Building performance optimisation tools for the decarbonisation of Swedish buildings

Nadja Baden and Seyedeh Marziyeh Taghizadeh

Master thesis in Energy-efficient and Environmental Buildings
Faculty of Engineering | Lund University

Lund University

Lund University, with eight faculties and a number of research centres and specialized institutes, is the largest establishment for research and higher education in Scandinavia. The main part of the University is situated in the small city of Lund which has about 112 000 inhabitants. A number of departments for research and education are, however, located in Malmö and Helsingborg. Lund University was founded in 1666 and has today a total staff of 6 000 employees and 47 000 students attending 280 degree programmes and 2 300 subject courses offered by 63 departments.

Master Programme in Energy-efficient and Environmental Building Design

This international programme provides knowledge, skills and competencies within the area of energy-efficient and environmental building design in cold climates. The goal is to train highly skilled professionals, who will significantly contribute to and influence the design, building or renovation of energy-efficient buildings, taking into consideration the architecture and environment, the inhabitants' behaviour and needs, their health and comfort as well as the overall economy.

The degree project is the final part of the master programme leading to a Master of Science (120 credits) in Energy-efficient and Environmental Buildings.

Examiner: Jouri Kanters (Division of Energy and Building Design)

Supervisor: Ricardo Bernardo (Division of Energy and Building Design)

Co-supervisor: Rafael Campamà Pizarro (Division of Energy and Building Design)

Keywords: Optimisation, Parametric design, Decarbonisation, Renovation

Publication year: 2023

Abstract

To mitigate climate change and reach Sweden's goal of becoming carbon neutral by 2045, or at an earlier stage of 2030 for 23 Swedish cities, urgent action is required to reduce greenhouse gas emissions. Due to the building sector's significant contribution to carbon emissions, a crucial aspect in achieving these goals is improving the energy efficiency of existing buildings, as most of these will still be in use by this time. To improve the energy performance of these buildings, various renovation measures can be applied, where the choice of the best strategies involves being confronted with multiple conflicting objectives such as reducing energy demand, minimising environmental impact, and managing costs, which can be assessed using advanced building performance simulation tools. Finding the most appropriate renovation solutions involves testing a significant number of different combinations, which can become a very time-consuming process; therefore, optimising the simulation process is essential, especially in large-scale renovations.

This study investigates the effect of using different existing building performance optimisation tools to accelerate the process of finding the most optimal renovation packages for Swedish buildings, with a focus on achieving decarbonisation in a cost-effective manner. A parametric model was created in Grasshopper to analyse three different optimisation tools, including Octopus, Wallacei and Opossum. Their performance was compared to Colibri components, which were used to carry out a 'brute force' to obtain results for energy use intensity, global warming potential and cost, for all possible renovation packages. The evaluation process was conducted in three main parts. The first involved testing 1260 combinations of different passive renovation measures on a simple shoebox model; the second considered both the same model and passive renovation measures, in addition to a number of active renovation measures, resulting in a total of 10 080 combinations to assess the optimisation tools' performance when considering a larger number of iterations; and the third consisted of testing only the passive measures on a more complex geometry, modelled after a real building.

The utilisation of optimisation tools proved to be very effective in accelerating the simulation and assessment process, while maintaining satisfactory precision in achieving optimal results, enhancing the applicability of parametric design as well as its practicality. Opossum was found to be the most efficient tool and reduced the total simulation time by 90 %, while upholding an acceptable level of accuracy in achieving optimal solutions. Additionally, Wallacei proved to be a feasible choice, as it provides the user with a number of useful post-processing features. For the real building model, the most optimal packages generally consisted of glass wool or cellulose fibre insulation at varying thicknesses, for both the walls and roof as well as the installation of a storm window. Although active measures were not applied to the real building model, the installation of a PV system was required for reaching carbon neutrality, as this was the only climate compensation considered in this study.

Acknowledgements

Abbreviations

A_{temp}	Heated floor area
AHU	Air handling unit
BBR	Boverket's Building Regulations
BOS	Balance of System
BPO	Building Performance Optimisation
CF	Cellulose fibre
CO_2	Carbon dioxide
DH	District heating
DHW	Domestic hot water
EA	Evolutionary Algorithm
EAHP	Exhaust air heat pump
EU	European Union
EUI	Energy Use Intensity
GA	Genetic Algorithm
GH	Grasshopper
GHG	Greenhouse gas
GUI	Graphical User Interface
GW	Glass wool
GWP	Global Warming Potential
HB	Honeybee
HF	Hemp fibre
HVAC	Heating, Ventilation and Air Conditioning
HypE	Hypervolume estimation algorithm
$kgCO_2eq.$	Kilograms of carbon dioxide equivalent
KSEK	Thousand Swedish krona
kWh	Kilowatt hour
kW_p	Kilowatt peak
LB	Ladybug
LCA	Life cycle assessment
LCC	Life cycle costing
MOO	Multi-Objective Optimisation
MSEK	Million Swedish krona
NPV	Net Present Value
NSGA	Non-dominated Sorting Genetic Algorithm
RBFOpt	Radial Basis Function Optimisation
SCOP	Seasonal Coefficient of Performance
SGBC	Swedish Green Building Council
SOO	Single-Objective Optimisation
SPEA	Strength Pareto Evolutionary Algorithm
UNFCCC	United Nations Framework Convention on Climate
W_p	Watt peak
WF	Wood fibre
WWR	Window-to-Wall Ratio
ZEB	Zero Emission Buildings

Terminology

Term	Definition
Brute force	An exhaustive search where all possible solutions to an optimisation problem are consecutively generated and evaluated.
Carbon neutrality	A state of having net-zero carbon emissions.
Convergence	An optimisation algorithm will converge when the evaluated solutions remain relatively unchanged, meaning that a satisfactory solution has been identified, or when another stopping criterion is met.
Heuristics	Intuitive problem-solving approaches, where prior knowledge or experience is used to make decisions quickly, without guaranteeing an optimal solution.
Parametric modelling	A computer-aided design (CAD) modelling technique that utilises adjustable parameter inputs to easily modify the design.
Search space	The array of all possible solutions for a particular optimisation problem.
Stochastic	A stochastic process involves probabilistic elements, thus producing unpredictable outcomes.

Contents

1 Introduction

Despite a progression towards the use of renewable energy sources, fossil fuels continue to be the main source of providing energy needs, resulting in large concentrations of greenhouse gases (GHG) in the atmosphere (United Nations Environment Programme, 2020). According to the Intergovernmental Panel on Climate Change (IPCC), the last decade has had the greatest amounts of GHG emissions to date, meaning that urgent action is required to reduce emissions and therefore keep global warming below the target of 1.5 °C (IPCC, 2022). One of the leading emitters is the building sector; in 2020, buildings accounted for 36 % of global energy demand and 37 % of energy related CO_2 emissions (United Nations Environment Programme, 2021). Within Sweden, the construction and real estate sector is responsible for one fifth of GHG emissions, while its importation of construction products also affects emissions in other countries (Boverket, 2020b).

According to a study, the building sector was also estimated to hold the greatest economic mitigation potential in 2030, giving it an important role in climate change mitigation (IPCC, 2007). The majority of existing buildings in the EU were constructed over 20 years ago, while around 90 % of these will still be in operation in 2050; this means large scale renovation is needed in order to decarbonise the building stock and meet European climate neutrality goals (European Commission, 2020). The improvement in energy performance due to renovation, results in a reduction in CO_2 emissions (Ramírez-Villegas et al., 2016); however, renovation processes also require a significant amount of materials and energy to execute (European Commission, 2019).

Computational simulation enables quantifying the building performance for proposed renovation measures, which can be used to achieve a specific target, i.e., balancing energy consumption, environmental impact, and costs (Clarke, 2001). A parametric modelling approach allows for the automated testing of a large number of potential renovation strategies; however, in large scale renovation this can be very time intensive. The time required to find the most feasible solutions can be substantially reduced by incorporating various optimisation methods.

1.1 Background

In order to mitigate climate change, a number of agreements have been concluded worldwide, one being the Paris Agreement under the 1992 United Nations Framework Convention on Climate Change (UNFCCC): this aims to "hold the increase in the global average temperature to well below 2°C above pre-industrial levels and pursuing efforts to limit the temperature increase to 1.5°C above pre-industrial levels" (UNFCCC, 2015). Europe also strives to be the first climate neutral continent with the formation of the European Green Deal, which addresses environmental problems by targeting net-zero GHG emissions by 2050 (European Commission, 2019).

Sweden was one of the first UNFCCC parties to adopt a net-zero GHG emission goal by establishing a climate policy framework in 2017, outlining that the country is to have net-zero emissions of GHG into the atmosphere by 2045 and should achieve negative emissions thereafter (Svensk Riksdag, 2021). To move in this direction, several milestone targets have been set, including a 63 % and 75 % decrease of GHG emissions by 2030 and 2040 respectively, compared to emissions in 1990 (Ministry of the Environment Sweden, 2020). In addition to this, Viable Cities created the Climate Neutral Cities 2030 initiative in 2019, bringing together 23 Swedish cities and six public authorities, with an aim to accelerate the climate transition and already achieve net zero GHG emissions by 2030 (Viable Cities, 2022).

With the building sector being one of the leading GHG emitters, an important aspect of reaching climate goals is to improve energy-efficiency in both existing and newly built buildings. To address this challenge, a renovation wave is needed within Europe; however, only 0.4 % to 1.2 % of buildings currently undergo energy-efficient renovation every year, which should at least double to reach the EU's climate objectives (European Commission, 2019). The post-war building stock within Europe, such as homes constructed between 1965 and 1974 as part of the Swedish Million Programme are already in need of renovation and have great energy-saving

potential, since there were no energy performance requirements for buildings at this time (Ministry of the Environment Sweden, 2020).

Various renovation measures can be applied to existing buildings to improve their energy performance, which involves being confronted with multiple conflicting objectives such as minimising energy demand, environmental impact, and costs. When multiple design variables are being considered simultaneously, Building Performance Optimisation (BPO) is a useful technique for finding the most feasible solutions among all possible cases. This method incorporates optimisation algorithms within building performance simulation to provide a set of optimal solutions (Si et al., 2016).

1.2 Objectives

This thesis will explore various optimisation algorithms used within computational calculations and identify those most relevant and commonly used for building performance analysis. This study then aims to evaluate and compare a selection of available optimisation tools within the program grasshopper (GH), which will aid the user in finding the most favourable solutions that balance the conflicting objectives of energy use, environmental impact, and cost. This will accelerate the calculation process of determining the most viable renovation packages to achieve carbon neutrality.

The study intends to answer the following questions:

- What optimisation algorithms are available for building performance analysis?
- Which objectives should be set for the optimisation of renovations towards carbon neutrality?
- Other than achieving a less time intensive simulation process, what criteria should be used to compare the performance different optimisation tools?
- How does a change in the number of tested iterations or the complexity of the building model geometry affect the performance of optimisation tools?
- Based on the results, which of the investigated optimisation tools is recommended?

1.3 Limitations

This study included several limitations and aspects that were disregarded throughout. When considering the analysis of renovation packages, other important values aside from the examined objectives were overlooked, including thermal comfort, indoor air quality, property value, moisture safety and daylight availability. The outcomes of this study were also heavily influenced by the chosen carbon neutrality definition of NollCO2, as other definitions would have reached different conclusions in terms of the performance of renovation packages. Additionally, the evaluated renovation measures within each package were limited to a chosen range, meaning that there will most likely be other variables which prove to be more optimal than those regarded in this study, i.e., it might be more effective to install a window type that is different from those that were analysed. Moreover, rather than investigating the renovation solutions themselves, this study was more focused on evaluating whether different tools reached similar solutions, while the investigation was limited to only one real building, so more testing on different typologies is required.

2 Literature review

A literature review was conducted to gain a deeper understanding of the topics associated with the aim of this study.

2.1 Carbon neutral definitions

There are a number of carbon neutrality calculation methods available, such as $NollCO_2$ (SGBC, 2022), Zero Emission Buildings (ZEB) (Fufa et al., 2016) and White Arkitekter's approach (White Arkitekter, 2020), with each one differing in their process. A comprehensive analysis of these methods can be found in a study evaluating the impact of different definitions on building design (Razna & Aive, 2022). $NollCO_2$ was implemented within this study, as it was considered to be a good representation of current and future carbon neutral certification within Sweden.

2.1.1 NollCO$_2$

$NollCO_2$ (SGBC, 2022) is a certification system created by the Swedish Green Building Council (SGBC). In this definition, climate neutrality equates to a total climate impact of zero, which is achieved by balancing a building's reduced climate impact, in line with the IPCC's 1.5° development path, with reductions or uptake of GHG emissions outside of the $NollCO_2$ project's system boundary. Table 2.1 lists the climate impacts and offsets included in this net zero balance. The standard SS-EN 15978:2011 is used for calculations of the climate impact during the building's life cycle while several measures are defined as offsets for this. Any installation of renewable electricity is considered as a replacement of fossil-fuelled electricity in the Nord Pool's electricity market, while improving the energy-efficiency of existing buildings and purchasing Continuous Cover Forestry (CCF) certificates or carbon credits from an approved organisation are also accepted as carbon offsetting measures. Since this certification is only intended for newly built constructions, it was adapted to the renovation project in this study.

Table 2.1. NollCO2 net-zero balance considerations

Climate impact	Offsets
Production* (A1-A3)	On/off-site renewable electricity delivered to the
Construction* (A4-A5)	grid
Replacement and refurbishment (B4-B5)	Energy efficient measures in existing buildings
Energy use* (B6)	Swedish CCF certificate and approved climate
Water use (B7)	compensation

*Contains a set limit

2.2 Optimisation

Optimisation ultimately entails finding the optimal or near optimal solution(s) to a particular problem (Floudas & Pardalos, 2009). Instead of using prior knowledge and intuition to formulate an ideal solution based on a certain number of iterations, optimisation provides an efficient and systematic approach to identify a satisfactory solution, without specifically evaluating every possible alternative (Ravindran et al., 2006).

2.2.1 Optimal problem formulation

The formulation of an optimal problem typically consists of identifying the decision variables, objective function, and constraints. This allows for the creation of a mathematical model describing the essence of the problem, which can in turn be solved by an optimisation algorithm (Hillier & Lieberman, 2010). The decision variables consist of the parameters that will be varied during the optimisation process, while others might remain fixed or change in accordance with other values (Deb, 2012). When formulating the objective function, the aim of the problem is expressed in terms of its decision variables and other parameters, and it should either be minimised or maximised (Arora, 2015). The constraints refer to any physical condition or limitations the

resulting design must adhere to, in order to maintain a functional relationship between certain parameters i.e., some parameters must be maintained at, above or below a particular value (Deb, 2012). If no constraints are specified, the problem is referred to as an unconstrained optimisation problem. Overall, the mathematical model should identify the decision variable values that minimise or maximise the objective function, whilst complying with the constraints (Hillier & Lieberman, 2010).

This study focuses on a black-box optimisation problem, where the relationship between decision variables and objectives is not explicitly defined by a mathematical equation, but by using computer simulations to evaluate a parametric model of a building (Wortmann, 2017b). In this type of optimisation, the evaluated design parameters are the decision variables, and computer simulations are used to calculate and optimise certain performance measures (i.e., objective functions), in order to identify the most favourable parameterisations (Costa & Nannicini, 2018). Computer simulations are a form of black box; this describes a process that will return an output when supplied with an input, yet there is no analytical availability of the intrinsic workings of the process (Audet & Hare, 2017). Therefore, black box optimisation methods do not require a mathematical formulation: these methods are discussed further in later sections.

2.2.2 Multi-objective optimisation

An optimisation problem can be referred to as single-objective or multi-objective. Single-objective optimisation involves one objective function, where the optimal solution relates to a single point in the design space, at which the value of the objective function is either minimised or maximised (Arora, 2015). Alternatively, two or more objective functions result in multi-objective optimisation (MOO); this gives rise to a set of points known as Pareto optimal solutions, which can be considered as equally favourable overall (Chankong & Haimes, 1983). Most real-world optimisation problems are of this type, as they involve multiple conflicting objectives, where the improvement in one objective results in degenerating another; these problems have no single best solution and the goal is to find good compromises instead (Chong & Zak, 2013).

All possible solutions can be represented in the decision space (search space) and the objective space, as shown in Figure 2.1. The decision space contains each feasible solution in terms of its decision variables and other parameters. Each point in the decision space corresponds to a point in the objective space, which is used to evaluate the quality of solutions in MOO; in this space, each solution is shown in terms of the values of its objective functions, where the number of dimensions is equal to the number of objectives (Anagnostopoulos et al., 2017; Talbi, 2009).
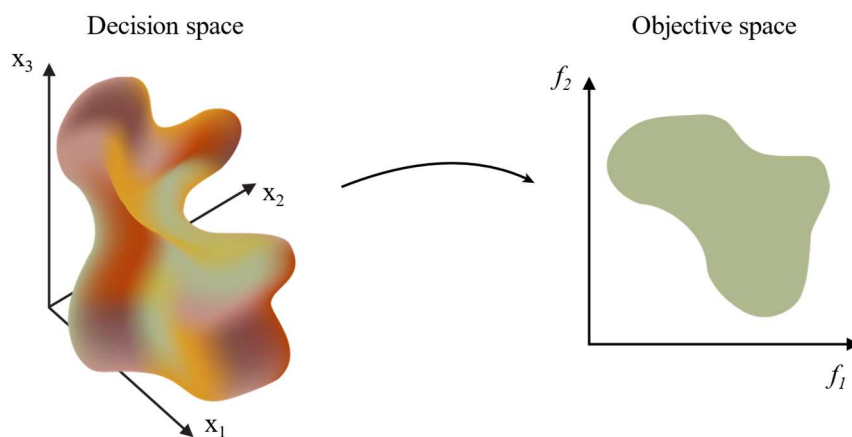

Figure 2.1. The decision space and objective space of an optimisation problem, adapted from (Deb, 2001)

The relative optimality of solutions within a multi-objective problem can be assessed using the principle of domination (Tan et al., 2005). In order to confirm whether one solution dominates another, the following conditions must be satisfied (Arora, 2015; Deb, 2001):

- Solution x_1 is no worse than x_2 in all objectives
- Solution x_1 is better than x_2 in at least one objective

To illustrate this, Figure 2.2 shows the objective space of an optimisation problem with two objective functions, where f_1 should be maximised while f_2 must be minimised. When comparing solution A and solution C, it is clear that solution C is better in both objective functions (it has a greater value in f_1 and a smaller value in f_2), meaning it dominates solution A. However, when looking at solution A and solution B, it cannot be said that either one dominates the other, since solution A is better in f_1 but worse in f_2. Solutions that are not dominated by any other solution are referred to as non-dominated solutions and form the Pareto optimal front, where any solution outside of this set will be dominated by one inside it. Therefore, it can be said that solutions of this set are superior to the rest (Coello Coello et al., 2007; Deb, 2001). When an optimisation problem contains three objective functions, the non-dominated set will form a Pareto surface instead.
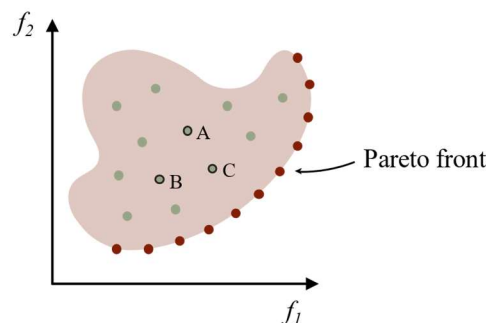
Figure 2.2. The Pareto optimal front

Since these Pareto optimal solutions are all of equal importance, it is desirable to identify as many as possible, while the obtained set should be diverse to ensure a good coverage of the Pareto front (Rostami et al., 2020). It should also be noted that multiple optimal solutions will only exist if the objectives are conflicting; otherwise, the resulting problem involves single-objective optimisation, which can be considered as a degenerate case of multi-objective optimisation (Deb, 2001).

2.2.3 The fitness landscape

The structure of an optimisation problem can be assessed through a fitness landscape; a concept which was first introduced by Sewall Wright within the field of biology, where the term fitness comes from "survival of the fittest" in natural selection and corresponds to an organism's ability to survive, mate, and reproduce (Talbi, 2009; Wright, 1932). The landscape was thought to have several peaks while the population would evolve by climbing uphill, thereby increasing fitness (Coello Coello et al., 2007; T. C. Jones, 1995). Much like the fitness landscape was initially used to examine and comprehend biological systems, it has also been applied to optimisation problems (Talbi, 2009).

Figure 2.3 shows an optimisation problem where points representing all possible solutions are visualised as a fitness landscape. For each solution, the decision variables (genes) dictate their quality (fitness), which is expressed as the height of the landscape. Therefore, the aim is to find the peaks, as these will contain the optimal solutions (Knowles et al., 2008). More specifically, the intention is to identify the highest peak, known as the global optimum; as these solutions equate the absolute optimum value of the objective function, while the other peaks are known as local optima (Deb, 2012). However, it is often impossible to conclude whether the current best solution is located at a local or a global optimum and some optimisation algorithms are more effective in finding global optima and avoiding local ones (Weise, 2009).

Figure 2.3.The fitness landscape

2.3   Classification of optimisation algorithms

Optimisation algorithms can be categorised in a number of ways, and the selection of an optimisation algorithm is dependent on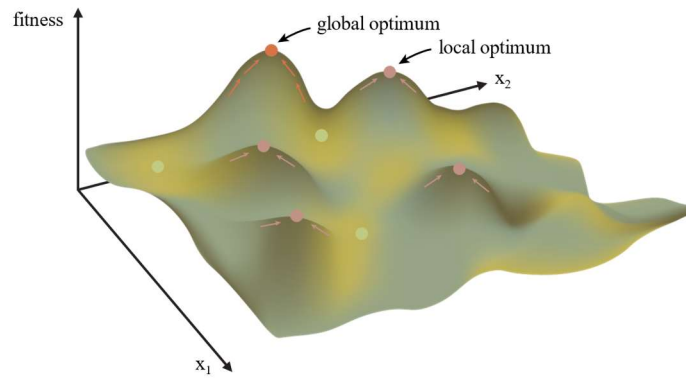 the type of optimisation problem that it should solve. Generally, algorithms have search principles that are either deterministic or stochastic: deterministic algorithms use specific rules when moving from one solution to another, where the same inputs will always yield the same results; alternatively, the transition principles in stochastic algorithms are probabilistic, so different solutions might be produced each time (Deb, 2012; Weise, 2009). Due to their randomness element, stochastic approaches are typically more effective in finding global optimal solutions and handling complex problems such as black box optimisation; however, the accuracy of the solution is often compromised, and more time is required for a higher level of precision (Deb, 2001; Kvasov & Mukhametzhanov, 2018). On the other hand, deterministic approaches can be considered unfeasible for problems with a complicated structure and a large search space, as an exhaustive search (brute force) often becomes the only option, making the process extremely time intensive (Weise, 2009).

This study is focused on black-box optimisation methods, which must obtain a balance between exploring the entire design space and focusing on a promising region to find an optimal solution (Wortmann, 2017b). These methods have been further categorised as follows: direct search, model-based and metaheuristics.

2.3.1   Direct search

Direct search is most commonly known as an unconstrained optimisation method which does not utilise derivatives (Kolda et al., 2003). The term seemingly first appeared in 1961, when Hooke and Jeeves described it as sequential examination of trial solutions, by means of a certain strategy (Hooke & Jeeves, 1961). Unlike gradient-based methods, which utilise the derivative of the objective function, direct search methods do not require any information about the gradient, but directly use objective function values to determine search directions (Deb, 2012; Rios & Sahinidis, 2013). Generally, the algorithm begins with a possible solution in the search space and explores a set of trial points; if one is found to be more optimal, it will replace the original solution and the same process continues until the algorithm converges; otherwise, a new set of trial points are examined (Audet & Hare, 2017). This means, the set of points create a direction of search, while the algorithm simply relies on the relative rankings of objective values to compare solutions (Lewis et al., 2000).

