity of the column significantly.

D.3 SCIA MODEL

1. Analysis model



2. Members

Name	Cross-section	Material	Length [m]	Beg. node	End node	Type
B1	CS1 - IPE500	S 235	5.000	N1	N2	beam (80)
B16	CS2 - O (80; 3; 100; 3)	S 235	7.000	N1	N5	beam (80)
B32	CS3 - Tube (89; 5)	S 235	8.602	N2	N21	beam (80)
B65	CS13 - HEA180	S 235	8.000	N20	N54	column (100)

3. Load cases

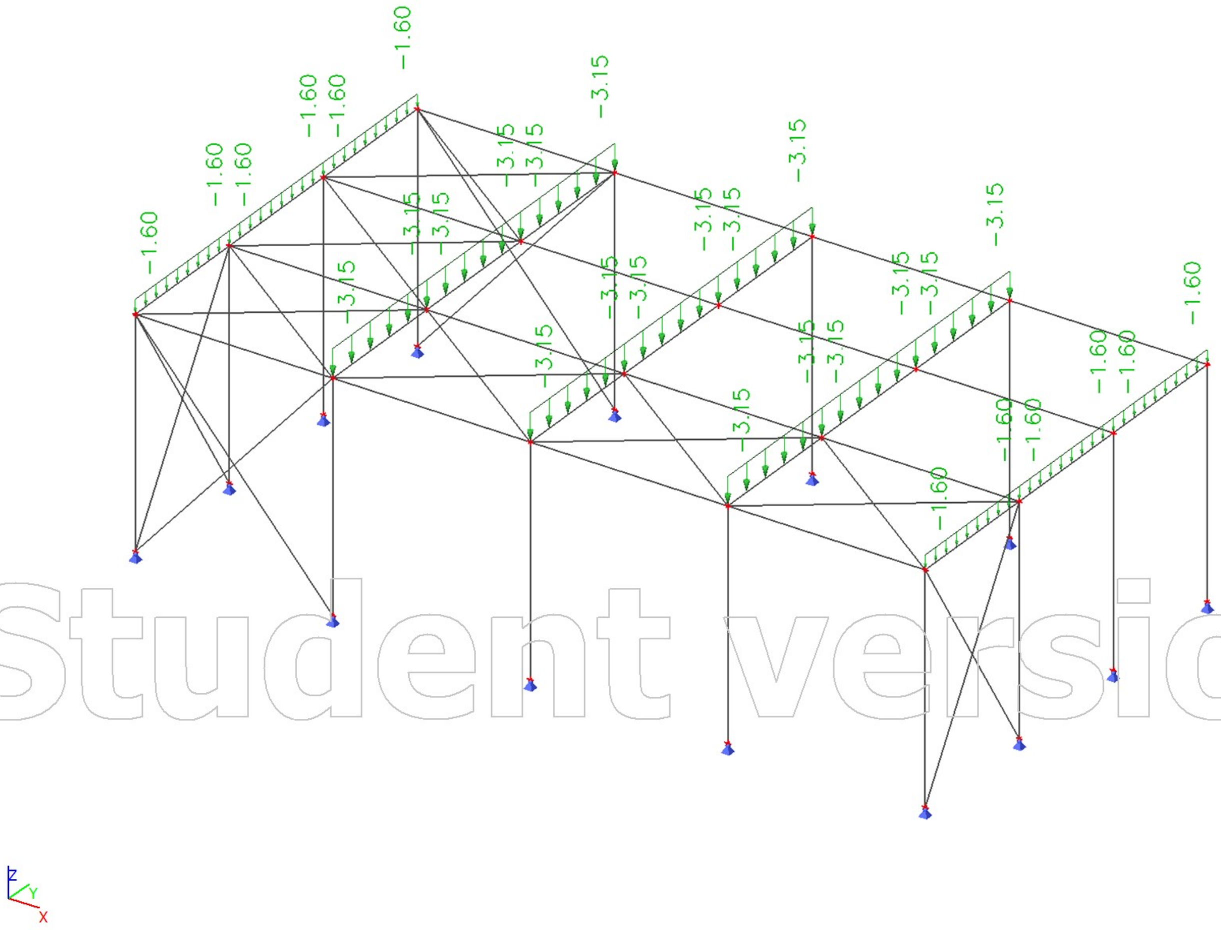
Name	Description	Action type	LoadGroup	Direction	Duration	Master load case
	Spec	Load type				
LC1		Permanent	LG1	-Z		
		Self weight				
LC2	PB	Permanent	LG1			
		Standard				
LC3	snow	Variable	LG2		Short	None
	Standard	Static				
LC4	wind	Variable	LG2		Short	None
	Standard	Static				

