Master's thesis

2022

Anders Fjetland Fredriksen, Fridtjof Høyer

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Anders Fjetland Fredriksen
Fridtjof Høyer

Towards a generative simulation-based tool for creating diverse building layouts optimized for energy efficiency

June 2022

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Towards a generative simulation-based tool for creating diverse building layouts optimized for energy efficiency

Anders Fjetland Fredriksen
Fridtjof Høyer

Engineering and ICT, Mechanical engineering
Submission date: June 2022
Supervisor: Andrei Lobov

Norwegian University of Science and Technology
Department of Mechanical and Industrial Engineering

Abstract

The construction industry is accountable for 17.5% of global emissions each year. Therefore the need to reduce these emissions is of great significance to the industry. Research has shown that changing different design elements of a building can influence the overall energy consumption. By parameterizing these elements, optimization models, such as Multi-objective genetic algorithms (MOGAs), can be used to optimize the buildings to a set of evaluation functions. In the case of green buildings, the evaluation function for the optimization model should be energy consumption. This thesis proposes a generative simulation-based tool that manipulates parameterized geometry in order to create diverse building layouts optimized for energy efficiency.

Three different design approaches were created for this study. The first is a floor plan approach optimizing interior and exterior factors. The last two approaches test different common building shapes to find the most optimal shape. The Honeybee application is used for transforming geometry into BIM elements. EnergyPlus is used as the environmental simulation for calculating the energy consumption of the building. Lastly, this thesis uses the MOGA hypervolume Estimation (HypE) as the optimization model.

The evaluation functions used in the study were to maximize sun exposure and minimize energy consumption. One optimized result was selected from each design approach and compared with a proposed architecture by *HAV eindom* on the same plot. All selected results from this thesis scored better than the proposed architecture on sun exposure and energy consumption.

There has not been sufficient research on optimizing parameterized exterior and interior factors, focusing on the building layout. This study combines these factors and creates a foundation for generative simulation-based tools opting for sustainable buildings.

Preface

Throughout the writing of this thesis, we have received a great deal of support and assistance.

We would first like to thank our supervisor, professor Andrei Lobov. Andrei has been our professor and supervisor the last three years. His insight and feedback have helped us understand and master the field of knowledge-based engineering (KBE), grasping many new challenges such as the one presented in this thesis.

We would also like to thank Multiconsult for their collaboration on our thesis. We would especially like to acknowledge Torkild Alstad and Øystein Meljænder-Larsen for feedback and guidance regarding the study and implementations.

Lastly, we would like to thank our family and friends. They have motivated us throughout our journey at NTNU and encouraged us when nothing seemed to work.

Contents

List of Figures

1 Introduction

1.1 Research Context

As the world demands more energy to fuel our increasing population, the need to transform one of the most polluting industries we have is becoming increasingly important. The construction industry account for 17.5% of global CO_2 emissions on a yearly basis [Hannah Ritchie and Rosado, 2020]. Figure 1.1 depicts the emission during a building's life cycle, where the percentage at the bottom shows how much pollution each phase contributes. Looking at the figure, 43% of all emissions produced during the building life cycle are created during the use phase. Minimizing the factors contributing to the energy use in this phase is crucial to minimizing the building industry's environmental impact.

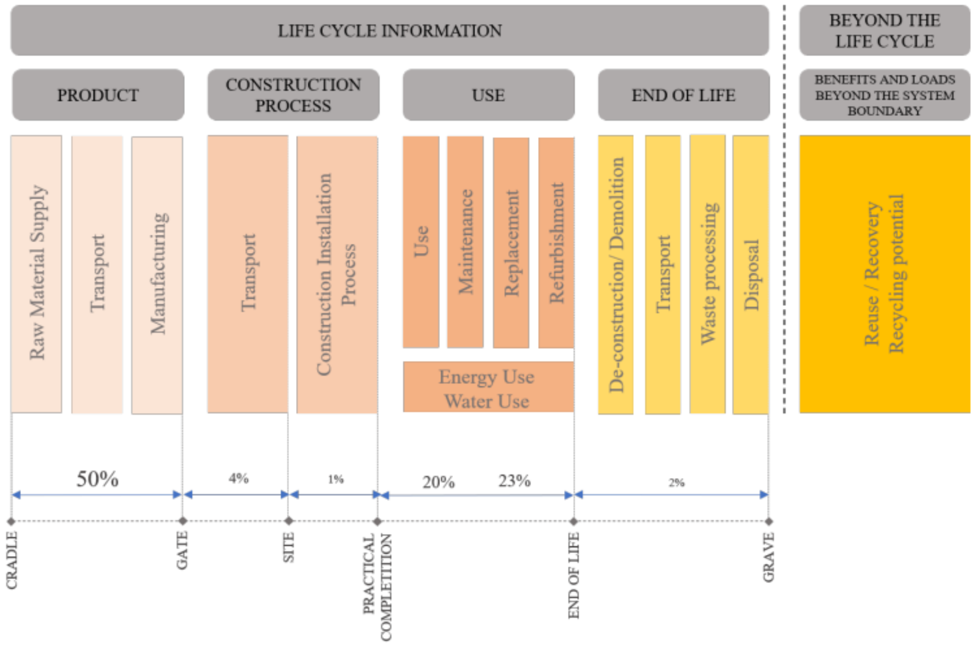


Figure 1.1: Overview of CO_2 emission in building projects.
[Felicita, 2021] as cited in [John Orr, 2020]

The construction of sustainable buildings can be expensive and time-consuming. As developers need to make a profit, using new tools and techniques is needed to keep costs down. Furthermore, the choices made in the early stages of building development affect the end result at a larger scale than decisions made in later stages [Ampanavos et al., 2022]. Therefore, the need to quickly create a foundation focusing on sustainable objectives could save both time and resources and minimize pollution.

As several antagonistic objectives are weighted during building design, the task can be both time-consuming and labor-intensive. The use of parameterized models and optimization methods have therefore become increasingly popular tools for building design [Evins, 2013].

1.2 Problem Definition

17.5% of global pollution is caused by the construction industry, whereas 43% is during the use phase of the building [Sawin et al., 2016]. Choices made in early design stages, such as building layout, material type, and window location, can reduce emissions, creating more sustainable buildings. In addition, utilizing parameterized models driven by optimization techniques can be implemented in the early stages to automate this development, opting to minimize energy consumption.

1.3 Thesis Objective

The main objective of this thesis is to develop a working solution for early-stage building design, focusing on sustainable goals and automation, minimizing what the users need to input. The solution should be able to manipulate geometry using complex simulations to obtain the optimal design. The corresponding solution should be viewed as a foundation for further work, enabling the user to quickly create a model for a building project. This will save time and resources early in the project and minimize energy consumption.

Implementation Objectives:

1. Asses which objectives and parameters should be used for optimization towards sustainable buildings.
2. Create a parametric geometry for the optimization model.
3. implement environmental simulations on the created geometry.
4. Implement a Multi-Objective optimization model.

1.4 Outline

The outline of the paper can be seen in Figure 1.2 below. For each section, a research question is presented.

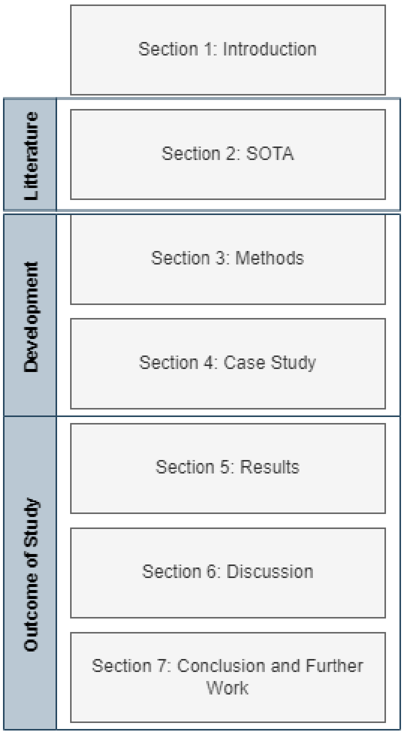


Figure 1.2: Outline for the paper

SOTA

What design factors affect the energy consumption of a building, and what are standard methods for optimizing these factors?

The first section consists of relevant literature that has been used to research relevant theory, architecture, and software.

Method

How can an extendable solution be developed, allowing for optimization and data gathering?

The second section describes the proposed architecture used to solve the problem definition. Finally, a detailed description of the solution workflow is given, showcasing the primary functions used.

Case Study

How can a solution be tested?

The third section will present a case study on a building plot at Bjørvika, Oslo, showcasing a scenario where the solutions will be used. Finally, a proposal from *HAV Eiendom* will be used for comparison based on environmental simulations.

Results

Which assumptions can be made from the optimization?

The fourth section presents the results from the case study. A set of optimal results will be presented for each design approach created in this thesis. Finally, one solution will be selected for each design approach, displaying fundamental building values and a 3D model of the finished building.

Discussion

What are the limitations and possibilities for parametric building design focusing on environmental factors?

In the fifth section, the thesis will be discussed. Questions regarding improvements, limitations, and the reliability of the study will be presented.

1.5 Scope

The problem definition states that this study is meant for the early stages of building design. The scope of this study will therefore focus on creating a building layout optimized for energy efficiency. It will parameterize building facade factors that will directly affect this layout. Different designs will be discussed and examined to achieve optimal results. Active design elements, such as water usage, interior lighting, and appliances, are outside the scope of this study, as well as analyzing building physics.

1.6 Relevance

Rising CO_2 emissions demand changing how we construct buildings, using precise simulations and optimization methods to reduce pollution. There have been many studies into optimizing parts of buildings using environmental simulations and optimization methods, such as GA. However, there is little research and methods for generating building layouts based on such simulations. In addition, incorporating both the exterior and interior parts of the model to be optimized has seen little research. Utilizing parameterized geometry to automate the design process can minimize time spent in the early design stages, focusing on environmental factors.

2 SOTA

What design factors affect the energy consumption of a building, and what are standard methods for optimizing these factors?

A literature review is conducted before creating a proposed architecture for this thesis. As the thesis objective is to parameterize buildings to reduce energy emissions, research on building factors that affect energy consumption has been conducted. Different design approaches have been reviewed to get inspiration when creating the geometry that shall be optimized. Different optimization models have also been evaluated, finding the one suited for the problem. Finally, a selection of tools and plugins have been analyzed and chosen based on the selected KPIs for this study.

2.1 Implementations for Evaluating Energy Consumption

To evaluate the energy consumption of a building, information about the building needs to be integrated. Building Information Modeling (BIM) is a process for managing information about built facilities. With BIM, one can create digital representations of assets throughout their lifecycle, for example, properties of building windows, walls, HVAC systems, and more.

Over the past few decades, BIM has undergone rapid growth due to the information contained in the models [Wen et al., 2021]. With BIM, more precise simulations can be created with the information gathered in the model. This information is essential when trying to optimize energy performance. The energy consumption of a building is most often calculated from building energy modeling programs (BEMPs). BEMPs use BIM properties to create an overall assumption of the total energy load on the building and its energy usage. It is important to note that BEMPs will not necessarily calculate a precise calculation of the energy consumption. However, with more detailed BIM constructions, the more correct results will come from the BEMPs.

Since BIM is an essential part of optimization calculations, choosing the proper framework is an important decision. Not only should the BIM software be able to generate and store building information, but it should also support generative design, have the possibility to integrate a programming language, and support weather simulations. In addition, because 3-D approaches necessitate expertise and experience, the program should have a quick learning curve. It is also beneficial if the BIM software has a large user base because assistance from more experienced users is not far away.

As part of development, choosing a BIM software can be a complex problem, and many programs have advantages and disadvantages regarding their field of study [Shishigin, 2016]. This study mainly considered two software: Rhinoceros 3D and Revit. These programs are commonly used software in the construction industry.

Revit is used solely as a BIM software that offers a digital representation of actual facilities [Shishigin, 2016]. It is a highly complex software with a visual programming language called Dynamo [dyn, 2021]. Dynamo can create generative design, and built-in simulations can be created within Dynamo. Dynamo has a reasonably large user base but does not support many simulation plugins.

Rhinoceros 3D (Rhino) is a computer-aided design (CAD) software but has the opportunity to become a BIM software with plugins such as Ladybug tools. Ladybug tools give the possibility to create BIM functionality directly inside Rhino [Gianpiero Evola, 2020]. Rhino also has a visual programming language called Grasshopper. Grasshopper has the same functionality as Dynamo but with a greater user base and supports more simulation plugins. Rhino also supports the integration of rhino models directly inside Revit with a plugin called Rhino.Inside.Revit. [rhi, 2020]. With another plugin called visualarq [Santos and Beirão, 2019] Revit functionality can also be created directly inside Rhino.

A critical part of building optimization is to provide the model with necessary building envelope information. Walls, roofs, windows, and floors, to name a few, are all part of the building envelope. Therefore, the BIM software must understand all parts in order for the finished model to integrate BEMPs with precision.

BEMPs will change results dependent on parameters provided in the building facade, such as window location, roof material, and wall material. The authors of Yang et al. [2017] try to change envelope parameters to achieve a fair trade-off between three variables; envelope construction cost (ENVCOST), envelope energy performance (ENVLOAD), and window opening rate (WOPR). This paper uses a BEMP called ENVLOAD that only considers the envelope energy performance. This can have some drawbacks since other factors such as building layout and HVAC systems are not considered. On the other hand, the run time of the algorithm can be significantly reduced when only considering envelope building factors.

Figure 2.1 shows the envelope parameters that can change to optimize energy efficiency.

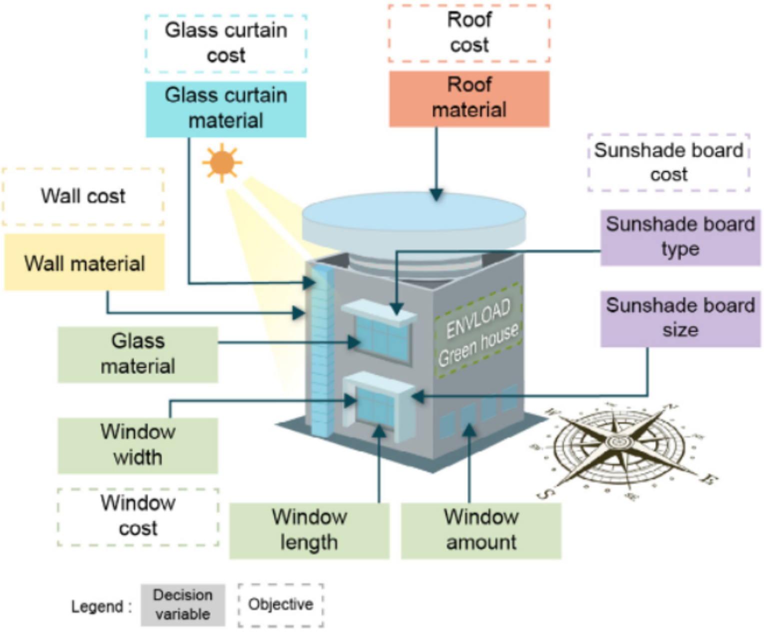


Figure 2.1: Illustration showcasing the factors that make up cost and env-load. [Yang et al., 2017]

2.2 KPIs for Energy Efficiency

Design factors are the controllable factors that are suitably varied in order to obtain the desired performance [Alkazraji, 2008]. These factors can be used to alter the model to be optimized for energy efficiency. Jiang et al. [2018] proposes a workflow for identifying important design factors to create energy efficient buildings. The approach takes advantage of BIM and ontology to facilitate the process of green building evaluation. Chinese evaluation standard green building (ESGB) is taken as an example to validate the feasibility of the proposed method. The standard evaluates the green certification of a building, as shown in Figure 2.2. ESGB can be used to understand what parameters affect the total consumption of a building. A combination of all should therefore be taken into consideration when trying to decrease building emissions. The selection of what parameters should be considered varies depending on the task. In the paper, Alwisy et al. [2019] a ranking system of green building design factors (GBDFs) is created. The conducted report is carried out based on the frequency of relevant publications in accordance with leading BEMPs such as EnergyPlus, eQuest and TRNSYS. The GBDF evaluation is divided into five groups: building systems, mechanical and electrical requirements, building design, weather conditions, and renewable energy. Within these groups, essential design factors are exterior walls, HVAC systems, building orientation, weather data, and solar radiation.

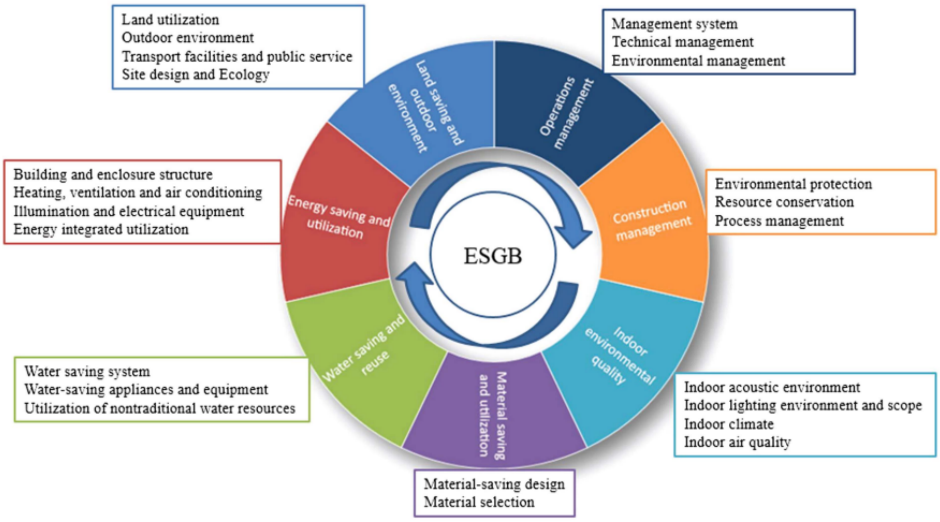


Figure 2.2: Index system of ESGB.
Jiang et al. [2018]

As stated above, GBDF highlights important building factors in energy optimization. These factors can be separated into two groups, passive and active design elements, and can be seen in Table 1 below.

These factors affect the energy consumption of a building as changing these factors will

	Active	**Passive**
Elements	HVAC	Envelope
	Hot Water	Width/Length
	Lighting	Orientation
	Appliances	Cooling Strategies
	Building Automation	Heating Strategies
		Thermal Energy Strorage
		Solar radiation

Table 1: The table shows both active and passive design elements.

effect the energy needed to keep a comfortable interior climate. Passive design elements encompasses the surrounding terrain and structures to improve the internal thermal comfort in the building [Huo et al., 2019]. To minimize the cooling and heating, different strategies can be utilized, such as altering the window properties or adding sun shades [Zhu and Lin, 2004]. The building layout can affect the energy consumption in relation to solar radiation, as the layout can be manipulated into shapes benefiting from the sun. Active design elements are based on reducing the energy consumption by altering the interior components of a building. HVAC systems, heating, water usage, and lighting are some of the elements considered active [Chen et al., 2015]. In general, implementing passive design elements is more cost-effective, as there is usually a one-time cost at installation, whereas active solutions have costs over time [Zhang et al., 2011].

2.3 Design Approaches for Building Optimization

Reviewing different design methods is an essential step before choosing a design path that should be used for optimization. This section describes some possible directions when parameterizing a design approach.

2.3.1 Floorplan Design Approach

When creating a building, the designer may have some set rooms that need to be implemented within the building. For instance, if a designer creates an office building, he will need toilets, offices, meeting rooms, and a hall. It is also essential that the hall is connected to all rooms and that meeting rooms are as close to the toilet as possible. This is a fundamental combinatorial packing problem, such as KNAPSACK and BIN PACKING, and falls under NP-hard problems [Klawitter et al., 2021].

In Verma and Thakur [2010], a comprehensive background check was done to determine the feasibility of the most used methods in floor planning. As a result, four techniques were tested and scored based on different categories:

- Additive Space Allocation
- Permutational Space Allocation
- Analogical Methods
- Genetic Algorithm

Some categories included constraint handling, runtime/complexity, and creating novel solutions. Genetic Algorithm (GA) is the only method scoring acceptable or higher in all categories. With the use of a Multi-objective genetic algorithm (MOGA), the issue of weighing the different constraints and fitness functions is feasible [Deb et al., 2002].

The different fitness functions used for optimization need to be predetermined by the developer. Configuring fitness functions, therefore, must reflect the solution's intended end goal. The intended goal of the algorithm is to generate a collection of rectangle shapes in close proximity while not creating an overlap. A proximity graph can be used to connect several nodes in a graph network [Yang et al., 2002]. The network will consist of nodes connected with edges. The edges can be given a Euclidean distance to get the distance from other nodes. When adding all the edges together, a number representing how close the rectangles are to each other is produced. Minimizing this number results in a compact floor plan, utilizing the space given in the best way.

In 2020 Egor et al. [2020] went out to create an automated floor plan solution to reduce time spent in the early phases of a building project. They found that the existing solutions for automated floor plans concentrated on smaller geometry, typically working well on houses and flats; however, they are unsuitable for larger buildings, such as public buildings and large offices. Their work resulted in a floor plan generator named *MagnetizingFPG*. A working floor plan is generated using a grid-based system, connecting boxes based on a distance graph. The finished solution was made into a Grasshopper plugin for others to use. Figure 2.3 shows an example of a working result.

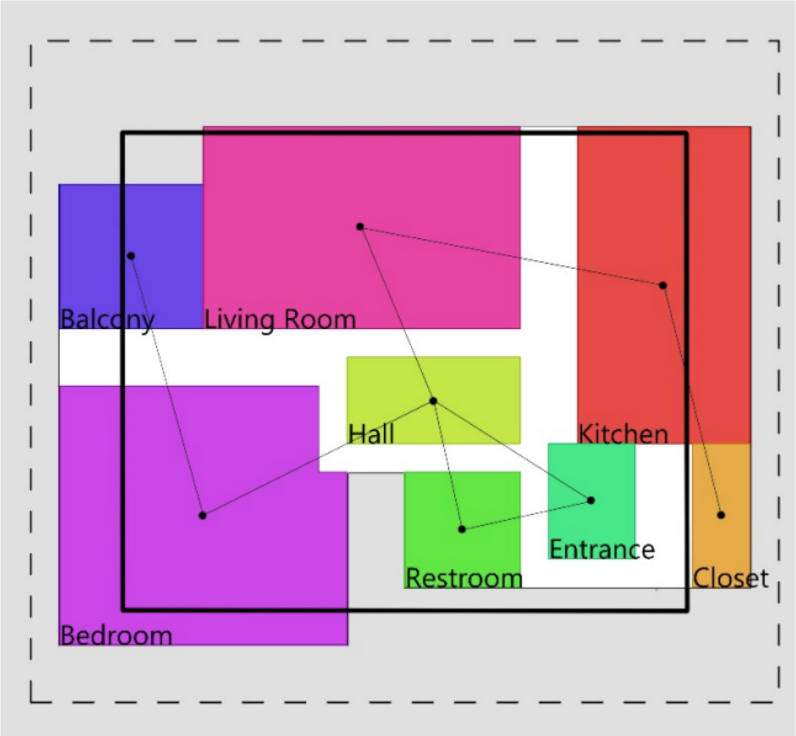
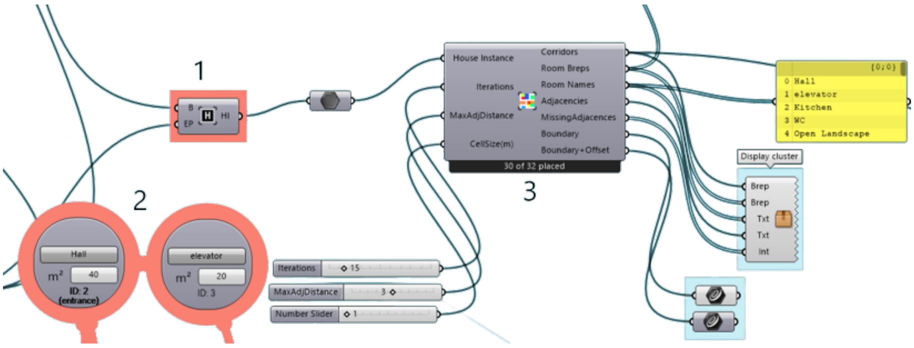


Figure 2.3: Example of final result with a small boundary. The rooms are placed inside the boundary, minus the offset shown with the stippled line. The rooms are connected with a corridor.