Direct search methods can be categorised into both local and global approaches, depending on their scope within the search space. Two well-known local direct search methods are the Hooke-Jeeves method and the Nelder-Mead simplex method (Hooke & Jeeves, 1961; Nelder & Mead, 1965). The Hooke-Jeeves method employs the coordinate directions of the search space to conduct a series of exploratory moves in order to gain insight into the objective function; successful moves are followed by pattern moves, which accelerate the search by determining an ideal direction based on previously acquired information. The Nelder-Mead simplex method consists of constructing a simplex (i.e., a collection of points in the search space), where the vertices are adjusted at each iteration by using one of five operators: reflection, expansion, outer contraction, inner contraction and shrinkage (Baeyens et al., 2016; Ravindran et al., 2006).

DIRECT is an example of a global direct search algorithm, with its name being an acronym for dividing rectangles (D. R. Jones et al., 1993). Figure 2.4 illustrates three iterations of DIRECT: in this method, the search space is partitioned into hyper-rectangles and the objective function value is assessed at each rectangles centre point; favourable rectangles are then subdivided for further search until the algorithm converges (D. R. Jones & Martins, 2021).
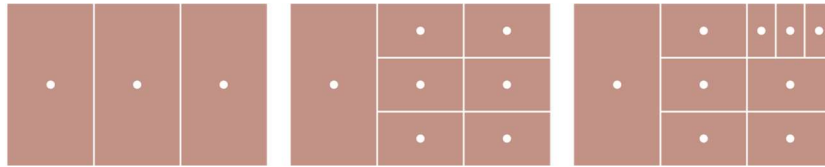

Figure 2.4. Three iterations of the DIRECT algorithm, adapted from (Wortmann, 2017a)

2.3.2 Model-based methods

Model-based methods attempt to accelerate the optimisation process by utilising an approximated model of a black-box problem (Wortmann & Nannicini, 2016). This model is known as a surrogate model, which mimics the behaviour of a high-fidelity simulation model in a simplified manner and makes predictions, by exploring the connection between the input and output variables (D. Yang et al., 2016). It is formed through the interpolation of a mathematical function by relating a set of chosen decision variables and their objective function values, in order to estimate the design space (Wortmann et al., 2015). Global model-based methods construct these models using various statistical (e.g. Polynomial Regression and Kriging) and machine learning-related (e.g. Neural Networks and Support Vector Machines) techniques (Wortmann, 2017b). Figure 2.5 shows three iterations of a model-based algorithm. At each iteration of the optimisation process, a promising solution is obtained and evaluated using a high-fidelity simulation model; the data is then used to update the surrogate model, increasing its complexity (Koziel & Yang, 2011). Thus, this method alternates between improving the accuracy of the surrogate model and using it as a guide to identify further solutions (Wortmann & Nannicini, 2016). As a result, the computational cost of the simulation is reduced, when compared to optimising the high-fidelity simulation model directly, while these methods typically only require a small number of iterations until a termination criterion is met (Koziel et al., 2011).

Figure 2.5. Three iterations of a global, model-based algorithm, adapted from (Wortmann, 2017a)

2.3.3 Metaheuristics

Metaheuristics comprise the primary subfield of stochastic optimisation, which employ elements of randomness to identify optimal solutions: they are generally applied to optimisation problems with a large search space and little available knowledge about the inner workings as well as the potential outcomes of the problem (Luke, 2016). A heuristic is an intuitive decision-making procedure, implemented within an optimisation algorithm: it guides the search process, without the guarantee of discovering a global optimal solution. Heuristics are typically created for a particular type of problem, whilst metaheuristics apply heuristics in an abstract manner to provide a more general strategy, which has the ability to escape local optima and can be applied to a number of different problems (Hillier & Lieberman, 2010; Weise, 2009). This involves an iterative process, combining the exploration of the search space with the exploitation of potentially feasible solutions (Abo-Hamad & Arisha, 2011). These two factors should be relatively balanced, as too much exploitation can cause the algorithm to become stuck in local optima whilst too much exploration can cause difficulty with convergence, thus slowing down the optimisation process. The selection and assessment criteria applied by metaheuristics are often nature inspired, such as ensuring the survival of the fittest solutions, by continually updating the best solution found thus far (X.-S. Yang, 2011). Metaheuristics are favoured for their easy implementation and applicability to a wide range of problems, regardless of complexity or size; however, they also tend to have a high computational

cost and poor convergence properties (Wortmann et al., 2015). These methods are either population-based (e.g., evolutionary algorithms and swarm intelligence) or trajectory-based (e.g., simulated annealing): while population-based algorithms use multiple candidate solutions in their search, trajectory-based algorithms only keep around a single candidate solution throughout the search process (Luke, 2016; X.-S. Yang, 2011). Figure 2.6 shows three iterations of a genetic algorithm: one of the most commonly used metaheuristics.

Figure 2.6. Three iterations of a genetic algorithm, adapted from (Wortmann, 2017a)

Evolutionary Algorithms (EA)

Evolutionary optimisation approaches mimic the behaviour of biological principles, to drive the search for an optimal solution (Pintér, 2002). They are population-based methods and refine a set of solutions through an iterative process, making them especially suitable for multi-objective optimisation, as they possess the ability to discover multiple optimal solutions in one iteration (Weise, 2009). Despite there being a number of different evolutionary algorithms (i.e, genetic algorithms, evolution strategy, evolutionary programming and genetic programming), they are all driven by evolutionary principles and possess operations of selection and search, where favourable solutions are duplicated and new solutions are created through a partial exchange of information (Deb, 2001). Genetic algorithms are the subset of EA's most relevant in this study; they are described further in the next section.

2.4 Genetic Algorithms (GA)

Genetic algorithms are a subcategory of evolutionary algorithms, rooted in the biological concept of natural selection (Goldberg, 1989). John Holland conceptualised GA's during the 1960's, being motivated by selective breeding principles to maintain desirable characteristics within a population (Holland, 1975; Reeves, 2010). The algorithm identifies each point as an individual with a set of genes (decision variables), giving them a certain fitness value, which will guide the evolution of favourable solutions (Rios & Sahinidis, 2013). A set of individuals (solutions) make up a population, and their genetic information is stored in strings known as chromosomes, with certain biological processes making changes to their composition, in turn creating new individuals (Audet & Hare, 2017). The algorithm keeps around a population of candidate solutions of which the population size is generally specified by the user and will affect the performance of the algorithm: an overly large population size could result in an unnecessary addition of computational time, whilst a small population size might cause premature convergence with less accurate results (Reeves, 2010).

2.4.1 The genetic algorithm process

The entire process of a genetic algorithm is illustrated in Figure 2.7. The population in its entirety will undergo a number of transitions, each one being referred to as a generation, until no further significant improvements occur or another stopping criterion is met (Arora, 2015). The initial population is typically created at random, after which the workings of a GA consist of three main operators to complete each subsequent generation: selection, crossover and mutation (Floudas & Pardalos, 2009).
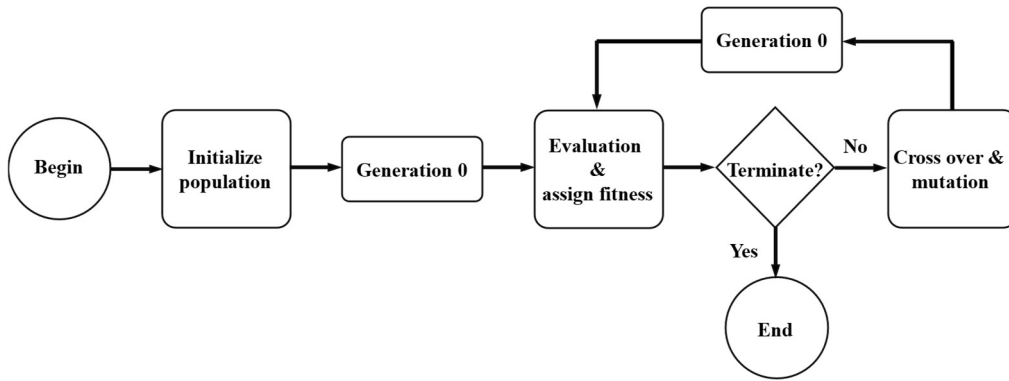
Figure 2.7. The genetic algorithm process

Selection

The selection operator ensures that the fittest members of the population survive and duplicate, while the least fit are eliminated, keeping the population size the same (Deb, 2001). Firstly, each candidate solution (individual) within the population, represented by a string of decision variables (genes), is evaluated to find their fitness, which corresponds to their objective function value (Chong & Zak, 2013). The solutions with higher fitness values are chosen to form a mating pool, for which a number of different selection procedures have been proposed (Reeves, 2010). A common method is tournament selection, where the selection takes place in a randomly created subpopulation that is given the opportunity to interact (Audet & Hare, 2017). As shown in Figure 2.8, solutions are compared in pairs, of which the fitter option will become part of the mating pool: each solution should be compared twice, giving them either one, two or no copies (Deb, 2001).
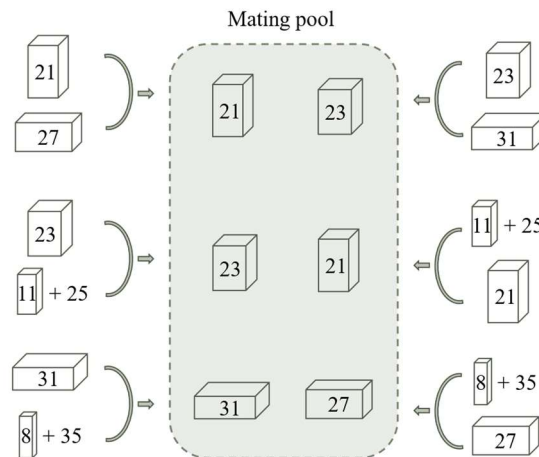
Figure 2.8. Tournament selection, adapted from (Deb, 2001)

Crossover

The crossover operator randomly combines parental solutions from the mating pool to create new offspring solutions, with different and potentially more favourable traits (Reeves, 2010). This involves exchanging substrings of two parent chromosomes, which can be done in a number of different ways. A one-point crossover is performed by assigning a crossing site and interchanging the parts on one side of this site, as shown in Figure 2.9. The crossover operation can also use multiple crossing sites, such as the two-point crossover illustrated in Figure 2.9 (Chong & Zak, 2013).
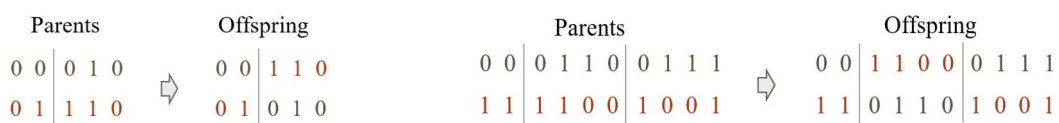
Figure 2.9. One-point crossover (left) and two-point crossover (right)

9

Mutation

The mutation operator randomly modifies an offspring solution, with one or more adjustments being made to the features of their chromosomes (Reeves, 2010). Although most mutations are disadvantageous or have no effect, this is mainly done to maintain diversity amongst the offspring and potentially cause better survivability, (Arora, 2015; Hillier & Lieberman, 2010). In the mutation operator, digits of a chromosome are randomly changed at a given probability, which is typically a low value (Chong & Zak, 2013). Figure 2.10 shows a chromosome where one digit has been changed to create a new solution.

Figure 2.10. The mutation operator

2.4.2   NSGA-II

NSGA-II (Deb et al., 2002) is a genetic algorithm, with a few distinct operators making it useful for MOO. The algorithm has three main features: using elitism, preserving diversity, and emphasising non-dominated solutions (Deb, 2012). After randomly initialising a parent population (P_t) of size N, the solutions are ranked and an offspring population (Q_t) of the same size is created using binary tournament selection, crossover, and mutation. The combined populations form a new population (R_t) of size 2N, of which all solutions are ranked based on their nondomination level. This process of non-dominated sorting involves assigning solutions with a fitness value, according to the non-dominated front that they belong to, as illustrated in Figure 2.11, where the initial front is given the best fitness value, followed by all subsequent fronts thereafter. The fact that previous population (P_t) members are also considered in this ranking process, describes a mechanism known as elitism. The population is cut down to its initial size N, with the best fronts being accepted first. When a partial front is needed to fill the remaining slots of the new population, members of this front are ranked according to a crowding distance calculation as shown in Figure 2.11, which reveals how close each solution is to their neighbouring solutions: the solutions within the least crowded regions are accepted to ensure a better spread (diversity) across the front. All other solutions are rejected, and the cycle continues with the newly formed population (P_{t+1}) becoming the parent population of the next generation.
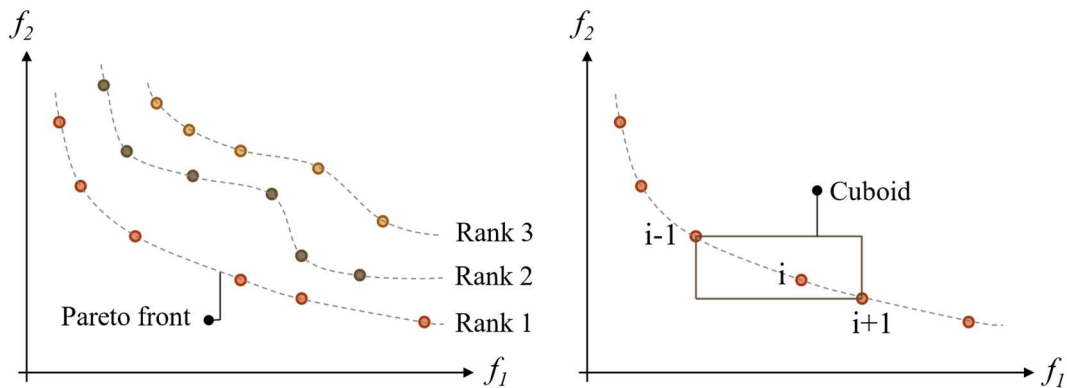

Figure 2.11. Non-dominated sorting scheme (left) and crowding-distance calculation (right)

2.4.3   SPEA-II

SPEA-II (Zitzler et al., 2001) is another type of genetic algorithm: it is similar to NSGA-II but involves a few different mechanisms. In addition to utilising a population, the algorithm keeps an external archive of a constant number of non-dominated solutions, and both archive and population members are given a strength (fitness) value as illustrated in Figure 2.12. This value is representative of the number of solutions the evaluated solution is dominated by as well as the number of solutions that those solutions dominate, so a lower value is more favourable. Environmental selection updates the archive set at each generation, by accepting all non-dominated solutions and when required, the best dominated solution thereafter, until the archive is filled: in the case that

the archive size exceeds the predefined limit, a truncation method is used to iteratively remove those members which have the minimum distance to another solution. The algorithm will terminate once a maximum number of generations is reached or another stopping criterion is met.
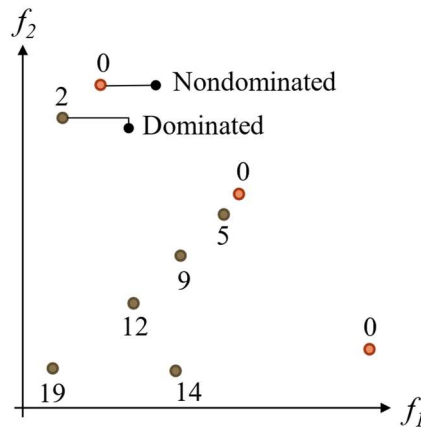
Figure 2.12. SPEA-II fitness assignment scheme, adapted from (Zitzler et al., 2001)

2.5   The hypervolume indicator

The hypervolume indicator is one of the most widely used indicators to evaluate the performance of multi-objective algorithms, as it simultaneously accounts for proximity to the Pareto front, diversity and spread (Guerreiro et al., 2021). It is also referred to as the "size of the space covered" and assesses the quality of Pareto front approximations in a single value, by measuring the size of the dominated portion of the objective space (Emmerich & Deutz, 2014). As illustrated for a minimisation problem in Figure 2.13, the indicator maps the set of non-dominated points found by an optimisation algorithm to a chosen reference point in order to measure the size of the dominated space; therefore, a greater hypervolume indicates a larger dominated region, and thus a better approximation of the pareto front.
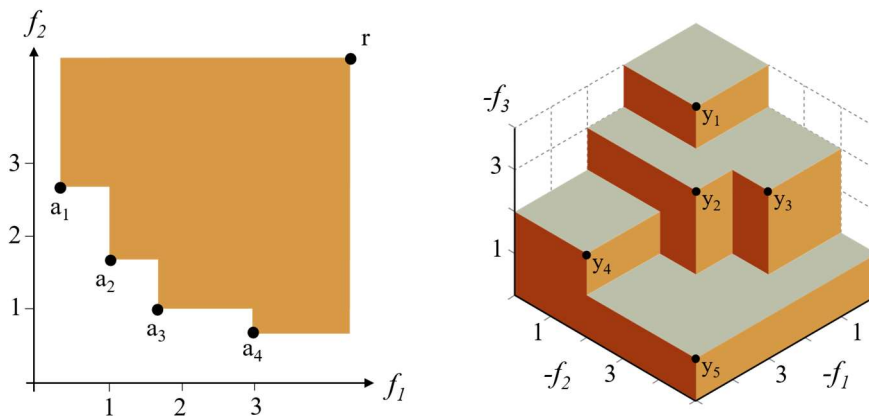
Figure 2.13. The hypervolume indicator, adapted from (Custódio et al., 2012)

2.6   Optimisation in grasshopper

There are a number of existing optimisation tools for GH, which incorporate the algorithms described in the previous sections. This section outlines a number of available tools for both single-objective problems and multi-objective problems: these are illustrated in Figure 2.14.

11
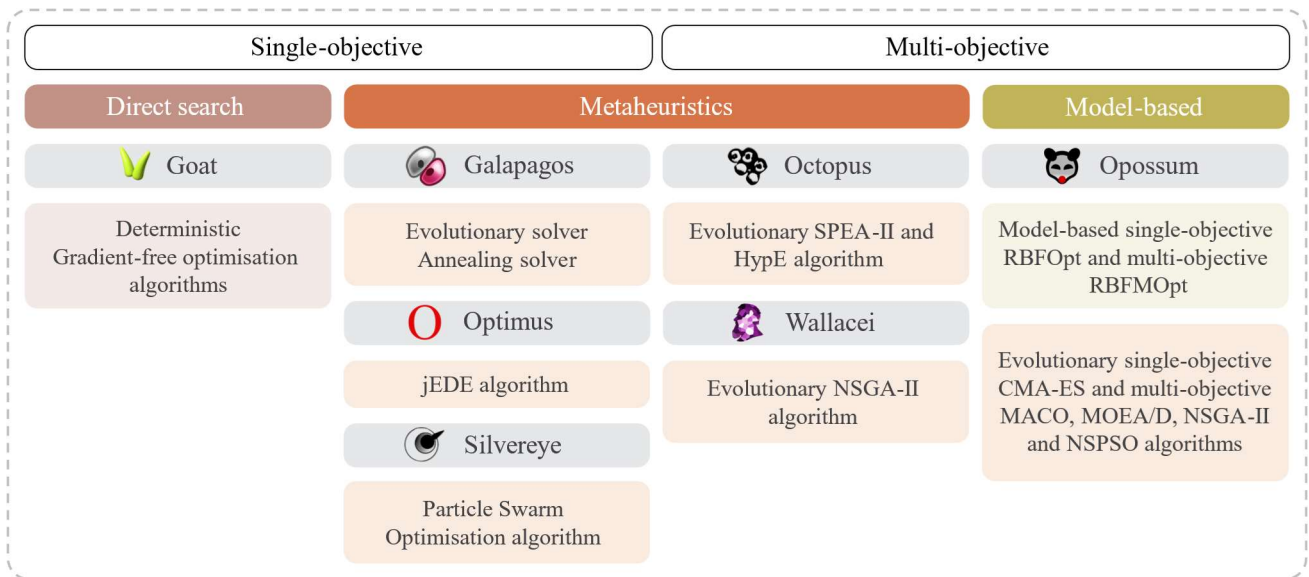
Figure 2.14. Optimisation tools in Grasshopper

2.6.1 Single-objective optimisation tools

Goat (Rechenraum GmbH, 2023) uses a mathematically rigorous approach to provide fast and deterministic results, where the same problem will yield the same optimal outcome with every optimisation run. The plugin contains several gradient free optimisation algorithms: constrained optimization by linear approximation (COBYLA), bound optimization by quadratic approximation (BOBYQA), subplex algorithm (Sbplx), DIRECT, and controlled random search 2 (CRS2). It is also inspired by the graphical user interface (GUI) of Galapagos and uses the mathematical optimisation libraries NLopt.

Galapagos (Rutten, 2013) employs two metaheuristics: the genetic algorithm (GA) and simulated annealing (SA). Although both have advantages and disadvantages, the GA solver is more successful in identifying reliable intermediate solutions early on, while the SA solver is more appropriate for navigating rough landscapes.

Optimus (Cubukcuoglu et al., 2019) is a metaheuristic tool, implementing the self-adaptive differential evolution with ensemble of mutation strategies (jEDE). The control parameters are updated in a self-adaptive approach where the algorithm is adapted according to the nature of the problem, while it uses three mutation operators, as opposed to one.

Silvereye (Cichocka et al., 2017) is based on the particle swarm optimisation (PSO) algorithm. It implements a ParticleSwarmOptimization.dll file, which is a shared library with employment of the core version of the PSO algorithm, and it also emulates the GUI design of Galapagos.

2.6.2 Multi-objective optimisation tools

Due to the focus of this thesis being a multi-objective optimisation problem, Octopus, Wallacei and Opossum were applied and investigated in this study.