4. Hinges

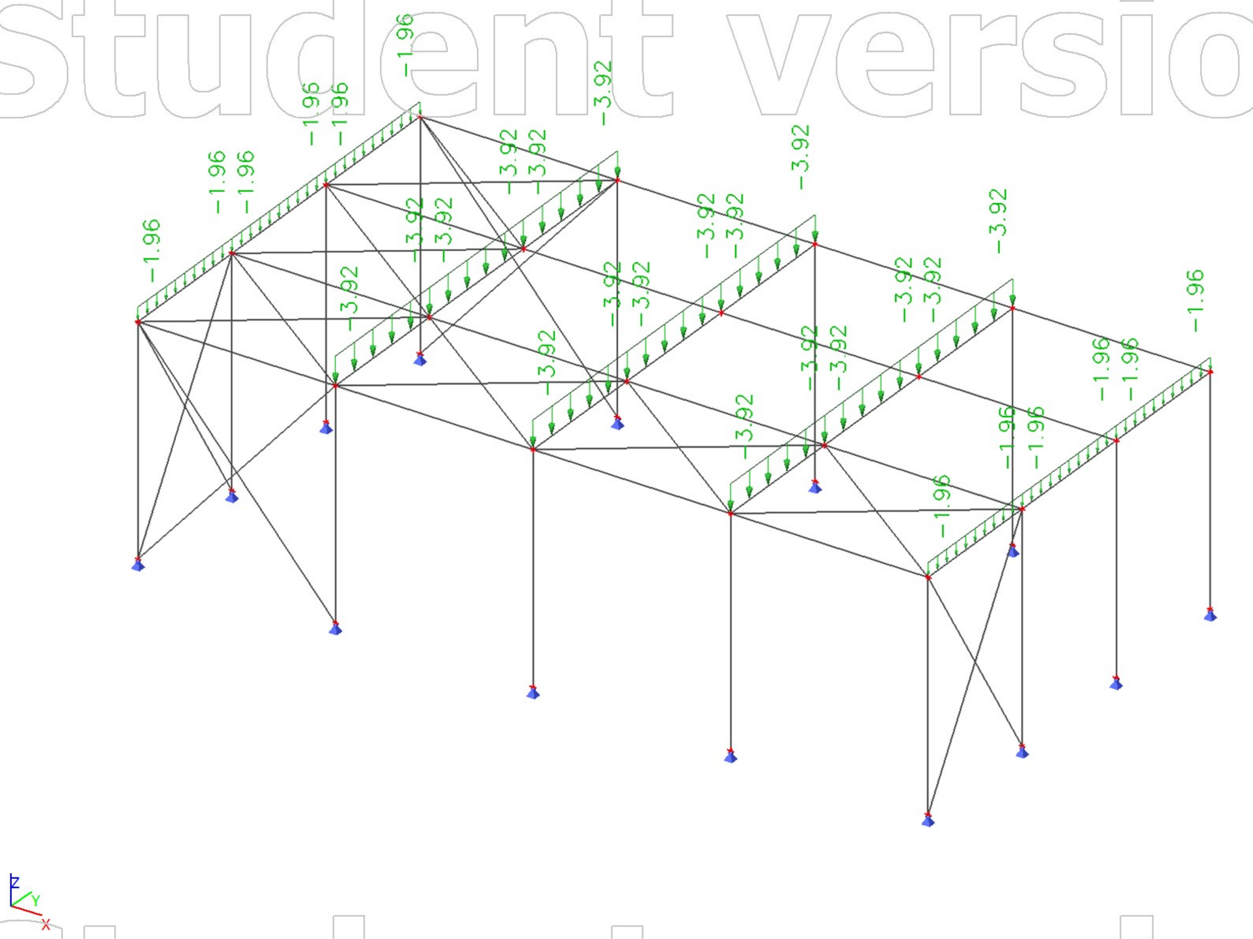
Name	Member	Position	ux	uy	uz	fix	fiy	fiz
H1	B44	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H2	B45	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H3	B46	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H4	B47	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H5	B48	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H6	B49	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H7	B50	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H8	B51	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H9	B52	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H10	B53	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H11	B54	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H12	B55	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free