[Egor et al., 2020]



```
{0;0}
0 Hall
1 elevator
2 Kitchen
3 WC
4 Open Landscape
```

Figure 2.4: The main components of *MagnetizingFPG* in Grasshopper, **1** House instance, **2** Room instance, **3** Main component running the algorithm

The algorithm works iteratively, serving as a quasi-evolutionary process [Egor et al., 2020], meaning it emulates the strategy of evolutionary algorithms. The house instance that contains information about all room instances requires a boundary curve and an entry point. The house instance is then sent to the main component together with numerical values to control the algorithm. The main components can be seen in Figure 2.4 above.

The main function for the floorplan generator being run iteratively can be seen below:

- **1** All room instances are connected to the corridor, creating a connection between all rooms.
- **2** The entrance room defined as one of the room instances are placed close to the entrance point.
- **3** Rooms are sorted by adjacency, enabling the algorithm to place rooms with the most adjacency constraints first. The corridor is then analyzed to see if a cell can be used to place a room, needing to meet the following requirements:
  - **1** The cell should be part of a corridor.
  - **2** The cell needs to be closer than the maximum adjacency threshold.
  - **3** Space should be available around the cell for the placement of new rooms.
- **4** If the cell is successfully selected, a room is placed on the cell, and point **3** is repeated for the subsequent rooms.
- **5** If the next room cant be placed, it indicates that the boundary is reached or that rooms were not placed closed enough.

The main iteration loop for the generator:

- **1** Run Main function 3-5 times, and keep the resulting branches in a list
- **2** Sort by number of room instances successfully placed.
- **3** Reverse the placement process back to a certain number of rooms, then start the placement process again, trying to optimize the number of rooms put down.

The iteration loop runs for n iterations, finally producing the best solution.

2.3.2 Polygon Shape Design Approach

The paper Wang et al. [2006] tries to find the most energy-efficient building outline based on polygon shapes where three main parameters can change: number of edges, angle, and length. Multiple shapes can be created simply by changing these parameters to find the most optimal building outline that can optimize a set of evaluation functions. Figure 2.5 displays 4 optimized shapes proposed in the paper.

Genetic algorithms are employed to optimize building shapes, in which the shape affects the building footprint. The impact of two alternative solutions on computational viability and productivity is examined. An optimization algorithm uses shape-related factors and a few other envelope-related plan factors, including window properties and shades as parameters. The profitability of a green design plan is determined by considering lifecycle costs as well as environmental effects when determining fitness values for a green project. As a result, developing an optimization model that incorporates both lifecycle cost and environmental consequences, as well as standards like ESGB, could provide a sufficient fitness function for the PoC proposed in this paper.

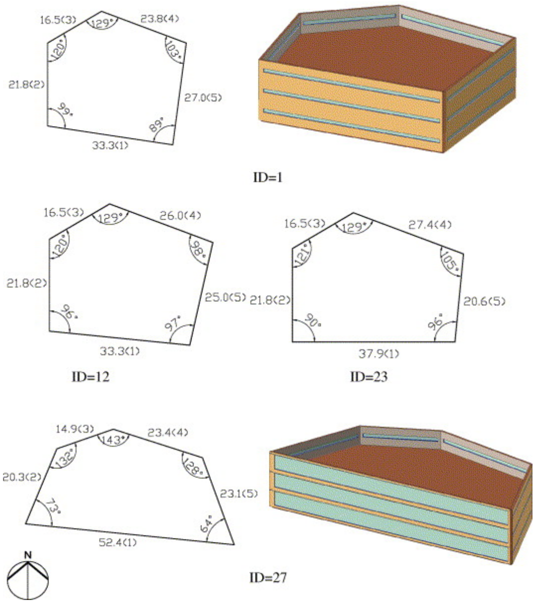


Figure 2.5: Two Pareto solutions proposed using polygon shapes.
[Wang et al., 2006]

2.3.3 Common Building Shapes Design Approach

Evaluating different building shapes and selecting the shape that optimizes the task at hand is a common selection method for finding an optimal solution [Geraldi et al., 2021] [Sazzad, 2015] [Zhang et al., 2017]. Some shapes used in these studies are shapes like H, I, L, C, and courtyard. The different studies' objective function varies, looking at aspects such as energy consumption and response to wind and earthquakes. The papers Geraldi et al. [2021], and Zhang et al. [2017] consider how different building shapes influence the energy consumption of the building, both using EnergyPlus for evaluation. The authors use a set of predefined buildings, not parametrizing the process, and thereafter evaluating each predefined building. For instance, Zhang et al. [2017] looks at seven predefined building shapes: I, L, C, H, courtyard, highrise, and H with an atrium shape. Each can rotate 90°, 180° and 270° creating a total of $7 * 4 = 28$ building designs. The solution space could be significantly increased with more parameterization of the different building shapes. Nevertheless, runtime and complexity will increase with a larger solution space.

2.3.4 Design Choices not Affecting Building Layout

As described in 2.2, multiple factors affect the overall building performance without changing the building outline/layout. As discussed, the building envelope is an important aspect when optimizing a building. Window wall ratio, building material, and sunshade will affect the overall energy performance of the building. Su and Zhang [2010] described how optimizing the WWR and window material can reduce the energy impact of the building by $9 - 15\%$.

Interior elements that contribute to energy efficiency, such as interior lighting, water usage, and appliances, do not affect the envelope or layout of the building.

2.4 Evolutionary Algorithms

Evolutionary algorithms (EA) are used inside many industries including design optimization [Gan et al., 2019] [Rodrigues et al., 2014] [Yi, 2019]. Within building optimization, multi-objective evolutionary algorithms (MOGA) are most often used, as more than one parameter needs to be optimized. With two or more optimization parameters, selections of the most suitable genomes in the fitness landscape is slightly more complicated than in Single objective evolutionary algorithms (SOEA).

The concept of dominance is one technique for selecting a genome. Given two solutions, one is said to dominate the other if the genome's fitness is at least as high for all objectives and strictly higher than one [Eiben et al., 2003].

Using the symbol $\preceq$ for dominance, we can say formally that $A \preceq B$ as:

$$A \preceq B \Leftrightarrow \forall_i \in \{1, 2, .., n\} a_i \geq b_i \land \exists_i \in \{1, 2, .., n\} a_i > b_i \tag{1}$$

The genomes not dominated by any other are called non-dominated solutions, more commonly known as the **Pareto front**. The Pareto front is used to select parents for new generations and will be part of the final solution when a termination condition is met. With a Pareto front, multiple solutions are created based on the set of evaluation functions added to the MOGA. The user can then select the preferable solution within the Pareto front.

Jonathan A. Wright [2002] investigates the use of a MOGA search approach. It seeks to find the best balance of energy cost and thermal comfort. The solution in Jonathan A. Wright [2002] will generate a Heating, Ventilation, and Air Conditioning (HVAC) system. The correctness of the MOGA is tested and reviewed, seeing that one can not solely depend on the algorithm to find the optimal solution. Another issue with MOGA is the algorithm's time complexity, which is dependent on the number of objective functions. The time complexity of a popular MOGA (NSGA), for instance, is $O(MN^2)$, where M

is the number of objectives and N is the size of the dataset [Journal and Computing, 2010]. As a result, a stack consisting of an artificial neural network (ANN) for classification followed by a MOGA for optimization is proposed [Laurent Magnier, 2010]. The factor of time related to optimizing might be prohibitively high in building applications, where assessments are performed for the most part by time-consuming simulations such as TRNSYS and EnergyPlus. Therefore implementing a new selection method such as ANN can decrease the time complexity or simply decrease the number of evaluation functions where this is a possibility [Laurent Magnier, 2010].

The most used optimization algorithm in the building design industry is genetic algorithms [Hamdy et al., 2016]. Grasshopper is comparable with multiple plugins that can integrate genetic algorithms. The most popular are Galapagos, Wallacei, and Octopus. The aforementioned optimizers have several properties in common. They all allow manipulation of the most common variables in a genetic algorithm, such as mutation factor, crossover probability, and the number of individuals and generations [Mirjalili, 2019].

2.4.1 GA Plugins

When integrating GA, one can integrate it from scratch or use already made tools and support for integration. The tools used and tested in this PoC are three plugins that can integrate SOGA and MOGA within Rhino. Galapagos is a built-in plugin for integration of SOGA, while Octopus and Wallacei are plugins for integration of MOGAs.

Galapagos implements two optimisation models, one using an evolutionary algorithm and one using simulated annealing [Rutten, 2013]. The genetic algorithm implemented in the plugin is a SOGA, so only one objective function can be maximized/minimized.

Octopus is a plugin made for Grasshopper by the University of Applied Arts Vienna, and Bollinger+Grohmann Engineers [Octopus, 2012]. It is based on the MOGAs SPEA-2 and HypE. It enables the use of several optimization functions, trying to generate a Pareto front for the best trade-off between the objectives. The interface allows the user to see the solution space in a 3D coordinate system. This allows for easy selection of a desired solution within the Pareto front.

Wallacei employs the NSGA-2 algorithm as the primary evolutionary algorithm and utilizes the K-means method as the clustering algorithm [wal, 2015]. Wallacei also comes with analytic tools to fully understand the evolutionary runs and make more informed decisions.

2.5 Overview of Tools Support for Selected KPIs and Desired Workflow

This subsection focuses on selected KPIs and existing tools supporting integration of the implementation objectives. To derive information on the energy consumption of a created building, integration of environmental simulations and BEMPs is a necessity.

2.5.1 Environmental Simulations

Environmental simulations are used to simulate and analyze the environmental impact, such as wind and sun, on a building design and its surroundings. A widespread tool for doing environmental studies is Ladybug tools.

Ladybug Tools is a suite of computer applications enabling users to run environmental analysis on 3D models through Grasshopper for Rhino, and Dynamo for Revit [Mostapha Sadeghipour Roudsari, 2021]. The programs aim at connecting CAD software with validated simulation engines, such as Openstudio, THERM, Radiance, and EnergyPlus. There are four main modules inside Ladybug tools, Ladybug, Honeybee, Dragonfly, and Butterfly, each specializing in a particular type of simulation.

Ladybug's primary function is to import and visualize weather data gathered from a *.epw* file Mostapha Sadeghipour Roudsari [2021]. *Epw* files are complex files containing data about weather from different geographical locations. This enables Ladybug to run weather-related simulations on complex geometry. It is often used in the early stages of a building project, as only the facade of the building is required to get satisfactory results. Ladybug utilizes mainly two values for computing the solar radiation and its effect on geometry Naboni et al. [2019]. The first is T_{mrt}, or mean radiant temperature. The MRT is first simulated by running the EnergyPlus engine on the surfaces of the geometry. Then, the view factor, meaning the exposure to sunlight, is computed using Rhino's ray-tracing solution. The equation can be seen below, where F is the fraction of the spherical view occupied by a given indoor surface and T is the temperature of that surface. Next, the surrounding geometry is used, calculating how many of the surrounding geometries are viewed from the face of the surface.

$$T_{mrt} = [\sum_{i=1}^{N} F_i * T_i^4] \tag{2}$$

On the exterior geometry, sky temperature is also accounted for, considering the longwave loss. The T_{sky} is shown below, where L_a is the longwave radiation from the sky, ε_p being the emissivity of the human body (0.95), and σ the Stefan-Boltzmann constant.

$$T_{sky} = \frac{L_a}{(\varepsilon_p * \sigma)^{\frac{1}{4}}} \tag{3}$$

Gianpiero Evola [2020] tested the viability of Ladybug simulation outputs. They created a case study to compare the results given by Ladybug to real-world measurements. They used TESTO 480 data logger connected to sensors for measuring the MRT in different locations, corresponding to direct sunlight and shaded areas. The experiment showed accurate simulation results for shaded areas, with 0.6 and 0.3 degrees in discrepancy at 13:00 and 14:00, respectively. However, in areas in direct sunlight, the generated MRT continuously exceeded 5 degrees in comparison with the measured temperature [Gianpiero Evola, 2020]. This is, however, in line with other simulation software, such as the ENVI-met model.

Honeybee introduces several features on top of Ladybug. By modeling more complex geometry, adding features such as rooms, doors, and apertures, the simulation engines can run simulations on both the exterior and interior, giving detailed results on energy consumption and interior comfort. Honeybee is directed more against interior simulations, using EnergyPlus and OpenStudio to simulate HVAC systems, indoor comfort, and material selection [Mostapha Sadeghipour Roudsari, 2021].

2.5.2 Building Energy Modeling Programs

To validate the building's environmental aspects and create objective functions for the EA, simulation engines need to be incorporated into the solution [Samaan et al., 2018]. Two of the most popular engines for environmental studies are EnergyPlus and OpenStudio.

EnergyPlus is a building performance simulation program, released in 2001 [Crawley et al., 2000]. The developers from ASHRAE saw the need for a modern simulation engine. Many of the existing solutions were starting to depreciate due to new hardware development and modularity that came with the newer programming languages. The engine is written in Fortran 90, laying the foundation for a modular code-base [Crawley et al., 2000]. EnergyPlus builds on the strengths of BLAST and DOE-2, which are older simulation engines. In addition, EnergyPlus introduced several new features, including:

- Heat balance load calculations.
- User-configurable HVAC system description.
- A modular structure, enabling third-party developers to extend the capabilities of the engine.
- Simple input - and output formats to enable front-end developers to create user-friendly interfaces.

Due to the modularity and simple formats, EnergyPlus is compatible with Honeybee for Grasshopper. Honeybee works as a third-party interface, enabling simulations of Rhino geometry in Grasshopper. An overview of EnergyPlus, its workflow, and how it connects to third-party software can be seen in Figure 2.6.

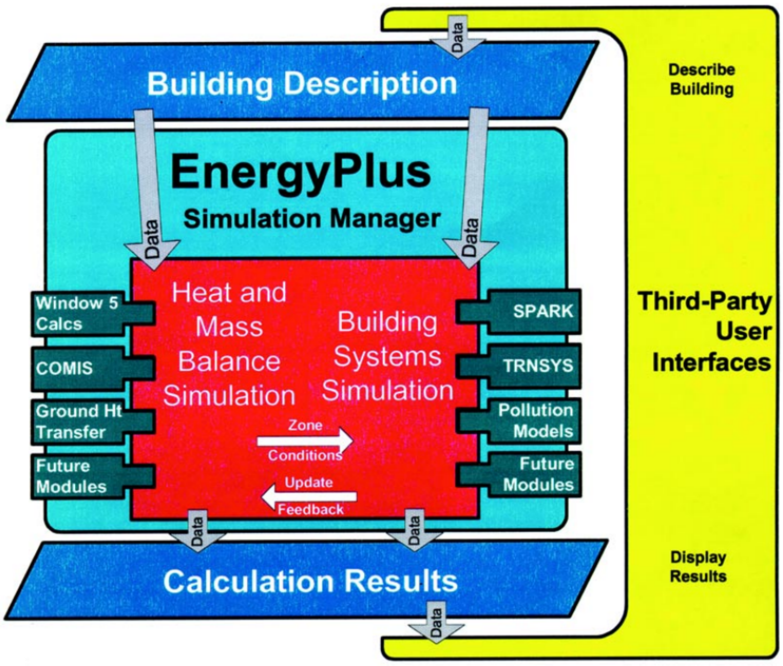


Figure 2.6: An overview of EnergyPlus and its workflow. Honeybee works as third-party user interface.

Crawley et al. [2000]

In 2011 Guglielmetti et al. [2011] went out to develop an open-source framework to take advantage of several simulation engines working together in one framework. The development team set several goals for the new software, named OpenStudio:

- Optimizing the runtime
- Well organized and object-oriented code
- Integration of several engines
- Optimization and sensitivity analysis
- Data display and reporting
- GUI interaction
- Command-line interface

OpenStudio primarily utilizes Radiance and EnergyPlus to simulate the daylight and HVAC systems for the building. Enabling the two engines to simulate their respective focus area gives OpenStudio great accuracy, and a broad use-case [Guglielmetti et al., 2011]. The core of the OpenStudio simulation framework is the *.osm* file, containing the model with all objects and parameters. Through Honeybee, one can use Honeybee objectives as input, together with a weather file (*.epw*), as well as simulation parameters. Sim parameters control simulation values such as time-step, simulation period, and shadow calculations. The OpenStudio command-line tool is opened from Grasshopper, running the simulation. The OSM component returns several parameters, such as an SQL file and an HTML file containing all simulation results. Honeybee allows the user to query the SQL file from within Grasshopper and retrieve data to be used for either optimization or visualization.

Both these simulation engines are used frequently in analysis and simulation tasks around the world, giving them a great reputation for providing accurate results [Bastos Porsani et al., 2021] [Gali and Yilmaz, 2012] [Rastogi et al., 2017].

2.6 Summary

Research into relevant theories and methods used in optimization and environmental analysis needs to be conducted to enable a solution to account for several different problems, creating a single result. As discussed, active and passive design elements influence the overall energy consumption of a building. There are countless different design choices that all have different benefits. Therefore one simply has to select a solution that one believes is the best fit for the problem at hand. As the first step of designing a building comes down to the layout, this should be a key factor when selecting the correct design direction. Building facades can always be added when a layout is created but should be appropriately used to calculate the correct energy consumption of the building. After choosing a building design direction finding a suitable optimization algorithm should also be made, where EAs are popular choices in building design optimization.

3 Methods

How can an extendable solution be developed, allowing for optimization and data gathering?

This section describes the workflow for implementing the thesis objectives. Implementation of geometry, BIM elements, BEMPs, and optimization models are presented. The PoC utilizes two different design approaches for generating the parameterized geometry used in the optimization process. The first method takes inspiration from the floor plan design approach, while the second focuses on utilizing pre-defined building shapes.

3.1 Architecture

In order to design energy-efficient buildings, selections and combinations of tools are essential. The proposed workflow mythology in this PoC combines parametric modeling, MOGA, and energy simulations. The software tools proposed are Grasshopper and Rhinoceros3d with the optimization model plug-in Octopus. Ladybug and Honeybee are used for environmental studies and to integrate BIM elements. EnergyPlus and OpenStudio are thereafter used to calculate energy consumption.

The architecture is divided into two steps, parameterization, and optimization. Parameterization consists of changing the characteristics of a building that can affect the overall thermal performance of the building [Touloupaki and Theodosiou, 2017]. The total energy performance is affected by many factors such as building layout, floor plan, HVAC systems, materials, and location. These factors will affect each other, and when creating an optimal building, all these factors should be taken into consideration. On the other hand, considering every factor affecting a building will make the program complex and lead to higher run times. In this paper, the layout of the building is taken into consideration, while other factors are set before the program is started.

The BIM/CAD software chosen for the PoC was Rhino, as it has a built-in visual programming language (Grasshopper) that can be integrated with the well-known environmental analysis tool, Ladybug. If a user prefers to use Revit, Rhino supports the integration of 3d models directly inside Revit using the plug-in, Rhino.inside.Revit [rhi, 2020].

Before running the simulation, constraints and specifications are made by the user. Constrains can make the program more efficient since unsatisfied results are removed before the simulations start. The parametric setup then creates an initial population of design solutions that are sent to the second part of the architecture, optimization. Finally, the Octopus plug-in is used for the optimization process, optimizing a set of fitness functions specified by the user.

As described in 2.4, Octopus support MOGA that gives the opportunity for multiple evaluation functions and the creation of a Pareto front that can be presented to the user. Octopus also supports two well-known optimization algorithms, SPEA-2, and HypE.

Octopus has a well-created user interface where changing parameters within the MOGA is simple, and the results are displayed intuitively.

For simulations, Ladybug and Honeybee are used, Ladybug to gather a visual understanding of the specified location and building, and Honeybee for energy simulations and BIM integration. Honeybee can create BIM elements that affect the energy modeling when working with OpenStudio and EnergyPlus. As the Ladybug suite of tools includes an extensive and comprehensive selection of components, as well as using certified engines for simulations, they were chosen for this PoC.

After running the MOGA, the results following the Pareto front will be displayed to the user. Figure 3.1 shows a visual representation of the architecture.

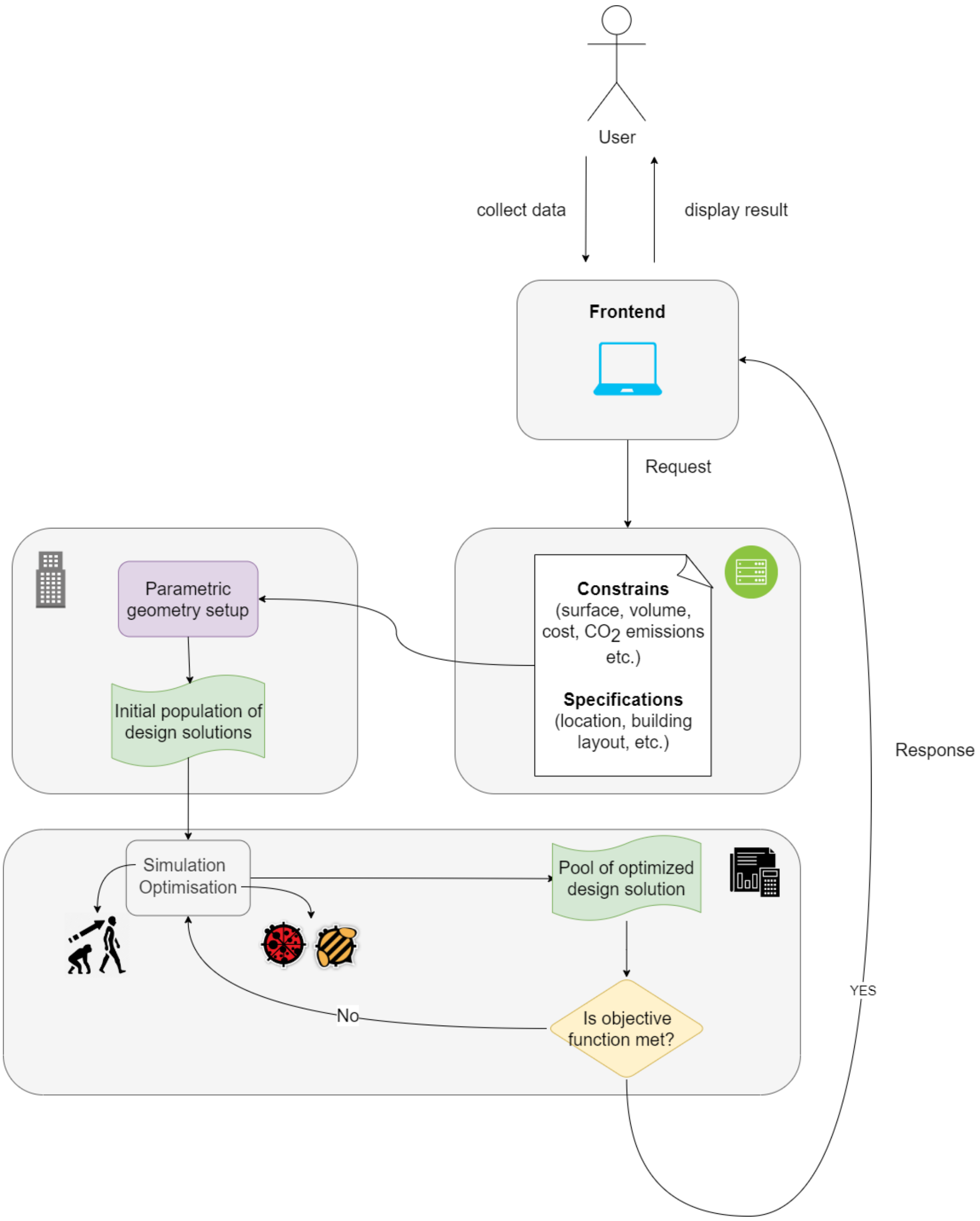


Figure 3.1: Proposed software architecture

3.2 Generating Geometry

The PoC utilizes two different design approaches for generating the parameterized geometry used in the optimization process. The first method takes inspiration from the floor plan design approach, while the second focuses on utilizing pre-defined building shapes.