Octopus

Octopus (Vierlinger, 2013) is based on the GUI of Galapagos and incorporates SPEA-II (section 2.4.3) as its main algorithm. The tool includes the choice of two different strategies for the reduction operator, which truncates the non-dominated set when it is too large to fit the desired archive size: SPEA-II and HypE reduction. It also contains the choice four different mutation strategies: Polynomial, Alternative Polynomial, HypE, and

Custom Mutation. In this study, the HypE (Bader & Zitzler, 2008) operator was utilised for both strategies. A picture of the plug-in can be seen in Figure 2.15.
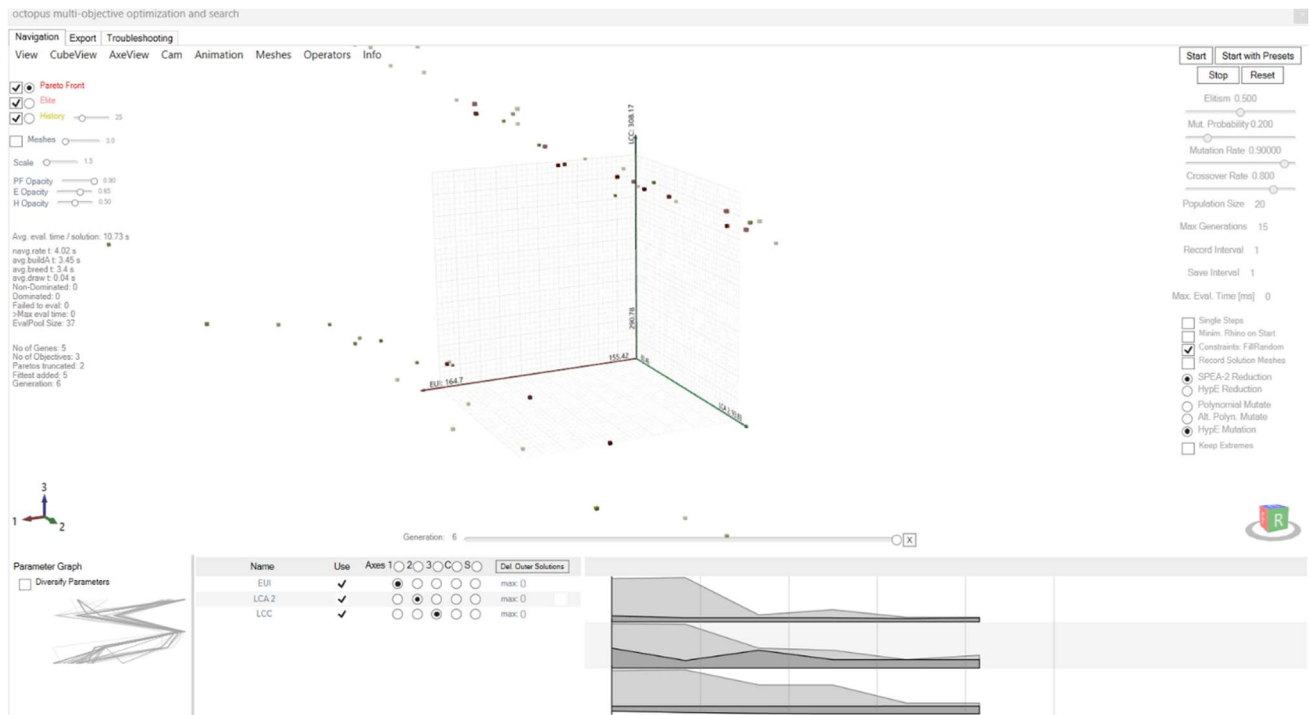
Figure 2.15. Octopus plug-in interface

Wallacei

Wallacei (Makki et al., 2022) integrates NSGA-II (section 2.4.2) as its primary algorithm, as well as utilising the K-means method as a clustering algorithm (Morissette & Chartier, 2013), to further analyse outputted results. The tool also employs the JMetal, LiveCharts and HelixToolkit libraries. Refer to Figure 2.16. for a visual representation of the plug-in.
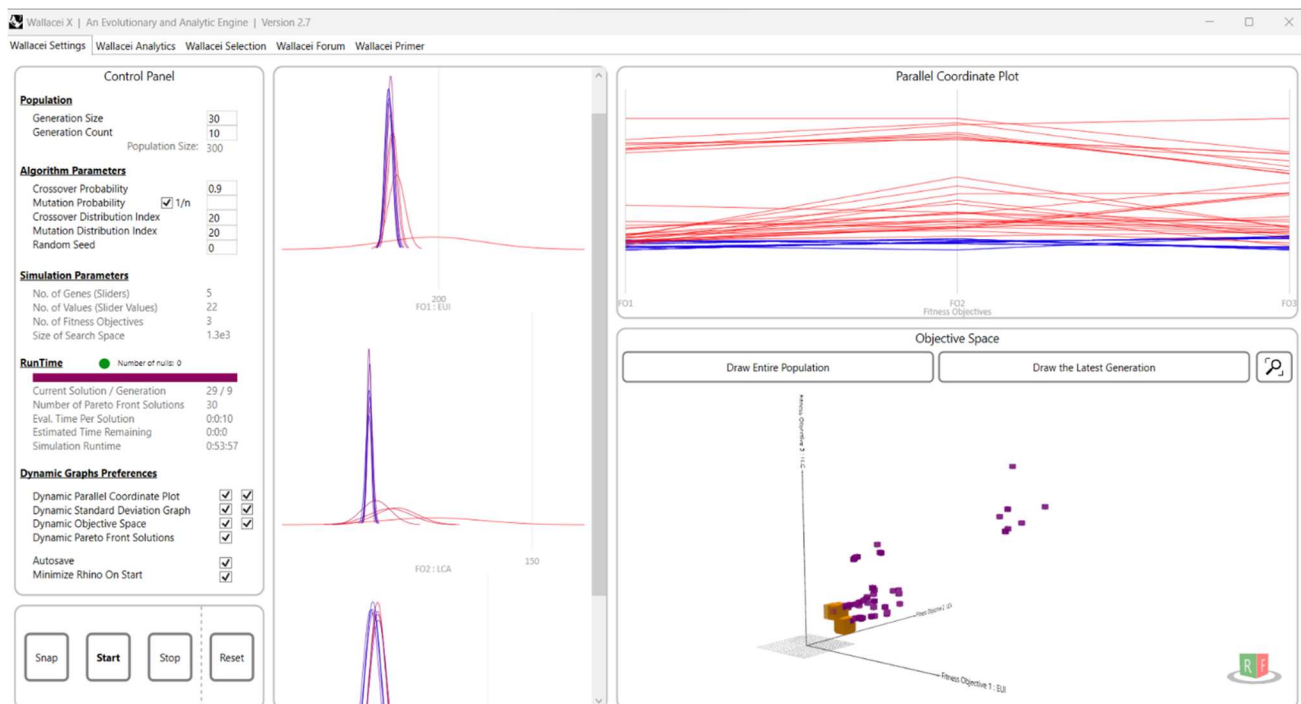
Figure 2.16. Wallacei plug-in interface

Opossum

Opossum (Wortmann, 2017b) can be used for both single-objective optimisation (SOO) and multi-objective optimisation (MOO): for SOO, the tool includes model-based RBFOpt and the evolutionary CMA-ES algorithm while for MOO, it employs model-based RBFMOpt, and evolutionary MACO (Ant Colony), MOEA/D, NSGA-II and NSPSO (Particle Swarm) algorithms from the Pygmo 2 library.

RBFOpt (Costa & Nannicini, 2018) is a model-based optimisation library, where a surrogate model is built to approximate the design space using radial basis functions (Gutmann, 2001). This is an interpolation technique, specified for problems where each evaluation of the objective function is expensive (i.e., a time-consuming computer simulation), meaning that the duration of the optimisation process is dominated by the function evaluations. The goal of this technique is to provide an estimation of the global optimum, using as few function evaluations as possible. RBFMopt is the multi-objective version of this algorithm and was implemented within this study. Figure 2.17. shows a snapshot of the plug-in.
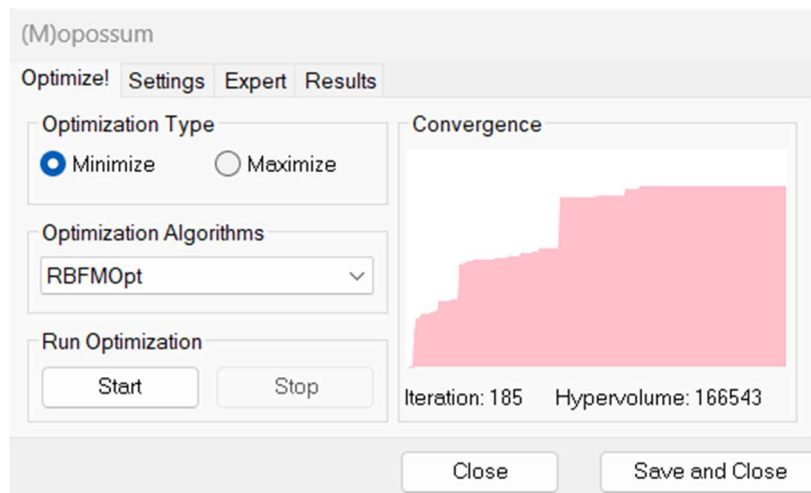
Figure 2.17. Opossum plug-in interface

3   Methodology

An overall workflow of the study is illustrated in Figure 3.1. The same process was carried out for three different alternatives: the first involved testing 1260 combinations of different passive renovation measures on a simple shoebox model; the second considered both the same model and passive renovation measures, in addition to a number of active renovation measures, resulting in a total of 10080 combinations; and the third consisted of testing only the passive measures on a more complex geometry, modelled after a real building.

For each of these three variations, a parametric model was created using the GH plugin within the software Rhino 3D. Most input parameter values were sourced externally and incorporated within the GH script numerically, while the energy model was defined using the Ladybug (LB) and Honeybee (HB) components within GH. Simulations were conducted to obtain three outputs for each iteration: the annual energy need in kWh/(m²·y), the environmental impact in $kgCO_2$ equivalent, and the cost in SEK.

Colibri components in GH were used to carry out a 'brute force' to obtain results for all possible iterations. Once the most favourable solutions were identified, an optimisation was performed using three different optimisation tools available in GH: Octopus, Wallacei, and Opossum. The results were compared to those found by Colibri and the performance of each optimisation tool was evaluated in terms of different criteria, including the time taken and the accuracy of the solutions found.
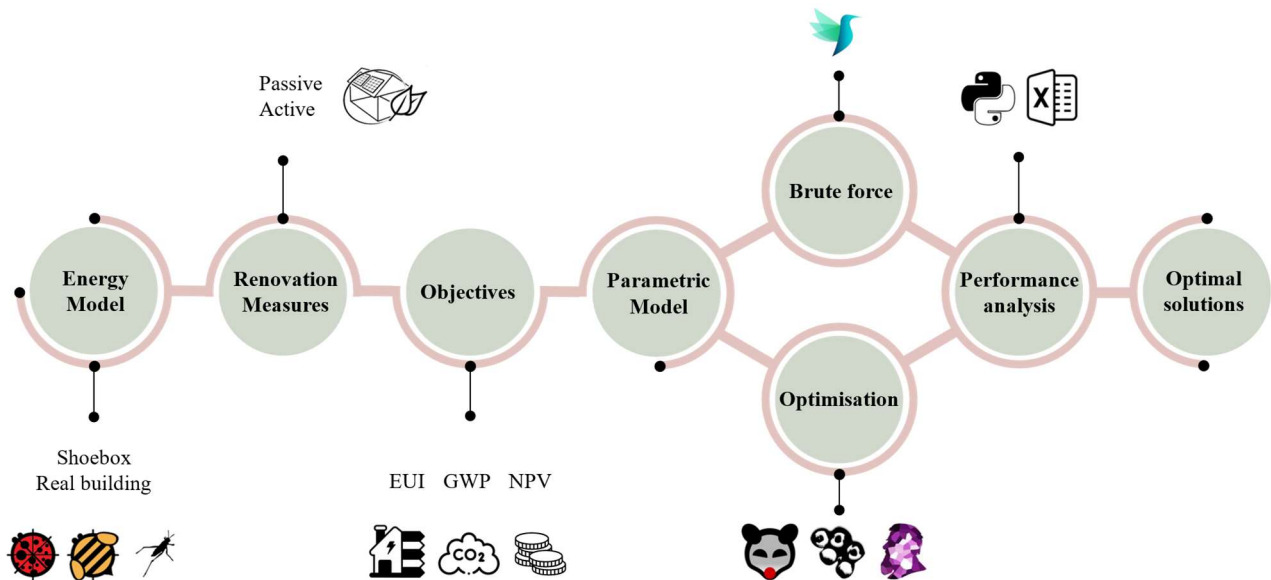
Figure 3.1. Project workflow

3.1   Shoebox model

The main energy model inputs for the shoebox model can be found in Table 3.1. The simulation model consisted of a simple geometry, measuring 5 m × 10 m × 3 m, with a 30 % window-to-wall ratio (WWR) on the south façade. Most inputs corresponded to recommended values from the Swedish building regulations (Boverket, 2017, 2020a) and SVEBY (SVEBY, 2012), while the airtightness was based on a prediction model for airtightness in Sweden (Zou, 2010). The *U*-values were calculated in HB, with the roof, wall, and ground compositions being modelled after typical constructions for a generic Swedish apartment building from the 1970's (Boverket, 2013). The exterior wall composition can be found in Figure 3.2.

Table 3.1. GH script input values for the shoebox model

Description	GH script value	Source/calculation method
Heated floor area (A_{temp}) / m^2	50	GH script
External wall area / m^2	85.5	GH script
Window area / m^2	4.5	GH script
People load / (people/m^2)	0.028	BEN2 / SVEBY
Heating setpoint / °C	21	BEN2 / SVEBY
Building airtightness / (l/(s·m²)) at q50	1.2	Airtightness prediction model
Ventilation air flow / (l/(s·m²))	0.35	BBR
Domestic hot water / (kWh/(m²·y))	25	BEN2 / SVEBY
Household electricity / (kWh/(m²·y))	30	BEN2 / SVEBY
Radiant fraction from electricity / %	70	BEN2 / SVEBY
Property electricity / (kWh/(m²·y))	15	SVEBY
External roof U-value / (W/(m²·K))	1.474	GH script
External wall U-value / (W/(m²·K))	0.376	GH script
Ground U-value / (W/(m²·K))	2.046	GH script
Window U-value / (W/(m²·K))	3.5	GH script
Weather file	Copenhagen	LB EPW map

Existing construction
13 mm gypsum board
95 mm mineral wool
80 mm concrete

Renovation
100 mm - 350 mm insulation
15 mm plaster

Inside   *Outside*
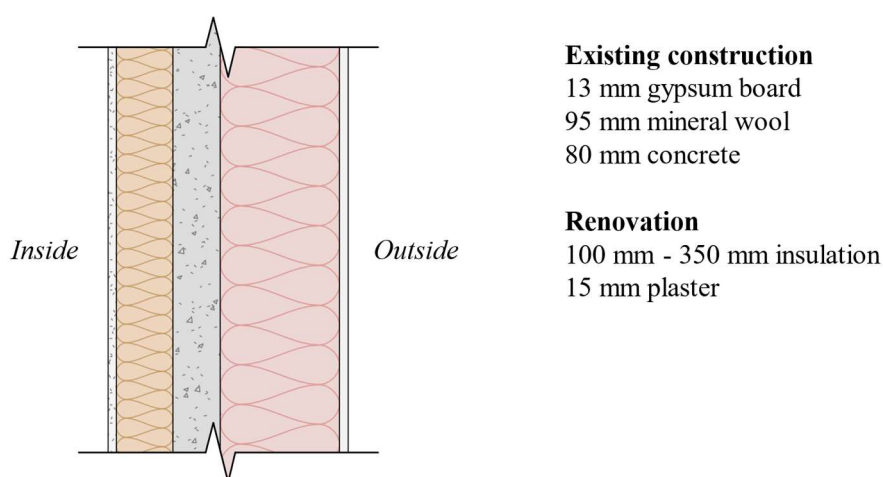
Figure 3.2. Exterior wall construction detail

3.1.1   Control method

In order to validate the results obtained by the simulation model, a control method was carried out. This consisted of making several simultaneous adjustments to the inputs of the GH script and a manual calculation method contained within an Excel spreadsheet. The results for annual heating energy need were compared after each change, to ensure they were viable, before adding more complexity to the GH script. The different adjustments and their results can be found in Appendix A.

3.2   Energy renovations

The renovation measures consisted of testing both passive and active measures, in an attempt to reduce the building's energy need. An overview of these can be found in Figure 3.3. The passive measures involved the reduction of heat losses through the façade, roof and glazing by decreasing their U-values, thus lowering the energy demand for heating. The active measures comprised installing a PV system to reduce the electricity being bought as well as integrating an exhaust air heat pump (EAHP) for the heating of domestic hot water (DHW), which would replace a portion of district heating (DH) with electricity.

Roof insulation type:
- EPS
- Glass wool
- Cellulose fibre

Thicknesses:
(100, 200, 300) mm

PV system size:
- No PV
- 20% roof coverage
- 50% roof coverage
- 80% roof coverage

Façade insulation type:
- EPS
- Glass wool
- Cellulose fibre
- Hemp fibre
- Wood fibre

Thicknesses:
(100, 150, 200, 250, 300, 350)mm

HVAC:
- Exhaust air heat pump

Window measures:
- Keep existing windows
- Install Storm windows
- Replace with triple glazed

Figure 3.3. Overview of considered renovation measures

3.2.1 Passive measures

The façade renovations involved adding a layer of insulation to the outer side of the exterior walls. Five different insulation types were tested: EPS, glass wool, cellulose fibre, hemp fibre and wood fibre, along with six variations in thicknesses ranging from 100 mm to 350 mm, with increments of 50 mm in between. The roof renovations included the addition of three insulation types of EPS, cellulose fibre and glass wool as well as three different thicknesses of 100 mm, 200 mm, and 300 mm.

The window renovations consisted of two different strategies: the first was adding an additional pane of glazing to the existing window, decreasing its U-value to 1.7 W/(m^2·K), while the second was replacing the existing window with a new triple glazed window, which had a U-value of 1.2 W/(m^2·K).

3.2.2 Active measures

Four different PV system sizes were tested for installation on the roof, taking up different proportions of its area: 20 %, 40 %, 60 %, and 80 %. This would enable a fraction of the electricity demand to be self-produced, whilst also allowing for any over-production to be sold back to the grid.

The installation of an EAHP was used for covering the DHW demand, whilst the space heating was still covered by DH. Consideration of a seasonal coefficient of performance (SCOP) of 3 meant that the same DHW need was met with 1/3 of the energy previously required, now in the form of electricity.

3.3 Life cycle assessment

To determine the environmental impact of each renovation measure, a life cycle assessment (LCA) was carried out. The approach taken as well as the boundaries and limitations of this assessment are defined in this section.

3.3.1 Goal and scope

The purpose of this LCA was to evaluate the environmental impact of each renovation measure in terms of their global warming potential (GWP), measured in kgCO$_2$ equivalent. A reference study period of 30 years

was chosen to be somewhat in line with the year 2050, at which the EU aims to be carbon neutral, and to determine the total impact, each functional unit was converted to the functional unit used in this study: $kgCO_2eq./(m^2 \cdot A_{temp})$.

System boundaries

The LCA's system boundaries aligned with those from NollCO$_2$ (SGBC, 2022), with some exceptions, and all considered stages are highlighted in Figure 3.4. The certification system follows the calculation standard SS-EN 15978:2011 to determine the climate impact of a building, while excluding the impact from outside of the system boundary (module D) as well as the end of life stage (modules C1-C4): this is set to zero since waste incineration in Sweden is required to be fossil-free by 2045, and NollCO$_2$'s calculation period of 50 years means that the earliest a building will be disposed of is in 2070. For the disposal of any replacements (module B4) required before 2045, an interpolated climate impact value was used. As modules B1-B3 are difficult to forecast and expected to have a marginal climate impact compared to the rest of the user stage, they are not included in the NollCO$_2$ definition, and since the renovation carried out in this study was considered as refurbishment, module B5 was also excluded from this study. Additionally, the impact of operational water use (module B7) was disregarded due to lack of available information, although the heating of water was included in the impact of operational energy (module B6). The expected service life of various building elements and products was also taken from NOllCO$_2$, which was 30 years for façade and roof elements as well as electrical systems, and 20 years for hot water systems. For façade and roof renovations, the impact of any extra materials and elements required for the installation of insulation, such as studwork, plaster, and fixings, were also excluded from calculations.
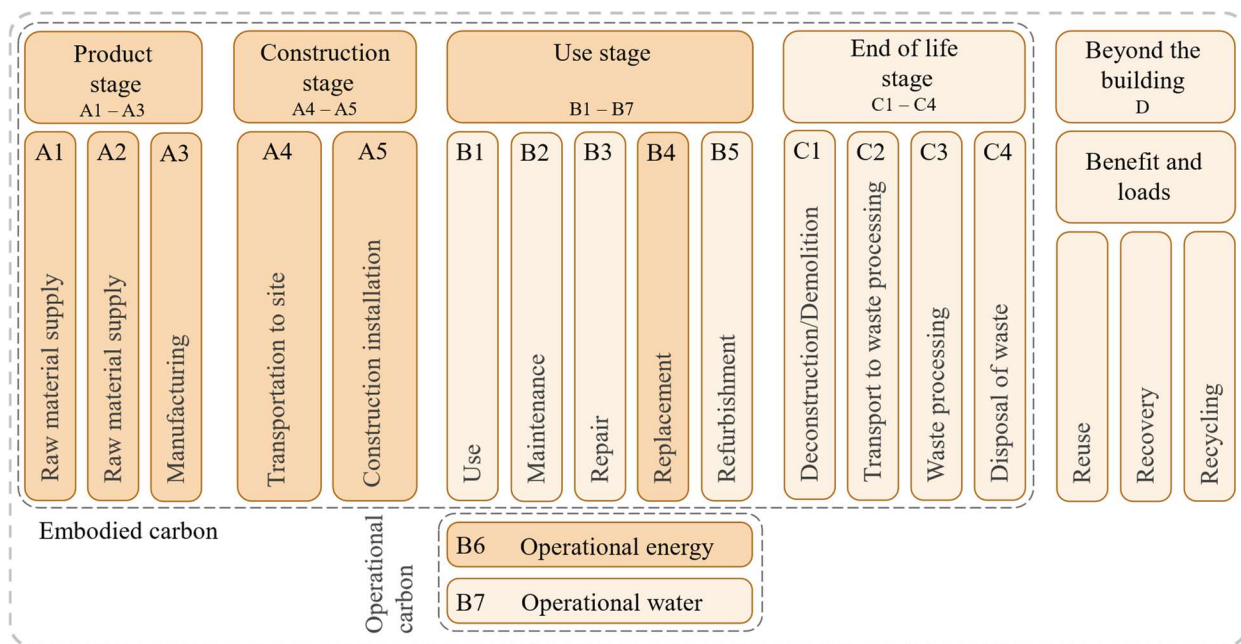
Figure 3.4. SS-EN 15978:2011 life cycle stages, with considered stages highlighted

Limitations

It should be noted that this LCA involved several simplifications and estimations. In addition to the exclusion of certain façade and roof elements, the heat pump calculations also included a lack of detail, as the values sourced from One Click LCA represented generalised data. The impact values obtained for A1-A5 were obtained from Boverket's climate database, which also contains generic values, specifically the transport value, which was not site specific.

3.3.2 LCA of passive measures

The environmental impact of all passive measures for modules A1-A5 was calculated using values from Brimstone: a GH component plugin which retrieves data from Boverket's climate databases. These values,

which can be found in Table 3.2, were then multiplied to their respective mass attributed with the amount of each element used. Other properties regarding the different insulation materials can be found in Appendix B. In addition to glazing, the value used for the storm window installation also included impact values for a sealant, wood trim and spacer, which was sourced from the One Click LCA database (One Click LCA, 2023), while for the installation of the triple glazed window, transportation (C2) and disposal (C4) of the existing window, was also considered. Since the service life for these elements was 30 years, the impact for disposal (B4) was zero.

Table 3.2. Environmental impact of passive renovation measures, in kgCO₂ eq./kg

	A1 − A3	**A4**	**A5**	**A1-A5**
Façade insulation				
EPS	3.200	0.035	0.282	3.517
Glass wool	0.860	0.035	0.077	0.972
Cellulose fibre	0.160	0.035	0.002	0.197
Hemp fibre	0.644	0.117	0.065	0.826
Wood fibre	0.297	0.035	0.028	0.360
Roof insulation				
EPS	3.200	0.035	0.282	3.517
Glass wool	0.890	0.035	0.080	1.005
Cellulose fibre	0.160	0.035	0.002	0.197
Window renovations				
Storm window	2.128	0.035	0.297	2.460
Triple glazed window	2.300	0.042	0.312	2.654

3.3.3   LCA of active measures

The environmental impact values of the active measures can be found in Table 3.3. The PV system impact for stages A1-A5 included a module impact of 797 $kgCO_2$ eq./kW_P, based on a PV module manufactured in China and used in Europe (Müller et al., 2021) as well as a balance of system (BOS) value of 240 $kgCO_2$ eq./kW_P, which considered elements such as mounting, wiring and installation, and was based on a study including a BOS carbon impact of a European energy mix (Friedrich et al., 2021). The inverter replacement was discarded, whilst the service life of other PV system elements was considered to be beyond the study period, resulting in an impact of zero. The EAHP carbon impact values were sourced from One Click LCA, and included the impacts of the HP unit, storage tanks, insulated pipe work and the refrigerant. For the replacement of the EAHP (B4), the impact is the depreciated value after 20 years, which was interpolated in accordance with NollCO₂.

Table 3.3. Environmental impact of active renovation measures

	A1-A5	**B4**
PV system / ($kgCO_2$ eq./kW_p)	1037	-
EAHP / ($kgCO_2$ eq.)	746	152

3.3.4   Operational energy (B6)

The environmental impact of operational energy was calculated using 2019 values from NollCO₂ (SGBC, 2022), which are shown in Table 3.4. Any electricity sold was assumed to reduce the environmental impact, by offsetting coal used within the Nordic Pool's electricity market. This value was linearly interpolated to zero by 2050 in accordance with the EU's carbon neutrality goal, whilst the impact from utilising DH or electricity from a Swedish mix were interpolated to zero by 2045, in line with Sweden's net zero target.

Table 3.4. NollCO₂ 2019 environmental impact and offset values, in kgCO₂ eq./kWh

Swedish district heating mix	**Swedish electricity mix**	**On-site consumption**	**Exported renewable electricity**
0.060	0.022	0	-0.820

3.4 Life cycle costing

To evaluate the economic feasibility and impact of each renovation option, an LCC analysis was carried out. This included the initial cost, the operational cost after implementing the measure, and the cost for any replacements required throughout the study period of 30 years. All costs are shown as a present value, and equations 1 and 2 were used to convert operational costs as well as any replacement costs from their future values.

$$P = F(1 + i)^{-N} \qquad\qquad [P] = SEK \text{ (1)}$$

$$P = A_1 \left[\frac{1 - (1 + g)^N (1 + i)^{-N}}{i - g} \right] \qquad\qquad [P] = SEK \text{ (2)}$$

Where F is the future value, P is the present worth, A_1 is the annual energy cost when $N = 1$, N is the number of years, g is the growth rate and i is the interest rate.