Name	Member	Position	ux	uy	uz	fix	fiy	fiz
H13	B56	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H14	B57	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H15	B58	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H16	B59	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H17	B60	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H18	B61	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H19	B62	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H20	B63	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H21	B64	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free
H22	B65	Begin	Rigid	Rigid	Rigid	Rigid	Free	Free

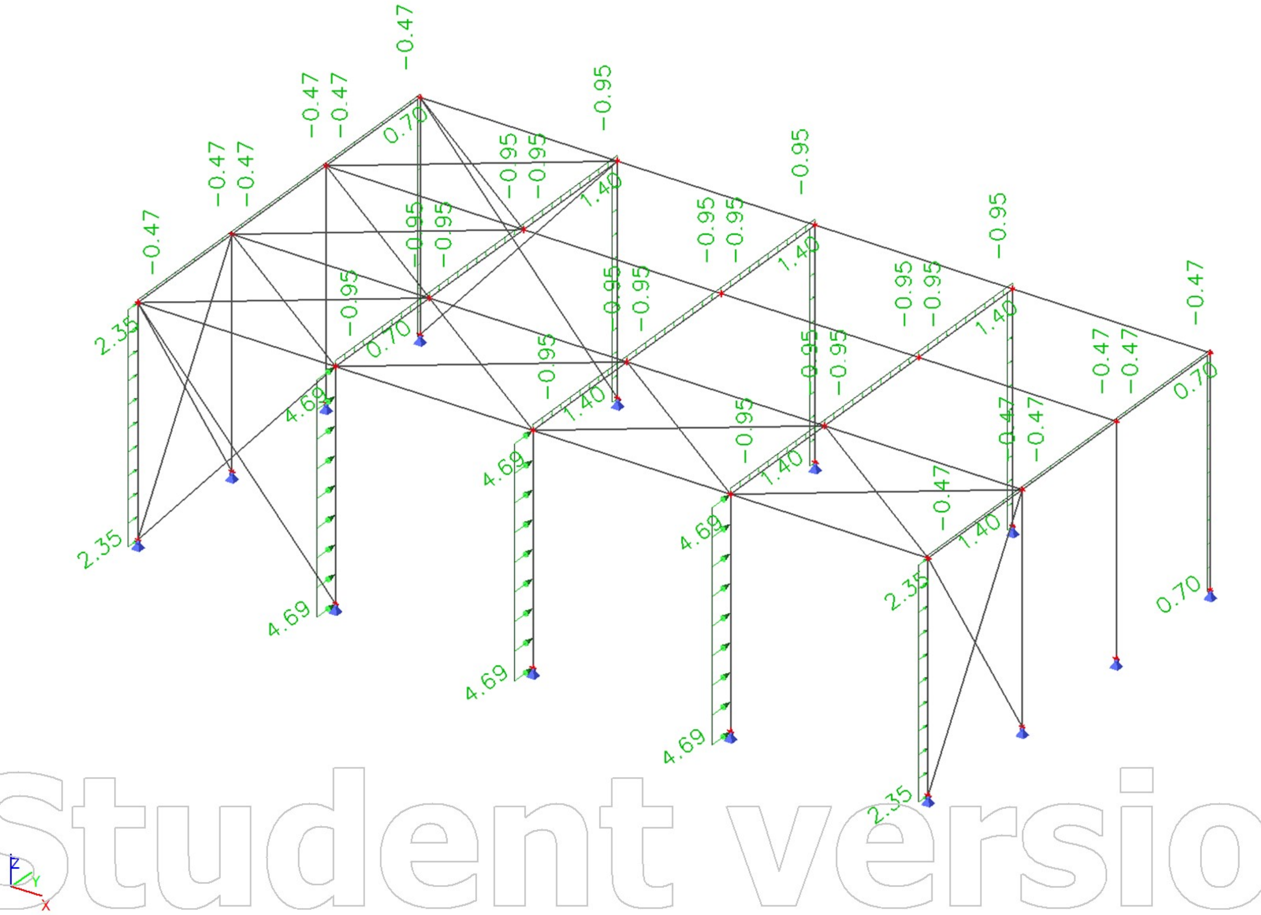
5. PB load



6. Snow load

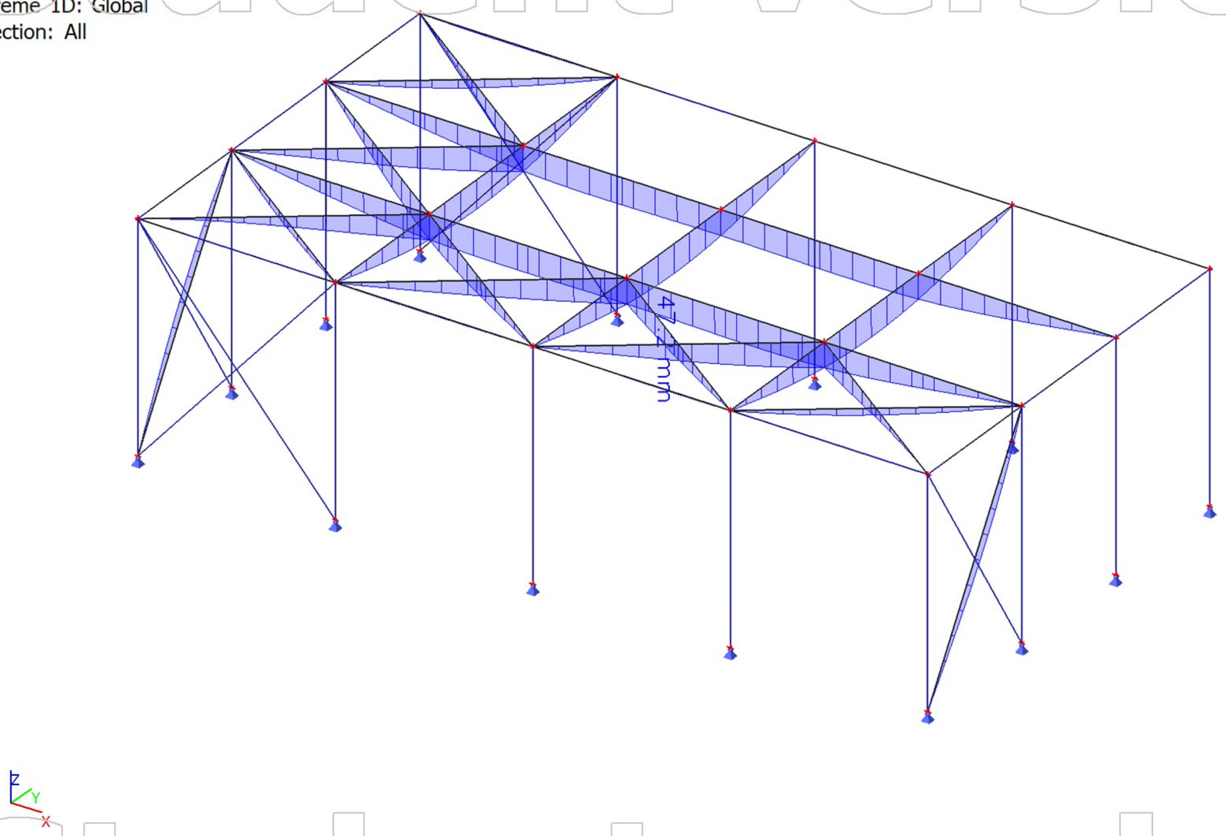


7. Wind load



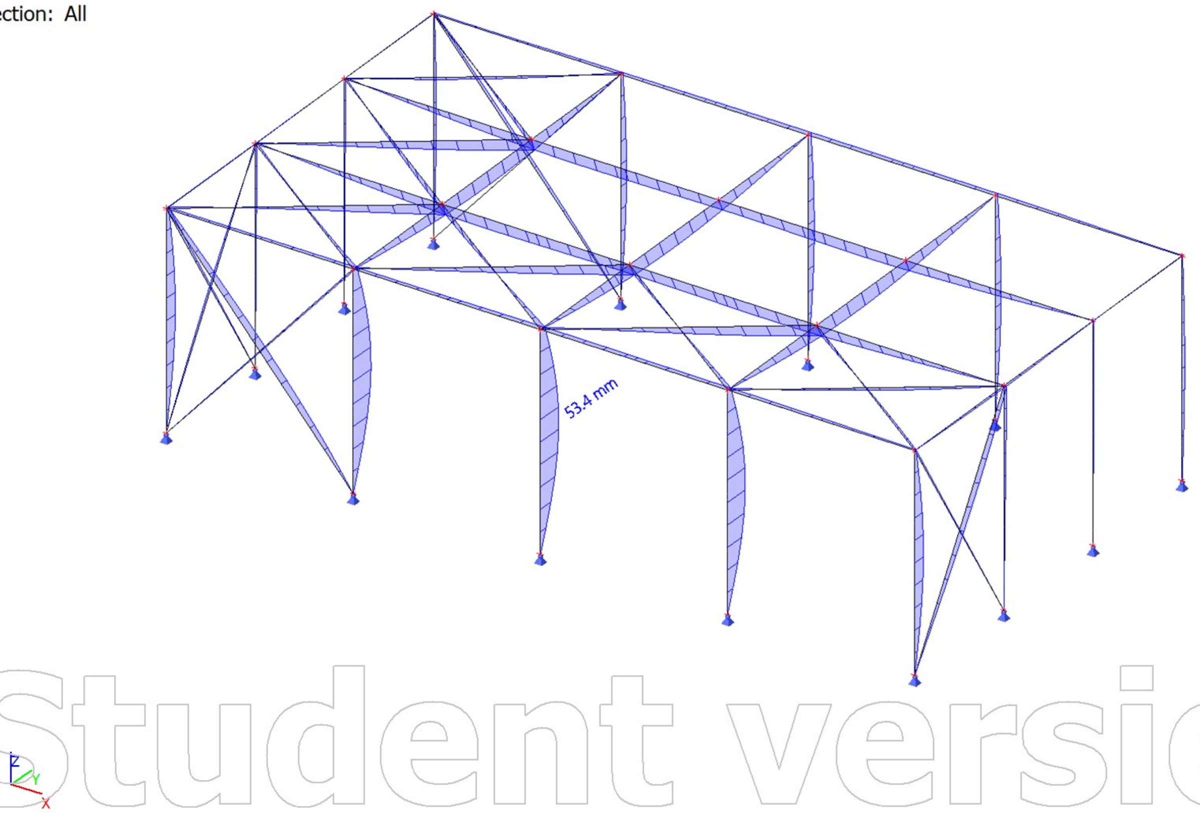
8. Deformation PB + snow

Values: **U_{total}**
Linear calculation
Combination: CO1
Coordinate system: Global
Extreme 1D: Global
Selection: All



