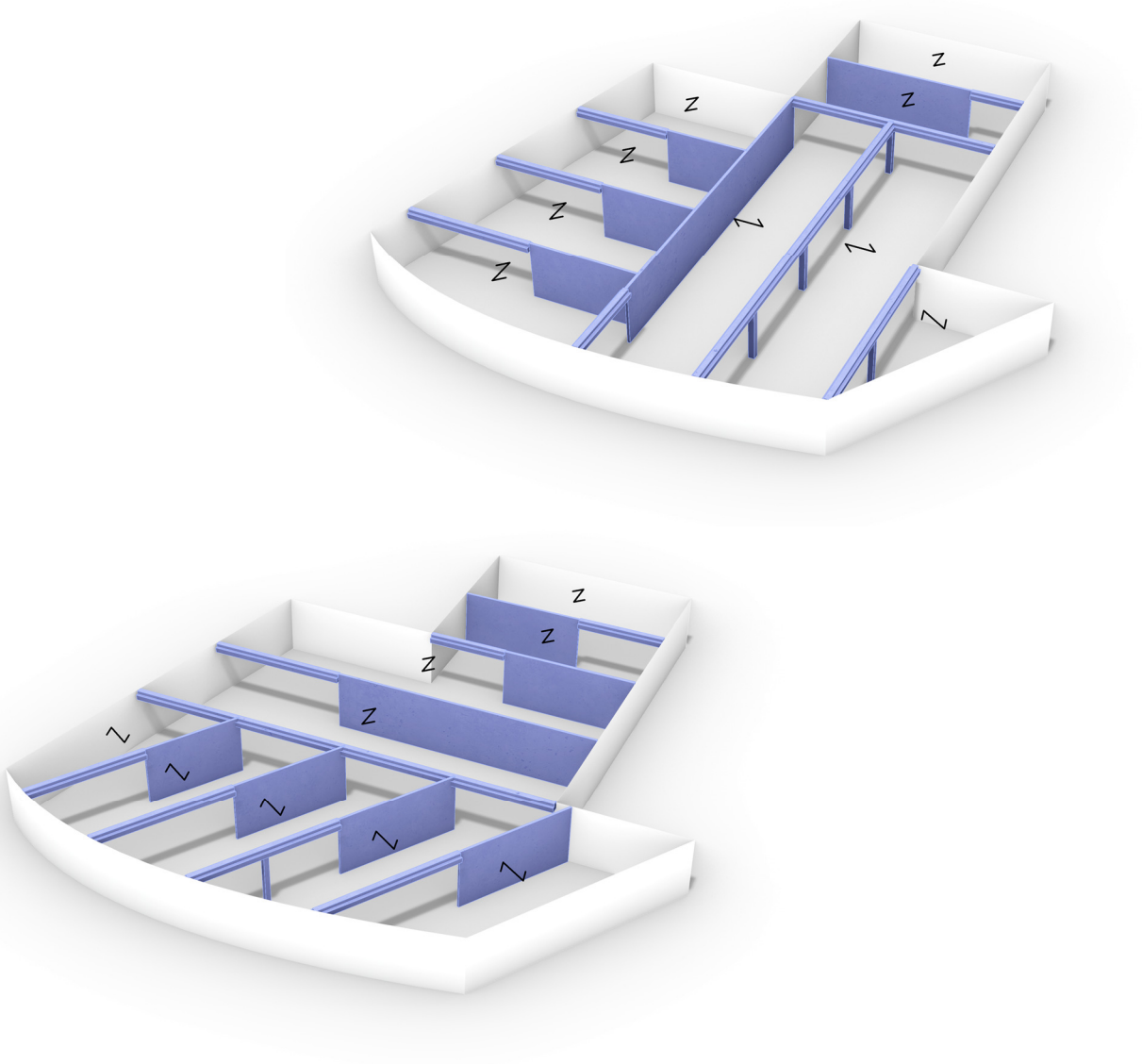
Conceptual Design Tool for Structural Layout Optimization in the Early Design Phase



PhD thesis

Lasse Weyergang Rahbek

April 2023

Conceptual Design Tool for Structural Layout Optimization in the Early Design Phase

Ph.D. Thesis

Author	Lasse Weyergang Rahbek
Thesis submitted	April 2023
Ph.D. supervisor	Prof. Poul Henning Kirkegaard, Department of Civil and Architectural Engineering, Aarhus University
Ph.D. co-supervisor	Associate Prof. Umberto Alibrandi, Department of Civil and Architectural Engineering, Aarhus University
Ph.D. co-supervisor	Carsten Rabjerg Terp, hamiconsult a/s

Graduate School of Technical Sciences, Aarhus University
The Ph.D. programme in Civil and Architectural Engineering

© Copyright by the author

ISBN: 978-87-7507-544-7
DOI: 10.7146/aul.490

AARHUS UNIVERSITY

Preface

The present dissertation, entitled "Conceptual Design Tool for Structural Layout Optimization in the Early Design Phase" was written as part of an industrial Ph.D. program at the Tectonic Design research group within the Design and Construction Department at Aarhus University in Denmark. The research was conducted between January 15, 2020, and April 11, 2023.

The Ph.D. project has been developed in collaboration between hamiconsult a/s, Innovation Fund Denmark, and Aarhus University and is part of a long-term strategy developed by hamiconsult a/s. One of the main objectives of this strategy is to ensure that the company is adapted to a consulting market with increasingly disruptive technologies and design methodologies where architects are taking advantage of several data streams available to them during design. The methodology of the design tool developed in this project can be applied generally. However, it is based on the Danish construction culture and specifically for using precast reinforced concrete elements.

The accepted Ph.D. project description was formulated back in 2019, at which time sustainability was already on the agenda. However, much can happen in just a few years. In January 2023, it became a legal requirement that all new construction projects in Denmark shall document the building's sustainability through an LCA (Life-Cycle Assessment) analysis. In the originally defined constraints, LCA was not intended to be part of the design objectives the tool would incorporate into its optimization process. This decision was made due to time limitations and to focus on the other defined objectives. However, it can be argued that by minimizing the mass of the building, the conditions for a successful LCA analysis are implicitly improved. It is also intended to incorporate LCA into the tool in the next planned version, as it is highly relevant commercially. However, it is also relevant from a scientific perspective to see the effect this inclusion will have on the algorithm's idea of an optimal design.

The author would like to take this opportunity to express my gratitude to all those who have contributed to the successful completion of this Ph.D. thesis.

A special thanks go to Professor Poul Henning Kirkegaard, the main supervisor whose guidance was instrumental in developing the design methodology. The author is also profoundly grateful to Associate Professor Umberto Alibrandi for contributing with insightful and inspiring discussions. Thanks to co-supervisor Carsten Rabjerg Terp who acted as a sparring partner throughout the project. Our discussions on all the structural theories and aspects were of great value. A special mention goes to Gerner Klærke and Christian Ohn for making the project possible and contributing to formulating the initial idea. The author thanks Jacob Güldner from NCC, Jens Erik Rasmussen,

and Nicolai Fuglsang Torp from Ginnerup Architects. Your input and feedback were of great value and improved the final result.

The author would also like to acknowledge the valuable collaboration of Markus Hudert, Assistant Professor. Hopefully, this collaboration can continue in the future. Thanks to Niels Martin Larsen, Associate professor, and Anders Kruse Aagaard, Associate Professor from Aarhus School of Architecture, for their collaboration.

Thanks to Kasper Fey Hansen, Martin Kristensen, and all my colleagues at hamiconsult for contributing to a good work environment.

Lastly, a special thanks go to my wife. Thank you for listening, being patient and supportive, and giving me three wonderful children, two of whom came into this world during this time. I sincerely could not have completed this project without you.

Reading guide

This Ph.D. thesis is submitted in partial fulfillment of the Ph.D. degree in accordance with the rules and regulations of the Graduate School of Technical Sciences, Aarhus University, and the Ministerial Order on the Ph.D. Degree Program. It takes the form of a monograph.

Figures and tables are numbered according to the chapter in which they appear. A list of references, figures, and tables can be found at the end of the thesis. References to literature, papers, and websites will be noted with a number in the order in which they are mentioned.

The following academic publications have been produced during this Ph.D. program:

- L. W. Rahbek, P. H. Kirkegaard og U. Alibrandi, »Parametric grid mapping design tool for freeform surfaces using a genetic algorithm« *Proceedings of the IASS Annual Symposium 2020/21 and the 7th International Conference on Spatial Structures Inspiring the Next Generation*, 2021.
- L. W. Rahbek, C. R. Terp, U. Alibrandi og P. H. Kirkegaard, »Stock optimization of naturally curved wood logs on freeform,« *Proceedings of the IASS 2022 Symposium affiliated with APCS 2022 conference*, September 2022.
- N. M. Larsen, A. K. Aagard, M. Hudert og L. W. Rahbek, »Timber structures made of naturally curved oak wood: prototypes and processes,« *Architecture, Structures and Construction*, 2022.

It is noted that these papers are not directly related to this dissertation, though they still contribute to the scholarly output of the research.

Summary in English

This thesis develops a conceptual design tool capable of generating optimized structural layout suggestions for building design in the early design phase. The structural layout of a building is the arrangement and design of the load-bearing elements that support the weight of the building and resist external forces. The structural layout in this project solely consists of prefabricated reinforced (RC) elements. The use of prefabricated RC elements is embedded in the Danish construction industry and will likely remain so in the foreseeable future. Therefore, there is great potential for more effective use of concrete in terms of sustainability and decreasing cost. The proposed design tool can help fulfill this potential.

Action Research (AR) is used to create the conceptual framework of the design tool. The AR analysis consists of semi-structured interviews and a co-creation workshop where architects, engineers, and contractors contribute to the development of the design tool to ensure that the final tool conforms to real-world practice. The final design tool is based on this framework and developed using four core principles: optimization, interactivity, dissemination, and automation.

A novel parametric modeling method is developed in the design tool called Adjacent Polygon (APoly) representation. The APoly method creates a dynamic parametric representation of a given building plan to generate diverse yet feasible structural layout suggestions.

The evaluation modules of different structural typologies are constructed using surrogate models in the form of Neural Networks. The surrogate models are combined in a hierarchical structure to create an algorithm capable of predicting the optimized geometry and corresponding cost for a structural element based on the external conditions inputted into the algorithm.

The entire network of prediction models is then combined with a meta-heuristic optimization algorithm in the form of a Genetic Algorithm (GA) to create a surrogate-assisted optimization framework. A repair algorithm is incorporated into the GA to increase the number of valid solutions generated during each optimization iteration to decrease the convergence time.

The performance and reliability of the design tool are validated through two groups of local and global case studies. The first group consists of parameter sensitivity studies on the local approximation modules for each structural typology. The second group of validation studies examines the design tool's effectiveness across relevant building plans and scenarios. The corresponding results demonstrate that the tool can effectively adapt to these different settings and produce optimized structural layout suggestions. It is also demonstrated that the design tool can conduct multi-objective optimization and produce a front of Pareto optimal solutions.

Summary in Danish

Denne afhandling udvikler et konceptuelt designværktøj, der er i stand til at generere optimerede strukturelle layoutforslag til den tidlige design fase. Det strukturelle layout er defineret som placeringen af bærelinjer med en tilhørende dimensionering af de bærende og stabiliserende elementer. I dette projekt vil det strukturelle layout udelukkende bestå af præfabrikerede armerede beton elementer. Brugen af betonelementer er en fast integreret del af den danske byggeindustri, hvilket formentlig ikke vil ændre sig i den nærmeste fremtid. Der er derfor et stort potentiale for at anvende beton mere effektivt af hensyn til de miljømæssige omkostninger, men der er også et økonomisk incitament i at optimere i forbruget af beton. Designværtøjet i dette projekt kan være med til at indfri dette potentiale ved at generere strukturelt optimerede layout forslag.

Action Research (AR) bruges til at skabe et konceptuelt framework af designværktøjet. AR-analysen består af semistrukturerede interviews og en co-creation workshop, hvor arkitekter, ingeniører og entreprenører bidrager til udviklingen af designværktøjet for at sikre at værktøjet er tilpasset de behov der er i reel praksis. Det endelige designværktøj er baseret på dette konceptuelle design framework og er udviklet ud fra fire kerneprincipper: optimering, interaktivitet, formidling og automatisering.

En ny parametrisk modelleringmetode er udviklet til designværktøjet kaldet Adjacent Polygon (APoly) repræsentation. APoly-metoden benyttes til at generere en dynamisk parametrisk repræsentation af et vilkårligt bygningsplan således at der kan dannes varierede men bygbare strukturelle layout forslag.

De forskellige strukturelle evalueringsmoduler er generet ved brug af surrogat modellering i form af neurale netværk. Surrogatmodellerne opsættes i en hierarkisk struktur for at skabe en algoritme, der er i stand til at forudsige den optimerede geometri, og approksimere de tilhørende materialeomkostninger. Hele systemet af surrogat modeller kombineres herefter med en meta-heuristisk optimeringsalgoritme i form af en genetisk algoritme, i en proces der er kendt som surrogat assisteret optimering. Den genetiske algoritme modificeres med en reparations procedure der øger antallet af valide løsninger ved brug af brugerinduceret viden. Værktøjets performance og pålidelighed er valideret gennem lokale og globale case-studier. De lokale undersøgelser består af en parameterfølsomhedsstudier på de lokale surrogatmodeller for at undersøge om de reagere som forventet på ændringen af input. Derefter undersøges designværktøjet i sin helhed på forskellige scenarier og bygningsgeometri. Resultaterne viste at programmet er i stand til at generere optimerede logiske løsninger der er tilpasset de forskellige omgivelser og laster der påvirker den. Til sidst eftervises det at værktøjet også er i stand til at udføre en multi-objektiv optimering of producere en front af Pareto-optimale løsninger.

Table of Contents

List of Acronyms

Symbol	Meaning
ACS	Average Cell Size
APoly	Adjacent Polygon
AR	Action Research
AI	Artificial Intelligence
ANN	Artificial Neural Network
BFGS	BFGS quasi-Newton backpropagation
BPS	Building Performance Simulation
CAD	Computer Aided Design
CD	Computational Design
CDP	Conventional Design Process
CGB	Conjugate Gradient with Powell/Beale Restarts
CGF	Fletcher-Powell Conjugate Gradient
CGP	Polak-Ribiére Conjugate Gradient
CNN	Convolutional Neural Network
CPM	Average cost per meter
CPM2	Cost per square meter
CPF	Column Price Factor
DAG	Directed Acyclic Graph
DNN	Deep Neural Network
FEM	Finite Element Method
Fck	Characteristic concrete class
GP	Gaussian Process
GA	Genetic Algorithm
GAN	Generative Adversarial Network
GH	Grasshopper3d
HypE	Fast Hypervolume-Based-Many-Objective Optimization Algorithm
IAR	Insider Action Research
IDE	Integrated Development Environment
IDP	Integrated Design Process
LCA	Life-Cycle Assessment
LM	Levenberg-Marquardt algorithm
LOD	Level Of Detail
ML	Machine Learning

MLP	Multi-layered perception
MARS	Multivariate Adaptive Regression Splines
NN	Neural Network
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
OSS	One-Step Secant backpropagation
OX	Ordered Crossover
PMX	Partially Matched Crossover
PBGD	Performance-Based Generative Design
PRG	Polynomial Regression
PCA	Principal Component Analysis
RBF	Radial Basis Function
RF	Random Forest
RNN	Recurrent Neural Network
RC	Reinforced Concrete
RR	Recess Ratio
SCG	Scaled Conjugate Gradient Algorithm
SLS	Serviceability Limit State
SPEA2	Strength Pareto Evolutionary Algorithm 2
SVM	Support Vector Machines
SU	Surrogate
UI	User Interface
ULS	Ultimate Limit State
W	Normalized width parameter value
WPF	Windows Presentation Foundation

Chapter 1 – Introduction

"There is never just one answer to a design problem, or even just one satisfactory answer – there are many"

– Peter Debney [1]

1.1 Background and motivation

The concept of optimized building design is hard to define as it requires multi-disciplinary collaboration between several relevant stakeholders. Some building design objectives are quantifiable, such as energy efficiency, cost, structural performance, and durability. Other objectives, such as aesthetics, adaptability, and function requirements, are more subjective in nature. These objectives will also be weighted differently for every new building project. These aspects make building design a very complex endeavor where there will be many acceptable solutions.

As building designers, we should strive to achieve a fully optimized holistic building design even though the concept may be unattainable because of the infinite number of variables and the many design objectives that are often contradictory. Also, the concept of an ideal building design may change over time. One of the most important contemporary challenges today is climate change, and the construction sector is responsible for nearly 40 percent of the annual CO2 emissions [2]. This CO2 contribution can roughly be divided into operational energy and embodied carbon. The embodied carbon is defined as the sum of CO2 emitted to produce and transport every building component [3]. For many years the focus has been on reducing the operational energy as this aspect was the main contributor to CO2. This effort has resulted in a significant decrease in operational energy, while the embodied energy from materials has long been an overlooked aspect in the discussion of sustainable construction [4]. This focus has shifted in recent years as embodied energy now exceeds operational energy for new constructions.

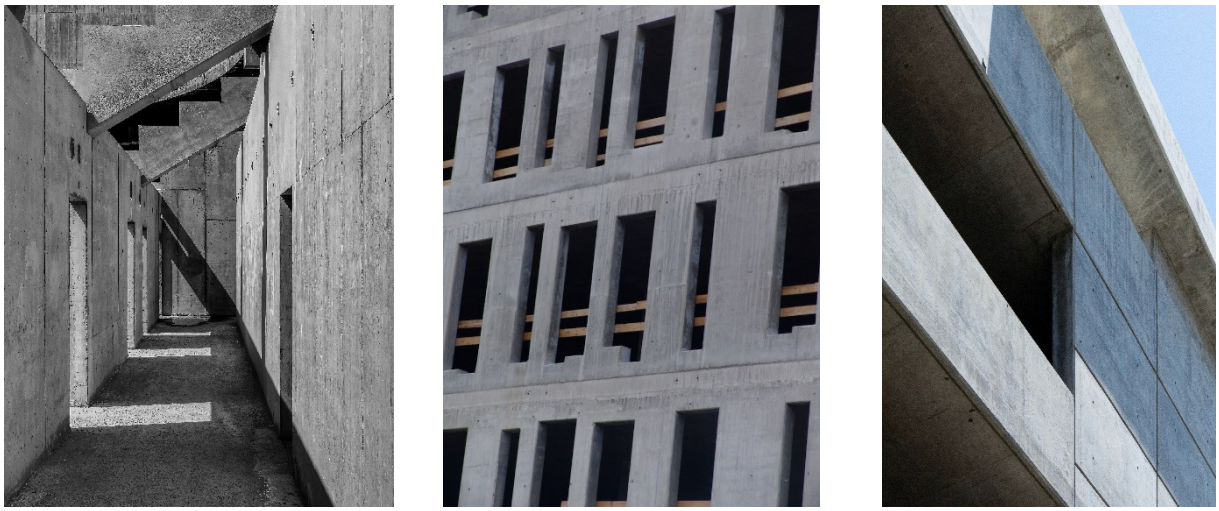
This shift is also reinforced politically in the Danish building regulation; a limit on CO2 emissions pr m^2 for buildings with more than 1000 m^2 is introduced as of January 1. 2023, for any new building structure [5]. Therefore, it can be argued that minimizing embodied carbon is one of the most essential design objectives in building design. From the structural engineer's perspective, it is, therefore, essential to optimize the load-bearing structure where the majority of embodied carbon is contained in new buildings. This project focuses on solving this issue

by introducing a new design methodology incorporated into a design tool that optimizes the structural layout in the initial design phase. The structural layout is defined as the arrangement and design of the load-bearing elements that support the weight of the building and resist external forces. The structural layout can consist of different structural typologies, but this project solely focuses on prefabricated reinforced concrete (RC) elements. The remaining part of section 1.1 introduces some of the themes relevant to this project, while section 1.2 provides an overview of state-of-the-art research in this field. Finally, section 1.3 defines the scope of the project and outlines the research questions, aims, and objectives.

1.1.1 Prefabricated RC elements

The most used building material globally is concrete [6]. This popularity is due to concrete's many beneficial properties, such as availability, affordability, high compressive strength, and the fact that concrete can be shaped into any form needed [7].

Reinforced concrete (RC) was first used in Denmark in 1891 to cast a slab, and the technique was adopted into the construction sector. Reinforced concrete was cast on-site in the first half of the twentieth century. This construction method is defined as in-situ concrete. After World War II, the construction sector had to adapt to become more efficient and began to use industrialized manufacturing methods to create prefabricated RC elements [8]. Prefabricated or precast RC elements are produced in a controlled environment, improving cost-effectiveness, quality control, safety, and construction speed. Different elements, such as beams, columns, walls, and slabs, can be produced and transported to the construction site, where they are assembled in a structural layout. An efficient structural layout takes architectural considerations, functional- and building code requirements into account, but the primary objective is to support and distribute the acting forces effectively.



(a) (b) (c)

Figure 1.1 – Examples of RC buildings. (a) image by Ricardo Gomez Angel. (b) image by Flickr user ACME. (c) image by Matt Reames.

Prefabricated RC elements can be used to create production halls or commercial buildings and are predominantly used to construct low and medium-rise housing buildings in Denmark [8]. The construction method is so embedded in the Danish construction culture that Denmark is the leading user of prefabricated concrete in Europe [9].

However, concrete is not the most sustainable material as it significantly contributes to carbon emissions, especially during manufacturing [10]. Despite the introduction of new, more sustainable materials, such as cross-laminated timber, it is projected that the global need for concrete will only increase in future years, as illustrated in Figure 1.2.

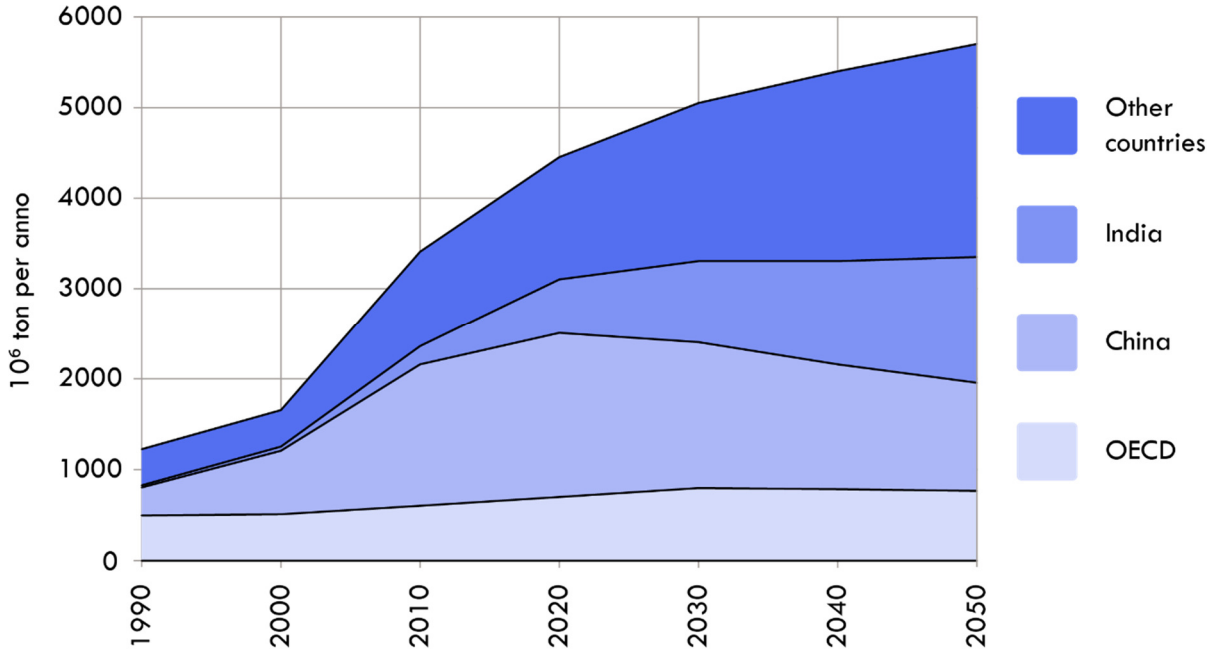


Figure 1.2 – Projected concrete production [12]

The growth of concrete production is more significant in developing countries than in the OECD countries, which primarily consist of Western countries. However, concrete will still be used in OECD countries because of the previously mentioned benefits and the existing culture in the construction sector. Therefore, it is vital to adopt a pragmatic approach to concrete as a building material and use it as effectively as possible to reduce the environmental consequences and comply with the legal requirements. There is also an economic incentive to reduce the building material used, which often can be a catalyst for change.

In continuation of the previously mentioned legal requirements in the building regulation, the concrete industry association released a guideline [11] for a series of measures to ensure that concrete structures are optimized to meet the specified functional requirements with the lowest possible carbon footprint. Some of the most relevant points regarding design optimization from the guideline are listed as follows:

- Assess the optimal choice for the statical system.
- Assess the calculation method (elastic or plastic).
- Optimize the reinforcement in conjunction with the geometry and material properties to achieve the lowest possible CO^2 footprint.
- Asses the load directions and the corresponding structural system

Other relevant points include materials reuse, topology optimization, and LCA. However, an important point in the guideline is to examine alternative structural layouts, and this analysis should be supported by involving the contractor and structural engineer in the early design phase to better integrate their knowledge and experience in the design [11], which can be done using an Integrated Design Process (IDP).

1.1.2 Integrated design process

Contemporary building design is a very complex endeavor, with many specialized disciplines such as HVAC, geotechnical engineering, fire-safety engineering, architecture, dynamic analysis, and more. Furthermore, due to this complexity, building design today is a strictly sequential process where the different disciplines work as separate entities without much cooperation across their respective fields. This approach is defined as a conventional design process (CDP). Despite the recent rise in digital tools and collaboration platforms, this is still the most common practice today [12]. Architects determine the building's geometry, massing, and overall form during the initial design phase. This task also includes placing the loadbearing and stabilizing elements. They typically carry out this task without involving engineering consultants [13] [14].

After the project has taken shape, structural engineers and other consultants start to carry out detailed calculations to realize the system provided by the architectural team. This approach is not ideal because a lot of the engineer's knowledge and know-how are not utilized to help create a more optimal form. The geometry of a building's structure directly determines the distribution and magnitude of the forces it must resist [15]. Therefore, the global configuration of the geometry holds the most significant impact on the structural performance of the building. So to achieve more effective designs, we need more collaboration among all participants [16].

This increased collaboration could be achieved by utilizing an IDP approach to building design. IDP can be understood as a collaborative building design method that emphasizes holistic design development [17]. This collaboration has to start early in the design process to have any beneficial impact. The basic concept of CDP and IDP is illustrated in Figure 1.3 and Figure 1.4, respectively.

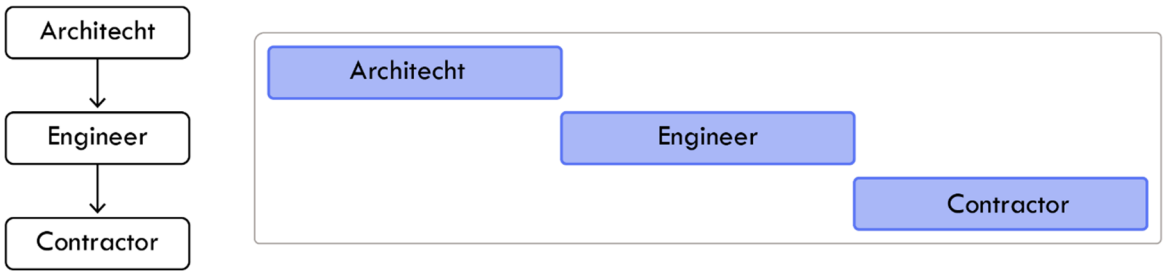


Figure 1.3 – The basic concept of a conventional design process (CDP)

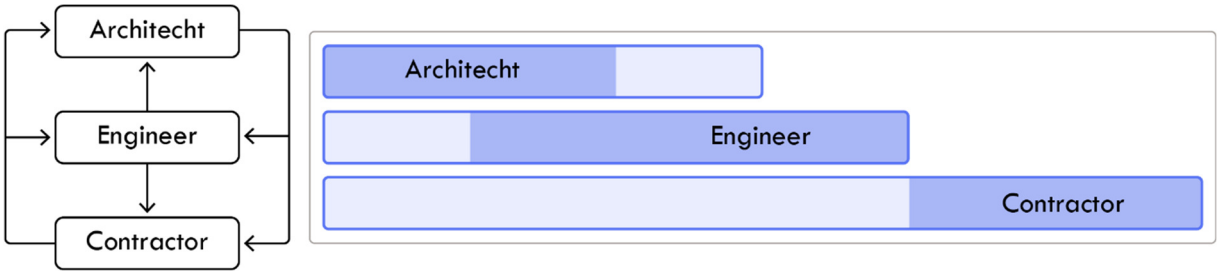


Figure 1.4 – The basic concept of an integrated design process (IDP)

It is noted that an IDP approach does not necessarily require more time but that the stakeholders are involved in the early design process in a joint collaboration to contribute their specialized knowledge to create a more cohesive design. This approach to building design is also supported by the MacLeamy curve, illustrated in Figure 1.5. It is noted that B. Paulson first introduced the concept of the MacLeamy curve in 1976 [18]. However, the concept is best known as the MacLeamy curve, and this notation will be used going forward.

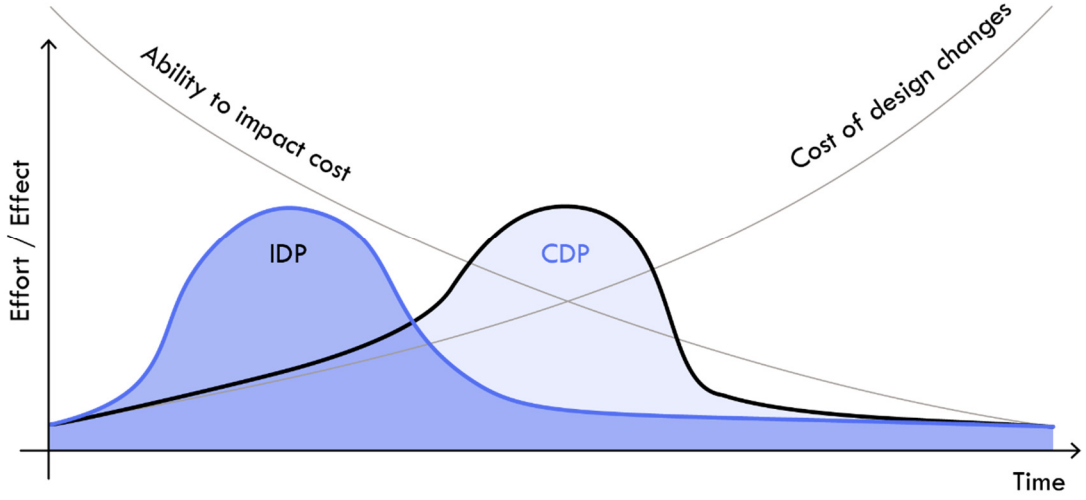


Figure 1.5 – Illustration of the MacLeamy curve

The MacLeamy curve is based on a pretty self-evident observation: An architectural project becomes more difficult to change the more developed it becomes [19]. While the MacLeamy curve is created from an architect's perspective, it can be argued that it is symptomatic for the whole building design phase and is, therefore, highly relevant to the structural engineers' work process. MacLeamy argued that increased focus on the initial design phase would benefit the project as a whole, as it is here that we have the most significant impact on the project. This phase in the construction process is often neglected because it is highly time-consuming, and therefore expensive [20]. MacLeamy stated that the shift to an IDP approach should be possible by increasing the use of automation [21].

This project also states that the use of computational design (CD) methods would support an IDP approach, as CD can facilitate more informed decision-making and enable collaboration among stakeholders. The concept of computational design is elaborated upon in the subsequent section.

1.1.3 Computational design

Computational design refers to the use of digital tools and techniques to generate, manipulate, and analyze design solutions. CD have rapidly gained popularity in the last couple of decades as a building design paradigm, which is also apparent in Figure 1.6. CD encapsulates a lot of methods, sub-categories, and terms. However, some terms are used inconsistently, and some overlap [22]. There is no commonly recognized taxonomy in CD, but the most used terms are parametric-, generative- and algorithmic design [23] [22]. CD also incorporates performance-based- performative- and evolutionary design and methods such as simulation analysis, repetitive task automation, and more.

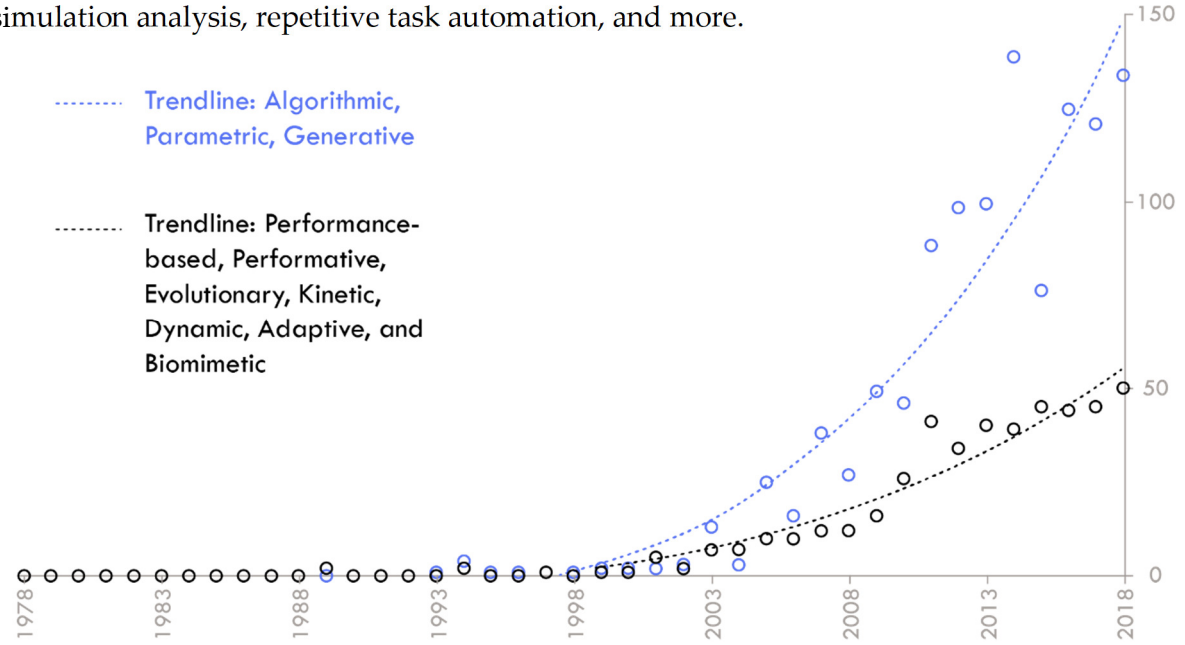


Figure 1.6 – Frequency of selected keywords in literature between 1978 and 2018. The x-axis defines the timeline and the circles and y-axis define the frequency use of the CD-related key word [20].

CD methods can generate more optimized-, sustainable- and aesthetically pleasing designs and provide real-time responses to design change, thus informing the designer of the potential consequence of each design variation. CD can also be used to create early design tools that facilitate collaboration between relevant stakeholders in the design process, such as architects, engineers, and contractors. These tools can generate efficient and robust design solutions optimized for the defined design objectives and go well beyond human intuition. All these aspects indicate that CD methods are well-suited for an IDP approach to building design. This strategy contrasts the CDP approach, where designers use their knowledge, experience, and intuition to create design solutions, often supported by computer-aided design (CAD) programs. This approach can only explore minimal design variations due to time- and resource restrictions [24].

1.1.4 Performance-based generative design.

Performance-based generative design (PBGD) can be defined as the generation of design solutions based on performance criteria and design restrictions [22]. PBGD is a sub-category within the field of generative design and CD.

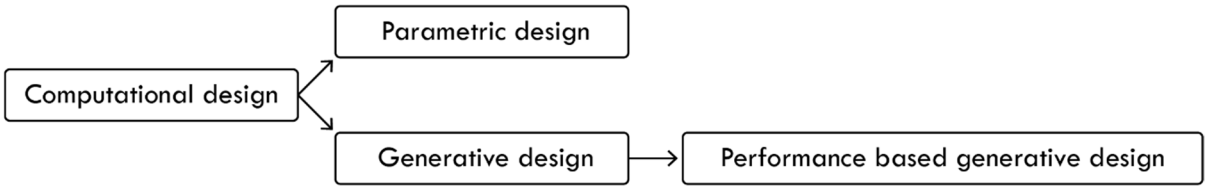


Figure 1.7 – The inherent terminology of performance based generative design.

This concept of PBGD enables a building designer to explore highly optimized design solutions that otherwise would have been near impossible to find through a manual trial-and-error procedure. PBGD of structural design in the context of optimization is a very complex task because of the many design variables, objectives, and constraints. These optimization problems can often be categorized as non-linear, non-continuous, and non-differentiable, excluding analytical solutions and numerical methods. However, these problems are well suited for machine learning (ML) methods often used in the PBGD models.

Generative design does not provide the exact answer, only better and worse ones. The user is still the principal designer, and the PBGD should be used as a support tool to inform the user of the advantages and disadvantages of different design options [1] [25]. The concept of PBGD is illustrated in Figure 1.8. Design variations are created from some base- geometry, settings, and constraints. The resulting design variations are denoted as phenotypes,

represented as gridshells in Figure 1.8. Each solution is associated with performance metrics to highlight their specific characteristics, in this example, using a radar graph.

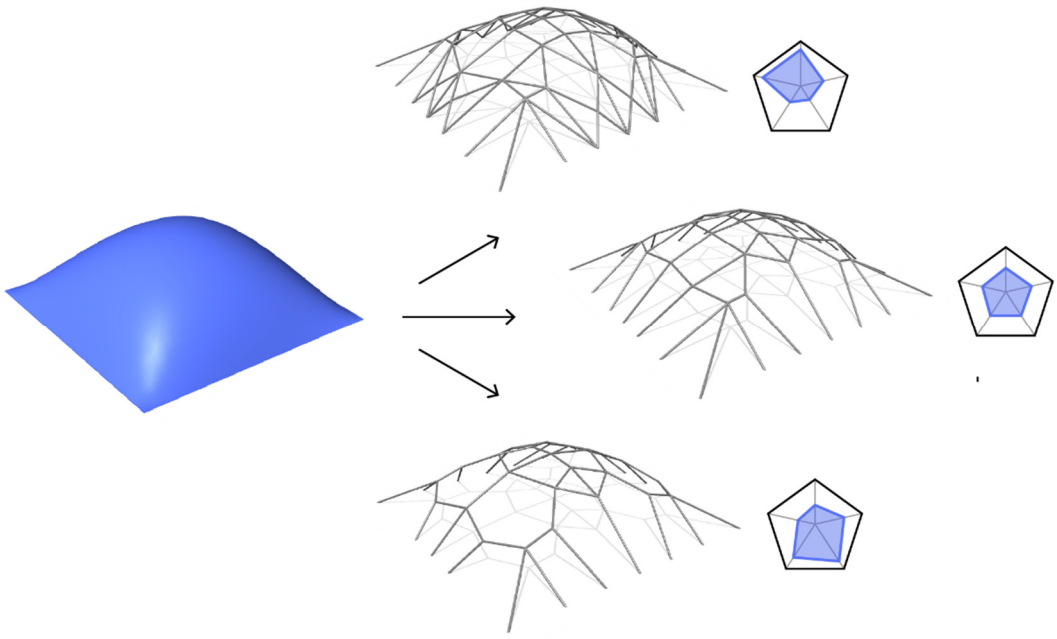


Figure 1.8 - The concept of PBGD

PBGD is not unanimously praised and recognized as the future of building design, and several blogs and articles present a case against the methodology [26] [27]. Some of the most relevant points are listed as follows:

1. Limited flexibility of design.
2. Optimization regarding a single load case.
3. Quantity over quality when generating solutions.
4. The methods do not adequately incorporate constructability and therefore generate impractical solutions.
5. PBGD only considers quantifiable metrics.
6. The process does not fit real-world design practice.
7. PBGD requires significant time and resources to set up and implement.

These points are all valid and should not be disregarded. However, these points should not necessarily be regarded as "deal-breakers" but rather a list of aspects to consider when designing a PBGD tool. Some of these issues are relatively easy to integrate, and others require much consideration and thought to solve—especially the three last points. PBGD is a data-driven process and therefore needs quantifiable metrics to run. For the most part, this is okay,

as many important metrics in building design are quantifiable, such as mass, cost, energy efficiency, and strain energy. There are also ways to incorporate the more subjective objectives into the design process, for example, through constraints and user-defined alterations to the design. There are also examples of aesthetic design objectives which have been quantified and incorporated into a PBGD algorithm to produce more uniform and aesthetic-looking designs [28] [29].

As to the issue of fitting into real-world practice, many PBGD tools indeed work more as a proof of concept. Instead of a design tool tailored to the need of the actual user. As stated in [26]: "*While it is trivial to show that generative design is possible, it is much harder to take the next step and show that generative design is useful*." Some research focuses on this problem [30] and suggests the problem could be solved by applying a research methodology such as Action Research to identify the end user's needs and how to incorporate the defined features properly. The issue of creating a generalized PBGD model is discussed in the following section 1.1.5

1.1.5 Generalized parametric models.

The use of parametric- or PBGD modeling is increasing, but we have still not seen a broad breakthrough in practical engineering. It requires skill, expertise, and time to develop a parametric or PBGD model, and sometimes it would be faster to solve the issue without CD methods. CD methods are, therefore, not always relevant, and the use of CD should be considered from case to case. Also, many parametric- and generative models for structural optimization are created for unique designs and cannot be applied generally. Examples include the Twist in Jevnaker, Norway, the Active Energy building, the European Central Bank in Frankfurt am Main, and the Forest Tower in Gisselfeld Monastery forest, Denmark. Therefore, it is argued that a greater economic perspective exists in creating PBGD models that work on more common design typologies without an increased setup time.

However, it is not easy, as a PBGD model operates on parameters. Therefore, creating a parameterization method that can be used generally on any design is necessary. This parameterization should also ideally consist of as few variables as possible to reduce the size of the solution space the optimization algorithm has to search through.

The complexity can be reduced by not creating a model that needs to encompass 100 percent of all possible design inputs. Instead, the aim should be to create a model that works for around 80 percent of all cases. The 80 percent limit is somewhat arbitrary, but the point is that it will often require more resources to incorporate the last 20 percent than the first 80 percent, and it is therefore not cost-effective to use resources on the last 20 percent.

A promising application for a generalized PBGD model in Denmark would be the development of a structural layout tool capable of generating optimized design proposals for the structural layout of prefabricated concrete elements. This statement is supported by all the reasons established in section 1.1:

- The use of prefabricated RC elements is an integral part of the Danish building culture and is likely to continue being so.
- The concrete industry association has released a series of recommendations to reduce concrete consumption, including exploring different structural layouts to a greater extent.
- The concrete industry association also recommended better and earlier collaborations between stakeholders, which could be supported using an IDP approach and an early design tool to enhance such collaboration.
- The construction industry is responsible for approximately 40 percent of the total CO2 emissions, and because concrete is one of the most widely used construction materials, then there is a significant potential in reducing the consumption of concrete by using it more effectively.
- There is a legal incentive to reduce the amount of concrete as the Danish building regulation introduced a limit on CO2 emissions per m^2 for new buildings.
- A tool that generates optimized structural layouts could reduce the total building cost.

1.2 State of the art

1.2.1 Literature review strategy

A literature search was conducted to supplement the set of already known papers relevant to developing conceptual design tools for layout optimization of RC elements. The purpose of this search was twofold.

Firstly, the search strategy focuses on finding relevant research on developing structural shear-wall layout optimization. This search focused on finding papers containing the term: "concrete shear-wall layout optimization" somewhere in the entire text. This approach was applied because too few potential papers emerged if this search focused on the title, abstract, or author-specified keywords. Paradoxically this resulted in too many results. Therefore, the search was adjusted with a "cut-off" date of January 2016. First, a preliminary screening identifies and removes irrelevant papers, such as review articles, because they do not include any methods. Book chapters and encyclopedia text are also removed. Then every paper that was not categorized as engineering was removed. This exclusion also involved papers in irrelevant subject areas. Then two review rounds were conducted to reduce the number of papers to a more reasonable amount. The first round reviewed papers based on their title alone. This review round was conducted as an inclusion process rather than an exclusion, meaning the title must be considered relevant to proceed to the second round. In the second review round, the abstract was read, and the paper's text was screened to determine its relevance to the project. The review procedure is shown in Figure 1.9.

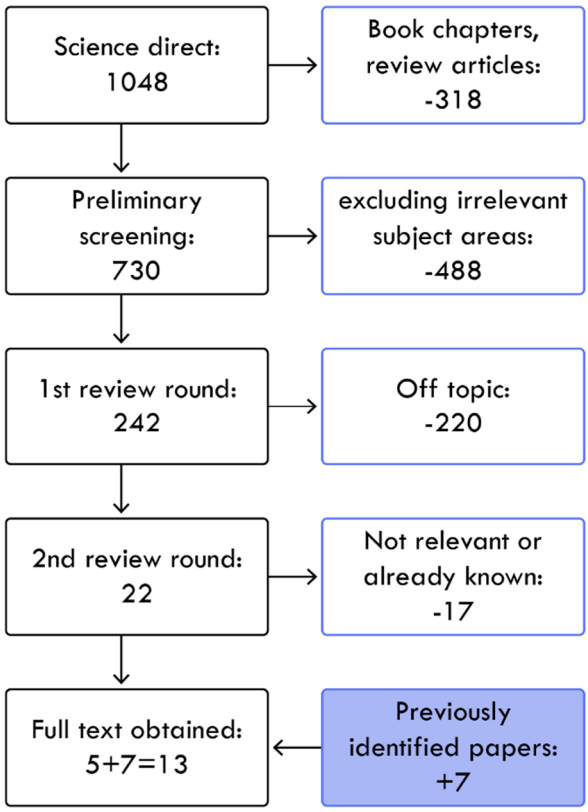


Figure 1.9 – 1st literature review procedure.

The second primary literature search strategy focused on exploring all the themes that could be relevant for the development of a generative design tool for structural layout optimization. There are many different relevant theories and methods, and the search focused on, though not limited to, the following search terms:

- Conceptual design, parameterization, performance-based design, Surrogate assisted optimization, automation in structural engineering, interactive optimization strategies.

1.2.2 Related work

Creating a tool to generate structural layout suggestions have been the subject of previous research. A. Retik [31] created such a tool in 1994. Many valuable properties were incorporated, such as cost estimation, user interactivity, and 3d illustrations. Though the tool also had some severe limitations. The tool generated suggestions based on a deterministic ruleset, so no performance-based optimization was involved. The tool also does not consider the lateral stability of the given structure, which, especially for taller buildings, can significantly impact the structural performance. The tool also only processes orthogonal-shaped buildings and a unidirectional load span.

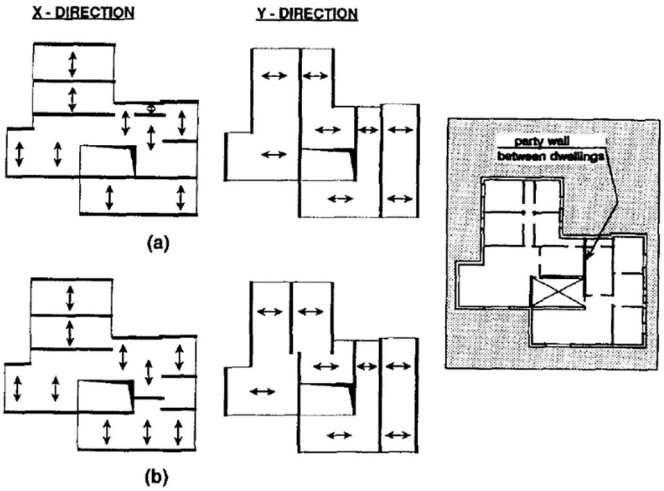


Figure 1.10 – Examples of alternative layout suggestions. Image by A. Retik and A. Warszawski [27].

Throughout the last two decades, there has been increased research on this topic. M. Jinjie combined FE analysis with an expert algorithm [32]. P. Zhao used a deep neural network (DNN) trained on existing data to generate a beam-slab layout [33]. S. Talatahari used a quantum-charged system search to optimize the shear wall layout of tall buildings [34]. T. Takada, Y. Kohama, and A. Miyamura created a proof of concept using a branch-and-bound method and combinatorial optimization to find the optimal allocation of shear walls in a 3d multi-story frame [35]. The objective was to minimize the total amount of shear walls while subjected to displacement-, torsional- and up-lift constraints. The method only works on rectangular planes and orthogonal lines and does not incorporate the slab into the design process.

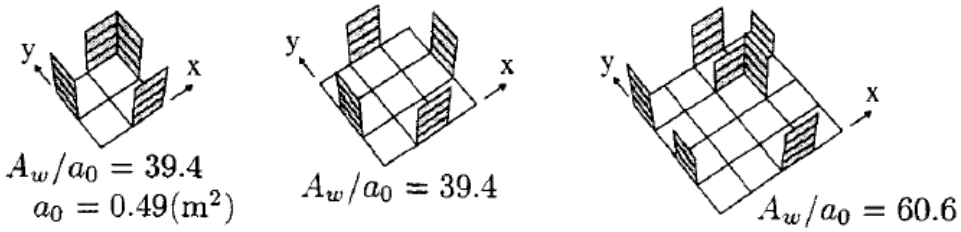


Figure 1.11 – Optimal allocation of shear walls. Image by T. Takada et al. [28].

Zhang and C. Mueller used the ground structure approach and a modified genetic algorithm to generate optimized shear wall layout designs [36]. They emphasized the importance of reducing the gap between architect and engineer with a tool that supports an IDP approach. Though the tool still had simplifications, such as being limited to orthogonal lines and simplified calculations, and it only considered shear walls in the structural system.

H. P. Lou et al. developed their shear-wall layout optimization tool using a tabu-search algorithm [37]. They utilized a surrogate model (SU) in the form of a support vector machine to reduce the cost of the model's evaluation.

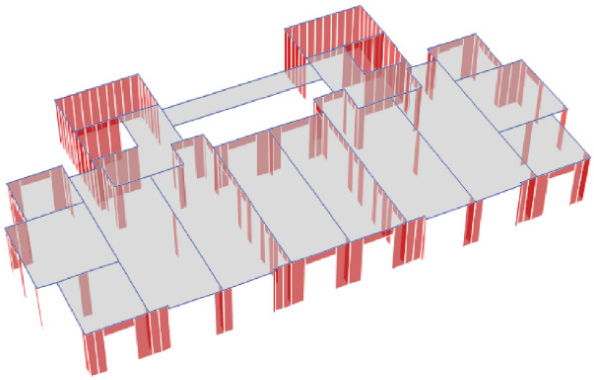


Figure 1.12 – Potential shear wall layout. Image by H.P. Lou et al. [33].

However, the method only optimizes the shear wall placement and does not incorporate structural elements such as beams and slabs into the design process. Furthermore, the method operates on a rigid set of module lines which limits the flexibility and variations of the design solutions. This issue is common within the addressed related work, and it reduces the flexibility of the design process when the solution space is confined to a predetermined set of potential lines. Furthermore, these lines must be orthogonal or at least linear in almost every reviewed project. Another issue is that the proposed methodologies do not incorporate all structural elements into one cohesive optimization process. These methodologies are also primarily created as a proof-of-concept and, therefore, not adapted for the need of real-world building design practitioners. Incorporating constructability is vital if the tool should be able to assist an IDP approach. H.P. Lou et al., Zhang, and Mueller does recognize and write about the potential of a more integrated design approach. They do, however, focus more on the barriers instead of on how to resolve them [37] [36].

This issue was the subject of P.B. Purup and S. Petersen. They proposed a research framework to develop building performance simulation (BPS) tools conformed to the design practices in the early design stage [30]. They also stated that, to their knowledge, no early HVAC design tools had been adopted in professional design practices.

This statement is partly also the case for structural engineering, though some exceptions exist. Søren Jensen a/s, BIG, Ramboll, and Arup use CD methodologies in their work. This

exception is also the case for Bolllinger+Grohmann. They utilize genetic algorithms to search through several constellations to find a solution that conforms to the given objective and constraints. This use of ML is evident in projects such as the Active Energy Building, Skybridge, and the European Central Bank in Frankfurt am Main. Common for all these projects was that the problem representation used was a simplification driven by the available technique.

1.3 Scope of the project

The objective of this Ph.D. project is to create a conceptual design tool to generate a catalog of optimized layouts of the load-bearing structures in the initial design phase. It is noted that the project's output is denoted as a design tool. In reality, it is more comparable to a framework featuring several design tool modules. This framework structure is necessary to accommodate the complexity of creating a tool containing all the required features. The definition of "design tool" is still telling, and this wording will be used interchangeably with "framework" and "design methodology" throughout the thesis.

The design tool is intended to inform the design with several relevant data streams such as cost, material type, and more. The tool is not intended to drive the design process but to support the design team in creating more successful designs that perform well regarding the defined design objectives and constraints. The tool is intended for real-world application, and in a real-world application, the design is often restricted to fit given criteria. These restrictions could be architecturally motivated. Therefore, the tool must be interactive and flexible to maximize design freedom. Another important aspect is the dissemination of the results.

If the tool is to achieve broad recognition, then the tool's procedure and output must be transparent to obtain the necessary credibility for real-world application. The responsibility relies solely on the engineer in collaboration with the architect. Therefore, the engineer needs to be able to obtain an overview and verify the static calculation results. Also, great dissemination of the results and calculations can be used directly in final static documentation, provided the calculation contains the required fidelity level. Thus, reducing the amount of work that an engineer would have carried out manually. This feature is also important as it is required for an industrial Ph.D. project to contain a commercial prospect. The commercial prospect resides in different aspects of the project. Automating engineering tasks enables hamiconsult to increase their capacity without increasing the need for "manpower." Automation is especially relevant when considering the report from "Engineer the Future" which states that there will be a high deficiency of engineers in the future [38]. Therefore, the tool would provide a possible way to reduce the workload while increasing the quality of the design solutions. The use of ML and optimization algorithms could reduce the amount of building material needed significantly. This reduction will decrease the cost of a given design

project and therefore increase hamiconsult's competitiveness. The tool is also an opportunity to reduce the risk of calculation errors. Engineers make mistakes, and not every mistake is found during the quality assurance process. A structural engineer will always verify the conceptual design tool's results, but the possibility of certain mistakes is eliminated (or at least reduced). The research generated from this project will also benefit the building industry as a whole since it explores the possibility of applying state-of-the-art methods as a complementing tool for the engineer.

1.3.1 Design limitations

The project has to introduce certain limitations to realize all these features due to practical constraints such as time and resources. Therefore, it is important to prioritize which features to incorporate by evaluating their influence on the final design and the resources required to implement them, as well as determining whether a particular feature will excessively increase the computational burden of the design tool in relation to its value.

This section outlines some of the most notable features that have not yet been integrated into the design tool:

- The tool will only focus on structures using precast RC elements.
- The tool does not consider other relevant building design disciplines, such as HVAC and LCA.
- Fire load calculations are not incorporated as the building's geometry changes in each iteration, making it difficult to assess the person load and the usage category. Moreover, conservative assumptions of fire conditions would result in oversizing the structural elements, leading to an imprecise representation of the building's geometry.
- Robustness in the context of a building's ability to resist unexpected failures or damage is not included. The necessary calculations to document a building's robustness vary based on its consequence class, which can change for each design iteration depending on the global geometry. In addition, assessing robustness requires an evaluation of the consequences of the failure of a given element [39], which is an iterative process that must be carried out for each element. These requirements make robustness considerations too resource intensive to include.
- Seismic load is not included in this version of the tool. In Denmark, seismic loads rarely govern the sizing of the stabilizing system. It would computationally be expensive to include these calculations in an optimization process as the structures change for each iteration.

- Certain limitations have been imposed on the geometric flexibility of the tool due to time constraints. The most apparent limitations are the absence of concrete cores and considerations for non-regular plan elevations.

Despite the absence of these features, the tool's ability to identify optimized structural layouts is not diminished, and it still serves as a proof of concept for the design methodology. The exclusion of these parameters is also not necessarily permanent, and section 6.3 reflects on how some of these features could be incorporated in future versions.

However, in an idealized scenario, the building design would be parameterized 1:1, every relevant building design discipline incorporated, and the evaluation time would be nearly instant. This goal is unrealistic, but this does not mean we should not strive for a truly holistic building design tool. Because the knowledge and experience for each research project accumulate and the design tools available improve with time. Hopefully, this project will add to this accumulated knowledge, and future research or versions of this tool will pick up the thread and incorporate the limitations defined in this project.

1.3.2 Research questions, aims, and objectives

All the aspects and considerations presented and discussed in Chapter 1 lead to the following research questions, aim, and objectives.

Research questions

- What features in an early design tool would support building designers real-world work practice?
- How can a parameterization method be designed to generate a range of diverse and practical solutions?
- What are the potential design strategies for generating optimized structural layout suggestions that incorporate all relevant structural elements within a single optimization process?

Research aim

- Identify the barriers for implementing early-stage design tools and features that would help building designer in their everyday work practice.
- Asses what parameterization- or combinations of parameterization encoding technique best fitted for the needs of the design tool.
- Assess the use of hierarchical surrogate modelling for rapid evaluation response.

Research objective 1

- Use action research as a structured approach to the concept development of the design tool.
- Use action research to identify features which will contribute with better user-interactivity and dissemination.

Research objective 2

- Develop a parameterization technique that can generate varied and relevant structural layouts on any given building plane.
- Incorporate user-interactivity into the parameterization methodology.

Research objective 3

- Develop a conceptual design tool that utilizes surrogate modeling to generate optimized structural layouts for prefabricated RC elements.
- Demonstrate the effectiveness of the design tool on relevant cases using different settings and constraint.

1.4 Organization of the Dissertation

The thesis is organized into six chapters. The outline of the chapters is summarized as follows:

Chapter 1 – Introduction

The background and motivation for this Ph.D. project are presented. Additionally, some of the design methodologies the project is based upon are discussed. Furthermore, a state-of-the-art is presented with a focus on related work. Lastly, the project's research questions, aim, and objectives are presented.

Chapter 2 – Methodological framework

The most relevant methods and theories that form the basis for the development of the tool are presented. This presentation is not extensive but focuses on the core principles and how they relate to the design tool. For a more in-depth review, reference is made to supplementary literature.

Chapter 3 – Action research analysis

Chapter 3 introduces the principles of action research (AR). The subsequent AR analysis was used as a structured framework for developing the overall concept of the design tool. The result of this analysis is presented with a focus on the final framework of the design tool.

Chapter 4 – The design tool

A thorough description of the finalized mechanism of the design tool is presented. Alterations from the original framework in Chapter 3 are discussed and displayed.

Chapter 5 – Validation studies

The tool's effectiveness and robustness in producing structural layout suggestions are validated through relevant case examinations. The analysis is split into two main sections, which can essentially be defined as a local- and global analysis of the design tool. First, the effectiveness of the separate modules is presented and analyzed. Secondly, the tool's ability as a unified unit to produce structurally optimized layouts is demonstrated and evaluated.

Chapter 6 – Conclusion

The primary results and conclusions of the project are presented. The main findings are discussed and reflected upon. A subsection in this chapter is dedicated to present future work and prospects concerning short- and long-term goals.

Chapter 2 - Methodological Framework

"All exact science is dominated by the idea of approximation."

– Bertrand Russell

Every scientific theory, -model, and -method is based on the idea of approximation. They are constructed to represent the observable physical phenomena occurring in the world. The theories are formulated using the scientific method through systematic observations, measurement, and experiments and then formulated into a theory that can be tested. The theory is then accepted or rejected based on its prediction accuracy, which can be measured using statistical methods. In time these may be replaced with newer theories that better represent the world. A quote attributed to George E.O. Box states: *"… All models are wrong; some, though, are more useful than others…"* [40]. We use these models in almost every aspect of modern life, and in structural engineering, they are used to getting the answer near enough, with failure not being an option [1].

Chapter 2 presents some of the most relevant theories used in this project to develop a conceptual design tool capable of generating optimized structural layouts for a building design. Section 2.1 discusses the topic of parametric modeling, including some of the key aspects in this field. Section 2.2 introduces optimization on a general level, while section 2.3 explains the mechanics of a Genetic Algorithm, a meta-heuristic optimization algorithm. Section 2.4 introduces the topic of surrogate modeling.