The fixed input values used within LCC calculations are shown in Table 3.5. The district heating price reflects a mean value, sourced from a study that evaluated pricing data for a number of providers within Sweden (Egüez, 2021). The purchasing and selling price for electricity was based on the Nord Pool spot price; it was considered as a fixed price with no price fluctuations throughout the study period (IEA, 2022). A tax reduction of 0.60 SEK/kWh, as currently offered by the Swedish government, was also included in the selling price. For the electricity price growth, an average value was obtained from changes in future Swedish electricity prices, predicted by the European Commission, until the year 2050 (Statista, 2023). Since the price growth for district heating is difficult to predict, the same average value for used. The interest rate was based on the latest value from February 2023, as decided by Sweden's central bank (Sveriges Riksbank, 2023).

Table 3.5. Input values used for LCC calculations

Description	Value	Source/calculation method
District heating price / (SEK/kWh)	0.85	District heating price study
Swedish electricity mix purchasing price / (SEK/kWh)	2.5	National survey report 2021
Swedish electricity mix selling price / (SEK/kWh)	2	National survey report 2021
Electricity price growth / %	0.5	Average of future predictions
District heating price growth / %	0.5	Average of future predictions
Interest rate / %	3	Sveriges Riksbank

The energy prices were multiplied by the annual amount of energy bought and sold throughout the study period, for both electricity and DH. The difference between these values gave the total annual operational costs, which were then added to the initial and replacement costs for each measure, to obtain the total net present value (NPV) for the 30-year calculation period.

3.4.1 Passive renovation costing

The section cost of each type of insulation material and thickness considered in this study can be found in Table 3.6. This includes the material cost, the labour cost for installation, and the hire of scaffolding: these were sourced from Wikells cost books (Wikells Sektionsfakta, 2022), with the exception of the material cost of the hemp fibre and wood fibre insulation, which was obtained from a manufacturer (Optimera, 2023). These values were then multiplied by the respective façade or roof area to provide their final initial cost within the GH script.

Table 3.6. Cost inputs for varying thicknesses of insulation, in SEK/m²

	100 mm	150 mm	200 mm	250 mm	300 mm	350 mm
Façade insulation						
EPS	1777	1888	2083	2252	2559	2727
Glass wool	1589	1728	1933	2183	2450	2590
Cellulose fibre	1615	1690	1864	2084	2375	2489

Hemp fibre	1654	1884	2101	2476	2706	2866
Wood fibre	1581	1771	1937	2293	2479	2850
Roof insulation						
EPS	495	-	621	-	852	-
Glass wool	147	-	217	-	277	-
Cellulose fibre	407	-	433	-	462	-

The costs associated with both window renovation measures were also sourced from Wikells cost books and can be found in Appendix B. The total cost of the storm window renovation consisted of a combination of costs for each individual component, including the additional pane of glazing as well as a sealant, wood trim and spacer channel. For the addition of a triple glazed window, the total cost includes that of the new window, and the demolition and removal of demolition materials of the existing window.

3.4.2 Active renovation costing

The cost of the PV system was sourced from a report, providing average pricing for turnkey grid-connected PV systems (IEA, 2022). A price of 13.56 SEK/W_p was selected and applied to all three PV system sizes: this was an average price for multi-family houses with a size of 50 - 100 kW_p. A value of 17.4 % of the system price was used for the cost of the inverter, which was replaced after 15 years.

The total cost concerning the installation of the EAHP system can be found in Appendix B. This included unit prices, labour costs for installation, and insulated piping as well as a heat pump replacement after 20 years, with removal and disposal of the existing unit. The NIBE S735 and the NIBE UKV 100 were selected for costing the heat pump unit and the accumulator tank (NIBE, 2023), whilst all other costs were sourced from Wikells cost books.

3.5 Real building model

The main inputs for the more complex energy model can be seen in Table 3.7. This was modelled after a residential building located in Skåne county. The building was constructed in 1942 and consisted of four floors and a basement, with a total of 37 apartments. A visualisation of the building model can be found in Figure 3.3.

Table 3.7. GH script input values for the real building model

Description	GH script value	Source/calculation method
Heated floor area (A_{temp}) / m^2	3168	GH script
External wall area / m^2	1396	GH script
Window area / m^2	282	GH script
People load / (people/m^2)	0.032	BEN2 / SVEBY
Heating setpoint / °C	21	BEN2 / SVEBY
Building airtightness / (l/(s·m²)) at q50	1.2	Airtightness prediction model
Ventilation air flow / (l/(s·m²))	0.35	BBR
Domestic hot water / (kWh/(m²·y))	25	BEN2 / SVEBY
Household electricity / (kWh/(m²·y))	30	BEN2 / SVEBY
Radiant fraction from electricity / %	70	BEN2 / SVEBY
Property electricity / (kWh/(m²·y))	15	SVEBY
External roof U-value / (W/(m²·K))	0.735*	Building data
External wall U-value / (W/(m²·K))	0.711*	Building data
Ground U-value / (W/(m²·K))	2.046	Building data
Window U-value / (W/(m²·K))	2.697*	Building data
Weather file	Copenhagen	LB EPW map

*These values include a thermal bridge factor of 20 %

The geometry of the building and its surroundings were created in GH, by retrieving values from a database containing building specific information such as a WWR for each façade, which was used to generate the

windows. The resulting building model took approximately 6 minutes to simulate, which was unfeasible for this study. To ensure a reasonable simulation runtime, slight simplifications were made to the model by combining the windows into a single glazing area for the north, east and west facades as well as removing unnecessary geometries from the surroundings. The model consisted of five thermal zones, with one for each floor and had a simulation runtime of 1 minute and 28 seconds. The initial building geometry and the simplified model can be found in Appendix C. Running the more complex model with all passive and active measures would be impractical; therefore, the active measures were disregarded to scale down the total simulation time to a reasonable number for this study.

3.6 Optimisation tools

After running all iterations with Colibri, several optimisation runs were executed with Octopus, Wallacei and Opossum. Each optimisation plugin was connected to sliders corresponding to the different decision variables as well as the three objective functions shown in Figure 3.5, where the aim was to identify the combination of decision variable values that would minimise the three objectives. All values can be found in Table 3.8. The results were retrieved as a list comprising the values found at each iteration of the algorithm, and post-processed in excel.
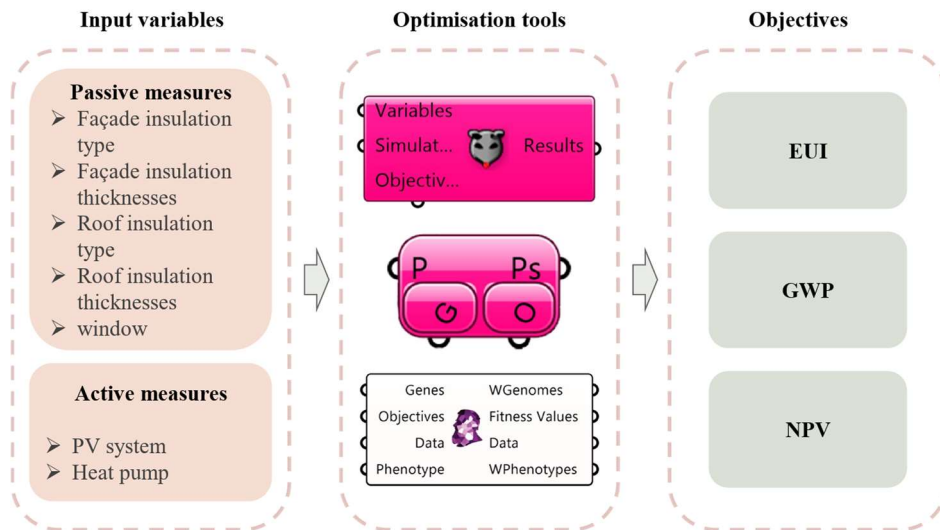
Figure 3.5. Simulation input variables and objectives

Table 3.8. Decision variables and objectives

Input parameters		
Façade insulation type	EPS	
	Glass wool (GW)	
	Cellulose fibre (CF)	
	Hemp fibre (HF)	
	Wood fibre (WF)	
Façade insulation thicknesses	0	mm
	100	mm
	150	mm
	200	mm
	250	mm
	300	mm
	350	mm
Roof insulation type	EPS	
	Glass wool	
	Cellulose fibre	
Roof insulation thicknesses	0	mm
	100	mm

		200	mm
		300	mm
Window type		Existing window (EW)	
		Storm window (SW)	
		Triple glazed window (TW)	
PV system size / (% of roof area)		0	%
		20	%
		50	%
		80	%
HVAC		No change	
		EAHP (COP = 3)	
Objectives			
Energy Use Intensity (EUI)		/ (kWh/m²A_{temp})	
CO₂ emissions (GWP)		/ (kgCO₂eq./m²A_{temp})	
Net Present Value (NPV)		/ (SEK/m²A_{temp})	

The different input settings for each tool can be found in Table 3.9. The multi-objective model-based algorithm RBFMOpt was selected for Opossum, whilst Octopus and Wallacei contained the genetic algorithms SPEA-II and NSGA-II, with the HypE algorithm being selected for the reduction and mutation operators in Octopus. Due to the stochastic nature of these algorithms causing a different result being produced for every run, each run was repeated three times. For the shoebox model, three different population sizes of 10, 20 and 30 were tested when only considering passive measures, whilst a population size of 30 was used for all remaining runs, which considered both passive and active measures and the real building model. The convergence criteria differed between each optimisation plug-in. For opossum, the algorithm was set to stop after a solution was repeated a certain number of times without showing any improvement in the results: three different values of 10, 20 and 30 were tested for this. The maximum number of generations for Octopus was set to 10; however, the algorithm could stop before reaching this number, on the premise that another integrated criterion was fulfilled. The only termination criterion for Wallacei was to specify a certain number of generations that the algorithm would execute, which was also set to 10.

Table 3.9. Simulation tools inputs/settings

Tools	Algorithm	Population size	Convergence factor
Colibri	-	-	-
Opossum	RBFMOpt	-	10, 20, 30 results without improvement
Octopus	SPEA-II & HypE	10, 20, 30	10 generations and other background criteria
Wallacei	NSGA-II	10, 20, 30	10 generations

3.6.1 Criteria for comparing optimisation tools

The performance of each optimisation plug-in was evaluated using a set of criteria, including the time taken for each run and the accuracy of the results, while the user interface of each tool was also reviewed. Two values were generated to assess how successful each plugin was in identifying the most optimal solutions. The first was an adaptation of the hypervolume indicator: a volume was calculated for each solution, using a reference point with coordinates corresponding to a value 10 % greater than the maximum value for each objective found with Colibri, after which the solutions with the 10 largest volumes found by both the optimisation tool and Colibri were added and compared, by calculating their percentage difference. Due to this indicator only considering values that would be located near the centre of the pareto front, another value was used to evaluate how successful each plug-in was in identifying the minimum value of each of the three objectives: this was done by calculating the percentage difference between the minimum value found by the optimisation tool and the minimum value from Colibri, before averaging the three percentages.

4 Results

This chapter presents the study's results, which were obtained through a three-step process. Firstly, the analysis focused on the performance of three optimisation tools for the shoebox model with only passive renovation measures. This involved comparing the time and accuracy of the results to those extracted from Colibri for all possible combinations. The same assessment process was then repeated for the shoebox model with both passive and active measures, allowing for an investigation into the optimisation tools' performance with a greater number of iterations. Finally, a simulation of a real case building was conducted to explore the impact of the optimisation tools on a more complex energy simulation.

4.1 Shoebox model with passive measures

The simulation performance outcomes for all runs considering the shoebox model with passive renovation measures are visible in Table 4.1. The table provides information about the total number of simulations and their corresponding time, based on the input settings for each plug-in. Additionally, it displays the optimisation tools' success rates in finding the hypervolume, which represents the 10 most optimal solutions that minimised all objectives, and in identifying the three extreme cases, which contained the minimum value for each objective.

The results of this set of simulations shows that all optimisation runs managed to reduce the simulation time by at least 74 % compared to Colibri. Regarding the optimisation tools' settings, the table shows that reducing the population size in Octopus and Wallacei as well as decreasing the required number of results without improvement in Opossum, lead to a decreased simulation time whilst simultaneously lowering the success rates. Although each optimisation run started with a random selection, and the number of iterations and total running time may vary with the same settings, the results did not show a significant difference in success rates.

Opossum generally showed the greatest time reduction, and although the shortest run for Octopus took a similar time of around 10 minutes, its success rates were lower, at 81 % and 91 % compared to 92 % and 99 %. As the stopping criterion for Wallacei meant that it would always execute 10 generations, its shortest run took approximately 18 minutes, while the runs with the same population size input only slightly differed in their results, as the total number of iterations did not change. Despite Octopus and Wallacei showing a relatively similar performance, Wallacei's fixed termination setting meant that Octopus often converged sooner; however, for the same number of generations, Octopus took longer, as it evaluates twice the number of population individuals within the initial generation.

Table 4.1. Total simulation performance results for passive measures, with shoebox model

	Plug-in	Population size input	Number of generations	Time	Time reduction	Total number of iterations	Success rate for hyper-volume	Success rate for minimum values
Passive measures	**Colibri**	-	-	3:50:28	-	1260	100 %	100%
	Octopus	30	5	0:33:37	85 %	180	96 %	99 %
			10	0:59:58	74 %	330	98 %	100 %
			7	0:44:56	81 %	240	99 %	100 %
		20	11	0:44:51	81 %	240	99 %	99 %
			7	0:29:30	87 %	158	98 %	100 %
			12	0:45:42	80 %	258	96 %	99 %
		10	4	0:09:40	96 %	50	81 %	95 %
			8	0:15:47	93 %	90	96 %	99 %
			6	0:13:13	94 %	70	88 %	99 %
	Wallacei	30	10	0:52:13	77 %	300	98 %	99 %
			10	0:53:57	77 %	300	97 %	99 %
			10	0:54:35	76 %	300	97 %	99 %
		20	10	0:36:05	84 %	200	97 %	98 %
			10	0:36:28	84 %	200	93 %	99 %
			10	0:36:35	84 %	200	95 %	98 %
		10	10	0:17:33	92 %	100	84 %	97 %
			10	0:17:47	92 %	100	90 %	97 %
			10	0:17:45	92 %	100	90 %	97 %
	Results without improvement input							
	Opossum	30	-	0:39:11	83 %	194	99 %	100 %
			-	0:39:57	83 %	186	100 %	100 %
			-	0:26:48	88 %	128	99 %	100 %
		20	-	0:10:42	95 %	52	96 %	100 %
			-	0:38:11	83 %	183	99%	100%
			-	0:25:18	89 %	126	99%	99%
		10	-	0:18:18	92 %	85	96%	99%
			-	0:09:33	96 %	42	92%	99%
			-	0:14:35	94 %	67	96%	99%

Figure 4.1. displays one set of the shoebox simulation results with passive measures, using Colibri and the optimisation tools. The scatterplots below provide a visual depiction of the difference between the number of iterations conducted using the optimisation tools and their respective areas of focus, compared to all 1260 combinations simulated with Colibri. Additionally, the size of each dot on the scatterplot represents the hypervolume for each combination, indicating its fitness regarding all objectives. For each tool, it is clear that their focus was within the region where the hypervolume was the largest, whilst fewer evaluations occurred in less favourable areas, where the hypervolume was smaller. It can also be seen that a significant number of iterations were avoided by the optimisation tools, causing them to be much more efficient than Colibri.
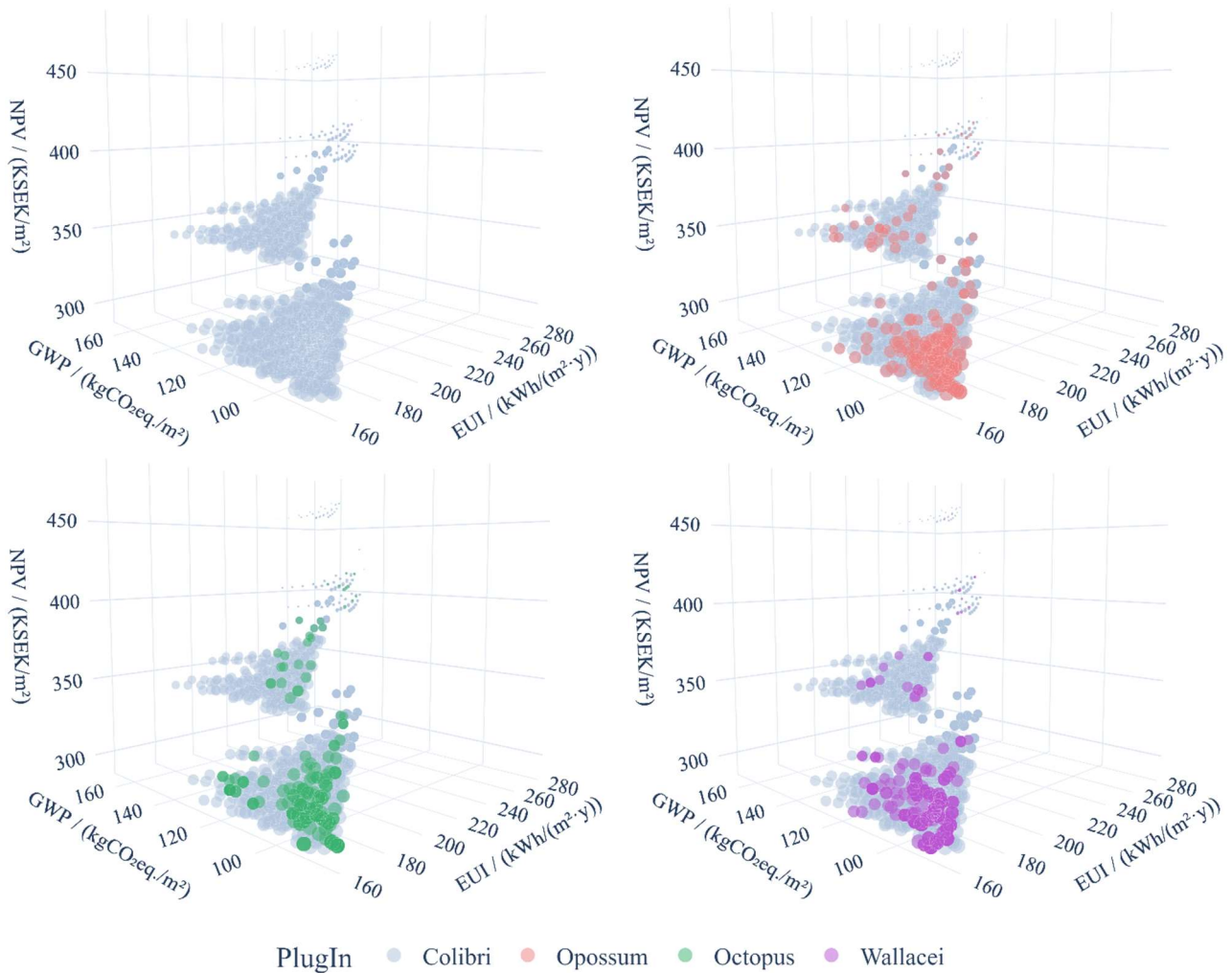
Figure 4.1. Shoebox model simulation results with passive measures, for Colibri and optimisation plug-ins

The hypervolume-based distribution of simulated solutions is illustrated in Figure 4.2. The adapted hypervolume indicator determines the proximity of a solution to the central region of the pareto optimal front, where a larger hypervolume implies a closer proximity. The graph includes a box plot illustrating the range of the entire set of solutions as well as the range of each quartile within this set, while each dot represents a single solution, and the concentration of solutions at each value is indicated by the width of the violin plot. It can be seen that Colibri's solutions are more widely dispersed, with a more even concentration throughout, while the solutions from the optimisation tools are more clustered toward the upper part of the graph with greater hypervolume values. Although the total range of solutions is similar for all plug-ins, the optimisation tools show a greater median hypervolume value than Colibri, while most of their solutions also have hypervolumes greater than Colibri's median.
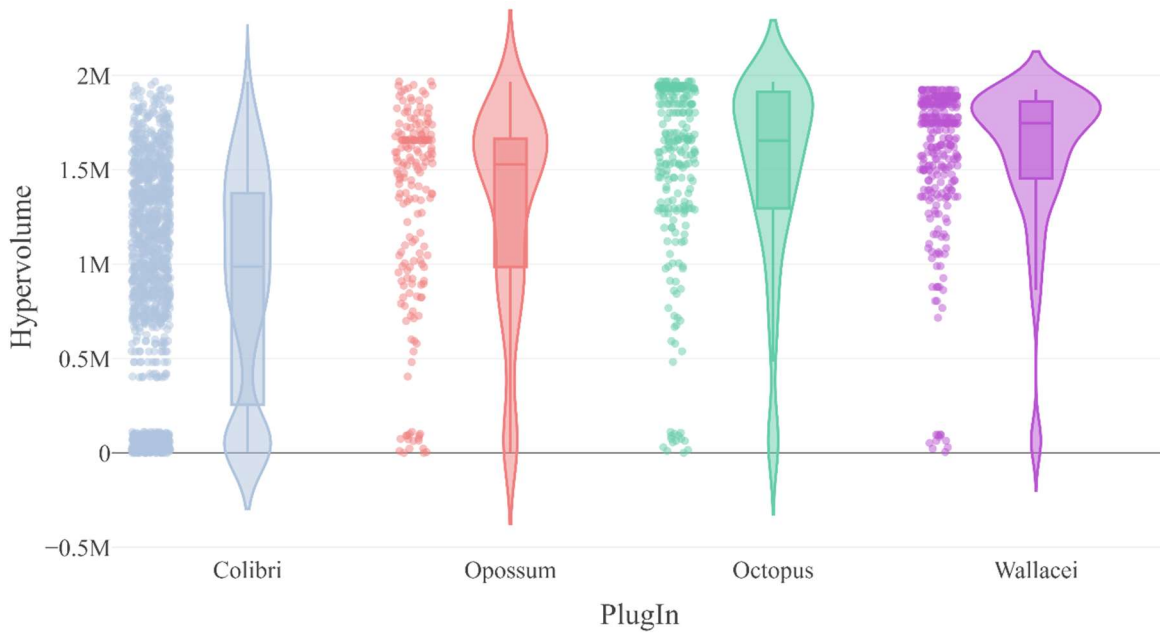
Figure 4.2. Violin plot of solution distribution and their hypervolume using Colibri and optimisation tools for the shoebox model with passive measures