3.2.1 Floorplan Design Approach

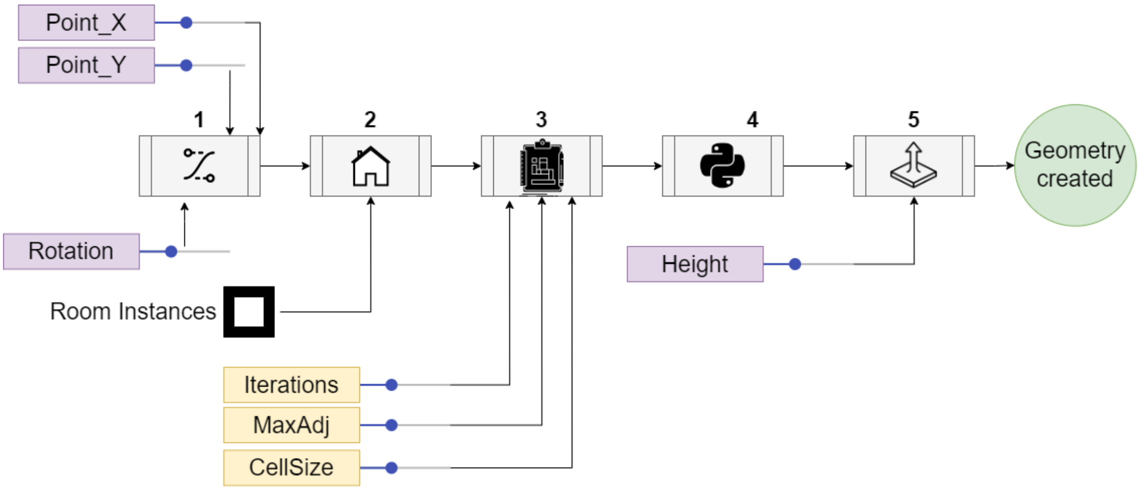


Figure 3.2: Workflow for generating geometry for floorplan design approach

The geometry is created in Grasshopper using built-in components and the open-source plug-in, MagnetizingFPG, described in Section 2. A flowchart describing how the geometry was created can be seen in figure 3.2, where the MOGA controls purple sliders. The scripts for the floorplan generation can be seen in Appendix B.

First, a boundary curve is selected and added to the Grasshopper script **1**. This acts as the limit for where you want the building to be placed. Next, to generate the floorplan, MagnetizingFPG is used. As discussed in Section 2.3, room instances must be set to tell the floorplan generator what rooms should be added. Room instances contain three important properties; size in m^2, name, and connections to other room instances, creating a graph network between all rooms. The graph network created in this PoC can be seen in figure 3.3, where each node indicates a room instance.

The graph network is used in the floor plan calculation, where the connection between room instances plays an important role. The optimization model used in MagnetizingFPG tries to place rooms according to the graph network, so connected room instances are as close together as possible. MagnetizingFPG needs a room to be placed first as it uses the starting room for connecting the rest of the network. The starting point **1** for the first room is placed in the center of the boundary curve. The point is parameterized to allow the optimization model to change this position. The finished room network, start point, and boundary curve are stored in a data component **2** to be later used by the floorplan generator. For the purpose of generating office buildings, a simple network of rooms was created, including; Hall, Kitchen, Elevator, Offices, Open Landscape, Meeting Rooms, and WC.

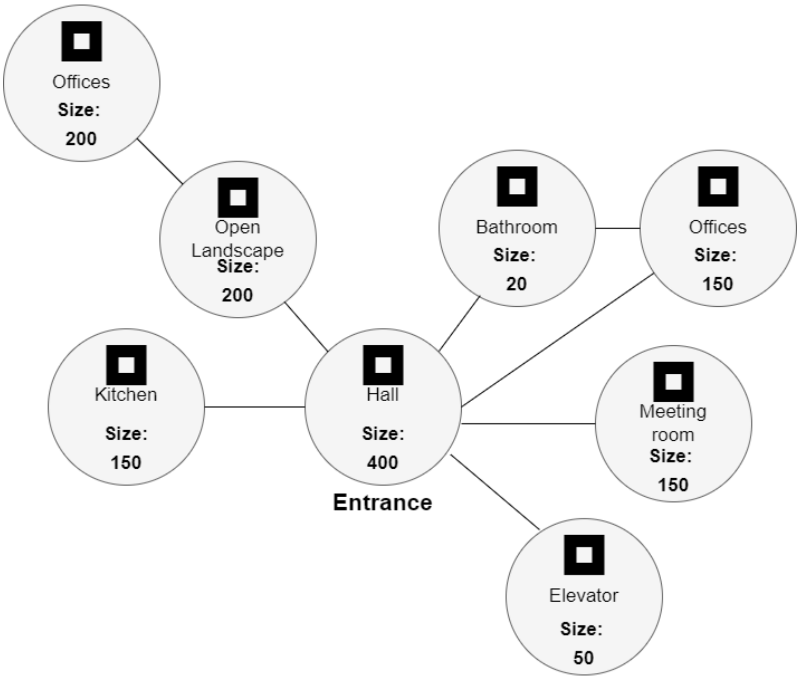


Figure 3.3: The graph network connecting all room instances together.

The graph network, boundary curve, and starting point are sent to the main MagnetizingFPG component **3**. The main component takes in the graph network and starts placing down the entrance room close to the entrance point. The area inside the boundary curve is made into a cell grid, allowing the generator to place down squares to make up rooms and corridors. The iterative process creates a corridor each time a room is placed, making sure all rooms are reachable through this corridor. The quasi-evolutionary process, as described in SOTA, looks at each room's connections and places down the room with most connection restrictions. The evolutionary process tries to minimize the distance between connected rooms in the network and keep all rooms inside the boundary. This process is repeated until all rooms in the network are placed. The generator then removes some of the rooms placed last and tries to minimize the graph distances based on the connections in the graph network. The component has three changeable inputs. The first is max adjacency, meaning the max amount of rooms that can be placed between two rooms connected in the graph network. Cell size defines the size of the cells in the grid. Finally, iterations control how many times the main floorplan component will remove rooms to try and find a more compact solution. An example of a floor plan generated in the Grasshopper script can be seen in Figure 3.4 below.

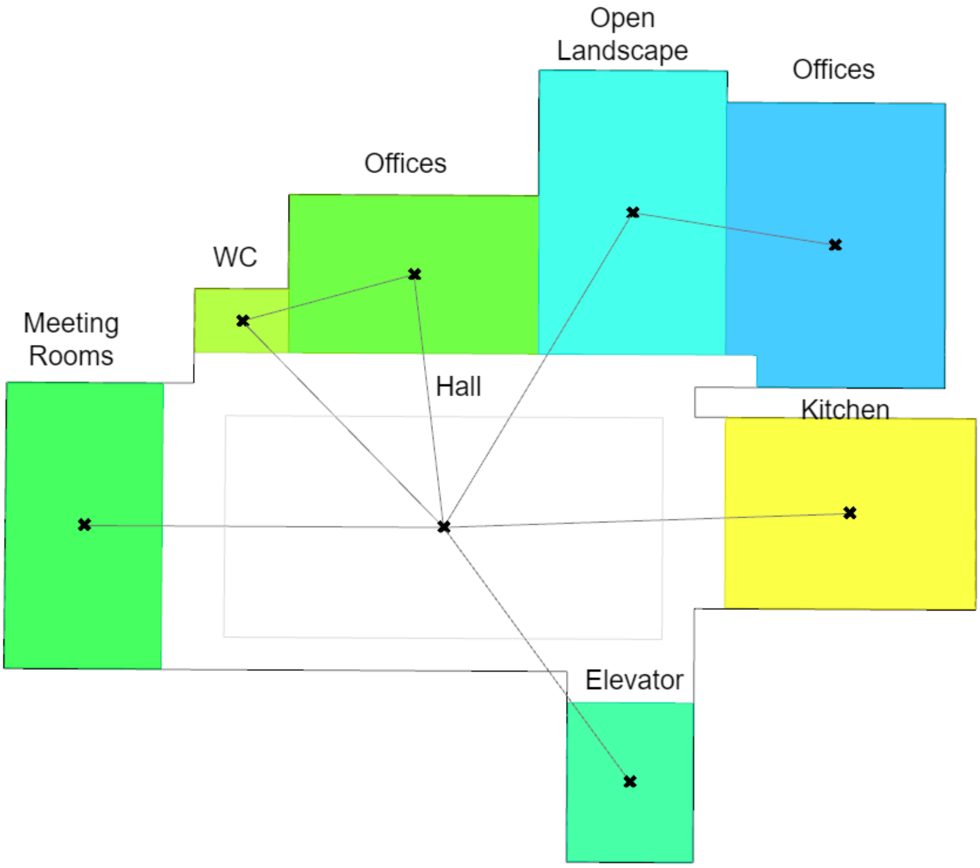


Figure 3.4: Example of floor plan generated by MagnetizingFPG

The MagnetizingFPG main component **3** outputs the rooms as a list of curves, together with a list corresponding to the name of each room. These lists are given to a python script **4**, sorting the rooms by name. This is done to allow different extrusion heights. After the rooms are sorted, they are extruded, and stories are created **5**. The floor plan is identical on each floor, allowing for elevators and toilets to be directly beneath each other. An extrusion of the floor plan generated earlier can be seen in Figure 3.5.

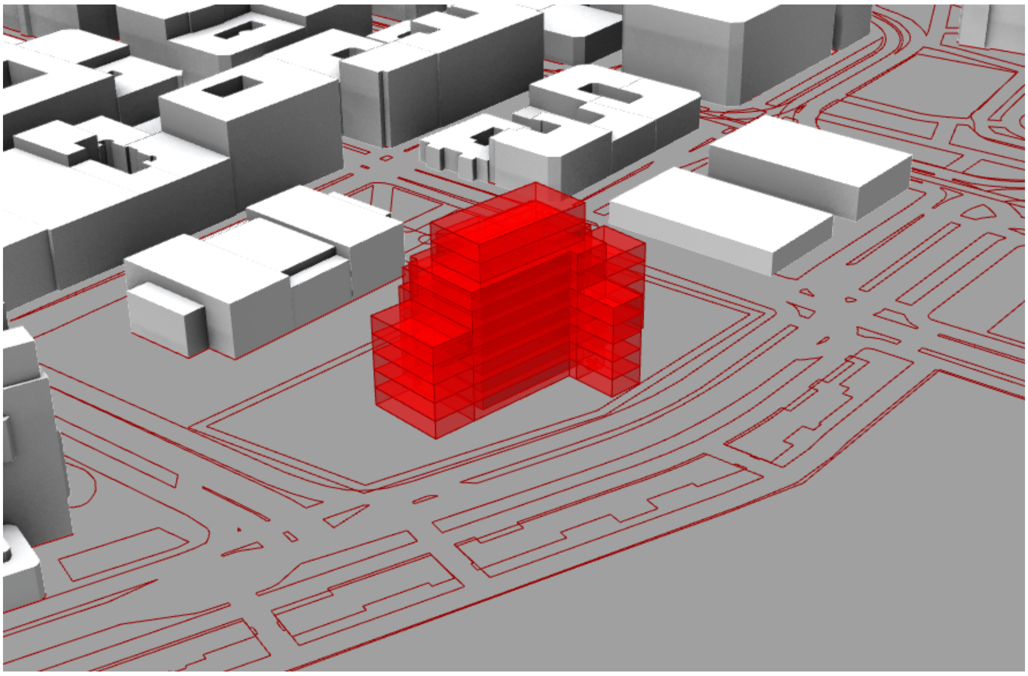


Figure 3.5: Example of extruded floor plan

3.2.2 Building Shape Design Approach

As discussed in SOTA, the building shape design approach evaluates different building shapes to find an optimal solution. The approach presented in this paper is divided into two separate ways to parameterize building shape geometry for optimization. The first approach has the opportunity to create courtyard and C-shape buildings. The parameterization of the design is created such that any combination of the genes will always make one of the two shapes. This type of implementation is presented to show how a design parameterization can be created if a specified building shape is important for the design process.

The other approach is presented using H-shape. New shapes such as rectangular layouts or L shapes can easily be implemented using a similar design parameterization approach used to create the H-shape. This implementation of common building shapes can create multiple design solutions, not just one specific shape given a combination of genes. This way of parameterizing the geometry allows the creation of countless building designs using a shape as its baseline.

Development of Courtyard and C-shape

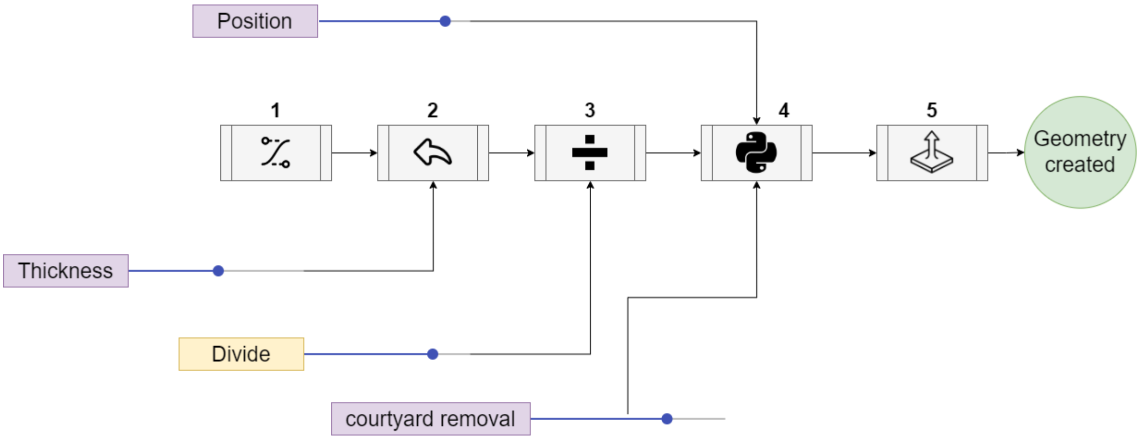


Figure 3.6: Geometry creation of C- and courtyard shape

The geometry is created in Grasshopper using its built-in components. The structure of the geometry can be seen in figure 3.6. The scripts for the C-shape generation can be seen in Appendix D.

First, the boundary curve for the building field is added to the Grasshopper script **1**. Then, this boundary curve is used directly as a baseline for creating the geometry, as the C/courtyard shape will be parallel to the boundary curve. This will be implemented by adding an offset to the boundary curve that will be used to create the geometry **2**. Using the boundary curve as the outer layer and the offset curve as the inner layer alone created the geometry for the courtyard, where the thickness changes depending on the offset distance.

For creating the C-shape, the courtyard geometry is divided into several segments the user can specify before running the optimization model **3**. The higher the number of divisions, the more design solutions can be created. However, an increase in the design space will affect the algorithm's time complexity. After dividing the courtyard shape, the removal of a specified number of segments at a specific position is implemented using a python component within the Grasshopper script **4**. The finished geometry is thereafter extruded. As seen in figure 3.6 the purple parameters will be used in the optimization model to find an optimal solution.

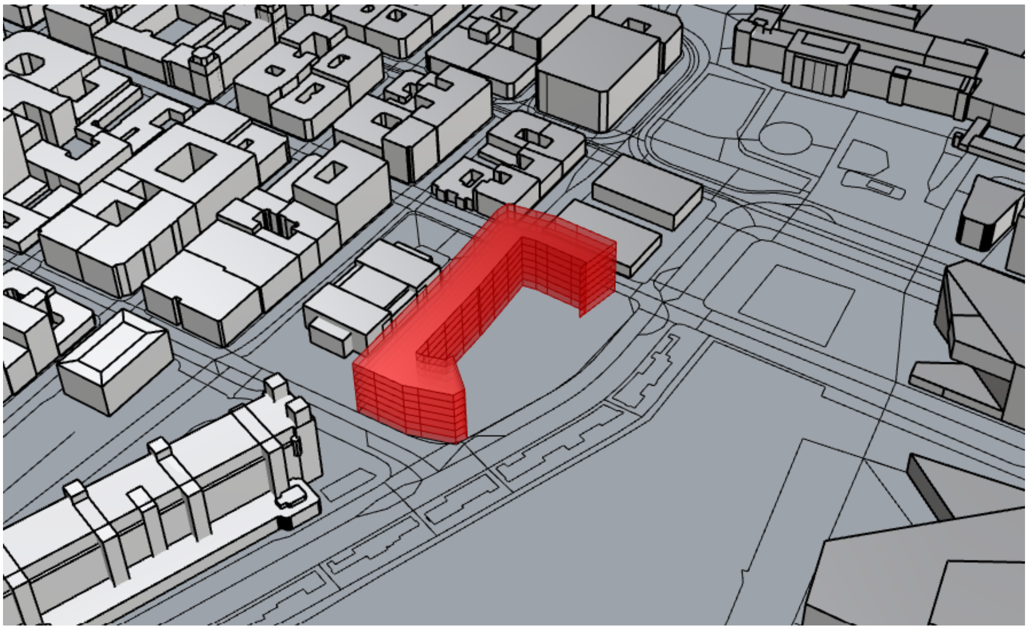


Figure 3.7: Courtyard/C-shape displayed on given location

Figure 3.7 displays the output of one specific gene combination from the cortyard/C shape script. As described, the thickness, location of the opening, and size of the opening are parameters that can be changed.

Development of Geometry Based on H-shape

The geometry based on H-shape is created such that a curve representing the shape can be rotated, change size, and position. The shape's position can be manipulated such that the curve can be inside the boundary, outside it, or part of the shape can intersect with the boundary. Only the intersecting parts between the shape and the boundary curve will be extruded. This is why multiple design solutions that do not necessarily represent a specific base shape can be created. For example, figure 3.8 displays the output of a genome that uses the H-shape as its baseline to create geometry that does not represent this shape. As seen in the image, the part of the H-shape not intersecting with the boundary curve will not be extruded. For this study, the H-shape is used to represent this approach, but multiple shapes can also be implemented using the same geometric setup used for creating this specific shape.

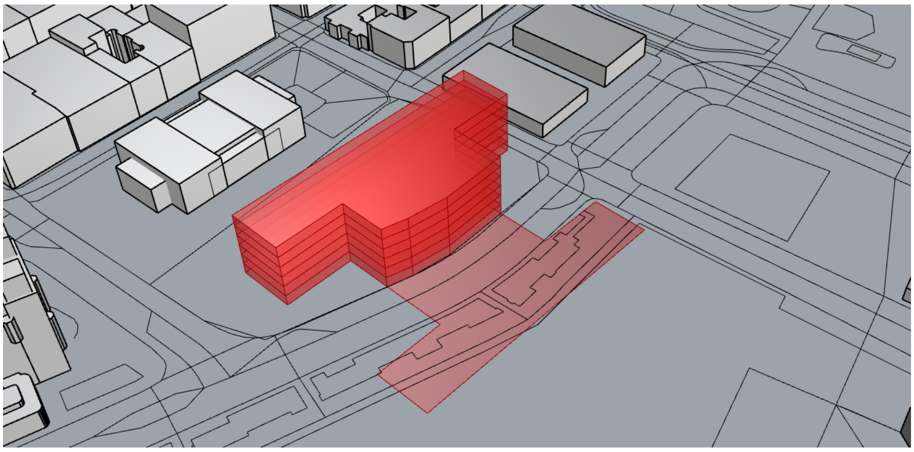


Figure 3.8: Geometry creation of H-shape

The flowchart of the geometric setup can be seen in figure 3.9. The scripts for the H-shape generation can be seen in Appendix C.

Like the other design approaches, the first step is to add the boundary curve to the Grasshopper script **1**. Thereafter three rectangles are created **2**, **3**, **4**, they change size in x- and y- position and is dependent on each other. Then a rotation is implemented **5** such that the shape has the opportunity to rotate 360°. Lastly, extrusion of the intersection between the shape and the boundary curve will be the final building design.

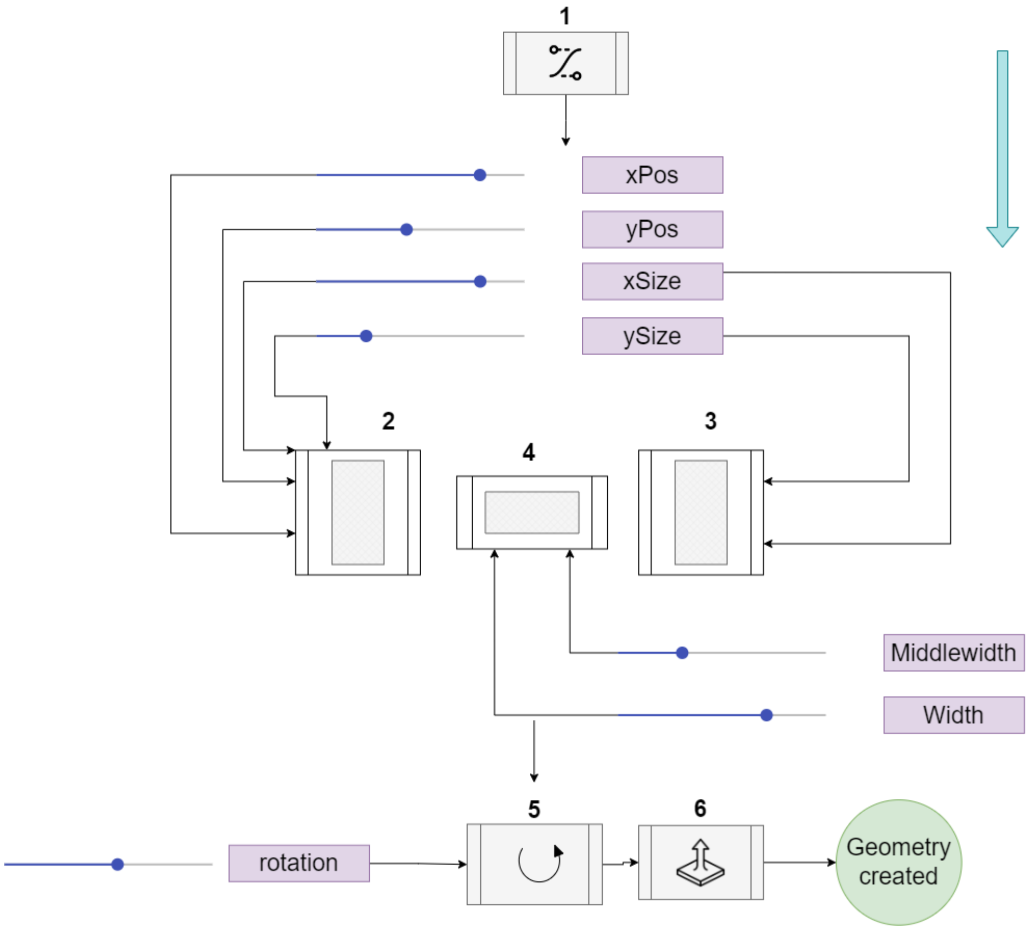


Figure 3.9: Geometry creation of H-shape

3.2.3 Integrating BIM Elements

Honeybee is used to integrate BIM elements, such as windows, shades, and rooms, into the geometry. This is mainly done to allow simulations on the building, as EnergyPlus needs Honeybee components to function.