Not all discussed methods were incorporated into the final design tool, though they were all considered. The inclusion and exclusion of these methods were determined through prototype testing. Finally, Chapter 2 establishes some of the terminology used in later chapters.

2.1 Parametric design

Defining parametric design or parametric modeling can be challenging, even for experts in the field, as noted by D. Davis [19]. Davis also outlines various definitions that have been utilized in parametric design. Therefore, it is important to state the definition used in the particular project's context. This principle also applies to any subcategories and concepts within the field of parametric design. This section will elaborate on relevant concepts, approaches, and definitions used in parametric design which are relevant for developing the design tool in this project.

In this project, parametric design is defined as the process of using algorithms to generate and manipulate geometric shapes, material properties, and other parameters that define the properties of a building design.

Parameters can be understood as values or variables that define the geometry, functional characteristics, or even the aesthetic expression of a building. This segmentation can occur in different ways. Parameters can represent dimensions, loads, materials, constraints, and other relevant aspects of building design. Manipulating these parameters enables the engineer or architect to explore different options and optimize the performance of a building or structure.

There is, in theory, no limit to how detailed a parametric model can be. However, this should not be the goal because the solution space becomes so vast that it will be impossible to navigate. It is important to reflect on the essential parameters and incorporate them into the model. A balanced parametric model is flexible and can generate varied solutions while still sustaining a manageable solution space.

Parametric modeling also requires a degree of upfront planning and reflection. A parametric model can vary and change effortlessly within the scope of its parameters. It is much harder to accommodate a requested change that was not anticipated when creating the model. This model adjustment can potentially mean a total rewrite of the parametric model [26].

Parametric design can be categorized based on different classifications or approaches, such as:

- Problem type, e.g., structural design, product design, landscape.
- Available software, e.g., Rhino + Grasshopper, Dynamo, Revit.
- Approach, e.g., Top-down, bottom-up.
- Encoding, e.g., binary encoding, permutation encoding, value encoding.
- Level of complexity, e.g., many variables, few variables.
- Shape grammar-based parameterization.

Numerous other categorizations and approaches to parameterization exist, as discussed in [41] [19] and [13]. Determining the correct categorization can aid in identifying appropriate methods for constructing the most efficient parametric model. The classifications listed in this section are emphasized since they are relevant to developing the design tool in this project. Sections 2.1.1 to 2.1.6 will further elaborate on these categories.

Finally, it is necessary to define the concept of parameterization. In the context of this project, parameterization of a building is essentially defining the design domain in which a building can change. It should not be confused with parametric modeling or encoding. Encoding in computing is defined as the process of converting one data format into another

[42]. Encoding and parameterization are closely related concepts. In this project, the following distinction is used to supplement the already established definitions: encoding is the process of converting design parameters into a machine-readable format, while parameterization is the process of linking these parameters together to create a parametric model.

2.1.1 Problem type

Parametric modeling based on the problem type refers to the industry in question, as parametric design is widely applied in everything from product design to building design. The nature of the problem significantly influences the parametric approach to solve it best. For example, when developing a parametric model for structural design, it is essential to consider the local and global geometry, material properties, connection types, and external forces.

In product design, it may be more important to ensure that the parametric model is highly geometrically flexible, enabling the exploration of a wide variety of solutions in terms of aesthetic and functional qualities.

Other relevant areas that use parametric design include aerodynamics, urban design, landscaping, and lighting design. Whatever the case, the problem at hand greatly affects the design and approach of parametric modeling. Examples of parametric design for different industries are shown in Figure 2.2 and Figure 2.1.

Figure 2.2 – Example of a parametrically designed structure in the Aliyev Cultural Center in Baku. Image by Flickr user Anton VG.

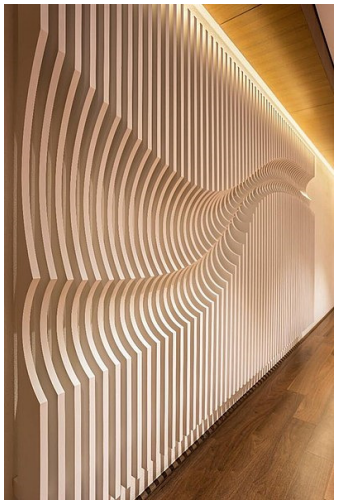


Figure 2.1 – Parametrically designed product. Image by Flickr user Sharan Sharma.

2.1.2 Parametric software

Parametric design can also be categorized based on the software platform used to implement the parametric model. Different software programs offer different capabilities and may have a unique approach to parametric modeling. One of the most popular software products is Rhinoceros3d + Grasshopper3d [43], also referred to as Rhino and Grasshopper, respectively. Rhino is 3D CAD software developed by Robert McNeel & associates and is used

in many different industries, such as architecture, engineering, and product design. The 3d graphical abilities in Rhino can manipulate, render, and animate NURBS representations, polygon meshes, and several other geometrical constructs, making it a versatile tool for complex 3d analysis.

Grasshopper is a visual-based programming language that runs within Rhino. Grasshopper is very intuitive and works by creating connections between different components with nodes in flowchart-style. The input then runs downstream and affects the following components until the final output is reached. This parametric approach allows for the fast creation of complex designs that can be easily manipulated, and is also known as Directed Acyclic Graph (DAG) [41].

Other relevant parametric software tools include Revit + Dynamo [44] and SolidWorks + DriveWorks [45].

2.1.3 Top-down or bottom-up parameterization

The approach category refers to the chosen parameterization strategy of a given problem. In building design, a top-down approach starts with a high-level concept or overall building form. This form is then broken down into smaller, more detailed components. The relationships and parameters define the overall structure; these relationships are then used to create and refine the smaller elements. This method contrasts with the bottom-up approach. Here the initial parameters are used as input to define individual elements or modules which are used to build up the overall structure.

The design of a gridshell can be used to illustrate the difference between these two strategies. In a top-down approach, the gridshell's overall shape is defined first, and the grid topology is projected into it, then dimensioning the individual grid elements and their connections. In a bottom-up approach, the process starts in reverse by defining the connections' capacity and gradually adding grid elements until a complete configuration is achieved. Dynamic relaxation is used to define the final shape of the gridshell.

While both approaches result in a gridshell design, the approach is vastly different and impacts the final outcome.

Another factor in parametric modeling that can significantly impact the final design is the choice of parameter encoding. The subsequent section will further elaborate on this topic.

2.1.4 Encoding types

As earlier defined, a parametric model uses a set of parameters to represent a given solution or design. This set is often denoted as a chromosome, with an apparent reference to biology, where a chromosome carries the genetic information representing an animal. This concept is illustrated in Figure 2.3.

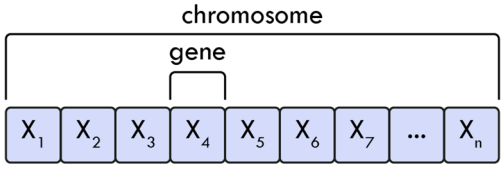


Figure 2.3 – Conceptual illustration of how parameter information is stored in a chromosome.

Each parameter in the chromosome represents an input that has been encoded into a data structure the algorithm can understand and operate. These encoded variables can then be used in one of the CD representations shown in Figure 1.7 to generate and manipulate different solutions. There are different types of encoding related to the nature of the problem it represents, and some of the most common types are illustrated in Figure 2.4.

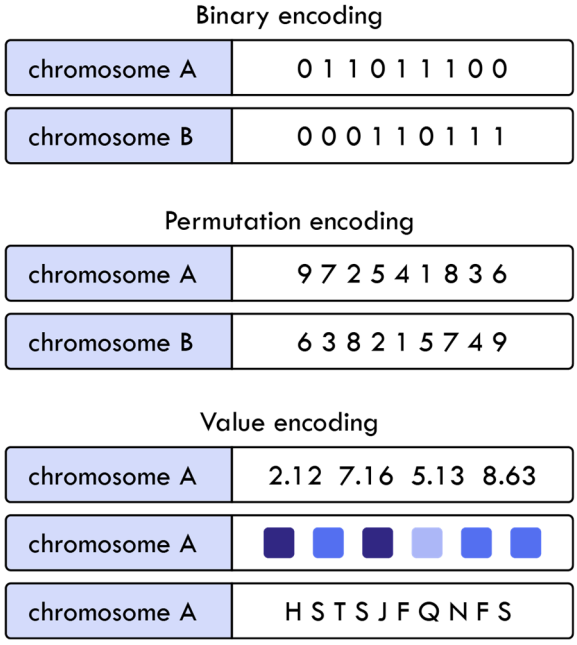


Figure 2.4 – Examples on the different encoding types.

Binary encoding represents a solution through a set of "one" and "zero" values. This encoding method is often used in structural engineering, as it is simple, and the values can represent the existence or nonexistence of structural elements. Permutation encoding is

utilized in problems where the order of variables is used to represent a given problem. This encoding type is relevant in problem areas such as stock optimization and bin packing problem and can be used to solve the traveling salesman problem. Value encoding, or real parameter encoding, is an array of values representing any possible property of elements, for example, real values, characters, or colors. For other encoding types, see [46] [47].

Figure 2.5 illustrates an example of how to represent a given grid structure through binary- and real-number variables. This encoding can turn the elements "on" and "off" and move the vertices of the structure; other parameters could have been added to represent each element's cross-section or color. The principle can be used to represent any given structure.

There are in principle no limitations on the number of parameters and encoding types used to represent a given structure. However, increasing the complexity can give rise to issues, which is discussed in the subsequent section.

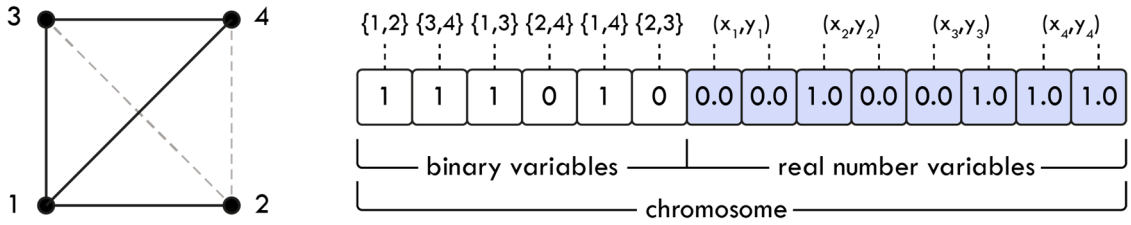


Figure 2.5 – Example of how to represent a grid structure through parameters.

2.1.5 Level of complexity

Complex problems can sometimes require a high number of parameters; this is an issue as the solution space increases exponentially with each added variable. This complexity can make it very hard for an optimization algorithm to navigate a solution space and converge toward optimum solutions. Also, the high number of parameters often correlates with an increased evaluation time, reducing the optimal conditions for the optimization algorithm. Specific measures can be implemented to deal with these issues:

- The first and most simple measure to consider is the possibility of a simpler representation of the problem with fewer variables.
- Surrogate models can also be implemented to construct approximation models of the problem to reduce the evaluation time significantly [48] [49].
- Dimensionality reduction techniques such as Principal Component Analysis (PCA) can simplify the problem. These techniques aim to reduce the number of parameters by identifying the most influential ones [50].
- Many-variable problems are well suited for meta-heuristic algorithms, such as genetic algorithms and particle swarm optimization. These algorithms have mechanisms that balance exploration and exploitation of the solution space [51] [52].

2.1.6 Shape grammar

Shape grammar or grammar-based design refers to the use of geometric rules to represent a structure or design. The methods operate on a set of initial shapes. These shapes are then manipulated and combined into more complex shapes; this process is recursive and can continue indefinitely. Shape grammars have the potential to produce more diverse solutions than using more traditional methods in parametric design and can be used to explore possibilities across typological boundaries [13] [53].

Rules in shape grammar can consist of different principles. Figure 1.1 illustrates how an initial shape can transform into new variations using different types of transformation, such as addition, subtraction, rotation, and reflection.

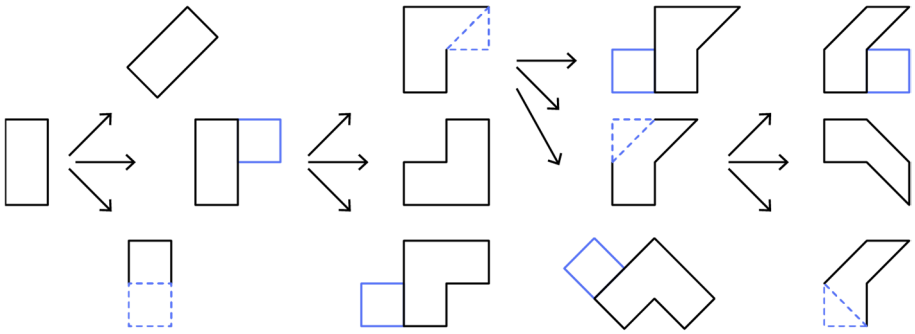


Figure 2.6 – Conceptual example of how shape grammar rules can create variations from an initial shape.

2.2 Optimization

Optimization is the act of obtaining the best result under given circumstances. Another relevant definition states that: optimization is the process of finding the variables that will provide the maximum or minimum goal value [52]. The general mathematical formulation of this statement can be formulated as:

$$\text{Find } \bar{X}=\begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \text{ which minimize or maximize f(X)} \tag{2.1}$$

Subjected to the constraints:

$$\begin{aligned} g_j(\bar{X}) &\leq 0 \quad j=1,2,\ldots,m \\ l_k(\bar{X}) &= 0 \quad k=1,2,\ldots,p \end{aligned} \tag{2.2}$$

Where:

$\bar{X}$	n-dimensional design vector
$g_j(\bar{X})$	j-dimensional inequality constraint
$l_k(\bar{X})$	k-dimensional equality constraint

The aim of structural design has always been to find optimal solutions. However, only the simplest problems can be solved analytically using differential calculus, which originated with Newton and Leibniz. More complex problems can be solved using numerical methods primarily developed in the middle of the twentieth century [52]. Even so, these methods also fall short when dealing with non-linear, non-continuous, and non-differentiable problems, which often is the case in structural engineering and CD. Conversely, these problems may be well suited for meta-heuristic optimization algorithms, which also can be combined with Machine learning methods to provide an effective approach to finding optimal solutions. Both concepts will be introduced in the following section.

2.2.1 Machine learning and optimization algorithms

Machine learning (ML) is a subfield of Artificial Intelligence (AI), and the terms are often used interchangeably. ML is defined as the ability of computers to learn and to find approximate functions without explicitly being programmed to do so [54]. ML can be divided into different categorizations, two of which are relevant to define concerning this project:

- Supervised learning vs. Unsupervised learning
- Classification vs. Regression

Supervised learning refers to the use of labeled data to train the algorithm to predict new data. In contrast, unsupervised learning involves finding patterns and relationships in unlabeled data without guidance [51]. Regression- and classification algorithms are both defined as supervised learning algorithms within the field of machine learning. The main difference between these two types of algorithms is the output they produce. A regression algorithm will output a real number representing a price, fitness value, size, and more. A classification algorithm, on the other hand, produces a categorial output. This output could be a simple binary, for example, “true” or “false,” or a multi-categorical output, such as “dog," “cat,” or “horse” [51].

In structural engineering, regression and classification ML models can be combined with meta-heuristic optimization algorithms to enhance the optimization process. This process is done by creating ML models that can approximate the response of computationally expensive objective functions. This concept is also known as surrogate modeling and is further detailed in section 2.4. The optimization algorithm can then use these ML models for fast evaluation and therefore achieve an increased exploration of the solution space with reduced computational cost.

Optimization algorithms can be divided into numerous categories based on their characteristics and approaches to optimization, as detailed in [55] [52] [51]. Specific categorizations will even intersect. The most relevant categorizations for this project are:

- Classical vs. Meta-heuristic
- Heuristic vs. Meta-heuristic
- Deterministic vs. Stochastic
- Population-based vs. Non-population based

Classical optimization refers to the numerical methods that originated with Newton and Leibniz. These methods are useful in finding the optimum solution for continuous and differentiable functions. Since most complex problems do not confine to these requirements, the classical optimization techniques have limited scope in practical application [52].

In contrast, meta-heuristic algorithms can be applied to optimization problems with little to no knowledge of the underlying problem. They can therefore be regarded as black-box solvers, also known as a general-purpose algorithms [51]. There are different properties that characterize metaheuristic algorithms [52] [56]: (i) they use stochastic principles to guide the search process. (ii) they all tend to find the global optimum solution and are most likely to find an optimum solution, but not necessarily all the time. (iii) they use strategies that imitate the behavior of natural phenomena like evolution, annealing in metallurgy, or imitating the behavior of certain species like ants, bees, or birds. (iv) They use higher-level heuristics to guide the search.

A heuristic can be defined as a rule or process used to approximate an answer. In contrast to a meta-heuristic algorithm, a heuristic algorithm is problem-dependent, meaning the algorithm exploits the underlying patterns of a given problem type. Also, a heuristic algorithm does not provide any information on the quality of an answer. The advantages of heuristic algorithms lie in their speed in reaching an approximate solution.

Meta-heuristic algorithms use higher-level strategies to provide a general framework of rules or heuristics to guide the search for a solution. Hence the name meta-heuristic, as it combines multiple heuristics into an iterative strategy that improves the answer by using a ranking procedure [57] [51].

A deterministic algorithm refers to an algorithm that uses predefined and rigid rules to navigate a solution space. A deterministic algorithm will always produce the same output given the same input as parameters. On the other hand, a stochastic algorithm uses randomness in its search process, making it effective in searching for solutions in complex non-linear, non-differentiable solutions spaces.

Non-population-based algorithms search the local region around an initial starting point in the solution space and will quickly converge toward a solution. However, these methods are unlikely to find the global optimum if the initial starting point is far from the initial starting point.

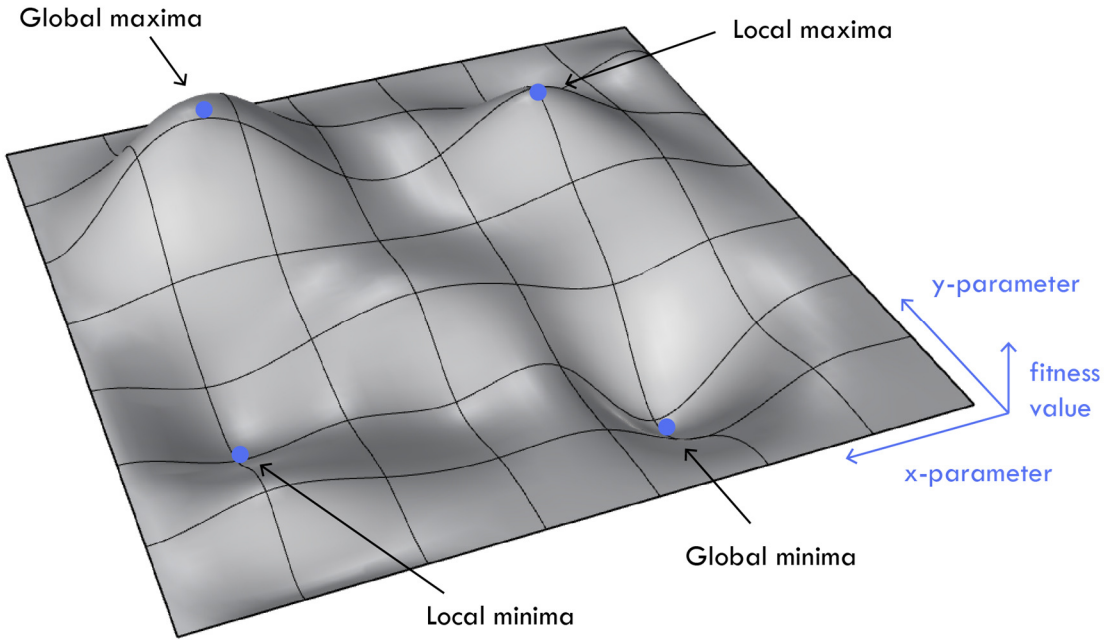


Figure 2.7 – Illustration of global and local extrema

Conversely, population-based algorithms are designed with mechanisms that enable them to escape local optima areas and are thus more likely to find the global optimum. However, they are more computationally expensive than local optimization algorithms and, therefore, slower.

Thus, the choice of strategy depends on a trade-off between computational cost and the accuracy of the final solution. This concept of global and local extrema is illustrated in Figure 2.7.

There are several other possible categorizations based on the nature of the optimization algorithm, and the categorization can also be based on the specific optimization problem, such as:

- Continuous vs. Discrete
- Single-objective vs. multi-objective
- Constrained vs. Unconstrained.
- Binary-, Permutation-, Value-encoding

Many more categories exist within the optimization algorithms, and they are sometimes even intersecting. This variety of options makes it nearly impossible to construct one cohesive overview. Figure 2.8 provides a simplified overview with examples of relevant algorithms. It is noted that many of these algorithms also contain different sub-versions. Combining optimization algorithms or parts of them into a larger optimization framework is also possible.

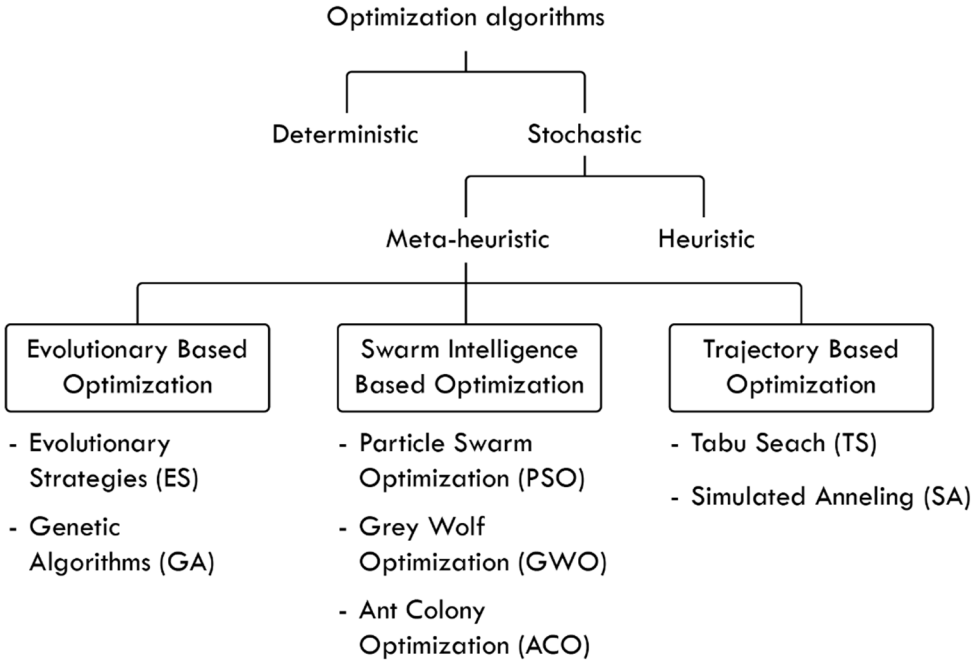


Figure 2.8 – Simplified overview of relevant optimization algorithms

2.2.2 Multi-objective optimization

Optimizing multiple objectives on a given problem is often relevant in structural engineering. The general mathematical formulation of optimizing with multiple objectives can be described as [52]:

$$Find\ \bar{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} which\ minimize\ or\ maximize\ F = \begin{Bmatrix} f_1(\bar{X}) \\ f_2(\bar{X}) \\ \vdots \\ f_k(\bar{X}) \end{Bmatrix} \tag{2.3}$$

Subjected to the constraints:

$$g_j(\bar{X}) \leq 0 \quad j = 1,2,\dots,m \tag{2.4}$$

Where:

$\bar{X}$	n-dimensional design vector
$g_j(\bar{X})$	inequality constraint

The objectives can often be conflicting, meaning that an improvement in one objective leads to a deterioration in another. The presence of conflicting goals makes it impossible to choose a solution that satisfies all the objectives in the best possible way. Thus, choosing a solution will require the definition of a satisfying trade-off between them.
In Multi-objective optimization, this task is defined as ranking, and the ranking methods can be categorized based on how they evaluate a solution. The most common ranking categories are dominance-based, aggregation-based, and indicate-based. Dominance based-ranking is the most widely used [55], and some of the most representative algorithms in this field are the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [58], the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [59] and the Fast Hypervolume-Based-Many-Objective Optimization Algorithm (HypeE) [60]. The dominance-based approach uses the concept of Pareto optimal solutions, also known as a Pareto front. A solution is Pareto optimal if it dominates all other solutions in at least one objective category. In other words, a feasible solution $\bar{X}$ is called Pareto optimal, if there exists no other non-dominant solution $\bar{Y}$ such that:
$f_i(\bar{Y}) \leq f_i(\bar{X})\ for\ i,j = 1,2,\cdots,k$ with $f_j(\bar{Y}) < f_i(\bar{X})$ for at least one j [52], where k denotes the number of objectives.

A Pareto front can visualize the trade-off between objectives and help the user make informed decisions. Figure 2.9 visualizes the concept of the Pareto front.

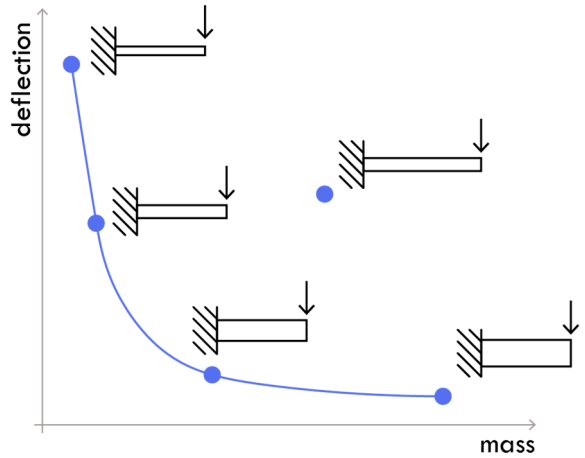


Figure 2.9 – Four Pareto-optimal solutions and one non-optimal solution. Recreated from [41].

Using a Pareto front to select a solution can also be described as ideal-based multi-objective optimization [55]. This approach contrasts with a preference-based multi-objective optimization strategy, simplifying the problem into a single-objective optimization problem by summarizing the different objectives with predefined weights. This method is known as the weighted sum method, or scalarization, and can be described mathematically as

$$F_{sum} = F_1 \cdot w_1 + F_2 \cdot w_2 \cdots + F_n \cdot w_n \tag{2.5}$$

Where:

F_n	Objective function
w_n	Weighting factor
F_{sum}	Summarized objective function

This simplification is a common strategy; however, there are multiple problems related to this approach. The main issue is the loss of information. Multi-objective optimization will often consist of conflicting goals, and there will not exist an optimal solution but a series of Pareto optimal solutions with each their advantage. Reducing multiple objectives into a single scalar value will make assessing the trade-offs between solutions hard. Moreover, the weighting of the objectives will be arbitrary and subjective as there is no precise way to evaluate the optimal trade-off between the objectives [55]. The resulting solution may not accurately reflect the preference or priorities of the decision-maker, and it may not even be Pareto-optimal. While a preference-based approach is not necessarily inappropriate, this project will employ an ideal-based approach for multi-objective optimization problems because of the aspects mentioned earlier and because it is deemed essential to engage the user in the decision-making process.

2.3 Genetic algorithm

A Genetic Algorithm (GA) is a meta-heuristic optimization algorithm inspired by the theory of evolution. It imitates the concepts of natural selection, crossover, and mutation, first proposed by Charles Darwin in his book *On the Origin of Species* [61]. The GA was developed by John Holland in 1975 [62] and is part of the evolutionary algorithm family. Other strategies include evolutionary programming and evolutionary strategies. They are based on the same principles, and in practice, it is hard to distinguish between them [63] [52]. The terminology used in GAs is derived from biology and defined in Table 2.1.

Table 2.1 – Basic terminology used in GA.

Terminology	Definition
individual	a representation of a candidate solution.
population	set of individuals.
gene	a feature of information describing an individual.
chromosome	set of genes representing an individual.
parent	individual selected for reproducing.
child	the offspring of two selected parents.
fitness	a measurement of an individual's quality in relation to defined performance objectives.

mating pool	a set of individuals selected for reproducing.
genotype	genetic representation of a solution.
phenotype	visual representation of a solution.

Genetic algorithms are very flexible and robust and accept almost any problem, as they can operate as a black box optimization algorithm, which means that the algorithm only operates with the input and corresponding output of a given function. A GA can be applied for multi-objective optimization. Unlike a gradient descent algorithm that can quickly get stuck in a local extrema point, a GA's stochastic mechanisms will balance exploitation and exploration in its search process. The basic framework of a GA is visualized in Figure 2.10.

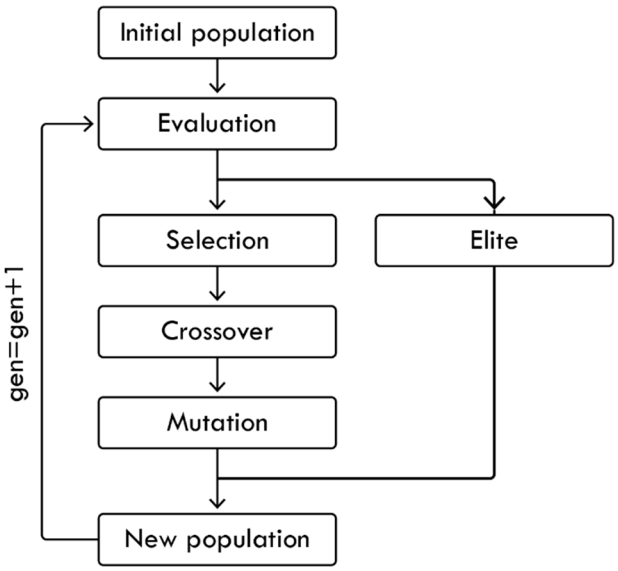


Figure 2.10 – The basic procedure of a GA

The basic procedure in a GA starts with the initialization of the first population, which is most often randomly generated. Every individual in the population is evaluated, and the "fittest" individuals are selected to reproduce. These individuals are then paired up, and their genetic data are merged to form the next generation of solutions. To add some randomness, a certain percentage of the solutions will have part of their genes mutated. The "elite" mechanism will retain a fixed number of the best-performing individuals for each generation to save their genetic data from any alterations due to the crossover or mutation mechanisms. These elite individuals are copied directly into the next generation, and this initiative will avoid decreasing the quality of the best solutions during the process. Generally, an elitism scheme will improve the algorithm's efficiency significantly [64] [65], and studies have shown that algorithms with elitism outperform algorithms without [58].

2.3.1 Selection

There are various selection methods in genetic algorithms, but their common purpose is to ensure the survival of genetic data, which will lead to optimal solutions. The selection process involves ranking the solutions based on their fitness. The individuals with the highest fitness are more likely to be selected for reproduction, while individuals with lower fitness are more likely to be discarded. This process can gradually improve the fitness of the solutions over multiple generations and will eventually result in convergence toward optimum solutions.

The simplest strategy involves an elitist ranking, where a certain percentage of the best solutions are selected for the mating pool. This method will quickly converge, but it comes at the cost of exploration. Most strategies involve a certain degree of randomness, and some of the most common strategies are roulette wheel selection and tournament selection.

The concept of roulette wheel selection is illustrated in Figure 2.11 (a). The wheel will spin randomly and collect the individuals needed for the mating pool. A fitter individual will demand a larger proportion of the wheel; this increases its chances of being selected. At the end of the selection process, some individuals will have been selected several times, and others will not have been chosen at all [65]. The selection probability p for a given individual i can be defined as:

$$p(X_i) = \frac{f(X_i)}{\sum_{j=1}^{n} f(X_j)} \tag{2.6}$$

Where n denotes the number of individuals in one generation, and f denotes the fitness of a given function. This method does not exclude any individuals from ever being selected but gives them a chance for survival proportional to their fitness value. Thus, this roulette selection prioritizes search space exploration at the expense of convergence time.

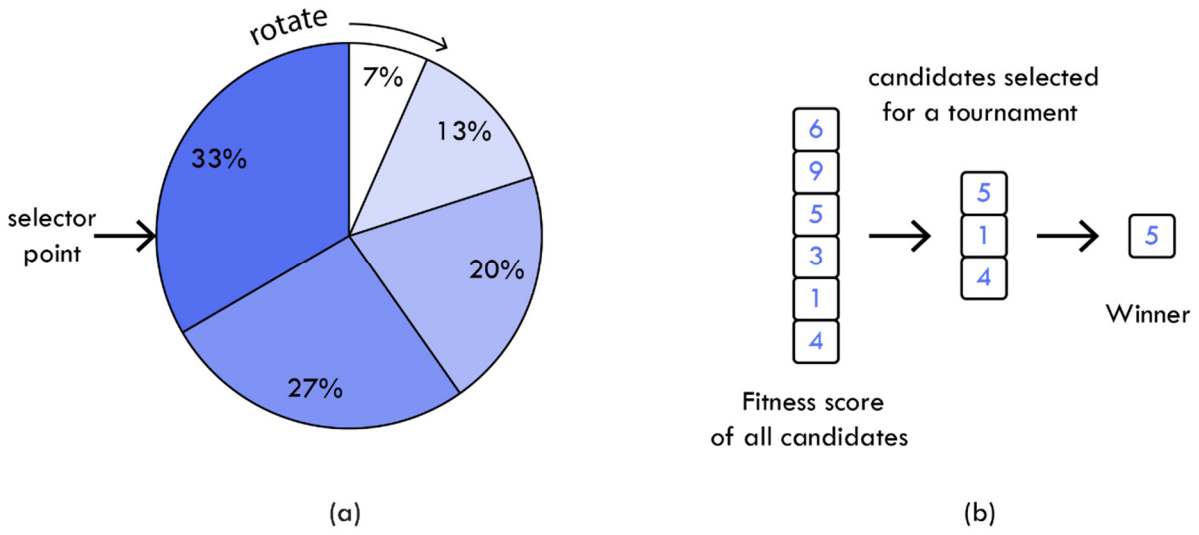


Figure 2.11 – (a) illustration of roulette wheel selection. The percentage illustrates the selection probability based on the fitness value. (b) Illustration of the tournament selection procedure.

The concept of tournament selection is illustrated in Figure 2.11 (b). The method works by randomly selecting x number of individuals from the population to participate in the tournament. The y number of best individuals is then selected for the mating pool. This process is repeated until the desired number of individuals in the next generation is reached. Adjusting the number of participants and winners will change the selection pressure. Decreasing the number of participants and increasing the number of winners will improve weaker candidates' chances of getting selected, so the selection pressure can be used to adjust the balance between exploration and exploitation.

2.3.2 Crossover

Following the selection phase, the crossover process, also known as recombination, can commence by randomly pairing individuals from the designated mating pool. Crossover involves the combination of chromosomes from the selected parents to produce offspring that exhibit traits of both predecessors. The crossover rate determines the probability of a pair reproducing, with pairs not selected for reproduction being transferred to the subsequent generation. Selecting an appropriate crossover rate requires a delicate balance between preserving the traits of successful solutions and enabling the exploration of the search space [65]. The crossover operator is traditionally considered as the "core" of the GA because it is the leading cause of variation and innovation of candidate solutions [64]. There are several methods for utilizing the crossover mechanism; two of the most commonly used for binary- and real-value encoding are One Point Crossover and Two Point Crossover. Other methods, such as Ring Crossover, Uniform Crossover, and simulated binary crossover, are defined in [17] [63] [66] [67].

One point Crossover works by randomly defining a position between the first- and the last gene in the chromosome. The process results in the creation of two new chromosomes by exchanging all the elements between the selected position. The concept of the One Point Crossover method is illustrated in Figure 2.12.

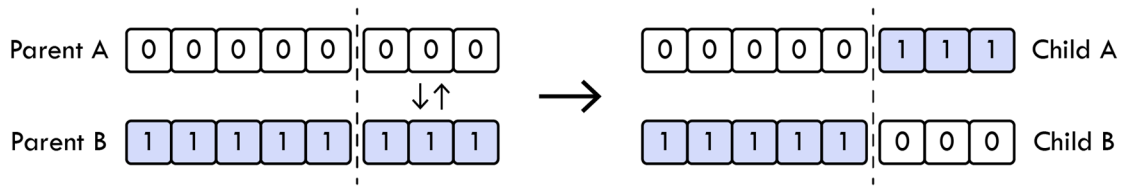


Figure 2.12 – The concept of One Point Crossover

Two Point Crossover operates similarly but with two division points instead of one. Two Point Crossover works roughly the same but with two division points. The two division points define a subset of the chromosomes, which is then exchanged between the two parent solutions to create the offspring. The concept of Two Point Crossover is illustrated in Figure 2.13.

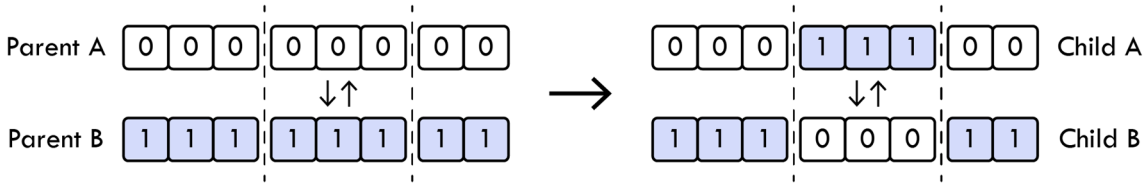


Figure 2.13 – The concept of Two Point Crossover

Crossover operations in permutation encoding problems are more complicated than in binary- and real-value encoding. The exchange of genetic sequences must be performed intelligently, promoting diversity while maintaining a valid permutation structure. Standard crossover methods in permutation encoding include Ordered Crossover (OX), Partially Matched Crossover (PMX). Other methods, such as Edge Recombination and the Cycle Crossover Rule, are defined in [68] [46].

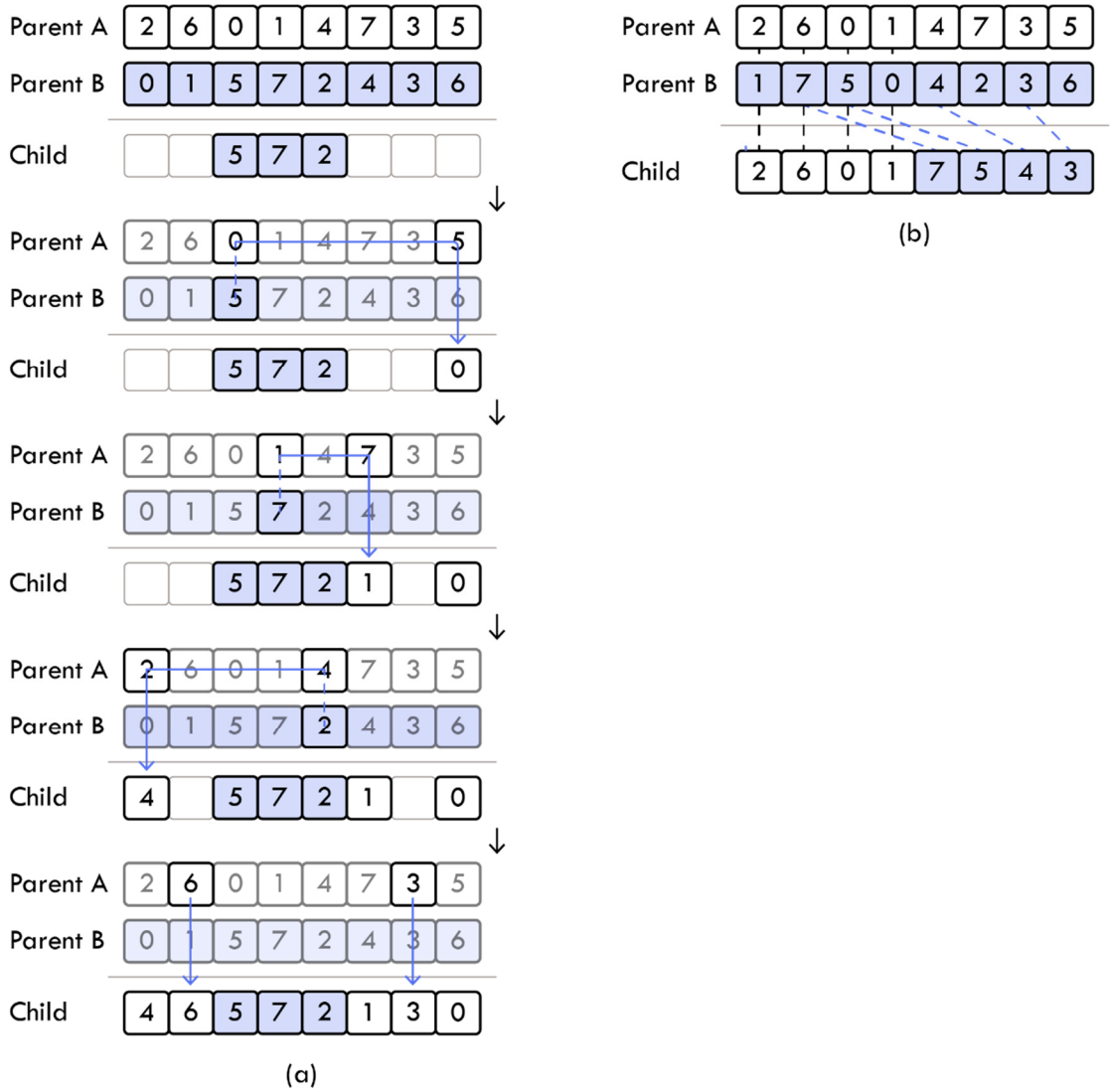


Figure 2.14 – (a) Illustrative example of the PMX procedure. (b) Illustrative example of the OX procedure.

crossover point is then placed in the offspring without alterations. The remaining elements from the second parent are then placed in the offspring in the order in which they appear. This process is illustrated in Figure 2.14 (b). The process is then repeated to create a second offspring but with the roles of the parents reversed.

PMX operates by selecting a subset of elements in parent B and transferring it to the offspring. Then each value replaced in parent A is moved to the corresponding location of the value in which it was replaced with. The remaining elements in parent A are then moved directly to the offspring. The process is again repeated to create offspring A but with the roles of the parents reversed. The process is best illustrated graphically as in Figure 2.14 (a).

2.3.3 Mutation

Sometimes the diversity generated through the crossover operator may prove insufficient, putting the process at risk of converging at a local optimum. The mutation operator is introduced to counteract this problem by enhancing the diversity of the population by introducing genetic variations to the chromosomes of different individuals. This diversification is achieved by substituting specific entries in the chromosome with random values that fall within the permissible range of the relevant variables. A simple concept of mutation in binary- and real-value encoding is illustrated in Figure 2.15. The mutation process can be regulated through the adjustment of two mutation parameters. The mutation probability determines the likelihood of a gene undergoing a mutation and is generally set to a low value. The mutation rate determines the extent to which the value could vary within a given interval. A low mutation rate results in minimal alterations, while a high mutation rate results in more substantial changes [65]. The mutation- probability and rate can also be adjusted according to the fitness using Adaptive Mutation. The average fitness is calculated, and the default mutation parameters of a given chromosome are then adjusted to provide a higher mutation rate for low-performing individuals and vice versa for high-performing individuals.

Additionally, the mutation probability and rate can be dynamically adjusted according to the fitness value through Adaptive Mutation. This adjustment involves the calculation of the average fitness, which is then used to modify the default mutation parameters for a given chromosome. High-performing individuals receive a lower mutation rate, while low-performing individuals receive a higher mutation rate, thus promoting greater diversity and potential for improvement.

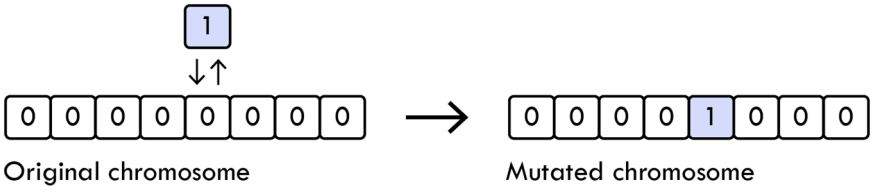


Figure 2.15 – Principle of mutation in binary- and real value encoding

There are specific mutation operators for permutation encoding problems. Some of the most populator mutation operators are Swap Mutation and Scramble Mutation. Swap mutation works by selecting two random genes in the chromosome and then swapping the values, so they exchange places. This procedure can be done more than once. The concept of Swap mutation is illustrated in Figure 2.16.

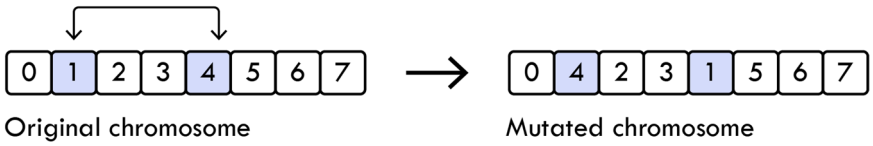


Figure 2.16 – The concept of Swap Mutation

Scramble Mutation operates by identifying a subset of values within a chromosome, which are then randomly shuffled to promote diversity. The concept of Scramble Mutation is illustrated in Figure 2.17. Other notable mutation methods include Displacement Mutation, Insertion Mutation, and Inversion Mutation.

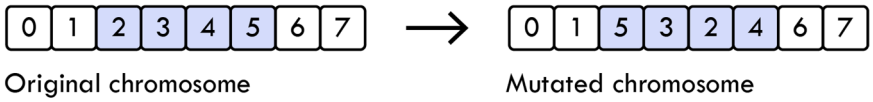


Figure 2.17 – The concept of Scramble Mutation

2.4 Surrogate modelling

Structural engineering is often a complex endeavor due to the frequent requirement for computationally heavy simulations such as finite element modeling, computational fluid dynamics simulations, and other resource-demanding tasks. Consequently, the optimization process becomes increasingly complex and may require numerous simulations to converge toward optimal results. Surrogate modeling (SU) offers a possible solution to reduce the computational burden as discussed in section 2.2.1. Surrogate modeling is a sub-field within ML and can be defined as *constructing an approximation of the response of a given function or model based on a limited number of expensive simulations*. A surrogate model is basically a "model of a model." It describes the relationship between inputs and outputs and works without any knowledge of the inner mechanism of the model it is trying to emulate [48]. The concept of a surrogate model is illustrated in Figure 2.18.

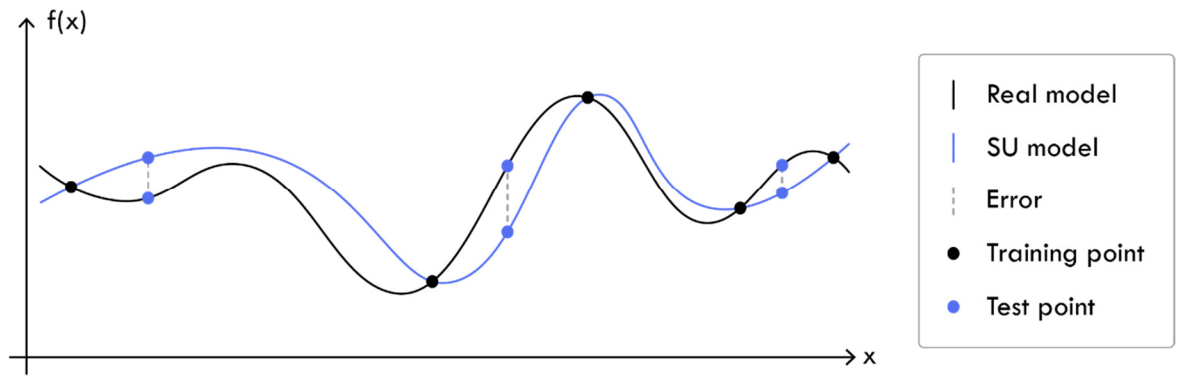


Figure 2.18 – Illustration of a one-variable surrogate model

Surrogate modeling is well suited for various applications in structural engineering, for example, in the conceptual design phase, as it can support the rapid-fire brainstorming sessions that are typical for conceptual design [13]. In this phase, there is no need for high-fidelity calculations which prevents any effective use of optimization algorithms.

Surrogate models, also known as metamodels, response surface models, and approximation models, is a field within ML that contains many sub-categories. Some common types of surrogate models include:

- Artificial Neural Network (ANN)
- Kriging
- Polynomial Regression (PRG)
- Radial Basis Function (RBF)
- Multivariate Adaptive Regression Splines (MARS)
- Gaussian Process (GP)

- Random Forest (RF)
- Support Vector Machines (SVM)

Surrogate models can also be categorized based on their different characteristics, such as:

- Interpolating- and non-interpolating models
- Regression- and classification models
- Global- and local models
- Single-output- and multiple-output models

Each SU method has its specific strength and limitations with regard to the simulation model it is trying to emulate. One of the main categories is interpolating- and non-interpolating models [69]. Non-interpolating models, such as ANN and PRG, minimize the sum of squared error between some predetermined functional form and the sampled data points. While non-interpolating models may lead to simple and interpretable functional forms, they may not be flexible enough to emulate highly non-linear correlations [70]. Interpolating methods such as Kriging exhibit increased flexibility by incorporating different basis functions (or kernels) built to predict the training points [69]. Because of their flexibility and precision, Kriging models are used in many projects, but they still have limitations. They are sensitive to dense sampling as it can overfit the model [71], and most importantly, they perform poorly on high-dimensional problems. In this context, "high dimensional" refers to a scenario with more than approximately twenty variables, as there is no universally agreed-upon definition of what constitutes high dimensionality.

Surrogate models can also be divided into regression- and classifications models and global- and local models [72]. Global models try to emulate the entire search space, while local models only emulate high-performing sub-regions of the search space. This concept is very much related to hierarchical surrogate modeling. It can be challenging to construct accurate Surrogate models to emulate high-dimensional, non-linear functions, especially if they emulate several objective values in the same surrogate model. A possible solution is splitting a surrogate model into different models in a hierarchical structure to simplify complexity.

A simplified concept of a hierarchical structure of SU models is illustrated in Figure 2.19.

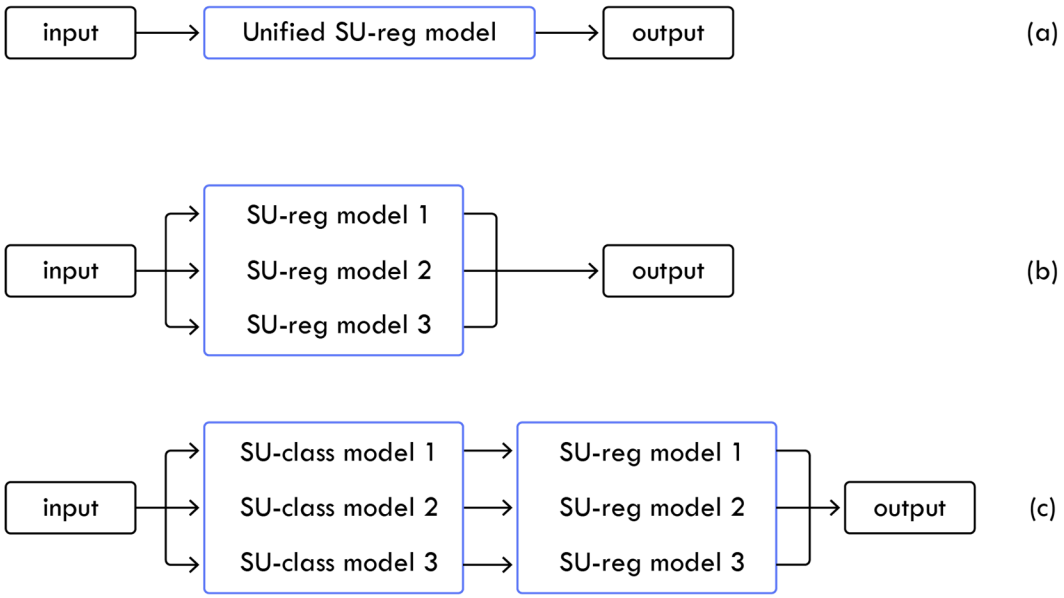


Figure 2.19 – (a) A simple surrogate model with multiple outputs. (b) The same surrogate model parallel surrogate models, each representing an objective value. (c) a hierarchical structure of surrogate models in series and in parallel.

Figure 2.19 (a) illustrates a single SU model that emulates the entire response spectrum and all three objectives. In Figure 2.19 (b), this same model is split into three separate SU models. This change has the advantage that a single surrogate model only needs to emulate one objective and therefore has better conditions for making accurate predictions. Another advantage is that more surrogate model types are available, as not all surrogate methods can generate multiple outputs in the same model.

However, in cases where a problem is highly non-linear and complex, this may not be sufficient. In Figure 2.19 (c), this same model is split into a hierarchical structure with six models that work in series and parallel. The first SU model in the series can be created as a classification model with a single purpose. This model should identify if a solution is expected to be in the top quartile percentile. The model acts as a gatekeeper, so the following regression model only needs to focus on these promising regions. This approach can reduce the search space the model has to emulate and avoids high non-linear response areas typically found in low-performing areas. This concept, also known as local SU modeling, can be as refined as necessary.

2.4.1 Sampling

The process of using simulations to construct a SU model is also known as sampling. Sampling is the initial step in any surrogate modeling framework. A set of multi-variables is defined, and for each sample, the corresponding objective value is computed using a high-fidelity model. Ideally, a surrogate model should be as precise as possible with as few samples as possible. Therefore, numerous sampling strategies, also called sampling plans, have been developed to maximize the information obtained with a minimum number of samples. These strategies can roughly be categorized as one-shot, sequential, exploitative, and exploration/space-filling, as illustrated in Figure 2.20. An exploitative strategy focuses on sampling in high-performing areas, while an explorative strategy samples in sparsely populated areas.

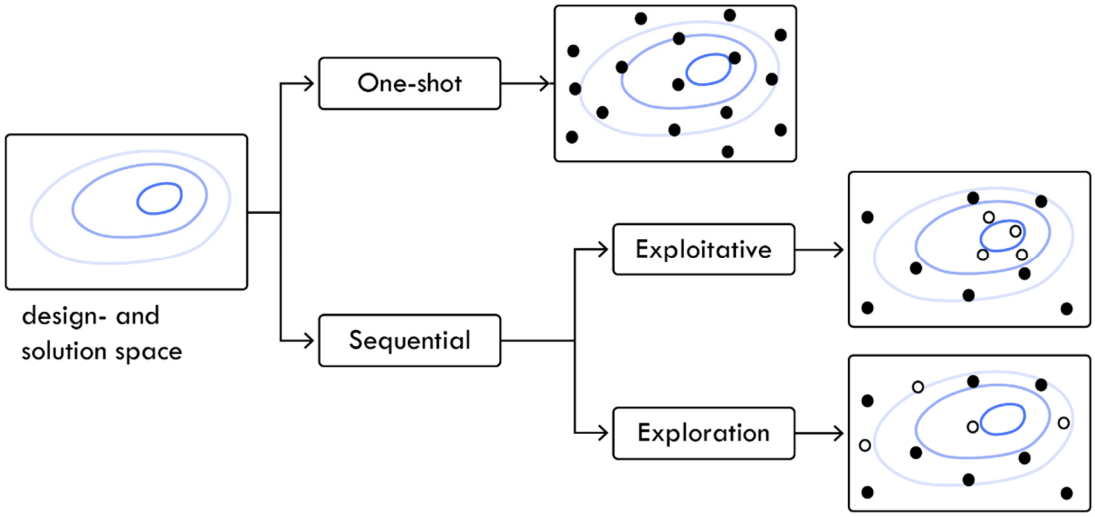


Figure 2.20 – The basic sampling strategies. Recreated from [59]

One-shot sampling generates all the sample points in one single step. This strategy is easy to implement and provides good coverage of the design space [73]. However, there is no way to determine the optimal sample size beforehand. Common one-shot sampling strategies include:

- Pseudo-random numbers
- Uniformly distributed numbers
- Random Latin hypercube
- Best Latin hypercube
- Sobol quasi-random
- Halton quasi-random

The pseudo-random numbers strategy is created using random number generator techniques. They are created to appear random from a statistical perspective, even though they are created using a deterministic algorithm. This sampling strategy tends not to perform well, as it is inherently not space-filling and can produce clusters in the design space. The uniformly distributed numbers strategy generates samples evenly spaced throughout the design space without any random elements, thereby maximizing the space-filling capabilities. However, this strategy also tends to perform poorly as the equally distanced samples can lead to systematic errors [48]. Typically, the most effective sampling strategies possess a combination of space-filling and stochastic properties, which are present in the Latin Hypercube, Sobol, and Halton sampling strategies. It is impossible to predict which sampling strategy will be the most effective for a given problem, and the choice also depends on the surrogate model. For instance, the Kriging model is particularly sensitive to sample clustering due to its reliance on Cholesky factorization. Therefore, the sampling strategy needs to consider this effect. In practice, the best approach is to compare the performance of each strategy through a comparative analysis. A more comprehensive description of these sampling methods can be found in [48] [71] [50] [49], which includes additional relevant sampling techniques.

Sequential sampling also referred to as adaptive sampling, sample enrichment, active learning, and dynamic training of surrogate models, is based on the concept of iteratively identifying new sample points in the most informative regions of the design space to reduce the number of samples required for accurate predictions. It is noted that informative regions can refer to sparsely populated regions or high-performing regions, also known as a global- and local search. A basic framework of sequential sampling is illustrated in Figure 2.21.

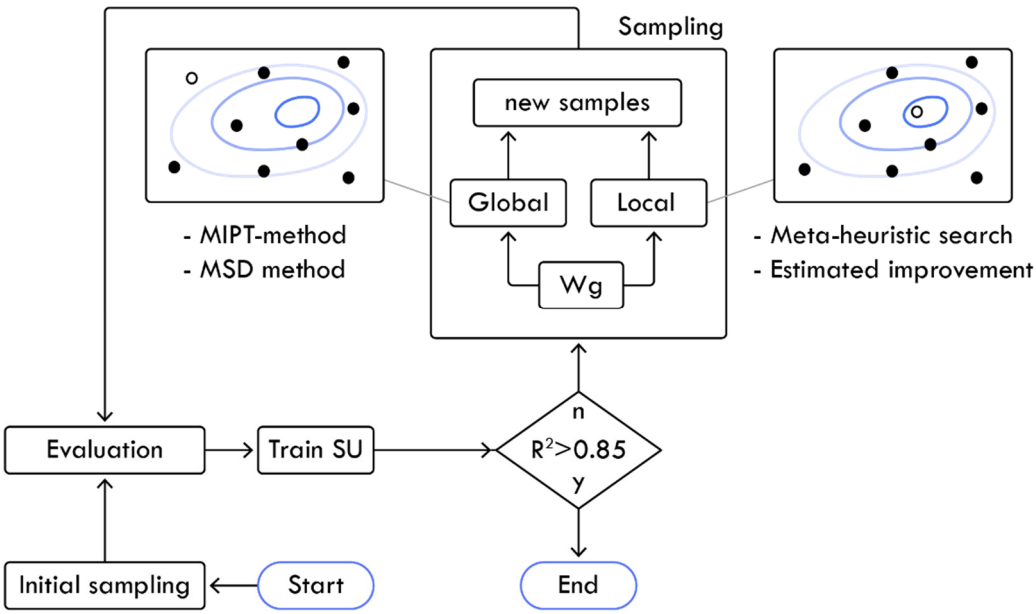


Figure 2.21 – Conceptual framework for a sequential sampling procedure

The process typically initiates with a one-shot sampling collection. These samples are evaluated and form the basis of the first training iteration of the SU model. If the accuracy of the model is not sufficient, the loop continues. New samples are generated through an adaptive sampling process. First, a weighting algorithm determines the ratio of new global and local samples. Different relevant weighting methods exist, as Decreasing-, Greedy- and Switch strategy [73] [74].

The new global samples are generated using a space-filling method suited for adaptive samplings, such as the MIPT method [75]. The local samples can be generated using a meta-heuristic optimization algorithm on the current SU model to identify high-performing regions and obtain samples from these regions. Another option is the Estimated Improvement method, explicitly designed for Kriging surrogate models. The sample collection process can be carried out either as a single-point or batch collection.

The collected samples are evaluated, and the SU model is updated. This process continues until the SU model reaches the designated level of accuracy.