Figure 4.3 and Figure 4.4 show an overview of the input parameters and results for the 10 most optimal solutions in terms of hypervolume, found by Colibri and each optimisation tool, for the shoebox model with passive measures. The exact values for each package can be found in Appendix D. It is clear that the packages found by all optimisation tools generally aligned with those from Colibri, as the coordinates follow similar paths, with most of them overlapping. All plug-ins found that cellulose fibre (CF) and glass wool (GW) were the best insulation types for both the walls and the roof, while keeping the existing window (EW) or installing a storm window (ST) was more favourable than a triple glazed window (TW). There was no clear optimal wall insulation thickness for any plug-in, as the thicknesses ranged between all possible values, with the exception of Wallacei's optimal solutions, which only contained thicknesses of 300 mm and 350 mm. Additionally, Colibri and Opossum found that 200 mm of insulation was generally best for the roof, while Octopus and Wallacei showed more variation. Although there were some slight differences in the parameters within the optimal packages found by each tool, it is clear that each one successfully minimised all objective functions, as the results aligned closely with those from Colibri.
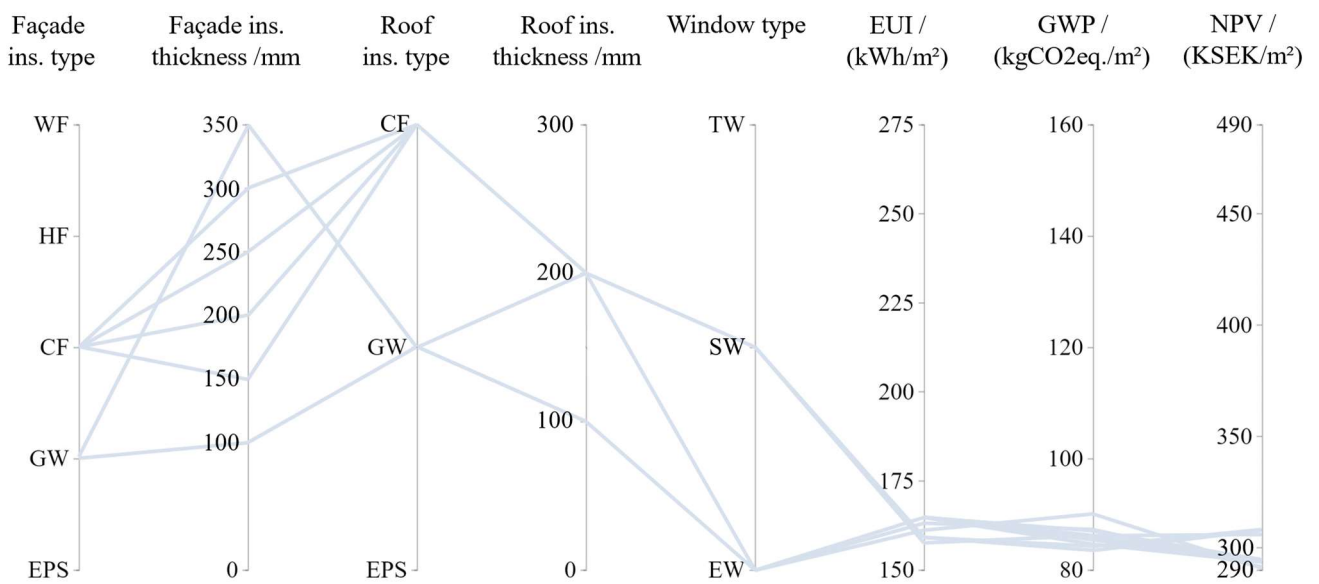
Figure 4.3. Input variables and results for the 10 most optimal renovation packages found by Colibri, for the shoebox model with passive measures
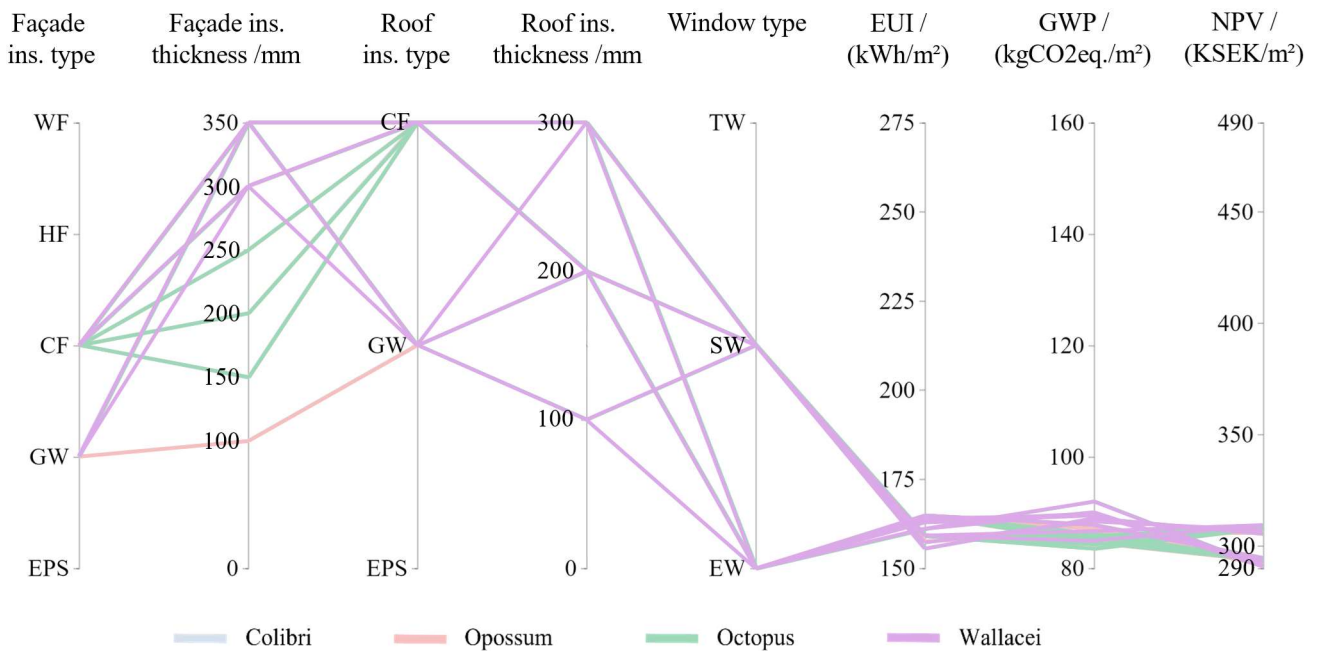
Figure 4.4. Input variables and results for the 10 most optimal renovation packages found by Colibri and each optimisation tool, for the shoebox model with passive measures

Table D.6 in Appendix D also shows the renovation packages that resulted in the minimum value for each of the three objectives, as found by each plug-in. The optimal package in terms of reaching the minimum energy use and cost both included the use of glass wool insulation with a 350 mm thickness, while the best window type was a triple glazed window for energy use and the existing window for cost. For a minimum environmental impact, the best renovation package consisted of cellulose fibre insulation with a 150 mm thickness for the wall and a 300 mm thickness for the roof, as well as the installation of a storm window. All optimisation tools managed to find either the same renovation packages or slight variations, while the minimum results for all optimisation tools were all within 0 % to 2 % of colibri's result. For the minimum energy use, all optimisation tools found most of the optimal parameters, with the exception of a different window type for Octopus and Wallacei, and a different insulation type for Opossum. Additionally, Opossum and Octopus both found the optimal renovation package for minimising the carbon impact, whilst Wallacei found a slightly different insulation thickness. Lastly, the optimal renovation packages for cost were found by both Octopus and Wallacei, whilst Opossum found a different insulation thickness.

4.2 Shoebox model with passive and active measures

The optimisation tools' performance results for all runs concerning the shoebox model with passive and active measures can be found in Appendix D. Table 4.2 is a representative of all runs, as it showcases one of the runs for each plug-in and provides information about the total number of simulations conducted and their corresponding time. It shows that optimisation tools drastically reduced the total simulation time for 10 080 iterations, from 30 hours and 49 minutes with colibri to around 1 hour with Octopus and Wallacei using genetic algorithms and to 23 minutes with Opossum using model-based algorithms, while maintaining a high success rate in finding optimal solutions. The table also indicates that the increase in the total number of combinations does not necessarily lead to a proportional increase in the time and accuracy of the results for the optimisation tools. As an example, the Opossum simulation converged after 112 iterations, which represents only 1 % of the total possible combinations, while still achieving a high success rate of 98 % for hypervolume and 100 % for finding minimum objective values; similarly, Octopus and Wallacei required only 330 and 300 simulations, respectively, to find optimal solutions.

	Plug-in	Population size input	Number of generations	Time	Time reduction	Total number of iterations	Success rate for hyper-volume	Success rate for minimum values
Passive & active measures	**Colibri**	-	-	30:49:05	-	10 080	100 %	100 %
	Octopus	30	10	0:59:55	97 %	330	98 %	99 %
	Wallacei	30	10	0:54:35	97 %	300	99 %	100 %
	Results without improvement input							
	Opossum	30	-	0:23:28	99 %	112	98 %	100 %

Figure 4.5 highlights all solutions found by one run with each the optimisation tool as well as the total number of iterations from Colibri, considering the shoebox model with passive and active measures. With the size of each dot representing the hypervolume, the graphs illustrate that the area of focus for each optimisation tool remained where the largest hypervolumes occurred and all objectives would be minimised. It should also be noted that the influence of climate compensation and the selling of electricity when adding different sizes of PV systems, resulted in a significant decrease in GWP as well as the NPV, causing the solutions to be split into four distinct sets of data.
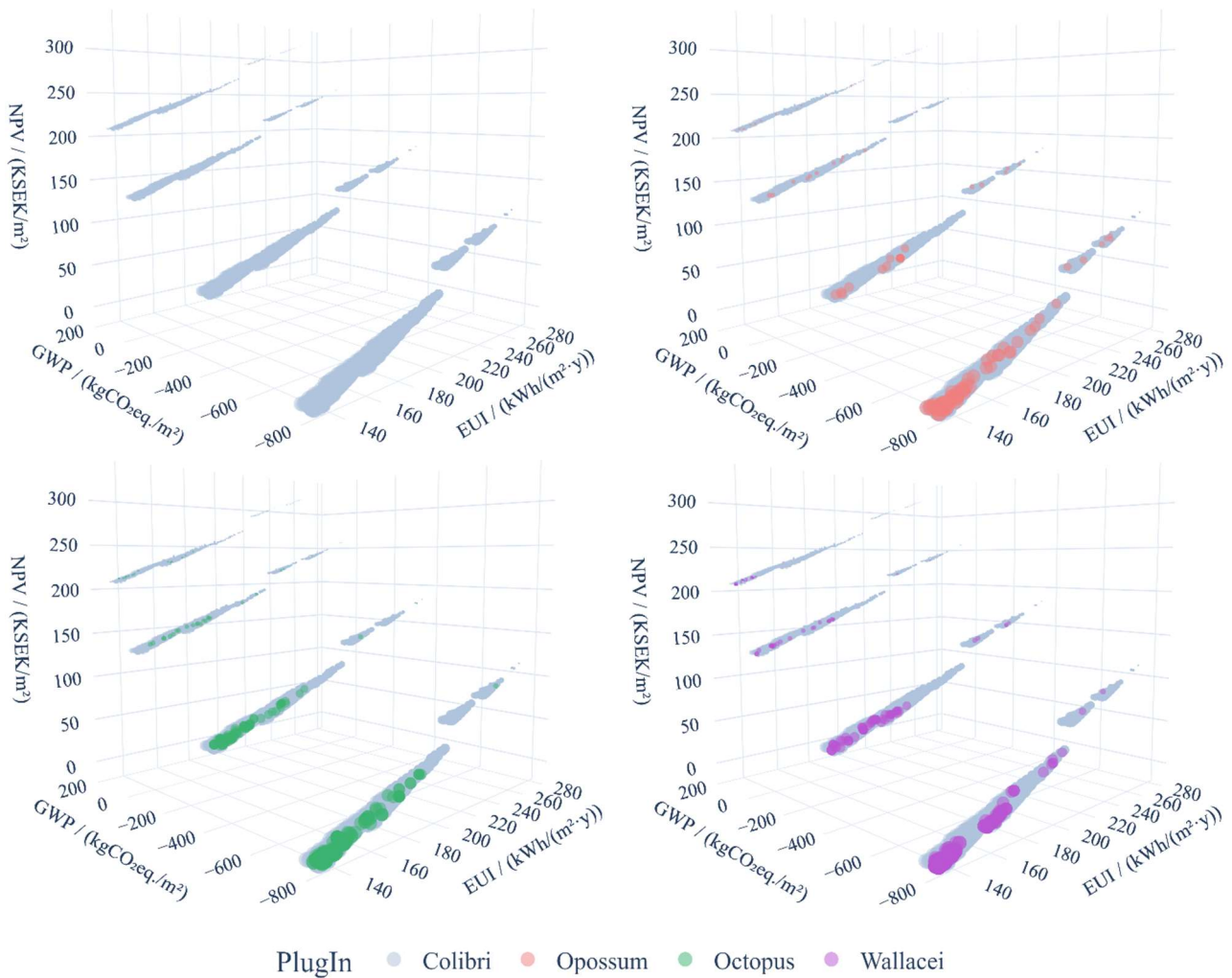
Figure 4.5. Shoebox model simulation results with passive and active measures, for Colibri and optimisation plug-ins

Figure 4.6. illustrates the distribution of solutions regarding the shoebox model with passive and active measures. Once again, each point alongside the violin plots represents a solution in terms of its hypervolume. The graph shows a clear contrast between the distribution and number of solution evaluations when using optimisation tools compared to Colibri. The width of the violin plot and the boxplot in the background indicate that Colibri tested many solutions that are far from the Pareto front and outside the problem's area of interest, whereas the hypervolumes of the simulation results with optimisation tools have a larger median, and their focus is on the upper part of the graph, as they avoided running numerous unnecessary simulations. Due to Opossum running fewer iterations of 112 compared to 330 and 300 for Octopus and Wallacei, its focus within the upper part of the graph is less obvious; however, their success rates remain similar.
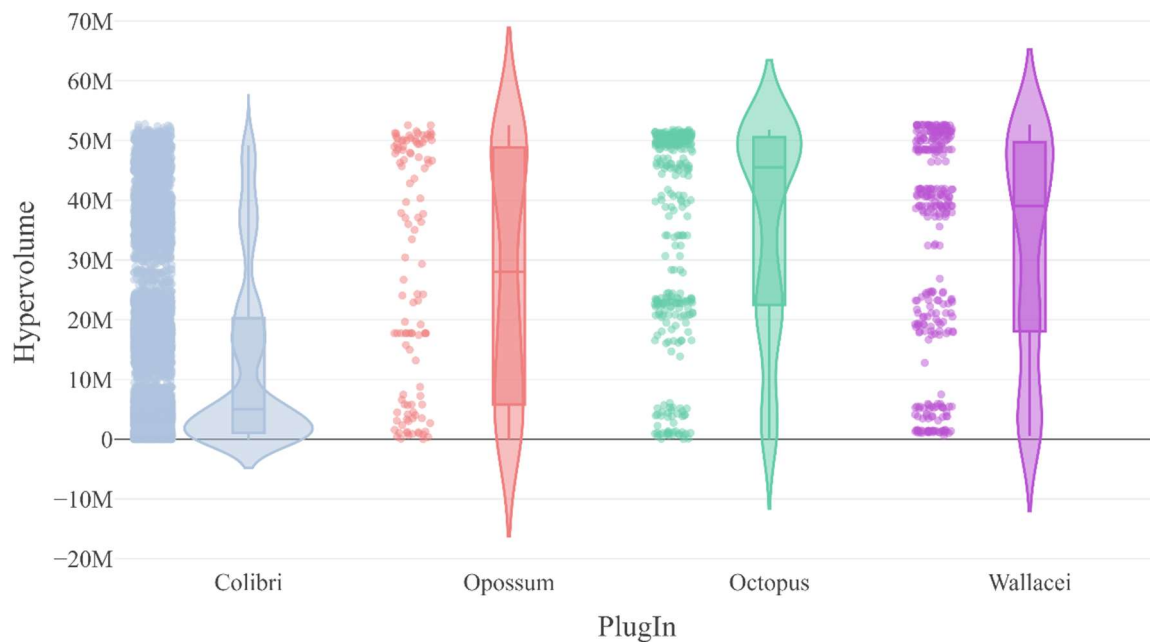
Figure 4.6. Violin plot of solution distribution and their hypervolume using Colibri and optimisation tools for the shoebox model with passive and active measures

Figure 4.7 and Figure 4.8 depict the 10 most optimal renovation packages, in terms of hypervolume, found by Colibri and the optimisation tools, for the shoebox model with passive and active measures. The exact values for each package can be found in Appendix D. It is clear that all optimisation tools successfully minimised the objective values, as they aligned closely with those extracted from Colibri, while the optimal variables slightly changed from those identified for only passive measures. The best insulation type identified by Colibri was mostly GW, and occasionally CF. Wallacei and Opossum found both GW and CF, while Opossum generally found GW for the wall and CF for the roof, and Octopus mostly found GW. Opossum and Octopus also identified some additional insulation types: wood fibre (WF) and hemp fibre (HF) for the wall and EPS for the roof. The optimal insulation thicknesses for Colibri varied; however, the best thickness was mostly 350 mm for the wall and 300 mm for the roof. For all optimisations tools, the best thickness was generally 300 mm or 350 mm for the wall and 300 mm for the roof. The optimal choices for each plug-in were the same for all other parameters: these consisted of either installing a storm window or a triple glazed window as well as an EAHP and the largest PV system, which covered 80 % of the roof area. The addition of PV also meant that all optimal solutions had a negative carbon impact, and therefore achieved carbon neutrality.
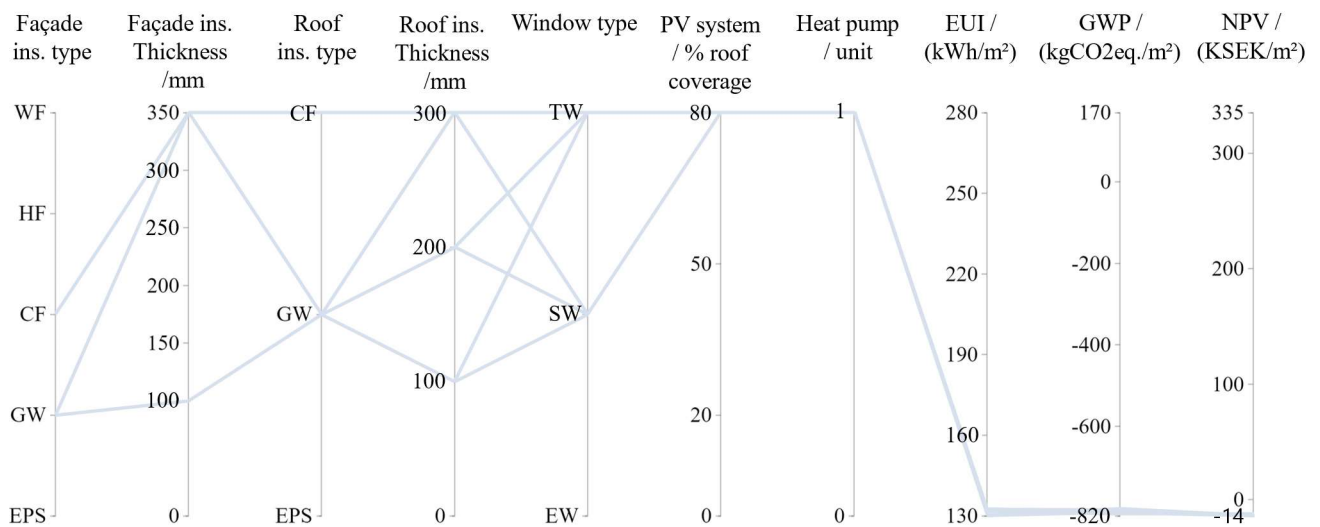
Figure 4.7. Input variables and results for the 10 most optimal renovation packages found by Colibri, for the shoebox model with passive and active measures

Figure 4.8. Input variables and results for the 10 most optimal renovation packages found by Colibri and each optimisation tool, for the shoebox model with passive and active measures

Table D.11 in Appendix D also shows the renovation packages that resulted in the minimum value for each objective, for Colibri and the optimisation tools. When including active measures, the optimal passive renovation parameters for minimising the energy use remained the same (GW insulation at maximum thickness and triple-glazed windows), while the best active strategies consisted of installing an EAHP and the largest PV system. This was also the optimal case for minimising costs, other than a different optimal window type, which was a storm window. For a minimum environmental impact, the optimal passive strategies also remained the same as previously (CF insulation with a 150 mm thickness for walls and a 300 mm thickness for the roof), while the optimal active measures included the largest PV system and no heat pump. All optimisation tools either found the same renovation packages or slight variations, with the largest difference from a minimum objective value being only 3 %.

4.3 Real building model with passive measures

Table 4.3 shows the simulation results for the real building model with passive and active measures, using Colibri and the optimisation tools. Originally, the generated model took approximately six minutes for each iteration, resulting in an estimated total of 5 days for all 1260 iterations. After simplifying the model, the time for each iteration reduced to 1 minute and 28 seconds, which was around eight times longer than one simulation with the shoebox model. As a result, simulating the building model with passive measures using both Colibri and the optimisation tools took significantly longer in total. All optimisation tools had high success rates, above 90 %; however, Opossum remained the most efficient optimisation tool, as it reduced the original time by 91 %, while Octopus and Wallacei had a similar time reductions of 76 % and 79 % respectively.

Table 4.3. Simulation performance results for passive measures, with real building model

	Plug-in	Population size input	Number of generations	Time	Time reduction	Total number of iterations	Success rate for hyper-volume	Success rate for minimum values
				Full model estimation				
	Colibri	-	-	5 days	-	1260	100 %	100 %
				Simplified model				
Real building with Passive measures	**Colibri**	-	-	32:33:31	-	1260	100 %	100 %
	Octopus	30	10	7:58:15	76 %	330	94 %	99 %
	Wallacei	30	10	6:58:07	79 %	300	97 %	100 %
	Results without improvement input							
	Opossum	30	-	2:54:08	91 %	112	96 %	100 %

Figure 4.9 shows the results for all iterations executed by Colibri and the optimisation tools, for the real building model with passive measures, where the hypervolume is indicated by the size of each point. Much like the case with the shoebox model, the area of focus for all optimisation tools was where the hypervolume was largest, meaning that they were successful in minimising all three objectives. It is also clear that a large number of the less favourable cases were not evaluated by the optimisation tools, allowing them to save a significant amount of simulation time.
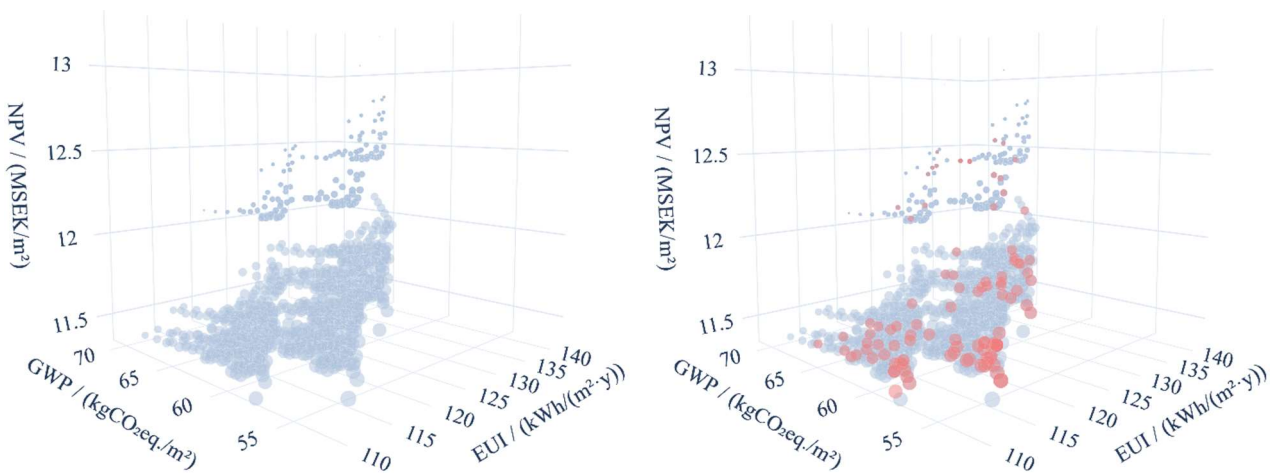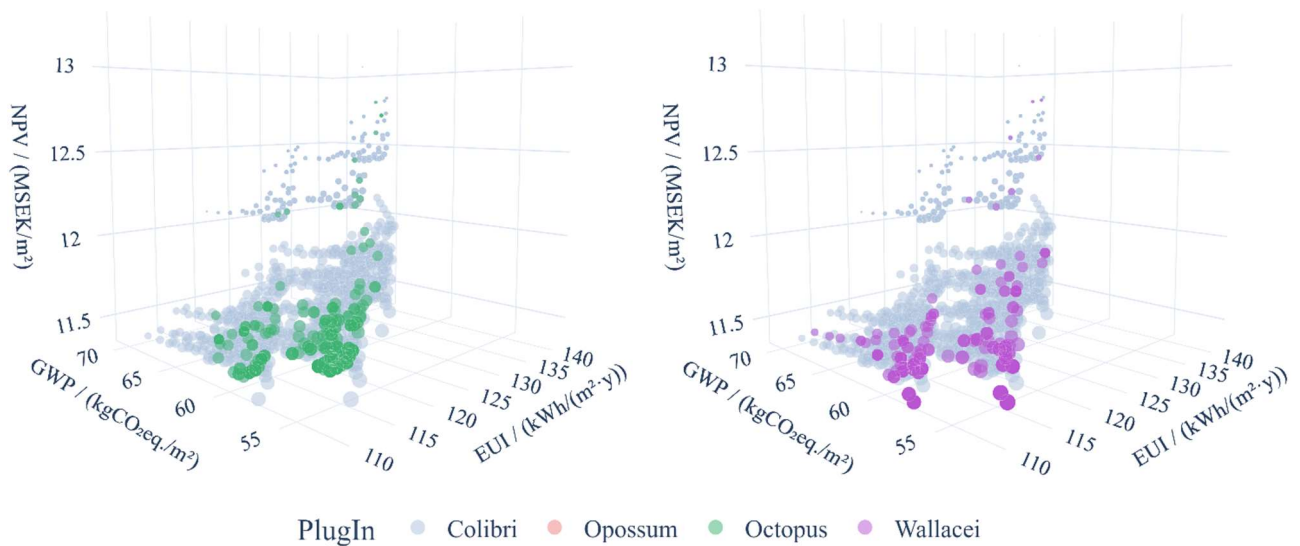
Figure 4.9. Real building model simulation results with passive measures, for Colibri and optimisation plug-ins