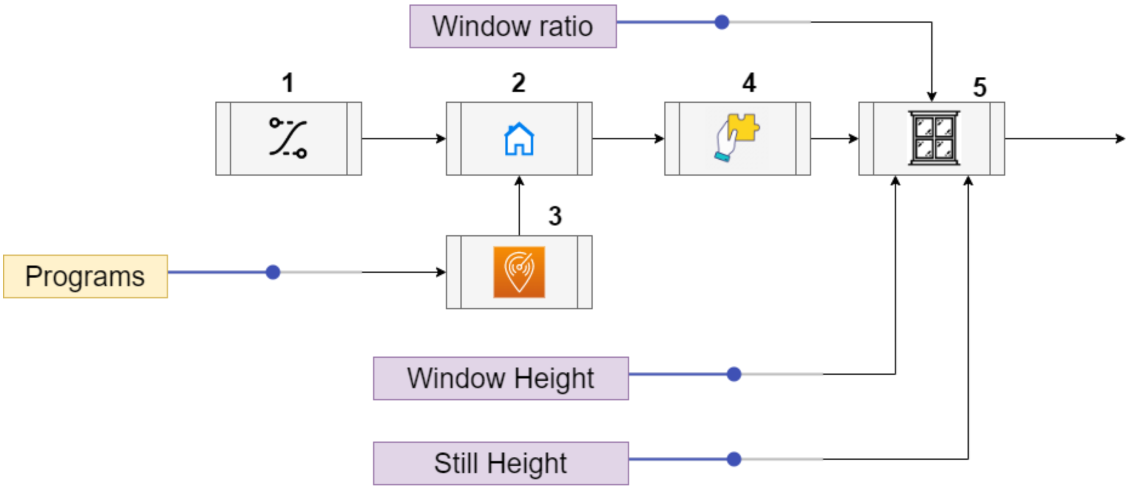


Figure 3.10: Honeybee workflow

When a functioning geometry has been created, as described in Section 3.2, we proceed with integrating the Honeybee components. The workflow of implementing honeybee components is shown in figure 3.10. The scripts for the Honeybee process can be seen in Appendix F.

It is important to notice that the geometry created using the floorplan and the building shape approaches can be extruded so that rooms are created with specific heights. In practice, each geometry surface is extruded such that n number of stories are created with a specific height at each story. in Appendix E, the scripts for extruding the geometries can be seen. The floor approach has the opportunity to extrude each individual room with a specific height. In contrast, the building shape approach can only extrude the geometry with the same height since no rooms are implemented in the geometry. The finished extruded geometry **1** is then sent to the honeybee room component **2**, which converts solid Rhino geometry to BIM elements, such as walls, interior ceilings, and roofs. The Honeybee room component allows several properties to be added. HVAC systems are set by a Boolean operator specifying if the room should have an HVAC system. Different programs can also be added, choosing from a pre-defined Honeybee library containing some generic programs, such as *Large Office* and *Retail* affecting design factors in the corresponding room. Programs contain different properties for the Honeybee rooms, such as material use and HVAC settings.

For the building shape design approach, rooms are not initially created. Therefore, the calculations will not be as precise since all buildings will contain a specific amount of rooms in each story. An estimation, on the other hand, may still be created. For the floor plan design approach, rooms are initially created. Therefore each story will contain a specific amount of rooms, emulating the interior of actual buildings.

To ensure that interior walls match the different rooms, the faces are split, and their surface area is matched to the adjacent walls **4**. This is done to ensure a correct energy model for multiple rooms, as conductive heat flow does not occur correctly if the walls do not match.

The layout of the windows is controlled with three values, the window wall ratio, window height, and sill height **5**. This allows the optimization model to control key aspects of the window properties during optimization. The same procedure is followed when integrating shades for each window. The shades are created as planes next to the windows, where depth, angle, and starting position can be altered. In Figure 3.11, a building has been configured with windows and shades. The optimization model can again change these parameters and, together with the apertures, have a significant say in the energy consumption of the whole building.

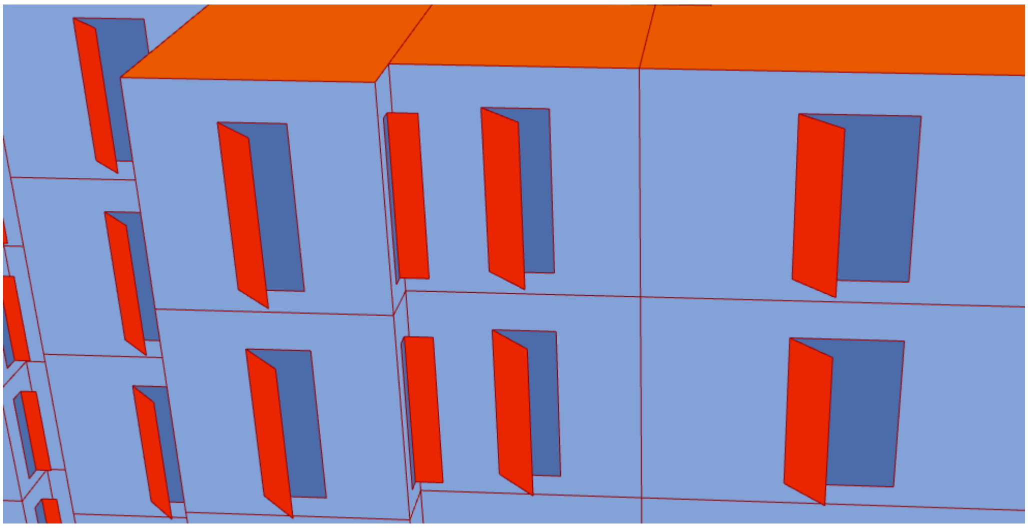


Figure 3.11: Shades and apertures on a building. Note the ratio and angle of the apertures and shades.

During the model creation the following Honeybee components have been added:

- Exterior walls
- Interior walls
- Roof
- Interior floor
- Windows
- Window shades
- HVAC system
- Interior ceilings

3.2.4 Simulation

Honeybee enables Grasshopper to communicate directly with simulation engines such as EnergyPlus and OpenStudio, described in Section 3. As EnergyPlus is a quicker approximation of the energy consumption, this engine is used for optimization in conjunction with the optimization model. OpenStudio produces more detailed results and is therefore used to test assumptions and final results. The scripts used for simulation can be seen in Appendix G.

EnergyPlus takes the Honeybee model as input, together with a .epw file, to generate correct weather data. The .epw file is gathered from the EnergyPlus website containing weather data from different geographical locations worldwide. The simulation component has several different outputs corresponding to different types of energy consumption, such as total load, cooling, and heating.

Another simulation done in this case study is to calculate the total solar radiation on the building facade. Ladybug's radiation analysis component is used to calculate the total solar radiation on the facade. It uses a sky matrix, calculating the MRT as described in Section 2. The sky matrix is created from the .epw file and data describing the simulation's time period. The geometry is taken in as Rhino geometry, allowing the use of neighboring buildings to provide shade. The total radiation on the geometry is returned when the simulation is complete.

3.2.5 Optimization

The optimization is done through Octopus, previously described in Section 2. For the case study, three fitness functions have been selected for optimization; minimizing the total output used by the building during a whole year, maximizing the floor area, and maximizing the total solar radiation on the facade.

Total output encompasses cooling, heating, lighting, and electric equipment. Since the solution does not add lighting and electrical equipment, these values are approximations based on the size of the building. Solar radiation calculates sun exposure on the exterior of the building. The energy consumption and the solar radiation result values are in kWh/m^2. As the total energy consumption and solar radiation is used for objective functions, the value is multiplied by the total floor area of the building. The objectives used can be seen in Table 2, with corresponding genes affecting their result.

	Energy Consump.	***Floor Area***	***Solar Radiation***
Genes	Orientation Placement Extrusion Height Window Properties Shade Properties HVAC Width/Length	Extrusion Height Width/Length	Orientation Placement Extrusion Height Shade Properties Width/Length

Table 2: Table depicting the objective functions used, with corresponding genomes affecting the objective score.

Octopus allows the user to change several parameters affecting the optimization process. For optimization algorithms, there are two choices given by Octopus, SPEA-2, and HypE. Bader and Zitzler [2011] has concluded that the HypE is generally the superior MOGA. Therefore HypE has been used for optimization in the solutions presented here. For mutation factors, there are three choices, polynomial, alt. polynomial, and hype mutation. As hype mutation allows for more diversification in the solutions generated, HypE is used as the mutation operator.

As the geometry generated in the floor plan design approach utilizes MagnetizingFPG to generate the floor plan solution, a distinct optimization workflow is put in place. The floor plan generator utilizes a quasi-evolutionary method for placement, meaning that Octopus can not control some of the parameters. This results in the floor plan being different with the same parameters set in Grasshopper. Octopus is therefore controlling the placement of the building, the rotation, amount of stories to be extruded, and the window properties. To retrieve the same buildings simulated in Octopus, a solution had to be made. To achieve this, the building's Honeybee rooms are saved in data recorders each time Octopus changes genes. To segment the different buildings, a dummy room is inserted at the end of each iteration, serving as a identifier for the script. The corresponding objective functions are then iterated through to find the Pareto dominant solution. When the best objective functions are found, the index is saved to be used for finding the corresponding building. The python script iterates through the rooms until the correct index is found, using the dummy rooms. Then the rooms corresponding to the best result are extracted and visualized in the Rhino viewport. The corresponding energy consumption and sun exposure are given as well. For later analysis, the gene values are also given for each solution.

3.3 Extendability

The extendability of a solution is a crucial part of developing software [Henttonen et al., 2007]. The objective functions used in this paper focus on environmental optimization. However, as different situations will call for different building evaluations, the need to create a modular and extendable solution is essential. The solution proposed in this paper is built to be extendable and modular. Different objective functions can easily be added or removed from the optimization model. Easy access to change evaluation functions is an important factor as different building projects often find some objectives more important than others. Using the developed parameterized design approaches used in this PoC, new evaluation functions can be found directly within the Grasshopper script and added as evaluations. For insistence in this PoC, total energy consumption is used as evaluation. If only cooling load is important for the user, this can be replaced with total energy consumption.

The Grasshopper script created for this study utilizes several different plug-ins, focusing on their respective part of the building design. An effort was put into making the scripts as understandable as possible, given the complexity and size of the script. Modules contributing to a specific objective were grouped together with short names describing their function. The script follows the natural progression of the workflow, going left to right. Understanding the script will help if new evaluations not provided within the script should be implemented.

3.4 Summary

In order to automate the process of building design, key changes have to be in place. The solution should need minimal input from the user, as users with varying knowledge of building design should be able to use the program. Therefore, the geometry should be parameterized, allowing for manipulation by the EA. Essential parts, such as apertures, should be controlled by the solution, generating data for the EA. This allows the optimizer to make calculated decisions based on simulated data. The users can then retrieve the building that meets their requirements.

4 Case Study and Implementation

How can a solution be tested?

4.1 Field A5 Bjørvika

Field A5 at Bjørvika Oslo is selected as the location for this case study. The location is currently used as a parking space for buses and owned by *HAV Eiendom* [Hav, 2022][Ådne Homleid, 2022]. Since 2021 *HAV Eiendom* has started planning the creation of a building design solution for the field together with *Studio Oslo Landskapsarkitekter and Byantropologene*. The A5 location is highlighted in green in figure 4.1.

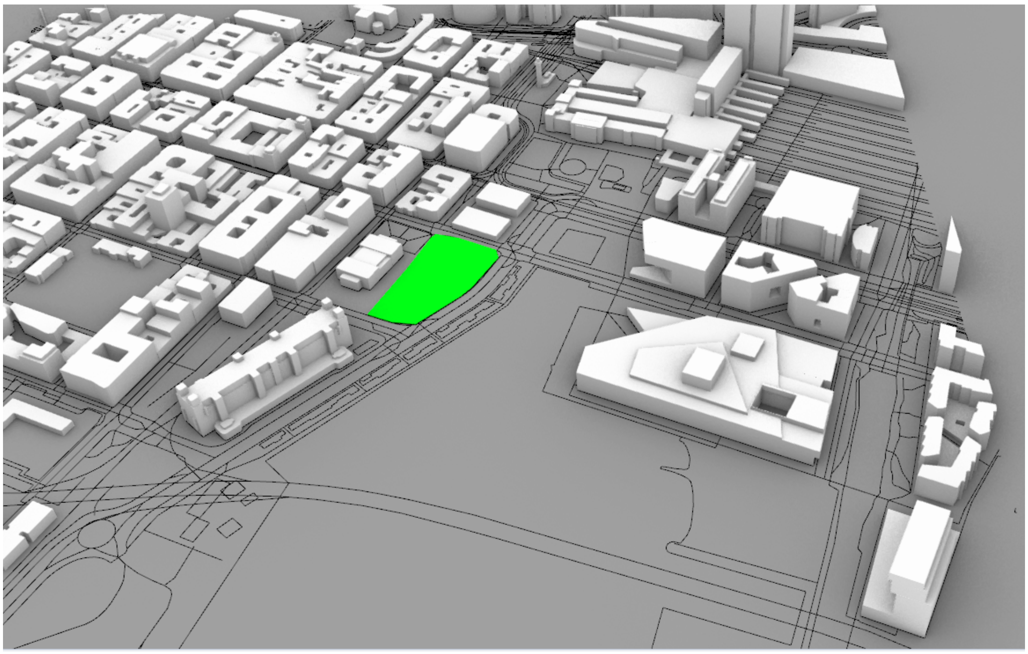
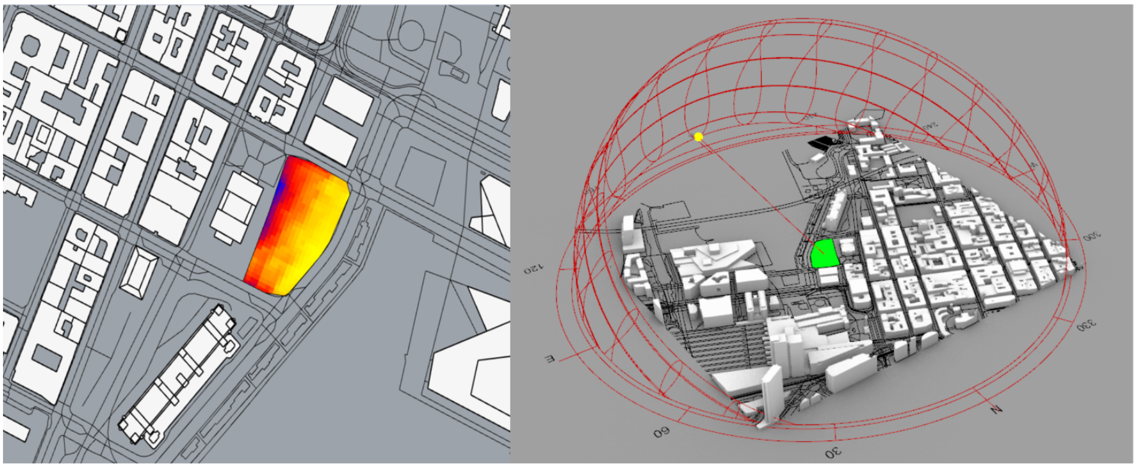


Figure 4.1: Field A5 Bjørvika

Bjørvika has changed a lot in the last decades, and as late as 2008, the Opera in Oslo, east of the A5 field, was built. When the politicians of Norway made a plan around the construction of Bjørvika, great emphasis was placed on facilitating jobs near Oslo Central Station [Ådne Homleid, 2022]. Environmental factors were the key reason for this decision as less traffic will occur for Oslo citizens if more offices are located in the central parts of the city. As environmental factors are an important part of creating office buildings at the A5 field, the building should also be optimized to minimize energy consumption.

Before starting the development of a proposed building design, in regards to energy consumption, it is important to explore the climate around the building field. As described in SOTA, ladybug tools can be used to visualize weather data, and environmental simulations can be conducted on a specified area to analyze the environmental impact. When optimizing the design model, having knowledge about the building field can help select the optimized result. Understanding the results will also allow the user to make knowledge-based decisions to help select the optimal solution.

This thesis uses energy consumption, area, and solar radiation as evaluation functions. Therefore, a visualization of the sun exposure on the A5 field and a visual representation of the sun path can be helpful for understanding results generated from the optimization model. In Figure 4.2 (**b**), the sun path around field A5 at Bjørvika during a whole year is shown. In the image, the sun (in yellow) is shown on the 21st of December, at 12 pm. The red curves indicate the sun's path for the rest of the year. It seems like most of the sun during a year will come from the south of the building site. How much direct sunlight is exposed in a year is shown in figure 4.2 (**a**), yellow indicates lots of exposure, while blue indicates less.



(a) Direct sunlight on study area

(b) Sun path around study area

Figure 4.2: Visualization of direct sunlight and sun path on study area

Given information about the sun's path and radiation on the study area, assumptions about the resulting building can be made. Norway has a maritime climate with cold winters and mild summers [Nor, 2020]. From this, one may assume that the building should be located around the most sunlight, as seen in figure 4.2 (**a**).

4.1.1 Proposed Design by *HAV Eiendom* at Field A5

To establish a comparison between the generated solutions and a real-life solution, an actual proposal for Trollbugata has been used. *Hav Eiendom*, *Grape Architects*, *SO-LA*, and *Byantropologene* have proposed a design solutions for the A5 field. The proposal will be modeled and run through the same simulations as the approaches presented in this thesis. The 3D model of the proposed building design can be seen in Figure 4.3 below. It is important to notice that the 3D model used in this thesis is a simplified model created to test the viability of the PoC. The model was created using an illustration booklet provided directly from *HAV Eiendom*. The simulation outputs will be discussed and compared to obtain a view of the viability of the solutions. A comparison of the results can be seen in Section 5.

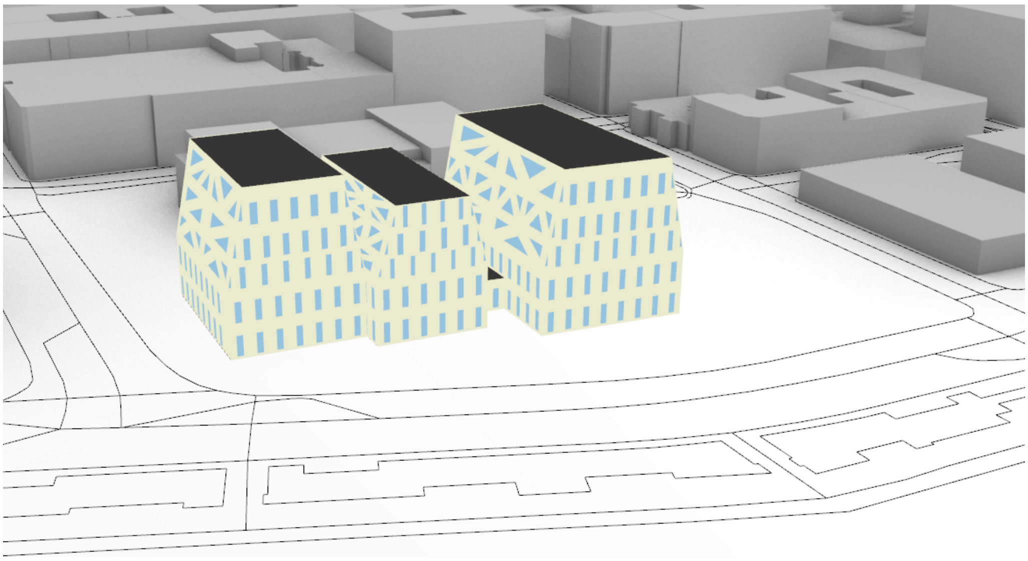


Figure 4.3: 3D model of the proposal from *HAV Eiendom*

The same objectives will be used for all proposals to ensure that the case study is fair. They will use the same surrounding geometry gathered from CADMapper.

4.2 Optimization Model

As described, the optimization model used for this case study is the Hypervolume Estimation algorithm (HypE) with HypE mutation. Since the two design approaches in this study vary in complexity, the termination condition used in the MOGA is a 24-hour runtime. This termination condition is implemented due to the restriction on available hardware. If adequate computational power was provided in this case study, solutions closer to the global optimum would be obtained using convergence as the termination condition. Elitism is set to 0.5, allowing the best individuals to be selected as parents without undergoing mutation while still allowing the mutation to happen on the individuals. Mutation probability is set relatively high to allow a higher diversification. According to Deb [2011a], the mutation probability should be set to $1/n$ decision variables. As the average genes of all design approaches are 7.66, the mutation probability should be $1/7.66 = 0.13$. As we want a more comprehensive solution space, the probability was set to 0.2. The mutation rate in Octopus does not affect the probability of mutation but rather the strength of the gene change. A high mutation rate means the genes change more when altered, creating greater diversity in the solutions. It is therefore set to 0.9. The crossover rate is set to 0.8 to allow chromosomes to change some of their parts, increasing the diversity of the solution space. The values discussed can be seen in table 13.

Parameters	Value
Elitism	0.5
Mut. probability	0.2
Mutation rate	0.9
Crossover rate	0.8
Algorithm	HypE
Mutation selector	Hype mutation
Population size	30
Max generations	Not set

Table 3: Values used in the HypE optimisation model

The evaluation functions used in this study, as described in Section 3 is as follows:

1. Maximize Sun radiation.
2. Minimize Energy consumption.
3. Maximize Total Area.

As described in SOTA, the most important aspect of green buildings is the total energy consumption of the building. This evaluation is affected by multiple factors, including the overall sun radiation on the building and the area. Sun exposure is also added as an evaluation function to facilitate the installation of solar panels.

The area plays an important role when optimizing solar radiation and energy consumption since there is a strong correlation between these properties. Since minimizing energy consumption and maximizing solar radiation are antagonistic objectives, they help keep the optimization process in check, never allowing one objective to dominate the others. The reason is that the design approaches used in this thesis allow the buildings to change their overall area. As solar radiation increases when there is a larger facade, while energy consumption minimizes when the overall size is lower, they are both dependent on the building area. For this reason, the area is implemented as the third objective function.

Octopus minimizes every evaluation function that is connected to the plugin. Therefore dividing the number by one will make Octopus maximize the given parameter. For this reason, both energy consumption and the total area are divided by 1.

Both building design approaches were run on an Intel(R) Core(TM) i7-10875H CPU @2.30GHz. The generations created using the optimization model vary from 14 to 18, depending on the design approach. As described average runtime for the optimization model is 24 hours.