2.4.2 Curse of dimensionality

The number of samples correlates with the prediction accuracy of a given SU model, and for high-dimensionality problems, we need more samples to achieve the same sample density. The number of samples directly affects the prediction accuracy of a surrogate model. For problems with a more significant number of variables, a higher sample density is required to achieve the same level of accuracy [48]. This statement is very intuitive. However, the extent of the required increase in sample density may not be immediately obvious. Suppose the sampling density per variable is denoted n, and k is denoted as the number of variables. In that case, the total required number of samples for reaching the same sample density can be calculated as n^k. For instance, only ten samples are needed in a one-dimensional problem with a required density of n=10. However, this number increased to 100 when adding one more variable to the problem, as illustrated in Figure 2.22.

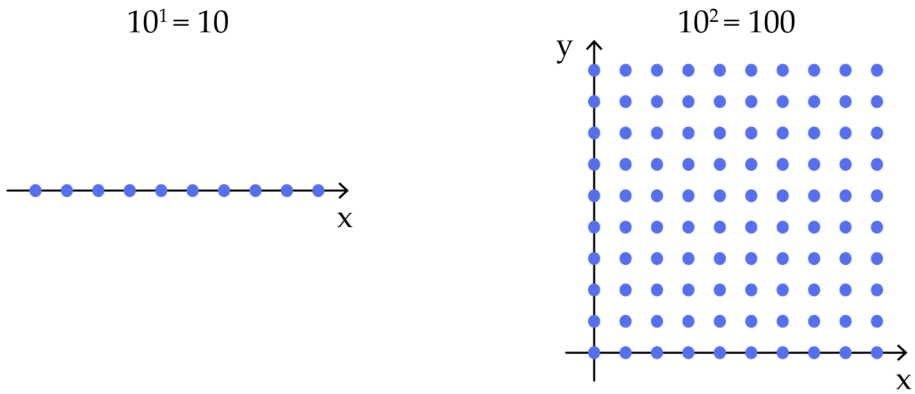


Figure 2.22 – Illustration of the ”Curse of dimensionality” going from one variable to two variables

Adding four more samples to a total of six variables, and we need a million samples instead of the original ten. This phenomenon is known as the "curse of dimensionality."

There are only limited options available to address this issue. One of the most straightforward solutions is simplifying the emulated model by reducing the number of variables [48]. More complex measures include the utilization of dimensionality reduction techniques, such as PCA, as mentioned in section [50]. Another option is to apply a hierarchical SU approach to decrease the required sampling density, as the SU model then only has to focus on high-performing regions [50].

2.4.3 Model validation

The accuracy of a surrogate model is assessed through a validation process. This valuation is done by evaluating its ability to make reliable predictions on new input, in other words, how well the model generalizes. This process commonly starts by splitting the available set of samples into a training set, which is used to train the SU model, and a validation set which is input-data the model has never encountered before [76]. If only a small dataset is available, then validation frameworks such as cross-validation and bootstrapping are available.

Numerous different error metrics are available to measure the prediction error, also denoted as the generalized error metrics. One of the most common error metrics is the correlation value R^2 which measures how well the predicted values correlate with the actual values. If an R^2 equals one, then there is a perfect correlation between the values and no error [77]. Forrester states that an R^2 above 0.8 indicates good prediction capabilities [48].

Table 2.2 lists some of the most common error metrics with the corresponding formula. A more comprehensive description and relevant error metrics can be found in [77] [78].

Table 2.2 – Common error metrics

Error metric	Formula
Mean Absolute Error (MAE)	$\frac{1}{m}\sum_{i=1}^{m}\left\|f_i - \hat{f}_i\right\|$
Mean Squared error (MSE)	$\frac{1}{m}\sum_{i=1}^{m}\left(f_i - \hat{f}_i\right)^2$
Root mean squared error (RMSE)	$\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(f_i - \hat{f}_i\right)^2}$
Correlation value R^2	$1 - \frac{\frac{1}{m}\sum_{i=1}^{m}\left(f_i - \hat{f}_i\right)^2}{\frac{1}{m}\sum_{i=1}^{m}\left(f_i - \bar{f}\right)^2}$

There are instances where one error metric can be misleading, and it can be hard to comprehend why a SU model performs as it does from just assessing quantitative data. Visualizing the prediction capability can provide greater insight into the mode's strengths and weaknesses and offers valuable information that can be used to improve it. One of the most used error visualization techniques for Surrogate modeling is the scatter plot showing predicted values versus real values. An example of this visualization is illustrated in Figure 2.23.

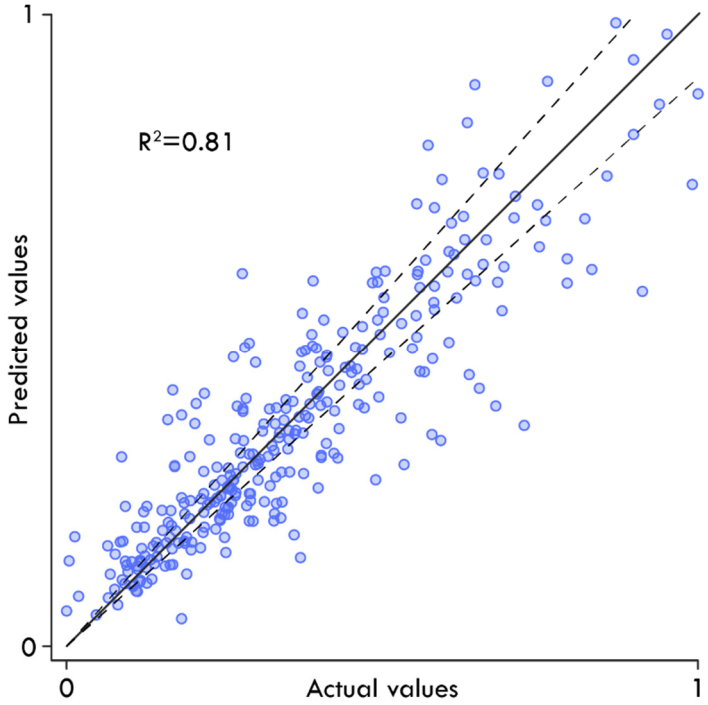


Figure 2.23 – Predicted values versus real values with a ±10% error margin.

2.4.4 Kriging

The kriging model was initially developed by the South African mining engineer Danie Krige in 1951 [79] and further developed by Matheron and Sacks et al. [48] and is one of the most widely used surrogate modeling methods in Engineering [76]. The Kriging method uses statistical interpolation to predict new data.

The prediction process relies on the mean and variance of the available training samples. These values allow the model to predict the value at a new point in the solution space by considering and utilizing information from all the surrounding data points.

The estimation of new data using the Kriging model can be described as a weighted sum of all the available sample data. The weighting factor is determined based on the spatial proximity between the new prediction point and each existing sample point. The weighting factor is derived from a basis function, and the Kriging model can also be considered part of

the RBF family. However, the basis function in a Kriging model is more flexible and can, therefore easier, emulate non-linear and complex data [77]. Although Kriging is more complicated than other radial basis function methods, it is, nevertheless, simply a sum of weighted basis functions [48].

The Kriging prediction function is formulated in equation 2.7, where a $\hat{f}(x)$ is the predicted value at location x [49] [48].

$$\hat{f}(x) = \hat{\mu} + \psi^T \Psi^{-1}(y - 1\hat{\mu}) \tag{2.7}$$

Where y is a vector of the sample points' function values and $\hat{\mu}$ is the estimated mean of the sample points' function value, and serves as a baseline for predicting the function value at new locations.

$$\hat{\mu} = \frac{1^T \Psi^{-1} \cdot y}{1^T \Psi^{-1} \cdot 1} \tag{2.8}$$

ψ^T is a vector of weights assigned to each sample point, which depend on the spatial relationship between the sample points and the prediction location x. Each weighting factor in ψ^T determines the influence of each basis function on the new prediction. The ith weight for the ith basis function is informed by the spatial relationship between sample points and the new prediction point, ultimately allowing the model to calculate a prediction as the sum of all sample contributions. The weighting factor is based on the spatial distance from point i to the new prediction point. The concept behind spatial weighting is that sample points close to the prediction point are more likely to provide information about the underlying pattern in the data and, thus, should be given more weight when making a prediction. Therefore, a minimal distance will result in $\psi_i \approx 1$ and reversely, a significant distance will result in $\psi_i \approx 0$ and consequently have very little influence on the final prediction. The ith basis function calculating this weight is formulated as follows:

$$\psi_i = cor\left[Y\left(x^i\right), Y(x)\right] = exp\left(-\sum_{j=1}^{k} \theta_j \left|{x_j}^i - x_j\right|^{p_j}\right) \tag{2.9}$$

Where k is the number of dimensions and p, and θ are nuisance parameters, respectively denoted as the smoothness- and width parameters. The nuisance parameters are adjusted through a meta-heuristic algorithm to maximize the prediction capability of the Kriging model. The nuisance parameter determines the shape of the basis function, and the effect of varying the parameters is best illustrated in Figure 2.24 and Figure 2.25.

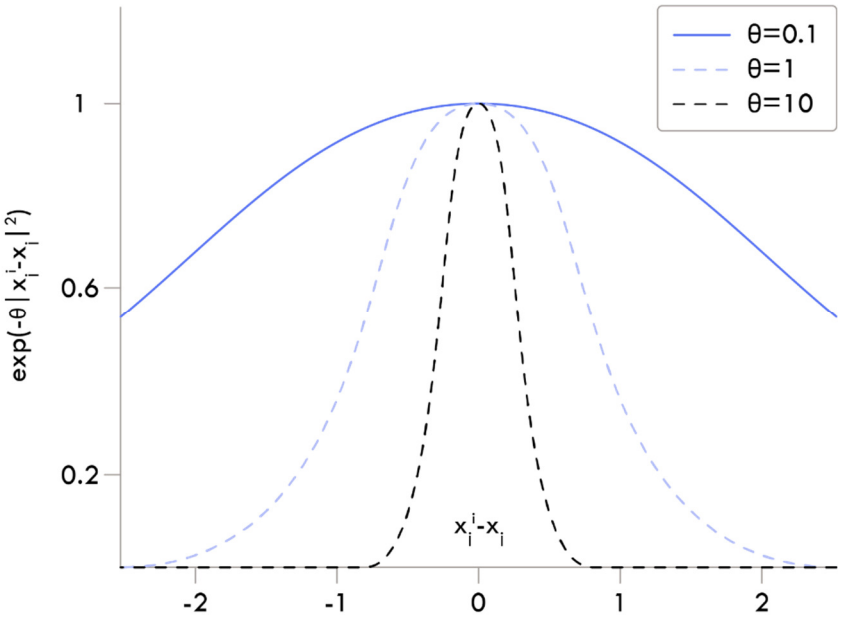


Figure 2.24 – Effect of varying θ values

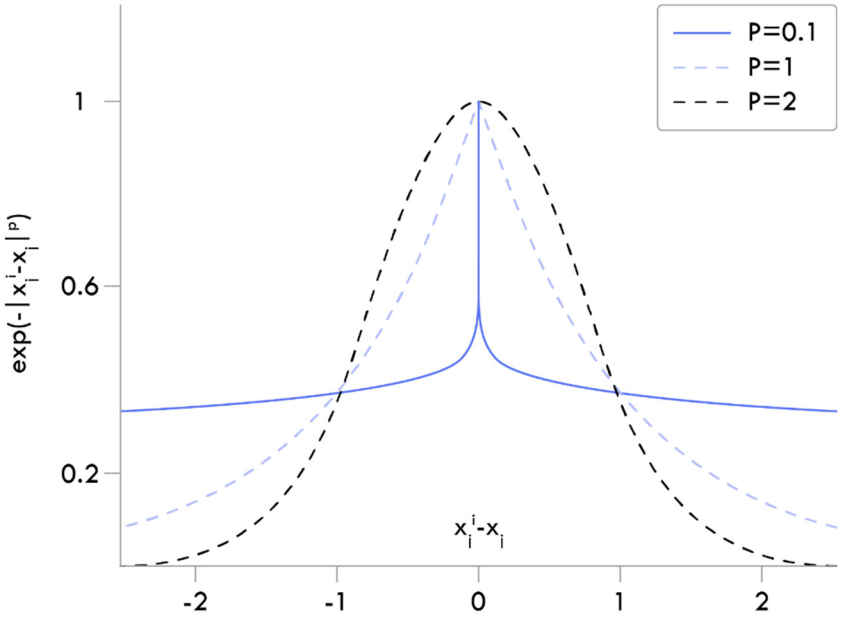


Figure 2.25 – Effect of varying P values

A vector of samples can be formulated as $X=\{x^1,x^2,\cdots,x^n\}^T$ where the corresponding values are formulated as $Y=\{y^1,y^2,\cdots,y^n\}^T$. The correlation between all these values is then calculated using equation 2.8 and stored in $n \times n$ correlation matrix, denoted as the Gram matrix, as shown in equation 2.9.

$$\Psi = \begin{pmatrix} cor\left[Y\left(x^i\right), Y(x^1)\right] & \cdots & cor\left[Y\left(x^i\right), Y(x^n)\right] \\ \vdots & \ddots & \vdots \\ cor[Y(x^n), Y(x^1)] & \cdots & cor[Y(x^n), Y(x^n)] \end{pmatrix} \tag{2.10}$$

The optimal choice of nuisance parameters is not known beforehand. They must be found through a sensitivity analysis. As previously mentioned, this is typically done using a meta-heuristic algorithm where the nuisance parameters are set as the variables, and the objective is to maximize the likelihood function defined in equation 2.10, which determines the likelihood of a given observation to be true. So, in other words, the prediction capability is optimized to construct a more robust SU model.

$$-ln(L) \approx -1\left(-\frac{n}{2}ln(\hat{\sigma}^2) - \frac{1}{2}ln(|\Psi|)\right) \tag{2.11}$$

Where:

$$\hat{\sigma}^2 = \frac{(y - 1\hat{\mu})^T \Psi^{-1}(y - 1\hat{\mu})}{n} \tag{2.12}$$

The Kriging model is widely recognized in literature, has shown good performance on different engineering problems, and can emulate highly non-linear problems. This ability is demonstrated in Figure 2.26, where the benchmark function Rastrigin is emulated using a Kriging model with 150 training samples.

However, the Kriging model also possesses some disadvantages as it is relatively complex to implement and does not perform well on high-dimensional problems. Additionally, if two sample points are too close to each other, the model may fail due to issues with Cholesky factorization.

The information presented in section 2.4.4 only provides a basic understanding of using Kriging as a SU model. There are additional important considerations for applying Kriging models, including sub-categories such as Universal Kriging. Furthermore, the prediction process may be enhanced by including noise parameters in the prediction process, so the model is not limited to solely interpolating through sample points. Adaptive sampling techniques have also been developed specifically designed for the Kriging mode. A more in-depth explanation of these and other aspects can be found in [49] [80].

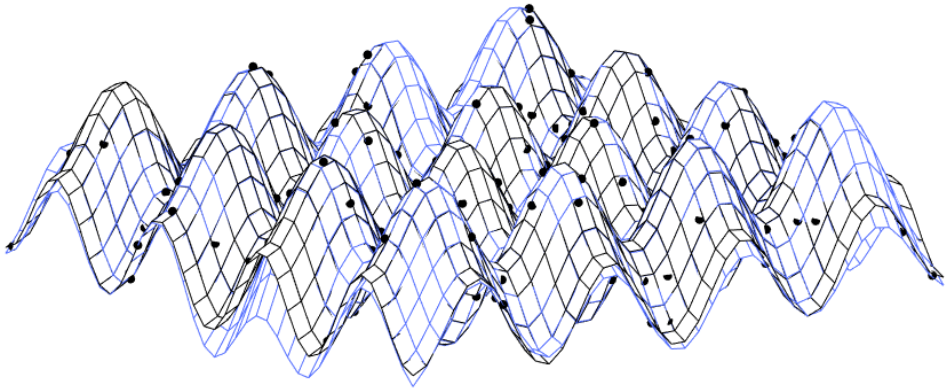


Figure 2.26 – Black indicates the real Rastrigin surface, blue indicates the corresponding surface plot emulated by a Kriging surrogate model with 150 training points.

2.4.5 Artificial Neural Network

An artificial neural network (ANN), referred to as a Neural Network, is an adaptive system that mimics the structure and function of the human brain. It uses interconnected neurons organized in a layered structure to learn from data. These features allow the neural network to identify patterns, classify data, and make predictions based on past experiences. Common type-variants of neural networks include:

- Multi-layered perceptron (MLP)
- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)
- Generative Adversarial Network (GAN)

The MLP network is also known as a feedforward network and will be explained in detail later. CNN is a deep-learning algorithm specifically designed for image recognition. RNN is a type of neural network designed to process sequential data where the output from the previous iteration is used as input for the current iteration [81]. GAN is a neural network that generates new data resembling the original data through a competition between a generator and a discriminator network [82]. A neural network can be categorized based on its task, characteristics, and structure, including:

- Classification vs. Regression
- Supervised learning vs. Unsupervised learning
- Deep vs. Shallow
- Generative vs. Discriminative

The distinction between classification and regression, as well as supervised and unsupervised learning, was established in section 2.2.1 and is likewise applicable in this context.

Neural networks can be classified based on their number of hidden layers. Shallow neural networks typically only consist of one or two hidden layers, while a deep neural network can consist of numerous layers [1]. Incorporating many layers can help the neural network effectively model highly non-linear data since a shallow network may lack the necessary complexity to emulate the training data accurately. However, it is important to note that deep neural networks are not inherently superior to shallow networks; their efficiency depends on the particular problem and available data. If an excessively complex deep neural network is employed for a simple problem, then the risk of overfitting increases, and the training time may be unnecessarily prolonged. A generative network is designed to generate new data samples based on the samples used for training. In contrast, a discriminative model makes predictions based on the input data. These methods, along with other relevant neural network methods and concepts, are described in detail in [81], [83], and [84].

This section outlines the fundamental properties of an MLP network, which can be considered a basic- or vanilla version of a neural network and among the most commonly used variants [1]. MLP is also referred to as a feedforward neural network with one or more hidden layers that uses an activation function to introduce non-linearity to its output. The activation function, also known as the transfer function, is an essential component of the neural network. Without an activation function, a neural would be a linear model where the output would be a weighted sum of its inputs, without any non-linear transformation [85].

The activation function determines when or to what extent a neuron will "fire." It accomplishes this by converting the sum of inputs, weights, and biases into another domain. Various activation functions exist; some use a hard threshold, where the output is zero unless the value exceeds a particular threshold. Other activation functions operate in the domain of $[-\infty, \infty]$, but many convert the values in the domains of [0,1] or [-1,1]. These restricted domains are acceptable as a linear transfer function can be incorporated in the output layer to properly re-scale the output into the appropriate domain.

Some relevant activation functions are listed in Table 2.3; the corresponding transfer functions are visualized in Figure 2.27.

Table 2.3 – Formula of relevant transfer functions

Transfer function	Formula
Tansig – Symmetric sigmoid transfer function	$\sigma(x) = \frac{2}{1 + e^{-2 \cdot x}} - 1$
Logsig – Logarithmic sigmoid transfer function	$\sigma(x) = \frac{1}{1 + e^{-x}}$
Satlins – Symmetric saturating linear transfer	$\sigma(x) = \begin{cases} -1, & x \leq -1 \\ x, & -1 \leq x \leq 1 \\ 1, & 1 \leq x \end{cases}$
Poslin – Positive linear transfer function	$\sigma(x) = \begin{cases} x, & x \geq 0 \\ 0, & x \leq 0 \end{cases}$

It cannot be known in advance which transfer function will be the most relevant for a given problem. There are certain transfer functions tailored to specific network types and issues. While some experiences can be drawn from similar issues, the choice of transfer function should be based on a sensitivity analysis. Additional transfer function methods and a more detailed description be found in [85].

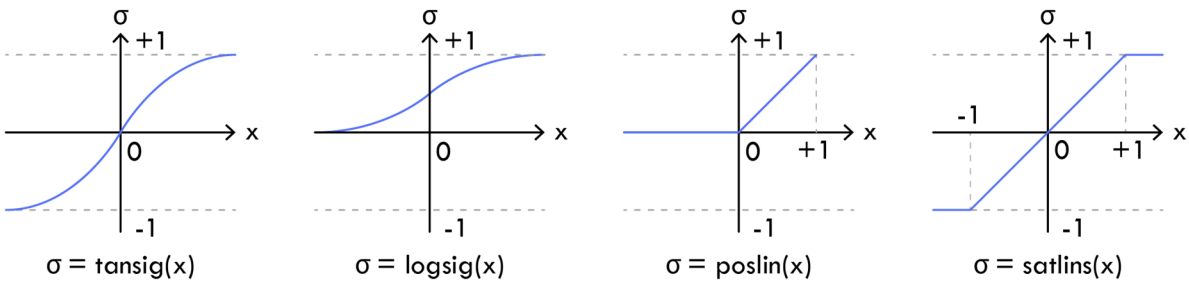


Figure 2.27 – Illustration of relevant activation functions

A feedforward neural network henceforth referred to as a neural network, consists of multiple layers that break down data into progressively simpler representations. A basic representation of the neural network's architecture is illustrated in Figure 2.28. A neural network is composed of an input layer, one or multiple hidden layers, and an output layer. Each layer encompasses a specific number of neurons, and all other neurons are interconnected with the preceding- and subsequent layers of neurons.

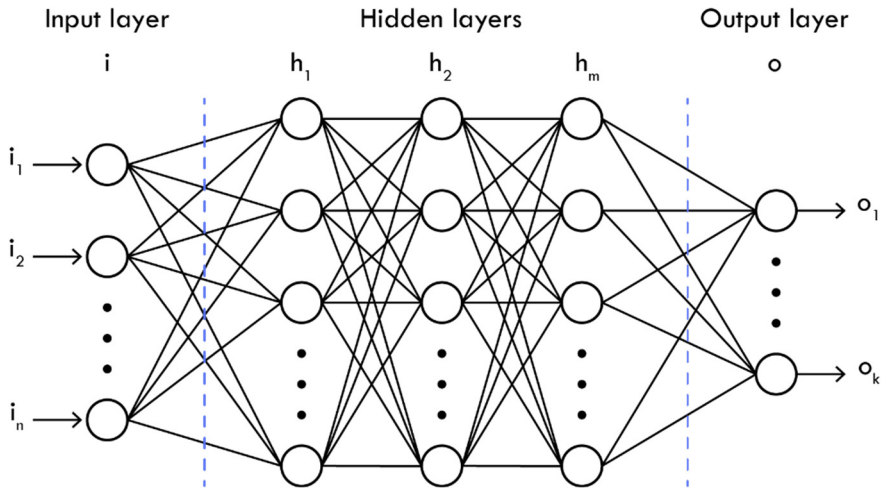


Figure 2.28 – A basic neural network architecture

The behavior of the network is defined by the connections between the neurons and the strength of these connections, which are represented by weight values and an associated bias value. These values or parameters are dynamically adjusted during the training process according to a predefined learning rule until the neural network correctly performs the desired task [81] [86].

If a neuron has low weight and bias values, it is unlikely to "fire," resulting in a minimal influence on the neurons in subsequent layers. Conversely, a neuron with a high bias and weight parameters will have a significant impact. A neuron can be considered an individual function with a simple summation of all input values multiplied by their corresponding weight, and finally, a bias value is added. The result of this summation is then passed to an activation function. This process is executed for all interconnected neurons throughout all layers. This process is illustrated in Figure 2.29 for one neuron.

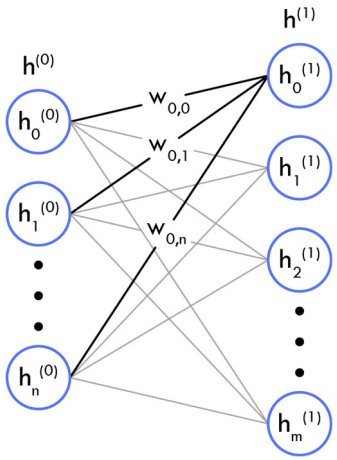


Figure 2.29 – Illustration of how input is transferred to one neuron.

The mathematical depiction of the summation in a single neuron is provided in equations (2.13) and (2.14). Expanding this methodology to all neurons between two consecutive layers can be described in matrix form, as shown in equation (2.15) [87], where σ represents the activations function determining the specific neuron's activity level.

$$h_0{}^{(1)} = \sigma\left(w_0{}^{(0)} \cdot h_0{}^{(0)} + w_1{}^{(0)} \cdot h_1{}^{(0)} + w_2{}^{(0)} \cdot h_2{}^{(0)} + b_0{}^{(1)}\right) \tag{2.13}$$

$$h_0{}^{(1)} = \sigma\left(\sum_{i=1}^{n} w_i \cdot h_i{}^{(0)} + b_0{}^{(1)}\right) \tag{2.14}$$

$$\begin{bmatrix} h_0{}^{(1)} \\ h_2{}^{(1)} \\ \vdots \\ h_m{}^{(1)} \end{bmatrix} = \sigma\left(\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,m} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,0} & w_{n,1} & \cdots & w_{n,m} \end{bmatrix} \cdot \begin{bmatrix} h_0{}^{(0)} \\ h_1{}^{(0)} \\ \vdots \\ h_n{}^{(0)} \end{bmatrix} + \begin{bmatrix} b_0{}^{(1)} \\ b_1{}^{(1)} \\ \vdots \\ b_m{}^{(1)} \end{bmatrix}\right) \tag{2.15}$$

The training can commence after the architectural structure of the neural network has been established. This training process involves the utilization of all the available labeled training samples, similar to the training of the Kriging model or any other SU model. The process is initiated by randomizing the weights and biases; the prediction output is then calculated and compared with the actual output. The squared difference between the predicted and actual output is then computed, and this value represents the prediction error or cost. The prediction error is then used to calculate the gradients of the weights and biases via a procedure known as backpropagation. This method is feasible because transfer functions are differentiable, thereby permitting gradient descent algorithms to utilize these errors or gradients to optimize the weights and biases.

This process is iterative and continues until a stopping criterion has been satisfied, such as convergence or completion of a predetermined number of iterations.

There are numerous backpropagation models, also referred to as learning models, available, including:

- Levenberg-Marquardt algorithm (LM)
- Scaled Conjugate Gradient Algorithm (SCG)
- Fletcher-Powell Conjugate Gradient (CGF)
- Conjugate Gradient with Powell/Beale Restarts (CGB)
- Polak-Ribiére Conjugate Gradient (CGP)
- One-step secant backpropagation (OSS)
- BFGS quasi-Newton backpropagation (BFG)

2.5 Summary

Chapter 2 presented several relevant theories and methods suitable for developing a generative structural layout tool. The general principle of parametric modeling was introduced, particularly in relation to building design. Some of the most imported topics related to this project were explained and illustrated.

The general principle of optimization was introduced, both for single-objective and multi-objective optimization. The Pareto front was introduced as a method to identify and visualize optimal solutions for a multi-objective optimization problem. Meta-heuristic optimization algorithms were introduced, and their characteristics and properties were outlined. Subsequently, the mechanisms behind the meta-heuristic algorithm Genetic Algorithm were explained in detail.

The fundamental definition of Machine Learning (ML) was introduced, focusing on how ML models can learn to emulate complex objective functions. This topic was further explored in section 2.4, which introduced surrogate modeling as a sub-field of ML. Some of the essential properties of surrogate modeling were introduced. It was emphasized how surrogate modeling can be combined with meta-heuristic algorithms to optimize complex and computationally intensive problems.

It should be noted that not all of the methods mentioned have been incorporated into the final design tool, but they have all been considered.

Chapter 3 – Action Research Analysis

"Analyzing is already design."

– Agostino Renna

Developing an early design tool that can generate structurally optimized layout suggestions requires careful consideration because the task affects multiple stakeholders in the field of building design. It is essential to analyze the task requirements and the current needs of building design professionals before proceeding with the development of such a tool.
An Action Research (AR) analysis was conducted to identify these needs and to provide a structural approach to develop and refine the proposed early design tool iteratively.

Chapter 3 presents the results of this AR analysis and is divided into two main sections. Section 3.1 provides a summary and structure of the basic theory behind AR and how it has been utilized; section 3.2 presents the results of the AR analysis and the resulting framework of the proposed design tool.

3.1 Action Research Methodology

AR is a research methodology that is primarily based on the social sciences. It is relevant for this project because the research goals defined in section 1.3.2 are primarily qualitative in nature. In general terms, AR provides a structured approach to solving a task iteratively. Furthermore, it also allows for active participation from the researcher. It is suitable for complex problems that span multiple fields of expertise, as is often the case in building design.

The variant that will be used for this analysis is defined as Insider Action Research (IAR). This variant has more or less the same methodology as AR. The main difference between IAR and AR is that IAR takes the role of the researcher within the organization into account and how that role can influence the research process. Insider action researchers must be aware of how their roles influence their perception of the world and must be able to decide when to step into and out of the multiple roles they hold within the organization [88].

This description corresponds well with the definition of the reflective researcher that is aware of his or her involvement by being self-reflective and self-critical during the entire research [30]. Brinkmann and Tangaard [89] also underline the importance of a reflective approach to qualitative research, meaning that the researcher should not blindly follow scientific methods and procedures. These methods are typically based on an ideal process, but reality rarely is ideal. This statement is complemented by Greenwood [90], stating that there

is no ideal form of AR. It depends on the scenario and research environment. Greenwood also states that it is possible to integrate different scientific theories and methods, qualitative and quantitative, into the applied AR method.

The typical approach to AR starts with the *framing phase*, where the research objectives and themes are concretized. This phase is followed by an iterative cycle consisting of four phases; the planning-, acting, observing, and reflecting phases, as illustrated in Figure 3.1.

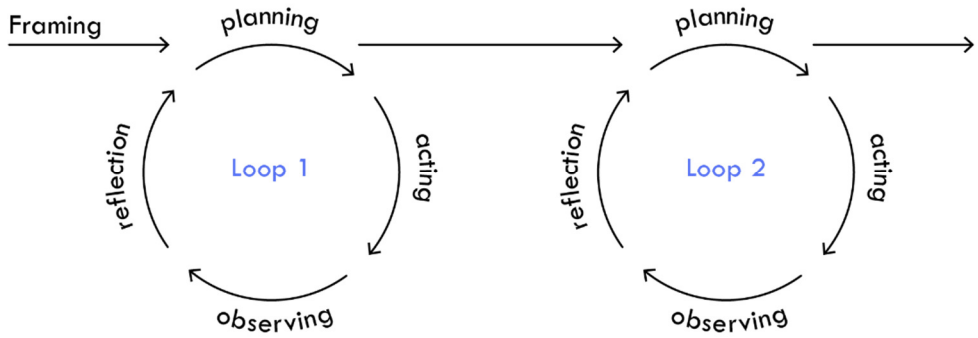


Figure 3.1 – Simplified concept of the Action Research iterations

The initial *planning phase* is used as preparation for the subsequent *acting phase*. Here sub-goals are defined, and relevant tasks are identified to aid in achieving the overall objective. The planned activities are then executed in the *acting phase*, followed by the *observing phase*, where data is collected and analyzed. During the *reflection phase*, the researcher evaluates the acquired results and feedback to determine how the gained knowledge can fulfill the primary objective. This phase is also utilized to identify areas that require improvement, which can then be addressed in the following cycle. This iterative process can be repeated as many times as necessary. Each loop allows for continuous improvement as each cycle builds on the knowledge gained in the previous cycle.

3.1.1 Research paradigm

Before using AR as a research methodology, it is important to define the applied research paradigm, as it establishes the overarching theoretical framework and perspective on which the research is based. A research paradigm is a set of assumptions and beliefs and can be formulated by addressing the three fundamental questions of research: The ontological question (What is reality?), the epistemological question (how can reality be examined?), and the methodological question (what is the relationship between the research and the subject of research?) [91] [92].

It is especially relevant to establishing the research paradigm in this Ph.D. project because, as mentioned, the research goals are primarily qualitative and, to a certain degree, subjective.

A relativistic approach is applied to the ontological question, which means that there is not one defined truth. The truth is defined by context. This approach is implemented because relevant stakeholders (engineers, architects, contractors) have their own worldviews on what defines a successful building design. Regarding the epistemological question, it is important to realize that building design requires expertise from multiple fields, and therefore, it is essential to obtain input from all relevant stakeholders.

Additionally, it is essential to acknowledge that this data is subjective. Hence, the researcher must maintain objectivity when analyzing the data. An interpretivism approach is therefore used to address the epistemological question, along with scientific methods from Participatory Action Research [93]. This approach implies that access to the required knowledge can only be obtained through social constructs. In this project, this is manifested through knowledge integration (understanding of needs and methods) between different disciplines rather than superficial knowledge sharing. For these reasons, active participation is essential to develop a successful tool.

3.1.2 Interview methodology

Information needs to be extracted from every building design-practitioners to construct a design tool that is conformed to their needs. This task is well suited for a qualitative interview methodology. A qualitative research interview attempts to understand the world from the subject's point of view [94]. Therefore, it is essential to develop questions that give participants a structure to explain and elaborate their cultural understanding of the topic [95]. Therefore, all questions must be open-ended to force the interviewee to think and reflect on the answer. This approach corresponds well with the semi-structured interview format, where the interviewer does not strictly follow a formalized list of questions. Instead, they give the interviewee more opportunities to express themselves fully. The transcript of the interviews is then examined using thematic analysis, which is a qualitative method for identifying, analyzing, and reporting themes within data [96].

3.2 Concept development of the design tool

The primary purpose of the applied AR analysis was to aid the concept development of the proposed design tool and to identify which features would support the needs of stakeholders involved in everyday building design. This goal was formulated into a research objective with corresponding sub-goals and themes as part of the framing face outlined in the AR cycle. It is noted that the research objective is formulated explicitly for the AR analysis in chapter 3, though it is derived from the main research aim defined in section 1.3.2.

Research objective:

- To develop a concept for an early design tool that generates optimized structural layouts and is conformed to everyday practices.

Sub-objectives:

- To obtain valuable input to the design tool from other representatives of the main stakeholders in building design (i.e., architects, engineers, constructors)
- To obtain a better understanding of their work procedure and needs.
- To demonstrate the benefit of an integrated design approach to building design.

Themes:

- Integrated design process, concept development, architectural engineering, structural layout optimization, and early design tools.

The iterative part of the AR analysis is outlined in Figure 3.2 and consists of three AR loops. The first loop was planned to obtain a better understanding of the design practice from the architect and contractors' perspective and to start identifying their needs in early design exploration. The second loop involved a co-creation workshop where structural engineers, architects, and contractors jointly discussed the challenges and opportunities of such a design tool. In the third loop, the framework of the design tool was refined and elaborated upon in greater detail based on prototyping and internal testing in the company.

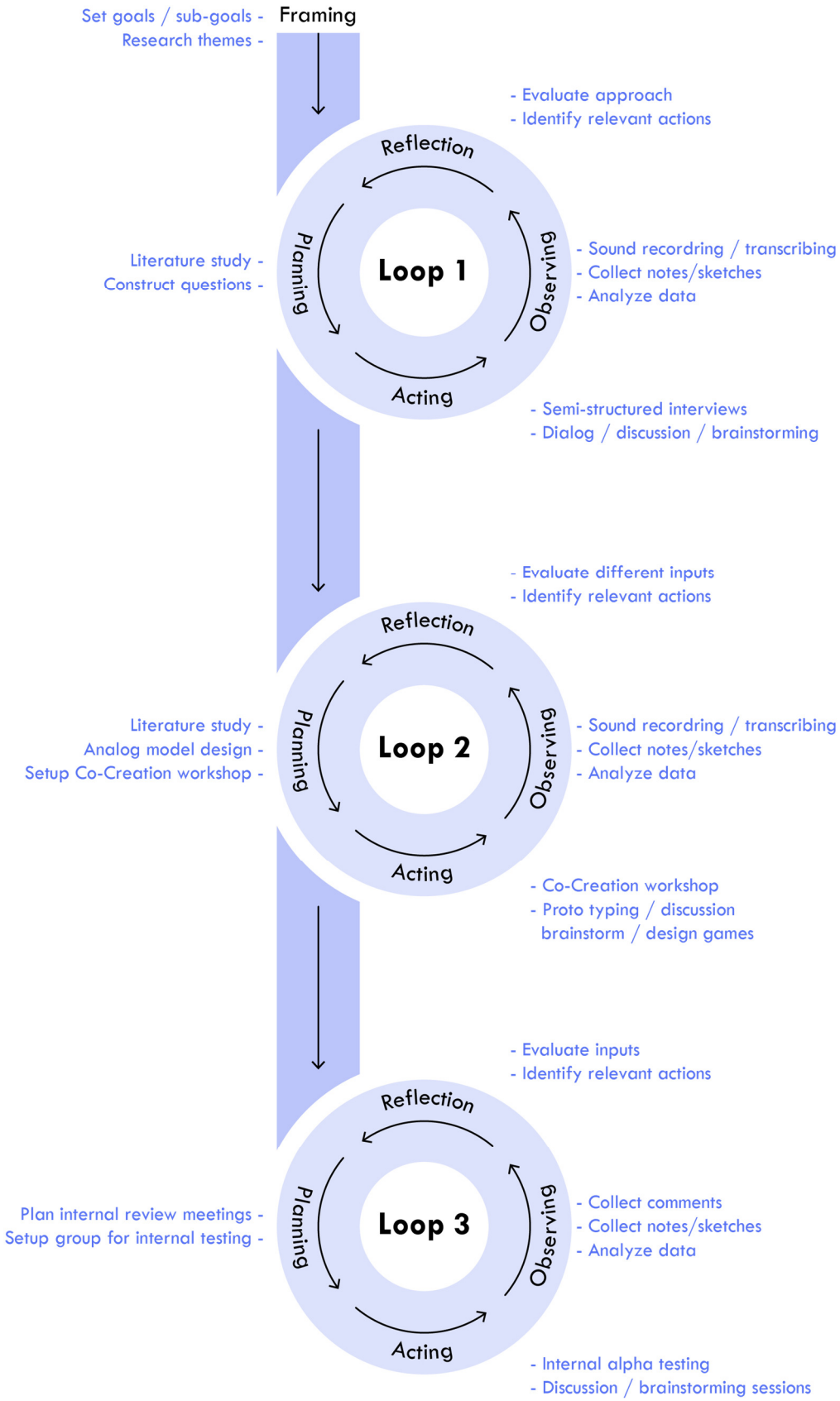


Figure 3.2 – Overview of the applied Action Research cycles

3.2.1 Loop 1 – Semi-structured interviews

The incremental development of the tool started with the first AR loop, which was dedicated to obtaining a better understanding of the design practice from an architecture- and contractors- perspective, but also to identifying their needs in early design exploration.

Only participants who grasped the general concept of computational design and early design tools were invited. This initial insight allowed for a more engaging discussion and relevant input. Representatives from the architectural company Ginnerup and the construction company NCC participated; they are henceforth referred to as the architect and the contractor.

A literature study was conducted in the *planning phase* focusing on early design tools, design practice, interview methods, and barrier analysis for using early design tools. The *acting phase* commenced with interviews in the participants' native environment. This familiar setting should make the interviewee more comfortable to answer more freely and express their opinion. This choice also allows the interviewer to observe the local language, daily routines, and power structures, thus providing a sense of what the interviewees will be talking about [95]. The interview data was collected in the *observing phase* and underwent a thematic analysis in the *reflection phase* with the following themes as a result:

- Integrated design process
- Miscommunication
- Importance of cross-disciplinary knowledge
- Dissemination
- Compatibility
- Design objectives and constraints
- Interactivity
- Information-level

During the interview, one of the first things that became evident was that the architect was already pursuing a more integrated design approach. They encouraged the associated engineers in a given project to provide their input early in the design process. This approach contradicts the prevailing practice observed in literature and among architectural firms. Typically, architects are hesitant to involve structural engineers initial design phase as they fear the creative process would be driven by quantifiable engineering objectives [30]. The architect was aware of their status as an outlier and reported that their inquiries were often met with resistance. The engineers attached to a given project would prefer to wait for the finished architectural project. C. Mueller also observed this issue "*the engineering team rarely provides advice or feedback to the architecture team on the form*" [13].

The architect reported that when engineers did engage in early design collaboration, it usually resulted in more successful designs and better communication throughout the remaining project period. All involved interviewees also highlighted the importance of communication, stating that miscommunication was the main reason for unsuccessful design projects. Conversely, successful designs were driven by a well-functioning communication channel and by the involvement of people with cross-disciplinary knowledge were involved.

When the architect and contractor were asked about their needs and requirement for early design tools, both parties prioritized features such as dissemination, compatibility, and interactivity. The contractor also highlighted the importance of reflecting on the available information level when creating early design tools. To illustrate this point, the interviewee cited a project where a tool was developed to optimize the placement of cranes during construction. However, the tool had to be discarded because the required level of information was not available as input at the time.

In addition, the interviews included a discussion on which design objectives should be incorporated into an early design tool. Various objectives were considered and debated, but the underlying commercial perspective was always present. Whether the objectives were minimization of mass, simplifying the workflow, or optimizing constructability, the overarching theme was to reduce costs.

The collected data and input from the interviews were processed in the *reflection phase*. As mentioned in section 3.1.1, it was recognized that each participant's definition of the truth would be influenced by their perspective of what constitutes good building design. With this in mind, the input collected was evaluated to determine which insights would provide the most value in a holistic perspective when developing an early design tool.

3.2.2 Loop 2 – Co-creation workshop

The defined sub-goals in loop two were to obtain relevant inputs for the design tool to a much greater extent than during the interviews and to persuade the participants to commit to an IDP approach.

The loop began with a planning phase, during which the most appropriate research methods were selected to fulfill the research objectives. These methods included a literature review, prototyping, co-creation, and design games. The co-creation in the *acting phase* manifested in a workshop, where the architects and contractors interviewed in loop one would participate along with structural engineers from hamiconsult a/s.

The accumulated knowledge obtained in loop one was used to develop the first analog prototype of the design tool. The design tool was presented during the workshop using digital sketches as a starting point for further refinement and development. Some of these sketches are illustrated in Figure 3.3, Figure 3.4, and Figure 3.5. Design games were also employed to stimulate creativity and initiate a further discussion among the participants on enhancing the tool.

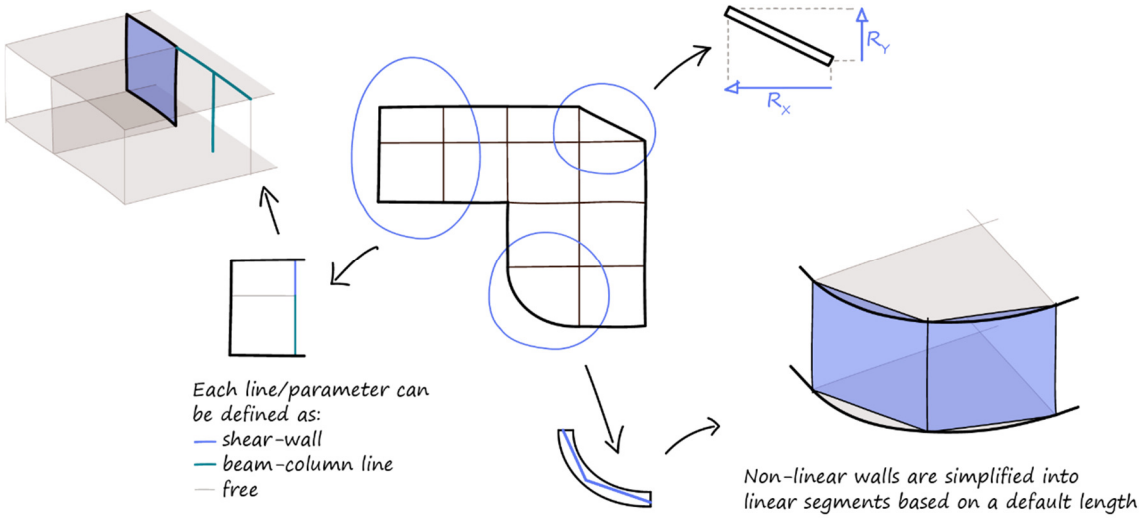


Figure 3.3 – Sketch used in the workshop to illustrate the proposed flexibility.

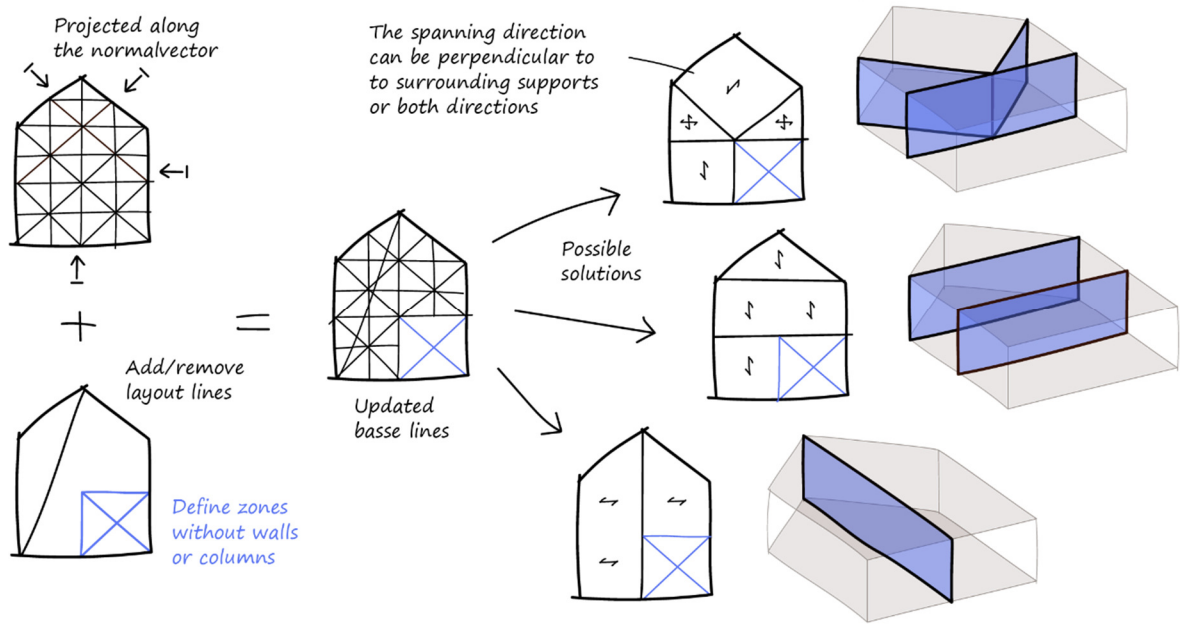


Figure 3.4 – Sketch used in the workshop to illustrate how the proposed parameterization can produce diversity.

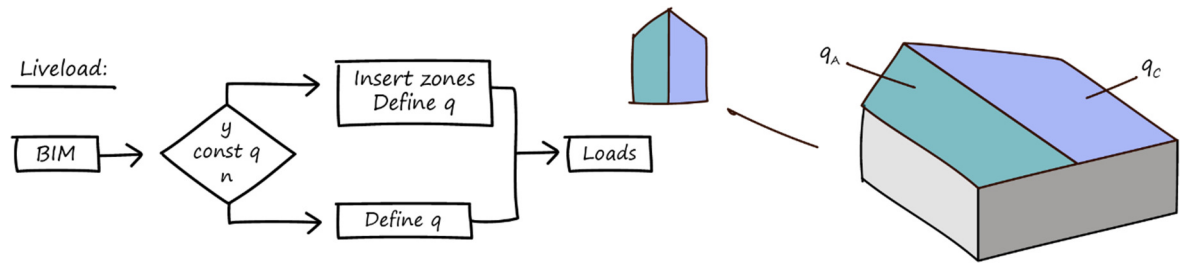


Figure 3.5 – Sketch illustrating how live loads could be applied interactively.

The discussion during the workshop brought up several interesting points, such as the fact that a design tool should not attempt to encapsulate every possible design option but instead focus on around 80 percent, as the last 20 percent would be too demanding to include.

Another vital talking point was the dissemination of results. Several examples were given of promising tools that lacked adequate dissemination. The problem was not always the quality of the information provided but rather the quantity of data. In some cases, the necessary information drowned the vast amount of data provided, which resulted in the tool not being integrated into the work process. Therefore, the importance of reflecting on which information to include was highlighted because sometimes less is more.

Other identified critical features that were identified for implementation included flexibility and interactivity. The setup time needed to be minimal, and the design freedom should be ensured through an interactive platform. Optimization algorithms should generate solutions that challenged the mental models that designers sometimes become fixated on. However, if the algorithm produced a solution that corresponded with the designer's initial idea, then it would work as a confirmation of the designer's approach.

Though the discussion was based on a holistic approach to the design tool, some topics are inevitably exclusive to a particular discipline within building design. As an example, the contractor emphasized the need for the design tool to have the capability of selecting suppliers and utilizing only the RC elements that were available from these associated suppliers. Although this was not a priority for all parties, everyone recognized such a feature's potential benefits.

Another relevant point brought up by the architect and contractor was that the engineer should be the primary user of the tool. However, the tool should work as a collaborative platform between the stakeholders and be able to facilitate communication and information sharing in the initial design phase.

All the discussions, comments, and inputs were collected through audio recording and analog sketches in the *observing phase*. The data underwent the same analysis in the reflection phase as in the first AR-loop to sort through many relevant suggestions. The chosen inputs

were then used to adjust and improve the proposed design tool following the first sub-objective of AR-loop 2.

Some of the noticeable changes that emerged from loop two were related to the global parametric representation. Initially, the story height for a building design was defined as a constant. However, this approach was deemed problematic, as a floor's interior ground clearance height would change depending on the applied slab type. Therefore, the story height parameter was changed from a constant variable to a dynamic parameter that directly depends on the defined minimum ground clearance. This modification means that the total floor-to-floor height is calculated as the sum of the defined minimum clearance height, the height of the structural elements used, such as beams and slabs, and a defined height for installations. A consequence of this change is that the building's height will dynamically change between each optimization iteration.

The architect highlighted this issue because it was a real-world problem they had encountered. In this case, the dilemma was whether to add a support line to reduce slab height or use a TT-slab to cover a longer span. Calculating the consequence of this choice is very resource-demanding because it would affect the entire structural system. Using the TT-slab may intuitively seem like the most resource-effective solution. However, this would, in turn, have resulted in a slightly taller building, which in turn would have increased the horizontal wind load, and the slenderness of the supporting walls, among other factors, and thus could have meant more mass. The design tool could help inform the architect of the consequences of these two options when implementing this change.

In summary, the first and second research objective is considered to be achieved. Several relevant inputs, both specific and general, were provided, all of which were highly relevant and contributed to the development of the tool. Regarding the second objective, the researcher noticed a growing enthusiasm as the workshop progressed. The participants expressed a commitment to collaborate on a relevant project where the tool could be applied.

3.2.3 Loop 3 – Refinement of the design tool

While loops one and two were executed based on a specific timeframe, loop three represented a more ongoing refinement process. This AR loop's goal was to further refine the tool through brainstorming sessions and native alpha testing of some of the tool's modules.

The *planning phase* consisted of a literature study on alpha testing in a native organization [97], preparing specific modules for release within the organization for testing, and organizing a group of alpha testers. The released modules were all related to wind force calculations. This decision was made because it was deemed that these modules could add value immediately and because the primary purpose was to assess whether the tool's user interface was intuitive

enough and suited the user's workflow. This goal could be achieved without releasing the entire package of modules, which at this point in the process, was not fully developed yet.

During the *acting phase*, some tool modules and a user guide were released to the organization. In addition, sessions were arranged with personnel in the organization to discuss whether the current features of the tool could accommodate the requirements identified in loops one and two.

The alpha testers were encouraged to provide feedback on their experience and report on possible improvements during the *observation phase*. Comments and sketches from the brainstorming sessions were collected, and all this data was reviewed during the *reflection phase.* It was observed that only around a third of the invited users utilized the modules continuously. However, those who did, integrated the modules into their workflow and contributed with valuable insights on potential improvements and bug reports.

The framework of the tool was reviewed during the brainstorming sessions, and it became apparent that the current parameterization method, outlined in Figure 3.4, was not flexible enough. The method was deemed too similar to existing methods and did not adequately produce varied and buildable solutions. Thus, a novel global parameterization methodology named Adjacent Polygon (APoly) representation was developed and implemented; the method is explained in detail in section 4.3.

3.3 Summary

Chapter 3 introduced the concept of Action Research, including its basic theory and how it can be used for concept development through a cyclical process of planning, acting, observing, and reflection.

The AR analysis resulted in many valuable improvements to the general methodology of the tool, and it also confirmed that there is a need for tools that can facilitate closer collaboration in the initial design phase. The inputs and data collected during the AR loops were evaluated based on their potential value and feasibility for implementation. The most relevant and requested features were identified and served as a guideline for the actual development. The features and requests are listed as follows:

- Flexibility and exploration should be prioritized over accuracy.
- The primary user of the tool should be the structural engineer.
- Provide optimized solutions without overwhelming the user with too many options.
- Solutions can be presented in 2D; a third dimension should only be added if it provides new information.
- A simple geometric representation such as surfaces, lines, and points are sufficient to convey a solution.

- Interactivity is crucial; users should be able to adjust solutions and create them independently.
- Dissemination is essential; the results are meaningless if they cannot be understood.
- The solutions should consider constructability, for example, by allowing for the selection of suppliers and taking actual construction practices into account.
- Dynamic constraints are preferred over static ones.
- Users should be able to insert their constraints into the parameterization process.
- Setup time should be minimized.

The desired level of interactivity, evaluation speed, and flexibility is only possible if many of the resource-demanding tasks are automated. Thus, a high level of automation has to be incorporated into the tool.

Chapter 4 – The design tool

"You cannot mandate productivity, you must provide the tools to let people become their best."

– Steve Jobs [98]

Greater collaboration and exploration in the initial design phase require more intelligent and flexible design tools. The first step to accomplish this goal, is to identify the key features that would most effectively achieve these objectives; this was the primary aim of chapter three. Chapter four, on the other hand, focuses on implementing these features into a design tool and presents the different modules in the framework. The design tool is built on four core principles: optimization, interactivity, dissemination, and automation, as illustrated in Figure 4.1.

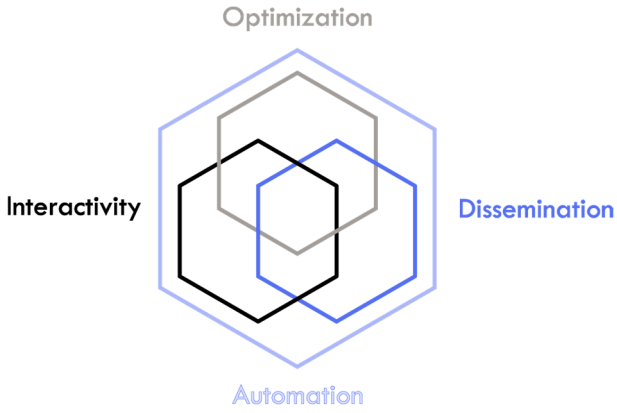


Figure 4.1 – The four core-principles of the design tool

These four principles are intended to add value for all stakeholders involved in the building design process. *Optimization* and ML algorithms enable the user to find high-performing solutions which otherwise would have been impossible through a manual trial-and-error procedure. The principle of *Interactivity* incorporates the user into the design process. This symbiosis of user and algorithm transforms the design tool into a supplementary tool that guides the design process instead of driving it. The focus on *dissemination* is also crucial. The design tool and the corresponding output are meaningless if not disseminated effectively, especially when the goal is to provide the user with information on high-performing building-design alternatives. Finally, *automation* is the principle that facilitates all the other core principles. Real-time building design exploration is made possible by automating parameterization, load calculation, and evaluation tasks.

The tool is also intended to fit into a broader context, as it is important to define where it fits into the general design process. Given the numerous design parameters, it would be practically impossible for a tool to generate fully dimensioned and highly detailed solutions in a single design iteration. Instead, a framework of nested design cycles is proposed, which involves a series of design iterations with decreasing global design freedom to compensate for an increasing level of calculation fidelity. Ideally, by following this strategy, one can move from a schematic design to a finished detailed design in a short period of time. A basic illustration of the proposed design cycles is shown in Figure 4.2.

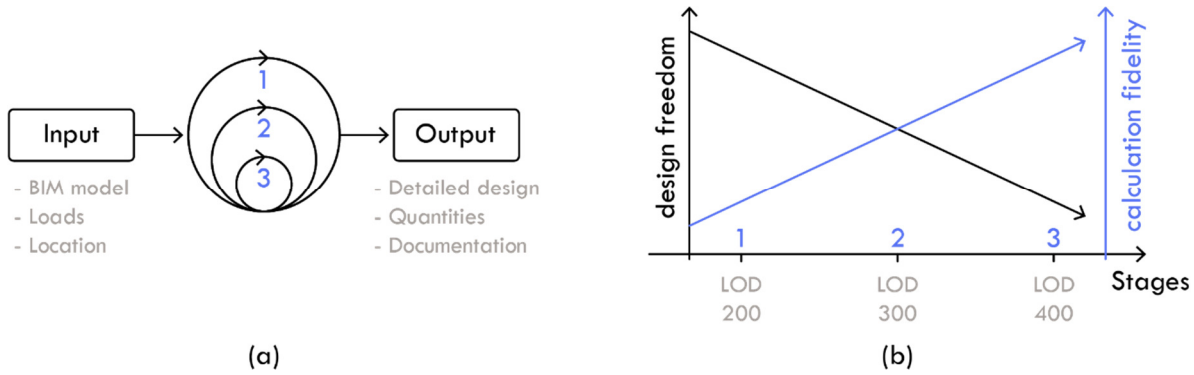


Figure 4.2 – (a) Design cycle overview, (b) Flexibility-Fidelity graph

The outer loop represents the first design cycle where the designer has the most control over the global geometry. As the design process advances, the geometry becomes more established, which is evident from the designated level of detail (LOD) associated with each design loop. LOD enables designers in the industry to specify the content and reliability of the geometrical representation and material properties at various stages in the design- and construction process. Visualization of the chosen LOD levels is demonstrated in Figure 4.3. A comprehensive overview of the entire spectrum of LOD levels can be found in [99].

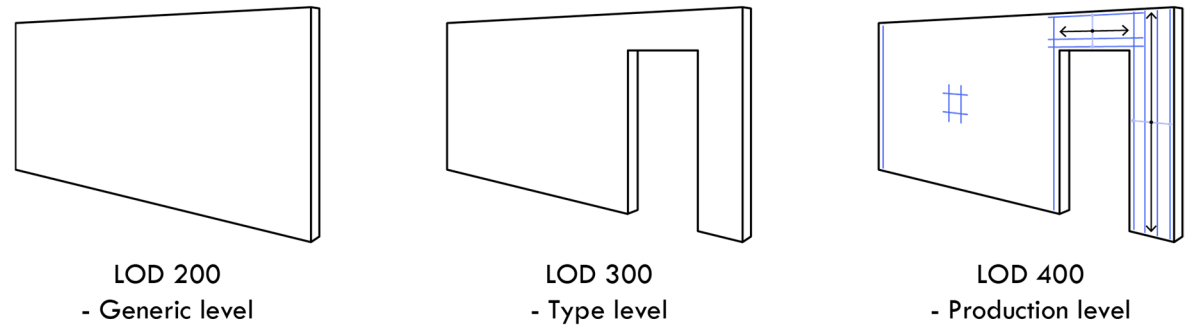


Figure 4.3 – Visualization of different LOD levels for a RC-wall.

The design loops are linked, so the output of one loop must match the following loop's required input. The first design cycle commences with a basic architectural shape as input, and the walls, slabs, and column-beam lines are generated into varied and optimized structural layout suggestions. The tool presented in this Ph.D. project is intended to operate the design iterations in design cycle one. The information level in this cycle is defined as LOD 200, where structural elements are defined at an expected level concerning geometry, placement, and corresponding material properties.

Design cycles two and three are intended for further development; only their fundamental principles are explained. The structural layout and geometry are fixed in cycle one, which frees up parameters for the design iterations in cycle two. Recesses can be inserted into walls, and the horizontal distribution of forces can be optimized using heuristic algorithms. The last cycle is intended as a validation cycle, where standard industrial tools are used to verify the final geometry, reinforcement, and material properties. The plans for the last two cycles are elaborated upon in section 6.3.2.

4.1 Software platform and user-interface

Programming is essential for any complex problem-solving process in computational design. This project considered different languages, such as Python, C#, and C++, which have different pros and cons. Python is very flexible, popular, and intuitive to use. Python also has access to useful ML- and mathematical libraries such as Numpy, Scipy, and TensorFlow. However, it is not as fast as C#/C++, which is paramount when dealing with the many thousand iterations required when using meta-heuristic algorithms. For that reason, Python is disregarded in this project. C++ is a little faster than C#, but it lacks flexibility as memory management has to be done manually. This issue is not the case for C#, as it is built within the .NET framework, and this has automatic memory management. C# is therefore chosen as the primary language.