Figure 4.10 and Figure 4.11 show the 10 most optimal renovation packages in terms of hypervolume, for the real buiding model with passive measures. The exact values for each package can be found in Appendix D. Colibri found that the best packages either included GW or CF insulation at varying thicknesses, and the installation of storm windows. The package with the largest hypervolume included GW insulation with a 100 mm thickness for the walls and a 300 mm thickness for the roof, in additon to a storm window: this gave an EUI of 111.2 kWh/(m²·y), a carbon impact of 52.2 kgCO$_2$eq/m², and a NPV of 11.5 MSEK/m². Overall, all optimisation tools had the same parameters within their 10 most optimal renovation packages. For the largest hypervolume, Wallacei found the same package as Colibri, while Octopus and Opossum's package varied slightly; however, the difference in their objective values was very minimal. Opossum's optimal package included CF insulation of 200 mm for the walls and 300 mm for the roof, along with storm windows, resulting in an EUI of 112.3 kWh/(m²·y), a carbon impact of 52.2 kgCO$_2$eq/m², and a NPV of 11.6 MSEK/m². The best renovation package for Octopus comprised of GW insulation with a 250 mm thickness for the walls and a 100 mm thickness for the roof as well as installing storm windows. This gave an EUI of 112.2 kWh/(m²·y), a carbon impact of 54.1 kgCO$_2$eq/m² and a NPV of 11.6 MSEK/m². All packages managed to successfully reduce the carbon impact from the building's original impact of 66 kgCO$_2$eq/m²; however, as no climate compensation measures (i.e., installing a PV system) were included for the real building, no package could reach carbon neutrality.
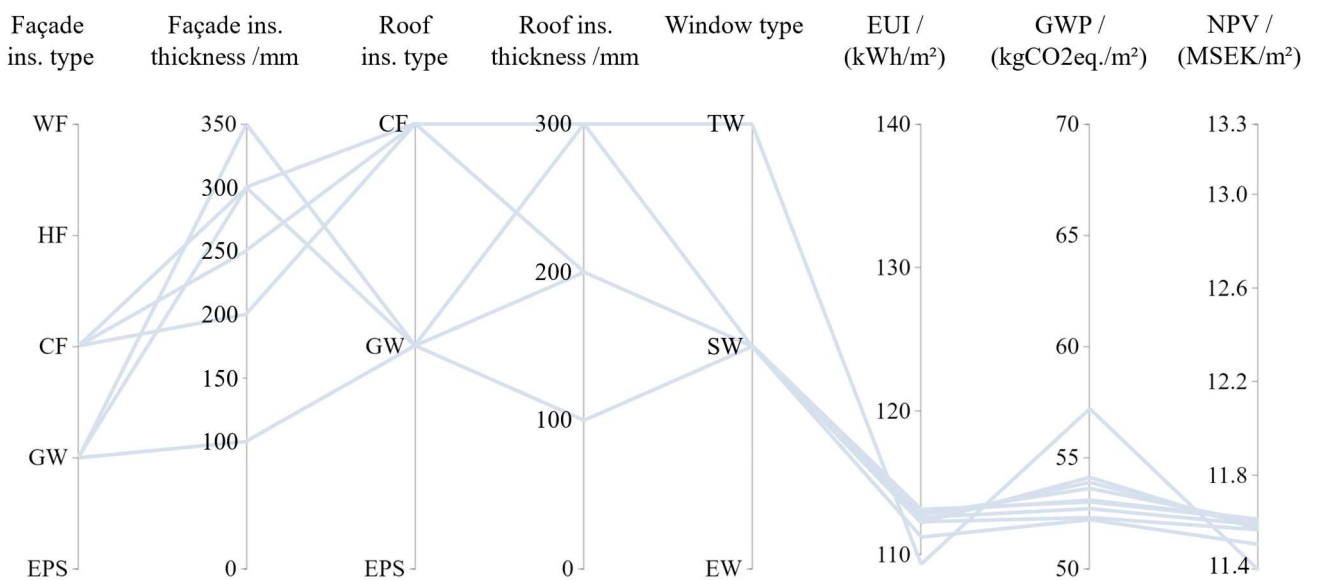
Figure 4.10. Input variables and results for the 10 most optimal renovation packages found by Colibri, for the real building model with passive measures
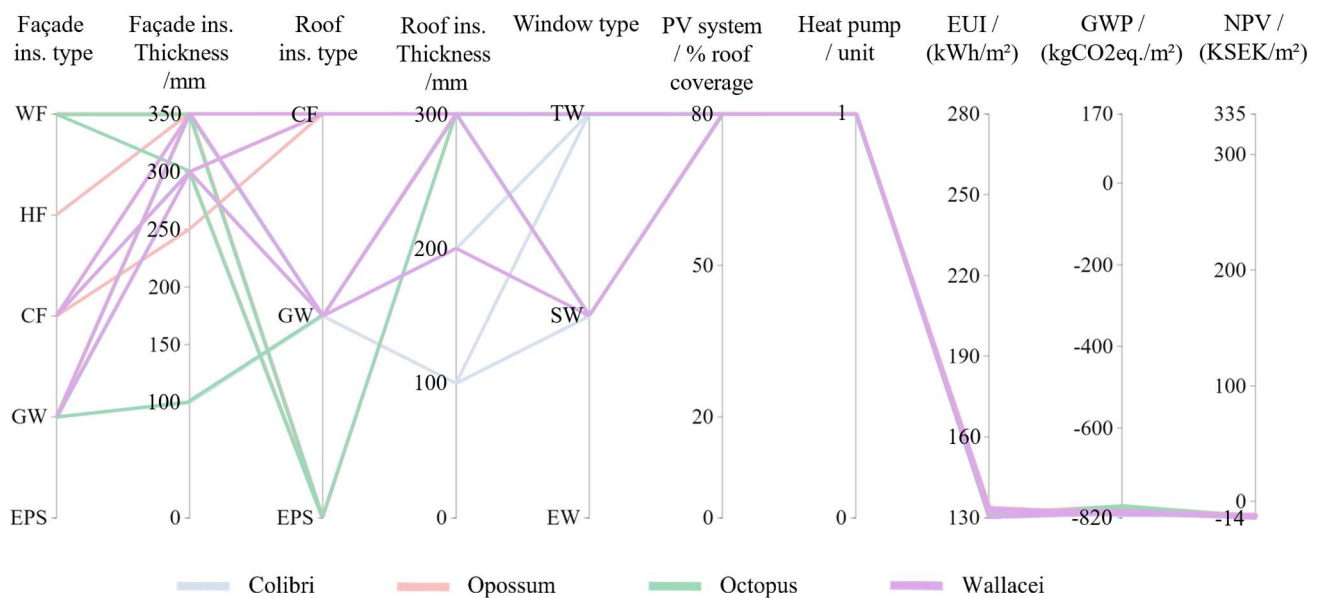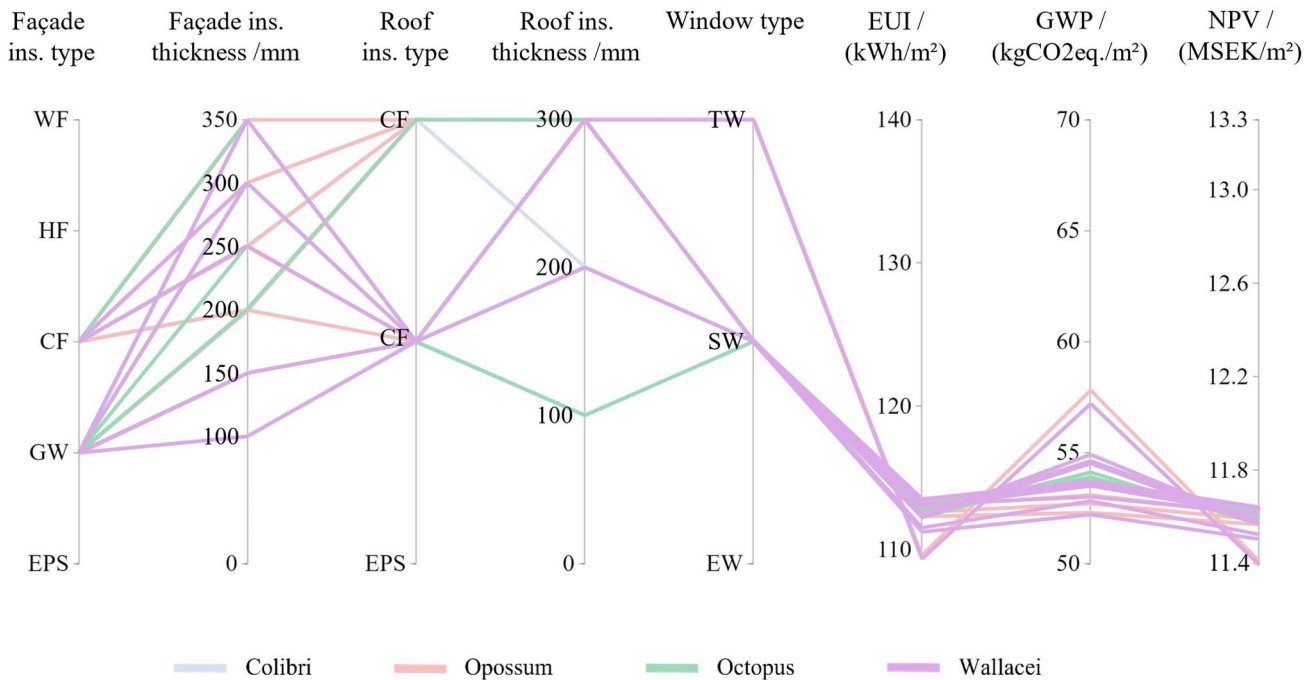
Figure 4.11. Input variables and results for the 10 most optimal renovation packages found by Colibri and each optimisation tool, for the real building model with passive measures

Table D.16 in Appendix D also shows the renovation packages that minimised each objective, for the real building model with passive measures. Colibri found that the renovation package most optimal in terms of both energy savings and costs consisted of GW insulation with a 100 mm thickness for the wall and a 300 mm thickness for the roof as well as the installation of triple glazed windows, giving an EUI of 109.3 kWh/(m^2·y) and a NPV of 11.4 MSEK/m^2. Wallacei found the same package, while Octopus and Opossum found slightly different insulation thicknesses; however, these still resulted in very similar EUI's of 110.1 kWh/(m^2·y) and 109.6 kWh/(m^2·y), and NPV's of 11.4 MSEK/m^2. For the lowest environmental impact, Colibri found that the best renovation package consisted of the same insulation type and thicknesses that minimised energy use and costs, in addition to installing storm windows: this resulted in a carbon impact of 52.2 kgCO$_2$ eq./m^2. While Wallacei also found this package, Octopus and Opossum had slightly different variables, although these still achieved very similar results of 53.5 kgCO$_2$ eq./m^2 and 52.3 kgCO$_2$ eq./m^2 respectively.

5 Discussion

This study revealed that the employment of optimisation plug-ins in GH can be extremely beneficial for parametric studies, by reducing the simulation time, while retaining the accuracy of the results. Each tool managed to substantially reduce the total simulation time compared to Colibri and was successful in identifying a reasonable number of optimal solutions. Although the main accuracy indicator used in this study failed to judge the diversity and spread of the pareto front, it is clear that each plug-in is able to converge in the most optimal region of the search space, whilst neglecting a large number of unnecessary iterations.

5.1 Overview of results

The inner workings of the algorithm are an important factor to consider when deciding which type of algorithm to employ, as this choice is problem dependent. The results showed that Opossum, which utilised a model-based algorithm, generally performed the best in terms of time, whilst still achieving a high level of accuracy. With the shoebox model and passive measures, the longest run for Opossum still reduced the total time by 84 %, whilst achieving both a hypervolume success rate and a minimum objective value success rate of 100 %. Since these methods specifically attempt to minimise the number of required function evaluations by approximating the design space with a surrogate model, they are likely more suitable for optimisation problems involving building performance simulations or other cases where function evaluations are more time intensive. Genetic algorithms are also beneficial, with the right input settings making them more efficient; however, they might be more suitable for optimisation problems that require less computational effort, as the number of iterations becomes less important with a less time intensive function evaluation.

Increasing the number of iterations with the shoebox model did not mean that the runtime of the optimisation tools also increased at a similar rate; however, a critical factor was increasing the time taken for each iteration, as this substantially increased the total simulation time. With an increase in the number of iterations from 1260 (4 hours) to 10 080 (31 hours), the runtime for each optimisation tool remained relatively similar and still did not exceed one hour, with the longest run for Opossum taking around 40 minutes compared to 35 minutes previously, while the success rates for hypervolume and the minimum objective values also remaining at similar values of 99 % and 100 %, compared to 100 % for both. Since changing the shoebox model to a real building model increased each function evaluation from 10 seconds to around 6 minutes, the total time taken with Colibri would have taken approximately 5 days. The simplified model still had a longer function evaluation of 1 minute and 28 seconds, so the total time taken with both Colibri and the optimisation tools increased at a similar rate. Although the number of iterations remained at 1260, the increased complexity of the model meant that the total simulation runtime with Colibri increased from approximately 4 hours to 33 hours, while with the addition of active measures (10 080 iterations), Colibri would have taken around 10 days. Therefore, without optimisation, this parametric analysis would have been infeasible. Additionally, in the real case, where the semi-automatically generated geometry had not been simplified and kept its original function evaluation time of 6 minutes, the total simulation time when considering both passive and active measures would have taken approximately 42 days. In this case, it would also be impractical to carry out a parametric analysis, even with the employment of optimisation tools.

5.1.1 Renovation packages

When testing the passive measures for the shoebox model, there were a lot of similarities between the most optimal renovation packages. These generally included either glass wool or cellulose fibre insulation for both the walls and the roof, and either keeping the existing window or installing a storm window. Although there was no obvious optimal insulation thickness, the optimal thicknesses tended to be balanced between both the walls and the roof, i.e., when the wall insulation thickness was at a higher value, the roof insulation thickness was generally at a lower value and vice versa. Additionally, when the wall insulation was at the maximum thickness of 350 mm, there was no difference in energy savings between the three different insulation thicknesses for the roof. After adding active measures, the optimal insulation variables remained relatively similar, whilst the other optimal measures comprised of either installing a storm window or a triple glazed window as well as an EAHP and the largest PV system, which covered 80 % of the roof area.

When changing the geometry from a shoebox model to a real building model, the optimal passive renovation packages consisted of the same measures, with the exception that the installation of storm windows was more prevalent. For solely minimising energy use and costs, the same package was found to be optimal for both: adding 100 mm of glass wool insulation to the walls and 300 mm to the roof, alongside installing tripe-glazed windows. This reduced the EUI by 32.4 kWh/(m²·y) and the NPV by 1.9 MSEK/m², while the carbon impact was lowered by 8.8 kgCO₂eq./m² over the 30-year study period. On the other hand, aiming for a minimum carbon impact gave a slightly different optimal package as it included the installation of storm windows rather than triple-glazed windows. The resulting reduction in GWP was 13.8 kgCO₂eq./m². This package was also the most optimal for balancing all three objectives, and it resulted in energy savings of 30.5 kWh/(m²·y) and a cost saving of 1.8 MSEK/m².

When comparing these results with a previous study that evaluated potential renovation packages for the decarbonisation of Swedish multi-family homes (Daya & Nolan, 2022), some similarities in the resulting most optimal packages can be identified. Much like this study, the most favourable renovation packages often included installing storm windows and cellulose fibre insulation for the walls; however, EPS and wood fibre insulation (not tested in this study) tended to be most optimal for the roof. Although active measures were only included for the shoebox model, the same study also found that installing an EAHP and a PV system with 80 % roof coverage was favourable. Due to the same NollCO₂ definition considered for climate compensation, cases with a PV system had the potential to achieve carbon neutrality, with larger PV systems being more likely in doing so.

5.2   Implementing optimisation plug-ins

To ensure the most favourable result, certain aspects should be kept in mind when working with optimisation algorithms and these GH tools. For genetic algorithms, the chosen population size will influence the outcome, so a reasonable size should be selected for a sensible compromise between speed and accuracy. If time is of less importance, a larger population size will allow for more accurate results, as the algorithm will execute more function evaluations per generation, whilst a smaller population size will involve fewer function evaluations and could cause the algorithm to converge too early, providing less accurate results. The user should also be aware that the algorithms integrated within these tools are stochastic, meaning they employ elements of randomness, which will lead to a different outcome for each optimisation run; regardless, the results revealed that the utilisation of these tools will be valuable, as they have the ability to identify optimal solutions in a shorter period of time compared to running every possible iteration. In most cases, running the optimisation more than once could still be advantageous as it will provide an extra degree of certainty whilst still being less time intensive than executing a 'brute force'. Alternatively, if the user does not want to solely rely on optimisation tools, they could also be used as a starting point for identifying the variables that appear within the most optimal solutions and use this information to narrow down the total number of iterations or evaluate some additional iterations with the most favourable variables. For example, if the best insulation thickness ranges between 150 mm to 250 mm, the user can limit the number of insulation thickness variables to be within this range, compared to previously tested values between 100 mm to 400 mm.

Given the successful performance of these optimisation tools for identifying optimal building renovation strategies in this study, the application of optimisation tools can be extended to other building-related areas, such as design stage problems and daylight analysis. For example, in the design stage, optimisation tools can be utilised to identify the most efficient building forms for reducing energy consumption and carbon emissions, while increasing the daylight availability within the space.

5.3   Octopus

Octopus contains some useful features alongside having a number of downsides. Although the main algorithm cannot be changed from SPEA-II, the interface allows the user to have control over mutation and crossover operators. A 'save interval' feature can be utilised, providing an interval of generations after which the Grasshopper file is saved to prevent data loss, in the case that Rhino crashes, while any unwanted situations can

be tracked by specifying a maximum evaluation time: when a function evaluation takes longer than this, the solution will be added to a collection that can be debugged afterwards, by reinstating them in the 'Troubleshooting' tab. Additionally, the interface provides a 3D visual of the solutions found, which is useful for an initial overview of the optimisation process and the solution history of each generation as well as identifying the Pareto front. Despite this, the tool has no post processing features and selecting a specific solution from the solution space only reveals the objective values, without specifying its input variables. Moreover, the interface does not contain an input for defining whether each objective should be minimised or maximised, while the user also has no control over the termination criteria of the algorithm, other than setting a maximum number of generations.

5.4 Wallacei

Much like Octopus, Wallacei has several useful attributes, while also lacking in some areas. The interface provides a lot of information regarding the optimisation run, including the evaluation time per solution, the total simulation runtime as well as estimating the remaining simulation time, which can be useful for the user, especially when simulating for longer periods of time. It will also identify any null solutions, giving the user an indication of any issues within the script: the algorithm will continue the optimisation process by replacing these solutions in case the user doesn't find the issue to be critical enough to stop the process. Again, there is no option of changing the algorithm from NSGA-II, yet the user has control over the mutation and crossover operators as well as a random seed option, which can be useful when comparing results after making a change, as it creates the same initial population. The interface also provides a lot of result visualisations and embedded features for analysing the optimisation process and recalling specific solutions, as can be found in Figure D.1 and Figure D.2 in Appendix D. Solutions can be clustered in each generation while a diamond fitness chart is provided for selected solutions, showing each objective in terms of their fitness value, although the interface does not contain a graph showing the input variables of a selected solution. Additionally, the user is able to export the phenotype and other specific data of any type associated with each solution, that is not used within the optimisation process. For example, obtaining the insulation volume or the amount of electricity sold to the grid. Unlike Octopus and Opossum, which require results to be exported for post processing in another environment (e.g., Excel or Python), Wallacei enables the user to post process the results within GH. However, in contrary to the maximum generation input in Octopus, Wallacei will not converge before reaching the number of generations specified, regardless of whether the results reveal a satisfactory improvement, since it does not define any other stopping criteria.

5.5 Opossum

Other than containing no result visualisations, with exception of the hypervolume trend, Opossum provides notably more user control for certain features than both Octopus and Wallacei. In addition to including choice between minimisation and maximisation within the interface, the user has the option of choosing between model-based as well as evolutionary algorithms, while the settings for these can also be changed within the interface. The termination criteria can also be defined in a number of different ways: the optimisation can be stopped based on a maximum number of iterations, a specific number of iterations without further improvement, and a time limitation. Lastly, the solutions are presented in a straightforward manner, as the tool provides the hypervolume, objectives and parameters for each solution, and sorts them based on their rank.

6 Conclusion

Given the urgency to undertake significant renovations of existing buildings for Sweden to attain carbon neutrality by 2045, or as early as 2030 for 23 Swedish cities, parametric modelling becomes a very useful approach, as it allows for the automation of testing a significant number of potential renovation strategies to identify the most optimal solutions for achieving this goal. However, exploring all possible solutions is extremely time consuming, making such an evaluation unfeasible. Optimisation can be a suitable approach to accelerate this process, thus saving time and costs, whilst still identifying the most optimal solutions.

This study evaluated the optimisation tools Octopus, Wallacei and Opossum within GH and found that all three plug-ins were effective in reducing the total simulation time, while retaining a pleasing accuracy of the optimal results. Opossum performed the best in terms of reducing the simulation time, while all tools achieved a high accuracy. When testing passive measures (1260 iterations) with the shoebox model, Opossums longest run took approximately 40 minutes, compared to 4 hours with Colibri, while obtaining a 100 % success rate in finding both the largest hypervolumes and the minimum objective values. After adding active measures (10 080 iterations), the longest run for Opossum still took a similar time of around 35 minutes, compared to Colibri taking 31 hours, with success rates of 99 % and 100 % respectively. When evaluating passive measures with a more complex geometry based on a real building, the total runtime with Colibri would have increased to approximately 5 days, while a slight simplification of this model resulted in a total runtime of 33 hours. In comparison, the optimisation run with Opossum reduced this time by 91 % as it took around 3 hours to execute, with a hypervolume success rate of 96 % and a minimum objective value success rate of 100 %. Although Opossum was the most effective in reducing the simulation time, choosing between optimisation tools could also depend on other factors such as the availability of certain features, i.e., if the inclusion of features concerning data analysis or the post processing of results within the GH tool is important to the user, Wallacei might prove to be a viable option.

For the real building model, the most optimal renovation packages generally included similar measures. When balancing all three objectives or considering only environmental impact, the same renovation package was deemed as optimal: adding 100 mm of glass wool insulation to the walls and 300 mm to the roof, alongside installing storm windows. Compared to the existing building, this resulted in a decrease in carbon impact of 13.8 $kgCO_2eq./m^2$ as well as saving 30.5 $kWh/(m^2 \cdot y)$ of energy and 1.8 $MSEK/m^2$ in costs, over the 30-year study period. On the other hand, aiming for a minimum EUI or a minimum NPV meant that the optimal package included the installation of triple-glazed windows rather than storm windows, while all other variables remained the same. The resulting energy savings for this package were 32.4 $kWh/(m^2 \cdot y)$, while 1.9 $MSEK/m^2$ were saved in costs, and the carbon impact decreased by 8.8 $kgCO_2eq./m^2$.