4.2.1 Parameterization

The genes used in each design approach can be seen in table 4. The interval of the different genes is set such that almost every combination of the genes will create a proposed design. For instance, the position genes used for creating the floorplan are set such that it will not exceed the boundary curve by setting the threshold to (-15) - 15. The genes have been selected to maximize the diversity the MOGA can create. While all approaches use different genes to generate the geometry, they all share the genes controlling windows. The process is identical for all approaches, as the Honeybee windows are added after geometry creation. The intervals for each gene have been set such that there is a large amount of variation for the building layout. These intervals could be decreased to decrease run time for the optimization, thereby reducing the possible solution space. As discussed in Section 2 there are passive and active design methods for minimizing energy consumption. Many active design elements are intricate, and the individual nature of each method creates difficulties in implementing automation for the installation. Due to the passive design method yielding larger benefits for reducing energy consumption, the genes are focused on altering the passive design elements, such as windows and parameters relating to the facade [Yu et al., 2019].

Design approach	Gene	Threshold	Interval
Floorplan	Position_X	(-15) - 15	1
	Position_Y	(-15) - 15	1
	Rotation	0 - 360	1
	Stories	3-10	1
C-shape	Thickness	(-31) - (-1)	1
	Opening pos.	0 - 19	1
	Snake length	0 - 10	1
H-shape	Position_X	(-100) - 50	1
	Position_Y	(-100) - 50	1
	Middle width	1 - 50	1
	X_size	1 - 50	1
	Y-size	1 - 100	1
	Width	0 - 50	1
	Rotation	0 - 360	1
All approaches	WWR	0.2 - 0.7	0.1
	Window height	1.5 - 3.5	0.1
	Sill height	0.2 - 1.5	0.1

Table 4: Genes used for each design approach

4.3 Summary

Field A5 at Bjørvika is used to test the methods created in this thesis. Weather simulations have been conducted on the field that will later be used to examine the building designs selected from the optimization model. These designs will thereafter be compared with a proposed architecture by *HAV Eiendom* on the same field. The optimization model used is the HypE algorithm that is terminated after 24 hours. Lastly, the gene thresholds for the different design approaches have been set to best fit the A5 field.

5 Results

Which assumptions can be made from the optimization?

As explained in Section 4, three different design approaches are used for generating the geometry in this PoC. In addition to the developed design approaches, a 3D sketch has been developed of the proposed design created by *HAV Eiendom*. The proposed design can be used as an indicator to see how well the optimized design solutions created in this thesis are compared to the solution made by *HAV Eiendom*. The goal of this section focuses on showcasing the different results from the optimization model and providing relevant observations regarding the feasibility and key differences.

As described, the proposed architecture uses the MOGA optimizing model, producing a Pareto front optimizing for the selected objective functions. As described in Section 2, the Pareto front consists of all combinations of genes, called genomes, that dominate the other genomes created in earlier generations. In other words, these genomes represent the best solutions generated during the optimization process. Since three evaluation functions are used in the case study, a Pareto front will be made in a 3-dimensional space. When comparing two and two objectives, all pairs will be antagonistic as described in section 4. Therefore, viewing the Pareto front in a two-dimensional space and comparing every evaluation with each other can give a more comprehensive understanding of the final results.

The results in this thesis are presented such that each design approach creates its respective Pareto front dependent on the same evaluation functions. The Pareto fronts for the three approaches can be seen in Figure 5.1, 5.2, and 5.5. Every single point represents a genome; yellow points denote earlier generations' genomes, and red denotes the Pareto front individuals. As stated, both solar radiation (SR) and total area (TA) are objective functions that are minimized; therefore, these objective functions are divided by 1. Subfigure **(a)** represents the correlation between solar radiation and energy consumption (EC). **(b)** shows the correlation between solar radiation and floor area. **(c)** displays the correlation between energy consumption and floor area, and **(d)** presents the Pareto front with the history of earlier individuals. The three-dimensional view of the Pareto front gives an understanding of the correlation between all objectives.

For each Pareto front created (Floor plan, court/C-shape, and H-shape), one solution is selected to compare the different results and see what output can be created within each design approach. In this thesis, a genome is selected such that there is a compromise between the three evaluation function. For instance, if the area varies between 9000 and 15000 m^2, a solution will be selected with an area as close to its average $(15000m^2 + 9000m^2)/2 = 12000m^2$ as possible. This procedure of selection will be done for all three evaluation functions. It is important to notice that this is only one way of selecting solutions, as all genomes in the Pareto front are optimal solutions.

The selected solution will be presented, displaying evaluation, and gene values for each selected building. These values will be the foundation for comparing the approaches in the Discussion. In addition, the building will be presented to give a visual representation of the final solution.

5.1 Floor Plan Design Approach

The generated Pareto front for the floor plan design approach with history can be seen in Figure 5.1.

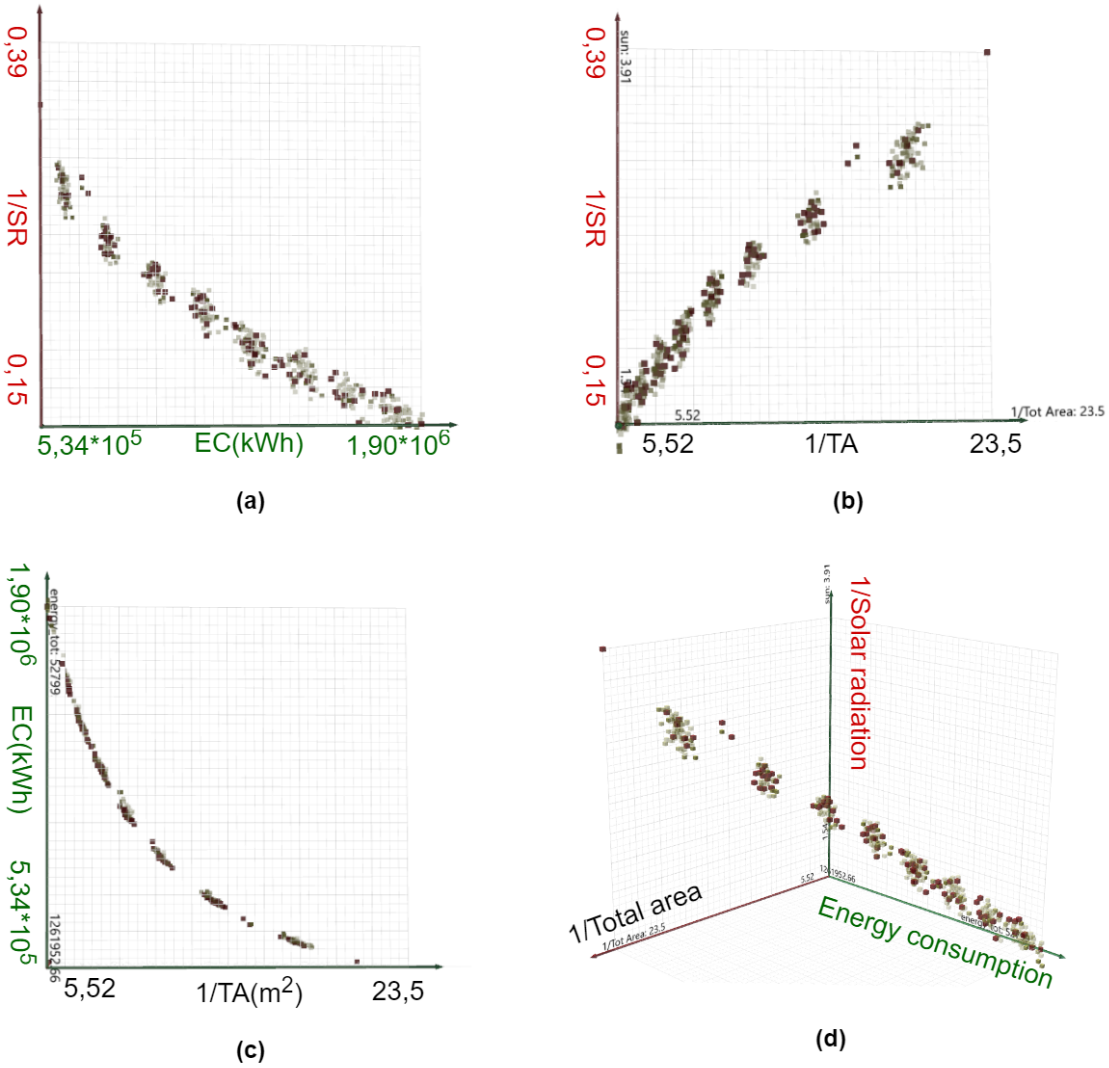


Figure 5.1: Generated Pareto front from the floorplan approach

Observations that can be made from Figure 5.1 regarding the floor plan design approach:

1. The height parameter heavily influences the three objective functions. Thereby, one

can see 8 clusters due to the height interval being set to 1 and the threshold going from 3 to 10, such that eight distinct stories can be created.

2. Total floor area and solar radiation have a strong positive linear correlation (**b**).
3. Solar radiation and energy consumption have a strong negative linear correlation (**a**).
4. Energy consumption and total floor area have a strong positive hyperbolic correlation (**c**).

The solution was retrieved using the Pareto front. As described in section 3 the floor plan design approach saves solutions in data collectors to retrieve the produced building corresponding to the objective functions found in the Pareto front. The values of the selected genome can be seen in Table 5. The selected building has a high total area and, therefore, also relatively high energy consumption.

Area	**Sun exposure**	**Energy consumption**
16869 m^2	605003 kWh	1.76 * 10^6 kWh

Table 5: The final results for the objective functions.

Genes	**Value**
Position X	-13
Position Y	-9
Rotation	194
WWR	0.2
Window height	2.1
Sill height	1.3
Stories	9/10

Table 6: Genes from the selected floorplan design

The values of each gene can be seen in Table 6. The window genes create small windows, as can be seen by the WWR gene. The finished 3D model can be seen in Figure 5.2. The building is centered to the east of the plot, facing south. Nine stories were created for all rooms, except the hall, being extruded to 10.

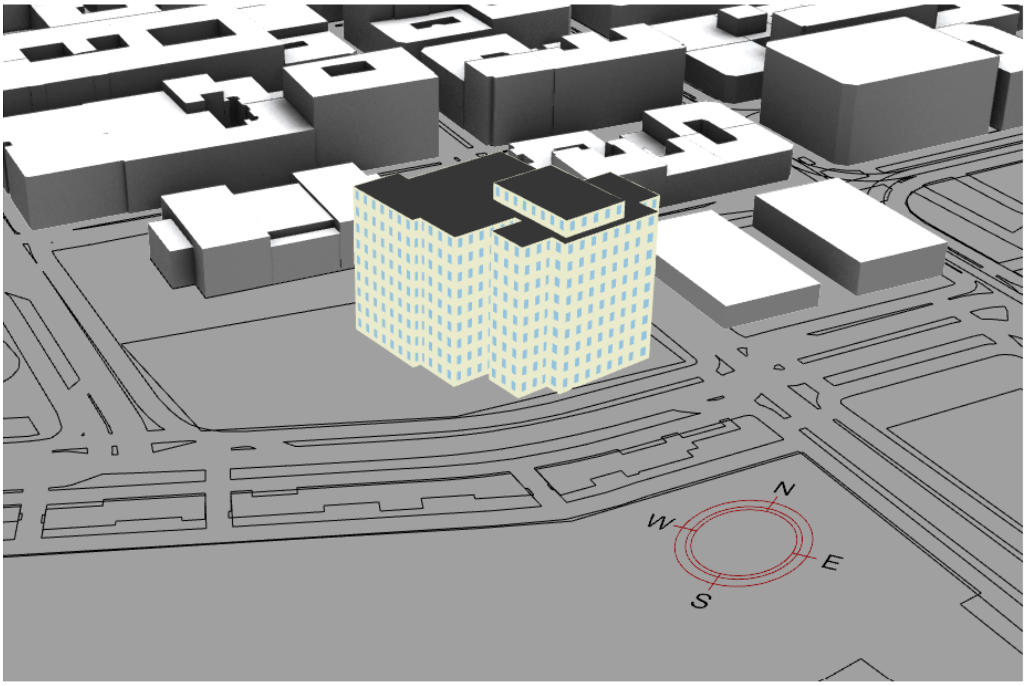


Figure 5.2: 3D model of the selected floorplan design

5.2 C/courtyard Design Approach

The generated Pareto front for Courtyard and C-shaped building with earlier history can be seen in figure 5.3.

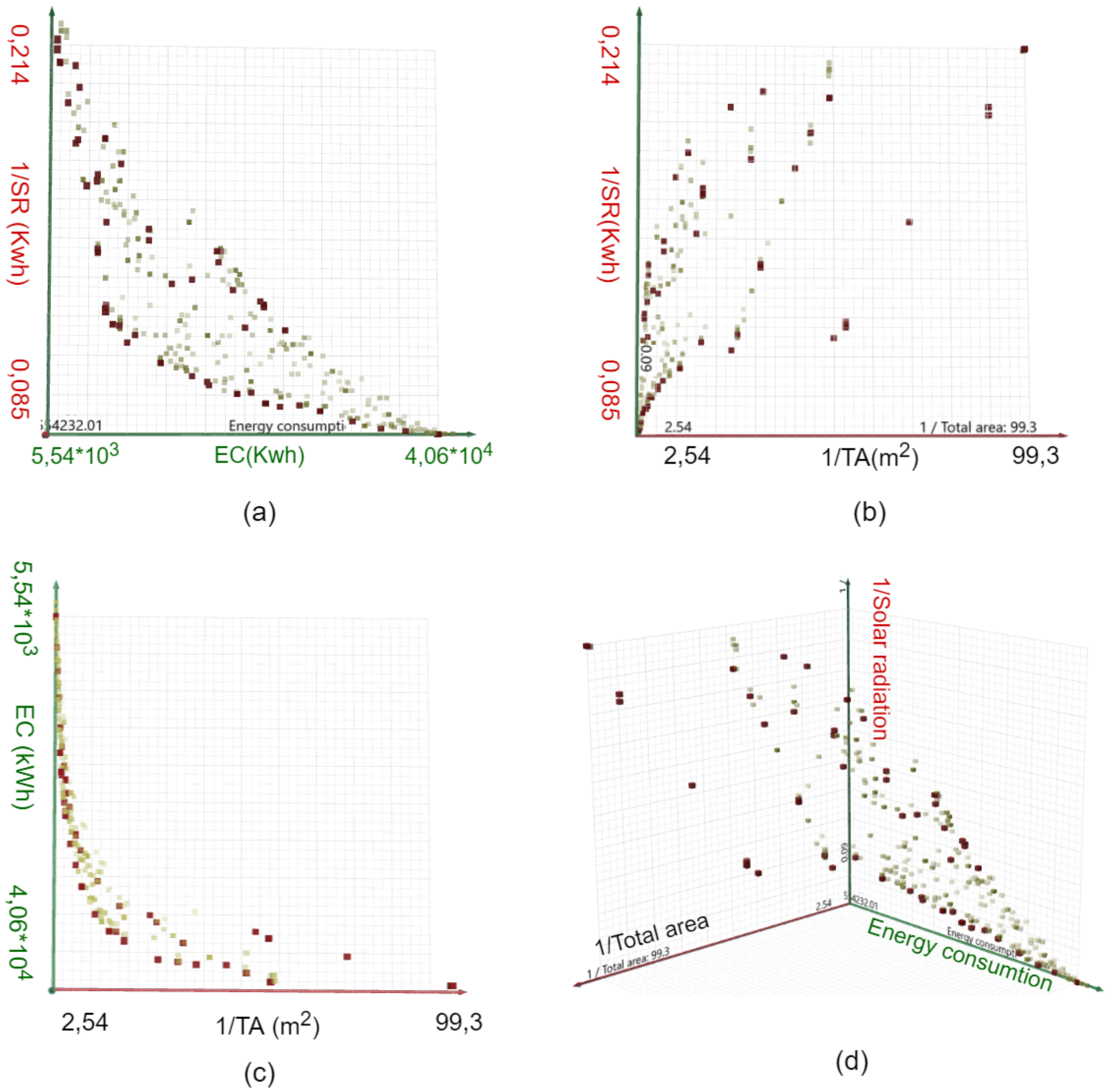


Figure 5.3: Generated Pareto front from the floorplan approach

Observations that can be made from Figure 5.3 regarding courtyard/C-shape:

1. Energy consumption and total floor area have hyperbolic correlation
2. Solar radiation and total area **(b)** can be viewed as having a hyperbolic correlation. This correlation is a weak positive, considering values with the same total area can have a wide interval of different solar radiation values
3. Solar radiation and energy consumption **(a)** have a weak positive hyperbolic correlation

From the Pareto front, one solution was selected. The values for the evaluation functions can be seen in Table 7. As seen in the table, the developed geometry has a lot of sun exposure regarding the total area.

Area	**Sun exposure**	**Energy consumption**
12811 m^2	749794 kWh	1.60 * 10^6 kWh

Table 7: The final results for the objective functions.

The values of each gene can be seen in table 8. The thickness is relatively small, and the geometry creates a C-shape instead of a courtyard since the *courtyard removal* is not set to 0. The genes controlling the window properties, such as WWR, are set to 0.2, creating small windows.

Genes	**Value**
Thickness	-12
Opening position	0
Courtyard removal	8
WWR	0.2
Window height	1.6
Sill height	0.4

Table 8: Genes from the selected Courtyard/C-shape design

The finished 3d model of the building can be seen in figure 5.4. The main facade faces south, while the facade facing north has been removed from the final solution. As shown in Figure 5.4, the most significant amount of solar radiation is found on the south side of the plot, where the building is located. As described, the building size thickness is slim, opting for a lower floor area.

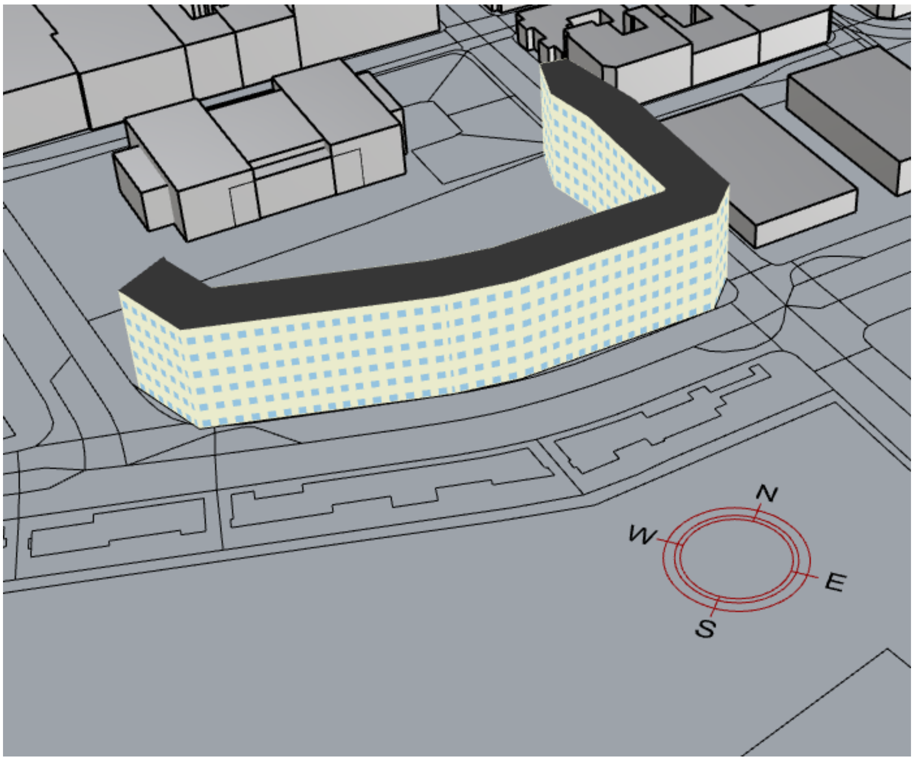
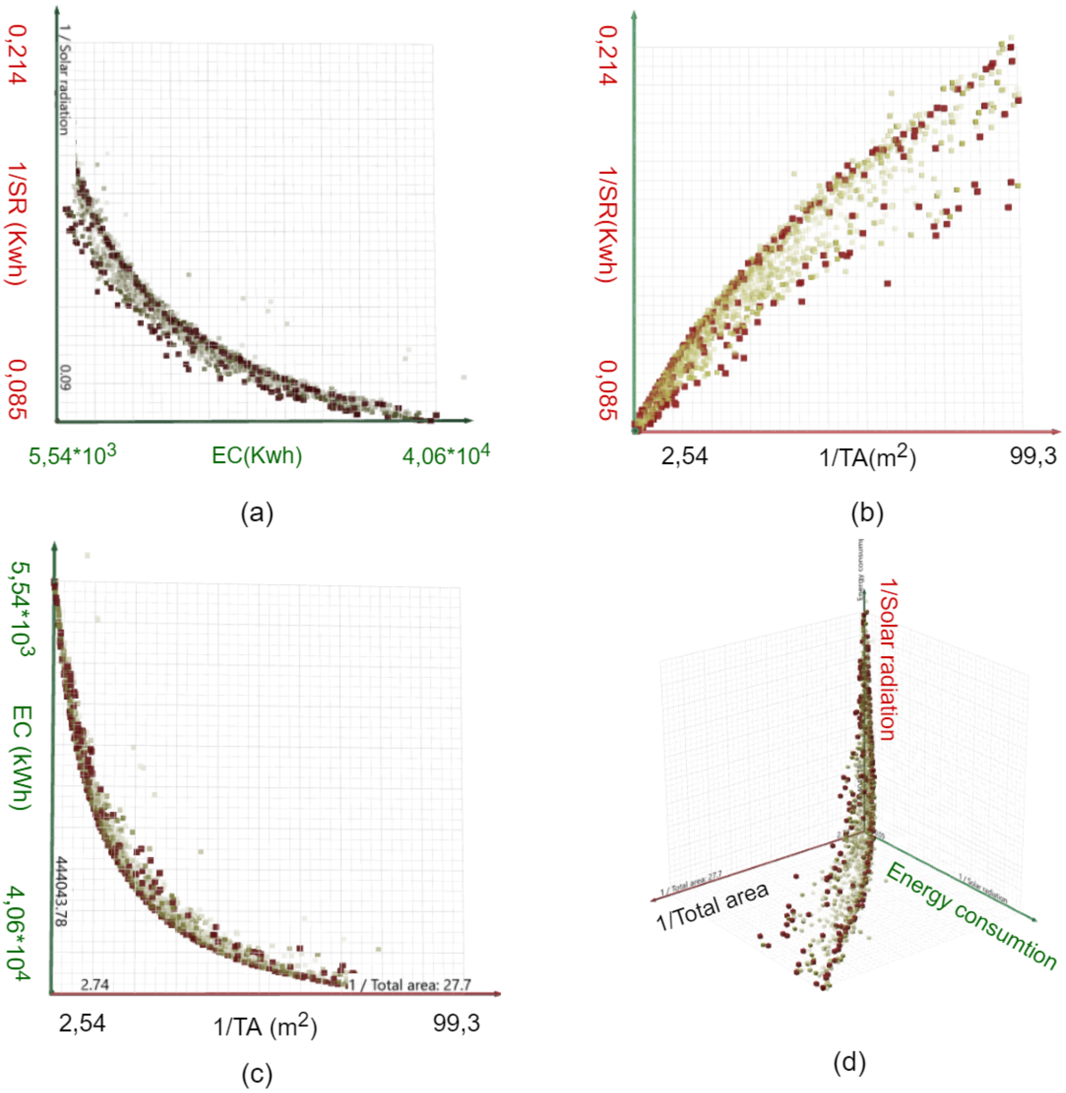


Figure 5.4: 3D model of the selected courtyard/C-shape design

5.3 H-shape Design Approach

The Pareto front generated with the H-shape can be seen in figure 5.5.