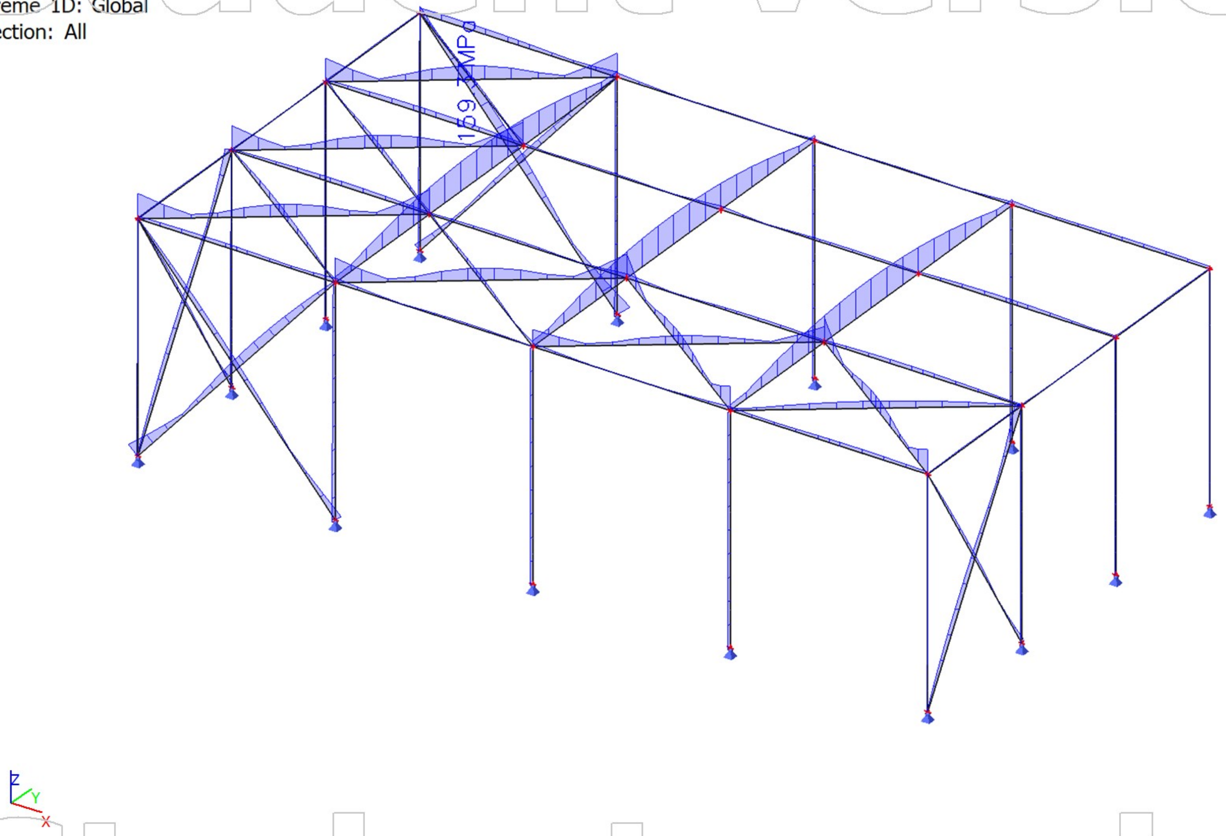
9. Deformation wind

Values: **U_{total}**
Linear calculation
Combination: CO2
Coordinate system: Global
Extreme 1D: Global
Selection: All



10. Stresses

Values: σ_E
Linear calculation
Combination: CO3
Coordinate system: Principal
Extreme 1D: Global
Selection: All



E

APPENDIX: IFC FILES INDUSTRIAL WAREHOUSES

A dataset of 25 industrial warehouses is provided by the companies Voortman Steelgroup, GB Steelgroup, and ASK Romein. Analysis of the data resulted in 8 files being not usable due to incomparability between generated designs, errors in data format and data incompleteness. Remaining warehouses are still difficult to compare directly against the generated data by Grasshopper. This due to simplifications in the parametric script. General remarks and comparison with the data is provided below.

E.1 REMARKS

General descriptions and remarks on the IFC data is presented below.

- *Specific function:* Immediately clear is the wide variety in designs. Additional floors for offices are visible, multiple different halls with each their own structural system, pitched roofs, different shapes, different span types. All designed specifically for a function and the client needs.

- *Size:* Most of the warehouses are very large in size, up to 440 m in length and 230 m in width is seen. However, at that size the structural systems are separated in blocks of roughly 100x100 m to prevent issues with e.g. temperatures.

- *Load transfer:* Almost all warehouses transfer the loads directly from the beams and trusses, through the columns, to the foundation. This result in very small purlins which are only used to prevent the beam or truss from buckling. These are so called 'drukkers' and are almost always constructed in Rectangular Hollow Sections (RHS). Portal distances range from 5 to 7 m. Load transfer by purlins is in some cases implemented. Portal distances of 10 to 12 m are found with IPE or HEA sections for the purlins.