Overall, the optimisation tools proved to be successful in accelerating the simulation process, by focusing on the area of interest regarding the objectives, while choosing the right input settings can further increase the effectiveness of their application, i.e., adjusting the population size or termination criteria. Due to the often unfeasible simulation runtime of parametric design, the implementation of optimisation tools becomes an important factor to consider and could even be the difference in determining whether the execution of a parametric design is possible or not.

The evident influence that optimisation tools can have on the efficiency and feasibility of parametric design makes the field of optimisation a valuable area to explore further, while many considerations could be made for future studies in this topic. Although this study already focused on a multi-objective optimisation problem by assessing energy use, environmental impact and costs, there are numerous other factors that can be used to assess the performance of a building, such as thermal comfort, indoor air quality and measures of daylight availability. It would be beneficial to increase the number of objectives to include more of these aspects, while a continuation of this study could also include different or additional renovation measures. Although the outcomes of this study can be applicable to similar buildings, these findings would be affected by a change in geometry; thus, future work should also consider the application of optimisation tools to other building typologies. Fewer simplifications or exclusions of certain aspects, specifically for LCA and LCC, might also increase the validity of the renovation packages that were deemed as the most optimal. Lastly, the different

optimisation methods could also be explored further, by analysing the performance of the evolutionary algorithms within Opossum or investigating the possibility of other optimisation methods within GH.

References

Abo-Hamad, W., & Arisha, A. (2011). *Simulation Optimisation Methods in Supply Chain Applications: A Review*. https://www.researchgate.net/publication/215642410

Anagnostopoulos, G. G., Deriaz, M., & Konstantas, D. (2017). A Multiobjective Optimization Methodology of Tuning Indoor Positioning Systems. *2017 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2017*, *2017-January*. https://doi.org/10.1109/IPIN.2017.8115908

Arora, R. K. (2015). *Optimization Algorithms and Applications*. Taylor & Francis Group.

Audet, C., & Hare, W. (2017). *Derivative-Free and Blackbox Optimization* (T. V. Mikosch, S. I. Resnick, & S. M. Robinson, Eds.). Springer International Publishing AG. http://www.springer.com/series/3182

Bader, J., & Zitzler, E. (2008). *HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization*. http://www.tik.ee.ethz.ch/

Baeyens, E., Herreros, A., & Perán, J. R. (2016). A Direct Search Algorithm for Global Optimization. *Algorithms*, *9*(2). https://doi.org/10.3390/a9020040

Boverket. (2013). *Optimal costs for energy efficiency - a basis according to Directive 2010/31 / EU of the European Parliament and of the Council on the energy performance of buildings*.

Boverket. (2017). *Boverkets författningssamling - BFS 2017:6 BEN 2 - Boverkets föreskrifter om ändring av verkets föreskrifter och allmänna råd (2016:12) om fastställande av byggnadens energianvändning vid normalt brukande och ett normalår.*

Boverket. (2020a). *Konsoliderad version av Boverkets byggregler (2011:6) – föreskrifter och allmänna råd*.

Boverket. (2020b). *Regulation on climate declarations for buildings proposal for a roadmap and limit values*. https://www.boverket.se/en/start/publications/publications/2020/regulation-on-climate-declarations-for-buildings/

Chankong, V., & Haimes, Y. Y. (1983). *Multiobjective Decision Making: Theory and Methodolgy* (A. P. Sage, Ed.; Vol. 8). Elsevier Science Publishing Co., Inc.

Chong, E. K. P., & Zak, S. H. (2013). *An Introduction to Optimization* (4th ed.). John Wiley & Sons, Inc.

Cichocka, J. M., Migalska, A., Browne, W. N., & Rodriguez, E. (2017). SILVEREYE – The Implementation of Particle Swarm Optimization Algorithm in a Design Optimization Tool. *Communications in Computer and Information Science*, *724*, 151–169. https://doi.org/10.1007/978-981-10-5197-5_9

Clarke, J. A. (Joe A.). (2001). *Energy simulation in building design*. 362.

Coello Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems* (2nd ed.). Springer Science+Business Media.

Costa, A., & Nannicini, G. (2018). RBFOpt: an open-source library for black-box optimization with costly function evaluations. *Mathematical Programming Computation*, *10*(4), 597–629. https://doi.org/10.1007/s12532-018-0144-7

Cubukcuoglu, C., Ekici, B., Tasgetiren, M. F., & Sariyildiz, S. (2019). OPTIMUS: Self-adaptive Differential Evolution with Ensemble of Mutation Strategies for Grasshopper Algorithmic Modeling. *Algorithms*, *12*(7). https://doi.org/10.3390/a12070141

Custódio, A. L., Emmerich, M., & Madeira, J. F. A. (2012). Recent Developments in Derivative-Free Multiobjective Optimisation. *Computational Technology Reviews*, *5*, 1–30. https://doi.org/10.4203/ctr.5.1

Daya, B., & Nolan, H. (2022). *Energy renovation packages for decarbonisation of Swedish multi-family homes* [Lund University]. https://tt-acm.github.io/DesignExplorer/?ID=BL_3FJoZRK

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Ltd.

Deb, K. (2012). *Optimization for Engineering Design : Algorithms and Examples* (2nd ed.). PHI Learning Private Limited.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Deepa, O., & Senthilkumar, Dr. A. (2016). Swarm Intelligence from Natural to Artificial Systems: Ant Colony Optimization. *International Journal on Applications of Graph Theory In Wireless Ad Hoc Networks And Sensor Networks*, *8*(1), 9–17. https://doi.org/10.5121/jgraphoc.2016.8102

Dorigo, M., Maniezzo, V., & Colorni, A. (1999). *Positive Feedback as a Search Strategy*. https://www.researchgate.net/publication/2573263

Egüez, A. (2021). District heating network ownership and prices: The case of an unregulated natural monopoly. *Utilities Policy*, *72*, 101252. https://doi.org/10.1016/J.JUP.2021.101252

Emmerich, M., & Deutz, A. (2014). Time Complexity and Zeros of the Hypervolume Indicator Gradient Field. In *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III* (pp. 169–194). Springer. http://www.springer.com/series/7092

European Commission. (2019). *The European Green Deal - Communication From The Commission To The European Parliament, The European Council, The Council, The European Economic And Social Committee And The Committee Of The Regions*.

European Commission. (2020). *Renovation Wave: doubling the renovation rate to cut emissions, boost recovery and reduce energy poverty*. https://ec.europa.eu/commission/presscorner/detail/en/IP_20_1835

Floudas, C. A., & Pardalos, P. M. (2009). *Encyclopedia of Optimization* (2nd ed.). Springer Science+Business Media.

Friedrich, L., Nold, S., Müller, A., Rentsch, J., & Preu, R. (2021). Global Warming Potential and Energy Payback Time Analysis of Photovoltaic Electricity by Passivated Emitter and Rear Cell (PERC) Solar Modules. *IEEE Journal Of Photovoltaics, Submitted*.

Fufa, S. M., Schlanbusch, R. D., Sørnes, K., Inman, M., & Andresen, I. (2016). *A Norwegian ZEB Definition Guideline*.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.

Guerreiro, A. P., Fonseca, C. M., & Paquete, L. (2021). The Hypervolume Indicator: Computational Problems and Algorithms. *ACM Computing Surveys*, *54*(6). https://doi.org/10.1145/3453474

Gutmann, H.-M. (2001). A Radial Basis Function Method for Global Optimization. *Journal of Global Optimization*, *19*, 201–227.

Hillier, F. S., & Lieberman, G. J. (2010). *Introduction to Operations Research* (9th ed.). McGraw-Hill. www.mhhe.com/hillier

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (1st ed.). The University of Michigan Press.

Hooke, R., & Jeeves, T. A. (1961). "Direct Search" Solution of Numerical and Statistical Problems. *Journal of the ACM*, *8*(2), 212–229.

IEA. (2022). *National Survey Report of PV Power Applications in Sweden 2021*. www.iea-pvps.org

IPCC. (2007). *Climate Change 2007: Synthesis Report. Contribution of Working Groups I, II and III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*. https://www.ipcc.ch/report/ar4/syr/

IPCC. (2022). *Climate Change 2022: Mitigation of Climate Change*. www.ipcc.ch

Jones, D. R., & Martins, J. R. R. A. (2021). The DIRECT algorithm: 25 years Later. *Journal of Global Optimization*, *79*(3), 521–566. https://doi.org/10.1007/s10898-020-00952-6

Jones, D. R., Perrttunen, C. D., & Stuckman, B. E. (1993). Lipschitzian Optimization Without the Lipschitz Constant. *Journal Of Optimization Theory And Application*, *79*(1), 157–181.

Jones, T. C. (1995). *Evolutionary Algorithms, Fitness Landscapes and Search* [University of New Mexico]. https://www.researchgate.net/publication/23740259

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1942–1948.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, *220*(4598), 671–680.

Knowles, J., Corne, D., & Deb, K. (2008). *Multiobjective Problem Solving from Nature From Concepts to Applications*. Springer.

Kolda, T. G., Lewis, R. M., & Torczon, V. (2003). Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *Society for Industrial and Applied Mathematics*, *45*(3), 385–482. https://doi.org/10.1137/S0036144502428893

Koziel, S., Ciaurri, D. E., & Leifsson, L. (2011). Surrogate-Based Methods. In S. Koziel & X.-S. Yang (Eds.), *Computational Optimization, Methods and Algorithms* (pp. 33–59). Springer.

Koziel, S., & Yang, X.-S. (2011). Computational Optimization: An Overview. In S. Koziel & X.-S. Yang (Eds.), *Computational Optimization, Methods and Algorithms* (pp. 1–12). Springer.

Kvasov, D. E., & Mukhametzhanov, M. S. (2018). Metaheuristic vs. deterministic global optimization algorithms: The univariate case. *Applied Mathematics and Computation*, *318*, 245–259. https://doi.org/10.1016/J.AMC.2017.05.014

Lewis, R. M., Torczon, V., & Trosset, M. W. (2000). Direct search methods: then and now. *Journal of Computational and Applied Mathematics*, *124*(1–2), 191–207. https://doi.org/10.1016/S0377-0427(00)00423-4

Luke, S. (2016). *Essentials of Metaheuristics*.

Makki, M., Showkatbakhsh, M., & Song, Y. (2022). *Wallacei - An Evolutionary Multi-Objective Optimization and Analytic Engine for Grasshopper 3D*. https://www.wallacei.com/

McMullen, P. R. (2001). An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives. *Artificial Intelligence in Engineering*, *15*(3), 309–317. https://doi.org/10.1016/S0954-1810(01)00004-8

Merkle, D., & Middendorf, M. (2005). Swarm Intelligence. In E. K. Burke & G. Kendall (Eds.), *Search Methodologies Introductory Tutorials in Optimization and Decision Support Techniques* (pp. 401–436). Springer.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, *21*(6), 1087–1092. https://doi.org/10.1063/1.1699114

Ministry of the Environment Sweden. (2020). *Sweden's long-term strategy for reducing greenhouse gas emissions*.

Morissette, L., & Chartier, S. (2013). The k-means clustering technique: General considerations and implementation in Mathematica. *Tutorials in Quantitative Methods for Psychology*, *9*(1), 15–24. https://doi.org/10.20982/tqmp.09.1.p015

Müller, A., Friedrich, L., Reichel, C., Herceg, S., Mittag, M., & Neuhaus, D. H. (2021). A comparative life cycle assessment of silicon PV modules: Impact of module design, manufacturing location and inventory. *Solar Energy Materials and Solar Cells*, *230*, 111277. https://doi.org/10.1016/J.SOLMAT.2021.111277

Nelder, J. A., & Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal*, *7*(4), 308–313.

NIBE. (2023). *Värmepumpar för en enkel och hållbar vardag*. https://www.nibe.eu/sv-se

Nikolaev, A. G., & Jacobsen, S. H. (2010). Simulated Annealing. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics* (2nd ed., Vol. 146, pp. 1–40). Springer. http://www.springer.com/series/6161

One Click LCA. (2023). *LCA Database of Building products: local and global data for your LCA*. https://www.oneclicklca.com/support/faq-and-guidance/documentation/database/

Optimera. (2023). *Optimera Bygghandel för Proffs*. https://www.optimera.se/bygghandel/proffs

Pintér, J. (2002). Global Optimization: software, test problems and applications. In P. M. Pardalos & H. E. Romeijn (Eds.), *Handbook of Global Optimization* (Vol. 2, pp. 515–569). Springer . https://doi.org/10.1007/978-1-4757-5362-2

Ramírez-Villegas, R., Eriksson, O., & Olofsson, T. (2016). Assessment of renovation measures for a dwelling area – Impacts on energy efficiency and building certification. *Building and Environment*, *97*, 26–33. https://doi.org/10.1016/J.BUILDENV.2015.12.012

Ravindran, A., Reklaitis, G. V., & Ragsdell, K. M. (2006). *Engineering Optimization Methods and Applications* (2nd ed.). John Wiley & Sons .

Razna, R., & Aive, N. T. (2022). *Climate-neutral buildings-impact of existing definitions on building design*. Lund University.

Rechenraum GmbH. (2023). *goat*. https://www.rechenraum.com/en/goat.html

Reeves, C. R. (2010). Genetic Algorithms. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics* (2nd ed., Vol. 146, pp. 109–140). Springer. http://www.springer.com/series/6161

Rios, L. M., & Sahinidis, N. V. (2013). Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, *56*(3), 1247–1293. https://doi.org/10.1007/s10898-012-9951-y

Rostami, S., Neri, F., & Gyaurski, K. (2020). On Algorithmic Descriptions and Software Implementations for Multi-objective Optimisation: A Comparative Study. *SN Computer Science*, *1*(5). https://doi.org/10.1007/s42979-020-00265-1

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

SGBC. (2022). *NollCO2 Nettonoll Klimatpåverkan - Manual1.1*. www.sgbc.se

Si, B., Tian, Z., Jin, X., Zhou, X., Tang, P., & Shi, X. (2016). Performance indices and evaluation of algorithms in building energy efficient design optimization. *Energy*, *114*, 100–112. https://doi.org/10.1016/J.ENERGY.2016.07.114

Statista. (2023). *Projected average prices of electricity for final demand sectors in Sweden from 2020 to 2050.* https://www.statista.com/statistics/720639/projected-electricity-price-sweden/

SVEBY. (2012). *Brukarindata bostäder.*

Svensk Riksdag. (2021). *Sweden's climate policy framework.* https://www.government.se/articles/2021/03/swedens-climate-policy-framework/

Sveriges Riksbank. (2023). *Policy rate, deposit and lending rate.* https://www.riksbank.se/sv/statistik/sok-rantor--valutakurser/styrranta-in--och-utlaningsranta/

Talbi, E.-G. (2009). *Metaheuristics From Design to Implementation.* John Wiley & Sons, Inc.

Tan, K. C., Khor, E. F., & Lee, T. H. (2005). *Multiobjective Evolutionary Algorithms and Applications.* Springer-Verlag.

UNFCCC. (2015). *Adoption of the Paris Agreement.* https://reliefweb.int/report/world/adoption-paris-agreement-enar?gclid=CjwKCAiA3KefBhByEiwAi2LDHOQNsw11eue68I6ee9FYzaEWud0ndQv_tmNTyNbfl_lSbMIWXJAHDxoCZvwQAvD_BwE

United Nations Environment Programme. (2020). *Emissions Gap Report 2020.* https://www.unep.org/emissions-gap-report-2020

United Nations Environment Programme. (2021). *2021 Global Status Report for Buildings and Construction Towards a zero-emissions, efficient and resilient buildings and construction sector.* https://globalabc.org/resources/publications/2021-global-status-report-buildings-and-construction

Viable Cities. (2022). *FAQ Viable Cities and the Climate Neutral Cities 2030 2.0 initiative.* https://en.viablecities.se/faq-23-stader

Vierlinger, R. (2013). *Multi Objective Design Interface* [University of Applied Arts Vienna]. https://doi.org/10.13140/RG.2.1.3401.0324

Weise, T. (2009). *Global Optimization Algorithms Theory and Application* (2nd ed.). http://www.it-weise.de/

White Arkitekter. (2020). *Carbon Neutral Buildings - Creating value through architecture.* https://whitearkitekter.com/carbon-neutral-buildings-creating-value-through-architecture/

Wikells Sektionsfakta. (2022). *Wikells Sektionsfakta.* https://www.sektionsfakta.se/User/Byggdelar/ByggdelsKapitel?bok=NYB

Wortmann, T. (2017a). Model-based Optimization for Architectural Design: Optimizing Daylight and Glare in Grasshopper. *Technology Architecture and Design*, *1*(2), 176–185. https://doi.org/10.1080/24751448.2017.1354615

Wortmann, T. (2017b). Opossum: Introducing and Evaluating a Model-based Optimization Tool for Grasshopper. *Proceedings of the 22nd Conference on Computer Aided Architectural Design Research in Asia (CAADRIA)*, 283–292. https://doi.org/10.52842/conf.caadria.2017.283

Wortmann, T., Costa, A., Nannicini, G., & Schroepfer, T. (2015). Advantages of surrogate models for architectural design optimization. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *29*(4), 471–481. https://doi.org/10.1017/S0890060415000451

Wortmann, T., & Nannicini, G. (2016). Black-Box Optimisation Methods for Architectural Design: An overview and quantitative comparison of metaheuristic, direct search, and model-based optimization methods. *Proceedings of the 21st Conference on Computer Aided Architectural Design Research in Asia (CAADRIA)*, 177–186. https://doi.org/10.52842/conf.caadria.2016.177

Wright, S. (1932). The roles of mutation, inbreeding, cross-breeding and selection in evolution. *Proceedings of the 6th International Congress of Genetics*, 356–366.

Yang, D., Sun, Y., Di Stefano, D., Turrin, M., & Sariyildiz, S. (2016). Impacts of problem scale and sampling strategy on surrogate model accuracy: An application of surrogate-based optimization in building design. *2016 IEEE Congress on Evolutionary Computation, CEC 2016*, 4199–4207. https://doi.org/10.1109/CEC.2016.7744323

Yang, X.-S. (2011). Optimization Algorithms. In *Computational Optimization, Methods and Algorithms* (pp. 13–32). Springer.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). *SPEA2: Improving the Strength Pareto Evolutionary Algorithm.*

Zou, Y. (2010). *Classification of buildings with regard to airtightness.* Chalmers University of Technology.

Appendix A

Table A.1. Control method process and results

Assessment	Description	Excel result / (kWh/(m²·y))	Simulation result / (kWh/(m²·y))	Margin of error / %
Transmission losses	Shoebox model (5 m ·10 m · 3 m), elevated by 3 m, with no doors/windows and a U-value of 0.2 W/(m²·K) assigned to all surfaces. Measured under steady-state conditions at a constant outdoor temperature of 0°C, with values for infiltration, ventilation and internal gains set to zero and a heating set point of 20 °C.	133.00	132.94	0.05
Ventilation losses	Addition of mechanical ventilation of 0.35 l/(s·m²).	206.90	207.39	0.24
Infiltration losses	Addition of an infiltration rate of 0.3 l/(s·m²) at q50.	218.89	220.39	0.68
Heat recovery	Addition of sensible heat recovery of 75 %.	163.70	164.56	0.52
Envelope losses	Base case with a changed wall U-value of 0.1 W/(m²·K).	101.72	103.61	1.82
Window losses	Base case with a window with an area of 3.75 m² and a U-value of 1.3 W/(m²·K).	147.77	146.28	1.01
Ground losses	Base case with no elevation and a ground temperature of 0°C.	133.00	135.28	1.69
Ground temperature	Changing the ground temperature to 10°C.	-	118.89	-
Thermal bridges	Base case with a changed roof U-value of 0.22 W/(m²·K).	137.00	138.39	1.00
Real outdoor conditions	Base case with a change to a Copenhagen weather file.	-	59.72	-

Appendix B

Table B.1. Insulation material properties used in energy simulations

Material	Thickness / m	Conductivity / (W/(m·K))	Specific heat / (J/(kg·K))	Bulk density / (kg/m³)
EPS	0.10 – 0.35	0.033	1692	16.0
Glass wool	0.10 – 0.35	0.035	1030	18.7
Cellulose fibre	0.10 – 0.35	0.039	2020	47.0
Hemp fibre	0.10 – 0.35	0.040	1600	36.0
Wood fibre	0.10 – 0.35	0.038	2100	50.0

Table B.2. Breakdown of window costs, in SEK
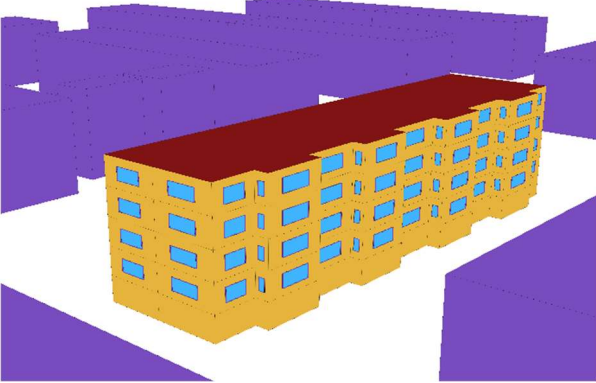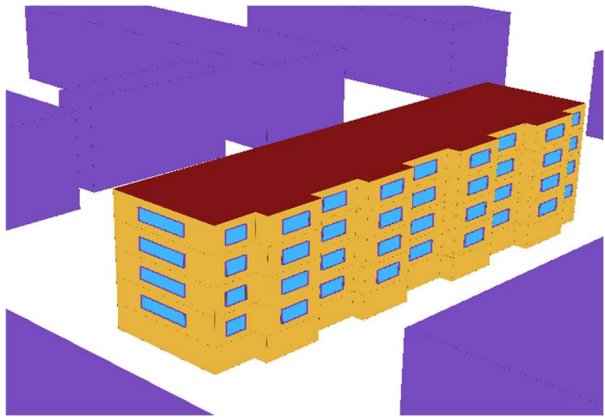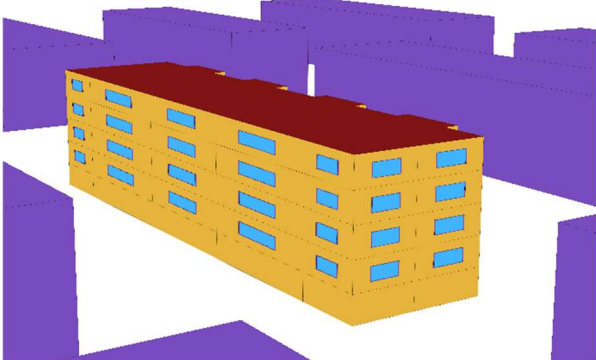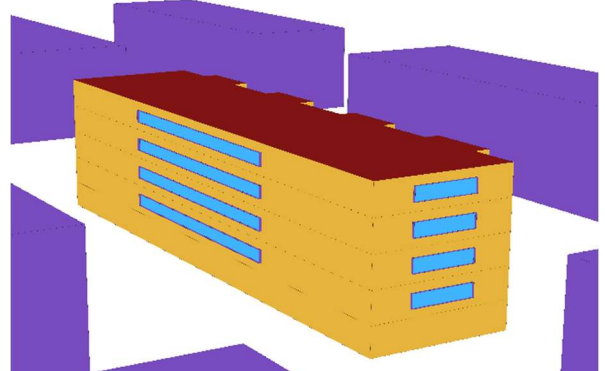
	Shoebox model	Real building model
Storm window		
Additional glazing layer	2741	171 815
Channel for spacer	1166	127 727
Acrylic sealant	878	96 141
Wooden trim painted white	698	76 410
Total cost	5484	472 092
Tripe glazed window		
Window type 1	-	174 424
Window type 2	-	495 192
Window type 3	15 058	794 772
Demolition of existing window(s)	1036	98 731
Removal of demolition material	1163	72 907
Total cost	17 257	1 636 025