C# provides access to many different .NET libraries, for example, ML.NET [100], which is a machine learning library that supports Python models when used together with NimbusML [101]. Word.Interop can also generate custom reports to document the static calculations. The GMAP API from Google [102] can also be operated using C#. This API provides different services, such as elevation values that can be used to calculate the orography factor used in wind load calculation. C# can also access Matlab's [103] substantial optimization algorithm library by compiling Matlab code into .dll files.

However, the main task for C# as the chosen programing language is to interact with the CAD software of choice. Here Rhinoceros 3d (Rhino) is chosen. Mainly because of its abilities as a 3d modeling software, but Rhino also works excellently as a software platform for engineering analysis, visualization, and other relevant analysis tasks. Rhino is very compatible

with other software packages, and it is backed by a very active community that develops and shares a lot of open-source plugins. Rhino also comes with the RhinoCommon library, which contains all the methods and properties available in Rhino and can be accessed using C#. Rhino also ships with Grasshopper (GH), which is a built-in visual programming language.

Various options were considered for the user interface (UI), such as text-based commands or a comprehensive graphical UI using Windows Presentation Foundation (WPF) or Windows Forms. However, as previously mentioned, the design tool is divided into separate modules that should be used independently of each other to maximize their usage potential. Therefore, it was determined that using a software-platform adapted to this approach was more appropriate. The decision was made to develop a GH library of custom components tailored to handle different stages and tasks in the design methodology. A visualization of this library is shown in Figure 4.4.

Figure 4.4 – Icons illustrating the different modules in the tool's library

In addition, building the user interface in the GH environment has the advantage that end-users can easily expand on the functionality by utilizing native GH components or the built-in C#/Python editor. This approach only requires a basic understanding of programming using an integrated development environment (IDE) such as Microsoft Visual Studio.

The flexibility of the design tool was improved using various functionalities such as buttons and sliders were incorporated into the components by adapting the GHCustomComponent C# library [104]. In addition, features were also coded in to allow users to interact with the design directly in the viewport.

The final user interface can therefore be described as a combination of a GUI and a simple DAG structure.

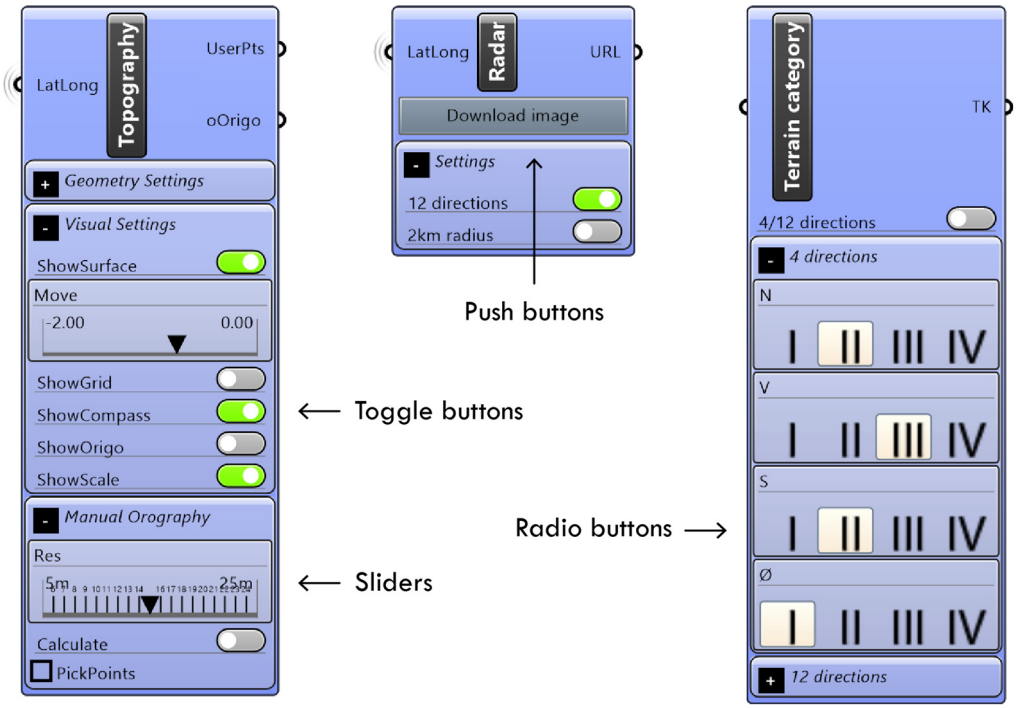


Figure 4.5 – Examples on the UI of the modules.

4.2 General framework

The framework utilized in design cycle one can be broadly divided into two main phases: initialization and optimization. Figure 4.6 is a simplified illustration of this framework.

All the necessary data required for the optimization phase is computed during the initialization phase. These tasks include parameterization, where the building's geometric degrees of freedom are defined. This process sets the overall framework for the optimization phase, as the solution space size is defined here. Therefore, this phase has a significant impact on the final design.

Furthermore, natural loads, live and dead loads, as well as various user inputs, are defined in phase one. These user inputs consist of mandatory inputs, such as defining the floor height, voluntary inputs that can influence the optimization process, and the shape of the final design. All these aspects are elaborated in detail in section 4.4 to 4.5.

The main purpose of the initialization phase is to pre-process as much data as possible to reduce the computational burden in the subsequent optimization phase. Here a meta-heuristic algorithm in the form of a modified genetic algorithm is utilized to find optimized structural layout suggestions. This phase is presented in detail in section 4.7.

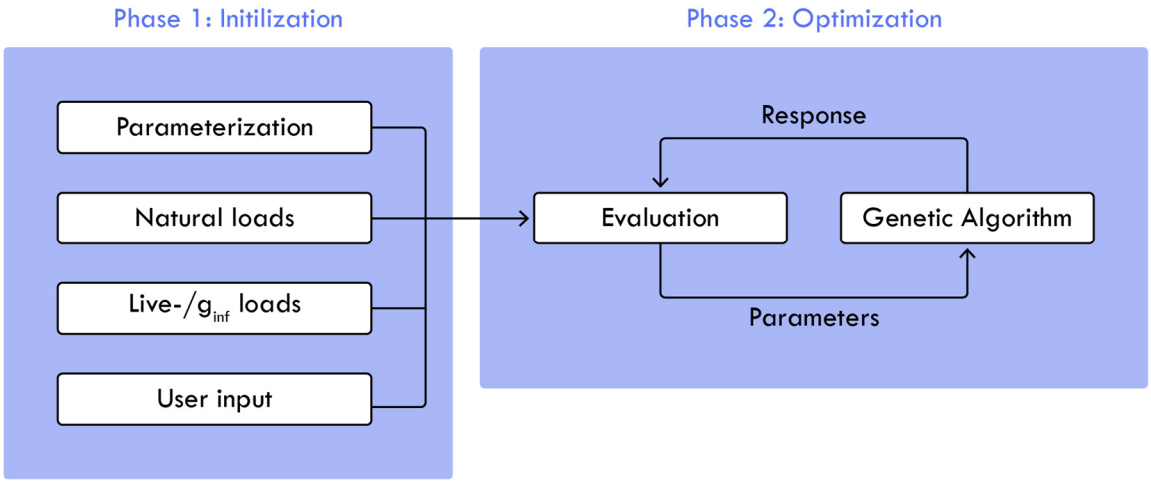


Figure 4.6 – The general framework of design cycle one.

A Finite Element Method (FEM) will not be utilized to evaluate the solutions. This decision is based on multiple factors. Firstly, the computational expense of setting up and running a FEM for each iteration is significant, especially given that the genetic algorithm may require thousands of iterations to converge. Additionally, this approach would limit the use of the design tool to devices equipped with licensed FEM software. Instead, a C# evaluation module is developed, which utilizes well-established theories from Eurocode 0-2 and widely recognized theory books like [105] [39] [9].

4.2.1 Design objective

Optimization is only possible when feedback is provided. Whether the optimization technique is gradient-based or probabilistic, this statement is true. The objectives guide the optimization algorithm in the solution space to find solutions with features that satisfy the defined objectives. Consequently, the choice of objectives significantly influences the final form of the design [29].

- **Minimizing cost:**

Minimizing mass is a common design objective for early-stage design tools. However, this objective is deemed too simplistic because the geometry of RC elements is strongly influenced by the amount of reinforcement used. Therefore, the optimization algorithm must consider the reinforcement-to-concrete ratio when designing an optimal RC element. This consideration is integrated into each RC evaluation module presented in section 4.6.

Therefore, the objective of minimizing the cost of the load-bearing structure is selected as the primary objective. This objective is related to minimizing mass but can be applied to structures of different material types.

Different material prices are defined based on available data from various suppliers; the specific material prices are listed in Appendix A. The SU models created in this project are derived from these prices. The models must be retrained if the relative relationship between the reinforcement and concrete costs changes significantly. The actual price is irrelevant as the cost is only used as feedback for the algorithm to identify cost-effective solutions. It is emphasized that the calculated cost only includes material costs and does not factor in other aspects such as transportation, manufacturing, and assembly costs. Nevertheless, the objective still provides an excellent basis for comparison, which is exploited in the optimization process.

The specific cost objective is calculated as the total cost for one floor in DKK divided by the floor area. This value is denoted as the average cost per square meter (CPM2). By using this definition, it becomes easier to compare different cases. Minimizing the cost will also implicitly lead to a reduction in embodied energy. It will be simpler to incorporate other design objectives, such as LCA in future versions because all material quantities are already extracted within the algorithm.

- **Minimizing the average cell size:**

Building design often has multiple objectives to consider, and they are not always based on material quantities or performance metrics. Sometimes objectives may be practical or aesthetic, and quantifying them can be challenging due to their subjective nature. This project includes an aesthetic objective, defined as the maximization of cell sizes as defined in equation 4.1.

$$ACS = \frac{\sum_{i=0}^{n} cellArea_i}{n} \tag{4.1}$$

Where:

$\boldsymbol{cellArea_i}$	The area in square meters for cell i.
$\boldsymbol{n}$	The total number of cells.

A cell is an area enclosed by load-bearing walls. The algorithm is expected to generate structural layout designs with larger open areas by maximizing the ACS value. This objective can be considered functional and aesthetic, as the larger open spaces make it easier to model or re-model a floor plan area.

4.3 Parametric representation

4.3.1 Global parameterization

The parameterization process defines how the overall configuration of structural elements can vary, including the placement of walls, column-beam lines, and the direction and length of the slab spans. The process can be divided into two parts: global and local. The global parameterization splits the building plane into smaller components or base shapes. These shapes are then further refined in the local parameterization process, where walls and column-beams lines are placed along with the direction and length of slab spans.

This novel approach to parameterization, denoted as Adjacent Polygon (APoly) representation, was developed to generate diverse yet feasible structural layout suggestions. The method initiates by generating a set of basis shapes, denoted as $\bar{A}$, as expressed in the mathematical notation shown in equation 4.2, where n denotes the total number of base shapes.

$$\bar{A} = \{x \in \mathbb{N} \mid 1 \leq x \leq n\} \tag{4.2}$$

There are no inherent limitations on the number of base shapes a building plan can be divided into, but increasing the number of polygons will exponentially increase the solution space. Therefore, simple shape grammar rules were incorporated into an algorithm to generate a finite set of basis shapes. The algorithm works by extending each linear line in the building plane into pieces using these lines. The user can add lines to create a specific basis shape if desired.

The base shapes can then be combined into APoly variations. The number of possible APoly shapes is calculated as the powerset of all the base shapes. A subset is then selected from these candidates by subjecting them to specific constraints. This concept can be illustrated using a simple example where a square building plane is divided into a set of four base shapes, denoted $\bar{A}$, in equation 4.3 and illustrated in Figure 4.7.

$$\bar{A} = \{1,2,3,4\} \tag{4.3}$$

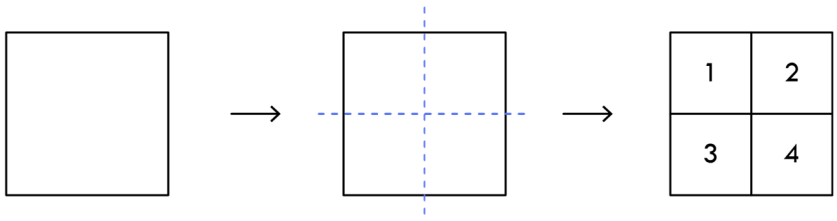


Figure 4.7 – A square building plane is divided into four base shapes.

The set of APoly candidates denoted $\bar{B}$, is then derived from the powerset (P) of $\bar{A}$, as shown in equation 4.4. The set $\bar{B}$ encompasses all feasible combinations of the base shapes.

$$\bar{B} = P(\bar{A}) = 2^n \tag{4.4}$$

The final set of APoly shapes denoted $\bar{C}$ is obtained as a subset of $\bar{B}$, as defined in equation 4.5.

$$\bar{C} \in \bar{B} \mid F(\bar{B}) \leq max_{sides} \tag{4.5}$$

Where:

$\boldsymbol{F(\bar{B})}$	A function that counts the number of linear edges in the APoly candidate
$\boldsymbol{max_{sides}}$	The maximum number of allowed linear edges

In basic terms, the new shape formed by the union of basis shapes should contain fewer or an equal number of linear edges as the number specified by the user. By default, this number is set to four. In the case of the four simple base shapes, this process would result in nine APoly shapes that meet the defined conditions, as illustrated in Figure 4.8.

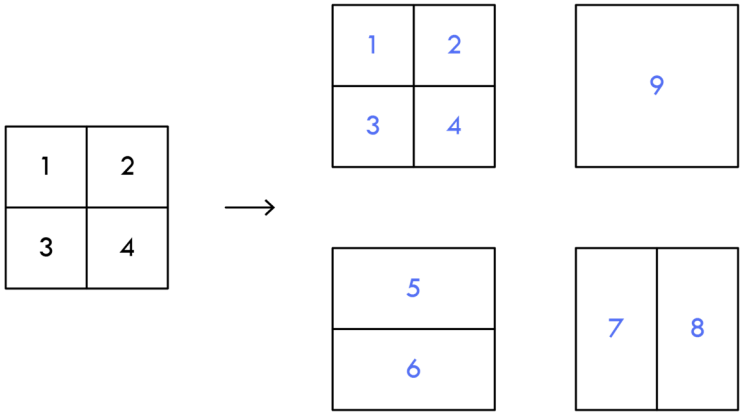


Figure 4.8 – Example of how the basic shapes are converted into valid AP shapes.

It is possible to form various configurations that fill the building plane by assembling the set of APoly candidates. However, some APoly shapes may overlap and cannot be combined. Therefore, it is necessary that the combined shapes are non-intersecting and do not have more sides than what has been defined beforehand, hence the naming of Adjacent Polygons.

Permutation encoding is used to generate variations of the APoly shapes. The technique utilizes a chromosome that contains a specific order of unique integers. Each number represents a particular APoly shape. In the case of the square building plane, a given solution could be represented by the following chromosome, denoted as x_1.

$$x_1 = \{5, 2, 1, 3, 8, 4, 9, 7, 6\} \tag{4.6}$$

The sequence determines the order in which the APoly shapes are placed. However, as mentioned earlier, some APoly shapes are mutually exclusive and cannot coexist. Information on which APoly shapes can and cannot be placed together is computed and stored in a matrix, denoted as the incompatibility matrix. For the case in question, the incompatibility matrix is illustrated in Figure 4.9. Row and column numbering represents specific APoly shapes, and a blue cross indicates that the two given APoly shapes are incompatible.

AP no.	1	2	3	4	5	6	7	8	9
1	X				X		X		X
2		X			X			X	X
3			X			X	X		X
4				X		X		X	X
5	X	X			X		X	X	X
6			X	X		X	X	X	X
7	X		X		X	X	X		X
8		X		X	X	X		X	X
9	X	X	X	X	X	X	X	X	X

Figure 4.9 – Example of the incompatibility matrix.

Using solution x_1 , APoly number 5 is placed first, and the sequence is updated by removing all options that cannot coexist with APoly number 5 according to the incompatibility matrix. The following eligible number is then APoly number 3. The sequence is updated again to check for incompatibility with APoly number 3, and the process continues until the building plane has been filled. Figure 4.10 illustrates the process for the solution x_1.

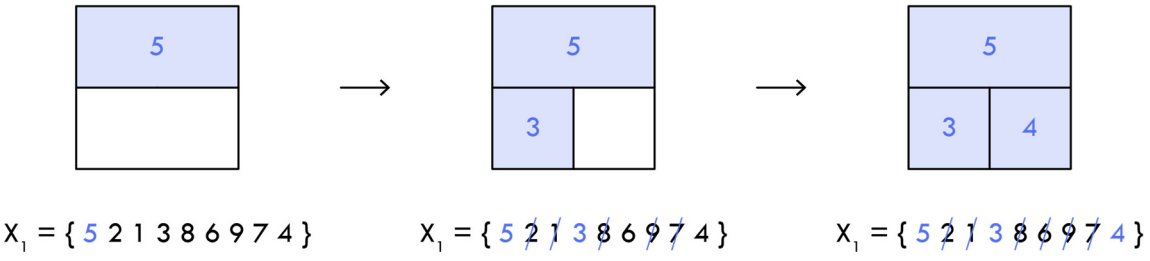


Figure 4.10 – Demonstration of how the permutation encoding determines the sequence of adjacent polygons.

In Figure 4.11 (a), a more complex example is depicted, where the blue shapes illustrate the valid APoly shapes that can be formed from the basic shapes. Figure 4.11 (b) illustrates a series of examples of final solutions that can be derived from the set of APoly shapes. The approach can be characterized as a top-down methodology, where the overall structure is deconstructed into more detailed components.

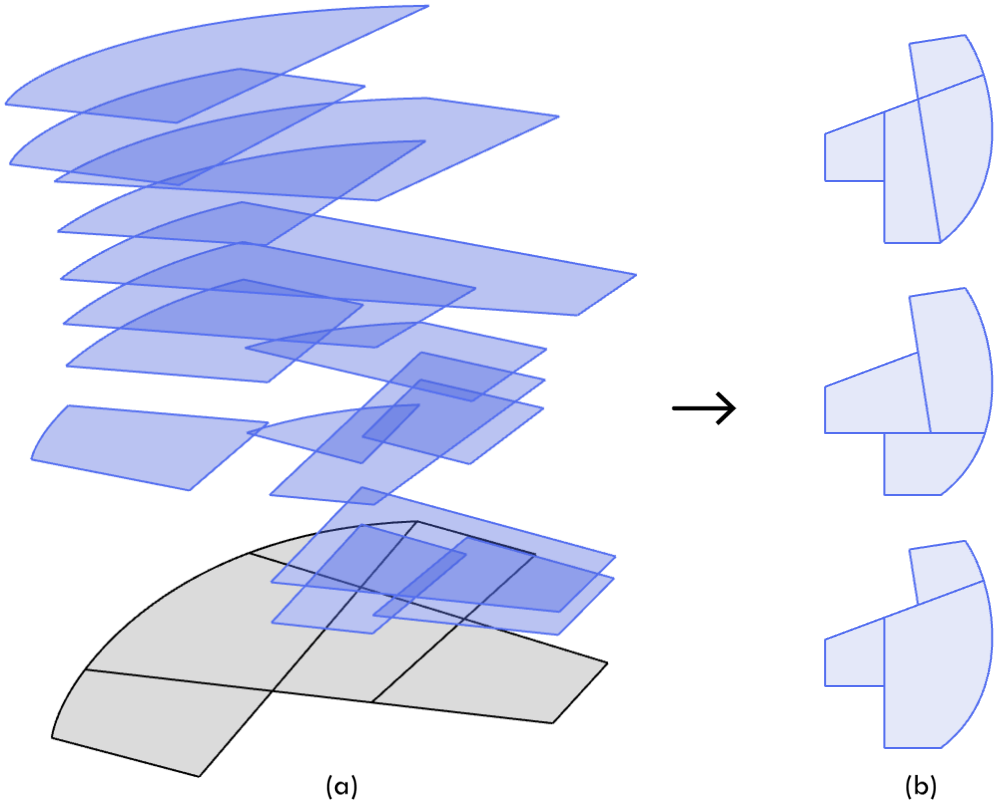


Figure 4.11 – Visualization of the use of simple shape grammar rules to divide an arbitrary building plane into it basic shapes. These shapes are transformed into APoly shapes that can be recombined to produce multiple solutions.

4.3.2 Local parameterization

Every APoly shape can be further detailed by specifying its span direction and length. This is accomplished using real-value encoding, where four parameters are associated with each activated APoly shape. These four parameter types, denoted p1, p2, p3, and p4, are normalized in the domain of 0-1. The objectives of the parameters are defined in Table 4.1.

Table 4.1 – Local parameter types

Parameter name	Objective
p1	Determines the load direction
p2	Determines the span length
p3	Determines the wall ratio on inner lines
p4	Determines the wall ratio on perimeter lines

The parameters operate in a sequential workflow from one to four, meaning that the value of the preceding parameter affects how the following parameter carries out its function. Each APoly shape possesses at least one linear edge, with each edge being assigned a number. Equally sized subdomains represent the number of edges within the primary domain of {0≤x≤1}. The p1 value can determine which edge activates by identifying the associated subdomain. This concept is illustrated in Figure 4.12, where it is shown that three different p1 values activate three distinct edges. The load direction is then calculated as the perpendicular vector to the activated edge.

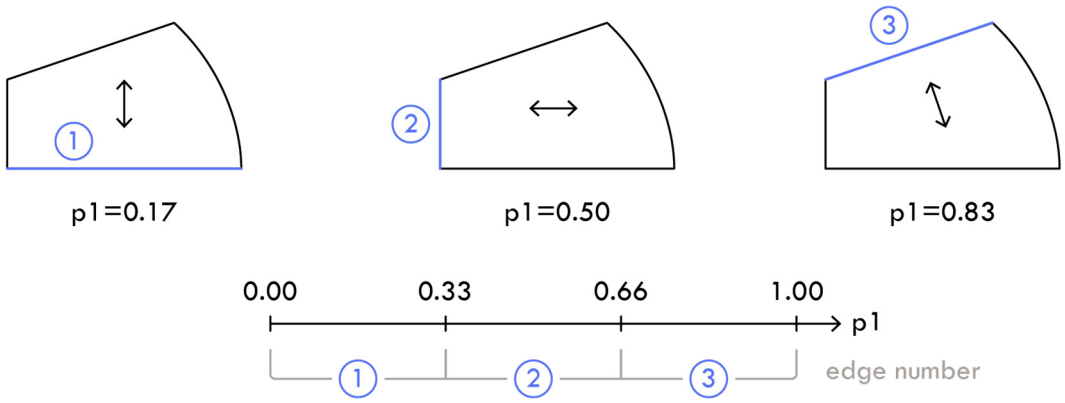


Figure 4.12 – A visual demonstration of how the p1 parameter determines the load direction.

The p2 parameter is responsible for determining the span length. This task is accomplished by remapping the p2 parameter to an integer that designates the active split value within the specified range. The split range is a series of numbers that determines how many times the total span length is divided into equal lengths, and it is calculated using equation 4.7. Transverse support lines are placed at each split, which can then be converted into a wall, a column-beam line, or a combination thereof.

$$S_{range} = \left\{ x \in \mathbb{Z} \middle| \left\lfloor \frac{L}{span_{max}} \right\rfloor \leq x \leq \left\lfloor \frac{L}{span_{min}} \right\rfloor \right\} \tag{4.7}$$

Where:

$\boldsymbol{L}$	The total span length in meters
$\boldsymbol{span_{max}}$	The maximum slab length, set to 16.8 meters
$\boldsymbol{span_{min}}$	The minimum slab length, set to 4.0 meters

The properties of the strongest slab type available in the slab database constrain the maximum length of the slab. The minimum length is set to a reasonable value from a practical and economic standpoint, as it is neither profitable nor logistically feasible to have support lines for every meter.

To illustrate this concept, consider a case where the total span length L is defined as 15 meters; the split range would then be calculated as $split_{range} = \{0, 1, 2\}$. This example is illustrated in Figure 4.13.

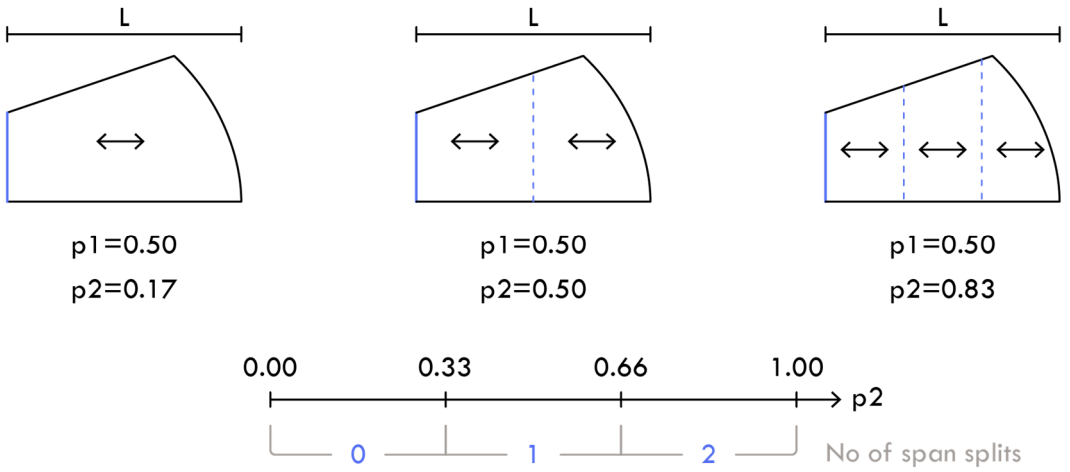


Figure 4.13 – A visual demonstration of how the p2 parameter determines the span length

Before defining p3 and p4, it is necessary to establish the terminology for what constitutes inner lines, perimeter lines, and outer lines. This definition is best illustrated in Figure 4.14, which expands upon the simple example presented in Figure 4.10.

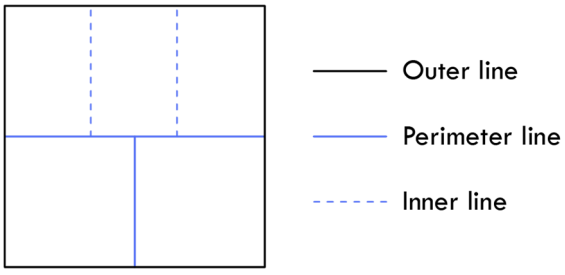


Figure 4.14 – Visual terminology of the different line variants

It is necessary to distinguish between these three line-type variants in the design methodology because each line type is differently parameterized when converted into structural elements. The *outer line* represents the external wall of a building, which can only be converted into a RC wall. However, an external wall is rarely solid. Therefore, a single parameter called the recess ratio determines the proportion of the wall that consists of openings. This parameter will affect the stiffness and mass of the external wall. The *perimeter*

line is defined as the outline of the APoly shapes that do not overlap with the outer lines. The *inner lines* are located within the APoly shape and, for practical reasons, will always be orthogonal.

The p3 and p4 parameters have the same function but operate on different elements, specifically the inner and perimeter lines. These parameters determine the wall ratio, which indicates the proportion of the line that should be converted into a RC wall element. The remaining line part is converted into a column-beam line. An example of this is illustrated in Figure 4.15.

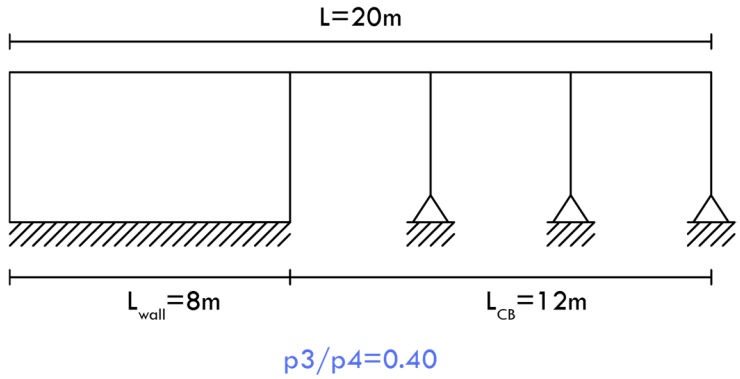


Figure 4.15 – Simple example on how the p3/p4 parameters can determine the wall ratio of a given line

As the parameter values are real numbers, solutions where the wall only fills 0.1% or 99,9% of the line may not make practical sense. In such cases, removing the wall entirely or letting it fill the entire line would be more appropriate. Therefore, the parameters are subjected to defined constraints that improve the constructability of the solution. The wall length defined as L_w, and the column-beam length is defined as L_{CB}, are calculated using equation 4.8.

$$
\begin{gathered}
if\,(w < w_{min}\ or\ w \times L < L_{min}):\\
L_{CB} = L\\
L_{wall} = 0\\
\\
else\ if\,(L - w \times L < L_{BCMin}):\\
L_{CB} = 0\\
L_{wall} = L\\
\\
else:\\
L_{wall} = L \times w\\
L_{CB} = L - (L \times w)
\end{gathered}
\tag{4.8}
$$

Where:

w	The wall ratio
w_{min}	Minimum wall ratio, defined to 0.2
L	Total length available
L_{min}	Minimum wall length, defined as 1m
L_{CBMin}	Minimum column-beam length, defined to 3m

4.3.3 APoly properties

A C# class named AdjPoly has been created, containing all the information required for a fast evaluation in the optimization process. This class includes, among other things, preprocessing the load area for every edge and every possible variation of the p1 and p2 parameters. The load area is calculated as an average of [m^2/m] for simplification, so it works independently of the p3 and p4 values. Figure 4.16 provides an example of the load area calculation. The black line represents a support line, the vector indicates the span direction, and the area is calculated for the curved edge. The load area is projected out with half a span length, as marked in blue, while the dark blue area indicates a contested area where the support line absorbs half of the area.

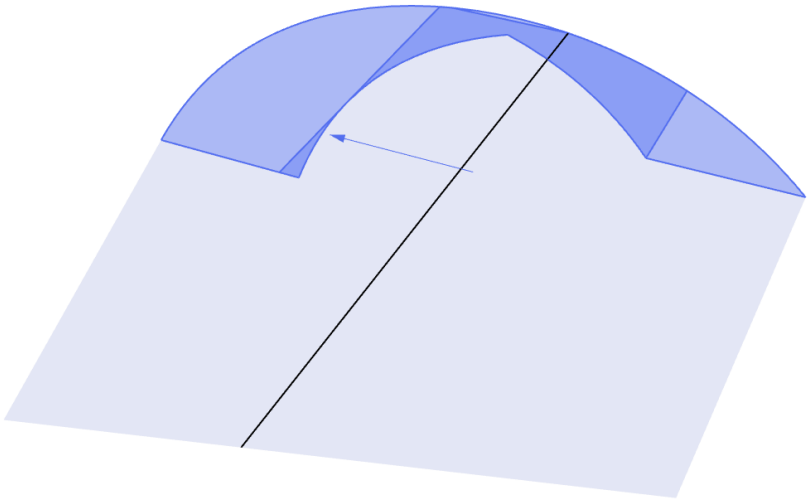


Figure 4.16 – Visual demonstration on how the unit area is calculated for a single side in an arbitrary APoly shape.

4.3.4 Design flexibility

The importance of interactivity in design tools was emphasized in the AR analysis presented in chapter 3. Consequently, three methods have been developed for generating solutions:

- **User-defined solution**
  This method allows the user to interact directly with the design methodology, to generate solutions fully based on user input without using any optimization algorithm. This is done using two modules. The first module, depicted in Figure 4.17 (a), enables users to customize the global parameterization process by creating custom APoly shapes. The second module, depicted in Figure 4.17 (b), allows users to adjust the local parameterization by defining slab directions and span length. This interaction occurs directly in the viewport by using simple mouse actions. The process is illustrated in Figure 4.18 (a), (b) and (c).

- **Hybrid solution**
  This approach allows the user to define the parameterization partially. The optimization algorithm will then finish the design with the remaining degree of freedom available. This method strikes a balance between user creativity and algorithmic assistance, allowing the user to provide some input while still benefitting from the optimization capabilities of the GA. This approach also supports the ambition that the tool should complement the design process and not dictate it. Figure 4.18 (d) illustrates an example of a partially defined solution.

- **AI-generated solution**
  This method utilizes the GA to take full control of the design process and generate complete efficient solutions. Besides the apparent advantage of using optimization, then it can sometime be beneficial for the designer to let the algorithm challenge the mental model that designers sometime get fixated on. Besides, the user always has the option to edit in the outputted solutions.

Overall, the design tool provides a range of options for generating building designs, allowing users to choose the level of control and assistance they prefer.

In addition to these primary options, users can also define the story height from 1 to 12, and the minimum ground clearance height in the range of 2.5m to 5m. It is assessed that this range is sufficient to meet the goal of the design tool to cover 80% of all building cases.

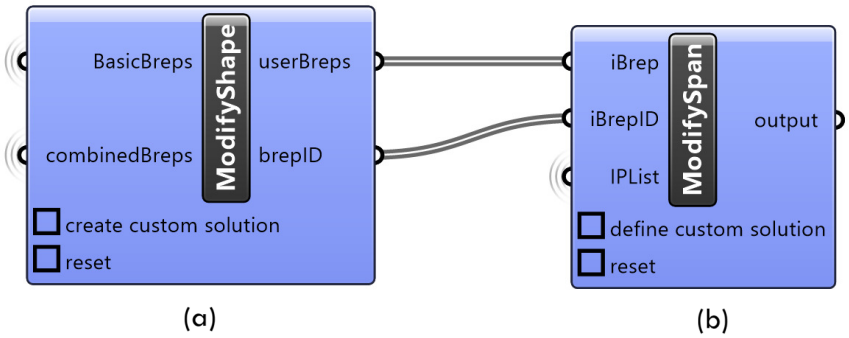


Figure 4.17 – (a) module to customize the global parameterization, (b) module to customize the local parameterization.

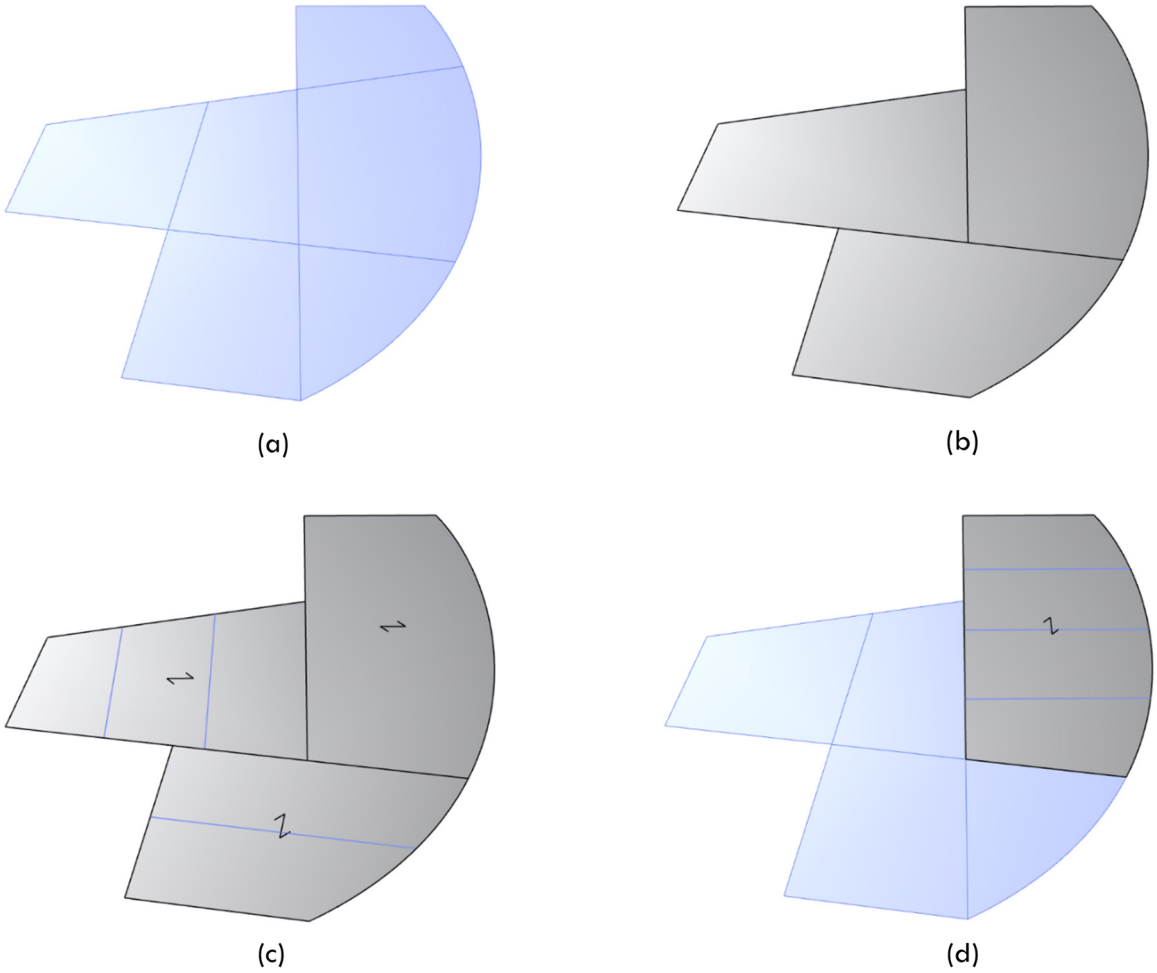


Figure 4.18 – (a) the basis shapes, (b) custom global parameterization, (c) custom local parameterization, (d) partial defined parameterization.

4.4 Loads

Simple unit loads are often used for proof-of-concept demonstrations. However, this is not viable for this project as it is intended for real-world practice. Consequently, the load calculation applied in this design tool must be of the highest fidelity level, equivalent to a level that can be used in the structural documentation. Load calculations at this level have distinct advantages, such as compensating for the lower fidelity level in the approximation models used in the evaluation models. Additionally, since the design tool is split into different modules, the high-fidelity load modules can be implemented even before the complete design tool is fully developed, thus generating value immediately.

As described in section 1.3.1, seismic loads are currently not included in the design tool. Snow loads are generated using simple theory, as the current geometrical limitations do not allow snow accumulation. The primary load types considered are dead load, live load, and wind load. These load types are typically the ones that govern the design of the structural elements. The applied load combinations depend on the specific structural typology being evaluated, which is further elaborated upon in section 4.6.

4.4.1 Self- and live load

The design tool calculates the dead load automatically, while the user defines the live load as a uniformly distributed surface load. The user defines the specific categories of live loads through an interactive process using a load module, as shown in Figure 4.19. The user selects the base shape to which the specific loaf should be assigned. The process continues until all base shapes have been assigned a live load category.

During the optimization process, there may be incidents where an APoly shape contains different categories; in such cases, the critical category is used for the entire AP shape. This simplification reduces the computational cost and is considered acceptable, as elements within the same field are typically preferred to be uniform to improve constructability.

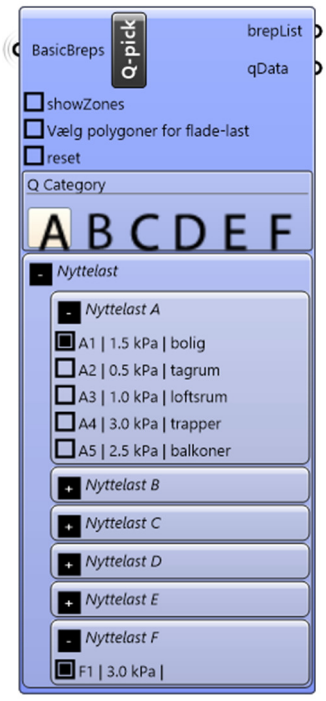


Figure 4.19 – Live load module

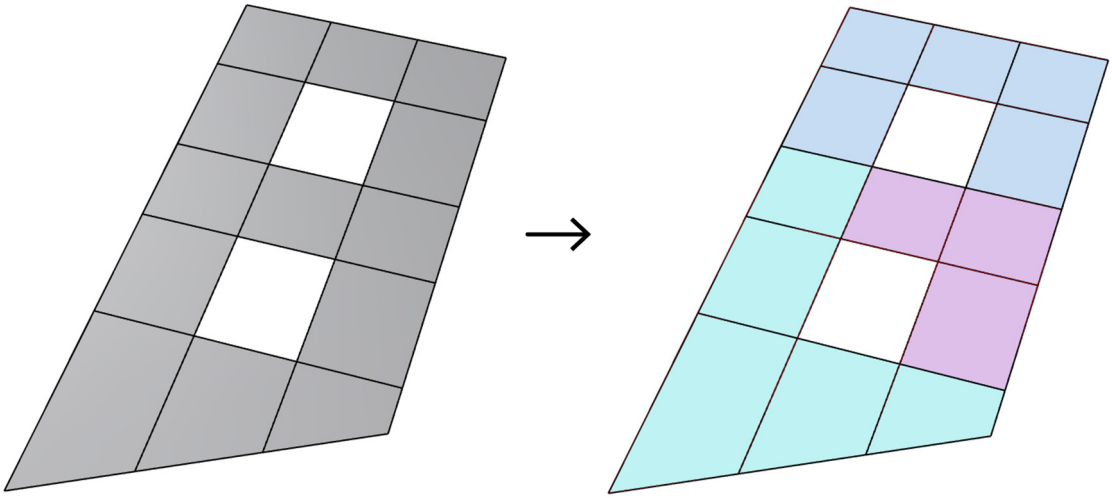


Figure 4.20 – Visualization of how the user designate live loads to the base shapes, each color represents a specific live load category.

4.4.2 Wind load

The wind load is one of the most significant forces that affect building design, particularly for tall structures. Therefore, it has been prioritized to make wind load calculations comprehensive, precise, and automated as much as possible. The high level of automation ensures that the design tool can operate without significant setup time. Calculations are based on Eurocode 1 and the national annex for Denmark [106] [107], which is a viable approach for most building shapes. While more complex and taller building projects may require wind tunnel analyses and CFD calculations, these cases are not included in the design tool's goal to handle 80% of all cases.

The most common wind calculation procedures can be generalized without significant complications. However, the more complex aspects of wind calculations are incorporated into independent modules, some of which may require user input. The remainder of section 4.4.2 elaborates on how these complex tasks are managed in the design tool.

- **Orography**:

The orography factor is an example of a wind load calculation aspect that is difficult to generalize. Typically, engineers manually analyze an elevation map and make an assessment based on the rules outlined in the national annex. In this project, this process is automated by utilizing an expert algorithm to find the most critical orography factor for the twelve primary wind directions. Additionally, the user can interact with the digital landscape and specify where the calculation should occur. The entire process is communicated through 3d environments, color gradients, and numerical values, as shown in Figure 4.21. The calculations underlying the analysis can be outputted for further review.

This approach demonstrates how the core principles of optimization, interactivity, dissemination, and automation, as shown in Figure 4.1, can be incorporated into the design tool's different modules, not just in the overall framework.

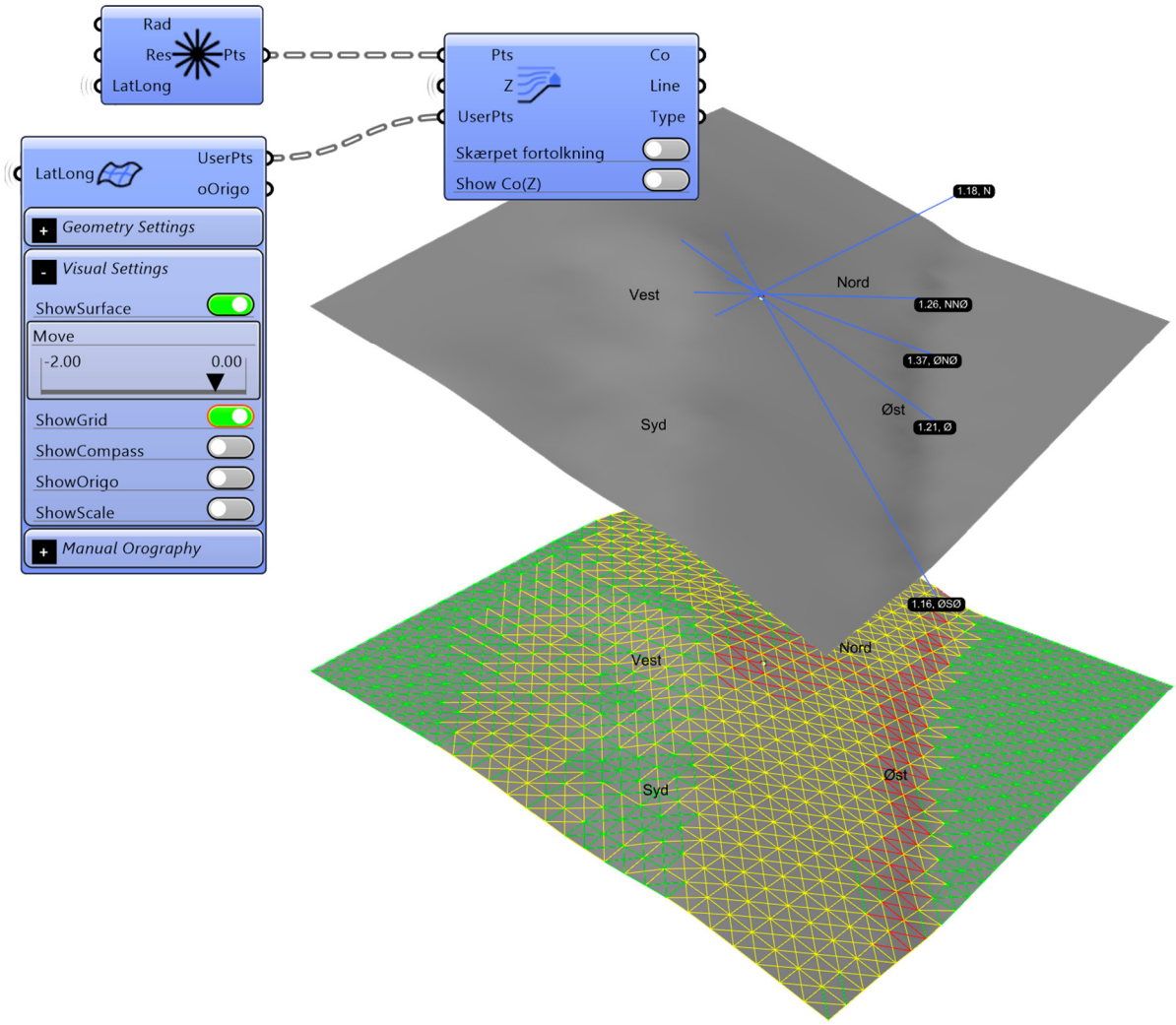


Figure 4.21 – Illustration of how the orography factor module operates.

- **Shape factors:**

Accurately determining shape factors requires wind tunnel testing or time-consuming CFD simulations. However, these methods are often impractical. The calculation methods available in the Eurocode standards are usually sufficient but only provide form factors for elementary geometric shapes. As a result, a building may be divided into simpler components, resulting in overly conservative load estimations. Neither of these options is particularly suitable for generative design, which requires a rapid evaluation response time. This issue is addressed by developing an algorithm based on the principle of how the Eurocode calculates vertical and

horizontal shape factors. The algorithm can be applied to any arbitrary geometric shape, as illustrated in Figure 4.22.

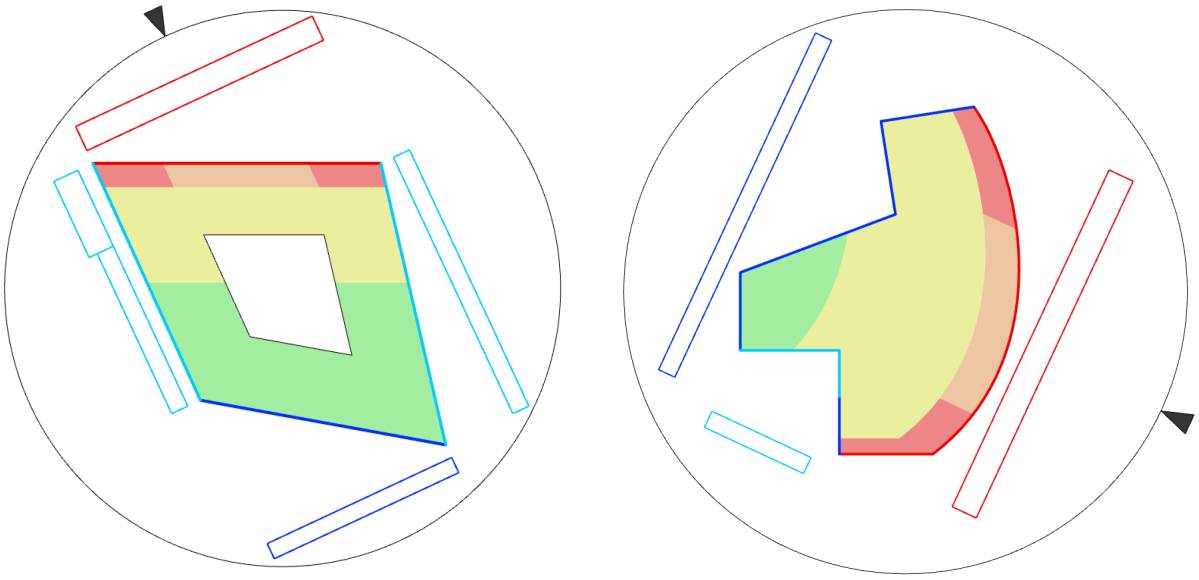


Figure 4.22 – Visualization of the horizontal and vertical shape factors. The black arrow indicates the wind direction.

- **Terrain category:**

In principle, choosing the terrain category could be automated by training a classification algorithm on a set of labeled radar images. However, this approach is not used as it is ultimately the engineer's responsibility to determine the terrain category. Nonetheless, specific sub-tasks can be automated. Radar images with correctly scaled divisions are provided to the user. Subsequently, the user can enter the correct categorization into a module. This minor feature may not significantly reduce time, but it accumulates with all the other automated tasks and ultimately frees up the engineer's time to focus on design tasks.

- **Critical wind direction:**

In computational design, evaluating every possible load scenario is often feasible. However, in generative design, one must constantly balance the increased computational load with the benefits added complexity can bring. In the case of wind calculations, it has been determined that evaluating the design for all 12 primary wind directions would impose too great a cost relative to the benefits it brings. However, using only one wind direction is not viable as the optimization algorithm does not have the necessary incentive to add any stiffness in the transverse direction, which could result in a suboptimal design. Therefore, a strategy is employed where a design is evaluated for two wind directions. The first direction, referred to as the primary direction, is the critical wind direction out of the 12 possible directions. This

design tool compares the wind load using the accumulated wind resultant instead of the local wind peak pressure. The secondary direction is the most significant wind load of the two perpendicular directions to the primary direction. Using these two wind directions as described, the algorithm will primarily place shear walls along these directions, and therefore implicitly be able to absorb horizontal wind loads in the remaining directions.

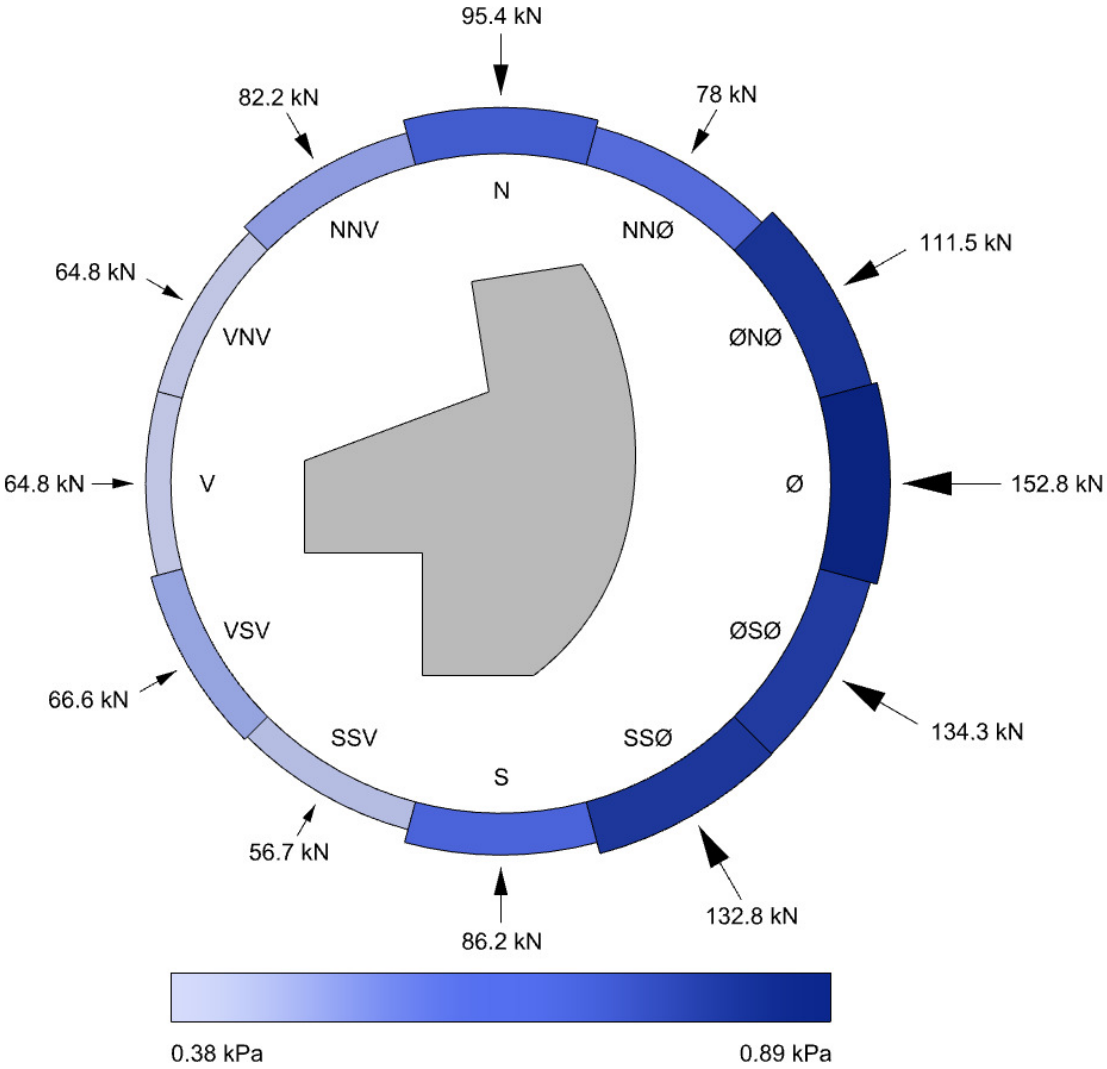


Figure 4.23 – Illustration of the critical wind directions with corresponding wind pressure values.

It should be noted that the wind directions in Figure 4.23 are depicted with equal spacing between them for clarity. In reality, the critical 12 wind directions are found by examining all wind directions with a resolution of 2° and the angles for all linear lines on the building plan. This examination is conducted in the preprocessing phase to reduce the computational load in the optimization phase. During the optimization process, the wind loads area subsequently updated only for the two critical wind directions, with respect to the revised total building height.

4.4.3 Horizontal distribution of forces

The distribution of horizontal forces is carried out for every optimization iteration, considering the individual shear walls' stiffness, placement, and orientation. It is common practice to set the maximum length of the shear walls to ten meters to avoid imposing unrealistic requirements on the joints and the internal stresses in the shear walls.

The elastic distribution of horizontal forces is executed using a modified version of the generalized distribution model, as outlined in [9]. The generalized model assumes the slab is infinitely stiff and that the stabilizing walls deform proportionally to the applied horizontal load. However, the basic version of the generalized model can only handle walls parallel to the primary axes, i.e., the x-axis and y-axis. Therefore, a modified version is necessary to handle walls that are oblique to the primary axes or curved walls. This section presents the methodology of the modified version to illustrate the alterations made to the original theory.

All wall elements are assigned a relative stiffness, denoted as S in the x- and y-directions. The general expression for this type of stiffness is defined in equation 4.9.

$$S = L^2 t \cdot |\vec{v}_{wall} \cdot \vec{v}_F| \tag{4.9}$$

Where:

$\boldsymbol{L}$	The total wall length in meters
$\boldsymbol{t}$	Default thickness, set to 0.15 meters
$\boldsymbol{\vec{v}_{wall}}$	The wall's unit vector
$\boldsymbol{\vec{v}_F}$	The unit vector of the horizontal force affecting the wall

The stiffness is adjusted based on the force exerted on the wall. This modification is achieved by multiplying the base stiffness with the dot product between the wall's unit vector and the horizontal force's unit vector. This procedure implies that walls more aligned with the exerted force will absorb a more significant proportion of the overall force than walls more perpendicular to the total force resultant.

As previously mentioned, all exterior walls are assigned a wall ratio parameter that determines the portion of the wall consisting of recesses. A differentiation is made between the stiffness of the interior walls, denoted as S, and the stiffness of the outer walls denoted as S_{OW}. This distinction controls the proportion of the horizontal forces absorbed by the outer walls. The stiffness is calculated using equation 4.10.

$$S_{OW} = (L - w)^2 t |\vec{v}_{wall} \cdot \vec{v}_F| \tag{4.10}$$

Where:

$\boldsymbol{w}$	The wall ratio for the outer wall element
$\cdot$	The dot product

The method handles the curved element by dividing them into sections that are a maximum of 10 meters long. The curve is then divided into smaller linear segments with a maximum length of 1 meter and scaled with respect to the total length, so the geometry operates in the realm of the stiffness values. The sub-elements are subsequently projected onto the two primary axes. The process is illustrated in Figure 4.24 for the projection onto the x-axis.

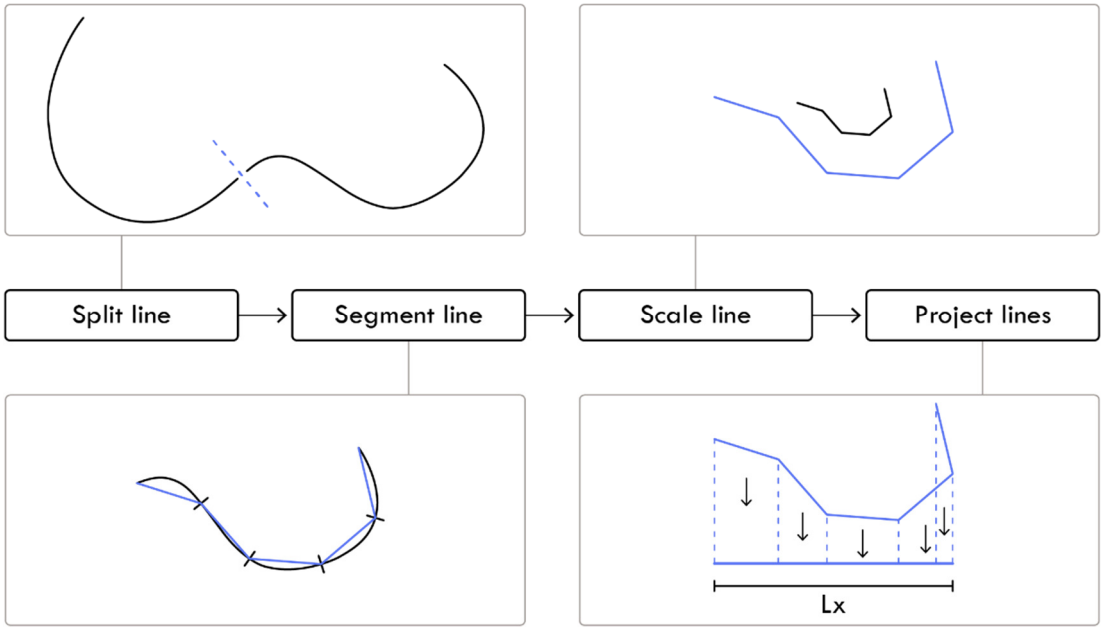


Figure 4.24 – Visualization of how the stiffness on the x-axis is found for curved line segments.