(a) (b)

(c) (d)

Figure 5.5: Pareto front from H-shape design

Observations that can be made from Figure 5.3 regarding H-shape:

1. Energy consumption are heavily dependent on both solar radiation and area.
2. There is not much diversity in the graphs where it seems like a convergence in the graph is being created
3. both graph (**a, c**) have a strong form of hyperbolic correlation.
4. graph (**b**) has a positive linear correlation.

From the Pareto front, one solution was selected. The values of the objective functions can be seen in Table 9. The total area and the sun exposure are important values to notice, which are relatively small.

Area	**Sun exposure**	**Energy consumption**
10911 m^2	471146 kWh	1.22 * 10^6 kWh

Table 9: The final results for the objective functions.

The values of each gene can be seen in table 8. The position of the gene is located such that the center of mass of the H-shape is outside the boundary curve. Therefore the building will not represent an H-shape.

Genes	**Value**
middleWidth	10
xPos	15
yPos	-54
xSize	42
ySize	99
Width	33
Angle	169
WWR	0.2
Window height	1.8
Sill height	1.3

Table 10: Genes from the selected H-shape design

The values of each gene can be seen in table 10.

The finished 3d model of the building can be seen in figure 5.6. The building is situated in the area with the most solar radiation facing south. The picture shows how the H-shape is positioned and which part of the shape is extruded to the final solution.

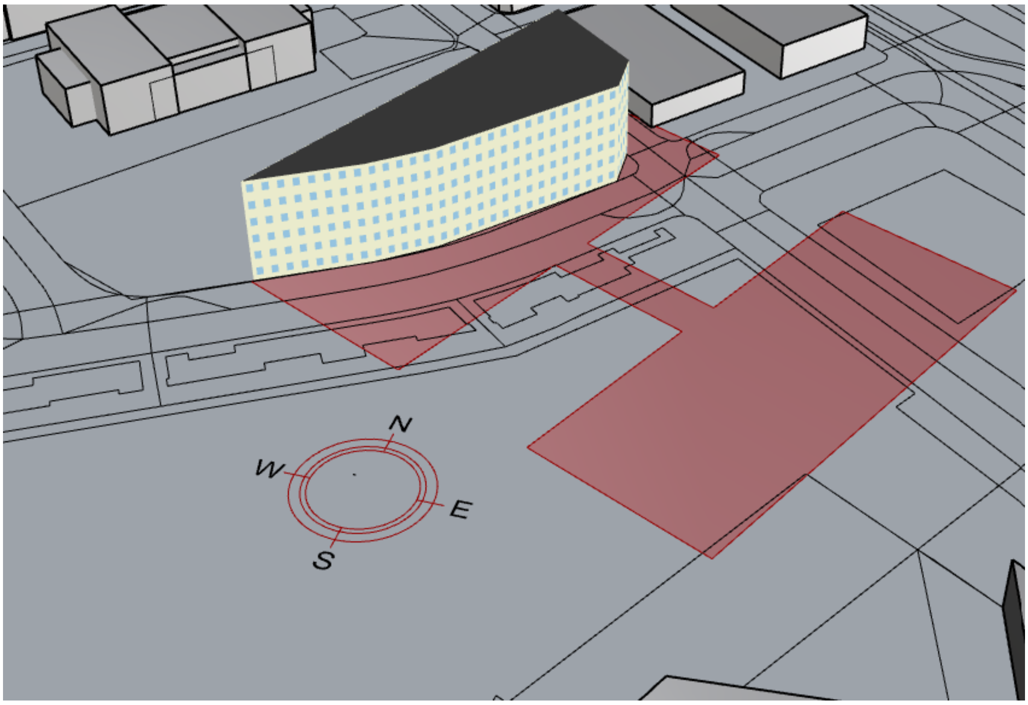


Figure 5.6: 3D model of the selected H-shape design

5.3.1 Proposal from *HAV Eiendom*

To analyze and compare the proposal from *HAV Eiendom* presented in Section 4, the same simulations have been run on the building. The simulation results are shown below in Table 11.

Area	**Sun exposure**	**Energy consumption**
13333 m^2	475949 kWh	$2.52 * 10^6$ kWh

Table 11: The final results for the objective functions.

5.4 Comparison

Table 12 is a comparison between the objective functions for all design approaches. Energy per area and sun exposure per area are added to highlight the energy consumption and sun exposure disregarding the overall area. As can be seen in the table, the floor plan approach has the highest area while at the same time having the lowest energy consumption per area. The C-shape approach has the highest sun exposure, mainly due to the long facade with relatively little area. While having the lowest area, the H-shape also has the lowest sun exposure due to much of the area being in the center of the building and not being radiated. All three design approaches beat the proposal from HAV Eiendom when it comes to energy consumption. HAV Eiendom has a marginally higher sun exposure than H-shape but with a much larger area.

Approach	Area (m^2)	Sun (kwh)	Energy (kwh)	Energy/area (kwh/m^2)	Sun/area (kwh/m^2)
HAV Eiendom	13333	475949	$2.52 * 10^6$	202.5	35.7
Floor plan	16869	605003	$1.76 * 10^6$	104.4	35.8
C-shape	12811	742794	$1.60 * 10^6$	124.9	58.0
H-shape	10911	471146	$1.22 * 10^6$	111.8	43.2

Table 12: Comparison of the selected building designs

5.5 Summary

The three approaches created show some correlation between the objective functions, where the area is the most influential. The three objective functions show that the floor plan design approach has the lowest energy usage per area. The C-shape approach seems superior for maximizing sun exposure, while H-shape produces acceptable values for all the objective functions. All three approaches beat the *HAV Eiendom's* proposal on sun and energy consumption.

6 Discussion

What are the limitations and possibilities for parametric building design focusing on environmental factors?

This section focuses on the results presented in Section 5. Discussion around the findings for each design approach and comparing their feasibility will be made. Moreover, the architecture and design choices will be discussed, looking at shortcomings and possible alterations that can be made to improve the solution.

6.1 On Results and Observations

The results found in this study can be divided into three parts for each design approach: The Pareto front, gene values, and the final 3d model. The produced Pareto fronts can be used to see how the different objective functions affect each other. Another important aspect for understanding the result is to look at the analysis conducted on the A5 field in section 4. Figure 4.2 shows the solar path and the solar radiation on the field depicting the sun's movements in the sky. Due to the sun's movement, the south-eastern side of the plot receives more radiation. This information can be used to gather knowledge regarding why the MOGA has chosen the genomes selected in this study, creating the 3d models presented in the results.

6.1.1 The Pareto Fronts

Looking at the Pareto fronts presented in Section 5, one can see that the area affects all the other evaluation functions integrated in this study. Solar radiation most often increases if the total area increases as more of the building facade will be exposed to solar radiation. This can be viewed when looking at all the Pareto fronts, sub-figure **(b)** in the Results. How much the area affects the total solar radiation of the buildings seems to vary among the three building design approaches. The approach with the weakest correlation is the C/courtyard approach, where one can see that buildings with the same total area can have many different values of sun exposure. On the other hand, the energy consumption and total area have a strong hyperbolic correlation for all the design approaches where one can see that more area will increase the total energy consumption of the building.

The Pareto fronts showing the correlation between solar radiation and energy consumption in sub-figures **(a)** shows a positive correlation, where an increase in solar radiation results in higher energy consumption. As described, both these variables are heavily dependent on the total area. When the solar radiation increases, so will likely the total area, and when the total area increases, energy consumption will be higher. Looking at the Pareto front comparing solar radiation and energy consumption, one can see that a building with higher energy consumption will have more solar radiation as well.

Comparing all objectives is important when interpreting and selecting a solution from the Pareto front. Utilizing the 3D dimensional view of the Pareto graphs can be beneficial for selecting genomes, where the correlation between all objective functions can be studied. As this study uses three evaluation functions, a 3-dimensional view of all the genomes is possible.

Looking at Figure 5.1 in Section 5 some assumptions can be made from the graphs. Stories have the most impact on the objective functions of all the genes used by the floorplan design approach. As the floor layout is set by a given number of rooms, the total area will largely remain the same for each story. The room instances only fluctuate a small amount from the set value. As total energy consumption and solar radiation are heavily dependent on the size of the building, adding and removing whole stories will have a severe impact. This creates clusters in the Pareto front, where the differences in each cluster can be attributed to different rotations and the slight change in room sizes.

6.1.2 Gene Combinations and Final 3d Models

The floor plan design approach is placed on the east side of the plot. The entrance point, often situated at the center of the building, was placed near the south-eastern part of the field. From Table 6, the final genes for the position were 13 units toward the water and 9 units to the east of the boundary center. Due to the way the floor plan is generated, the layout of the floorplan is not affected by the MOGA trying to maximize the solar radiation. Therefore, the solution cannot fully utilize the simulation by situating more area in the southern part of the field.

The MOGA controls all aspects of the building's overall shape, orientation, and position in the building shape design approach. This allows all genes to be affected by the simulation process. Both solutions produced using the building shape design approach have their position on the southern side of the plot. For instance, the C-shape solution has removed all facade on the northern side of the plot, maximizing the solar radiation. The H-shape was also situated on the southern side of the plot, trying to maximize the radiation.

6.1.3 Comparison

The performance of each approach is given in Section 5. One of the most important comparisons in this case study is the differences in energy per area and sun per area. Since all the solutions have a different total area, they give a lot of informative information about the building's performance. As discussed earlier, this thesis compares the three design approaches developed, against the proposed design created by *HAV Eiendom.* This design compares how an actual proposed architecture for the A5 field performs in comparison with the solutions created in this thesis, based on the three selected evaluation functions. It is important to notice that the proposed design created by *HAV Eiendom* does not use the same evaluations when the design was created. Therefore, one should see superior results regarding energy consumption and solar radiation for the proposed design solutions created for this study. *HAV Eiendom* tries to optimize their design solution for logistics of goods delivery, creation of a new park, and views from surrounding buildings, to mention a few. Since this study focuses on green building designs, energy consumption and sun exposure are used for evaluation functions. As discussed in section 3.3 the program architecture used in this PoC can change its evaluation functions. Therefore the evaluations used by *HAV Eiendom* could be implemented using the same program architecture presented in this thesis.

As seen in table 12, all the solutions selected from the Pareto fronts have better energy and sun exposure than the proposed architecture by *HAV Eiendom.* The floor plan design approach does have the lowest energy consumption per area. This could be because the floor plan solution will create compact building designs since the magnetizingFPG plug-in tries to minimize the distance between the generated rooms. This feature will also affect the solar radiation on the building since less building facade is exposed to solar radiation compared to the building shape design approaches. The building shape design approach that has the highest amount of solar radiation is the courtyard/C-shape design approach. This design approach will, as described, create a courtyard/C shape no matter the combination of genes. Since this approach creates a building around the boundary curve of the field, a lot of building facade will be exposed to the sun. Therefore, this design approach will create the building with the most sun exposure. On the other hand, the H-shape approach performs well on both energy consumption and solar radiation. The H-shape design selected from the Pareto front consists of a simple shape, having little facade to be radiated compared with the C-shape. In table 12 one can see that the selected C-shape has better sun per area than both the selected floorplan and H-shape approach.

Another important aspect is that all the solutions have different computation times. The computation time depends on how many combinations of genomes can be created using an optimization approach. In comparison, using a brute force method would increase the computational time dramatically. The solution space varies between all the proposed design approaches, where the most computationally heavy is the H-shape approach. Calculating the number of possible design solutions that can be created is fairly simple, multiplying the threshold interval for each gene. To recap, all the genomes and thresholds can be seen in table 4. For the H-shape approach, the number of combinations would be $1.6 * 10^{17}$ possible gene combinations. As each genome, on average, took approximately 40 seconds to finish, calculating sun exposure, total area, and energy consumption, the total time to run a brute force search on the solution space would be $2.05 * 10^{11} years$.

Since all models use the same Honeybee components for simulation, some assumptions can be made regarding the genes affecting the same properties for each design approach, in this case, window properties. All design approaches use the same genes within the MOGA to optimize the window shape and form. The genes *WWR*, *window height* and *sill height* all reach close to the same values. These genes only affect the total energy consumption, as floor area and sun exposure do not change based on these genes. The final solution for all design approaches resulted in small window sizes. The comparison between the genes can be seen below:

	Floorplan	C-shape	H-shape
WWR	0.2	0.2	0.2
Window height	2.1	1.6	1.8
Sill height	1.3	0.4	1.3

Table 13: The genes controlling the windows, retrieved from the final solutions.

There has been a trend in office design to build large windows or whole facades out of glass. This design characteristic is appealing visually, as well as allowing a lot of natural light into the office space; however, the energy consumption of the building could increase as a result [Persson et al., 2006]. Therefore, this may be why the MOGA opts for smaller windows, as daylight simulations inside the building are not used as an objective function.

6.2 Reliability of the Study

Ensuring the reliability of the study is crucial in determining the solution's viability. As the main objective of the thesis is to optimize building layout based on set objective functions, the correctness of the simulations needs to be high. Utilizing the simulation tools incorporated in Honeybee allows for highly precise results. As discussed in Section 2, regarding Honeybee, simulation tests show minimal differentiation between real-world measurements. Honeybee utilizes EnergyPlus and OpenStudio for simulations. These simulation engines have an excellent reputation for being accurate, as discussed in Section 2, enabling the resulting objective functions to be credible. Weather data corresponding to the correct location will also increase the reliability of the results. The weather data is credible because the data is gathered from Fornebu, Oslo, only 7,5 kilometers from our case site. Using a combination of these tools to see how they can be used to generate an optimal building layout with parameterized design is not a well-documented research direction. This thesis shows the different opportunities within this field. The solutions have many improvements but are a step forward for the construction industry, where the focus on environmentally friendly construction has become an important subject. As creating a building layout often is the first step when creating a building, it can also be viewed as one of the most important decisions to be made.

As the solution presented in this thesis is meant for early prototyping, the complexity has been reduced, allowing for *normal* computers to have an acceptable run time. Therefore, certain elements of Honeybee are not used, such as more detailed modeling of the interior, materials, and sun shades. This, in turn, can result in less accurate predictions regarding energy consumption. Although not all design factors are added to the energy consumption calculations, this study does not build a finished design, but a proposed building layout that can be evaluated and redesigned later in the design stages. The most crucial aspect of this PoC is to see how different building layouts affect the overall energy efficiency of the building. Therefore, adding every design factor for correct calculations is unnecessary.

6.3 Limitation

Generating a building from scratch and optimizing every factor that may affect a building's overall energy consumption seems like a far-fetched concept. Therefore this thesis tried to utilize new parametric design concepts that have the opportunity to be optimized. These design concepts are pretty narrow, allowing for a shorter runtime and opting for the most important objectives. As discussed in section 3.3 this study has a lot of opportunities for further extension. New genomes and evaluation functions can be added or removed, allowing for tailored solutions where green buildings may not be the desired objective.

When first creating a design solution, a design path must be made. Therefore, selecting a design approach that suits the construction site's needs is an important decision. However, creating new parameterized design solutions for every new building location is time-consuming and can be seen as inefficient. Therefore a set of multiple parametric design solutions should be made as standards so building industries can use the program if a design solution is already created within the program. In this study, two methods were conducted to parameterize deign approaches. These approaches, floorplan and building shapes can only create a small portion of every single design choice that can be made, but shows the possibility within the field of building optimization.

Generating BIM models and adding information to the model directly affects the values of the chosen BEMP, in our case, EnergyPlus. This study does not include many design factors that will affect the results of EnergyPlus. Most design factors are set before the algorithm is run, such as material type and HVAC. Many active design factors are also not included in the study. To achieve correct results, all these factors should be included, and therefore this study will only create an estimate of the actual energy consumption of the building. For the purpose of this study, an estimation of the energy consumption is adequate as the primary purpose is to find the layout of the building.

To use the program in its current state, knowledge about Rhino/Grasshopper is necessary. As Octopus provides a user interface that can run the program and display results, the user can run the program if the evaluation functions are set. On the other hand, the possibility of changing objectives, genes, and constraints has to be done directly within the script, which can be a tedious task.

6.3.1 Floorplan Design Approach

The floorplan approach utilizes the plug-in MagnetizingFPG, as discussed in Section 3. This enables pre-made components to be used and integrated into the Grasshopper script. The immediate drawback when using plug-ins is the limiting configurability. As MagnetizingFPG uses optimization techniques inside the component, users have no choice but to use the predetermined objective functions available to the plug-in. Therefore, the main optimization process using the environmental objectives in Octopus cannot alter how the floor plan is generated, resulting in several issues. As Octopus tries to optimize the genes given to it, like rotation and position, it will always have a different layout, meaning the objective functions could be affected by the random floorplan. This results in the MOGA's inability to see the correlation between good values for the genes and a good score for the objective functions, thereby promoting sub-optimal individuals from each generation.

Using the Octopus interface, the building shape design approach allows users to reinstate solutions they find attractive. Reinstating solutions like this results in a building being generated from scratch using the same genes. For example, generating buildings with the same genes for the floor plan approach can result in different floor plans. Therefore, the objective functions found in Octopus will not necessarily represent the building reinstated from Octopus. To combat this issue, a completely new solution for retrieving the optimal solution had to be made and will be discussed closer in Section 6.4.

6.3.2 Building Shape Design Approach

The way building shapes were created for this PoC does vary. The courtyard/C-shape design approach always creates the desired shape, while the H-shape design approach can create multiple shapes. These two methods of building shapes can be used to create interesting designs to be optimized for a given building field. Now only one shape is created for each approach, not testing if any other shape is more optimized. Therefore the building shape design approach is not yet finished, as the purpose was to test what building shape performs the best from a set of evaluations. Adding new shapes to the two design approaches would add this integration where the optimization model can find the best shape for a given location. To extend the solution, the program user should also be able to select what type of building shape will be tested out.

6.4 Challenges

Several challenges were encountered during this thesis. Possible solutions will be discussed in Section 6.5 below.

One of the greatest challenges met during the development relates to the floor plan design approach and the issues presented by MagnetizingFPG, mentioned earlier in this section.

Seeing that the solutions produced by Octopus can not be reinstated using the Octopus interface, a different solution needed to be made, described in Section 3. When using Octopus' interface to reinstate solutions, the genomes are saved in the script as non-volatile data. However, the data collectors used in the floorplan solution only save data if the script is not closed. Therefore, the data would need to be saved by internalizing the data collectors after the simulation is finished, which removes all connections to other components, or saving the objectives to a non-volatile file.

The optimization algorithm used in this case study is the hypE algorithm. For all genetic algorithms, time complexity can be a major issue [Oliveto and Witt, 2014]. In addition, a local optimum can be found, meaning the most optimal solution is not selected. Due to the long calculation time needed for Honeybee and EnergyPlus, the number of generations used in this case study is insufficient for finding a convergence around the solution, as the termination condition was set to 24 hours. One of the key reasons genetic algorithms have long run times is the selection phase within the algorithms [Deb, 2011b]. Implementing neural networks within the GA, as shown in section 2 is a possibility that potentially could decrease the overall run time. It is important to notice that this implementation of the GA could still provide valuable data even if the optimal Pareto front is not found. The Pareto front created will still give information that can be used to select a possible design solution and information about different design variations.

One of the key reasons genetic algorithms were used in this study has to do with the minimal data needed for the algorithm to run efficiently. However, since evolutionary algorithms are computationally heavy, the implementation of reinforcement learning was considered during this study. This approach has been implemented in earlier building optimization tasks [Mocanu et al., 2019] [Hao et al., 2020]. Implementing reinforcement learning could significantly reduce computational time, resulting in better solutions. The drawback would be that the reinforcement model would need large amounts of data to help reward the algorithm when finding good solutions. The main reason reinforcement learning was not implemented in this study has to do with how the solutions are presented. Only one solution will be returned, giving the user fewer choices when retrieving a solution. Since this is a study focusing on the early design stages, it is important that the user is presented with multiple solutions that can be further developed.

6.5 Further Works

Many improvements can be made with the solution created for the PoC in this thesis. First, implementing a more user-friendly interface instead of the script that is now created will allow users with little knowledge about Grasshopper to efficiently use the program. The optimization model can also be more optimized for the problem at hand, minimizing the time complexity. Another key approach in this thesis that has a lot of improvements is the floor plan design approach, where the optimization model should be better developed for the design approach creating more optimal solutions. Lastly, a more complex integration of BIM elements should be implemented for better energy simulations.

6.5.1 User-Friendly Interface

For now, the solution is only created directly inside Rhino/Grasshopper showcasing the potential of such a program in the building industry. There are two possible directions for reducing the program's complexity and creating a user-friendly solution.

The first approach could include a web server allowing users to access the program via a website. The program could be launched on a server in a virtual machine hosted by one of the numerous cloud providers. Autodesk Forge might be used to show the model and essential building information for users through the Rhino.inside.Revit plug-in [rhi, 2020], allowing complicated BIM-models to be viewed and manipulated directly in the browser [Yan, 2017]. The key advantage of such a solution is its flexibility, as it allows the software to be used on various devices. The 3D-modelling tool could also be created from the ground up, moving away from CAD programs such as Rhino, allowing for more tailor-made solutions. Both Ladybug tools and EnergyPlus are open source applications that can be added as APIs in a web server, allowing the usage of already used tools presented in this thesis directly inside a server solution [Hon, 2018][ene, 2020].

The other possibility is to incorporate the solution into Revit. The plug-in could be used by all Revit/Rhino users and assist in the early stages of development. The advantage of such a connection is the ease with which the model may be used in subsequent projects because all the data is already inside the Revit/Rhino ecosystem.

The user interface should also have the possibility to easily change constraints and parameters in the application, so there is no need to go inside the Grasshopper script to make changes. Widely used evaluation functions in building design should also be available to select without altering the script.

6.5.2 Optimisation Model

In order to create a useful program, the run time needs to be reduced to reach convergence. Costumers of the program may not want to wait multiple days for optimal results to be presented. To speed up the algorithm, some new methods can be implemented. Depending on the design problem, the implementation of reinforcement learning can be added instead of using genetic algorithms. This integration only presents one solution. However, this could be enough for the desired design problem in some situations. Moffaert et al. [2014] shows how to implement a multi-objective reinforcement learning algorithm (MORL) such that more than one solution can be retrieved. Implementing faster selection methods within the MOGA could be a step towards a faster optimization algorithm. As discussed, using neural networks has been tested and shown to affect the algorithm's time complexity. Another approach could be to implement knowledge-based engineering (KBE). KBE is the system where an engineer can re-use information that has been stored to generate new solutions [Zhang and Lobov, 2021]. Storing optimal solutions for later works can help decrease time complexity, especially since KBE can help optimize the creation of new genomes when initiating the MOGA. When the desired Pareto front is created, the genomes can be stored in a database. The information provided with the Pareto front can then later be used for initiating the MOGA instead of random genomes. As the PoC is heavily dependant on accurate weather data, and the building location can vary, using genomes from the same locations could increase the credibility and accuracy of the solution.