Table B.3. Breakdown of EAHP costs, in present values

Description	Cost / SEK
EAHP initial cost incl. installation	153 078
Accumulator tanks initial cost incl. installation	9664
Insulated pipework incl. installation	1040
EAHP replacement cost	84 756
EAHP removal cost	956
EAHP disposal cost	118
Total cost	249 613

Appendix C

Table C.1. Simplification of the real building energy model

Initial model retrieved from database	Simplified model
Simulation time per iteration: 6 minutes	Simulation time per iteration: 1.4 minutes
Total simulation runtime for 1260 iterations ≈ 126 hours Total simulation runtime for 10080 iterations ≈ 42 days	Total simulation runtime for 1260 iterations = 35 hours Total simulation runtime for 10080 iterations ≈ 10 days

Appendix D

Table D.1. Total simulation performance results for passive and active measures, with shoebox model

	Plug-in	Population size input	Number of generations	Time	Time reduction	Total number of simulations	Success rate for hyper-volume	Success rate for minimum values
Passive & active measures	Colibri	-	-	30:49:05	-	10080	100%	100%
	Octopus	30	10	1:00:14	97 %	330	98%	98%
			10	0:59:55	97 %	330	98%	99%
			10	0:59:47	97 %	330	99%	100%
	Wallacei	30	10	0:54:35	97 %	300	99%	100%
			10	0:52:44	97 %	300	96%	99%
			10	0:53:24	97 %	300	100%	100%
	Results without improvement input							
	Opossum	30	-	0:23:28	99 %	112	98%	100%
			-	0:34:31	98 %	180	99%	100%
			-	0:08:59	100 %	48	93%	96%

Table D.2. Top 10 optimal renovation packages for the shoebox model with passive measures using Colibri, where CF = Cellulose Fibre, GW = Glass Wool, EW = Existing Window and ST = Storm Window

Façade insulation type	Façade insulation thickness /mm	Roof insulation type	Roof insulation thickness /mm	Window type	EUI / (kWh/m²)	GWP / (kgCO$_2$ eq./m²)	NPV / (KSEK/m²)	Hyper volume
CF	150	CF	200	EW	164.70	84.61	293.32	1966624.65
GW	100	GW	200	EW	163.15	87.29	291.41	1948809.46
CF	200	CF	200	EW	164.70	85.40	293.61	1945480.25
CF	150	CF	200	SW	159.09	83.46	307.87	1932613.88
CF	250	CF	200	EW	164.70	86.19	293.99	1923601.83
GW	350	GW	100	EW	161.03	90.08	290.78	1922889.47
CF	300	CF	200	EW	164.70	86.05	294.49	1921760.09
GW	100	GW	200	SW	157.53	86.15	305.97	1916235.42
CF	200	CF	200	SW	159.09	84.25	308.17	1911837.22
CF	300	CF	200	EW	164.70	86.98	294.49	1900614.02

Table D.3. Top 10 optimal renovation packages for the shoebox model with passive measures using Opossum, where CF = Cellulose Fibre, GW = Glass Wool, EW = Existing Window and ST = Storm Window

Façade insulation type	Façade insulation thickness /mm	Roof insulation type	Roof insulation thickness /mm	Window type	EUI / (kWh/m²)	GWP / (kgCO$_2$ eq./m²)	NPV / (KSEK/m²)	Hyper volume
CF	150	CF	200	EW	164.70	84.61	293.32	1966624.65
GW	100	GW	200	EW	163.15	87.29	291.41	1948809.46
CF	200	CF	200	EW	164.70	85.40	293.61	1945480.25
CF	150	CF	200	SW	159.09	83.46	307.87	1932613.88
CF	250	CF	200	EW	164.70	86.19	293.99	1923601.83
GW	100	GW	200	SW	157.53	86.15	305.97	1916235.42
CF	200	CF	200	SW	159.09	84.25	308.17	1911837.22
CF	300	CF	200	EW	164.70	86.98	294.49	1900614.02
CF	250	CF	200	SW	159.09	85.04	308.54	1890280.24
CF	350	CF	200	EW	164.70	87.77	294.68	1880645.52

Table D.4. Top 10 optimal renovation packages for the shoebox model with passive measures using Octopus, where CF = Cellulose Fibre, GW = Glass Wool, EW = Existing Window and ST = Storm Window

Façade insulation type	Façade insulation thickness /mm	Roof insulation type	Roof insulation thickness /mm	Window type	EUI / (kWh/m²)	GWP / (kgCO₂ eq./m²)	NPV / (KSEK/m²)	Hyper volume
CF	150	CF	200	EW	164.70	84.61	293.32	1966624.65
CF	200	CF	200	EW	164.70	85.40	293.61	1945480.25
CF	150	CF	200	SW	159.09	83.46	307.87	1932613.88
CF	250	CF	200	EW	164.70	86.19	293.99	1923601.83
CF	200	CF	200	SW	159.09	84.25	308.17	1911837.22
GW	350	GW	100	SW	155.42	88.93	305.33	1890645.92
CF	250	CF	300	SW	159.09	85.04	308.54	1890280.24
CF	350	CF	300	EW	164.70	87.77	294.68	1880645.52
GW	350	GW	200	EW	161.03	91.96	290.78	1877759.24
CF	300	CF	300	SW	159.09	85.83	309.04	1867543.87

Table D.5. Top 10 optimal renovation packages for the shoebox model with passive measures using Wallacei, where CF = Cellulose Fibre, GW = Glass Wool, EW = Existing Window and ST = Storm Window

Façade insulation type	Façade insulation thickness /mm	Roof insulation type	Roof insulation thickness /mm	Window type	EUI / (kWh/m²)	GWP / (kgCO₂ eq./m²)	NPV / (KSEK/m²)	Hyper volume
GW	350	GW	100	EW	161.03	90.08	290.78	1922889.47
CF	350	GW	300	EW	162.76	89.50	292.52	1892538.80
GW	350	GW	100	SW	155.42	88.93	305.33	1890645.92
CF	300	CF	200	SW	159.09	84.91	309.04	1888039.79
CF	350	CF	300	EW	164.70	87.77	294.68	1880645.52
GW	350	GW	200	EW	161.03	91.96	290.78	1877759.24
GW	300	GW	100	EW	163.15	89.75	292.88	1876871.77
CF	300	GW	300	EW	163.92	89.39	293.62	1866107.12
CF	350	GW	300	SW	157.15	88.35	307.07	1860489.38
CF	350	CF	300	SW	159.09	86.62	309.23	1848013.58

Table D.6. Solutions with minimum value of each objective for the shoebox model with passive measures using Colibri and optimisation tools, where CF = Cellulose Fibre, GW = Glass Wool, EW = Existing Window, ST = Storm Window and TW = Triple glazed Window

	Façade insulation type	Façade insulation thickness /mm	Roof insulation type	Roof insulation thickness /mm	Window type	EUI / (kWh /m²)	GWP / (kgCO₂ eq./m²)	NPV / (KSEK/ m²)	Hyper volume
Colibri									
Min EUI	GW	350	GW	100	TW	155.31	95.64	356.08	1255658
Min GWP	CF	150	CF	300	SW	159.09	83.46	307.87	1932613
Min NPV	GW	350	GW	100	EW	161.03	90.08	290.78	1922889
Opossum									
Min EUI	CF	350	EPS	300	TW	156.65	106.1	357.40	1057861
Min GWP	CF	150	CF	300	SW	159.09	83.46	307.87	1932614
Min NPV	GW	100	GW	300	EW	163.15	87.29	291.41	1948809
Octopus									
Min EUI	GW	350	GW	100	SW	155.42	88.93	305.33	1890645
Min GWP	CF	150	CF	300	SW	159.09	83.46	307.87	1932614
Min NPV	GW	350	GW	200	EW	161.03	91.96	290.78	1877759
Wallacei									
Min EUI	GW	350	GW	100	SW	155.42	88.93	305.33	1890645
Min GWP	CF	300	CF	200	SW	159.09	84.91	309.04	1888040

							EUI	GWP	NPV	
Min NPV	GW	350	GW	100	EW	161.03	90.08	290.78	1922889	

Table D.7. Top 10 optimal renovation packages for the shoebox model with passive and active measures using Colibri, where CF = Cellulose Fibre, GW = Glass Wool, ST = Storm Window and TW = Triple glazed Window

Façade ins. type	Façade ins. thickness /mm	Roof ins. type	Roof ins. thickness /mm	Win. type	PV size / % of roof area	Heat pump / unit	EUI / (kWh /m²)	GWP / (kgCO$_2$ eq./m²)	NPV / (KSEK /m²)	Hyper volume
GW	350	GW	100	SW	80	1	130.4	-811.5	-13.9	52757047
GW	350	GW	200	SW	80	1	130.4	-809.6	-13.9	52656220
GW	350	GW	300	SW	80	1	130.4	-807.7	-13.9	52555423
GW	350	GW	100	TW	80	1	130.3	-804.6	-13.8	52410397
GW	350	GW	200	TW	80	1	130.3	-802.8	-13.7	52309533
GW	350	GW	300	TW	80	1	130.3	-800.9	-13.7	52208698
GW	100	GW	300	SW	80	1	132.5	-814.3	-13.7	52138456
CF	350	GW	300	SW	80	1	132.1	-812.1	-12.5	51982009
GW	350	CF	300	SW	80	1	132.1	-809.6	-12.3	51825160
GW	100	GW	300	TW	80	1	132.4	-807.4	-13.5	51797413

Table D.8. Top 10 optimal renovation packages for the shoebox model with passive and active measures using Opossum, where CF = Cellulose Fibre, GW = Glass Wool, WF = Wood Fibre, HF = Hemp Fibre ST = Storm Window and TW = Triple glazed Window

Façade ins. type	Façade ins. thickness /mm	Roof ins. type	Roof ins. thickness /mm	Win. type	PV size / % of roof area	Heat pump / unit	EUI / (kWh /m²)	GWP / (kgCO$_2$ eq./m²)	NPV / (KSEK /m²)	Hyper volume
GW	350	GW	300	SW	80	1	130.4	-807.7	-13.9	52555423
WF	350	CF	300	SW	80	1	132.7	-809.4	-11.8	51549868
GW	300	CF	300	SW	80	1	133.3	-810.5	-11.5	51346853
GW	300	GW	300	TW	80	1	132.4	-801.2	-12.1	51253889
CF	350	EPS	300	TW	80	1	131.6	-794.2	-12.7	51242232
WF	350	CF	300	TW	80	1	132.6	-802.5	-11.6	51183010
GW	300	CF	300	TW	80	1	133.2	-803.6	-11.3	51009696
CF	250	CF	300	TW	80	1	134.0	-808.5	-11.3	50987084
HF	350	CF	300	SW	80	1	133.1	-802.1	-11.0	50914579
GW	300	EPS	300	TW	80	1	132.0	-790.2	-12.4	50856082

Table D.9. Top 10 optimal renovation packages for the shoebox model with passive and active measures using Octopus, where CF = Cellulose Fibre, GW = Glass Wool, WF = Wood Fibre, ST = Storm Window and TW = Triple glazed Window

Façade ins. type	Façade ins. thickness /mm	Roof ins. type	Roof ins. thickness /mm	Win. type	PV size / % of roof area	Heat pump / unit	EUI / (kWh /m²)	GWP / (kgCO$_2$ eq./m²)	NPV / (KSEK /m²)	Hyper volume
GW	100	GW	300	TW	80	1	132.4	-807.4	-13.5	51797413
WF	350	GW	300	SW	80	1	131.9	-807.0	-12.5	51796360
GW	300	GW	300	SW	80	1	132.5	-808.0	-12.2	51593709
HF	300	GW	300	SW	80	1	133.3	-812.2	-11.6	51455972
WF	350	GW	300	TW	80	1	131.9	-800.1	-12.3	51426810
GW	350	EPS	300	TW	80	1	130.9	-789.3	-13.2	51296971
WF	300	GW	300	SW	80	1	133.1	-807.8	-11.6	51280708
GW	300	GW	300	TW	80	1	132.4	-801.2	-12.1	51253889
GW	300	EPS	300	SW	80	1	132.1	-797.0	-12.5	51197427
HF	300	GW	300	TW	80	1	133.2	-805.3	-11.5	51118730

Table D.10. Top 10 optimal renovation packages for the shoebox model with passive and active measures using Wallacei, where CF = Cellulose Fibre, GW = Glass Wool, ST = Storm Window and TW = Triple glazed Window

Façade ins. type	Façade ins. thickness /mm	Roof ins. type	Roof ins. thickness /mm	Win. type	PV size / % of roof area	Heat pump / unit	EUI / (kWh /m²)	GWP / (kgCO₂ eq./m²)	NPV / (KSEK /m²)	Hyper volume
GW	350	GW	200	SW	80	1	130.4	-809.6	-13.9	52656220
GW	350	GW	300	SW	80	1	130.4	-807.7	-13.9	52555423
GW	350	GW	300	TW	80	1	130.3	-800.9	-13.7	52208698
CF	350	GW	300	SW	80	1	132.1	-812.1	-12.5	51982009
GW	350	CF	300	SW	80	1	132.1	-809.6	-12.3	51825160
GW	300	GW	300	SW	80	1	132.5	-808.0	-12.2	51593709
GW	350	CF	300	TW	80	1	132.1	-802.7	-12.1	51456387
CF	300	GW	300	SW	80	1	133.3	-812.2	-11.6	51455972
GW	300	CF	300	SW	80	1	133.3	-810.5	-11.5	51346853
GW	300	GW	300	TW	80	1	132.4	-801.2	-12.1	51253889

Table D.11. Solutions with minimum value of each objective for the shoebox model with passive and active measures using Colibri and optimisation tools, where CF = Cellulose Fibre, GW = Glass Wool, EW = Existing Window, ST = Storm Window, TW = Triple glazed Window

	Façade ins. type	Façade ins. thickness /mm	Roof ins. type	Roof ins. thickness /mm	Win. type	PV size / % of roof area	Heat pump / unit	EUI / (kWh/ m²)	GWP / (kgCO₂ eq./m²)	NPV / (KSEK /m²)	Hyper volume
Colibri											
Min EUI	GW	350	GW	100	TW	80	1	130.3	-804.6	-13.8	52410397
Min GWP	CF	150	CF	300	SW	80	0	159.1	-820.3	5.8	40777639
Min NPV	GW	350	GW	100	SW	80	1	130.4	-811.5	-13.9	52757047
Opossum											
Min EUI	GW	350	GW	300	SW	80	1	130.4	-807.7	-13.9	52555423
Min GWP	GW	100	GW	300	EW	80	1	138.1	-813.1	-8.6	49412299
Min NPV	GW	350	GW	300	SW	80	1	130.4	-807.7	-13.9	52555423
Octopus											
Min EUI	GW	350	EPS	300	TW	80	1	130.9	-789.3	-13.2	51296971
Min GWP	GW	100	GW	300	EW	80	0	163.1	-816.5	9.2	38878895
Min NPV	GW	100	GW	300	TW	80	1	132.4	-807.4	-13.5	51797413
Wallacei											
Min EUI	GW	350	GW	300	TW	80	1	130.3	-800.9	-13.7	52208698
Min GWP	CF	300	CF	200	SW	80	0	159.1	-818.9	7.0	40575049
Min NPV	GW	350	GW	200	SW	80	1	130.4	-809.6	-13.9	52656220

Table D.12. Top 10 optimal renovation packages for the real building model with passive measures using Colibri, where CF = Cellulose Fibre, GW = Glass Wool, ST = Storm Window and TW = Triple glazed Window

Façade insulation type	Façade insulation thickness /mm	Roof insulation type	Roof insulation thickness /mm	Window type	EUI / (kWh/m²)	GWP / (kgCO₂ eq./m²)	NPV / (MSEK/m²)	Hyper volume
GW	100	GW	300	SW	111.2	52.2	11.5	1178935
CF	200	CF	300	SW	112.3	52.3	11.6	1092942
CF	250	CF	300	SW	112.6	52.7	11.6	1047254
CF	300	CF	200	SW	113.0	53.0	11.6	1012642
GW	100	GW	200	SW	113.1	53.0	11.6	1004632
CF	300	CF	300	SW	113.0	53.1	11.6	1002671
GW	350	GW	100	SW	112.2	54.1	11.6	993993
GW	300	GW	100	SW	112.5	53.9	11.6	990747
CF	300	GW	300	SW	112.8	53.6	11.6	988385

| GW | 100 | GW | 300 | TW | 109.3 | 57.2 | 11.4 | 987292 |

Table D.13. Top 10 optimal renovation packages for the real building model with passive measures using Opossum, where CF = Cellulose Fibre, GW = Glass Wool, ST = Storm Window and TW = Triple glazed Window

Façade insulation type	Façade insulation thickness /mm	Roof insulation type	Roof insulation thickness /mm	Window type	EUI / (kWh/m²)	GWP / (kgCO$_2$ eq./m²)	NPV / (MSEK/m²)	Hyper volume
CF	200	CF	300	SW	112.3	52.3	11.6	1092942
CF	250	CF	300	SW	112.6	52.7	11.6	1047254
CF	300	CF	300	SW	113.0	53.1	11.6	1002671
CF	300	GW	300	SW	112.8	53.6	11.6	988385
CF	350	GW	300	SW	112.7	53.8	11.6	983871
CF	250	GW	300	SW	113.0	53.5	11.6	980472
CF	350	CF	300	SW	113.3	53.5	11.6	959340
GW	200	CF	300	SW	113.2	53.7	11.6	959023
CF	200	GW	300	SW	113.4	53.6	11.6	953298
GW	150	GW	300	TW	109.6	57.8	11.4	931456

Table D.14. Top 10 optimal renovation packages for the real building model with passive measures using Octopus, where CF = Cellulose Fibre, GW = Glass Wool, ST = Storm Window

Façade insulation type	Façade insulation thickness /mm	Roof insulation type	Roof insulation thickness /mm	Window type	EUI / (kWh/m²)	GWP / (kgCO$_2$ eq./m²)	NPV / (MSEK/m²)	Hyper volume
GW	350	GW	100	SW	112.2	54.1	11.6	993993
GW	300	GW	100	SW	112.5	53.9	11.6	990747
CF	350	GW	300	SW	112.7	53.8	11.6	983871
CF	250	GW	300	SW	113.0	53.5	11.6	980472
GW	350	GW	200	SW	112.2	54.5	11.6	972772
GW	250	GW	100	SW	113.0	53.8	11.6	968096
GW	200	CF	300	SW	113.2	53.7	11.6	959023
GW	350	GW	300	SW	112.2	54.9	11.6	951550
GW	150	GW	200	SW	113.4	53.6	11.6	950497
GW	300	GW	300	SW	112.5	54.6	11.6	949033

Table D.15. Top 10 optimal renovation packages for the real building model with passive measures using Wallacei, where CF = Cellulose Fibre, GW = Glass Wool, ST = Storm Window and TW = Triple glazed Window

Façade insulation type	Façade insulation thickness /mm	Roof insulation type	Roof insulation thickness /mm	Window type	EUI / (kWh/m²)	GWP / (kgCO$_2$ eq./m²)	NPV / (MSEK/m²)	Hyper volume
GW	100	GW	300	SW	111.2	52.2	11.5	1178935
GW	150	GW	300	SW	111.5	52.8	11.5	1119039
GW	100	GW	200	SW	113.1	53.0	11.6	1004632
GW	100	GW	300	TW	109.3	57.2	11.4	987292
CF	250	GW	300	SW	113.0	53.5	11.6	980472
GW	350	GW	200	SW	112.2	54.5	11.6	972772
GW	350	GW	300	SW	112.2	54.9	11.6	951550
GW	150	GW	200	SW	113.4	53.6	11.6	950497
GW	300	GW	300	SW	112.5	54.6	11.6	949033
CF	300	GW	200	SW	113.5	53.7	11.6	939216

Table D.16. Solutions with minimum value of each objective for the real building model with passive measures using Colibri and optimization tools, where CF = Cellulose Fibre, GW = Glass Wool, EW = Existing Window, ST = Storm Window and TW = Triple glazed Window

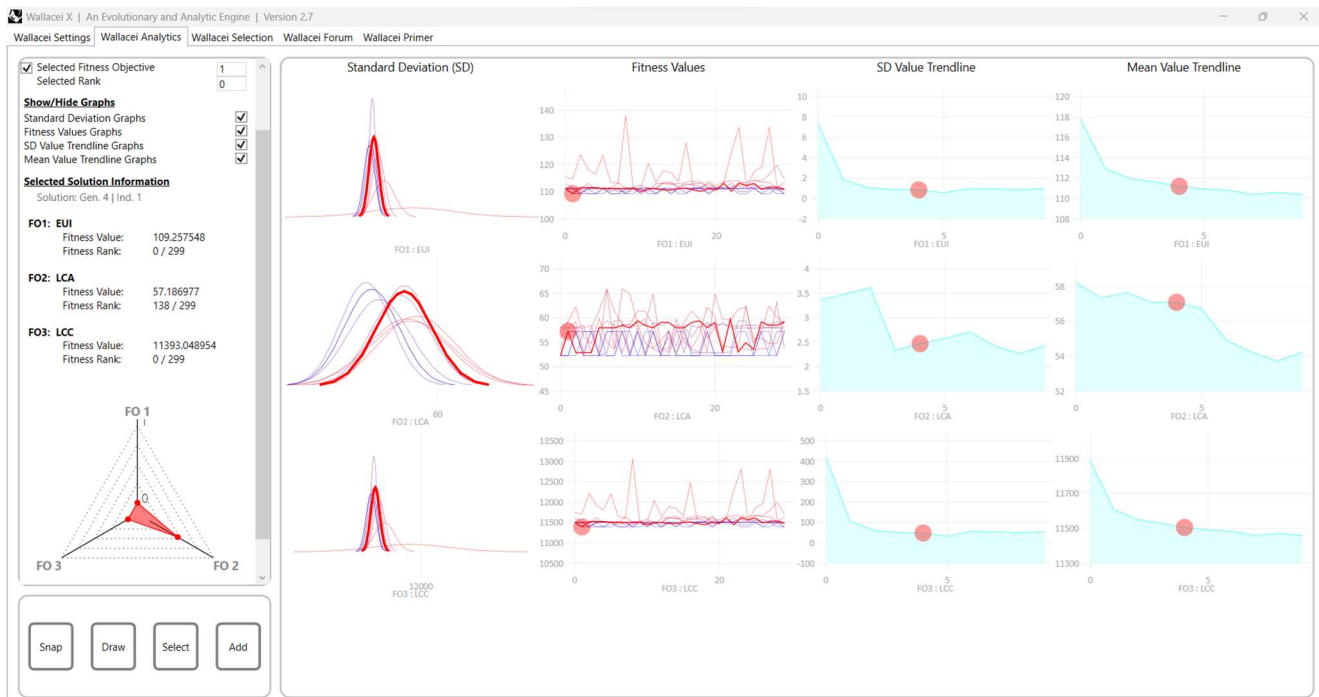|	Façade insulation type	Façade insulation thickness /mm	Roof insulation type	Roof insulation thickness /mm	Window type	EUI / (kWh /m²)	GWP / (kgCO₂ eq./m²)	NPV / (MSEK /m²)	Hyper volume
Colibri									
Min EUI	GW	100	GW	300	TW	109.3	57.2	11.4	987292
Min GWP	GW	100	GW	300	SW	111.2	52.2	11.5	1178935
Min NPV	GW	100	GW	300	TW	109.3	57.2	11.4	987292
Opossum									
Min EUI	GW	150	GW	300	TW	109.6	57.8	11.4	931456
Min GWP	CF	200	CF	300	SW	112.3	52.3	11.6	1092942
Min NPV	GW	150	GW	300	TW	109.6	57.8	11.4	931456
Octopus									
Min EUI	GW	350	GW	100	TW	110.1	59.0	11.4	822186
Min GWP	CF	250	GW	300	SW	113.0	53.5	11.6	980472
Min NPV	GW	350	GW	100	TW	110.1	59.0	11.4	822186
Wallacei									
Min EUI	GW	100	GW	300	TW	109.3	57.2	11.4	987292
Min GWP	GW	100	GW	300	SW	111.2	52.2	11.5	1178935
Min NPV	GW	100	GW	300	TW	109.3	57.2	11.4	987292

Figure D.1. Wallacei analysis features

Figure D.2. Wallacei recalling and exporting features