The method operates with a default thickness, as the actual thickness remains unknown until the evaluation process commences. The thickness differentiates between linear wall elements and those derived from curved walls. In the case of curved wall elements, their projection onto the primary axes may result in overlapping; hence an adjusted thickness is determined for these elements. This calculation is presented in equation 4.11. Finally, the total stiffness for the entire set of n wall elements is computed, as expressed in equation 4.12.

$$t_{adj} = \frac{\sum L_i \cdot t}{L} \tag{4.11}$$

Where:

$\boldsymbol{L_i}$	Denotes the smaller projected segment lines on a given axis

$$Sx = \sum_{i=0}^{n} Sx_i$$
$$Sy = \sum_{i=0}^{n} Sy_i \tag{4.12}$$

Subsequently, the coordinates for the global shear center are calculated as (x_{SC}, y_{SC}), as shown in equation 4.13.

$$x_{SC} = \frac{\sum_{i=0}^{n} x'_i}{Sx}$$
$$y_{SC} = \frac{\sum_{i=0}^{n} y'_i}{Sy} \tag{4.13}$$

Where:

x'	Denotes the distance along the x-axis from a given origin to the centre point of wall i.
y'	Denotes the distance along the x-axis from a given origin to the centre point of wall i.

The torsional stiffness Iw is then calculated as

$$I_W = \sum_{i=0}^{n} Sx_i \cdot {x_i}^2 + \sum_{i=0}^{n} Sy_i \cdot {y_i}^2 \tag{4.14}$$

Where:

$$x_i = {x'}_i - s_{SC}$$
$$y_i = {y'}_i - y_{SC} \tag{4.15}$$

The torsional moment Mw is calculated positively in a counterclockwise direction using equation 4.16 and illustrated in Figure 4.25.

$$M_W = (FY \cdot x_{FY}) - (FX \cdot y_{FX}) \tag{4.16}$$

Where:

FX	The wind resultant in kN along the x-axis
FY	The wind resultant in kN along the y-axis
x_{FY}	The distance on the x-axis from FY to the centre point of the shear centrum (SC)
y_{FX}	The distance on the y-axis from FX to the centre point of the shear centrum (SC)

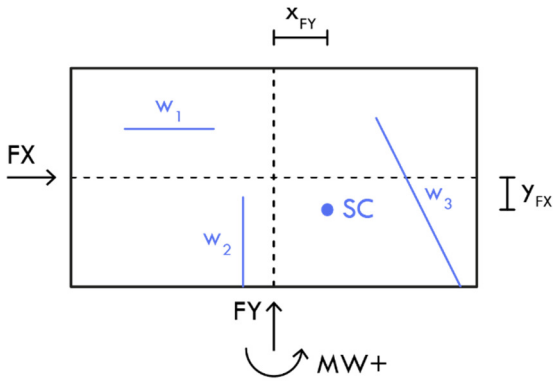


Figure 4.25 – Illustration of how the torsional moment is calculated.

The ideal forces are then calculated using equation 4.17.

$$FX_i = Sx_i \cdot \left(\frac{FX}{Sx} + x_i \cdot \frac{MW}{IW}\right)$$
$$FY_i = Sy_i \cdot \left(\frac{FY}{Sy} + y_i \cdot \frac{MW}{IW}\right) \quad (4.17)$$

Where:

$\boldsymbol{FX}$	The wind resultant in kN along the x-axis
$\boldsymbol{FY}$	The wind resultant in kN along the y-axis

When the ideal forces are exerted on a wall, they will not always perfectly align with the wall's directional vector, leading to a residual force. equation 4.19 demonstrates the computation of this residual force, which is visualized in Figure 4.26. The calculation method involves converting the wall's directional vector into a unit vector and multiplying it by a factor α, derived from equation 4.18. This factor α describes the relationship between the most significant resultant force exerted upon the wall and its corresponding reaction resultant.

$$\alpha = max \begin{cases} \alpha_x = \dfrac{\overrightarrow{F_X}}{\overrightarrow{W_X}} \\ \alpha_y = \dfrac{\overrightarrow{F_Y}}{\overrightarrow{W_Y}} \end{cases} \quad (4.18)$$

Where:

$\boldsymbol{\overrightarrow{F_X}}$	The wind resultant along the x-axis
$\boldsymbol{\overrightarrow{F_Y}}$	The wind resultant along the y-axis
$\boldsymbol{\overrightarrow{W_X}}$	The wall's reaction along the x-axis
$\boldsymbol{\overrightarrow{W_Y}}$	The wall's reaction along the y-axis

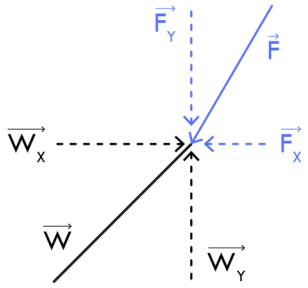


Figure 4.26 – Visualization of how the residual force occurs.

The residual force F_{res} is then calculated as the difference between the reaction vector and the force vector, as shown in equation 4.19.

$$\overrightarrow{F_{res}} = \vec{W} \cdot \alpha + \vec{F} \tag{4.19}$$

This process is repeated for every wall element, and the total residual forces are applied back to the building and distributed to the different walls. Therefore, the problem is an iterative approximation that can be repeated until a desired level of accuracy is achieved. Testing has shown that the residual forces were virtually eliminated after two iterations. However, this project will only utilize one iteration to reduce the computational load, which is deemed acceptable since it was observed that the inaccuracy in the equilibrium was insignificant and always nearly perpendicular to the direction of the load. Thus, there will always be sufficient load capacity to absorb the residual forces in the transverse direction. As previously mentioned, the building will also be calculated for wind loads in the transverse direction.

4.5 Constructability

Constructability can be defined as the utilization of construction knowledge and experience to achieve superior designs [108]. In this project, constructability manifests itself through a range of different measures. As mentioned, the proposed design tool operates in design cycle one, which uses a low level of detail (LOD). Therefore, some constructability measures may be more suitable for other loops where the level of detail is greater. Nonetheless, constructability can still be integrated into the design methodology to generate more buildable designs. In general, incorporating constructability considerations into the design process reduces the risk of major changes later and provides a more accurate representation of the final design. Constructability is also interpreted as increased flexibility, enabling designers to consider practical and aesthetic requirements. Some of the constructability measures incorporated into the design tool are listed as follows:

- Ensuring flexibility by customizing solutions as explained in section 4.3.4.
- Implementing parameters to control the amount of horizontal force transferred to the outer walls.

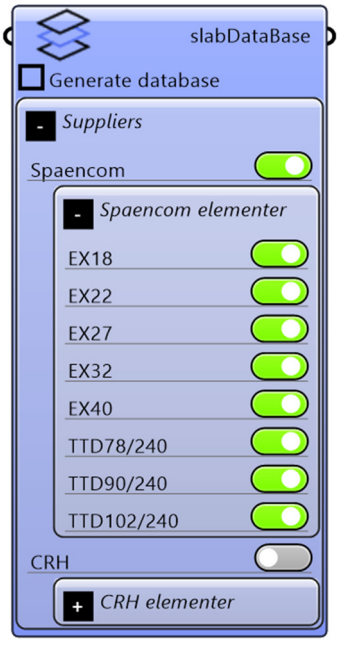


Figure 4.27 – Slab database module

- Implement parameters that incentivize the algorithm to use fewer columns in the beam-column lines.
- Ensuring that no shear wall is longer than ten meters.

Another important constructability aspect is the ability to choose a specific supplier of RC elements. This feature is especially relevant since contractors often pre-select suppliers for a particular project. By incorporating this information, the algorithm can generate solutions that solely utilize RC elements available from the chosen supplier. Figure 4.27 illustrates the available options, where the user may select from various slab types and two distinct suppliers. Future versions of the design tool incorporate additional suppliers and distinguish between other structural element types. These future measures are discussed in-depth in section 6.3.

4.6 Evaluations modules

This section presents an overview of the evaluation process for the utilized structural element types, along with the parameters used to construct surrogate models that predict their geometry and cost. The design tool is applied to a building structure comprised of four distinct RC element types:

- RC Slabs
- RC Beams
- RC Columns
- RC Walls

Evaluation modules are created for all these structural types, including a beam-column system evaluation module built using the column- and beam evaluation modules. All evaluation modules serve as the basis for a range of surrogate models that can predict the geometry and cost of the different RC element types using a series of inputs. Surrogate modeling was necessary to achieve the fast response time required for a performance-based generative design approach.

The surrogate models incorporated in the tool are constructed and trained using many optimized samples. Each sample is divided into two categories: the external parameters, which serve as input to the surrogate model, and the design parameters that the surrogate model aims to predict.

The external parameters describe the conditions that affect the elements, such as load values and geometrical constraints. These input parameters are generated using a one-shot sampling strategy. An adaptive sampling strategy is considered unnecessary as the evaluation

modules are computationally cheap to execute. The Sobol sampling strategy is chosen to execute a one-shot sampling approach because previous experience with sensitivity analyses has shown that the method is effective in producing space-filling yet stochastic samples.

The process of generating the corresponding output parameters, denoted as the optimized design parameters, is more complicated. Each sample has to undergo an optimization process to find the optimal design parameters to create the database on which a surrogate model is trained. This process is illustrated in Figure 4.28. The complexity of the evaluation model dictates the burden of this process. A brute force algorithm can be used to find the optimized design parameters for simple element typologies with few design parameters.

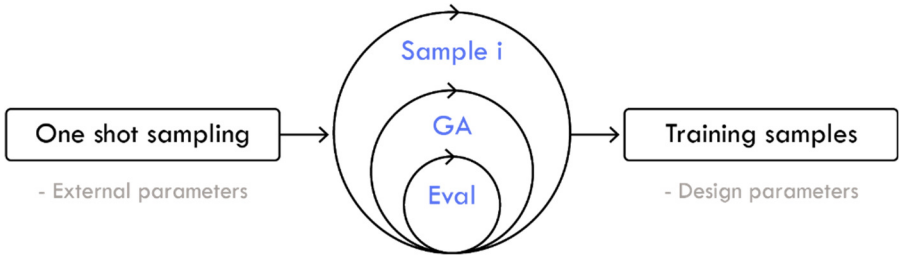


Figure 4.28 – Generalized representation of how optimized training samples are generated.

A comprehensive performance comparison analysis was conducted to identify which surrogate model had the best prediction capabilities. The analysis was carried out on the column evaluation module, and the following methods and libraries were tested:

- Kriging
- ML.NET
- Matlab regression library
- Convolutional Neural Networks

The ML.NET and Matlab regression libraries contain methods such as Support Vector Machines, Decision Trees, Gaussian Models, and more. The models were tested for their ability to predict different design parameters, including the reinforcement size (As size), reinforcement level (As Lv), the width of the column (W), concrete class (Fck), and cost. The performance assessment was based on various metrics, such as the correlation value R^2, as shown in Table 4.2.

Table 4.2 – The correlation value R^2 shown for different surrogate model types and prediction objectives.

Model	As size	As Lv	W	Fck	Cost
NN multiple outputs	0.52	0.68	0.99	0.76	0.99
NN single output	0.64	0.77	0.99	0.74	0.99
Kriging with noise	0.23	0.41	0.97	0.36	0.96
Matlab reg. library	0.23	0.42	0.97	0.40	0.96
ML.NET library	0.14	0.42	0.98	0.36	0.80

During testing, it became apparent that the Kriging model was sensitive to the number and density of samples and often failed due to the Cholesky factorization collapsing. As a result, it was decided to continue working with neural networks with a single output, as these models had demonstrated good prediction performance and a high level of robustness. However, the optimal configuration also depends on the problem that the NN tries to emulate, as high nonlinear problems require more layers than problems that scale more linearly. Therefore, conducting a grid search analysis for each NN surrogate model is necessary to achieve optimal performance. Although this approach is computationally expensive, it is easy to set up and reliable for finding the best settings. The range of hyperparameters in the grid search analysis is differentiated for regression models and classification models as follows:

Hyperparameter	Range
Learning functions for regression	{SCG, CGF, CGB, CGP, LM}
Learning functions for classification	{SCG, CGF, OSS, BFG}
Transfer functions	{Tansig, Logsig, Satlins}
Number of layers	{1, 2, 3, 4}
Number of neurons	Number of parameters * {1, 2, 3, 4}

Table 4.3 – Range of hyperparameters used for the grid search. The abbreviations are detailed in section 2.4.5

Every grid search is conducted by finding up to ten promising neural network settings. These models are then executed 30 times to further examine their performance, given that the process is stochastic and yields fluctuating performance levels. The choice of algorithm is then based on the mean and variation value of the R^2 and MSE error metrics. The final settings for each surrogate model are listed in Appendix B.

4.6.1 RC Slab module

As mentioned in section 4.5, the range of available slab types is restricted to those offered by chosen suppliers. Therefore, a dynamic slab library is constructed based on the user's selection. This comprehensive database encompasses all material, structural, and geometrical properties for each specific slab option. In essence, the sole design parameter is the choice of slab type, thereby enabling the utilization of a brute force algorithm to determine the optimal slab for every possible combination.

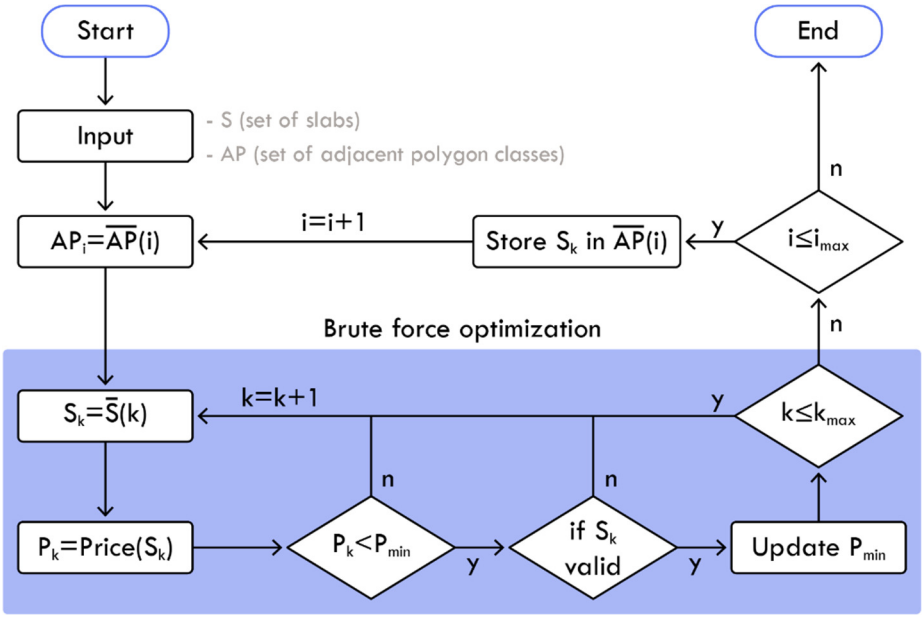


Figure 4.29 – Illustration of how the brute force algorithm finds the optimized slab(k) for every APoly shape(i).

The brute force sorting algorithm is illustrated in Figure 4.29. The algorithm operates on a set of slabs denoted as $\bar{S}$, and a set of valid APoly shapes denoted as $\overline{AP}$. The process iterates through every APoly shape (i) and checks every slab option (k) in the database. Firstly, it is verified that the current slab is less expensive than the current best option. If it is, it undergoes a validity check to verify compliance with all relevant constraints in the ultimate limit state (ULS) and serviceability limit state (SLS). If the slab satisfies the conditions, the choice of slab for this combination and the minimum price is updated. It should be noted that Figure 4.29 provides a visual simplification of the process since the investigation is actually performed for every possible load direction and span length in the APoly shape, as depicted earlier in Figure 4.12 and Figure 4.13.

The constraints examined for the slab include:

- Verification of the maximum bending moment in ULS.
- Verification of the balance load to ensure long-term deformations.
- Verification of the cracking moment.
- Verification of the natural frequency.

It is assumed that the eigenfrequency must not be less than 5 Hz since a more detailed acceleration analysis is unlikely to be successful otherwise. The eigenfrequency is calculated based on the transformed cross-section, while the manufacturer provides the remaining structural properties.

4.6.2 RC beam module

Chapter 3's AR analysis concluded that it was important that the generated elements were available at the chosen suppliers. In the current version of the program, only prestressed KB beams are available since these beam types are suitable for supporting the available slab variants. Unlike slab elements, KB beams can be produced in so many variations that it is practically impossible to examine all possible combinations. Therefore, the GA approach shown in Figure 4.28 is employed.

The supplier dictates the design freedom by specifying how their structural element products may vary. This information was then used to construct the parametric model, where the degree of freedom is visualized in Figure 4.30

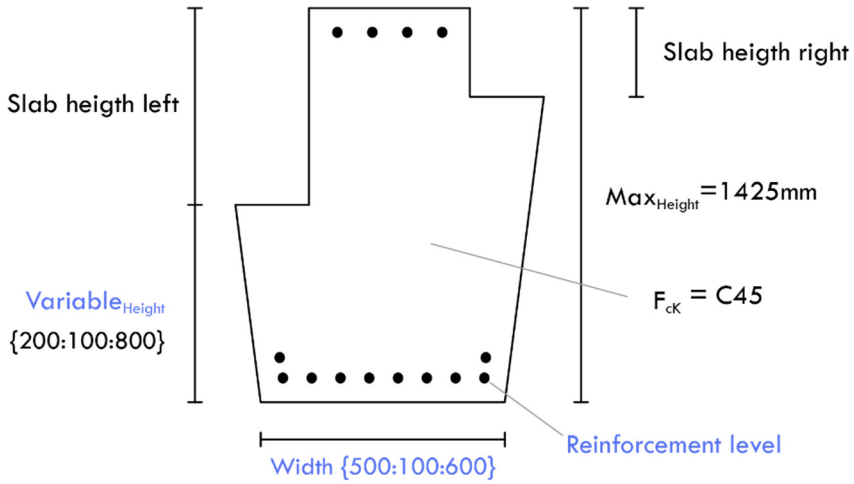


Figure 4.30 – Illustration of the variables for a KB beam, blue indicates design parameters.

Another important conclusion from the AR analysis was that the beam models must adequately account for the ULS and SLS constraints to ensure that the elements used are realistic in a detailed analysis. Therefore, the following verification analyses in ULS and SLS are included in the evaluation model:

- Verification of the maximum bending moment.
- Verification of the balance load to ensure long-term deformations.
- Verification of deformations under short-term and long-term loads.
- Verification of the eigenfrequency.

It is worth noting that the current analysis does not account for shear reinforcement. This decision was made based on the assessment that including the aspect would not significantly impact the final geometry or cost difference.

In addition to the listed verification procedure, it is ensured that the cross-section is neither over- nor under-reinforced and that cracks will not occur due to prestressing forces. These constraints are achieved by limiting the minimum and maximum degree of reinforcement available and imposing geometrical limitations. The reinforcement level parameter is then converted from a normalized domain to the range determined by the prementioned minimum and maximum reinforcement degree. The reinforcement level parameter is thereby translated into an actual number of prestressed reinforcement bars. This converted range varies depending on the external parameters. All the external parameters for the RC beam elements are listed in Table 4.4, along with their defined range.

Table 4.4 – External parameters for the RC beam evaluation model, with corresponding range.

External parameters	**Range**
Beam Length [m]	{3-12}
Characteristic live load (q) [kN/m]	{0-126}
Characteristic dead load (g) [kN/m]	{0-104}
Live load category	{A, B, C, D, E, F}
Slab height left [mm]	{180-1020}
Slab height right [mm]	{180-1020}

It is important to note that the range defines the spectrum on which the surrogate model has been trained. Therefore, the model cannot predict a solution if an input value falls outside this range. The limit values were established based on experience and an analysis of the maximum expected values given the current geometric constraints encoded in the parametric

methodology. The design parameters the GA can operate with, and their corresponding range are listed in Table 4.5.

An interesting observation emerged when analyzing the optimized samples: the wide version of the KB was never considered cost-effective. Consequently, the SU model has never encountered a scenario with a wide beam during the training sessions and thus will not select this option. This exclusion makes sense from a static perspective as the beam height is squared while the width is multiplied by 0.5 when calculating the stiffness. However, other factors may necessitate a wider beam, such as providing sufficient bearing length for larger slab structures. Such considerations could be incorporated into the evaluation module in future versions.

Table 4.5 – Design parameters for the RC beam evaluation model, with corresponding range.

Design parameters	Range
Variable height [mm]	{200:100:800}
Width [mm]	{500:100:600}
Reinforcement level	Depends on geometry

A training set consisting of 2500 optimized samples was generated and used to construct two surrogate models for the KB beam model, one for predicting geometry and the other for predicting the cost. The prediction plots of these two models are visualized in Figure 4.31.

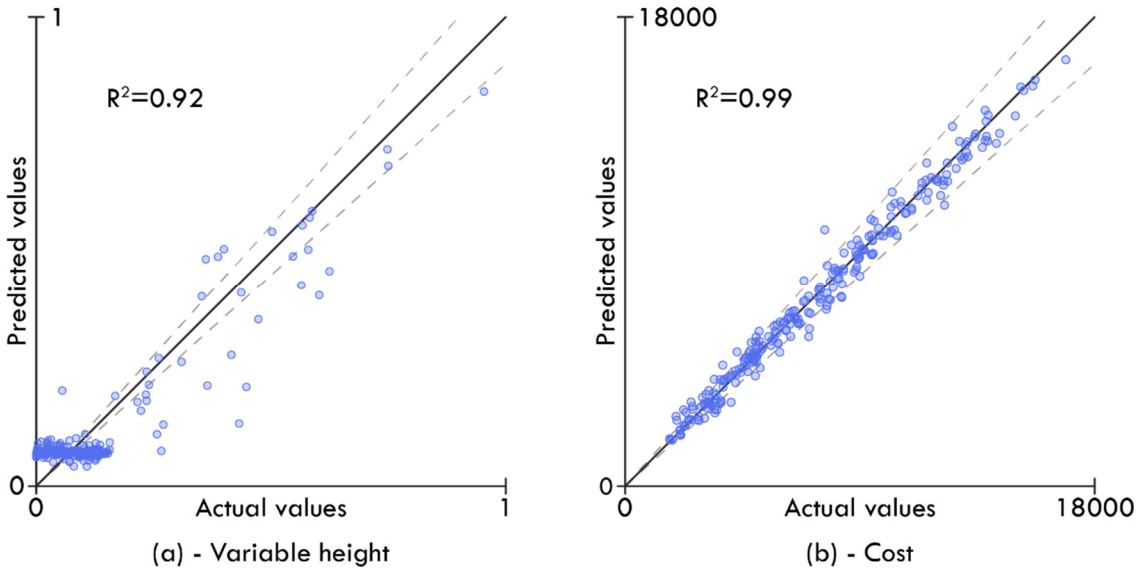


Figure 4.31 – Illustration of the prediction plots for the KB beam evaluator, (a) prediction of the variable height parameter, (b) prediction of the cost output

In Figure 4.31 (a), clustering around the lower values is observed. This clustering is partly due to the parameter being remapped to a specific interval, thus representing the lowest and

most common interval. Additionally, the variable height parameter only represents the height under the slab shelf. Therefore, the applied slab height implicitly dictates a significant portion of the height. Furthermore, there is a correlation between large slab heights and the need for high cross-sections, which explains why many training examples do not require extra height.

Figure 4.31 (b) illustrates a uniform distribution and a high correlation value because the cost is almost linearly correlated to external parameters such as slab height, span length, and load values, making it easier to predict. Based on these observations, it is concluded that the surrogate models exhibit good prediction capabilities, as demonstrated by the error metrics and prediction plots.

4.6.3 RC column module

The parametric model developed for the RC column is also limited to generating geometry that the selected suppliers can provide. For the sake of simplicity, only square columns, which are the most common variant, are used. Due to the numerous design parameters and variations, a GA is again used to generate optimal samples, as shown in Figure 4.28. The degree of freedom is visualized in Figure 4.32.

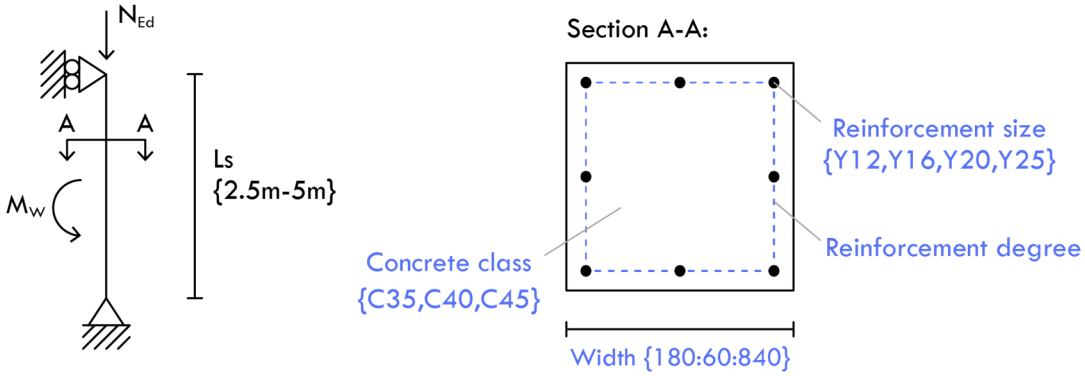


Figure 4.32 – Illustration of the variables for a RC column, blue indicates design parameters.

Table 4.6 outlines the design parameters and their corresponding ranges based on the selected suppliers' design options. While it would have been possible to simplify these options by utilizing a singular reinforcement diameter or concrete grade, it is believed that incorporating these additional degrees of freedom allows the algorithm to better adapt to the external forces that affect the column.

Like the RC beam, the range of reinforcement degree is dependent on other parameters. By default, the range is determined by how many reinforcement bars can be physically inserted into the cross-section. However, these default values can be overridden following the rules for over and under-reinforced cross-sections.

Table 4.6 – Design parameters for the RC column evaluation model, with corresponding range.

Design parameters	Range
Width [mm]	{180:60:840}
Reinforcement size [mm]	{Y12, Y16, Y20, Y25}
Reinforcement level	Depends on geometry
Concrete class	{C35, C40, C45}

Table 4.7 outlines the external parameters. The height of the column is denoted as Ls, which represents the buckling length of the column. The range of this parameter is based on experience, and it can be increased in future versions if necessary. However, any changes will require retraining the surrogate models. The buckling length is in the evaluation model defined as the story height minus the slab height, as illustrated in Figure 4.33. H_{column} is the only user input and consists of the sum of the minimum ground clearance height and an optional extra installation height.

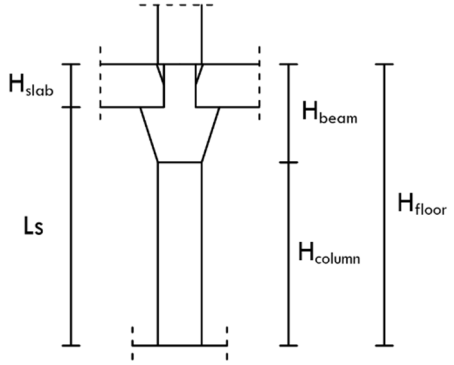


Figure 4.33 – Illustration of the different height values used in relation to the column calculations.

The other external parameters are the normal axial force and the bending moment that affect the column in ULS. The range of forces that affect the column is based on the maximum loads that can occur with the current limitations in the parameterization model.

Table 4.7 – External parameters for the RC column evaluation model, with corresponding range.

External parameters	Range
Ls [m]	{2.5-5}
NEd [kN]	{0-32000}
M_W [kNm]	{0-300}

It is considered sufficient to investigate the column in the ULS to ensure a realistic column design. The column is statically treated as a non-tensioned reinforced, simple supported column. Buckling is considered, along with first and second-order effects. A detailed

evaluation is carried out using an interaction diagram for several reasons. Firstly, constructing an interaction diagram is computationally cheaper than iteratively finding a strain value that creates equilibrium in the cross-section. Secondly, engineers in the organization are accustomed to using an interaction diagram. Furthermore, additional load combinations can easily be included later.

A training set of 2500 samples was generated and optimized using the GA. Subsequently, a grid search analysis was performed to determine the optimal hyperparameters of the neural network. Two surrogate models were constructed to predict the column width and cost, respectively. The corresponding prediction plots of these two models are visualized in Figure 4.34.

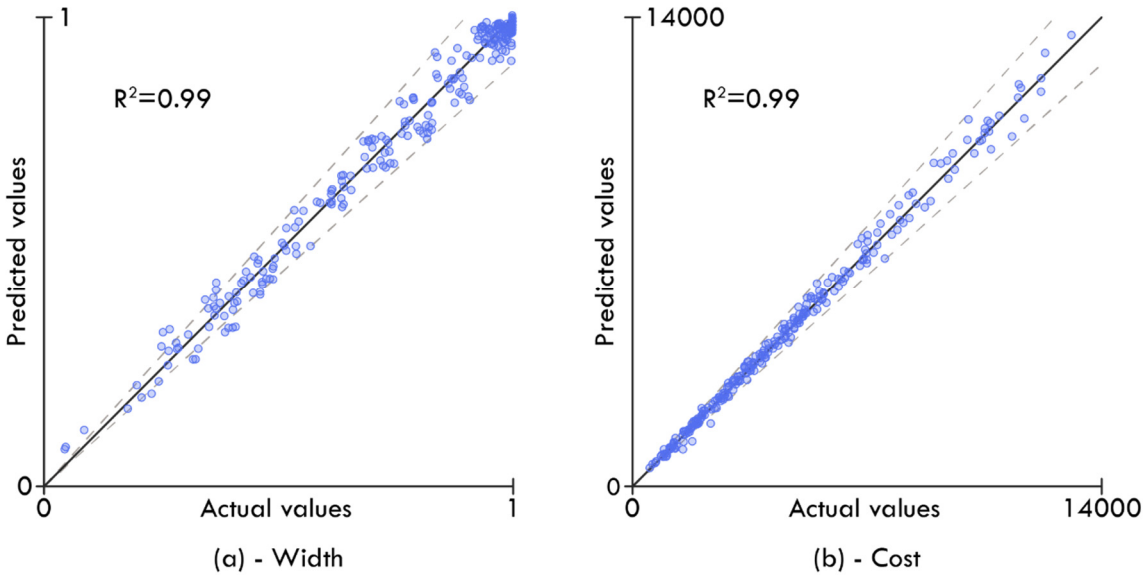


Figure 4.34 – Illustration of the prediction plots for the RC column evaluator, (a) prediction of the variable height parameter, (b) prediction of the cost output in DKK

Figure 4.34 (a) exhibits small clusters due to the normalized input being remapped to widths that vary in small intervals. The dense clustering around the 1:1 area results from custom alterations made in the first generations of solutions in the GA, where all width parameters are set to one to ensure that some solutions would be valid in the first generation. As the optimization progresses, the algorithm is incentivized to reduce cost by reducing the design parameters. However, the optimization process is stochastic, and an optimal solution is not always guaranteed. Consequently, a larger proportion of solutions with maximum width may occur, leading to clustering. Increasing the number of generations and population size or including a solution with all its design parameters set to one could potentially reduce clustering. However, this is not considered a pressing issue. It can be concluded that both models demonstrate satisfactory prediction capabilities, with only a few outliers observed.

4.6.4 RC beam-column line module

The main goal of this module is to estimate the optimal distance between columns in a beam-column module. The distance is the only design parameter specified in Table 4.8 and illustrated in Figure 4.35. A brute force algorithm is employed, using a resolution of 1000 steps over the range of the design parameter to determine the optimal distance.

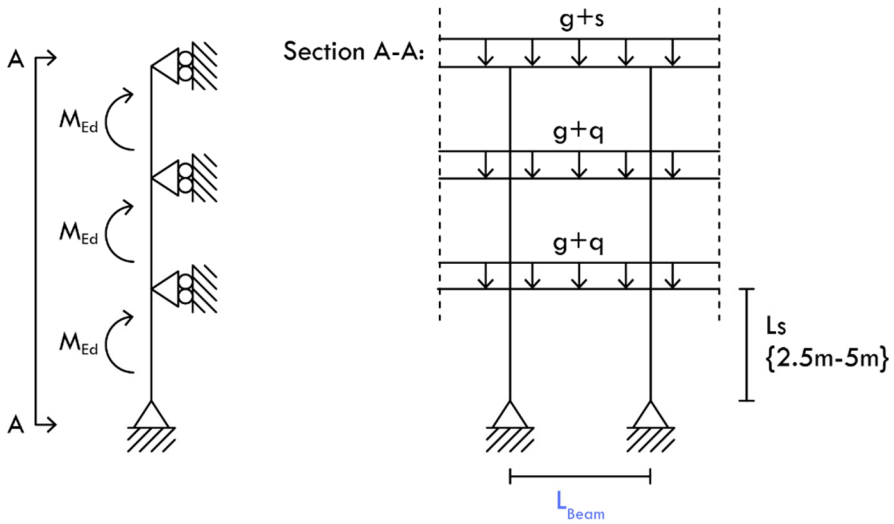


Figure 4.35 – Illustration of the variables for a RC beam-column module, blue indicates design parameters.

Table 4.8 – Design parameters for the RC beam-column evaluation model, with corresponding range.

Design parameters	Range
Beam Length [m]	{3-12}

Table 4.9 – External parameters for the RC beam-column evaluation model, with corresponding range.

External parameters	Range
Ls [m]	{2.5-5}
NEd [kN]	{0-32000}
MEd [kNm]	{0-300}
Beam Length [m]	{3-12}
Characteristic live load (q) [kN/m]	{0-126}
Characteristic dead load (g) [kN/m]	{0-104}
Live load category	{A, B, C, D, E, F}
Slab height left [mm]	{180-1020}
Slab height right [mm]	{180-1020}
Column price factor (CPF)	{1-4}

The external parameters for the beam-column module are listed in Table 4.9 and mainly consist of the same parameters as those specified for the RC column and RC beam. The unique external parameter is a user-defined parameter denoted as the column reduction factor. This factor is included to reduce the number of columns, even if it would be more expensive. It is typically used from the architect's perspective. The column price factor (CPF) is multiplied by the cost of the columns, resulting in a fictitious higher cost. This adjustment alters the cost ratio between columns and beams and incentivizes the algorithm to reduce the number of columns.

The optimal distance between columns also depends on the total length of the beam-column line. However, the decision was made not to treat the total length as an external parameter, as doing so would result in a substantial and complex solution space. Additionally, setting an upper bound for the parameter would be necessary, which could severely limit the tool's use cases and result in errors when the module is applied in practice. Therefore, an alternative approach was adopted, where the beam-column line is assumed to be infinitely long, and the cost is calculated as an average cost per meter (CPM) using equation 4.20.

$$CPM = \frac{(cost_{column} \cdot CPF + cost_{beam})}{L_{beam}} \tag{4.20}$$

Where:

$\boldsymbol{cost_{column}}$	Total cost of one column
$\boldsymbol{cost_{beam}}$	Total cost on one beam
$\boldsymbol{CPM}$	Cost per meter (DKK/m)

The RC beam-column surrogate model employs a hierarchical approach, where the evaluator is built based on the existing surrogate models for the RC beam and RC column. These surrogate models form the basis for the new surrogate model, effectively emulating their response. In other words, an approximation model is developed by combining two existing approximation models. This concept is illustrated in Figure 4.36.

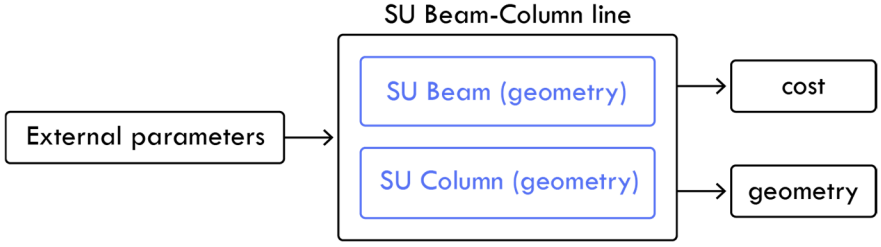


Figure 4.36 – Principal illustration of hierarchical surrogate modelling approach used to predict the geometry and cost of the SU Beam-Column line.

Figure 4.36 is a simplistic representation of the hierarchical surrogate modeling approach. The cost of the SU beam-column module is based on the predicted geometry of the beam and column modules rather than their cost prediction functions.

Because the beam-column line is treated as infinite, the optimal design parameters that are outputted are likely, not divisible by the actual length, potentially resulting in very short outer beam lengths. Therefore, an approach is used where the total length of the beam-column line is divided by the ideal beam length, resulting in a number that indicates the idealized number of divisions. This number is a natural number and is then rounded up and down to find the two nearest integers. These integers can be converted into two different beam lengths divisible by the length of the entire beam-column line. These two values are then inputted into a function that calculates the CPM value using the existing column and beam SU models and the beam lengths as input. This approach allows for investigating which beam spacing is the most cost-effective using equation 4.21.

$$if\left(F_{cost}\left(\frac{L_{tot}}{\left\lceil\frac{L_{tot}}{L_{ideal}}\right\rceil}\right) < F_{cost}\left(\frac{L_{tot}}{\left\lfloor\frac{L_{tot}}{L_{ideal}}\right\rfloor}\right)\right):$$

$$L_{best} = \frac{L_{tot}}{\left\lceil\frac{L_{tot}}{L_{ideal}}\right\rceil} \tag{4.21}$$

$$else:$$

$$L_{best} = \frac{L_{tot}}{\left\lfloor\frac{L_{tot}}{L_{ideal}}\right\rfloor}$$

Where:

$\boldsymbol{F_{cost}}$	A function that calculates the CPM of a beam-column line using the beam length as input and the existing SU models for the column and beam.
$\boldsymbol{L_{tot}}$	The total length of the beam-column line.
$\boldsymbol{L_{ideal}}$	The idealized beam length.
$\boldsymbol{L_{best}}$	The beam length resulting in the most cost-effective beam-column line and is divisible by the total length.
$\lceil - \rceil$	Ceiling function to round up to the nearest integer.
$\lfloor - \rfloor$	Floor function to round down to the nearest integer.

The principle of this approach is visualized in Figure 4.37, where the y-axis represents CPM, and the x-axis represents the beam length. The graph itself illustrates the cost-benefit for the distance ratio between columns. The shape of this graph will change according to the values of the external parameters.

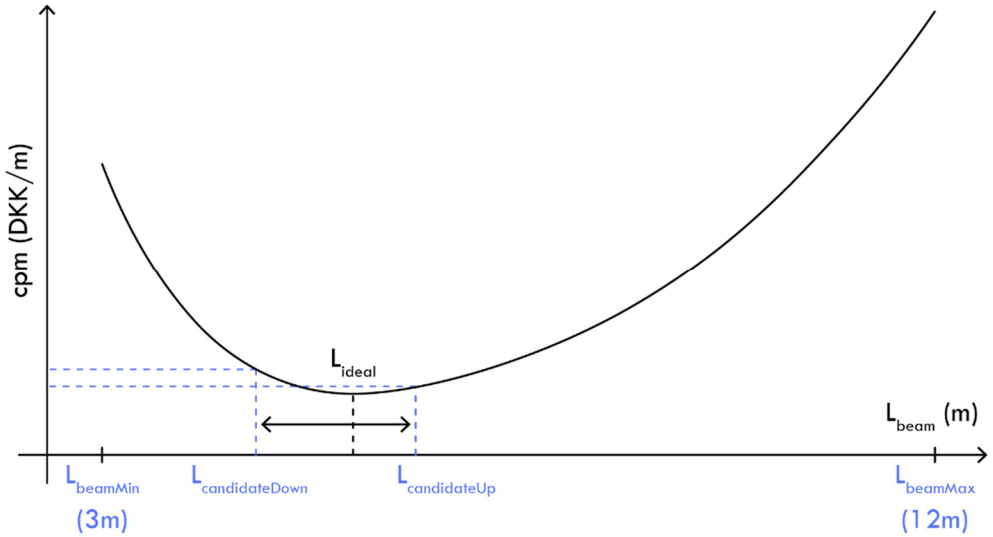


Figure 4.37 – Illustration of how the ideal beam length is rounded up and down to lengths that are divisible with the total beam-column length. The most cost effective of the two candidate lengths are then used.

The surrogate model used to predict the idealized beam length for the beam-column system was trained on 3000 training samples, and the resulting prediction plot is shown in Figure 4.38.

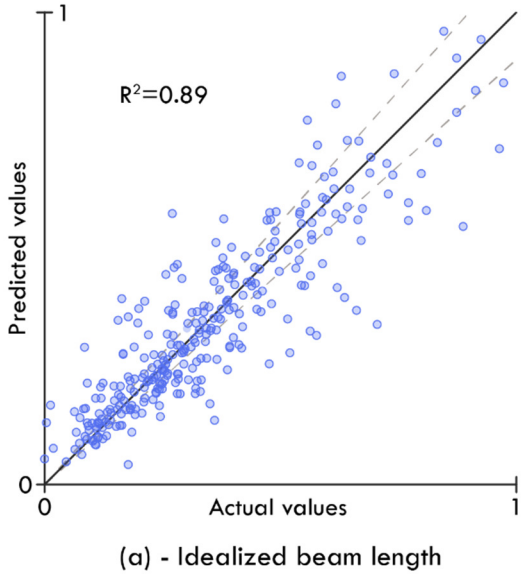


Figure 4.38 – Illustration of the prediction plots for the surrogate model that predicts the idealized length.

It can be observed that the correlation value is slightly smaller than the remaining prediction models. This difference can be partly attributed to the more significant number of external parameters in the evaluation model and the fact that the evaluation is based on two existing surrogate models. A correlation value of 0.89 is still considered very good, and the increased uncertainty is not a significant issue as the predicted value is rounded up and down. Therefore, this variation is very unlikely to impact the final beam spacing.

4.6.5 RC wall module

RC element suppliers tend to offer wall element components that are relatively similar. Therefore, the module is considered to work generally, even though minor variations may occur. In such cases, the procedure used by the chosen suppliers is implemented. A distinction is made between external and internal walls to consider that external walls are affected by wind loads acting perpendicular to the surface. Additionally, a parameter denoted the recess ratio determines the percentage of the external walls that consist of recesses. The static system and the degrees of freedom are illustrated in Figure 4.39.

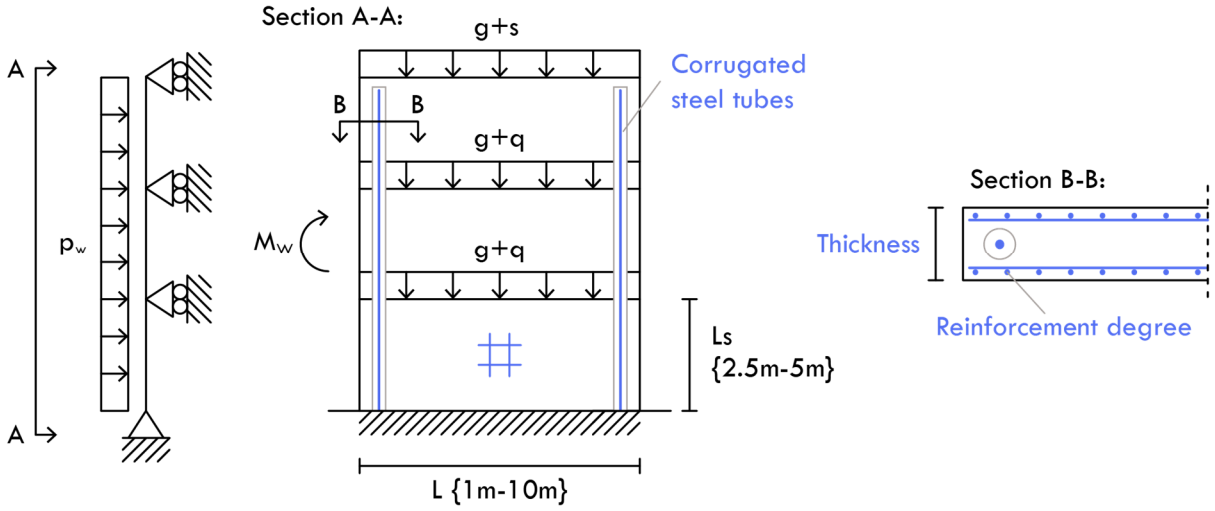


Figure 4.39 – Illustration of the variables for a RC wall section, blue indicates design parameters.

Table 4.10 presents the design parameters and their respective ranges. Brute-force optimization is employed since only the reinforcement ratio fluctuates within a continuous interval. A resolution of 1000 steps is applied for the reinforcement ratio, in addition to the predetermined intervals of the other external parameters. This results in a total of 14000 iterations required for each training sample. Although the brute-force approach is more computationally expensive than the GA approach, the benefit of increased accuracy outweighs the additional computational burden.

The thickness range is established based on industrial standards used by all suppliers. The reinforcement level parameter is remapped from a normalized value to a range dependent on geometry and the boundaries for under and over-reinforced cross-sections. Unlike the RC column module, the reinforcement size is not specified; it is expressed in [mm²/m]. This simplification requires an estimate of the edge distance, which is achieved using a simple linear formula on the conservative side based on the reinforcement level. The final design parameter is defined as a Boolean value determining whether corrugated pipes will be integrated into the wall element. Corrugated pipes serve several purposes in practice. For instance, robustness requirements may dictate their placement to reduce the risk of a progressive collapse. They may also be included to secure the wall against sliding failure. In this scenario, they only increase the wall's capacity to resist overturning. Generalizing the design of corrugated pipes is challenging, as several different factors and parameters must be considered. Therefore, a simplistic approach is used to parameterize the corrugated pipes. Firstly, corrugated pipes can only be used for walls with a minimum thickness of 180mm. Secondly, it is impossible to calculate an exact capacity, as this value depends on several factors, including whether there is sufficient ballast in the foundation or whether the foundation can function as a stabilizing baseplate.

All constructions below the foundation footing are beyond the scope of design cycle one; therefore, a conservative assessment of the capacity is calculated, which increases linearly with the thickness of the wall. The exact cost of corrugated pipes is hard to determine since the cost objective is tied to material prices. Based on internal discussions, a factor four is multiplied by the cost of the reinforcement used in the corrugated pipe. This multiplication accounts for the additional overlapping- and transverse reinforcement needed and the material cost of the corrugated pipe itself. This simplification and multiplier value is deemed acceptable for the level of detail defined for design cycle one and because the algorithm in the training examples used corrugated pipes in 19.6 percent of all cases. If the algorithm used the corrugated pipes every time or never, it would indicate that the cost was calibrated incorrectly. It is also noted that the expected percentage of use was within this range since corrugated pipes can only be used on wall thicknesses of a minimum of 180mm, as previously mentioned.

Table 4.10 - Design parameters for the RC wall evaluation model, with corresponding range.

Design parameters	**Range**
Thickness [mm]	{100,120,150,180,200,240,300}
Reinforcement level [mm²/m]	Depends on geometry
Corrugated steel tube	{true, false}

Table 4.11 outlines the external parameters. The height and load parameters are defined based on the same considerations as those used in the RC column evaluator. It should be noted, however, that higher loads may occur for external walls due to the recess parameters which can result in concentrated loads, as indicated in blue in Table 4.11. The recess ratio is set to a range between 0, which indicates no recess at all, and 80 percent. This range is deemed realistic, as exceeding it would result in insufficient material to absorb the forces acting on the wall element.

Table 4.11 - External parameters for the RC wall evaluation model, with corresponding range. The parameters in blue are only applicable for outer walls.

External parameters	Range
Ls [m]	{2.5-5}
Number of floors	{1:1:12}
Wall length [m]	{1-10}
Live load category	{A, B, C, D, E, F}
Characteristic live load (q) [kN/m]	{0-126} / {0-315}
Characteristic dead load (g) [kN/m]	{0-104} / {0-260}
Characteristic wind load (p_w) [kN/m]	{0,34-12}
Recess ratio (RR)	{0-0.8}

It should be noted that snow load is not included as an independent parameter. This exclusion is because it can be implicitly calculated from the size and category of the live load since the current geometric limitations do not allow for snow accumulation. The wall length is limited to the range between one and ten meters due to practical limitations in production and transportation and static considerations defined in section 4.4.3. Based on an assessment of the weighting between the level of calculation detail and the resulting increased complexity, it is deemed sufficient to examine the walls based on the following three STR/GEO load combinations defined in Eurocode 0 [109]:

- 1: Dominant wind with favorable load
- 2: Dominant wind with unfavorable load
- 3: Dominant live load

Regarding the vertical load, the wall is evaluated using an interaction diagram similar to the column module. Concerning the overturning moment, it is verified that the wall's center point remains with the cross-section of the wall to prevent overturning. There is no need to verify sliding; a potential issue is not dimensioning for the geometry of the wall itself and can

be easily resolved using corrugated pipes. The maximum stress resulting from the overturning moment is also included in the calculations.

As described earlier, a training set consisting of 8000 samples for both the inner and outer walls was created and optimized using a brute-force approach. The relatively large number of samples was necessary because finding a solution may not always be possible based on the randomly generated external parameters. For example, if the length of a wall is short and the wind load is very high, the wall will overturn regardless of its thickness or the use of corrugated pipes. Therefore, creating two more types of surrogate models for each wall type was necessary. First, a classification model was trained to predict whether it is possible to find a solution for the given wall length and external conditions. Then, a regression model was created to predict the geometry and cost. This model was only trained on samples where it was possible to find an optimal solution.

The confusion matrix, shown in Figure 4.40, displays the prediction performance of the classification model. This model was evaluated using a validation subset of 400 samples that the surrogate model had not encountered during training.

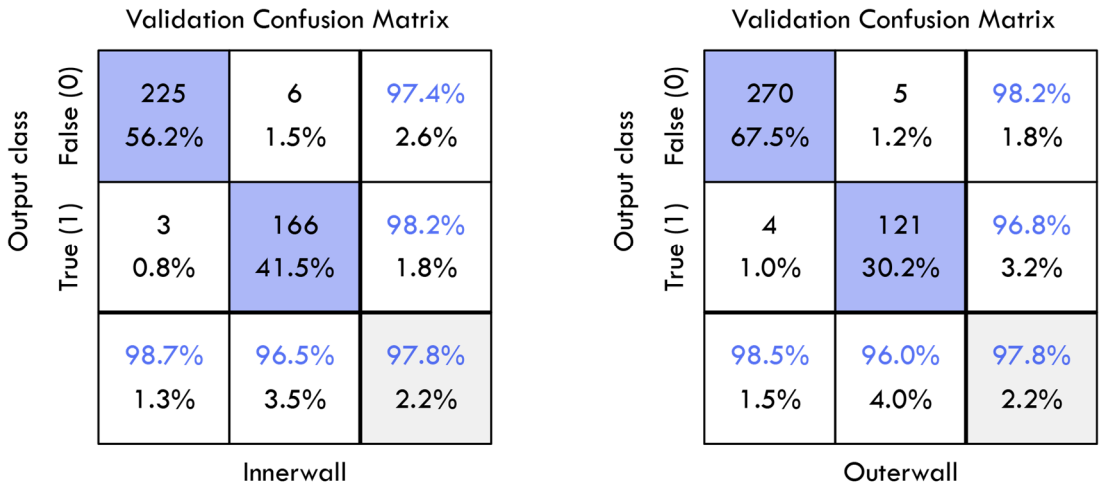


Figure 4.40 – Confusion matrix for the classification models of the innerwall- and outerwall modules. Blue indicates a correct prediction. False designate cases where a solution could not be found, true designate cases where a solution could be found.

Both models demonstrate good performance with few errors. It can be assumed that the errors that occur are close to the decision boundary, meaning that a solution incorrectly classified as valid is likely to be close to a valid solution, and the corresponding geometry will not be significantly incorrect. It should also be noted that fewer valid solutions were found for the outer walls than the inner walls, even though the inner walls have twice the maximum load area. This difference occurs because the outer walls are also affected by the horizontal

wind acting perpendicular to the wall surface, as well as the recess ratio increasing the load on the outer walls.

A valid solution was obtained in 2461 out of the initial 8000 samples for the inner wall regression model. These valid samples were used to train the surrogate regression model, which predicts thickness and cost based on external parameters. The corresponding prediction plots are visualized in Figure 4.41. It is noted that distinct clusters emerge in subfigure (a) due to the actual thickness being defined in specific intervals, as indicated in Table 4.10. While a multiple classification model could have been used, a regression model can also predict specific intervals. Based on the prediction plots and error metrics, it can be concluded that both models exhibit good prediction properties.

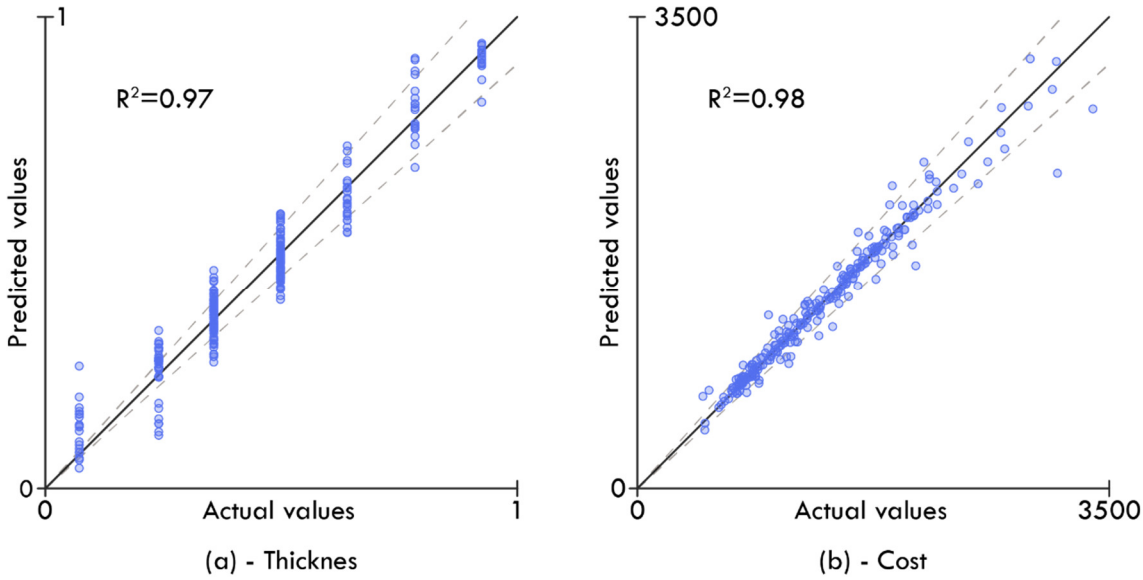


Figure 4.41 – Illustration of the prediction plots for the RC inner wall evaluator, (a) prediction of the thickness parameter, (b) prediction of the cost output in DKK

For the outer wall module, 2338 valid samples were generated from the initial batch of 8000 samples. As with the inner wall, two surrogate models were constructed to predict the thickness and cost of a given wall based on external parameters. The corresponding prediction plots are visualized in Figure 4.41 and Figure 4.42. It can be observed that both models exhibit similar behavior.

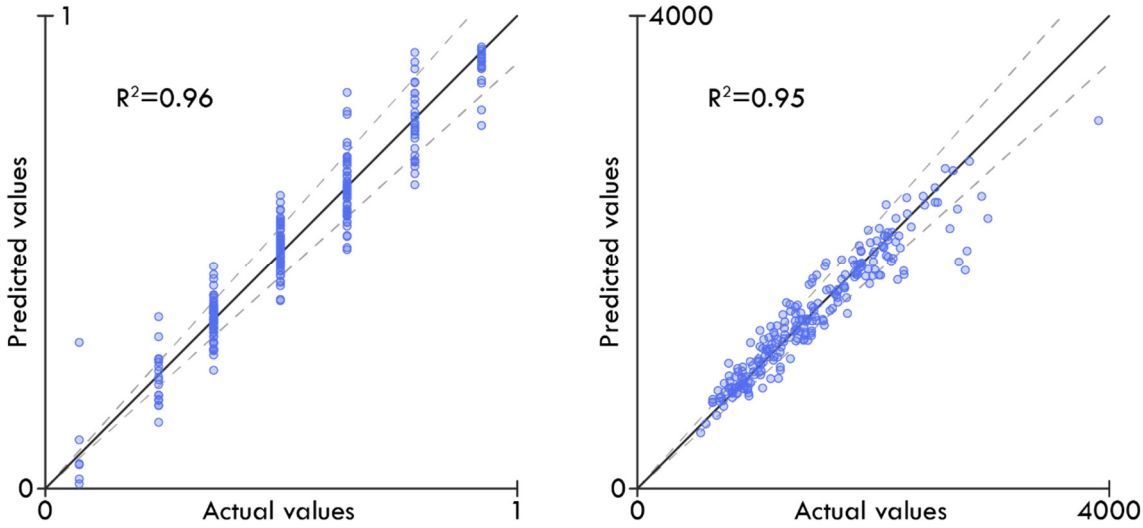


Figure 4.42 – Illustration of the prediction plots for the RC outer wall evaluator, (a) prediction of the thickness parameter, (b) prediction of the cost output in DKK/m.

However, the surrogate models for the outer walls are slightly less accurate, which can likely be attributed to fewer training samples, and because the outer wall modules contain two additional external parameters, increasing the solution space. Nevertheless, the models still exhibit excellent prediction capabilities.

4.7 Optimization phase

The general optimization framework utilized in the design tool is illustrated in Figure 4.43. The process can be described as a surrogate-assisted optimization using a modified genetic algorithm that combines real-value and permutation encoding.

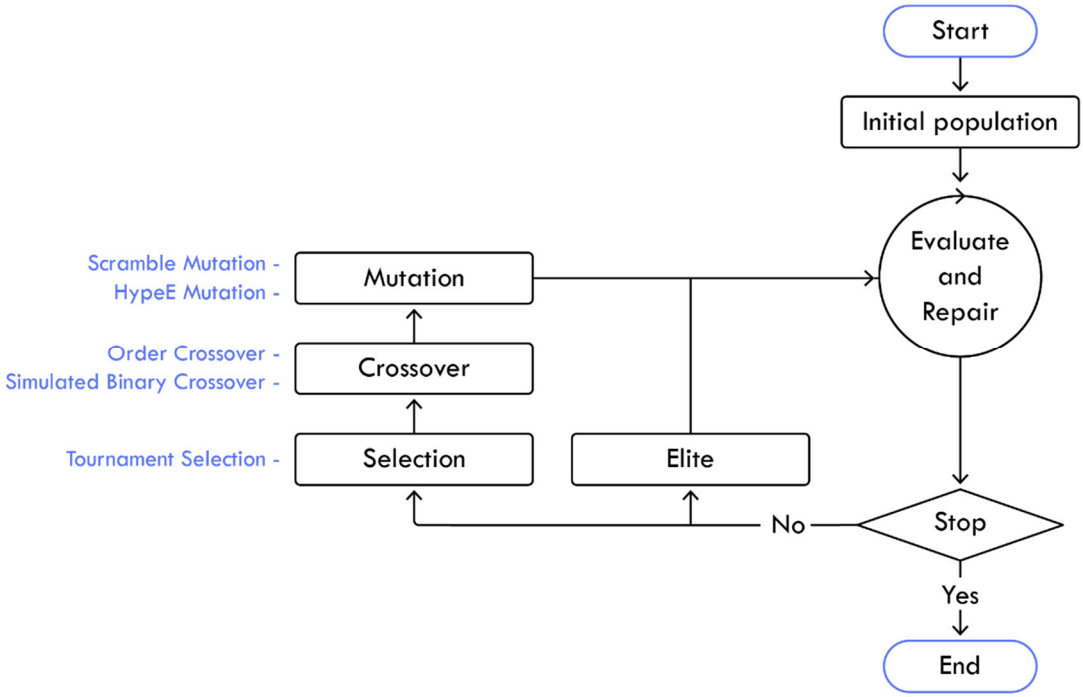


Figure 4.43 – Illustration of the general optimization procedure.

The optimization procedure illustrated in Figure 4.43 can be utilized to create AI-generated or hybrid solutions, as detailed in section 4.3.4.

Tournament selection is utilized as the selection mechanism, and the solutions' parameters are divided into their real-value and permutation encoding components. Simulated Binary Crossover and HypeE Mutation [60] are used for the real-value parameters, while Order Crossover and Scramble Mutation are applied to the permutation parameters.

Some of the adjusted solutions derived from the GA operators will be flawed, as the stabilizing walls may be too short to withstand the horizontal forces. Therefore, it may be advantageous to integrate a repair algorithm that operates on the chromosome of different solutions and increase the parameters defining the shear wall's length. Incorporating user-induced knowledge into the optimization process has shown great potential in reducing convergence time in previous research, such as [28] [29]. The "Evaluate and Repair" module in Figure 4.43 represents this part of the algorithm, and the entire process is further elaborated in Figure 4.44.