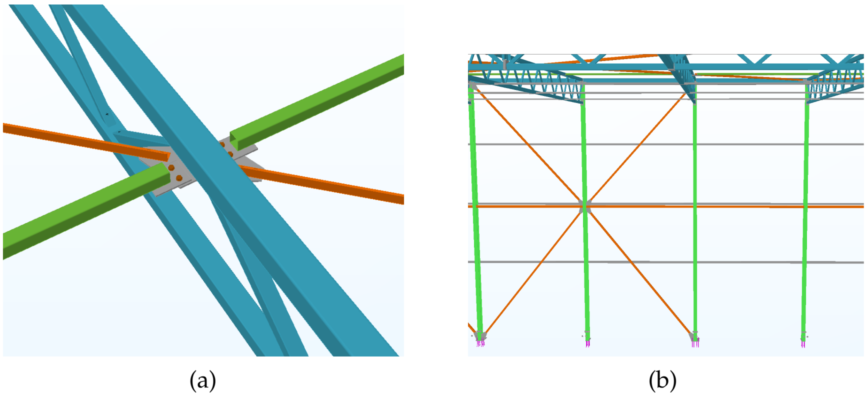
- *Stability:* Up to 20 m in height is recorded in data. The stability of such a warehouse is very elaborate with in some cases entire roofs covered with braces. These are constructed in a square grid (optimal in force distribution) and applied at the outer edges of the roof. With increased lengths, additional braces are placed in the middle. Application at the outer edges reduces compression in the beam members when horizontal wind forces are present. The façade bracing is almost always below the braces of the roof, although functionality wise (e.g. door placements) this can change at certain locations. The braces are often constructed in L shape profiles. These are very efficient in transferring tensile forces compared to their weight. The shape also prohibits deflections due to self-weight.

- *Trusses:* From spans between 25 m and 60 m trusses are used. Often the trusses are only made out of diagonals (Warren truss) and supported in lateral direction on both upper and lower part by 'drukkers'. A mix of profiles are used often with different sections for upper, lower and diagonal members. Most seen is the RHS ranging from KK140 to KK200. A rather interesting other combination is seen with as upper chord an HEA section and as lower chord a 90 degrees turned IPE profile. The diagonals are then welded to the web of the IPE section.

- *Columns:* The façade columns are typically constructed in HEA sections but sometimes IPE or KK. Interesting to mention is that often the columns are laterally supported by 2 to 3 additional steel elements ('drukkers') preventing buckling in the weak axis. The mid columns are often larger than the façade columns. Purlin sections at those locations are very large as only a limited amount of mid columns are placed (not every portal frame have one). Loads are therefore first transferred to purlin and then to mid column.

- *Combination truss and mid column:* The combination of a span with trusses and extra mid columns is seen in about half of the warehouses. It's an efficient way to span large widths.

In Figure E.1 two details of industrial warehouses are given.



(a)

(b)

Figure E.1: (a) Typical connection of a purlin, wind brace and a truss or beam member. Also visible is the use of L sections and RHS. (b) The façade of warehouse showing the braces and other stabilising façade elements.

PARAMETERS	1	2	3
length	44	78	55
width	35	18.5	25
height	14	14	7
por.fr.dis.	5.5	5.2	5
pur.dist	5.5	6	5
beam	KK180x5	IPE450	KK160x6
beam GH	RHS200x120x12	IPE550	RHC180x100x10
column	HEA260	HEA260	HEA180
column GH	HEA260	HEA330	HEA140
h truss	1.5	N/A	1
h truss GH	1.75	N/A	1.25

Table E.1: Three IFC's compared with Grasshopper solution

E.2 COMPARISON GRASSHOPPER

Out of the 17 designs it is challenging to find comparable design within the limits of the Grasshopper scripts. A total of 6 designs are provided in Table E.1 and Table E.2. It presents an overview of chosen beam and column sections by the structural optimisation and sections in the IFC dataset. This two types are found to be most significant in determining the sustainability score (See Table 4.7). The warehouses are given in picture (Figure E.2) to underline the difference between the conceptual design implementation in Grasshopper and real-world data. Comparable is used loosely here, as the IFC data is rather different.

Warehouse 1 is cut from the IFC file as it contains multiple heights and variations in span. This assumption has influence on the found sections as a certain degree of stability is provided by the entire structure. However, as Table E.1 shows the found sections for truss and column are very comparable. Warehouse 2 on the other hand, shows the limitations of the Grasshopper script. Both the beam and column are significantly larger. This is caused by the combination of height and large length, but also by the disregarded other 3/4 of the warehouse in the IFC model. Similar results are found with warehouse 5. The combination of a large length results in unrealistic column sections.