6.5.3 Optimising the Floor Plan Design Approach

To take full advantage of a floor plan geometry, a tailored solution should be put in place. To achieve this, the floor plan algorithm would have to be able to reproduce solutions from genes, not allowing any form of randomness in the process. This needs to be in place for the optimization algorithm to see correlations between the objective functions and floor layout. In addition, this would allow the Octopus UI to reinstate solutions, making the program easier to use.

Another approach would be to create MagnetizingFPG from scratch, taking inspiration from Egor et al. [2020] paper. As the plug-in uses its own objective functions, namely, max adjacency and boundary, being able to alter the source code could allow the implementation of several new fitness functions to the quasi-evolutionary process used. Inputs could be taken from the simulations used in this study to force the floor plan layout to take into consideration the environmental evaluation functions as well. This means that the use of Octopus would become depreciated, as all optimization happens in the quasi-evolutionary process, producing the final result.

The floor plan design approach has the possibility to implement different building programs from Honeybee, where the different rooms can be given specific properties to increase the correctness of the EnergyPlus simulations. An example would be to assign the bathroom from the floorplan design approach with the bathroom program when initializing the Honeybee rooms. This would allow the model to have accurate water usage. In addition, facade properties such as windows could be removed from rooms not needing them, such as bathrooms and elevators.

6.5.4 Optimising the Design Approaches for more Precise Energy Simulations

More simulation elements should be included to further increase the credibility and complexity of the solution. To improve the correctness of EnergyPlus, a more detailed layout, including lighting, water usage, appliances, and a more detailed and parameterized HVAC system should be included. Some of these elements can be implemented using the room programs. As described in Section 3,the lighting and electrical equipment is only an estimate because these elements have not been included in the model. Implementing them will increase the number of elements the MOGA can configure, possibly improving the solution further. Due to the complexity of the model and limited computational power, window shades and neighboring buildings had to be disabled during the optimization process. They, therefore, do not influence the simulations with regard to solar radiation and energy consumption. To run optimization with a large enough amount of individuals and generations, more computing power is needed. Since most normal personal computers will not have the necessary performance, one idea could be to move the solution to a cloud provider, as discussed earlier. By utilizing a virtual machine, and scaleable CPU usage, the optimization period could be reduced significantly.

6.6 Summary

The methods presented in this thesis show a possible direction of how early stages of building design can be automated, opting to decrease energy consumption. The results generated from the study show that the approaches used in this thesis perform better than the proposal by *HAV Eiendom*. However, the study has many limitations, and improvements to the design, optimization model, and simulations should be conducted for a better product.

7 Conclusion

The goal for this thesis was to implement an application that generate a building based on environmental simulations and optimization methods, as stated in the Introduction.

The architecture was developed with automation in mind, letting the MOGA control all aspects of the geometry. Three design approaches were selected with their respective strengths and weaknesses. Building parameters were chosen based on passive design choices, focusing on building layout and window properties. The floor plan approach utilized the MAgnetizingFPG plugin, allowing complex floor plans to be generated. However, as the plugin works by using a quasi-optimization process, the MOGA cannot control the layout of the floor plan, creating several issues. The building shape design approaches were created based on widely used building shapes. They do not have an interior and focus on the overall shape and position of the building. All approaches were made into Honeybee models, allowing for adding BIM elements, such as windows. The window properties were parameterized as well and controlled by the MOGA. To simulate the energy consumption, HVAC systems were added, allowing simulation engines to run environmental simulations on the model. Honeybee allows integration with EnergyPlus and OpenStudio, both widely used engines for environmental simulation. The total output, sun exposure, and area were used for objective functions. The total output calculated from the EnergyPlus component was minimized, reducing the energy consumption from heating and cooling. Ladybug's solar radiation component calculated the total sun exposure on the facade, trying to maximize the radiation. The floor area is also maximized, creating antagonistic pairs of objectives. The octopus plugin was used for the optimization process due to its ease of use and helpful user interface.

The case study was conducted on plot A5 in Bjørvika, Oslo. Since the location was undeveloped and another proposal from Hav Eiendom was available, the site was used. HAV Eiendom's proposal was modeled to be compared with the three design approaches created in this thesis. To ensure a fair comparison, all buildings were compared on the same objective functions, using the same BIM elements.

One final solution was picked from each design approach. Out of the three approaches, the floor plan approach used the least energy per area, with a final score of 104.4 kWh/m^2. The C-shape approach was able to maximize the sun exposure to 58.0 kWh/m^2, resulting in the highest exposure of all approaches. The H-shape approach produces acceptable results for both sun exposure and energy consumption. All three approaches developed in this thesis beat HAV Eiendoms proposal on both sun and energy consumption per area.

The results show that all approaches have correlations between the objective functions. The area is the most influential, especially for the floorplan approach, where clusters formed in the Pareto front due to the stories gene. Comparing the placement of the final solution with the A5 field study shows that all approaches were able to position themselves at the southern part of the field, maximizing sun exposure. The window properties ended on similar values, with the three approaches opting for smaller windows, with a WWR of only 0.2.

There has not been conducted sufficient research into optimizing and generating parameterized building models based on interior and exterior factors. This study combines these factors, creating diverse building layouts that are an important aspect for early stage building design. The outcome of this study sets the foundation for complex generative simulation tools that could push the industry towards a more efficient and sustainable future. As the solution is made to be extendable, numerous evaluation functions can be implemented such that the program is not locked to only focus on environmental optimisation.

7.1 Further Work

To better understand the implications of these results, future studies should address the key concerns regarding limitations discussed in Section 6.

A UI needs to be put in place to create a user-friendly application. A similar solution could be implemented as more applications are accessible through a web server. The application is hosted on a server and accessible through a user-friendly website, displaying the model and starting the optimization process. Another approach would be creating a UI inside Rhino, allowing Rhino - and Revit users to run the application efficiently.

Introducing reinforcement learning could speed up the optimization process to reduce runtime complexity. Studies show that other optimization issues have successfully reduced runtime using neural networks and could be a possible addition.

To combat the issue relating to the floorplan approach, two possible approaches could be developed. First, by altering the source code or developing a new solution based on MagnetizingFPG, more objective functions could be considered in the optimization process inside the plugin. A second approach could be to develop a similar algorithm from the ground up, generating identical results from the same genes and being controlled by the MOGA.

A more detailed model will result in more accurate simulations. By implementing additional BIM-elements to the interior, the MOGA could optimize more aspects of the building.

Bibliography

Max Roser Hannah Ritchie and Pablo Rosado. Co and greenhouse gas emissions. *Our World in Data*, 2020. https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions.

Maria Felicita. Timber high-rise buildings: Parametric study of the influence of preliminary design parameters on the wind-induced dynamic response of timber high-rise buildings. 2021.

Will Arnold John Orr, Orlando Gibbons. A brief guide to calculating embodied carbon. 2020.

Spyridon Ampanavos, Mehdi Nourbakhsh, and Chin-Yi Cheng. Structural design recommendations in the early design phase using machine learning. In *Communications in Computer and Information Science*, pages 190–202. Springer Singapore, 2022. doi: 10.1007/978-981-19-1280-1_12.

Ralph Evins. A review of computational optimisation methods applied to sustainable building design. *Renewable and sustainable energy reviews*, 22:230–245, 2013.

Janet L Sawin, Freyr Sverrisson, Kristin Seyboth, Rana Adib, Hannah E Murdock, Christine Lins, Isobel Edwards, Martin Hullin, Linh H Nguyen, Satrio S Prillianto, et al. Renewables 2017 global status report. 2016.

Qing-Jie Wen, Zi-Jian Ren, Hui Lu, and Ji-Feng Wu. The progress and trend of bim research: A bibliometrics-based visualization analysis. *Automation in Construction*, 124:103558, Apr 2021. doi: 10.1016/j.autcon.2021.103558. URL `https://www.sciencedirect.com/science/article/pii/S0926580521000091`.

D. S. Shishigin. On choosing the technology of application software integration with a cad-system. *Tr. SPIIRAN*, 47(11):211–224, 2016. URL `http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=trspy&paperid=899&option_lang=eng`.

Nov 2021. URL `https://dynamobim.org/`.

Cristina Magrì Giuseppe Margani Luigi Marletta Emanuele Naboni Gianpiero Evola, Vincenzo Costanzo. A novel comprehensive workflow for modelling outdoor thermal comfort and energy demand in urban canyons: Results and critical issues. *Energy and Buildings*, 216:109946, 2020. ISSN 0378-7788. doi: https://doi.org/10.1016/j.enbuild.2020.109946. URL `https://www.sciencedirect.com/science/article/pii/S0378778820300773`.

Dec 2020. URL `https://aecmag.com/bim/rhino-7-brings-rhino-and-grasshopper-inside-revit/`.

Deborah M. Santos and José Nuno Beirão. Integration of bim and generative design for earthbag projects. *Lecture Notes in Mechanical Engineering*, page 102–109, Sep

2019. doi: 10.1007/978-3-030-29041-2_13. URL `https://link.springer.com/chapter/10.1007/978-3-030-29041-2_13`.

Ming-Der Yang, Min-Der Lin, Yu-Hao Lin, and Kang-Ting Tsai. Multiobjective optimization design of green building envelope material using a non-dominated sorting genetic algorithm. *Applied Thermal Engineering*, 111:1255–1264, 2017. ISSN 1359-4311. doi: https://doi.org/10.1016/j.applthermaleng.2016.01.015. URL `https://www.sciencedirect.com/science/article/pii/S1359431116000703`.

D. Alkazraji. Design approach. *A Quick Guide to Pipeline Engineering*, page 9–22, 2008. doi: 10.1016/b978-1-84569-490-6.50002-3. URL `https://www.sciencedirect.com/topics/engineering/design-factor`.

Shaohua Jiang, Na Wang, and Jing Wu. Combining bim and ontology to facilitate intelligent green building evaluation. *Journal of Computing in Civil Engineering*, 32, 09 2018. doi: 10.1061/(ASCE)CP.1943-5487.0000786.

Aladdin Alwisy, Samer BuHamdan, and Mustafa Gül. Evidence-based ranking of green building design factors according to leading energy modelling tools. *Sustainable Cities and Society*, 47:101491, May 2019. doi: 10.1016/j.scs.2019.101491. URL `https://www.sciencedirect.com/science/article/pii/S2210670718306000`.

Xiaosen Huo, Ann T.W. Yu, Amos Darko, and Zezhou Wu. Critical factors in site planning and design of green buildings: A case of china. *Journal of Cleaner Production*, 222:685–694, 2019. ISSN 0959-6526. doi: https://doi.org/10.1016/j.jclepro.2019.03.123. URL `https://www.sciencedirect.com/science/article/pii/S0959652619308145`.

Yingxin Zhu and Borong Lin. Sustainable housing and urban construction in china. *Energy and Buildings*, 36(12):1287–1297, 2004.

Xi Chen, Hongxing Yang, and Lin Lu. A comprehensive review on passive design approaches in green building rating tools. *Renewable and Sustainable Energy Reviews*, 50: 1425–1436, 2015.

Xiaoling Zhang, Andrew Platten, and Liyin Shen. Green property development practice in china: Costs and barriers. *Building and Environment*, 46(11):2153–2160, 2011. ISSN 0360-1323. doi: https://doi.org/10.1016/j.buildenv.2011.04.031. URL `https://www.sciencedirect.com/science/article/pii/S0360132311001338`.

Jonathan Klawitter, Felix Klesen, and Alexander Wolff. Algorithms for floor planning with proximity requirements. *CoRR*, abs/2107.05036, 2021. URL `https://arxiv.org/abs/2107.05036`.

Manisha Verma and Manish K Thakur. Architectural space planning using genetic algorithms. In *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, volume 2, pages 268–275, 2010. doi: 10.1109/ICCAE.2010.5451497.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197, 2002. doi: 10.1109/4235.996017.

Jianhua Yang, V. Estivill-Castro, and S.K. Chalup. Support vector clustering through proximity graph modelling. In *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.*, volume 2, pages 898–903 vol.2, 2002. doi: 10.1109/ICONIP.2002.1198191.

Gavrilov Egor, Schneider Sven, Dennemark Martin, and Koenig Reinhard. Computer-aided approach to public buildings floor plan generation. magnetizing floor plan generator. *Procedia Manufacturing*, 44:132–139, 2020. ISSN 2351-9789. doi: https://doi.org/10.1016/j.promfg.2020.02.214. URL `https://www.sciencedirect.com/science/article/pii/S2351978920308003`. The 1st International Conference on Optimization-Driven Architectural Design (OPTARCH 2019).

Weimin Wang, Hugues Rivard, and Radu Zmeureanu. Floor shape optimization for green building design. *Advanced Engineering Informatics*, 20(4):363–378, 2006. ISSN 1474-0346. doi: https://doi.org/10.1016/j.aei.2006.07.001. URL `https://www.sciencedirect.com/science/article/pii/S1474034606000450`.

Matheus Soares Geraldi, Veronica Martins Gnecco, Antonio Barzan Neto, Bárbara Augusta de Mafra Martins, Enedir Ghisi, Michele Fossati, Ana Paula Melo, and Roberto Lamberts. Evaluating the impact of the shape of school reference buildings on bottom-up energy benchmarking. *Journal of Building Engineering*, 43:103142, Nov 2021. doi: 10.1016/j.jobe.2021.103142. URL `https://www.sciencedirect.com/science/article/pii/S2352710221010007`.

Md. Mahmud Sazzad. Effect of building shape on the response to wind and earthquake. *International Journal of Advanced Structures and Geotechnical Engineering*, Nov 2015. URL `https://www.researchgate.net/publication/303945216_Effect_of_building_shape_on_the_response_to_wind_and_earthquake`.

Anxiao Zhang, Regina Bokel, Andy van den Dobbelsteen, Yanchen Sun, Qiong Huang, and Qi Zhang. The effect of geometry parameters on energy and thermal performance of school buildings in cold climates of china. *Sustainability*, 9(10):1708, Sep 2017. doi: 10.3390/su9101708. URL `https://www.mdpi.com/2071-1050/9/10/1708/htm`.

Xing Su and Xu Zhang. Environmental performance optimization of window–wall ratio for different window type in hot summer and cold winter zone in china based on life cycle assessment. *Energy and Buildings*, 42(2):198–202, Feb 2010. doi: 10.1016/j.enbuild.2009.08.015. URL `https://www.sciencedirect.com/science/article/pii/S0378778809002060`.

Vincent J.L. Gan, H.K. Wong, K.T. Tse, Jack C.P. Cheng, Irene M.C. Lo, and C.M. Chan. Simulation-based evolutionary optimization for energy-efficient layout plan design of high-rise residential buildings. *Journal of Cleaner Production*, 231:1375–1388, 2019. ISSN 0959-6526. doi: https://doi.org/10.1016/j.jclepro.2019.05.324. URL `https://www.sciencedirect.com/science/article/pii/S0959652619318608`.

Eugénio Rodrigues, Adélio Rodrigues Gaspar, and Álvaro Gomes. Automated approach for design generation and thermal assessment of alternative floor plans. *Energy and Buildings*, 81:170–181, 2014. ISSN 0378-7788. doi: https://doi.org/10.1016/j.enbuild.2014.06.016. URL `https://www.sciencedirect.com/science/article/pii/S0378778814004988`.

Yun Kyu Yi. Building facade multi-objective optimization for daylight and aesthetical perception. *Building and Environment*, 156:178–190, 2019. ISSN 0360-1323. doi: https://doi.org/10.1016/j.buildenv.2019.04.002. URL `https://www.sciencedirect.com/science/article/pii/S0360132319302343`.

Agoston E Eiben, James E Smith, et al. *Introduction to evolutionary computing*, volume 53. Springer, 2003.

Raziyeh Farmani Jonathan A. Wright, Heather A. Loosemore. Optimization of building thermal design and control by multi-criterion genetic algorithm. *Energy and Buildings*, 34(9):959–972, 2002.

Journal and Computing. Improved nsga-ii based on a novel ranking scheme. 2, 2010. URL `https://arxiv.org/ftp/arxiv/papers/1002/1002.4005.pdf`.

Fariborz Haghighat Laurent Magnier. Multiobjective optimization of building design using trnsys simulations, genetic algorithm, and artificial neural network. *Building and Environment*, 45(3):739–746, 2010.

Mohamed Hamdy, Anh-Tuan Nguyen, and Jan L.M. Hensen. A performance comparison of multi-objective optimization algorithms for solving nearly-zero-energy-building design problems. *Energy and Buildings*, 121:57–71, Jun 2016. doi: 10.1016/j.enbuild.2016.03.035. URL `https://www.sciencedirect.com/science/article/pii/S0378778816301724`.

Seyedali Mirjalili. Genetic algorithm. In *Evolutionary algorithms and neural networks*, pages 43–55. Springer, 2019.

David Rutten. Galapagos: On the logic and limitations of generic solvers. *Architectural Design*, 83(2):132–135, 2013.

Octopus. Octopus, Dec 2012. URL `https://www.food4rhino.com/en/app/octopus`.

2015. URL `https://www.wallacei.com/about`.

Chris Mackey Mostapha Sadeghipour Roudsari. Ladybug tools. `"https://www.ladybug.tools/"`, 2021.

Emanuele Naboni, Marco Meloni, Chris Mackey, and Jerome Kaempf. The simulation of mean radiant temperature in outdoor conditions: A review of software tools capabilities. In *IInternational BPSA conference*, 2019.

Mina Michel Samaan, Osama Farag, and Magdi Khalil. Using simulation tools for optimizing cooling loads and daylighting levels in egyptian campus buildings. *HBRC journal*, 14(1):79–92, 2018.

Drury B Crawley, Linda K Lawrie, Curtis O Pedersen, and Frederick C Winkelmann. Energy plus: energy simulation program. *ASHRAE journal*, 42(4):49–56, 2000.

R Guglielmetti, D Macumber, and N Long. Openstudio: An open source integrated analysis platform; preprint. 12 2011. URL `https://www.osti.gov/biblio/1032670`.

Gabriela Bastos Porsani, Kattalin Del Valle de Lersundi, Ana Sánchez-Ostiz Gutiérrez, and Carlos Fernández Bandera. Interoperability between building information modelling (bim) and building energy model (bem). *Applied Sciences*, 11(5):2167, 2021.

Gözde Gali and AZ Yilmaz. Problems for energy certification of complex buildings through simplified methods. In *Proceedings of the first building simulation and optimization conference, Loughborough, UK*, pages 10–11, 2012.

Ankush Rastogi, Jun-Ki Choi, Taehoon Hong, and Minhyun Lee. Impact of different leed versions for green building certification and energy efficiency rating system: A multifamily midrise case study. *Applied energy*, 205:732–740, 2017.

Eleftheria Touloupaki and Theodoros Theodosiou. Energy performance optimization as a generative design tool for nearly zero energy buildings. *Procedia Engineering*, 180:1178–1185, 2017. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2017.04.278. URL `https://www.sciencedirect.com/science/article/pii/S187770581731785X`. International High-Performance Built Environment Conference – A Sustainable Built Environment Conference 2016 Series (SBE16), iHBE 2016.

Johannes Bader and Eckart Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, Mar 2011. doi: 10.1162/evco_a_00009. URL `https://direct.mit.edu/evco/article-abstract/19/1/45/1363/HypE-An-Algorithm-for-Fast-Hypervolume-Based-Many`.

Katja Henttonen, Mari Matinlassi, Eila Niemelä, and Teemu Kanstrén. Integrability and extensibility evaluation from software architectural models–a case study. *The Open Software Engineering Journal*, 1(1), 2007.

Apr 2022. URL `https://haveiendom.no/fra-motorvei-til-allmenning-ett-steg-naermere-park-og-publikumstilbud-i-bjorvika-vest/`.

Ådne Homleid. Hav eiendom har sendt inn forslag for bjørvika-tomt, Apr 2022. URL `https://www.bygg.no/hav-eiendom-har-sendt-inn-forslag-for-bjorvika-tomt/1495167!/`.

2020. URL https://climateknowledgeportal.worldbank.org/country/norway/climate-data-historical.

Kalyanmoy Deb. Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing*, pages 3–34. Springer, 2011a.

Zhongqi Yu, Zhonghua Gou, Feng Qian, Jiayan Fu, and Yiqi Tao. Towards an optimized zero energy solar house: A critical analysis of passive and active design strategies used in solar decathlon europe in madrid. *Journal of Cleaner Production*, 236:117646, 2019.

Mari-Louise Persson, Arne Roos, and Maria Wall. Influence of window size on the energy balance of low energy houses. *Energy and buildings*, 38(3):181–188, 2006.

Pietro S. Oliveto and Carsten Witt. On the runtime analysis of the simple genetic algorithm. *Theoretical Computer Science*, 545:2–19, Aug 2014. doi: 10.1016/j.tcs.2013.06.015. URL https://www.sciencedirect.com/science/article/pii/S0304397513004817.

Kalyanmoy Deb. *Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction*, pages 3–34. Springer London, London, 2011b. ISBN 978-0-85729-652-8. doi: 10.1007/978-0-85729-652-8_1. URL https://doi.org/10.1007/978-0-85729-652-8_1.

Elena Mocanu, Decebal Constantin Mocanu, Phuong H. Nguyen, Antonio Liotta, Michael E. Webber, Madeleine Gibescu, and J. G. Slootweg. On-line building energy optimization using deep reinforcement learning. *IEEE Transactions on Smart Grid*, 10(4):3698–3708, Jul 2019. doi: 10.1109/tsg.2018.2834219. URL https://ieeexplore.ieee.org/document/8356086.

Jun Hao, David Wenzhong Gao, and Jun Jason Zhang. Reinforcement learning for building energy optimization through controlling of central hvac system. *IEEE Open Access Journal of Power and Energy*, 7:320–328, 2020. doi: 10.1109/oajpe.2020.3023916. URL https://ieeexplore.ieee.org/abstract/document/9195858.

Wei Yan. Cumincad : Cumincad papers : Paper ecaade2017$_0$33 : $wp - bim : .2017.URL.$

2018. URL http://www.ladybug.tools/apidoc/honeybee/index.html.

2020. URL https://nrel.github.io/EnergyPlus/api/python/.

Kristof Moffaert, Ann Nowé, Peter Auer, Marcus Hutter, and Laurent Orseau. Multi-objective reinforcement learning using sets of pareto dominating policies. *Journal of Machine Learning Research*, 15:3663–3692, 2014. URL https://jmlr.org/papers/volume15/vanmoffaert14a/vanmoffaert14a.pdf.

Liang Zhang and Andrei Lobov. Extending design automation by integrating external services for product design. In *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, pages 1–6, 2021. 10.1109/INDIN45523.2021.9557486.

Appendix

A Code-Base

The scripts used in this thesis can be found on:

`https://github.com/anderf2706/Generative-Simulation-based-Tool-for-Sustainable-buildings`

B Floorplan Design Approach

The Grasshopper script generating the floorplan using MagnetizingFPG. Figure B.1 shows the boundary curve and centered point being calculated.