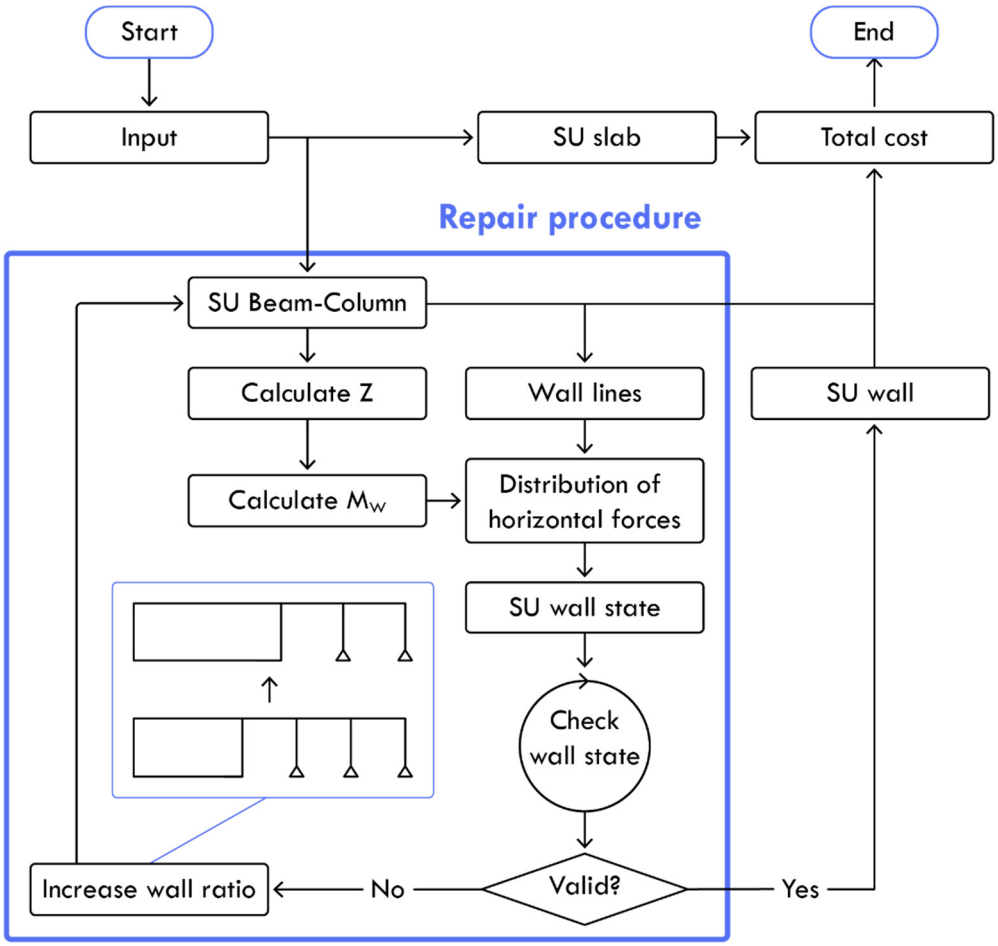


Figure 4.44 – Illustration of the evaluation and repair procedure.

The process commences by generating the slabs and beam-column systems using the surrogate models. The story height is determined by the geometry of the beam-column system, which dictates the height of the shear walls. The total building height is then computed, and the resulting wind force affecting the building is updated accordingly. Next, the horizontal forces are distributed among the shear walls, and the surrogate model that determines whether a wall solution can be found is used on every wall line. The corresponding wall lengths are extended if the SU model predicts one or more wall lines to fail. The process is then repeated, and updated beam-column systems are created, which may result in a new building height. The loads are updated and redistributed as before, and all wall lines are rechecked. As a result of the updated stiffness, the forces are now distributed differently. Therefore, it is necessary to verify that all walls can handle the loads affecting them again. This iterative process continues until all walls are valid, or a wall cannot be extended further. After this process, the geometry of the walls can be defined, and the total cost of the building structure can be determined as the sum of the slabs, beams, columns, and walls.

It is important to note that the repair procedure does not optimize in the opposite direction, i.e., by removing unnecessary shear walls or changing the global geometry. Firstly, this process occurs indirectly through the optimization process in the GA. Additionally, it is essential that the repair algorithm does not undermine the evolutionary principles in the GA but only makes minor adjustments to speed up the convergence process.

If a hybrid solution is required, the user may dictate specific parameters. However, this does not significantly alter the overall optimization process. An algorithm ensures that user-defined parameters remain untouched by the crossover and mutation mechanisms. The algorithm reorders permutation parameters for any operated-upon chromosomes to ensure that user-defined parameters are first located in the chromosome.

4.8 Summary

Chapter 4 presents the final design tool and elaborates on implementing the conceptual framework in Chapter 3. The design tool, also referred to as a design methodology, was built using four core principles: optimization, interactivity, dissemination, and automation. The chapter also explains how the design tool was envisioned in a larger context of nesting design loops, each with an increasingly associated Level of Detail (LOD). In this project, the design loop was intended for the outermost design loop, referred to as design loop one.

Various relevant programming languages were explored, and their advantages and disadvantages were discussed. Based on a balance between user-friendliness and computing speed, C# was chosen for this project, while Rhino was selected as the software platform for the design tool. Rhino provides access to the RhinoCommon library and offers a 3D engine to visualize the structural layouts of the building.

The design tool's general framework was outlined and divided into two main phases: initialization and optimization. The chapter presents the design objectives of CPM2 and ACS and the load modules. Additionally, a novel method, named the Adjacent Polygon (APoly) representation, was introduced, which can be used to create diverse yet logical structural layout suggestions. The evaluation modules use surrogate and hierarchical surrogate modeling approaches to achieve fast approximated responses. The chapter also presents the accuracy of these models using prediction plots and the correlation value as the error metric.

Lastly, the optimization phase is presented with a focus on how user-induced knowledge is incorporated into the optimization framework. This incorporation is achieved using a repair algorithm that operates on the chromosomes of each individual to increase the number of valid solutions in each generation.

Chapter 5 – Validation studies

"*The form of an object is a diagram of forces.*"

– D'Archy [110]

When utilizing evolutionary optimization algorithms, it is obvious to draw parallels to nature. The structural framework of all living organisms, animal or plant-based, has adapted and optimized over time in response to the environment and physical forces affecting them. For instance, elephants have evolved to become the largest land animal in existence today because their size provided them immunity from local predators. As their size increased, so did their bones, which needed to withstand the gravitational forces acting upon their large mass [65]. Nevertheless, an elephant's bone is only as thick as they need to be, as nature creates the most efficient structure with the least possible amount of material. As Darwin stated in *On the Origin of Species*: "*for it will profit the individual not to have its nutriment wasted on building up a useless structure*" [61].

Nature's structural optimization is also evident when observing the structural shape of trees. Trees grow tall and spread their branches to maximize the surface area of leaves, thereby increasing the amount of photosynthesis they can produce. However, this growth pattern comes at a cost, as trees are subjected to the physical constraints imposed by wind and gravitational loads. Consequently, the branches of trees are more robust and thicker at the base of the trunk, where the sectional forces are the largest. The size of the branch then progressively becomes thinner towards its end, where the forces are at their minimum.

This realization that physical laws and principles of evolution govern patterns and shapes in nature was also evident to Scottish biologist and mathematician D'Archy Wentworth Thompson, who stated, "*The form of an object is a diagram of forces*." This statement implies that it is possible to understand why a natural object is formed as it is by studying the environment to which it has adapted.

This approach is also highly relevant to the examinations carried out in Chapter 5. Assuming the algorithm has generated optimal results that are adapted to the specified constraints and forces acting on the building, it should be possible to derive why the algorithm produced a particular structural configuration. These validation studies can then be used to confirm that the algorithm offers rational and efficient solutions. The studies are categorized into two sections: a parameter sensitivity study of the local modules in section 5.1 and the evaluation of the design tool on diverse building plans and settings in section 5.2.

5.1 Validation of the structural modules

In Chapter 4, the prediction efficiency of the structural modules was demonstrated. Section 5.1 conducts a parameter sensitivity study to observe how changes in parameters affect the module's response. This study is visualized using sensitivity plots, demonstrating how sensitive a model is to change in a given parameter. This examination also offers an opportunity to assess whether the response is logical, indicating that the surrogate models behave as expected.

The x-axis in the sensitivity plot represents the specified free parameter for which the response is being examined. The parameter change occurs through high-resolution increments of 1000 steps, translating into a continuous response graph. The y-axis represents the response, which is the objective that the given SU model is designed to predict.

It is not possible to visualize the interrelated spectrum of all parameters. Instead, the change in response is demonstrated for one parameter at a time while the remaining parameters are fixed. The value of each external parameter is visualized in their normalized domain under each sensitivity plot, except for the live load category, which is set to zero unless another value is specified. The value of zero represents the typical residential load. It should also be noted that the dead load and live load parameters are combined into one parameter denoted as "vertical load" for the sake of simplicity.

A series of the most relevant sensitivity studies are selected and presented in the following subsections to help evaluate the validity of the structural modules used in the final design tool.

5.1.1 Wall module

Given the similarity in structure and behavior between the inner and outer wall modules, it is determined that sensitivity studies will be conducted solely on the inner wall module. This decision is made because the conclusions derived from the sensitivity studies on the inner wall module will also apply to the outer wall module.

A scenario is examined with a relatively high overturning moment, while the vertical load is defined as the free parameter. It should be noted that the length of the wall is set to its maximum value to ensure that a solution can be found when the vertical load is low. In Figure 5.1, an initial high CPM and thickness value is observed, which quickly decreases before experiencing a slight increase. This behavior can be explained by the fact that the overturning moment dominates when the vertical load is low. The wall increases its self-weight and utilizes corrugated tubes, which increases the cost, to compensate. When the vertical load on the wall is increased, the wall can more easily withstand the overturning moment, resulting in the use

of less material. However, when the vertical load becomes too high, the wall requires more material to withstand the load, causing the cost to increase again.

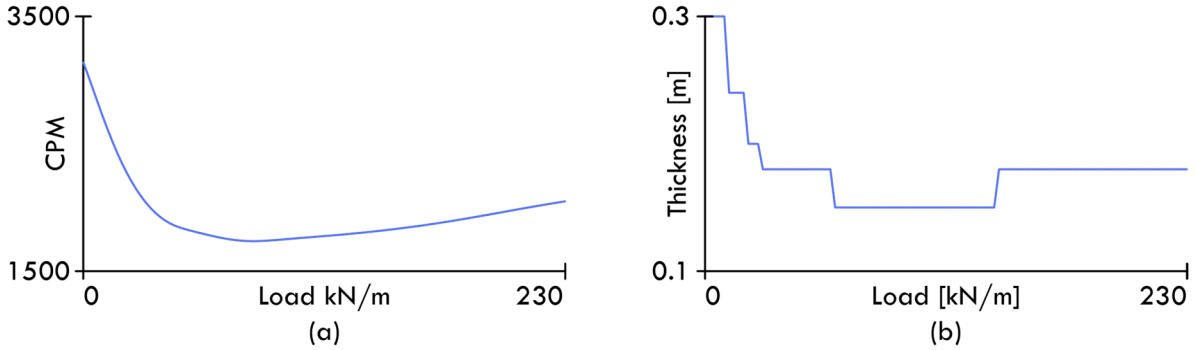


Figure 5.1 – Sensitivity plot for the RC wall module in relation to the load value, with (a) illustrating the response of the cost per meter (DKK/m) objective, and (b) illustrating the response of the wall thickness objective.

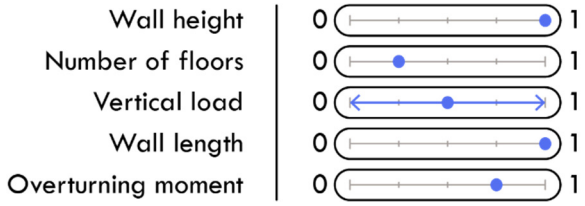


Figure 5.2 – External parameter settings that were used for the sensitivity plot in Figure 5.1

A scenario is examined in which the number of floors represents the free variable. The length of the wall is maximized to minimize the influence of the overturning moment. In Figure 5.3 (a) and (b), a steep increase is observed initially, followed by a more gradual increase throughout the rest of the range. This change is likely due to the live load reduction factor, which activates after a couple of floors, which reduces the percentwise increase of the live load and is reflected in the material cost.

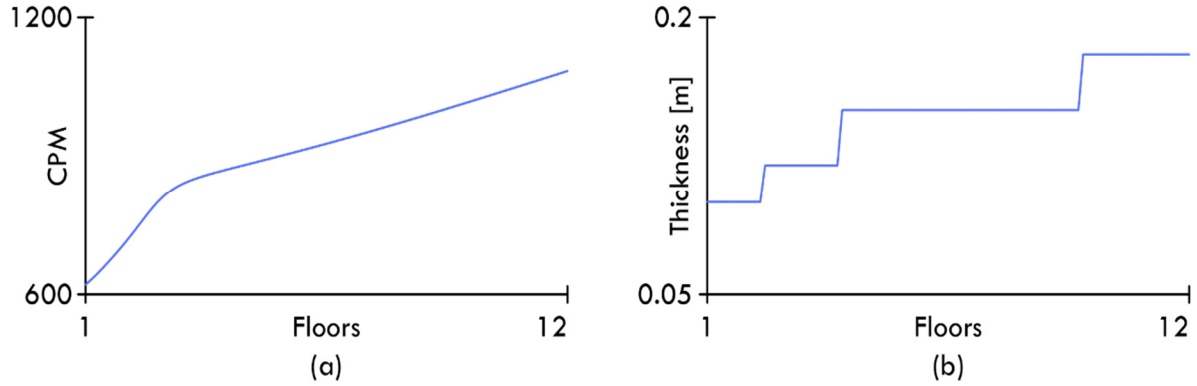


Figure 5.3 – Sensitivity plot for the RC wall module in relation to the number of floors, with (a) illustrating the response of the cost per meter (DKK/m) objective, and (b) illustrating the response of the wall thickness objective.

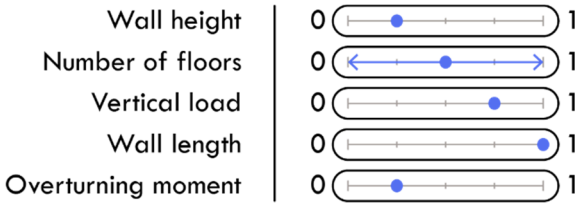


Figure 5.4 – External parameter settings that were used for the sensitivity plot in Figure 5.3

5.1.2 Beam module

Figure 5.5 illustrates the sensitivity plot in relation to the beam length parameter across the entire range. The vertical load parameter is fixed at the 0.75 percentile to amplify the impact of the beam length variation. An accelerating increase in the CPM and beam height response can be observed. This graph shape was expected because the beam length is squared in the bending moment calculation.

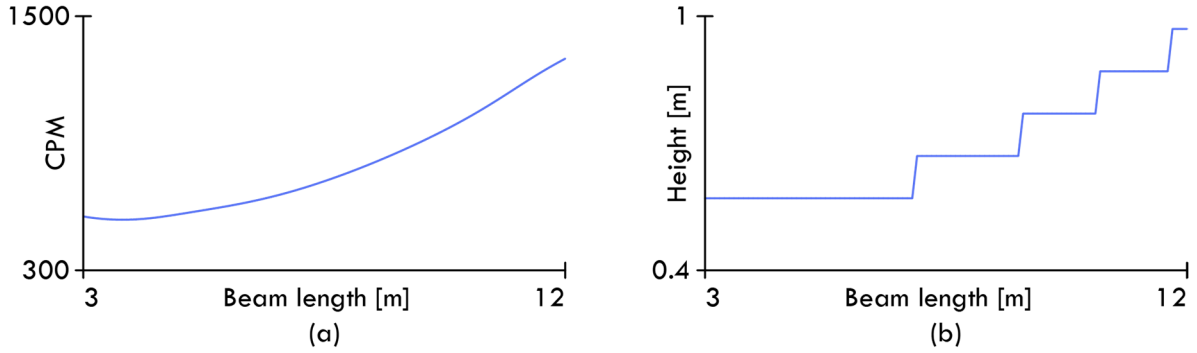


Figure 5.5 – Sensitivity plot for the RC beam module in relation to the beam length, with (a) illustrating the response of the cost per meter (DKK/m) objective, and (b) illustrating the response of the beam height objective.

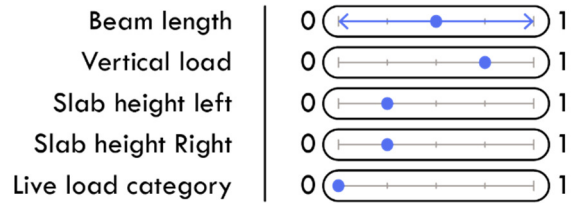


Figure 5.6 – External parameter settings that were used for the sensitivity plot in Figure 5.5

Figure 5.7 illustrates the sensitivity plot for a scenario where the vertical load is set as the free parameter. The beam length is fixed at the 0.75 percentile, and the live load category is set to category F, which corresponds to traffic load, to amplify the impact of the variation. As expected, a relatively linear increase in cost and beam height can be observed in Figure 5.7 (a) and (b).

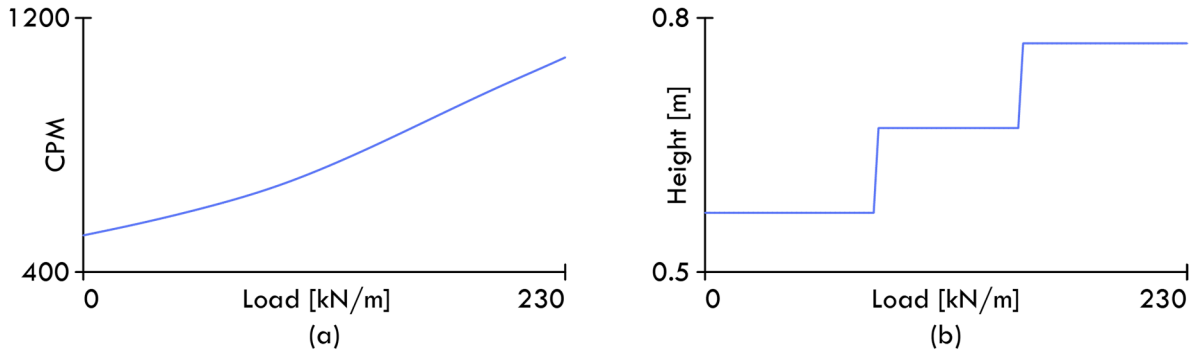


Figure 5.7 – Sensitivity plot for the RC beam module in relation to the load, with (a) illustrating the response of the cost per meter (DKK/m) objective, and (b) illustrating the response of the beam height objective.

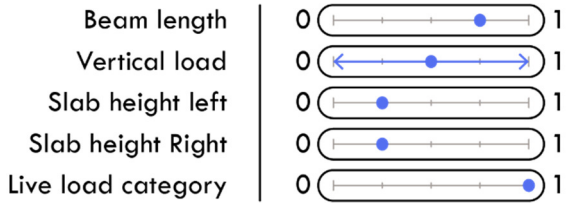


Figure 5.8 – External parameter settings that were used for the sensitivity plot in Figure 5.7

An interesting effect is demonstrated in Figure 5.9, where the height slab on the left side is fixed at its maximum value, and the parameter for the right slab height is free. A decrease in cost is observed as the slab height on the right-side increases. This effect occurs because the beam ledge's height decreases to accommodate the increasing slab height, and because no other external parameters change, no increase in the total height is necessary. Consequently, the beam's cross-sectional area decreases, which is manifested in Figure 5.9 (a).

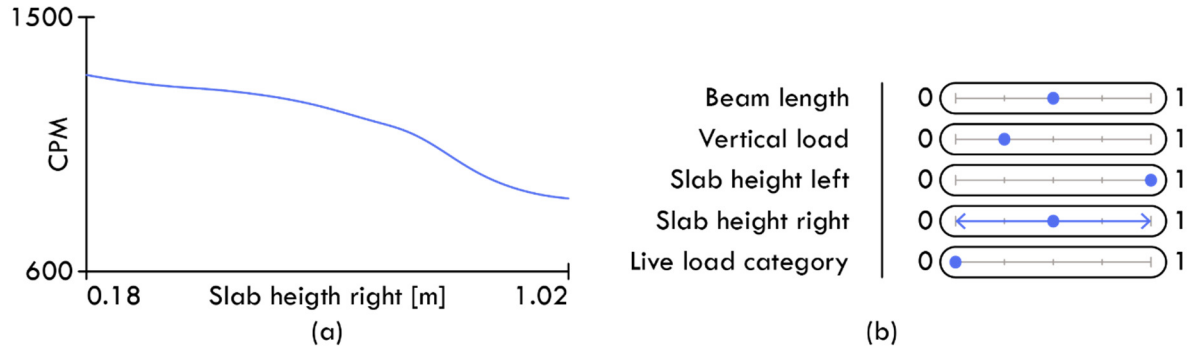


Figure 5.9 – (a) Sensitivity plot for the RC beam in relation to the right slab height and the cost per meter (DKK/m) objective, and (b) illustrating the corresponding external parameters.

5.1.3 Column module

A scenario is examined where the beam length is defined as the free parameter, resulting in a linear increase of the normal force acting on the column. It is noted that the beam length in this context, can also be understood as the column spacing. As expected, the cost and column width increase as the beam length and load area increase, as the column requires more material to withstand the increasing force affecting it. The cost is observed to be moderately accelerating in Figure 5.10 (a), likely due to the second-order effect being triggered and because the weight of the above columns' mass increases, as shown in Figure 5.10 (b).

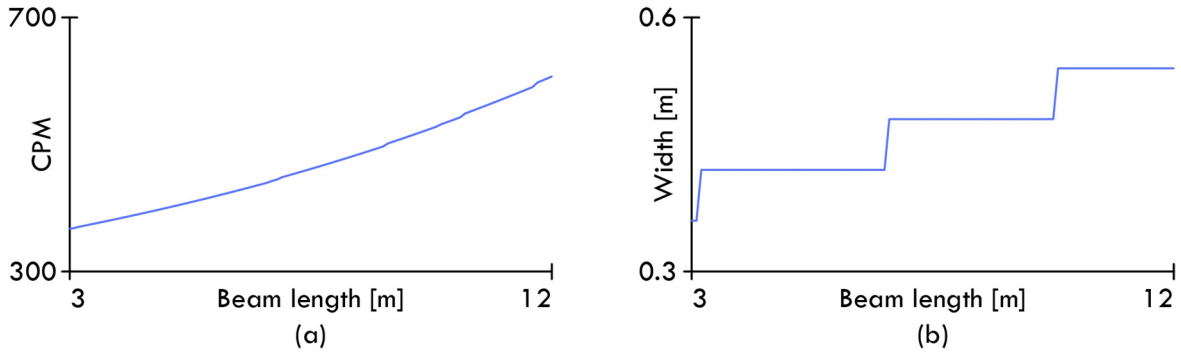


Figure 5.10 – Sensitivity plot for the RC column module in relation to the beam length, with (a) illustrating the response of the cost per meter (DKK/m) objective, and (b) illustrating the response of column width objective.

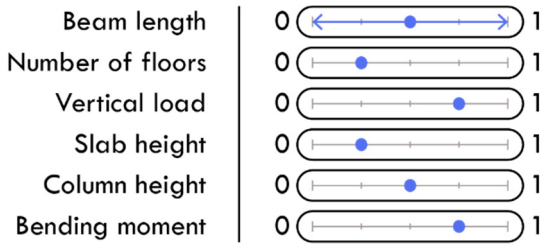


Figure 5.11 – External parameter settings that were used for the sensitivity plot in Figure 5.10

Figure 5.12 illustrates an analysis where the number of floors is defined as the free parameter. The SU model demonstrates a pattern of cost increases occurring in intervals corresponding to an increase in the number of floors. There are 12 plateaus visible in Figure 5.12 (a) corresponding to each available floor number, and the distance between them are roughly equal, which was expected given the parametric definition.

Figure 5.12 (b) shows a relatively linear increasing trend in the column width response. However, the slope of the increase flattens slightly, which can be attributed to the scaling principle of the column, where the cross-section area increases by a more significant proportion with each width increment.

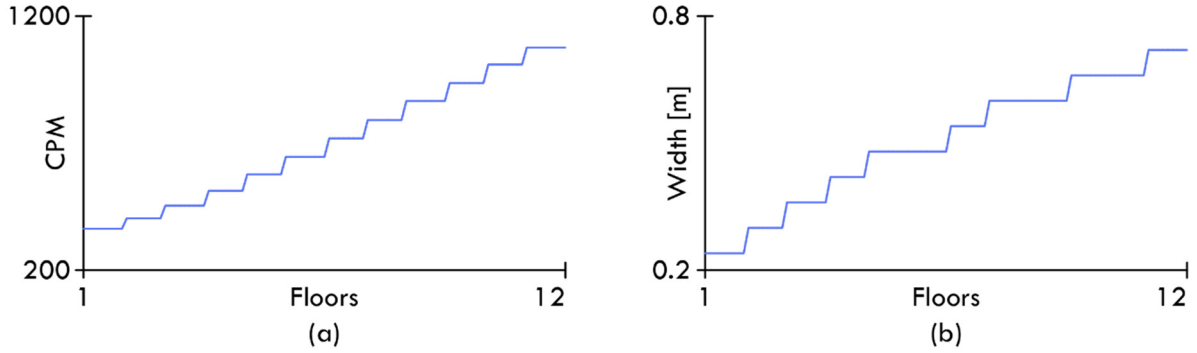


Figure 5.12 – Sensitivity plot for the RC column module in relation to the number of floors, with (a) illustrating the response of the cost per meter (DKK/m) objective, and (b) illustrating the response of column width objective.

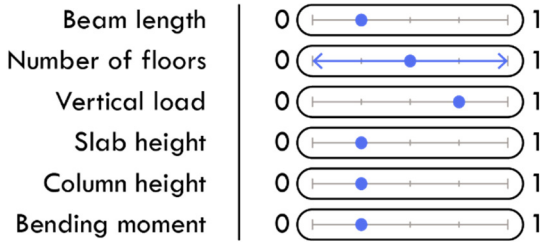


Figure 5.13 – External parameter settings that were used for the sensitivity plot in Figure 5.12

5.1.4 Beam-column line module

The methodology for defining the optimal distance between columns in the beam-column SU model has been outlined in section 4.6.4. This section presents a series of analyses where two sensitivity graphs are plotted using the beam length as the free parameter and two different values of a relevant external parameter to illustrate their impact on the optimal column spacing. It is noted that all unspecified external parameters are set to their 0.25 percentile value in the normalized domain, except for the live load category, which is defined as category A, corresponding to a standard residential live load.

An analysis is conducted to evaluate the impact of the CPM parameter, which was designed to artificially increase the cost of columns, thereby encouraging the algorithm to reduce the number of columns. The black curve in Figure 5.14 represents an unaltered column price, while the blue curve illustrates a column price multiplied by a factor of three. It is evident that the curve's minimum point has shifted to the right, increasing the distance between the columns as expected. It is emphasized that the adjusted column price is only utilized to determine the column spacing, while the actual price of the column is used in the overall algorithm of the design tool.

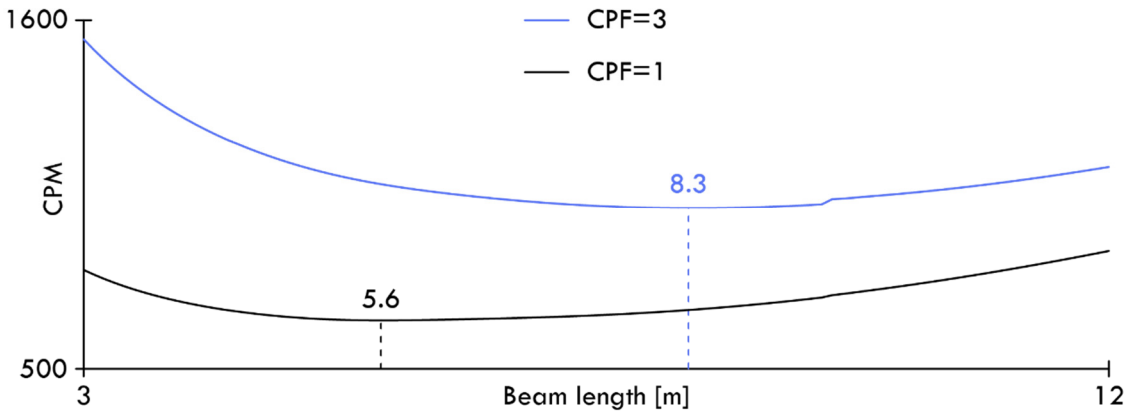


Figure 5.14 – Sensitivity plots for the RC beam-column line module in relation to two different values of the Column Price Factor (CPF).

Figure 5.15 illustrates the difference in using the minimum and maximum effective column length, represented by the black and blue curves, respectively. It can be observed that increasing the effective length results in a higher column spacing. This effect was expected, as an increased column length leads to a greater self-weight from the columns above and a more significant influence from the second-order effects. All these aspects increase the load on the columns and consequently increase the price per column, incentivizing the algorithm to utilize fewer columns.

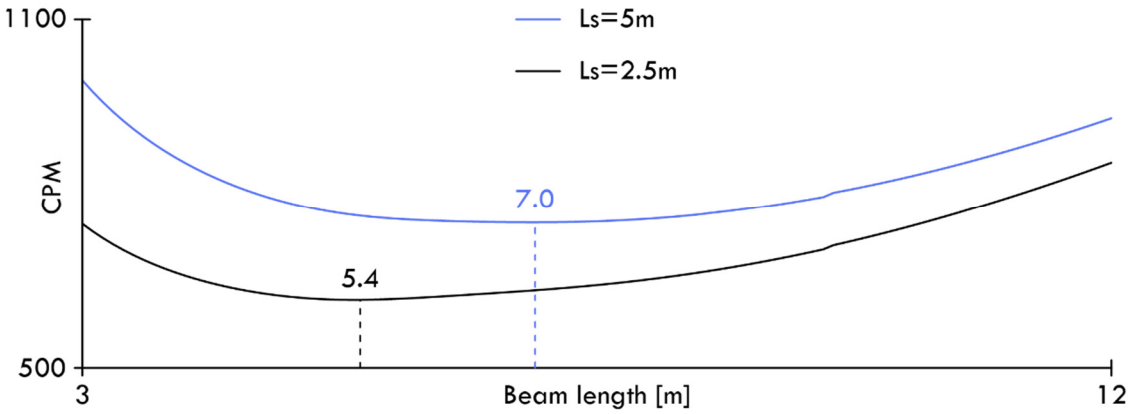


Figure 5.15 – Sensitivity plots for the RC beam-column line module in relation to two different values of the effective column length (Ls)

In Figure 5.16, the black and blue curves visualize the effect of utilizing the minimum and maximum line load, respectively. As expected, when the line load consists only of the self-weight of the beam, the algorithm will choose a larger spacing distance between the columns. However, when the line load is maximized, the column spacing is almost halved to reduce the bending moment in the beam and decrease the load on the columns.

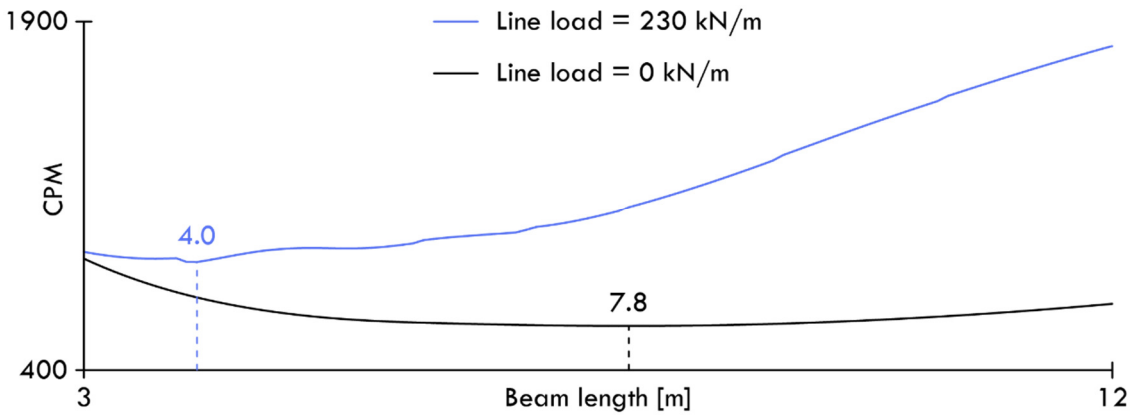


Figure 5.16 – Sensitivity plots for the RC beam-column line module in relation to the minimum and maximum line load value.

Figure 5.17 illustrates the difference between using one or twelve floors, as visualized by the black and blue curves, respectively. It can be observed that the overall shape of the curves is similar, as the parameter does not affect the prerequisites regarding the beam, but only the axial force which affects the columns. Therefore, the relative cost ratio between columns and beams only changes slightly, reflected in the small reduction of the column spacing, to decrease the accumulated load on the columns.

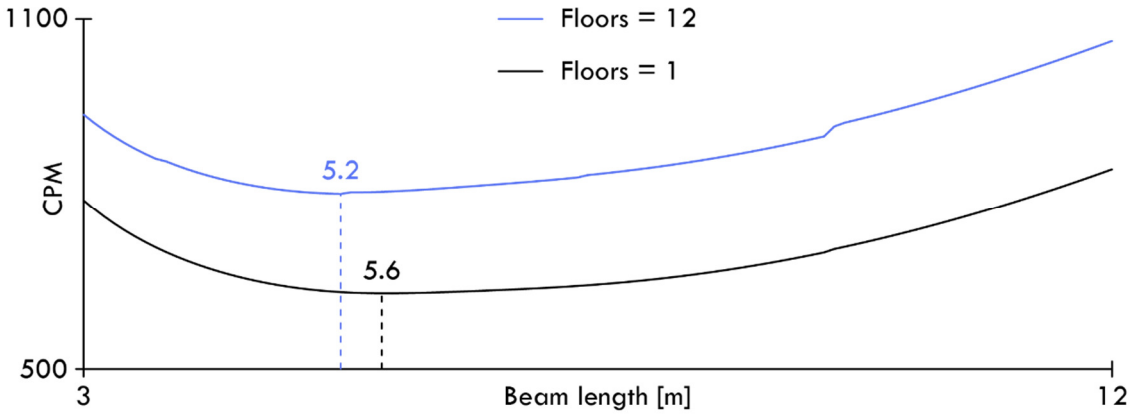


Figure 5.17 – Sensitivity plots for the RC beam-column line module in relation to the minimum and maximum number of floors.

5.2 Validation of the design tool

The analysis presented in section 5.1 should be considered as local investigations of the structural modules. This section employs a global approach, wherein the design methodology as a whole is tested on various building planes. The general design methodology is illustrated in Figure 4.43 and Figure 4.44, and the solutions generated consider the interconnected relationships between all the structural modules with respect to the specified design objective.

These examinations aimed to investigate whether the algorithm converges towards optimized solutions. The design objective is the minimization of material cost; therefore, optimized solutions should be characterized by an efficient structural layout that still complies with the defined physical and practical constraints. A series of specific scenarios are examined to analyze why the algorithm converged toward a particular solution, aiming to demonstrate that the design tool generates optimized and logical solutions.

The base settings used in the design tool are listed in Table 5.1, while those utilized in the GA are listed in Table 5.2. If an analysis deviates from the main settings, it will be specified in the corresponding subsection.

Table 5.1 – Base settings for the design tool

Design tool setting	Value
Live load size	1.5 [kN/m²]
Live load category	A1 – Residential area
Variable dead load	1 [kN/m²]
Number of floors	4
Installations height	0.4 [m]
Ground clearance height	2.5 [m]
Terrain category	III
Orography factor	1
Outer wall recess ratio	0.5
CPF	1

Table 5.2 – Base settings for the GA

GA setting	Value
Generations	100
Population	200
Elite ratio	1:10
Crossover rate	0.70
Mutation probability	0.05
Mutation rate	0.35

5.2.1 Different CPF on a simple building plane

As shown in Figure 5.18 the design tool has been applied to a building plan with an L-shaped layout. This layout was selected for the initial test due to its simplistic shape, which reduces the solution space. This simplification makes it easier to analyze the solution from a static and economic perspective before applying the design tool to more complex building shapes. The L-shaped layout is transformed into five different AP shapes, allowing for a reduction in population size to 100.

Two investigations are conducted, one using standard settings and one using a CPF value of three, to examine how a change in CPF values affects the final results of the algorithm.

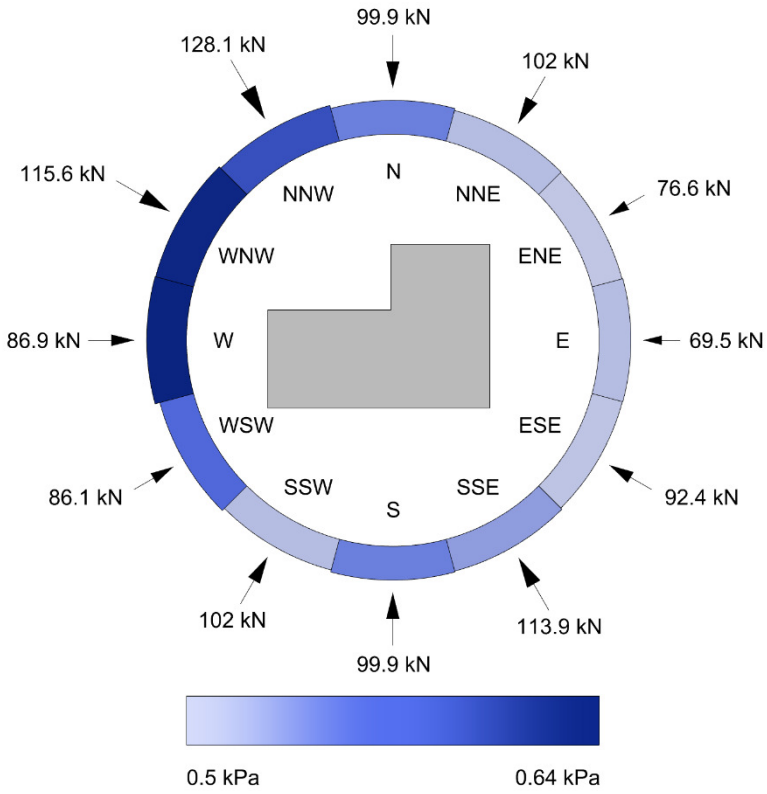


Figure 5.18 – Illustration of the wind resultants and corresponding wind pressure values for 12 directions.

Figure 5.19 (a) and (b) demonstrate a rapid convergence towards an optimum with a minimum CPM2 value of 393.5 and 410.6, respectively, corresponding to a 4.3% increase. As expected, the algorithm also reduces the number of columns from six to four, as illustrated in Figure 4.20.

As illustrated in Figure 5.22, the algorithm has chosen to insert a wall element, even though it is not required from a static point of view regarding the overturning moment. It is assumed that this decision is due to the algorithm's incentive to minimize the number of columns used. The algorithm can avoid placing an additional intermediate supporting column by inserting

the wall. The choice of the slab is not affected by the different CPF values, as both tests employ the same type of hollow core slab of EX18-5L9.3.

From a general static and geometric perspective, it is noticeable that the algorithm has chosen to halve the span using beam-column lines. While it was statically possible for the algorithm to employ TT-slabs, this solution was economically inferior, despite the potential reduction in beam-column lines. This potential reduction was not enough to compensate for the increased cost per square meter of the slab and the increased height of the exterior walls, as the minimum height must always be maintained.

Based on these observations, it can be concluded that the design tool has generated solutions that make sense from both an economic perspective and that increasing the CPF value will reduce the number of applied columns. As a result, the design tool can be tested on more complex building shapes.

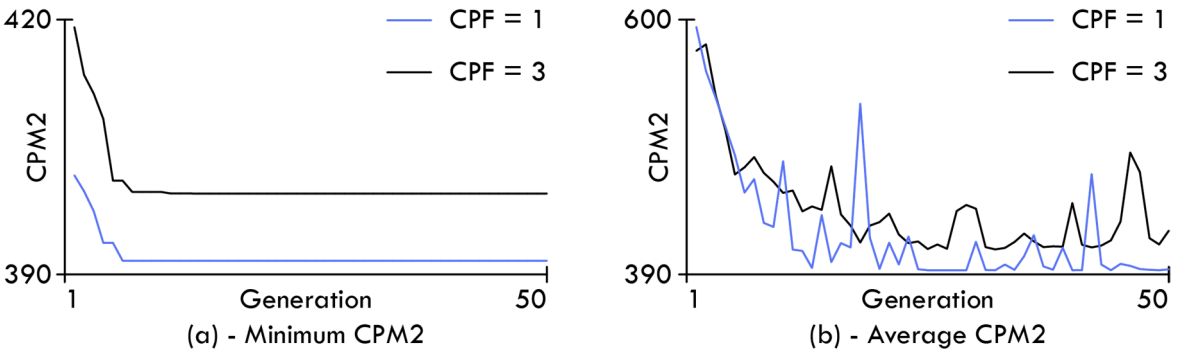


Figure 5.19 – Convergence plot for both the minimum CPM2 (a) and the average CPM2 (b) values.

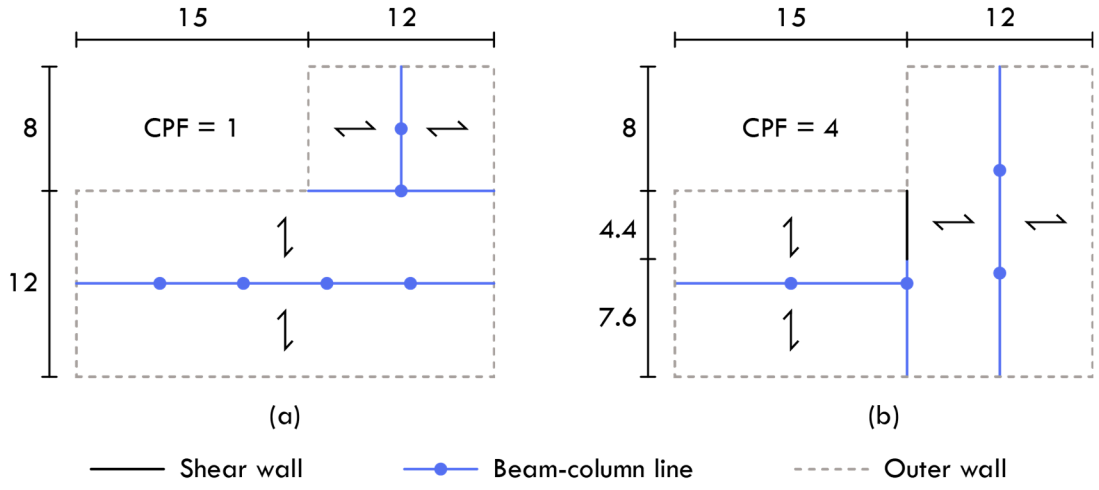


Figure 5.20 – Plan view of the structural layout, with (a) representing CPF =1 and (b) representing CPF=3.

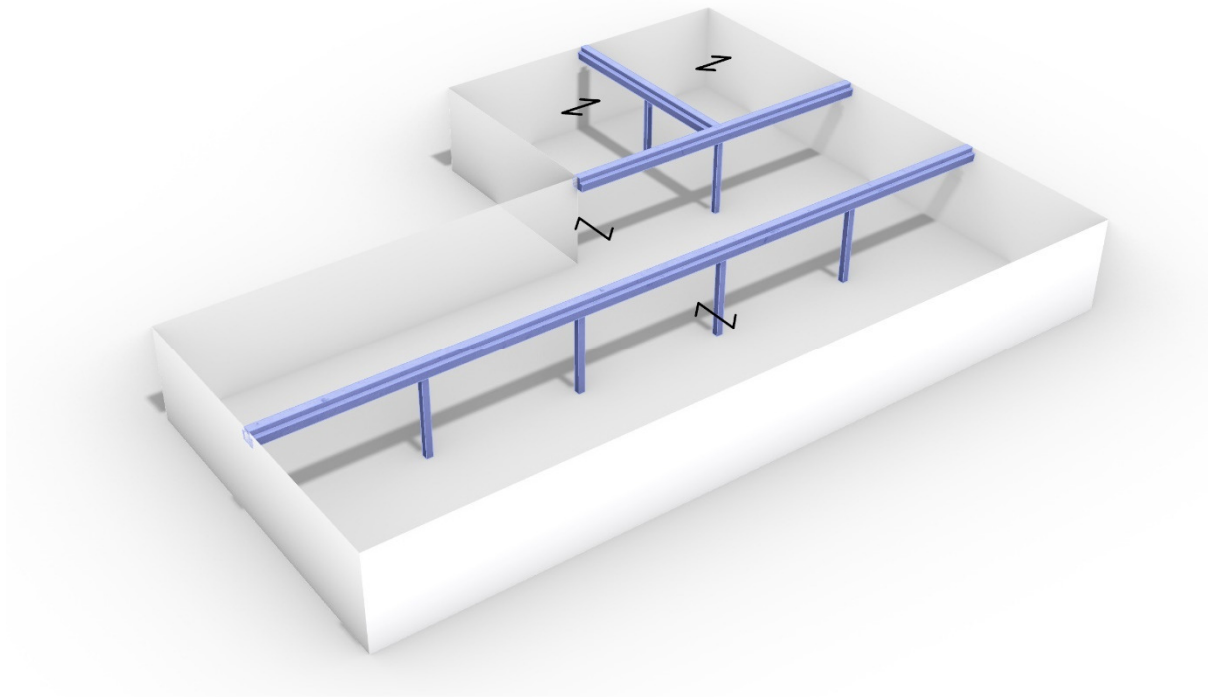


Figure 5.21 – 3d visualization of final solution using CPF=1, CPM2 objective = 393.5.

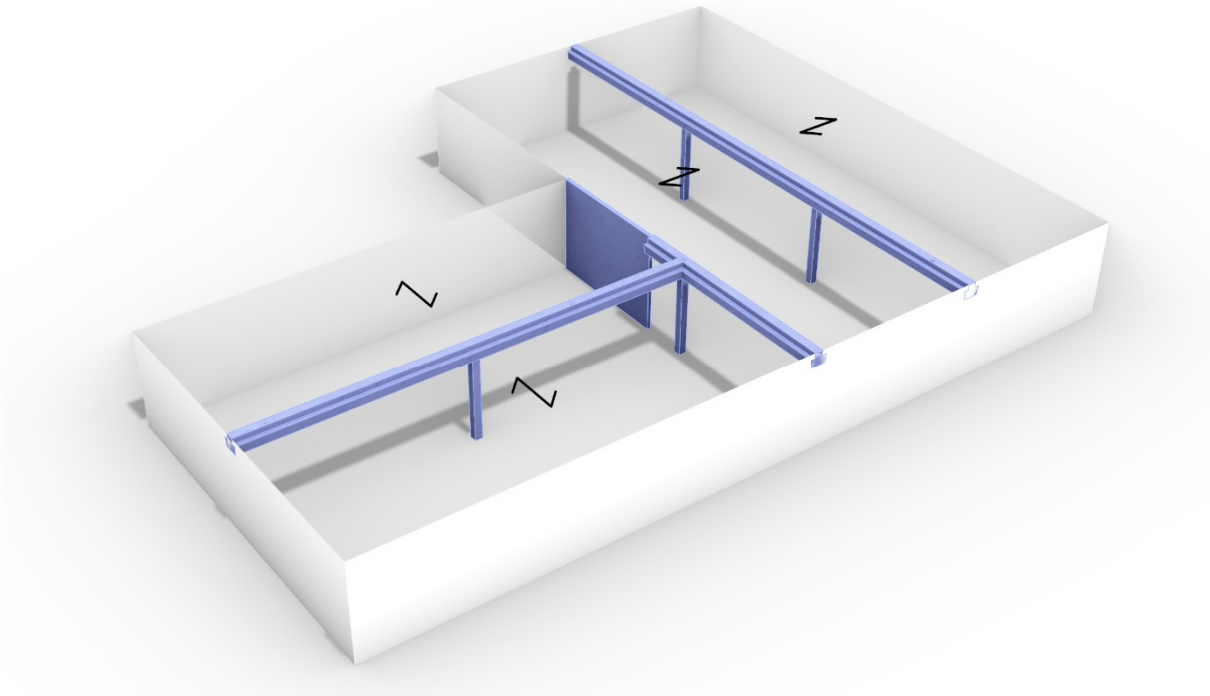


Figure 5.22 – 3d visualization of final solution using CPF=3. CPM2 objective = 410.6.

5.2.2 Multiple floors

This analysis explores the effects of varying floor counts. As additional floors are added, vertical and horizontal loads increase, resulting in an anticipated increase in the use of shear walls. Additionally, cross-sectional dimensions are expected to increase to accommodate the more significant vertical load affecting the structural elements. The analysis employs standard settings as input. Figure 5.29 (a) illustrates the geometric shape and dimensions of the building plan used in the analysis, while Figure 5.23 shows the critical wind directions. Although the wind direction is only indicated for four floors, the relationship between the wind results remains constant for solutions with eight and twelve floors.

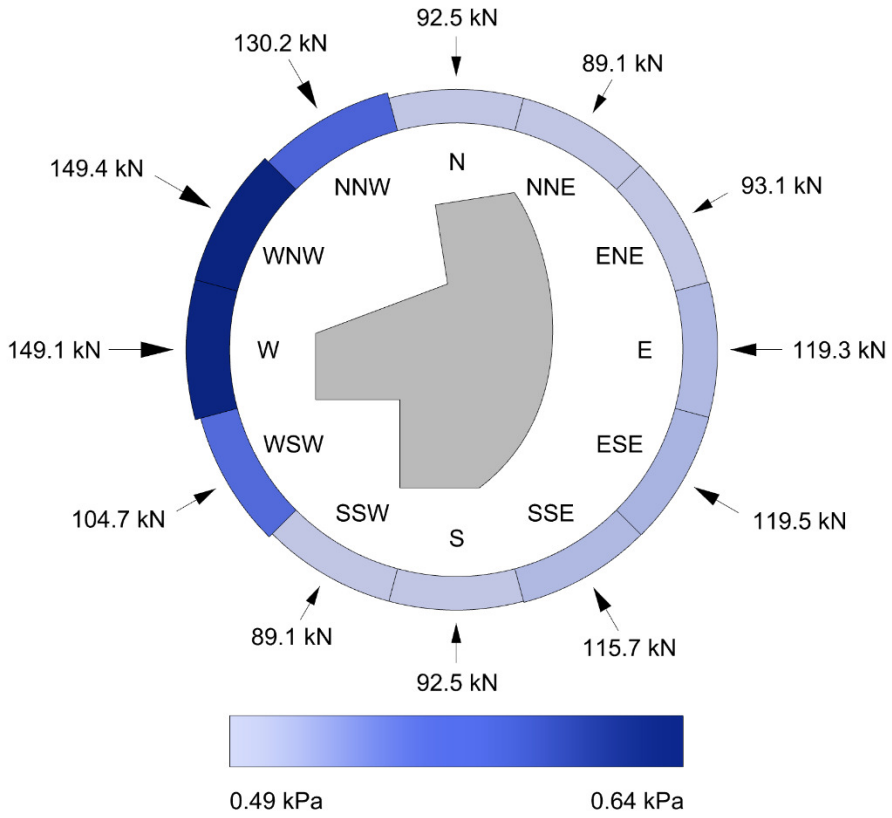


Figure 5.23 – Illustration of the wind resultants and corresponding wind pressure values for 12 directions.

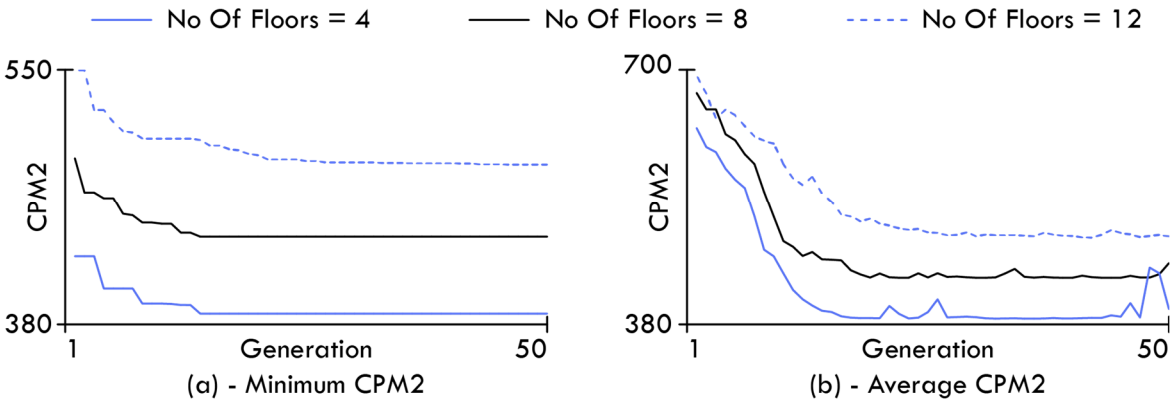


Figure 5.24 – Convergence plot for both the minimum CPM2 (a) and the average CPM2 (b) values.

Figure 5.24 (a) demonstrates a convergence towards an optimum, with a minimum CPM2 value of 387.2, 438.4, and 488.7 for the solutions with four, eight, and twelve floors, respectively. The convergence graphs show a relatively similar shape, although it takes slightly longer for the solution with twelve floors to converge. This difference can be attributed to the increased horizontal forces and vertical loads, which makes it more difficult for the algorithm to find valid solutions.

The increase in CPM2 values was expected, as the structure requires more material to support the additional load. Nonetheless, the percentage increase is relatively linear, indicating that the cost of the slab still dominates the structure's overall cost, as demonstrated in Figure 5.25. However, it is noticed that this proportion decreases as the number of floors rises since the structure requires more material in the wall sections to withstand the affecting forces.

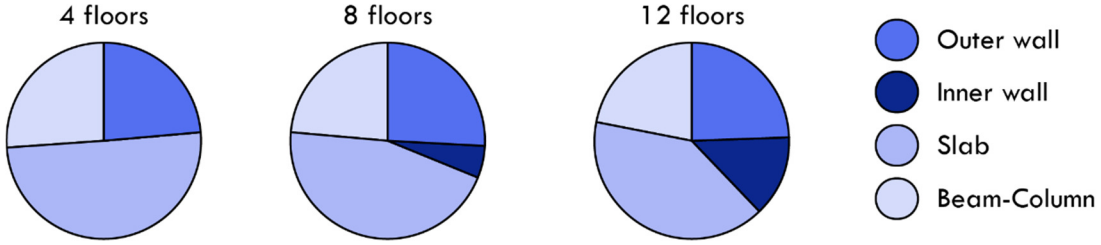


Figure 5.25 – Cost distribution for the different number of floors.

The final results are presented in Figure 5.26 to Figure 5.30. It is noted that in the four-floor solution, no shear walls were added, as the outer walls of the building can still absorb all the horizontal forces. As a result, the algorithm minimizes cost by reducing the contribution from slabs, accomplished by designing efficient beam-column layouts that allow for the utilization of EX18 hollow core slabs. This slab type has the smallest height dimension in the slab database and is, therefore, the most cost-effective option.

As the number of floors increases to eight, the algorithm can still rely primarily on the outer walls, but it does need to include a shear wall to reinforce them. It is also noted that the algorithm utilizes primarily the same beam-column line configuration, implying that the system is the most efficient option. However, in the extended section of the building plan, the span direction has been altered, presumably to add vertical load to the outer shear walls and to increase their stabilizing capacity.

For twelve floors, more significant modifications can be observed. The algorithm has added a beam-column line, likely to distribute the increased vertical load across multiple columns. However, the most significant change is the increased use of shear walls in both directions. This change indicates that it is no longer feasible only to redirect forces to the outer walls.

Additionally, it is worth noting that if the walls were combined into a resultant, the vector direction of this resultant would align well with the critical wind resultant from the WNW direction, as depicted in Figure 5.23.

Based on these observations, it can be concluded that the algorithm responds logically to changes in the number of floors.