Warehouse 4 has a pitched roof and mid columns. The first is not taken into account in the Grasshopper model. Comparable beam sizes are found but column section is much smaller. No clear reason, apart from the pitched roof, is found. Warehouse 6 on the other hand shows

PARAMETERS	4	5	6
length	55	97	37
width	50	150	14
height	11	9	15
por.fr.dis.	4	6	5
pur.dist	5.5	6	5
beam	IPE330	IPE450	IPE500
beam GH	IPE360	IPE450	IPE400
column	IPE330	KK400x8	HEA360
column GH	HEA220	HEA240	HEA360
max.span	16	18	N/A

Table E.2: Three IFC's compared with Grasshopper solution

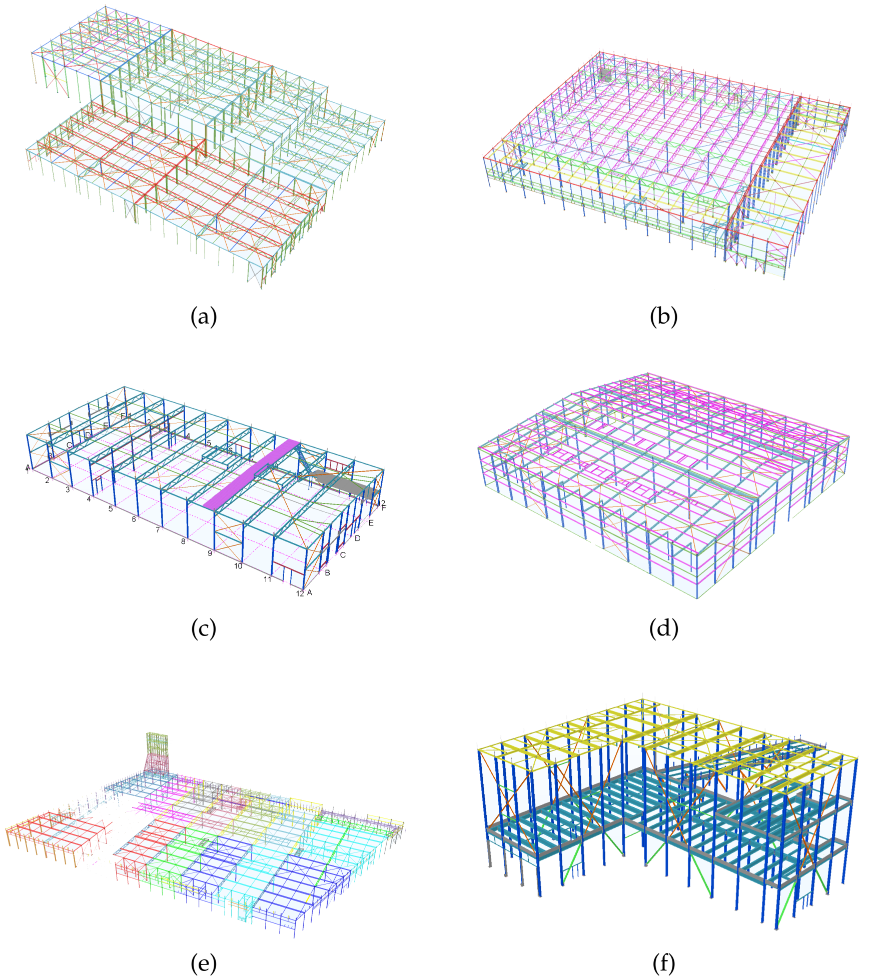
PARAMETERS	MIN	MAX	AVG
length	37	180	89
width	14	150	64
height	7	20	12
por.fr.dis.	4	12	6.3
pur.dist	4	12	5.7
max span	16	32	25
l/w ratio	0.65	4.2	1.8

Table E.3: The minimum, maximum and average values of the given parameters for the 17 warehouses

very comparable section although the shape is not compatible with the script. Only a small part of the IFC file is taken for this comparison.

E.3 PARAMETER VARIATION

The IFC data is also used to create a more realistic generated dataset. This is done by deleting data points which do not comply to the set rules (Section 5.3). In Table E.3 the parameters are given in minimum, maximum and average values.



(a) (b) (c) (d) (e) (f)

Figure E.2: The following warehouses are linked to the tables E.1 and E.2. (a) Warehouse 1 with trusses. (b) Warehouse 2 with beams. (c) Warehouse with trusses. (d) Warehouse 4 with beams and mid columns. (e) Warehouse with mid columns and trusses. (f) Warehouse 6 with beams