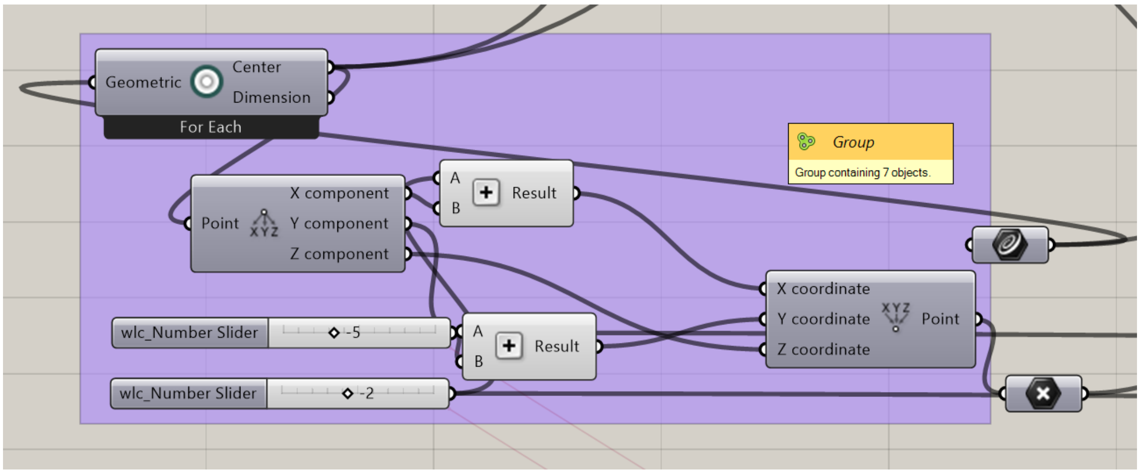
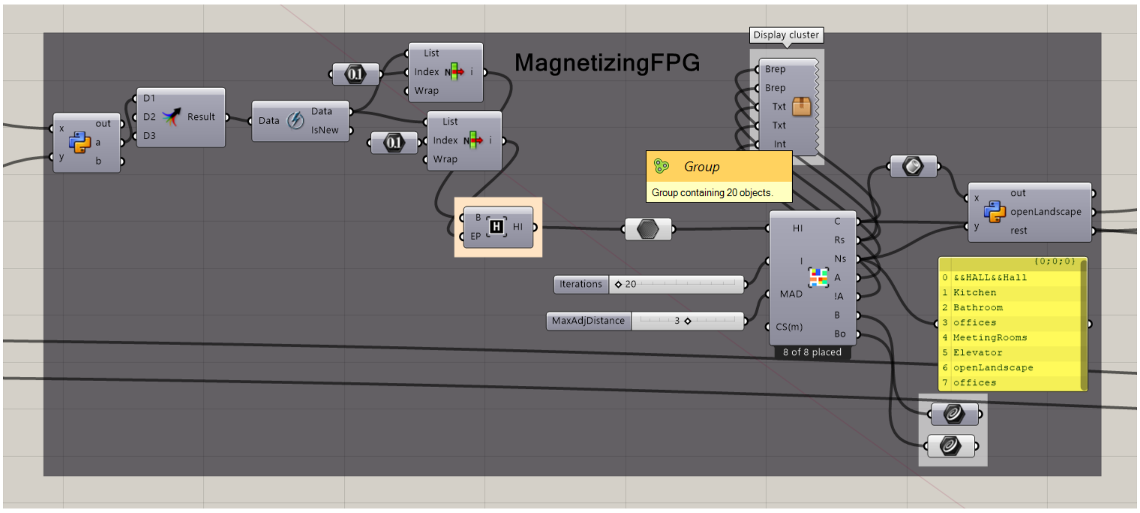


Figure B.1: Boundary curve and centered point being calculated

Figure B.2 shows the MagnetizingFPG component producing the final floor plan.



Group containing 20 objects.

Iterations 20

MaxAdjDistance 3

8 of 8 placed

```
{0;0;0}
0 &&HALL&&Hall
1 Kitchen
2 Bathroom
3 offices
4 MeetingRooms
5 Elevator
6 openLandscape
7 offices
```

Figure B.2: MagnetizingFPG component producing the final floor plan

C H-shape Design Approach

The script generating the H-shape design approach can be seen in figure C.1.

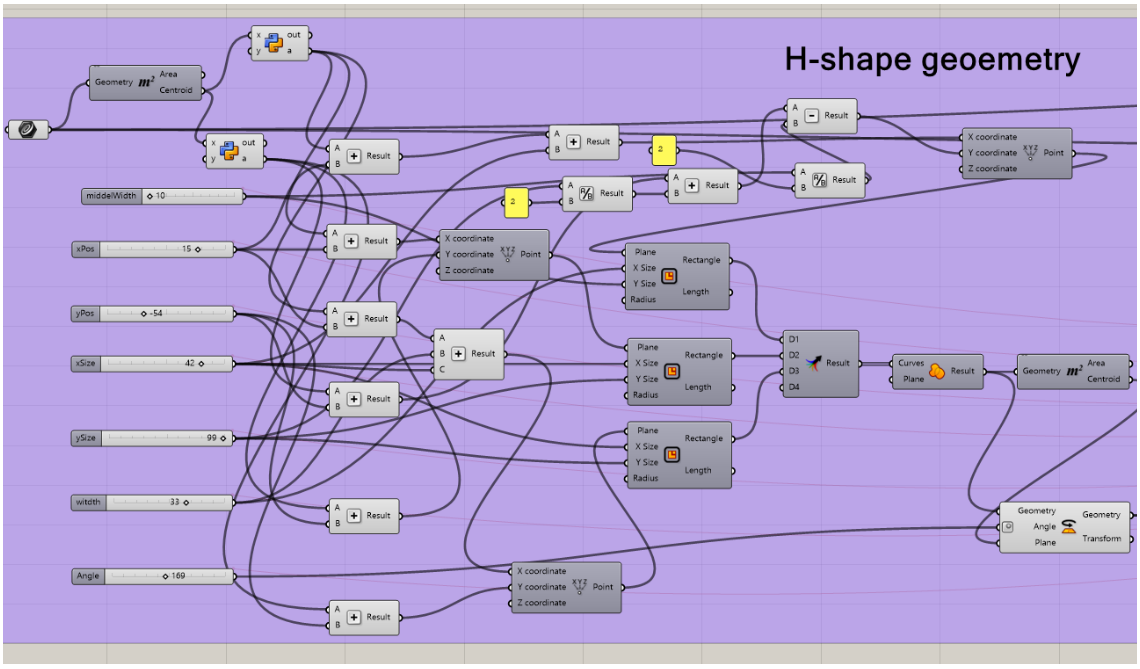


Figure C.1: generating the H-shape design approach

D Courtyard/C-shape Design Approach

Figure D.1 displays the setup for creating and removing the base curves for the courtyard/C-shape.

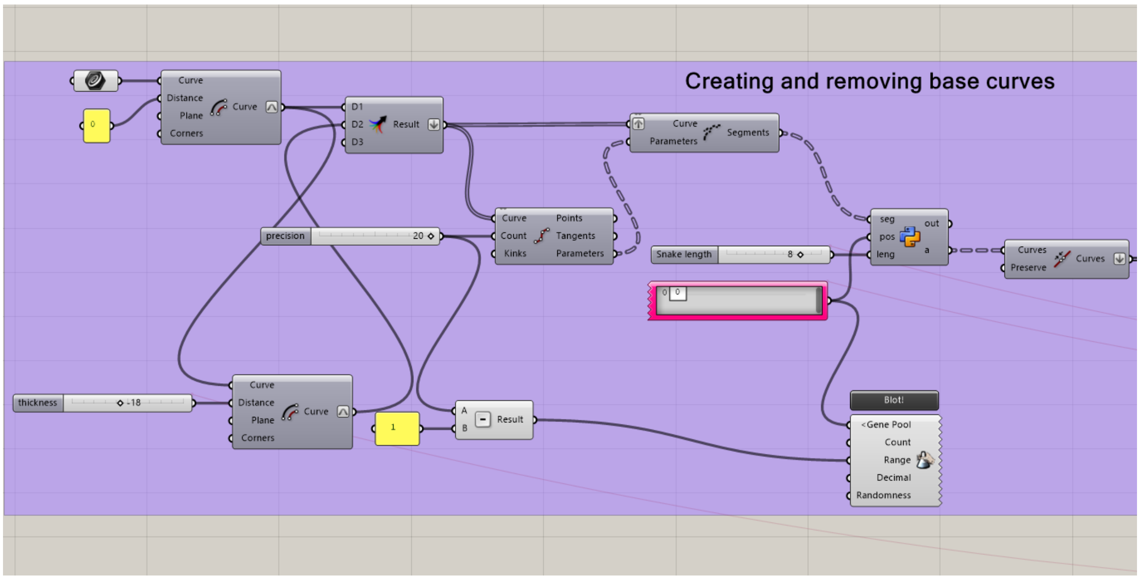


Figure D.1: Creating and removing the base curves for the courtyard/C-shape

To create a C-shape, parts of the courtyard shape is removed, this is done inside the python component seen in the image above. The code for removing parts of the courtyard curve can be seen below.

```
#Courtyard is made if leng = 0
if(leng == 0):
    a = seg
#Remove snake length from the given position
else:
    for i in range(leng):
        if(pos >= len(seg)):
            seg.pop(0)
        else:
            seg.pop(pos)
#Return curves
a = seg
```

After removing parts of the courtyard, the curves are connected to then be extruded. The implementation of this step can be seen in figure D.2.

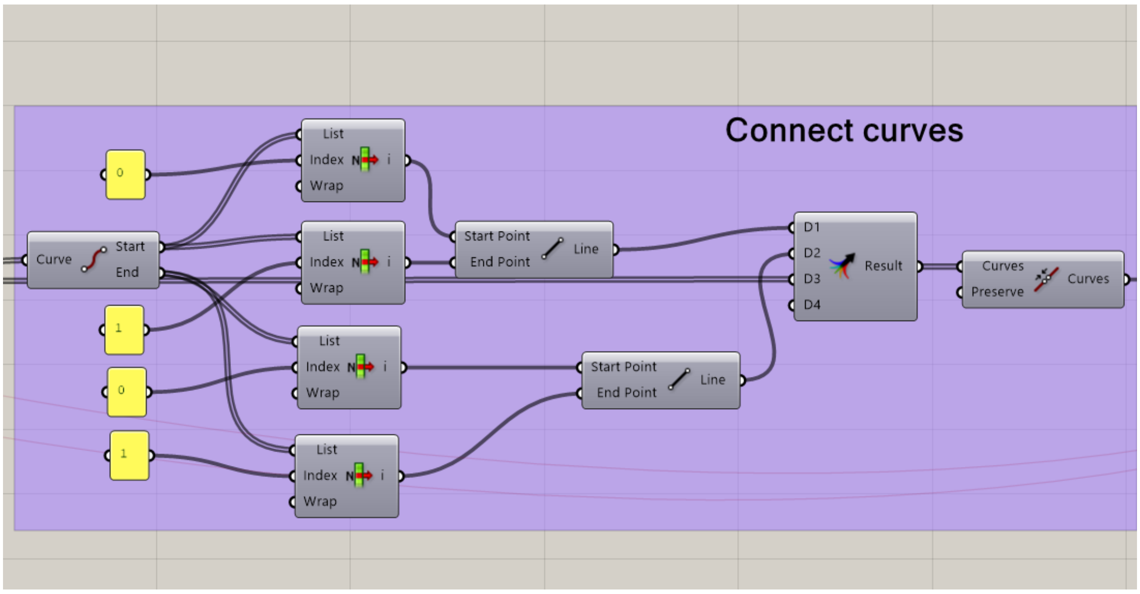


Figure D.2: Connecting curves

E Extrusion

Figure E.1 shows the extrusion of the floorplan design approach, where stories are created, and each room type has the possibility of being extruded in a different height.

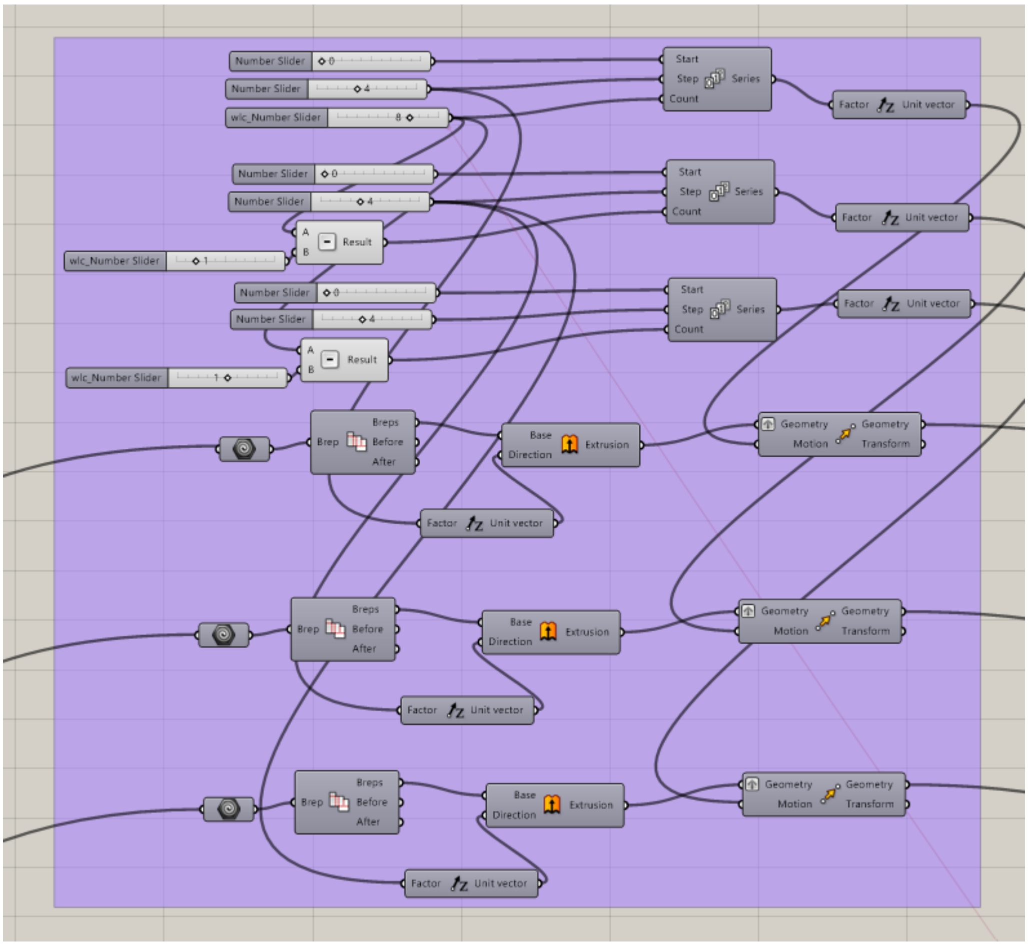


Figure E.1: Extrusion of the floorplan design approach

As the building shape approaches do not implement rooms when the geometry is created, the finalized geometry will be extruded to a set height. How this is implemented can be seen in figure E.2.

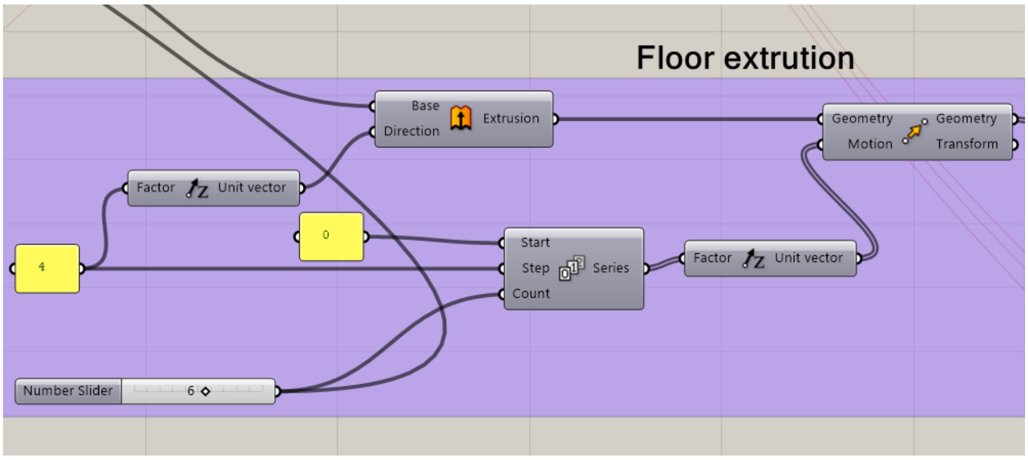


Figure E.2: Extrusion of the building shape design approaches

F Honeybee

Figure F.1 and F.2 shows the process from taking the geometry created using the floorplan - or building shape design approaches. The final product is a functioning Honeybee model.

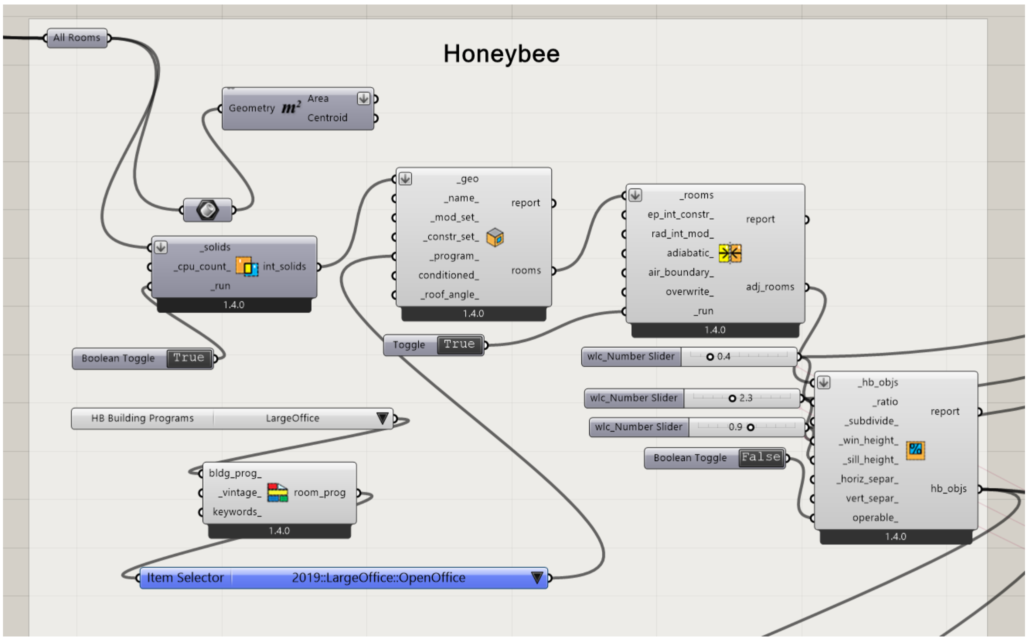


Figure F.1: Implementing BIM elements

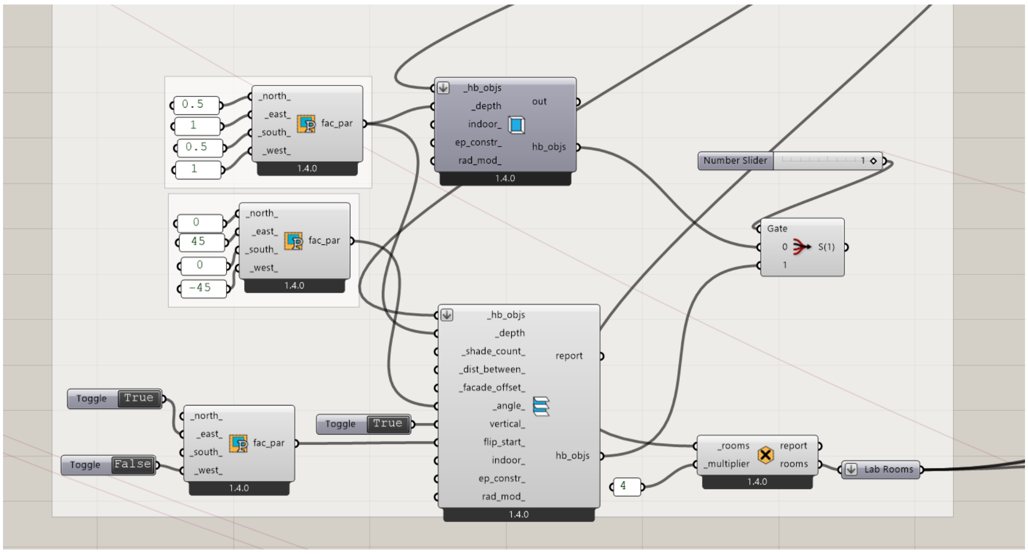


Figure F.2: Adding shades to the design

G Simulation

Figure G.1 shows the EnergyPlus component taking in the finished honeybee model to be analysed. The output is the total energy consumption which is used as an objective function.

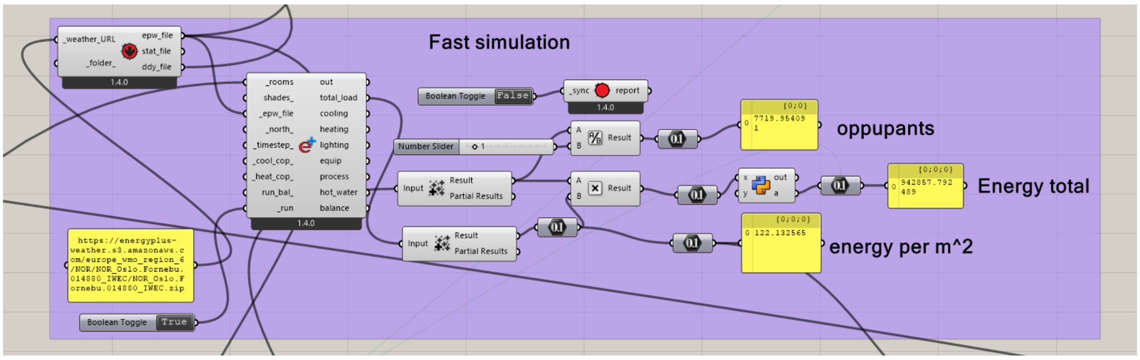
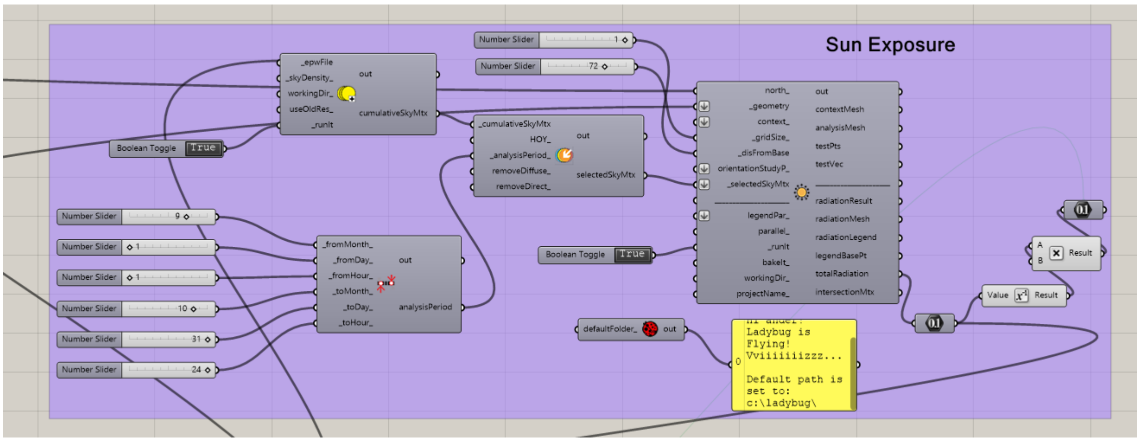


Figure G.1: Energy simulation setup

Figure G.2 shows the Ladybug solar radiation component taking in the geometry and returning the total solar radiation which is used as an objective function.



```
Ladybug is
Flying!
Vviiiiiiizzz...

Default path is
set to:
c:\ladybug\
```

Figure G.2: Calculation of sun exposure