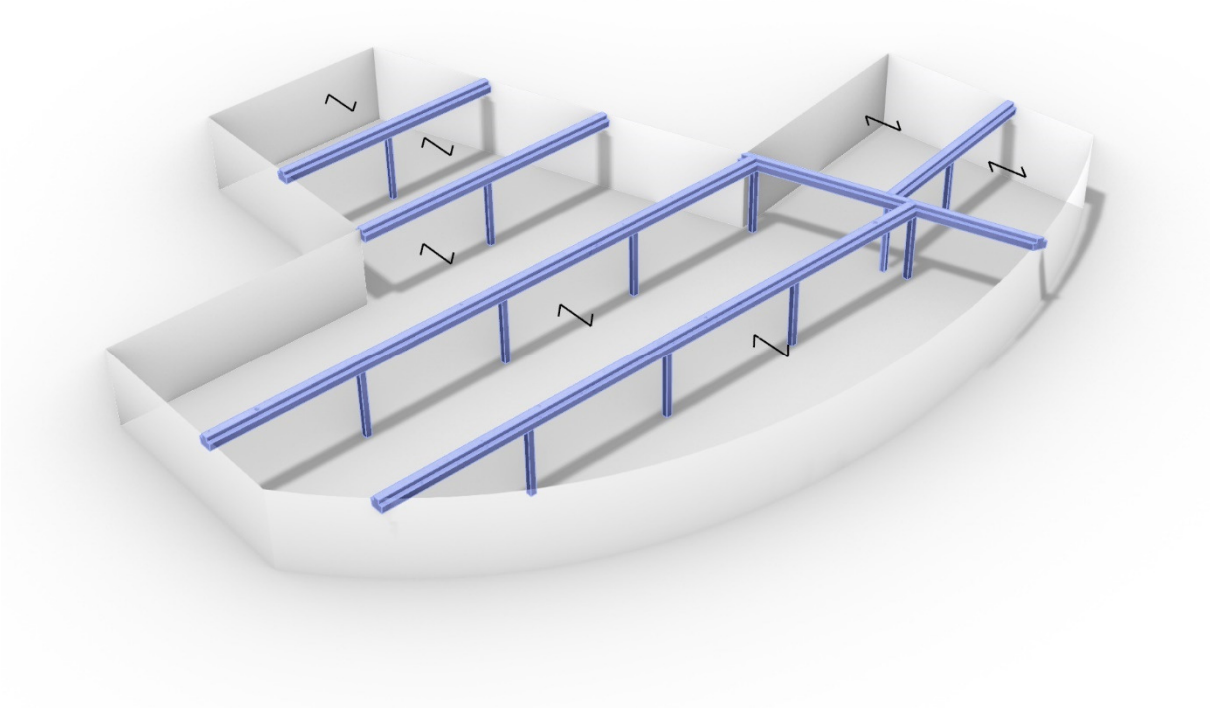


Figure 5.26 – 3d visualization of final solution using 4 floors, CPM2 objective = 387.2

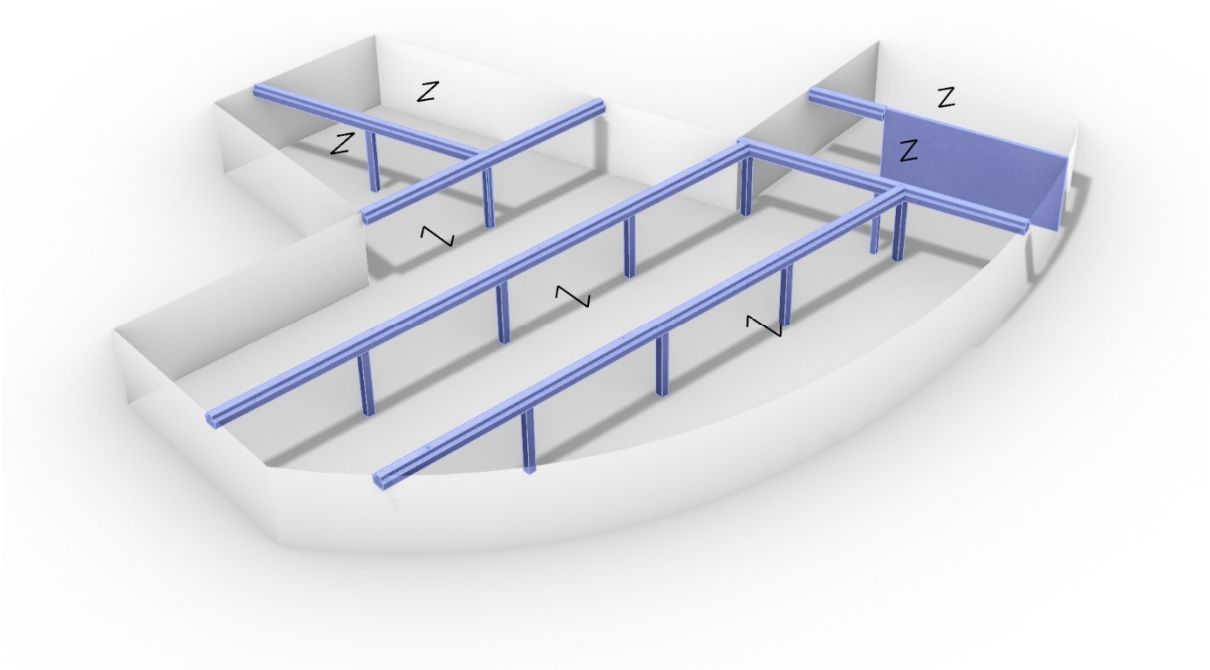


Figure 5.27 – 3d visualization of final solution using 8 floors, CPM2 objective = 438.4

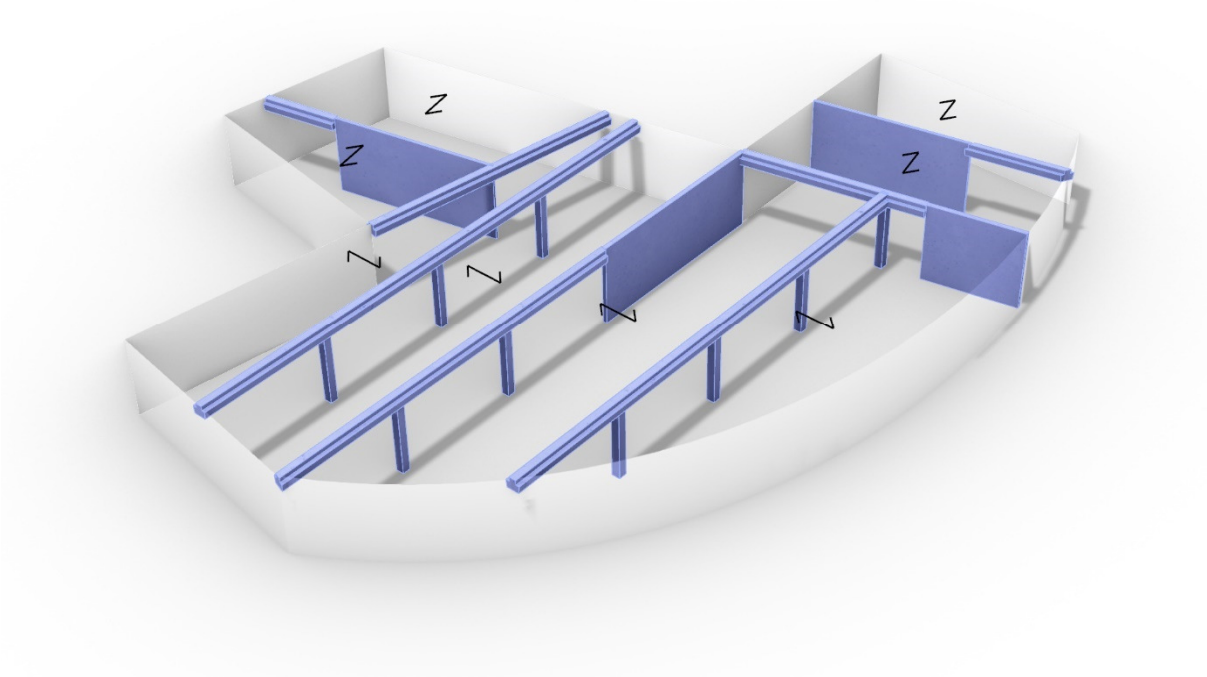


Figure 5.28 – 3d visualization of final solution using 8 floors, CPM2 objective = 488.7

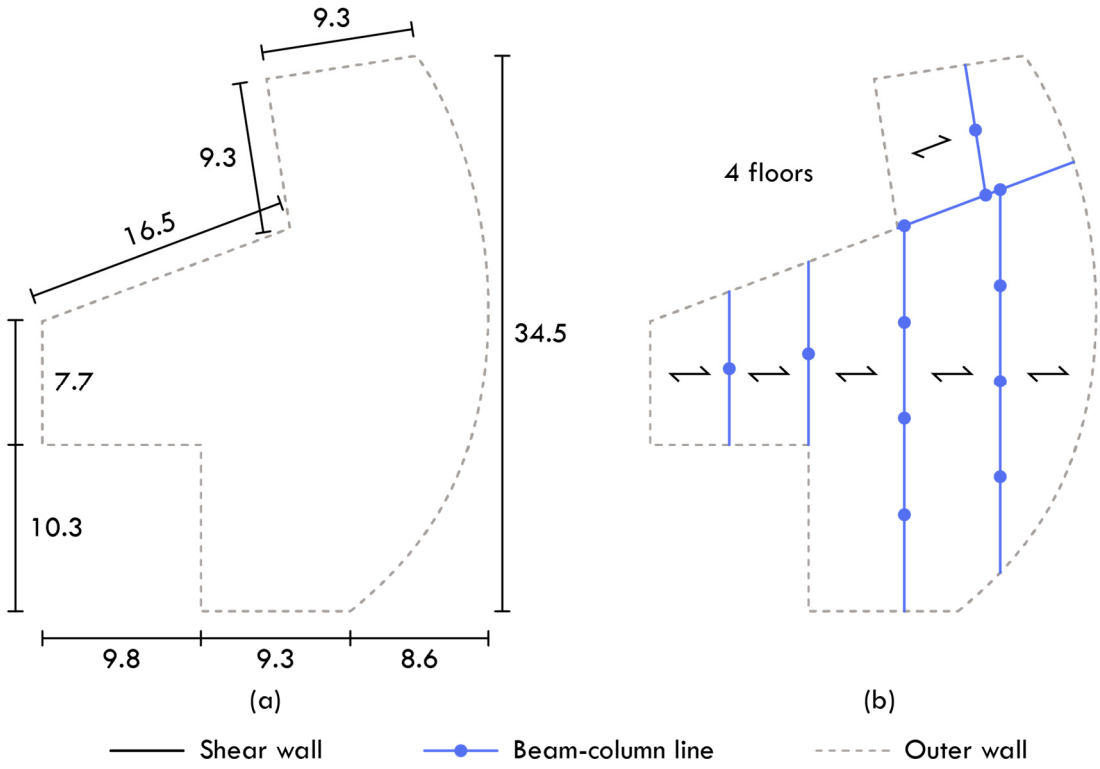


Figure 5.29 – Plan view of the structural layout, with (a) representing CPF =1 and (b) representing 4 floor.

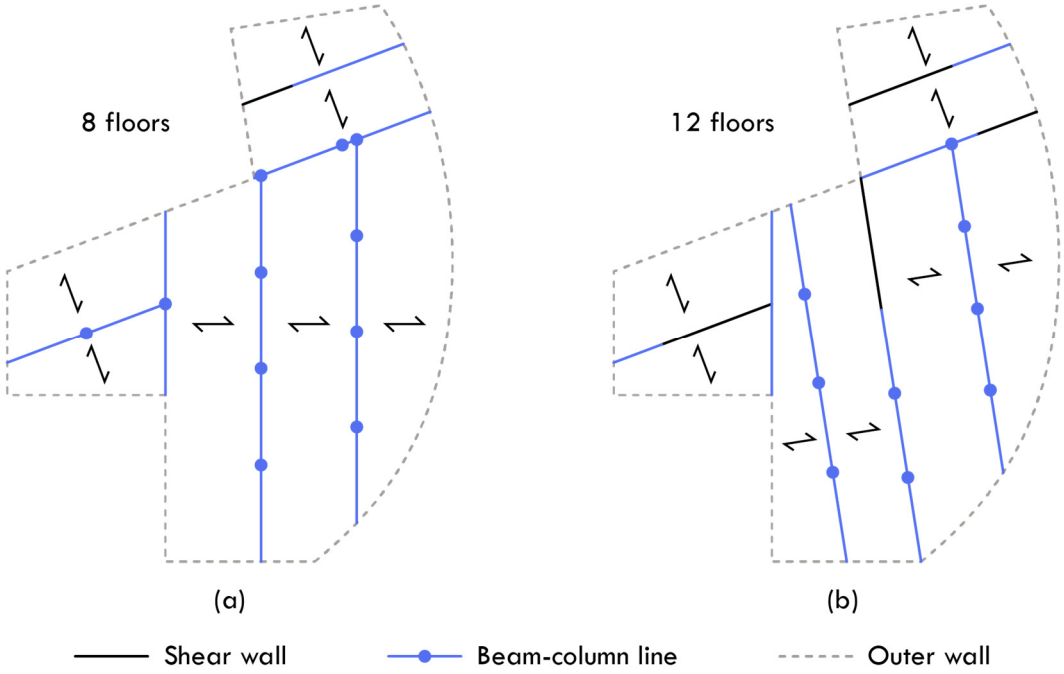


Figure 5.30 – Plan view of the structural layout, with (a) representing 8 floors and (b) representing 12 floors.

5.2.3 Limited slab options

This analysis examines the impact of a reduced slab database on the final design. Specifically, the option to select hollow core slabs has been eliminated. The algorithm can only select from different versions of the TT slab type, which is suited for longer spans due to its large cross-section. Consequently, the algorithm is expected to utilize larger spans than in the previous examinations. The algorithm is applied to the same building plan used in Figure 5.26, with identical settings except for the reduced slab database.

Figure 5.31 demonstrates a rapid convergence toward an optimum CPM2 value of 655.0, corresponding to a 69.2% increase compared to the solution that used hollow core slabs. This increment was expected as the algorithm is forced to use TT-slabs, which have a higher cost per square meter, and because the overall building height increases. The rapid convergence can be attributed to the reduced slab database, which decreases the solution space the algorithm has to search through.

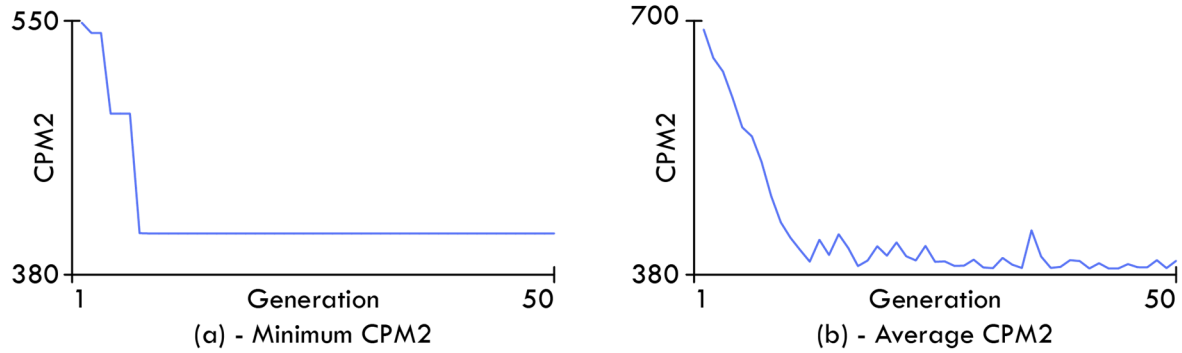


Figure 5.31 – Convergence plot for both the minimum CPM2 (a) and the average CPM2 (b) values.

Figure 5.32 and Figure 5.33 illustrates the final solution. It is evident that the algorithm has generated a substantially different solution than the one presented in Figure 5.26. As expected, the algorithm has widened the distance between beam-column lines to maximize material usage. This optimization is also reflected in the algorithm's decision to utilize TTD78/240 slabs throughout the structure, as it was the TT-type with the lowest height available.

As a consequence of the algorithm being forced to use the TT-slab type, the overall floor-to-floor height increases from 3.38m to 3.98m since the algorithm must always ensure the defined minimum clear height. Using TT-slabs also means that the beam's height and stiffness increase. Therefore, even though the beam experiences a significant increase in load, the column spacing remains comparable to that seen in Figure 5.26.

Although using TT-slabs in the solution increases the cost per square meter, this strategy may present additional benefits, including practical and aesthetic considerations. Table 5.3 shows that the solution with TT-slabs has an almost double ACS value. A larger open area provides a better design freedom. Furthermore, the cost assessment is solely based on material

prices, and there may be other factors to consider, such as the number of work processes and crane lifts.

Based on these considerations and results, it can be concluded that the availability of structural elements significantly impacts the design of the structural layout and final cost.

Table 5.3 – ACS values for both the solution with all slab types available and the solution with limited options.

Case	ACS
All slab types	82.6 [m²]
Only TT-slabs	144.5 [m²]

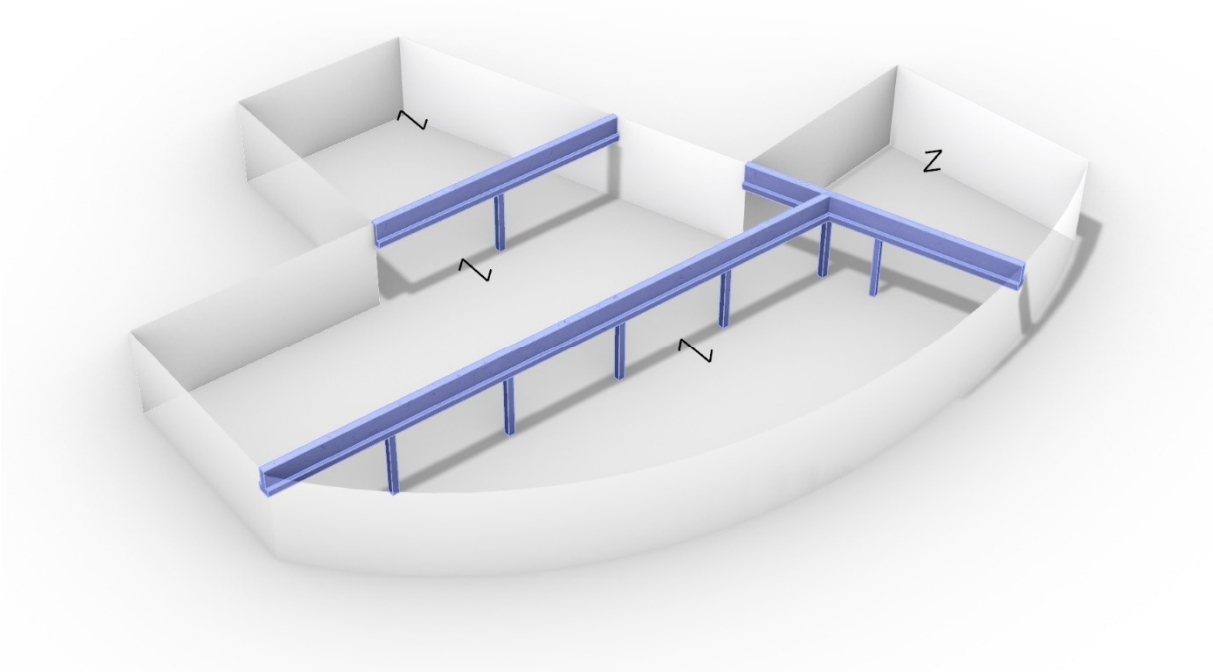


Figure 5.32 – 3d visualization of final solution with limited slab options, CPM2 objective = 655.0

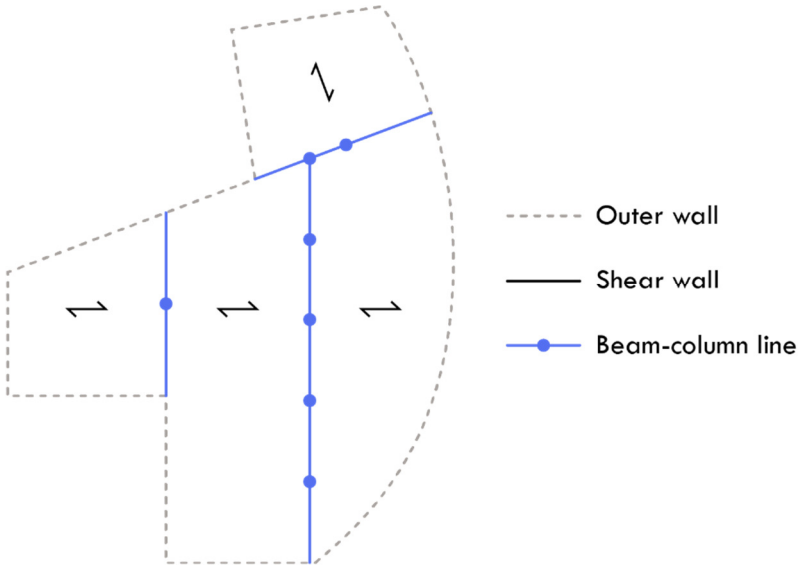


Figure 5.33 – Plan view of the structural layout suggestion using a limited slab database.

5.2.4 Different recess ratio settings on the outer wall

This analysis investigates whether the outer wall recess ratio (RR) parameter functions as intended by redirecting horizontal forces within the structure. As described in section 4.6.5, the RR parameter denotes the percentage of the outer walls containing recesses, such as windows and doors. A high RR value reduces stiffness, as less material is available to resist the overturning moment. Furthermore, outer walls with a high RR value will be more affected by vertical load and wind loads perpendicular to the surface due to a concentration of forces on a smaller area, further reducing the element's ability to counteract the overturning moment. Therefore, the algorithm is expected to add inner shear walls to increase the structure's capacity against the overturning wind forces for the solution with a high RR value.

The analysis is carried out on a block building, as this building type has a high proportion of outer walls relative to the plan area, which will emphasize the impact of the RR parameter on the final design. The building plan and geometry are shown in Figure 5.34 (a), while the critical wind direction is illustrated in Figure 5.34 (b). The algorithm conducts two analyses, one with a low RR value set to 0.2 and another with a high RR value set to 0.8. The number of floors is set to six to amplify the effect, but standard settings are used otherwise.

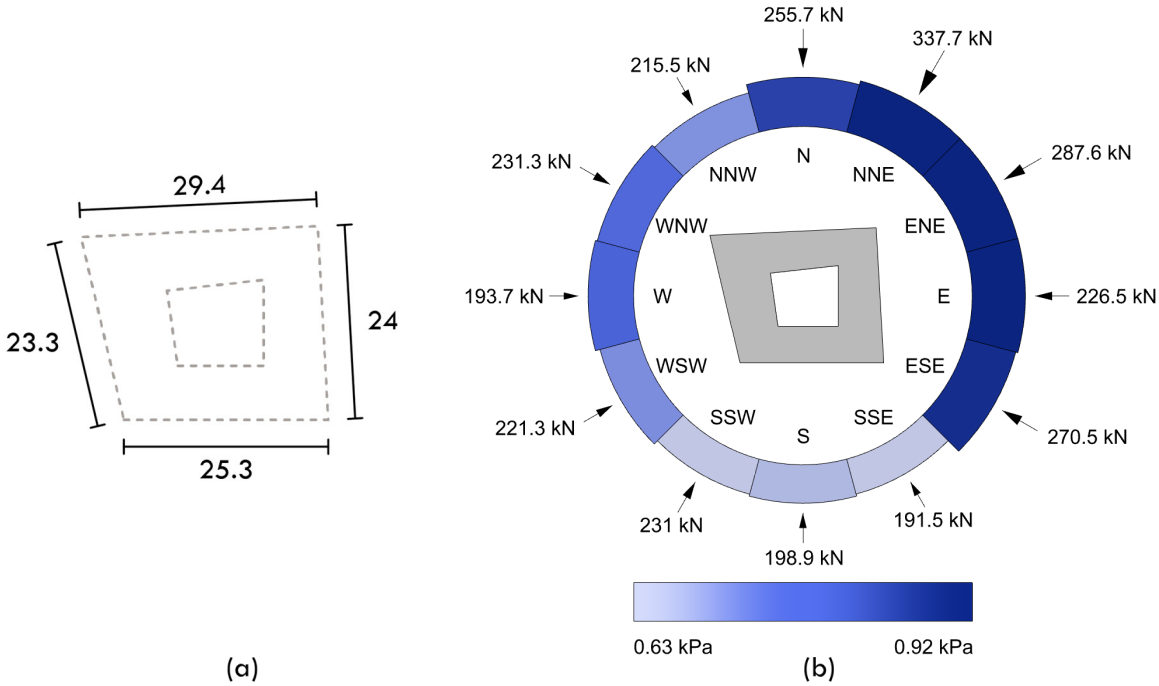


Figure 5.34 – (a) Illustration of the wind resultants and corresponding wind pressure values for 12 directions. (b) The geometric dimensions of the applied building plan.

Figure 5.35 demonstrates a convergence towards an optimum with a minimum CPM2 value of 380.2 and 407.9, respectively, for the solution with RR=0.2 and RR=0.8. The price for the outer wall is determined based on the total surface area. Consequently, the price difference reflects the variation in the wall thickness and reinforcement level in the outer walls, as well as the difference in the inner structural layout. Therefore, it is reasonable that the value with the RR=0.8 is slightly higher, even though the floor height for this solution has decreased from 3.47m to 3.17m.

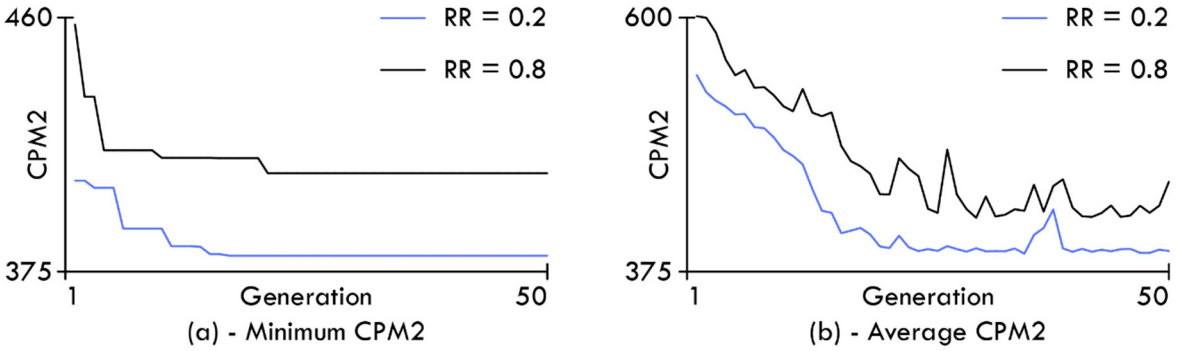


Figure 5.35 – Convergence plot for both the minimum CPM2 (a) and the average CPM2 (b) values.

Figure 5.36 to Figure 5.38 illustrates the final solutions. From a general static perspective, the algorithm has opted to orient the slab direction to utilize the permanent load-bearing lines in the outer walls, thus reducing material usage. As expected, the algorithm has increased the number of shear walls to counteract the overturning moment better. Additionally, it is observed that the algorithm has applied the same fundamental layout system across two distinct optimization iterations, indicating that the configuration is efficient.

Based on these results, it can be concluded that the RR value has the intended effect and can be used to redirect horizontal forces to and from the outer walls.

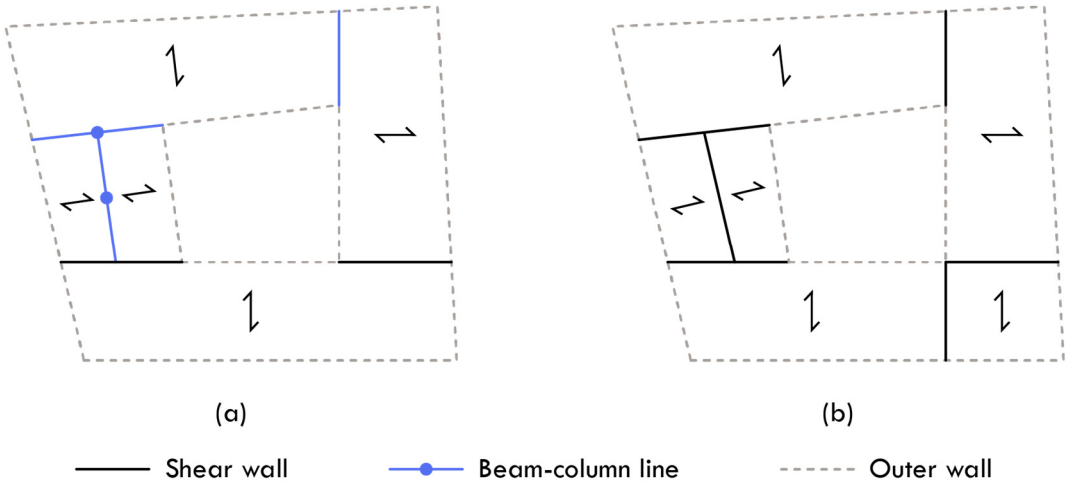


Figure 5.36 – Plan view of the structural layout, with (a) representing RR=0.2 and (b) representing RR=0.8

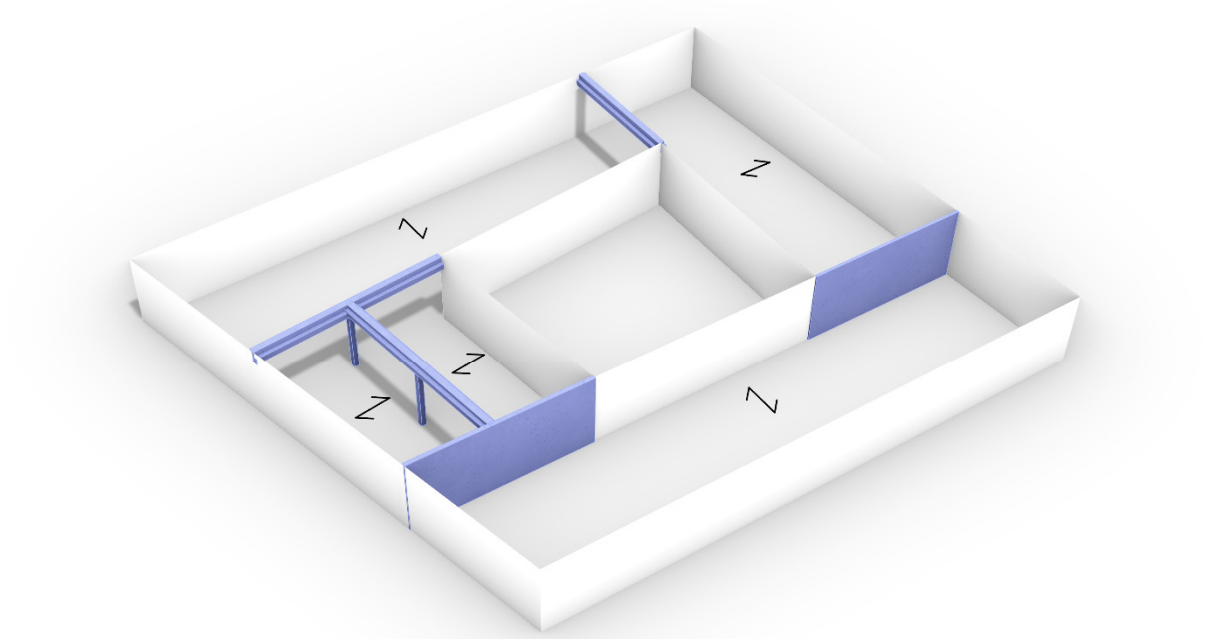


Figure 5.37 – 3d visualization of final solution using RR=0.2, CPM2 objective = 380.2

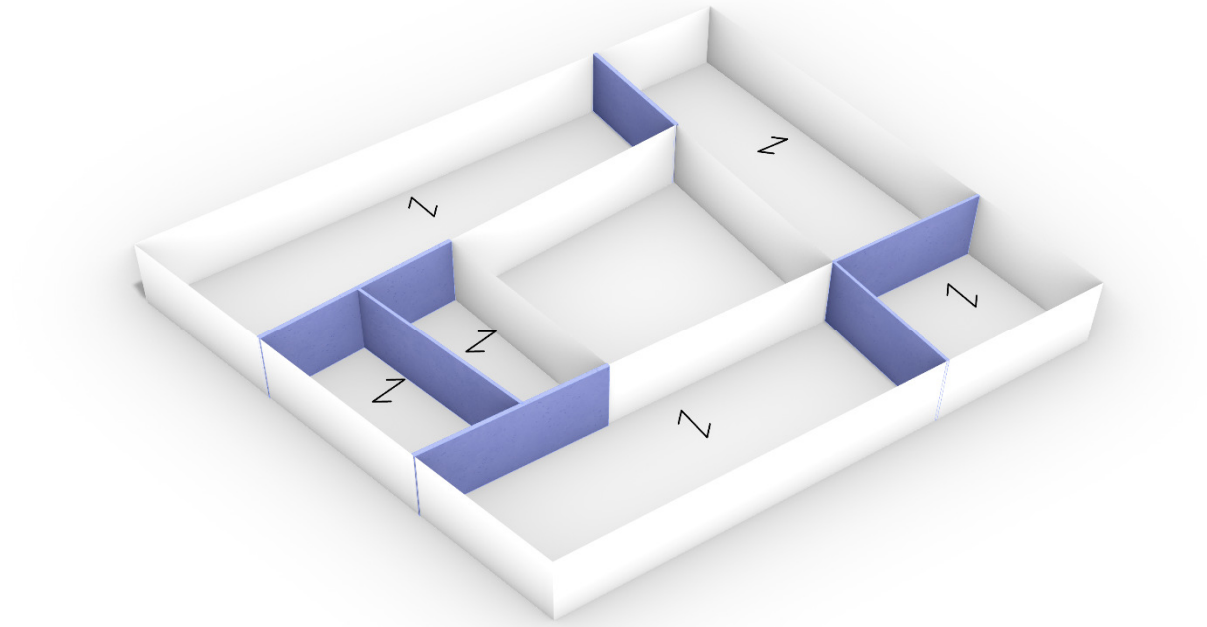


Figure 5.38 – 3d visualization of final solution using RR=0.8, CPM2 objective = 407.9

5.2.5 Change of live load

This analysis investigates the impact of vertical load on the optimized structural layout. The analysis is conducted on a building shape with a relatively large surface area, making changes in the inner structural layout more apparent. The building plan and geometry are depicted in Figure 5.39. The critical wind load is not illustrated, as it does not influence the final design. The analysis will be carried out on two cases, one with a surface load equal to 1.5 kN/m2, corresponding to category A (residential load), and another case using a surface load equal to 7.5 kN/m2, corresponding to category E (traffic load).

Figure 5.40 demonstrates a convergence towards an optimum with a minimum CPM2 value of 359.8 and 388.2 for the solutions with live load A and live load B, respectively.

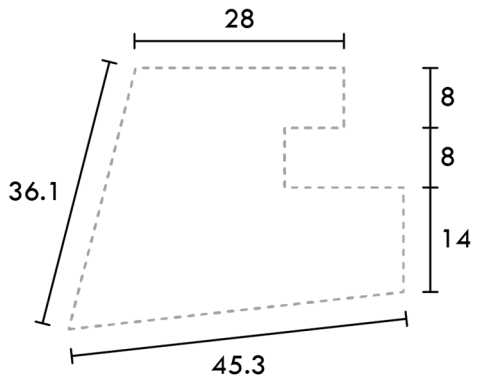


Figure 5.39 – The geometric dimensions of the applied building plan.

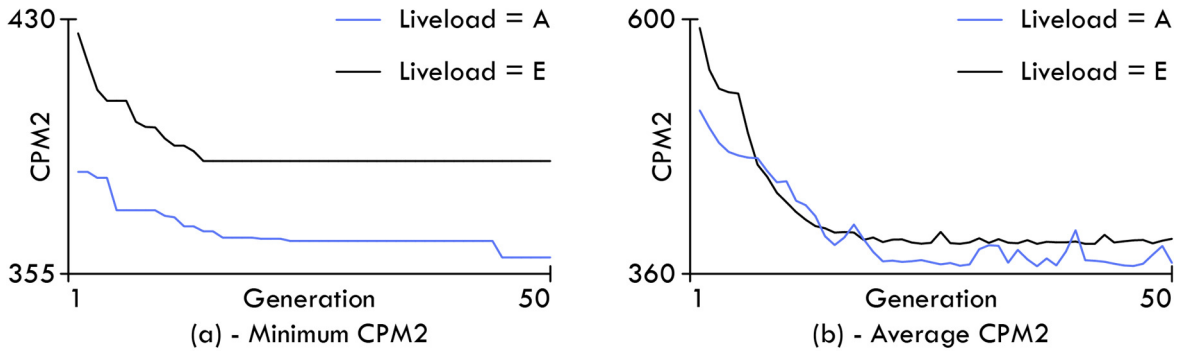


Figure 5.40 – Convergence plot for both the minimum CPM2 (a) and the average CPM2 (b) values.

Figure 5.41 to Figure 5.43 illustrate the final solutions. As expected, the structural layout is dominated by beam-column lines, as this is the most cost-effective option. It is evident that the algorithm still favors the utilization of EX18 hollow core slabs, even across load categories. The variance in the structural layout lies in the density of beam-column lines and the column spacing. This approach is reasonable from both a static and economic perspective and is consistent with the findings of previous studies in Chapter 5.

Figure 5.41 (a) shows that only a single shear wall is utilized in the small wing. It is assumed that the algorithm has placed a wall here instead of a beam-column line because it uses an EX32 hollow core slab to cover the span. If the algorithm had used a beam-column line, the floor height would have increased because of the defined parametric rules to ensure the minimum clear height. Additionally, none of the surrounding slabs have a bearing on the wall; thus, the minimum thickness can be used, which is more cost-effective. This approach is logical, considering the defined constraints and design objectives. However, from a practical and static viewpoint, this wall does not make sense, as Figure 5.41 (b) illustrates that the outer walls can withstand the overturning moment. The wall is present because the rules dictate that all edge curves in an APoly shape must have a function. Thus, the placement of this wall serves as an example of how the algorithm can still be refined, such as by allowing non-vertically loaded support lines to be empty.

However, considering the defined rules, it can be concluded that the algorithm has generated logical and cost-effective solutions based on the different live load values.

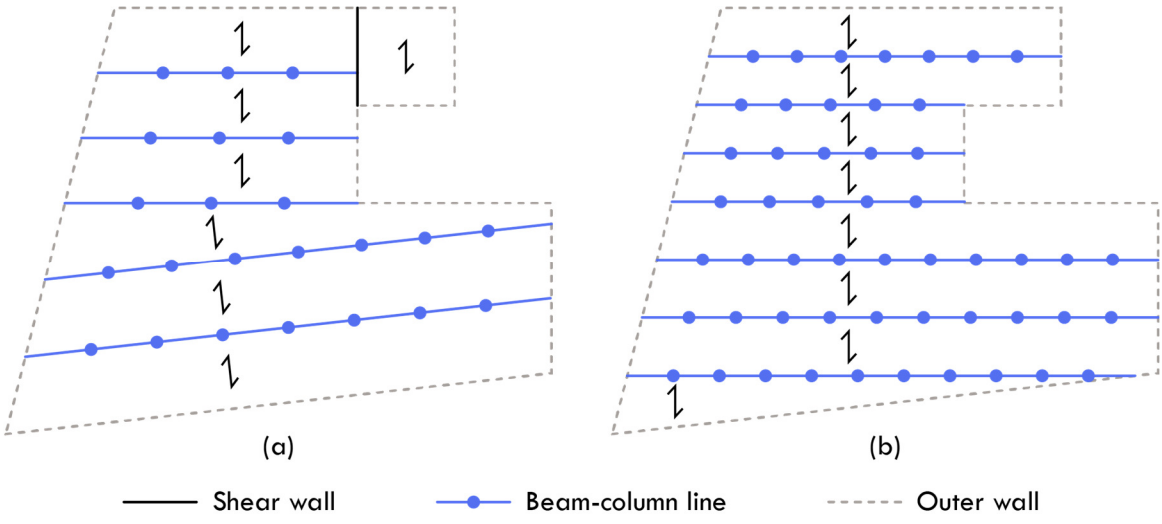


Figure 5.41 – Plan view of the structural layout, with (a) representing live load category A and (b) representing live load category B.

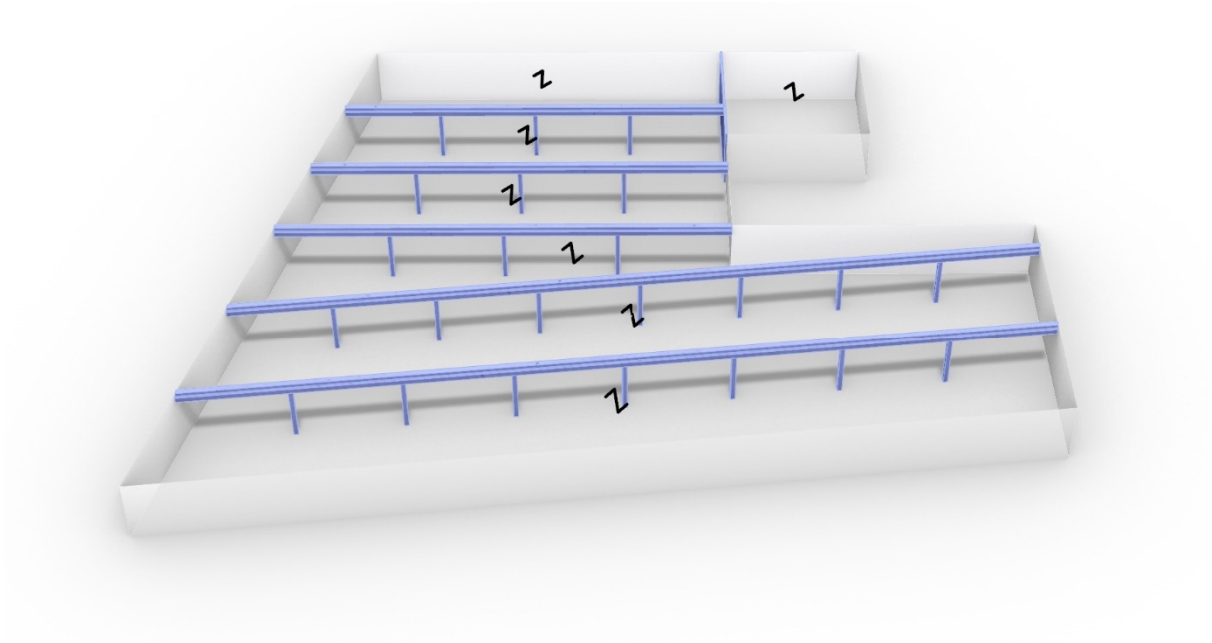


Figure 5.42 – 3d visualization of final solution using live load category A, CPM2 objective = 359.8

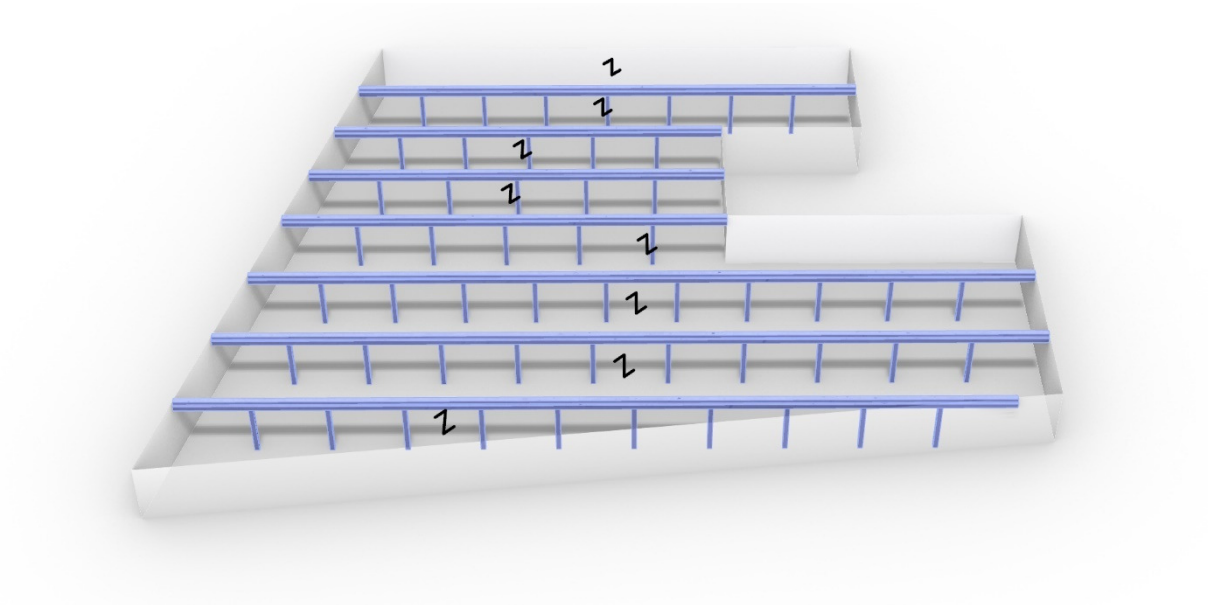


Figure 5.43 – 3d visualization of final solution using live load category E, CPM2 objective = 388.2

5.2.6 Different locations

This analysis investigates the influence the location can have on the optimized structural layout of a building. Two optimization iterations will be performed with the same settings, except for the placement of the building. Both analyses will be placed in a real-world location using the wind load values dictated by the location. Both locations are selected to significantly impact the building with wind load from a specific direction. In this case, from a Northern and Eastern direction, clearly illustrated in Figure 5.46.

The variation in load outcomes is primarily attributed to unfavorable terrain categories and orography values for specific directions. The terrain categories are defined based on the radar images found in Appendix A. The orography values are visualized in Figure 5.47.

The geometry of the affected building plan is illustrated in Figure 5.45 (b). The building is designed with teen floors to ensure that the wind load significantly impacts the final structural layout. Otherwise, standard settings are applied.

Figure 5.44 demonstrates a convergence towards an optimum with a minimum CPM2 value of 461.3 and 483.6 for the solution with the critical Northern and Eastern directions, respectively. It is noted in the graph that the algorithm took longer to reach an optimum in this analysis compared to previous studies. This scenario can be attributed to the significant wind load the algorithm has to manage in addition to the vertical loads.

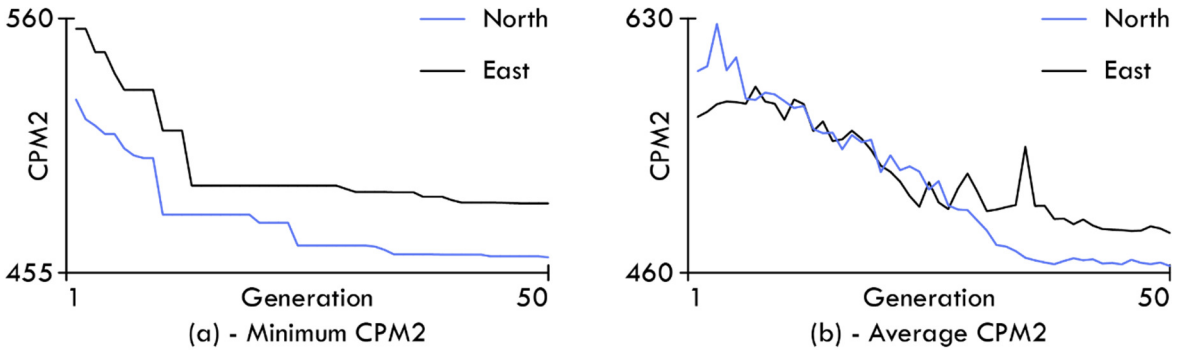


Figure 5.44 – Convergence plot for both the minimum CPM2 (a) and the average CPM2 (b) values.

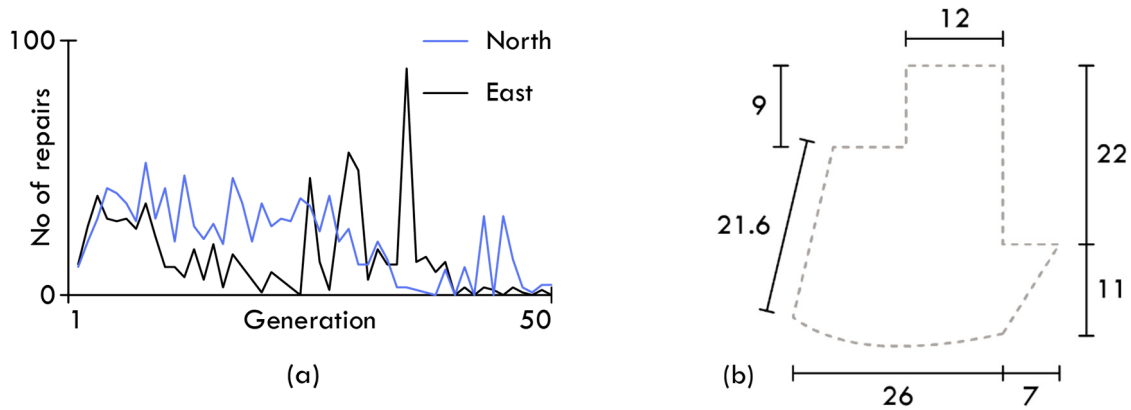


Figure 5.45 – (a) Number of repairs conducted by the algorithm. (b) The geometric dimensions of the applied building plan.

Furthermore, it can be observed that the algorithm utilizes more material for the solution in the critical Eastern direction compared to the critical Northern direction. This observation is apparent since the critical wind resultant is more significant for the eastern direction than the critical Northern direction, as demonstrated in Figure 5.46.

Figure 5.45 (a) illustrates the number of repairs per generation. There is no apparent pattern in the number of generations, but it is noted that the repair algorithm has been considerably more active in these optimization iterations than previously observed. This activity is logical since the repair algorithm is designed to increase the length of the stabilizing wall where necessary. It can therefore be concluded that the repair algorithm outlined in Figure 4.44 works as intended and contributes to increasing the number of valid solutions in each generation.

The optimized results are illustrated in Figure 5.48 to Figure 5.50. As expected, the algorithm has allocated more shear walls along the critical wind directions to withstand the overturning moment. This strategy was especially evident for the case with the critical Eastern direction, as this value was exceptionally high. Additionally, it is noted that the algorithm also optimized the positioning of the beam-column lines to minimize the number of columns, indicating that it still takes the vertical loads into consideration. This approach is further evident in the resulting structural layout, which has been designed to accommodate the use of EX18 hollow core slabs.

In general, it can be concluded that the building's location can significantly impact the final structural layout, provided that the building is highly exposed to wind loads.

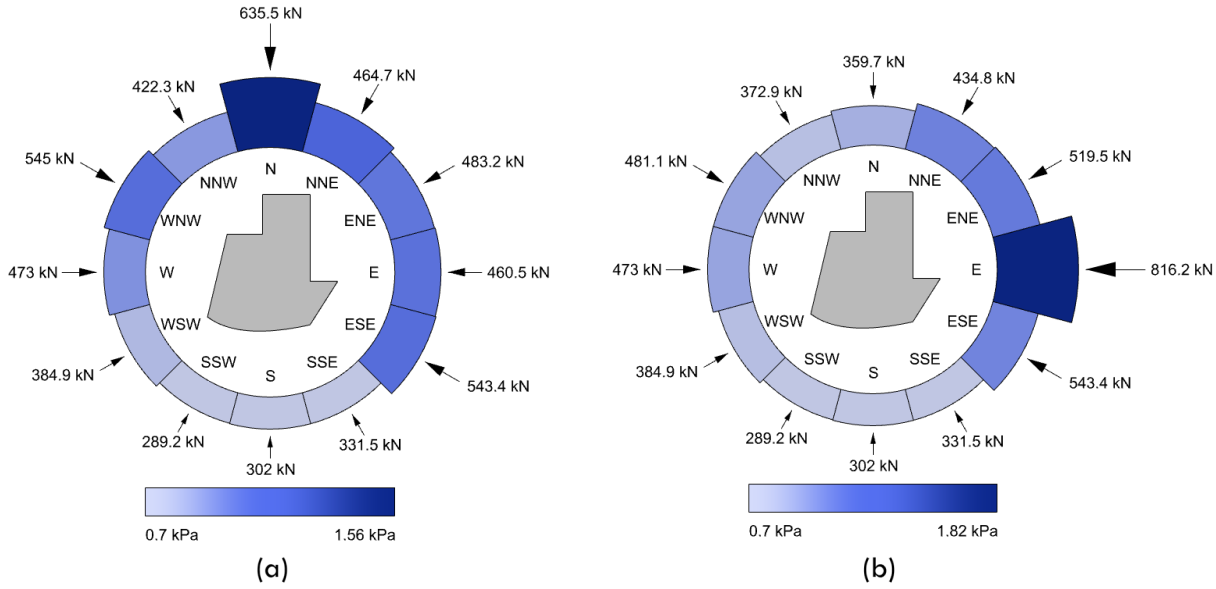


Figure 5.46 – (a) Wind resultants and corresponding wind pressure values for the critical Northern direction. (b) Wind resultants and corresponding wind pressure values for the critical Eastern direction.

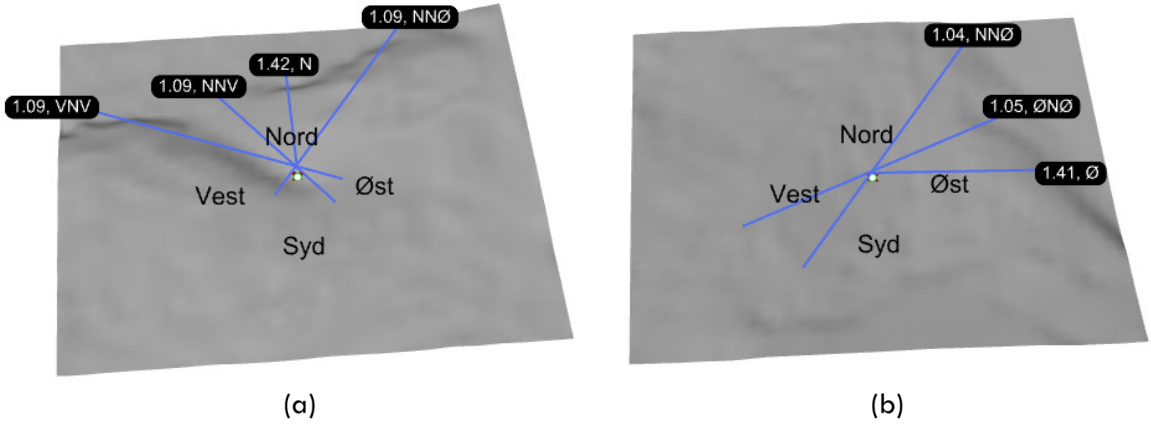


Figure 5.47 – (a) Orography values for the critical Northern direction. (b) Orography values for the critical Eastern direction.

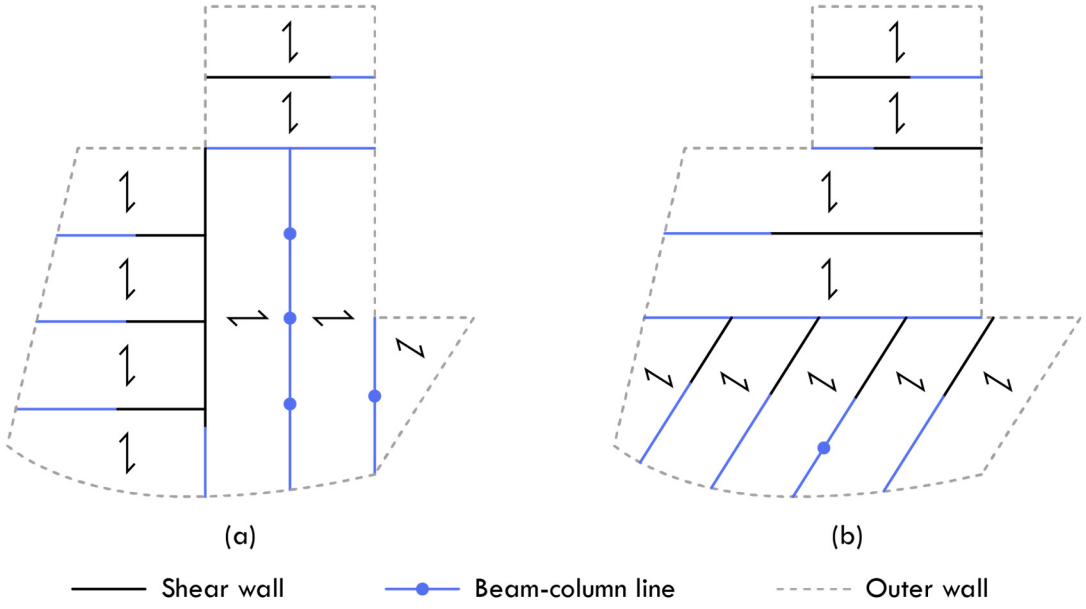


Figure 5.48 – Plan view of the structural layout, with (a) representing the critical Northern direction and (b) representing the critical Eastern direction.

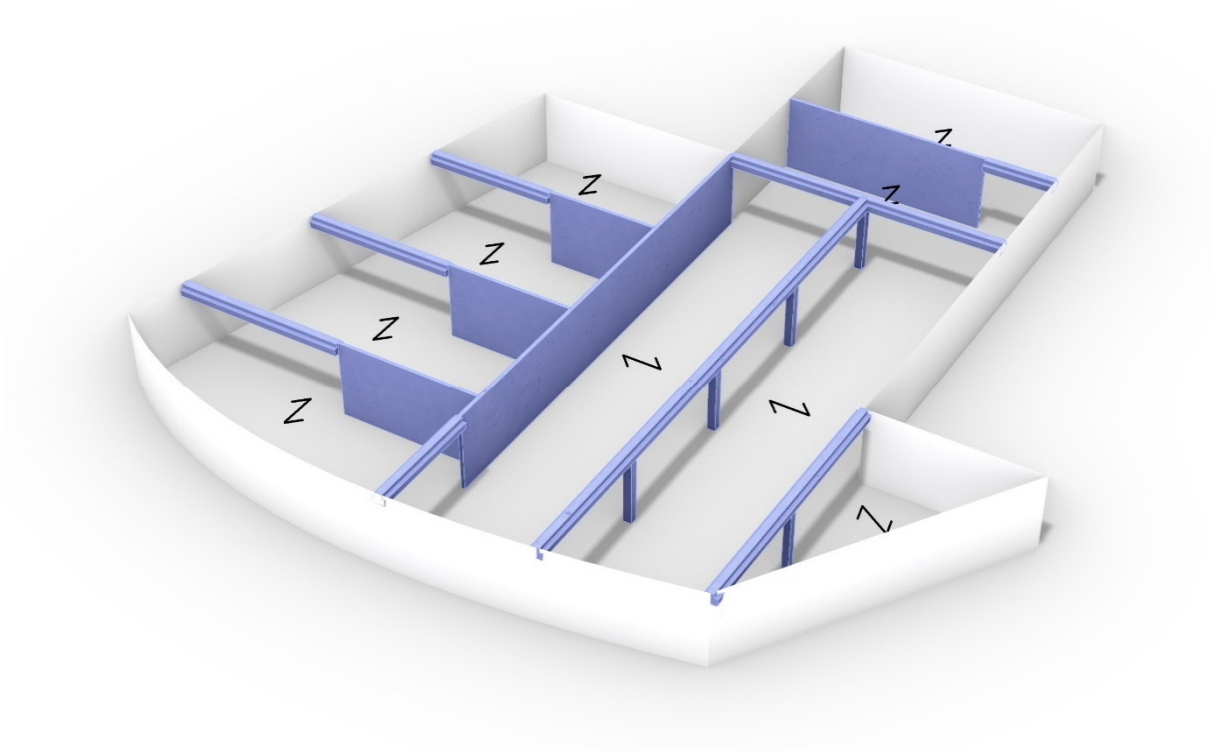


Figure 5.49 – 3d visualization of final solution for the critical Northern direction, CPM2 objective = 461.3

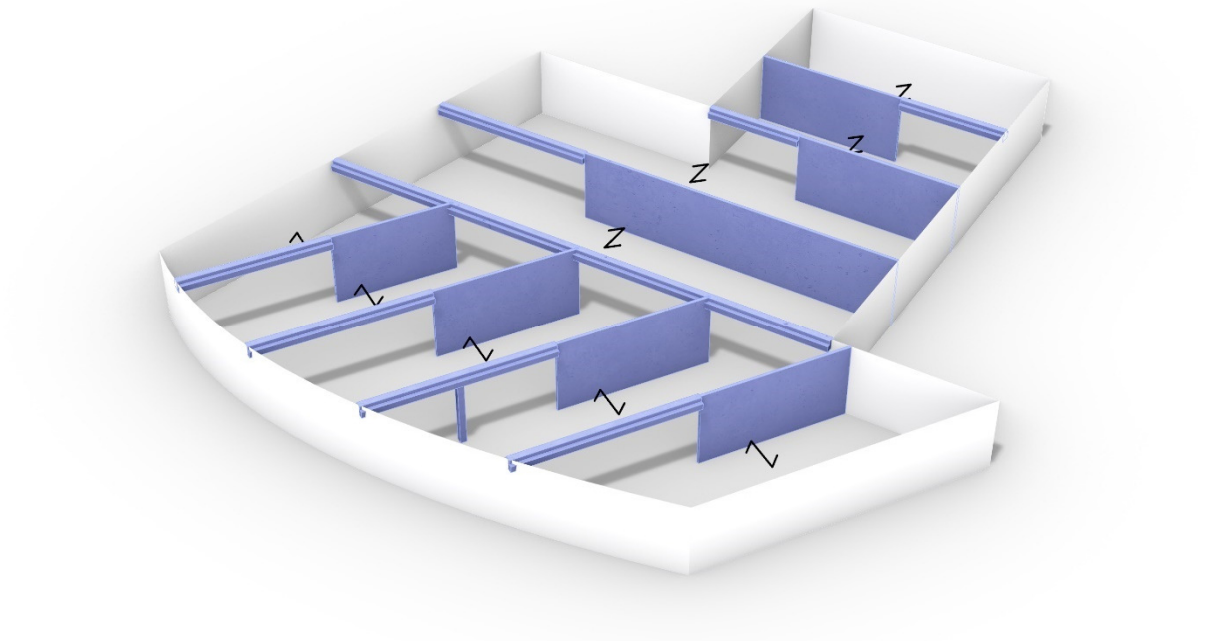


Figure 5.50 – 3d visualization of final solution for the critical Eastern direction, CPM2 objective = 483.6

5.2.7 Hybrid solution

As described in section 4.3.4, a hybrid solution is a partially defined solution where the user has already designed a part of the structural layout. The algorithm then adapts to the user-defined inputs when completing the remaining sections of the structural layout. The effectiveness of the algorithm's ability to adapt to user input is examined and compared to a similar solution where the algorithm had complete design freedom.

The analysis builds upon the previous case study, which examined the critical Northern wind direction, as shown in Figure 5.49. The population size in the GA is increased to 500 to account for a higher proportion of solutions that are likely to fail in the first generation. However, standard settings have been applied in all other aspects. The custom input is illustrated in Figure 5.52 (a), where an open area without any interior load-bearing lines has been specified. This input reflects a plausible functional requirement that could occur in real-world scenarios.

Figure 5.51 (a) demonstrates a convergence towards an optimum with a minimum CPM2 value of 472.5, a 2.4 percent increase compared to the non-restricted optimization. This result is very satisfying as it was expected to be even higher; it can likely be attributed to the unrestricted optimization not having fully converged, as indicated in Figure 5.44 (a).

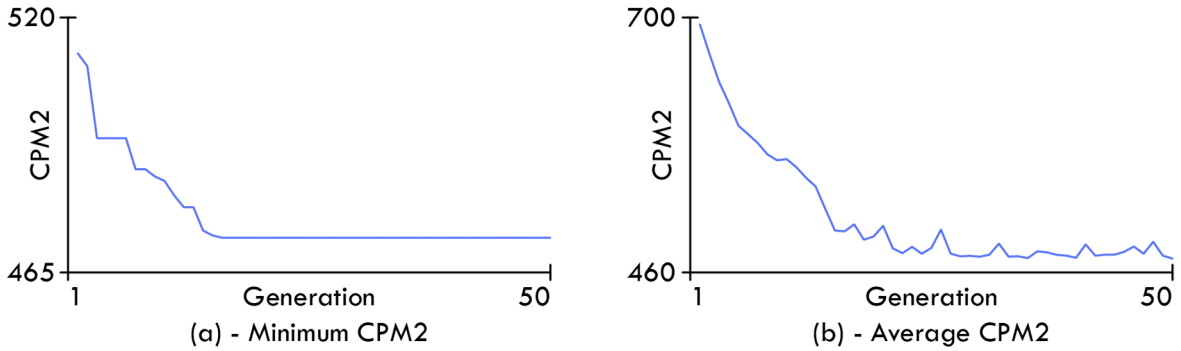


Figure 5.51 – Convergence plot for both the minimum CPM2 (a) and the average CPM2 (b) values.

The final solutions are illustrated in Figure 5.52 (b) and Figure 5.53. It can be observed that the algorithm has chosen to position the shear walls around the fixed APoly shape, which was not included in the customized inputs. Additionally, the algorithm has opted to use a similar structural layout, indicating that the layout is efficient.

In summary, the algorithm can effectively incorporate user input and generate optimized hybrid solutions based on the defined objectives and constraints.

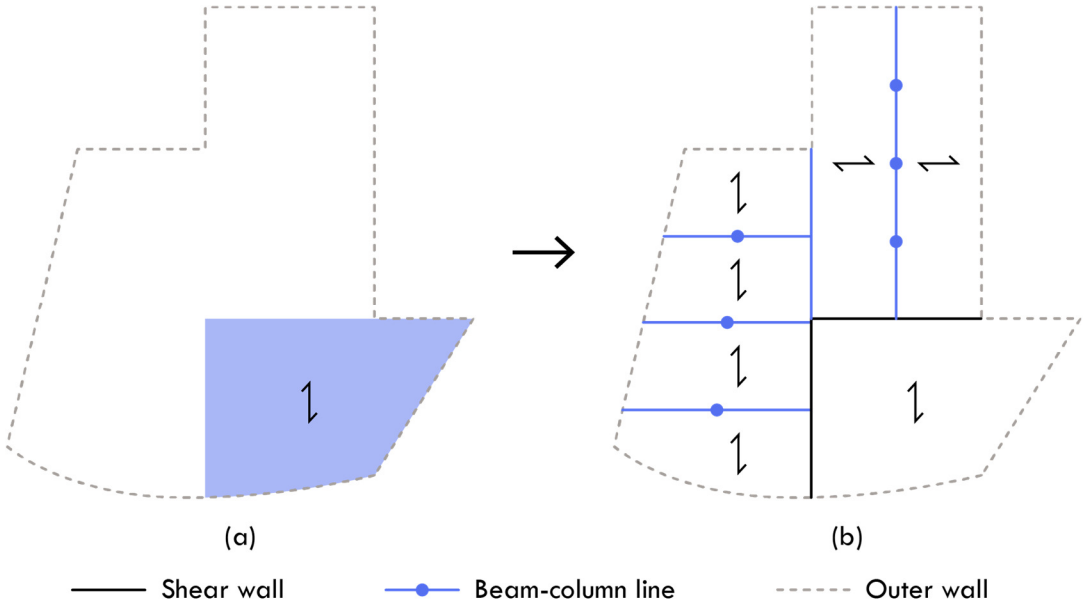


Figure 5.52 – (a) User defined APoly shape and slab direction. (b) Final plan view of the structural layout

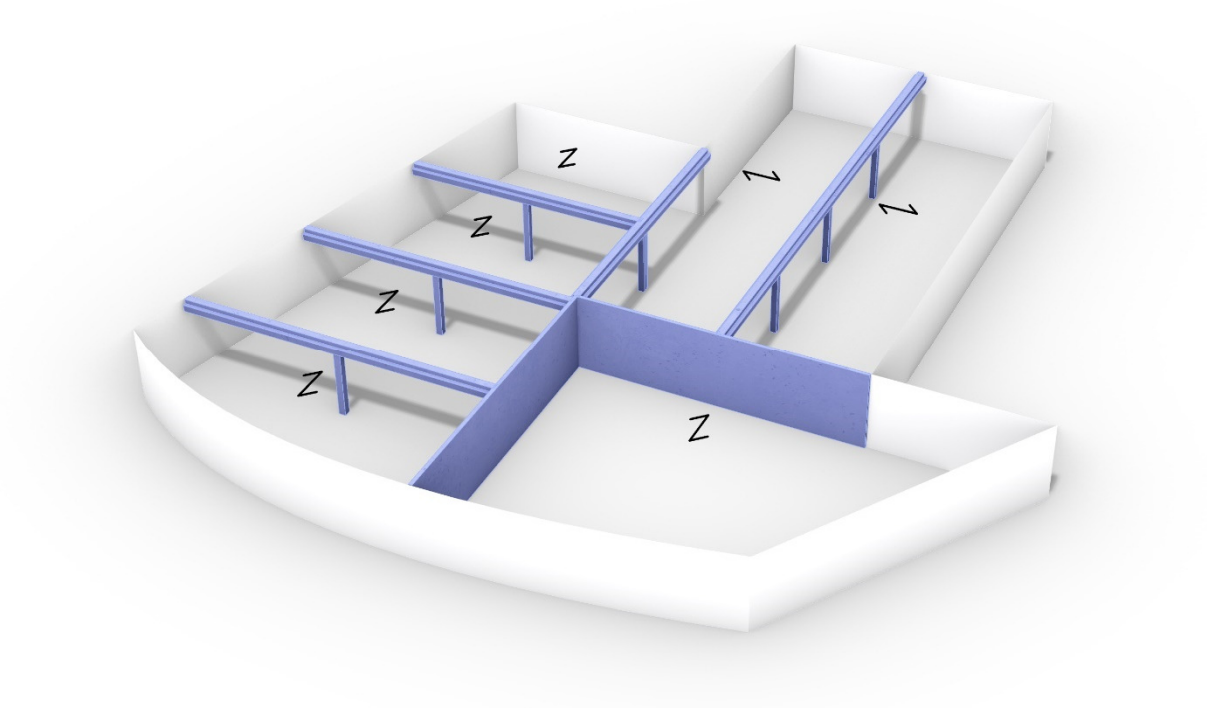


Figure 5.53 – 3d visualization of final hybrid solution, CPM2 objective = 472.5

5.2.8 Multi-objective optimization

A MOO analysis is conducted to investigate whether the algorithm can generate optimized solutions that take multiple objectives into account simultaneously. This analysis employs the CPM2 objectives used in all previous case studies and the ACS objective. These two objectives are in conflict, such that improving one objective will lead to a deterioration in the other. Consequently, selecting a solution will require determining a satisfactory trade-off between the design objectives.

The analysis is performed on the same building plan as shown in Figure 5.26 with the same assumptions, except for an increase in the initial population in the GA from 200 to 300. The HypE algorithm will be employed to select the elite individuals in each generation.

Regarding the two defined objectives, solutions at each end of the spectrum are expected to resemble the solution shown in Figure 5.26 when the CPM2 objective is prioritized and the solution shown in Figure 5.32 when the ACS objective is prioritized.

The convergence plot for the CPM2 objective is presented in Figure 5.54 (a). The ACS convergence plot is not displayed as it only changed twice and thus does not provide much information. The convergence values also represent only the extreme values of the Pareto front and therefore offer limited information about whether the Pareto front is still being optimized. However, as shown in Figure 5.54 (b), the number of invalid solutions has significantly decreased, suggesting that the algorithm has converged.

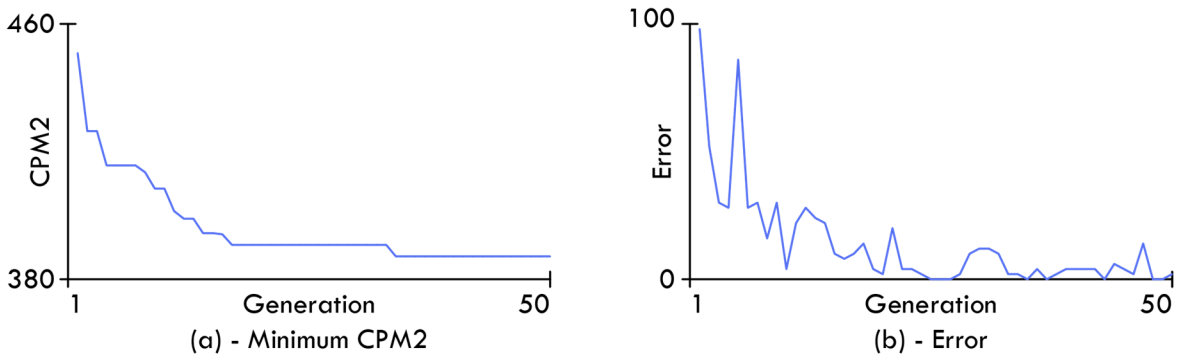
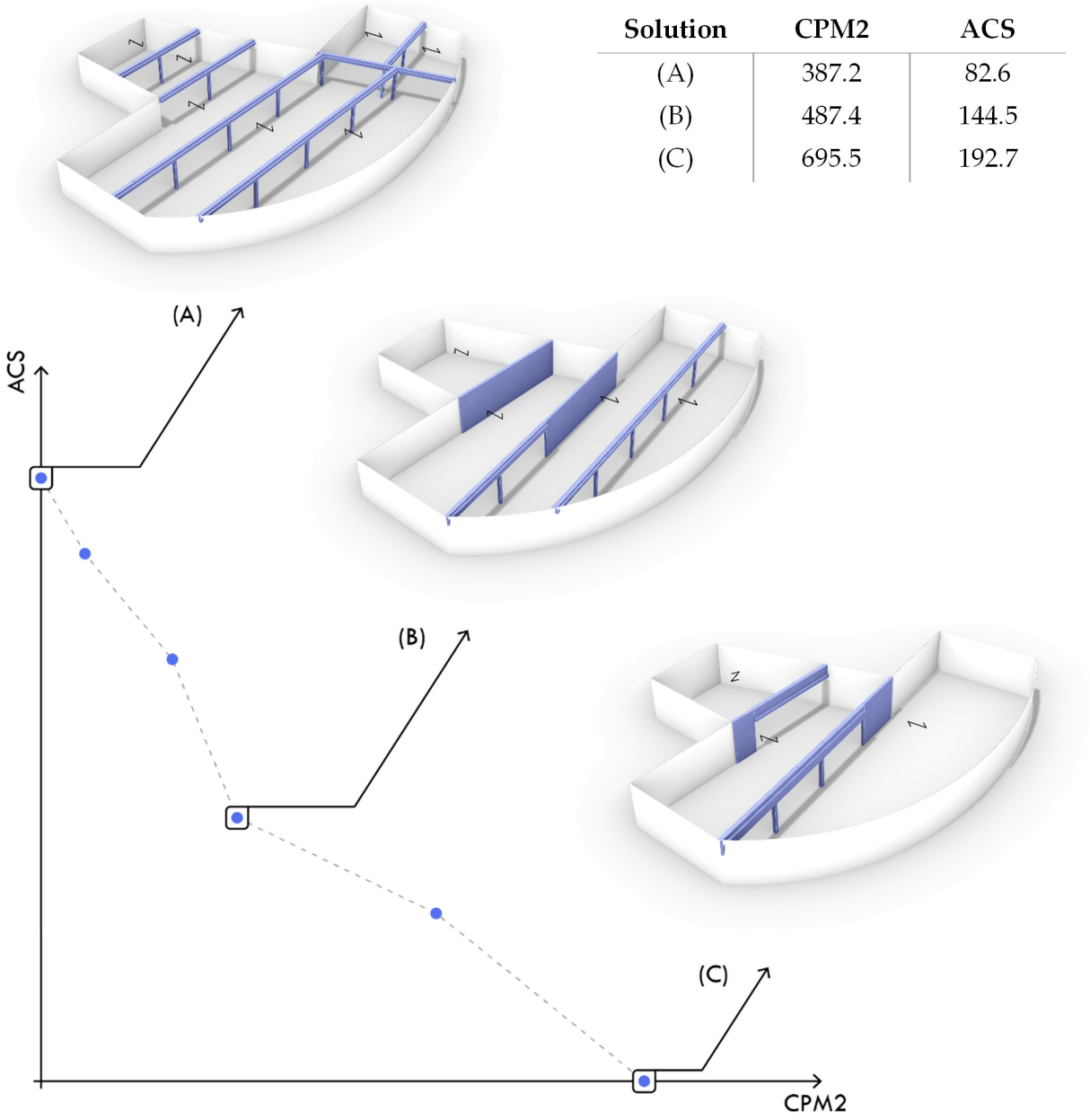


Figure 5.54 – (a) Convergence plot for the minimum CPM2 objective.
(b) Number of invalid solutions in each generation.

The trade-off between the objectives is clearly depicted in the Pareto front in Figure 5.55. Solution (A) is the same as the solution found in Figure 5.26, while solution (C) closely resembles the solution presented in Figure 5.32, although with a slightly higher CPM2 value. While it is not guaranteed that two optimization sessions will converge toward the same solution, given enough time, the mutation operators would ensure that the same solution would eventually be found. Solution (B) represents a solution where the objectives are weighted roughly equally, which is also evident from the objective values presented.

It is noted that the density of solutions in the Pareto front is moderately low, possibly due to the relatively simple building plan used in this analysis. Future sensitivity analysis may demonstrate whether the SPEA2 or NSGA-II algorithm can produce a more densely populated Pareto front. However, the previous case studies in sections 5.2.2 and 5.2.3 indicate that the solutions at the outer spectrum of the Pareto front are correct, and a strong trade-off can also be observed, as expected. Therefore, it can be concluded that the algorithm can incorporate multiple objectives and generate structurally valid solutions optimized with respect to the defined design objectives.



Solution	CPM2	ACS
(A)	387.2	82.6
(B)	487.4	144.5
(C)	695.5	192.7

Figure 5.55 – Visualization of the pareto front for the MOO analysis using the ACS and CPM2 objectives.

5.3 Summary

In Chapter 5, the performance and reliability of the design tool were validated through a series of local and global case studies. The validation studies were divided into two main groups.

The first group, presented in section 5.1, consisted of parameter sensitivity studies conducted on the structural approximation modules. These studies evaluated the modules' responsiveness to changes in different parameters and assessed whether the response was logical and consistent with expectations. The studies were conducted on the wall, beam, column, and BC line evaluation modules. All sensitivity studies demonstrated a logical response that was consistent with expectations. Therefore, it can be concluded that the modules function as intended and can be combined in the final design tool.

The second group of validation studies examined the design tool's effectiveness across relevant building plans and scenarios. The corresponding results demonstrated that the tool could adapt to different scenarios and settings and produce optimized results based on the design objective and active constraints.

It was also demonstrated that the tool could produce solutions adapted to the user interactivity features as defined in Chapter 3. These features enable the user to influence or define sections of the structural layout suggestions.

Furthermore, it was demonstrated that the design methodology could conduct multi-objective optimization and produce a front of Pareto optimal solutions.

All studies produced excellent results, confirming the tool's ability to generate efficient, logical, and optimized solutions.

Chapter 6 – Conclusion

"Technology is there to be used to help you be creative"

– Nicholas Grimshaw [111]

Chapter 6 provides the final conclusions of this project and highlights that the developed design tool is more than just a proof-of-concept. It is, in its current form, a valuable tool for exploring and evaluating various structural layout suggestions. Chapter 6 is divided into three sections:

- Section 6.1 summarizes chapters 1 to 5 and presents the conclusions, including the project's novelty.
- Section 6.2 reflects on and discusses some of the main findings.
- Section 6.3 is dedicated to presenting and discussing future improvements and prospects concerning short- and long-term goals.

6.1 Summary of the thesis

Chapter 1:

The primary goal of this project is to develop a tool that could generate optimized structural layout suggestions using prefabricated reinforced concrete (RC) elements. Chapter 1 outlines the research aims, questions, and objectives to achieve this goal.

The background and motivation for the project are presented, emphasizing the need for more effective use of concrete due to environmental, legal, and economic considerations. The argument is made that better design exploration in the early design phase using the proposed design tool could indeed lead to more effective use of concrete. It is argued that such a tool could support an integrated design process (IDP) approach, which involves earlier and more efficient collaboration between architects, engineers, and contractors. State-of-art is also presented with a focus on research developing structural layout tools and early design tools for building design.

Chapter 2:

Chapter 2 focuses on the fundamental mechanics of the theories and methods applied in constructing the design tool. It introduces the concept of parametric design and relevant approaches to this topic. Additionally, it specifies terminology and definitions of concepts used in this project.

The chapter also introduces the general optimization principle for single-objective and multi-objective optimization. It explains the difference between ideal-based and preference-based multi-objective optimization, arguing that a preference-based approach is most suitable for this project. This approach is implemented in the form of a Pareto front representation.

The characteristics of meta-heuristic algorithms are defined, with a detailed explanation of the mechanisms of a Genetic Algorithm. Machine learning (ML) is also introduced, focusing on how ML models can learn to emulate complex objective functions. This topic is further explored through the concepts behind surrogate modeling, which is a sub-field of ML. The essential concepts behind surrogate modeling are presented, including sampling strategies, error metrics. Different specific surrogate model strategies are explained in detail. The chapter emphasizes that surrogate modeling can be combined with meta-heuristic optimization algorithms to solve complex and computationally expensive engineering problems.

It is noted that not all of the defined methods are incorporated into the final design tool, but they are all considered. The chapter also provides a common terminology used in the subsequent chapters.

Chapter 3:

Chapter 3 introduces the concept of Action Research (AR), including the fundamental theory, and details how it can be used for concept development through a cyclical process of planning, acting, observing, and reflection.

The AR analysis uses three loops for the concept development and employs semi-structured interviews and co-creating workshops where architects, engineers, and contractors contribute to the development of the design tool. The inputs and data collected during these sessions are evaluated based on their potential value and feasibility for implementation. The most relevant and requested features are identified and serve as a guideline for the actual development of the design tool.

Chapter 4:

Chapter 4 presents the final design tool and elaborates on implementing the conceptual framework from Chapter 3. The design tool, also referred to as a design methodology, is built using four core principles: optimization, interactivity, dissemination, and automation. The design tool is envisioned in a larger context of nesting design loops, each with an increasingly associated Level of Detail (LOD). In this project, the developed design loop is intended for the first design loop with a LOD set to 200.

Various relevant programming languages are explored, and their advantages and disadvantages are discussed. Based on a balance between user-friendliness and computing

speed, C# is chosen for this project, while Rhino is selected as the software platform for the design tool.

The design tool's general framework is outlined and divided into two main phases: initialization and optimization. During initialization, user input is applied to calculate and prepare the information needed for the evaluation modules. This phase utilizes the novel method developed for this project named Adjacent Polygon (APoly) representation, which essentially defines how a building's parametric representation can change dynamically.

The evaluation modules are presented, focusing on applying surrogate and hierarchical surrogate modeling approaches to achieve fast approximated responses. Their accuracy is demonstrated using prediction plots and error metrics.

Lastly, the optimization phase is presented with a focus on how user-induced knowledge is incorporated into the optimization framework. This incorporation is achieved using a repair algorithm that operates on the chromosome of each individual to increase the number of valid solutions in each generation.

Chapter 5:

Chapter 5 validates the performance and reliability of the design tool through a series of local and global case studies. The validation studies are divided into two main groups.

The first group consists of parameter sensitivity studies on the structural surrogate evaluation modules, including the wall, beam, column, and beam-column line modules. All sensitivity studies demonstrate a logical response that is consistent with expectations. Therefore, it can be concluded that the modules operate as intended and can be combined in the final design tool, which is explored in the second group of studies.

The second group of validation studies examines the design tool's effectiveness across relevant building plans and scenarios. The corresponding results demonstrate that the tool can adapt to different scenarios and settings and produce optimized results based on the design objective and active constraints.

It is also demonstrated that the tool can produce solutions adapted to different settings of the user-interactivity parameters, which was incorporated based on the AR analysis in Chapter 3.

Furthermore, it is demonstrated that the design methodology can conduct multi-objective optimization and produce excellent results, confirming the tool's ability to generate efficient, logical, and optimized solutions.

6.1.1 Overall conclusions

Based on the results of this project, the following conclusions can be made:

- The AR method was effectively utilized as a framework for developing early design tools, allowing for the identification of weaknesses and opportunities for improvement.
- AR was successfully applied to identify relevant user interactivity and dissemination features, which would aid building design practitioners in real-world practice.
- The proposed APoly representation methodology can produce varied and relevant structural layout suggestions for any given building plan.
- The APoly representation methods allow for user interaction, enhancing the algorithm's purpose as a supportive tool in the design process.
- Surrogate modeling can produce optimized structural RC elements with high accuracy and instant response time.
- Hierarchical surrogate modeling can be applied to more complex issues, such as determining the optimal BC lines while maintaining satisfactory accuracy levels.
- The repair algorithm successfully increased the percentage of valid solutions in each generation, especially when horizontal wind loads dominate the structure.
- The design tool can take into account multiple objectives and produce a Pareto front of valid and optimized solutions.

After conducting four parameter sensitivity studies and eight validation studies, it can be concluded that the design tool can generate valid and efficient solutions optimized with respect to the defined objectives and constraints. The program's ability to perform robustly under different scenarios and conditions further validates its use as an early design tool. Therefore, it can be concluded that the program is suitable for generating structural layout suggestions.

Novelty:

The novelty of this research is present in different aspects:

- The proposed APoly representation method adopts a distinct approach compared to existing methods, offering greater design flexibility that accounts for the placement and dimensioning of slabs, beams, columns, and walls within a single optimization procedure.

- The approach of using hierarchical surrogate modeling to predict the entire optimized geometry of the structural elements, such as the shear walls and beam-column line on finite length, is believed to be a novel approach in the context of structural layout optimization. Previous research has mainly focused on prediction performance metrics such as utilization values and deformations levels and then utilizing meta-heuristic algorithms to determine optimized geometry. This methodology is commonly referred to as surrogate-assisted optimization. Similarly, the design tool utilized in this project also employs surrogate-assisted optimization but at a global level, which defines the overall structural layout.
- The development of the tool was inspired by previous research focusing on HVAC tools [30], which utilized AR. This approach was adapted and structured to suit the current design tool's concept development process, which focused on generating optimized structural layout suggestions.
- The repair algorithm developed for this project aimed to increase the number of valid solutions in each generation and decrease the convergence time. The approach of incorporating user knowledge into the framework of the GA was inspired by prior research [28] [29] done during the developer's Ph.D. program.
- In conclusion, the combination of all applied methods and approaches, whether novel or known, has resulted in a new robust, flexible design tool that can produce valid and optimized solutions for the initial design phase.

6.2 Reflection and discussion

Developing an early design tool to generate optimized structural layouts has required various theories and methods, leading to numerous reflections on the interrelationships between these methods and their impact on the final design. As stated in the introductory quote in Chapter 1, there is no one correct answer but rather many valid approaches that can be used to achieve the same goal. This section will reflect on and discuss some of the strengths and weaknesses of the proposed design methodology, thereby identifying potential areas for improvement.

6.2.1 The importance of parameterization

The research focused on developing PBGD models tends to prioritize performance metrics and the chosen optimization technique over the parametric representation. This preference may be because parametric modeling is challenging to quantify, and there may be several acceptable ways to parameterize the model, depending on the problem being addressed. However, it can be argued that the choice of the parametric representation has a far more significant impact on the final results than the choice of the optimization algorithm, especially for complex problems such as structural layout optimization.

The present design tool is constructed from various principles, such as surrogate modeling, genetic algorithms, and evaluation modules utilizing static theory. However, the parametric representation is by far the most crucial aspect. The proposed APoly representation method is particularly well-suited for structural layout optimization for two reasons. Firstly, relatively few parameters can generate numerous varied and relevant solutions. Secondly, these parameters can be easily manipulated through user interactivity and can even serve as constraints, as demonstrated in the hybrid solution example in section 5.2.7

However, there are still aspects that can be improved. For instance, in some results, a BC line is generated for lines with no load area attached. This issue can be easily resolved by introducing simple “if” rules to the algorithm that removes elements with no structural purpose.

Furthermore, it is necessary to investigate how the parametric representation method can be adapted to use concrete cores. Incorporating cores would require some human input, at least to determine the requested size and quantity. Implementation could occur through a separate optimization iteration before the primary optimization phase, where the cores are placed in the most efficient locations within the building plan to reduce deformation and torsion. The concrete core could either be inserted directly into a structural layout and then replace the overlapping structure. It could, however, also be incorporated into the algorithm that defines the APoly shapes. However, these considerations require tests and examinations

to determine the optimal approach for implementing concrete cores into the design methodology.

6.2.2 Use cases

The primary use case for the tool is to explore different design options during the initial design phase. This exploration can be done using any of the three levels of ML involvement or a combination thereof. The primary use case will typically occur in a competition scenario with limited time to determine and dimension a potential structural layout. The proposed design methodology would have a significant advantage over the current approach, as the user can receive instant feedback on specific configurations when using the "user-defined solution" approach or leverage the capability of ML to explore optimized solutions. Additionally, the user can adjust these solutions as needed. This feature can potentially tailor an optimized solution to fit the functional needs that may be present. Ultimately, this method can generate more accurate solutions than what can be achieved through standard rough estimate calculations.

The tool is built using a modular principle, which means each module can be used in other scenarios, maximizing the project's value creation. For example, if a specific task only requires the determination of an optimized BC geometry, the corresponding BC module can be used to solve this problem. The wind load modules are another excellent example of adding value. Unlike many other modules, the wind modules are not based on an approximation algorithm but rather on high-fidelity calculations and can be used in static documentation.

The tool, in its current state, is highly effective and robust. However, as described in the previous section, many aspects can still be improved and further developed, such as the implementation of concrete cores and additional topics will likely arise as the design tool is used in its entirety.

6.2.3 Automation

Automation is not a typical research topic, but it was a necessity to achieve the limited setup time requested by the interviewed architects, engineers, and contractors. For this reason, automation was defined as one of the four key aspects in Figure 4.1.

As previously mentioned, the primary use case of the tool would be during the initial design phase to explore design options. Fewer resources are typically allocated for this stage, and the design tool has to be tailored to this reality. Cost is one of the primary factors for any building project, so if an early design tool is to be successfully implemented, automation must be incorporated into the tool's overarching framework.

6.2.4 Constructability

The significance of the constructability measure became increasingly apparent throughout the process. During initial testing, it was discovered that the algorithm sometimes placed large slabs on thin walls. While investigations showed that it was statically permissible regarding the wall's load capacity, such a scenario would occur in actual practice as the slab bearing length would be far too small. As a result, a constraint was inserted to prevent this and ensure that the solutions generated were buildable.

Constraints and user-defined rules such as these are essential, as they increase the constructability of the generated solutions. However, constructability also entails other considerations, such as not using stabilizing walls longer than ten meters to avoid overestimating the building's to absorb the overturning moment and setting minimum height as a constraint instead of a constant input value. These considerations may seem obvious, but they are often only highlighted through conversations with relevant stakeholders or through actual use.

Implementing more functional constructability constraints, such as minimum thickness regarding acoustic requirements, could also be relevant. The tool's flexibility could also be enhanced if these requirements were interactive, allowing the user to enable or disable them in the same way the user can add or remove structural elements from the database the algorithm uses for evaluation.

6.2.5 AR as a structured approach for concept development

Research methods from the social sciences are not often used in structural engineering research projects that focus on creating early design tools. This occurrence happens because the research tends to be more theoretical, with a greater focus on constructing a proof-of-concept than creating a tool that conforms to everyday design practice.

AR provided this project with a platform where discussions, inputs, and reflections from different stakeholders helped develop a more refined tool than its original starting point. Including at least two AR loops is recommended, as this allows participants to see that their input is being incorporated into the solution. This approach can form the basis for a more nuanced discussion, resulting in a more developed and relevant final product.

It should be noted that an AR analysis requires a significant amount of time and resources to complete. However, this expense is greatly outweighed by its improvements to the design tool's framework.

6.2.6 Evaluation response time

Creating a working design tool by attaching a FEM engine as an evaluation engine would have been easier and faster to implement. However, FEM calculations are ill-suited for running

numerous evaluation iterations to achieve an optimized result. It also limits the tool's use, as it depends on a FEM license. Consequently, it was chosen to code methods for defining load cases, performing evaluations, and distributing horizontal forces. Although this required significant development time, the outcome was faster than if a FEM model were implemented, even if the FEM model was only used to determine the load on the structural elements.

Another crucial aspect of the evaluation speed was pre-calculating as much information as possible before the optimization process. For example, the optimal choice of the slab was found for each APoly shape, for each possible load direction, and for each slab length, as described in section 4.3.2. This strategy was possible because the choice of the slab was not affected by external wind loads. This information was stored in the APoly C# class created so the algorithm could access this database for each iteration.

Another, and even more important feature for the evaluation speed, was the use of surrogate models. This principle allowed for instant dimensioning of the structural elements, so the primary time used in a single evaluation was allocated for executing the parameters into a structural layout and calculating and distributing the external forces that affect the structural elements.

6.2.7 Design objectives

The primary CPM2 design objective showed a clear tendency that it was generally more cost-effective to use multiple but smaller elements, at least to a certain limit. However, other factors may also contribute to the total cost, such as transportation and the number of work processes, as mentioned in section 4.2.1. Therefore, if a more precise cost calculation is desired, these factors must be incorporated, requiring a more detailed understanding of the different cost factors. The RC suppliers who were approached in this study were not willing to disclose how they assess the total cost, as it is considered a trade secret. Other suppliers may be more open to collaboration, and contractors may contribute to further refining the cost objective.

It may also be relevant to test other design objectives, such as minimizing the number of turning elements or incorporating more functional constraints, but only if they add value to the design process.

6.2.8 User interaction and dissemination

User interaction was a significant priority in the project, as it was crucial that the tool supported the creative design process rather than dictating it, as previously emphasized. This priority was also shared by the co-creation workshop participants, along with clear dissemination of the final results. Interactivity and dissemination were included as part of the four core principles for the design tool for these reasons.

The implemented interactivity features enhance the user's ability to influence the final design. For example, three measures have been implemented to reduce the number of elements: the CPF value, the ability to reduce the number of available elements, and the ACS objective. The effect of all these measures has been demonstrated in the case studies presented in Chapter 5. Using a factor like CPF can also be extended to other structural elements, which may be more advantageous than removing options from the algorithm.

Generally, user interaction and dissemination features are expected to be improved continuously as users tend to report on possible areas for improvement. Early alpha testing has already resulted in several relevant requests for new features or adjustments to existing ones. This entire process could also be more systematized by using relevant theories from the social sciences.

6.3 Future work

The design methodology has shown promising results, providing a robust foundation for further development. This section elaborates on some aspects that can be improved and further developed; the subjects are divided into short-term and long-term goals. Some of these planned features can relatively easily be implemented, while others are in the preliminary stage and only discussed in theory.

There is great potential in using early design tools. However, development has numerous challenges, especially if one aims to create a generalized tool for real-world practice. The design limitations listed in section 1.3.1 will also be addressed in future versions. There are numerous directions that a further development process can take, which will be explored in the following subsections.

6.3.1 Short-term goals

The process of finding the different APoly shapes is currently the bottleneck in the design methodology due to the large number of possibilities that the algorithm has to check using the powerset function, as shown in equation (4.4). This process is non-recurrent, and when completed, the APoly shapes can be generated and used in a fast optimization process. The building shapes applied in Chapter 5 were simple enough, but due to the nature of the problem, this can quickly become an issue for more complex building shapes. A mockup expert algorithm has been formulated as a potential alternative to the powerset function. However, unlike using the powerset function, this function will not guarantee that every possible combination is found. Alternatively, users can define the APoly shapes themselves, as the number of basis shapes the default algorithm generates are typically manageable.

Generalizing a design tool is associated with significant challenges, mainly because there are so many different scenarios in terms of geometry, loads, and evaluations that need to be considered. It was defined as a goal that the design tool should be able to handle 80 percent of all use cases. It is acknowledged that the tool is not there yet. One of the most important future features is that the tool should be able to handle non-regular plan elevations, and also, by expanding the library of structural elements, the design tool can use.

With the inclusion of these features and the capacity to incorporate concrete cores, it can be argued that the design tool is equipped to handle 80 percent of all use cases. However, implementing these features also imposes other challenges that must be addressed. For example, including non-regular plan elevations may result in snow accumulation to such an extent that it becomes the dominating factor for the design of elements such as the slabs. Implementing snow accumulation can be resource-demanding if it is to be as automated as the current wind load calculations.

As previously mentioned, it is planned to implement LCA as a design objective, which requires estimating the total reinforcement quantity. This implementation could be achieved by adding another SU model in a series after the SU models that predicts the final geometry. There is expected to be a high correlation between the reinforcement quantity and the cross-sectional area of an RC element. Using the geometry value and all the other known external parameters should result in a SU model that can accurately predict the amount of reinforcement, assuming a constant reinforcement quality.

There is also expected to be a strong correlation between the CPM2 and LCA objectives. Therefore, it is speculated whether the LCA objective is better suited as a constraint. For example, this constraint could be implemented by ensuring that the embodied energy is less than the legal requirement. However, this would require defining the proportion of embodied energy typically contributing to the overall carbon footprint. Literature analysis and interviews with experts in this field would help to determine how such a requirement could best be formulated and implemented.

It is planned to ensure vibrations comfort induced by rhythmic person loads by calculating the limit acceleration values based on the method defined in [112]. These values will then be compared against the requirements defined in EC1 [113] instead of using a minimum eigenfrequency threshold value. This change will result in a more accurate representation of the comfort functionality, which can be easily integrated into the SU model.

Various measures are planned to increase the evaluation speed. For instance, the selection and mutation operators were selected based on experience, but a performance study is

necessary to identify the optimal operators and settings for this specific problem. Other relatively simple measures can also be implemented, such as using a stopping criterion instead of letting the optimization run for a finite number of generations. Bottlenecks in the code can also be identified and replaced with more efficient methods.

6.3.2 Long-term goals

The design loop was initially conceived as part of a larger context of nested design loops, with each subsequent loop increasing the LOD and calculation fidelity. This progression is achieved by fixing a set of parameters at the end of each design loop and then releasing other parameters for modification in the next loop. The visualization of this approach is depicted in Figure 4.2. This project has solely focused on the implementation of design loop 1.

Future efforts will be focused on implementing design loop 2, where the LOD will increase to 300. This implementation will include recesses into the geometry and estimating the overall geometry at a level that will not change for the rest of the design process.

Another feature planned for design loop 2 is developing a plastic redistribution method of the horizontal forces inspired by the method presented in [9]. This method designates a relative stiffness to every shear wall but only changes in very rough intervals and uses few parameters. The schematics of a potentially better method have been drafted. This method would consider load values, height, local and global geometry, and other parameters to determine the relative stiffness of the shear walls. The function itself would be calibrated using a meta-heuristic algorithm. The algorithm would require thousands of training examples which could be provided by the design tool in this project. The objective would be to minimize the highest utilization value of the shear walls. This proposed method is still vaguely defined and would require much time and resources. However, an improved distribution method could provide considerable value, and its performance could be compared with the method defined in [9] and other relevant methods found in the literature.

The design methodology is not limited to prefabricated RC elements, and it can be adapted for other structural typologies such as steel and wood. However, this would require a more extensive analysis of how to logically define and compare the cost of different material types. Additionally, numerous expert rules would likely be introduced to ensure constructability, as joints across typologies are not always practical.

In terms of implementation, a hierarchical surrogate approach could be used, where initial models predict the most suitable structural material for a given location and then activate another surrogate model that dimensions the specific structural element. Furthermore, increased use of timber elements would necessitate rethinking the load distribution, as timber does not possess the same plastic properties as reinforced concrete and steel.

Reference

Bibliografi

[1] P. Debney, Computational Engineering, London: The Institution of Structural Engineers, 2020.

[2] C. Ionescu, T. Baracu, G.-E. Vlad og H. Necula, »The historical evolution of the energy efficient buildings,« *Renewable and Sustainable Energy Reviews,* 2015.

[3] UCL Engineering, »ucl.ac.uk,« 2014. [Online]. Available: https://www.ucl.ac.uk/engineering-exchange/sites/engineering-exchange/files/fact-sheet-embodied-carbon-social-housing.pdf.

[4] NG ZINK, »ng-zink.dk,« 2022. [Online]. Available: https://ng-zink.dk/artikler/byggeindustrien-er-en-gigantisk-byrde-klimaet.

[5] »danish housing and planning authority,« 19 October 2022. [Online]. Available: https://bpst.dk/da/Byggeri/Baeredygtigt-byggeri/NY-Klimakrav-i-bygningsreglementet#.

[6] N. Nain, R. Surabhi, N. Yathish, V. Krishnamaurthy, T. Deppa og S. Tharannum, »Enhancement in strength parameters of concrete by application of,« *Construction and building materials,* nr. 202, pp. 904-908, 2019.

[7] N. Z. Muhammad, A. Shafaghat, A. Keyvanfar, M. Z. A. Majid, S. Ghishal, S. E. M. Yasouj, A. A. Ganiyu, M. S. Kouchaksaraei, H. Kamyab, M. M. Taheri og Shirdar, »Tests and methods of evaluating the self-healing efficiency of concrete: A,« *Construction and Building Materials,* nr. 112, pp. 1123-1132, 2016.

[8] Betonelement-foreningen , »betonportal,« 2003 . [Online]. Available: http://www.betonportal.dk/index.htm.

[9] J. F. Jensen, Beton element byggeriers statik, 1. red., Polyteknisk forlag, 2010.

[10] F. Belaïd, »How does concrete and cement industry transformation contribute to mitigating climate change challenges?,« *Resources, Conservation and Recycling Advances,* november 2022.

[11] D. Mathiesen og J. S. Hansen, »Branchevejledning - Designoptimering af betonkonstruktioner,« Bæredygtig Beton initiativet, 2022.

[12] T. Hamidavi, S. Abrishami og R. M. Hosseini, »Towards intelligent structural design of buildings: A BIM-based solution,« *Journal af building engineering,* årg. 32, 2020.

[13] C. T. Mueller, »Computational Exploration of the Structural Design Space,« Boston, 2014.

[14] L. Wang, W. Shen, H. Xie, J. Neelamkavil, Pardasani og Ajit, »Collaborative coneptual design - state of the art and future trends,« *Computer-Aided Design*, årg. 34, nr. 13, pp. 981-996, 2002.

[15] A. J. MacDonald, Structure and Architecture, 3. red., Routledge, 2019.

[16] I. J. Palormar, J. L. G. Valldecabres, P. Tzortzopoulos og E. Pellicer, »An online platform to unify and synchronise heritage architecture information,« *Automation in Construction,* årg. 110, February 2020.

[17] R. Deutch, BIM and Integrated Design: Strategies for Architectural Practice, Wiley, 2011.

[18] B. C. Paulson, »Designing to reduce Construction Cost,« *Journal of the Construction Division,* December 1976.

[19] D. Davis, »Modelled on Software Engineering: Flexible Parametric Models in,« 2013.

[20] Y. Reich, »Machine Learning Techniques for Civil Engineering Problems,« *Microcomputers in Civil Engineering,* pp. 295-310, 1997.

[21] M. Ball, »informed infrastructure,« 7 June 2015. [Online]. Available: https://informedinfrastructure.com/15197/change-leader-interview-bim-aids-process-but-further-promise-lies-in-interoperability/.

[22] I. Caetano, L. Santos og A. Leitao, »Computational Design in architecture: Defining parametric, generative, and algorithmic design,« *Frontiers of Architectural Research,* 2019.

[23] A. Beim, L. B. Jensen, P. A. Sattrup og K. Negendahl, Informing sustainable architecture: Introduction, Copenhagen: Polyteknisk forlag, 2018.

[24] M. Ramage, »Trimble,« 21 April 2022. [Online]. Available: https://constructible.trimble.com/construction-industry/what-is-computational-design.

[25] M. Dimcic, »Artificial intelligence in architecture,« *Frontiers of Science and Technology,* 2017.

[26] D. Davis, »danieldavis,« 20 february 2020. [Online]. Available: https://www.danieldavis.com/generative-design-doomed-to-fail/.

[27] R. Tara, »engineering,« 11 january 2022. [Online]. Available: https://www.engineering.com/story/an-engineers-answer-to-generative-design.

[28] L. W. Rahbek, P. H. Kirkegaard og U. Alibrandi, »Parametric grid mapping design tool for freeform surfaces using a,« *Proceedings of the IASS Annual Symposium 2020/21 and the 7th International Conference on Spatial Structures Inspiring the Next Generation,* 2021.

[29] L. W. Rahbek, C. R. Terp, U. Alibrandi og P. H. Kirkegaard, »Stock optimization of naturally curved wood logs on freeform,« *Proceedings of the IASS 2022 Symposium affiliated with APCS 2022 conference,* September 2022.

[30] P. B. Purub og S. Petersen, »Research framework for development of building performance simulation tools for early design stages,« *Automation in Construction,* årg. 109, 2020.

[31] A. Retik og A. Warszawski, »Automated Design of Prefabricated Building,« *Building and Environment,* årg. 29, nr. 4, 1994.

[32] M. Jinjie, S. Qingxuan og H. Zhijian, »Optimal Design of Tall Residential Building with RC Shear Wall and with rectangular Layout,« *International Journal of High-Rise Buildings,* årg. 3, nr. 4, 2014.

[33] P. Zhao, W. Liao og X. Lu, »Intelligent design method for beam and slab of shear wall structure based on deep learning,« *Journal of Building Engineering,* årg. 57, 17 June 2022.

[34] S. Talatahari og M. Rabiei, »Shear wall layout optimization of tall buildings using,« *Frontiers of Civil and Structural Engineering,* pp. 1131-1151, 2020.

[35] T. Takada, Y. Kohama og A. Miyamura, »Optimization of shear wall allocation in 3D frames by branch-and-bound method,« *Transactions on the Built Environment,* årg. 52, 2001.

[36] Y. Zhang og C. Mueller, »Shear wall layout optimization for conceptual design of tall buildings,« *Engineering Structures,* pp. 225-240, 22 February 2017.

[37] H. Lou, B. Gao, F. Jin, Y. Wan og Y. Wang, »Shear wall layout optimization strategy for high-rise buildings based on conceptual design and data-driven tabu search,« *Computers and Structures,* 2021.

[38] L. Wessel, »ing.dk,« 2018. [Online]. Available: https://ing.dk/artikel/ny-prognose-ingenioermanglen-er-langtfra-afblaest-0.

[39] B. C. Jensen og S. O. Hansen, Bygningsberegninger, 1. red., Valby: Nyt Teknisk Forlag, 2010.

[40] P. McCullag og J. Nelder, Generalized Linear Models, 2. red., New York: Routledge, 1983, p. 532.

[41] J. E. Harding og P. Shepard, »Meta-Parametric Design,« *Design Studies,* pp. 73-95, 2017.

[42] techtipnow, »techtipnow,« 2023. [Online]. Available: https://techtipnow.in/encoding-scheme-definition-and-types/.

[43] Robert McNeel & associates, »Rhino3d,« 2023. [Online]. Available: https://www.rhino3d.com/.

[44] Autodesk, »autodesk.com,« 2023. [Online]. Available: https://www.autodesk.com/products/dynamo-studio/overview.

[45] SolidWorks, »solidworks.com,« 2023. [Online]. Available: https://www.solidworks.com/partner-product/driveworks-pro.

[46] S. Sivanandam og S. Deepa, Introduction to Genetic Algorithms, 2008.

[47] A. Dutta, »geeksforgeeks.org,« 2023. [Online]. Available: https://www.geeksforgeeks.org/encoding-methods-in-genetic-algorithm/.

[48] A. I. J. S. A. K. A. J. Forrester, Engineering Design via Surrogate Modelling A Practical Guide, 1. red., Chichester, West Sussex: John Wiley and Sons Ltd., 2008, p. 238.

[49] A. I. Forrester og A. J. Keane, »Recent advances in surrogate-based optimizaiton,« *Progress in Aerospace Sciences 45,* pp. 50-79, 10 January 2009.

[50] F. A. C. Viana, C. Gogu og R. T. Haftka, »Making the most out of surrogate models: tricks of the trade,« *Proceedings of the ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDET/CIE 2010,* august 2010.

[51] J. Brownlee, Clever Algorithms - Natur-Inspired Programming Recipes, 1. red., 2011.

[52] S. S. Rao, Enginering Optimization - Theory and Practice, 5. red., Hoboken, New Jersey: John Wiley and Sons, Inc., 2019.

[53] N. Gu og P. a. Behbahani, »Shape Grammars: A key Generative Design Algorithm,« i *Handbook of the Mathematics of the Arts and Sciences*, Springer Nature Switzerland AG 2021, 2021.

[54] H. Salehi og R. Burgueno, »Emerging artificial intelligence methods instructural engineering,« *Engineering Structures,* pp. 170-189, May 2018.

[55] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Chichester: John Wiley and Sons, Ltd, 2001.

[56] C. Blum og A. Roli, »Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison,« *ACM Computing Surveys,* pp. 268-308, January 2001.

[57] K. Sørensen, »Metaheuristics - the metaphor exposed,« *International Transactions in Operational Research,* årg. 1, pp. 3-18, 2013.

[58] E. Zitzler, M. Laumanns og L. Thiele, »SPEA2: Improving the Strength Pareto Evolutionary Algorithm,« Computer Engineering and Networks Laboratory (TIK), Zurich, 2001.

[59] K. Deb, A. Pratap, S. Agarwal og T. Meyarivan, »A fast and elitist multiobjective genetic algorithm: NSGA-II,« *IEEE Transactions on Evolutionary Computation,* årg. 6, nr. 2, pp. 182-197, April 2002.

[60] J. Bader og E. Zitzler, »HypE: An algorithm for Fast Hypervolume-Based Many-Objective Optimization,« *Evolutionary Computation,* årg. 1, pp. 45-76, 2011.

[61] C. Darwin, On the Origin of Species, Dover Publication, 2006.

[62] J. H. Holland, Adaptation in Natural and Artificial Systems, Michigan Press, 1975.

[63] A. E. Eiben og J. Smith, Introduction to Evolutionary Computing, 2. red., Leiden: Leiden Center for Natural Computing, 2017.

[64] S. Adriasenssen, P. Block, D. Veenedaal og C. Williams, Shell Structures for Architecture - Form Finding and Optimization, Taylor and Francis Ltd, 2014.

[65] L. W. Rahbek og M. K. Feld, »Parametric Grid Mapping Design Tool for Freeform Surfaces,« 2015.

[66] Y. Kaya, M. Uyar og R. Tekin, »A Novel Crossover Operator for Genetic Algorithms: Ring Crossover,« *DBLP,* January 2011.

[67] K. Deb og R. B. Agrawal, »Simulated Binary Crossover for Continous Search Space,« *Complex Systems,* pp. 115-148, 1995.

[68] L. D. Whitley, T. Starkweather og D. Fuquay, »Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator,« *Proceedings of the 3rd International Conference on Genetic Algorithms,* pp. 133-144, June 1989.

[69] C. Wang, Q. Duan, W. Gong, A. Ye, Z. Di og C. Miao, »An evaluation of adaptive surrogate modeling based optimization with two benchmark problems,« *Environmental Modelling and Software,* årg. 60, pp. 167-179, 2014.

[70] D. R. Jones, »A Taxonomy of GLobal Optimization Methods Based on Response Surfaces,« *Journal of Global Optimization,* nr. 21, pp. 345-383, 2001.

[71] J. N. Fugh, A. Fau og U. Nackenhorst, »State-of-the-Art and Comparative Review of Adaptive Sampling Methods for Kriging,« *Archives of Computational Methods in Engineering,* 25 July 2020.

[72] K. Crombecq, L. De Tommasi, D. Gorissen og T. Dhaene, »A novel sequential design strategy for global surrogate modeling,« *Proceedings of the 2009 Winter Simulation Conference (WSC)*, 2009.

[73] J. N. Fuhg, »Adaptive surrogate models for parametric studies,« Hannover, 2019.

[74] P. Singh, D. Deschrijver og T. Dhaene, »A balanced Sequential deisgn strategy for global surrogate modeling,« *Proceedings of the 2013 Winter Simulation Conference*, 2013 .

[75] K. Crombecq, E. Laermans og D. T., »Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling,« *European Journal of Operational Research*, pp. 683-696, 30 May 2011.

[76] S. Tseranidis, N. C. Brown og C. T. Mueller, »Data-driven approximation algorithms for rapid performance evaluation and optimization of civil structures,« *Automation in Construction*, 6 February 2016.

[77] S. Tseranidis, »Approximation Algorithms for Rapid Evaluation and Optimization of Architectural and Civil Structures,« Thessaloniki, 2015.

[78] A. Botchkarev, »Performance Metrics (Error Measure) in Machine Learning Regression, Forecasting and Prognostic: Properties and Typology,« *Interdisciplinary Journal of Information, Knowledge, and Management*, årg. 14, pp. 45-76, 2018.

[79] D. Krige, »A Statistical Approaches to Some Basic Mine Valuation Problems on the Witwatersrand,« *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, årg. 52, pp. 119-139, 1951.

[80] J. P. Kleijnen, »Kriging metamodeling in simulation: A review,« *European Journal of Operational Research*, pp. 707-716, 17 October 2009.

[81] G. Dreyfus, Neural Networks - Methodology and Applications, 2005.

[82] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, Ozair, Sherjil, A. Courville og Y. Bengio, »Generative Adversarial Networks,« *arXiv e-prints*, June 2014.

[83] I. Goodfellow, Y. Bengio og A. Courville, Deep Learning, MIT Press, 2016.

[84] I. Goodfellow, Y. Bengio og A. Courville, Deep Learning, MIT Press, 2016.

[85] J. Feng og S. Lu, »Performance Analysis of Various Activation Functions in Artificial Neural Networks,« *Journal of Physics: Conference Series*, 2019.

[86] MathWorks, »mathworks.com,« 2023. [Online]. Available: https://se.mathworks.com/discovery/neural-network.html.

[87] G. Sanderson, »3blue1brown,« October 2017. [Online]. Available: https://www.3blue1brown.com/topics/neural-networks.

[88] D. Coghlan og R. Holian, »Editorial: insider action research,« *Action Research,* årg. 5, pp. 5-10, 1 March 2007.

[89] S. Brinkmann og L. Tanggaard, Kvalitative metoder: en grundbog, Hans Reitzel, 2013.

[90] D. Greenwood, »Pragmatic Action Research,« *International Journal of Action Research,* årg. 3, nr. 1+2, pp. 131-148, 2007.

[91] P. C. S. Taylor og M. N. Medina, »Educational research paradigms: From positivism to pluralism.,« *Colleage Research Journal,* årg. 1, nr. 1, pp. 1-16, 2011.

[92] V. Sonne-Ragans, Anvendt videnskabsteori - reflekteret teoribrug i videnskabelige opgaver, 2. red., 2019, p. 300.

[93] S. Kemmis og R. Mctaggart, »Participatory Action Research: Communicative Action and the Public Sphere,« i *Handbook of qualitative research* , Sage Publishing, 2017.

[94] S. Kvale, Doing Interviews, London: Sage: Publications, 2007.

[95] S. Chen, »Chapter 16: Interviews and focus groups,« i *Research Methods in Physical Education and Youth Sport*, K. Armour og D. Macdonald, Red., Routledge, 2012, p. 376.

[96] V. Braun og V. Clarke, »Using thematic analysis in psychology,« *Qualitative Research in Psychology,* årg. 3, nr. 2, pp. 77-101, 2008.

[97] L. Crispin og J. Gregory, Agile Testing - A Practical Guide for Testers and Agile Teams, 1. red., Boston: Pearson Education, Inc, 2009, p. 577.

[98] wired, »wired,« 1997. [Online]. Available: https://www.wired.com/1997/07/chairman-jobs-the-guessing-goes-on/.

[99] DiKon, »DiKon - Digital Konvergens,« 2021. [Online]. Available: https://www.dikon.info/publikationer-2/.

[100] Microsoft, »ML.NET,« 2023. [Online]. Available: https://dotnet.microsoft.com/en-us/apps/machinelearning-ai/ml-dotnet.

[101] Microsoft, »NimbusML,« [Online]. Available: https://github.com/microsoft/NimbusML.

[102] Google, »Google Maps Platform,« [Online]. Available: https://developers.google.com/maps.

[103] MathWorks, »Matlab,« 2023. [Online]. Available: https://se.mathworks.com/products/matlab.html.

[104] A. Torabi, »parametriczoo,« 16 May 2021. [Online]. Available: https://www.parametriczoo.com/index.php/products/grasshopper-ui/.

[105] B. C. Jensen, Betonkonstruktioner, 1. red., København: Nyt Teknisk Forlag, 2008.

[106] European Committee for Standardization, »Eurocode 1: Last på bærende konstruktioner - Del 1-4: Generelle laster – Vindlast,« 2007.

[107] Energi styrelsen, »DS/EN 1991-1-4 DK NA:2015 - Eurocode 1-4,« 2015.

[108] D. Arditi, A. Elhassan og Y. C. Toklu, »Constructability Analysis in the Design Firm,« *Journal of Construction Engineering and Management*, årg. 128, nr. 2, April 2002.

[109] European Committee for Standardization, »Eurocode 0: Projekteringsgrundlag for bærende konstruktioner,« 2007.

[110] D. W. Thompson, On Growth and Form, Cambridge University Press, 1917.

[111] J. Tusa, On Creativity: Interviews Exploring the Process, London: Methuen Publishing Ltd, 2004.

[112] J. S. Knudsen, N. Grathwol og S. O. Hansen, »Vibrational Response of Structures Exposed to Human-induced Loads,« *Dynamics of Civil Structures*, årg. 2, pp. 151-158, 2017.

[113] European Committee for Standardization, »EN 1991-1-1 DK NA:2013, Nationalt anneks til Eurocode 1: Last på bygværker - Del 1 - 1: Generelle laster - Densiteter, egenlast og nyttelast for bygninger«.

[114] Rahbek, Bare en test, Aarhus: Egmont , 2014.

[115] P. Foraboschi, M. Mercanzin og D. Trabucco, »Sustainable structural design of tall buildings based on,« *Energy and Buildings*, pp. 254-269, 2 September 2013.

[116] M. Imbabi, C. Carrigan og S. McKenna, »Trends and developments in green cement and concrete technology,« *International Journal of Sustainable Built Environment*, årg. 1., pp. 194-216, 2012.

[117] A. D. Eslamlou og S. Huang, »Artificial-Neural-Network-Based Surrogate Models for Structural Health Monitoring of Civil Structures: A Literature Review,« *Buildings*, November 2022.

Figure list

Table list

Appendix A

The following table lists the material prices used in the SU models. It should be noted that some of the values averaged based on the data available from suppliers. All prices are converted to cubic meters.

Material	DKK / m³
C35 Concrete	1792
C40 Concrete	1891
C45 Concrete	1990
B550B (Y) reinforcement	42547
Prestressed reinforcement	45547

Appendix B

The applied settings used for the different neural network models are listed as follows:

KB – Price prediction

Setting type	Setting value
Training function	trainlm
Transfer function	logsig
No of layers	3
No of neurons	12
No of training samples	2500

KB – Height prediction

Setting type	Setting value
Training function	trainlm
Transfer function	satlins
No of layers	4
No of neurons	6
No of training samples	2500

RC Column – Price prediction

Setting type	Setting value
Training function	trainlm
Transfer function	logsig
No of layers	3
No of neurons	10
No of training samples	2500

RC Column – Width prediction

Setting type	Setting value
Training function	trainlm
Transfer function	tansig
No of layers	2
No of neurons	10
No of training samples	2500

BC Column – Optimized column spacing prediction

Setting type	Setting value
Training function	trainlm
Transfer function	logsig
No of layers	3
No of neurons	9
No of training samples	3000

Inner RC wall – Price prediction

Setting type	Setting value
Training function	trainlm
Transfer function	logsig
No of layers	3
No of neurons	14
No of training samples	2461

Inner RC wall – Thickness prediction

Setting type	Setting value
Training function	trainlm
Transfer function	tansig
No of layers	3
No of neurons	7
No of training samples	2461

Inner RC wall – Validity state prediction

Setting type	Setting value
Training function	traincgf
Transfer function	satlins
No of layers	3
No of neurons	42
No of training samples	2461

Outer RC wall – Price prediction

Setting type	Setting value
Training function	trainlm
Transfer function	logsig
No of layers	2
No of neurons	18
No of training samples	2338

Outer RC wall – Thickness prediction

Setting type	Setting value
Training function	trainlm
Transfer function	logsig
No of layers	4
No of neurons	9
No of training samples	2338

Outer RC wall – Validity state prediction

Setting type	Setting value
Training function	trainbfg
Transfer function	tansig
No of layers	3
No of neurons	18
No of training samples	8000

Appendix C

The following radar images was used to determine the terrain category used in the validation study in section 5.2.6.

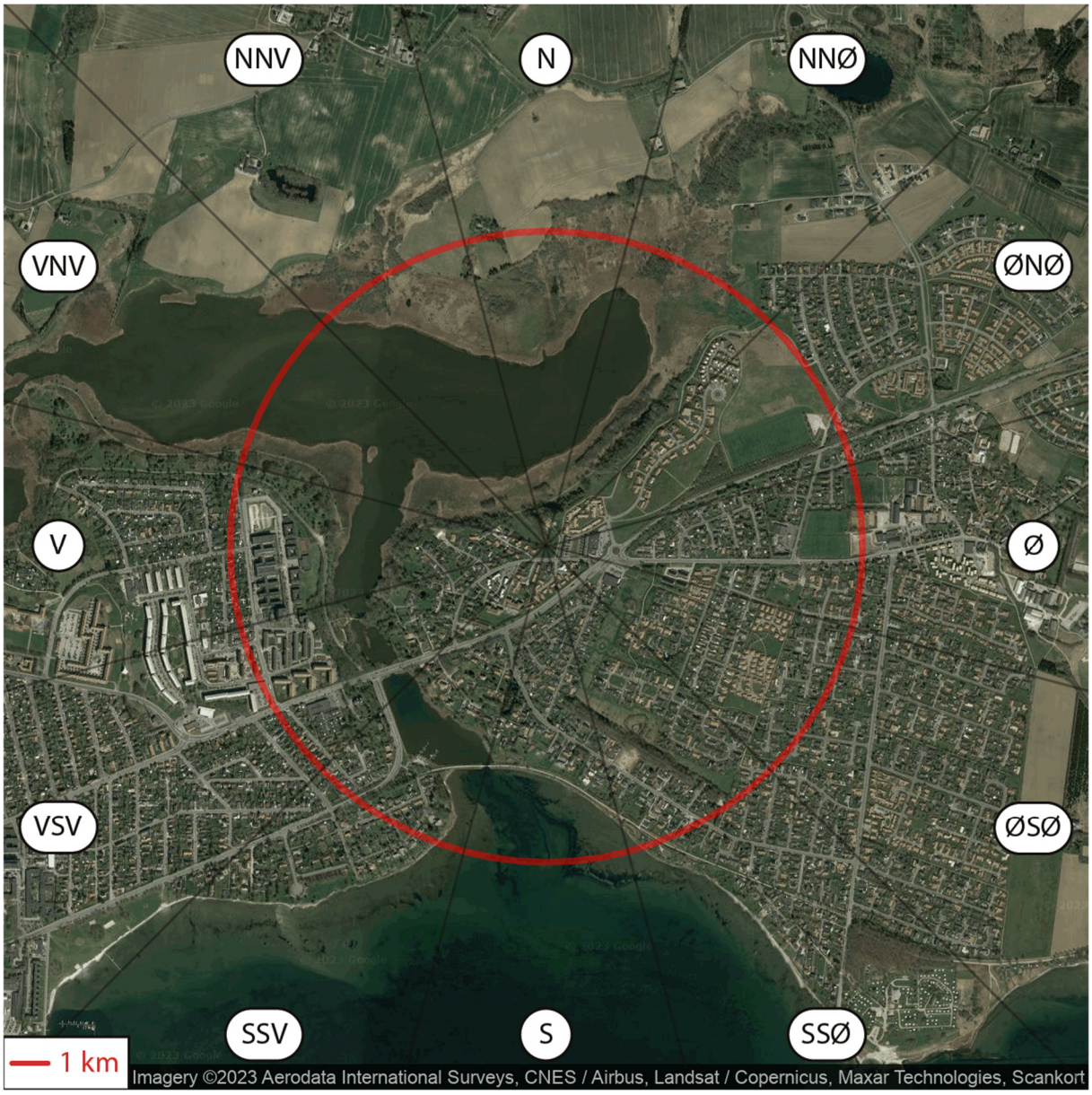


Figure 1 – Radar image at location: 55.872615200208635, 9.900330266895853

Figure 2 – Radar image at location: 56.127254302136436, 10.2121679787756