TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

DISSERTATION

A COMPUTATIONAL ENVIRONMENT FOR BUILDING ENCLOSURE SYSTEMS DESIGN AND CONTROL SUPPORT VIA DYNAMIC SIMULATION-ASSISTED OPTIMIZATION

ausgeführt zum Zwecke der Erlangung des akademischen Grades

eines Doktors der technischen Wissenschaften

unter der Leitung von

Univ. -Prof. Dipl. -Ing. Dr. techn. Ardeshir Mahdavi

E 259-3 Abteilung für Bauphysik und Bauökologie

Institute für Architekturwissenschaften

eingereicht an der

Technischen Universität Wien

Fakultät für Architektur und Raumplanung

von

Hamidreza Shirdel

Matr. Nr.: 1129188

Guglgasse 8, 3 3 9B, 1110-Wien

Wien, im September 2018

Kurzfassung

In den letzten Jahrzehnten machte eine Reihe von Entwicklungen die globale Optimierung großer multidimensionaler Designoptionsräume möglich. Zu diesen Entwicklungen gehören das Wachstum von Rechenleistungen, die Entstehung hochentwickelter Optimierungsalgorithmen und neue Techniken zur Ableitung von rechnerisch hocheffizienten Meta-Modellen. Neben ihren Versprechen beinhalten diese Entwicklungen auch einige potenzielle Nachteile. Beispielsweise können Metamodelle gelegentlich das Verhalten "nicht-konventioneller" und komplexer Designs nicht erfassen. Ein weiteres kritisches Problem betrifft die potentiell undurchsichtige Natur großer, globaler Optimierungsübungen, die sie für die Bereitstellung intuitiver Unterstützung in einem natürlichen iterativen Entwurfsprozess weniger zugänglich machen. In diesem Zusammenhang untersucht diese Arbeit das Potenzial eines neuartigen Ansatzes zur iterativen globalen Optimierung lokal optimierter Attribut-Cluster von Gebäudeentwurfslösungen. Dadurch werden Cluster von Entwurfsraumattributen (d.h. Sets ontologisch verwandter Aspekte von Entwürfen), die für typische Gebäudegestalter als zusammengesetzter, jedoch kohärenter Aspekt eines Entwurfs verständlich sind, mehreren Durchgängen von lokalen simulationsgestützten Optimierungsläufen unterzogen. Anstatt jeder Variablen eines komplexen Designs im Rahmen einer globalen Single-Pass-Optimierungskampagne eine einzelne Dimension zuzuordnen, zielen mehrere iterative Optimierungsschritte auf kohärente Cluster dieser Attribute ab und verfolgen diese, bis das Gesamtdesign der erwarteten Leistung entspricht (oder bis eine

weitere Leistungsverbesserung bevorsteht). Die Dissertation berichtet über mehrere Tests dieses Ansatzes und dokumentiert die Vorteile der Methode (d.h. die Verwendung von Original-Simulationsmodellen statt Meta-Modellen sowie die iterative, transparente und intuitive Navigation des Entwurfsraums).

Die Leistung der Implementierungen des vorgeschlagenen Ansatzes über Optimierungsfallstudien, welche verschiedene Systembetriebsoptionen beinhalteten, wird veranschaulicht (z.B. zufälliger Zyklus zwischen Attribut-Clustern gegenüber vordefinierten Sequenzen sowie unterschiedliche Komplexitäten der Gebäude).

Die vorgeschlagene Methode liefert optimierte Lösungen, die sich von denen eines globalen, eimaligen Referenz-Optimierungslaufs praktisch nicht unterscheiden, soweit die Werte der Indikatoren der Energieeffizienz und die verbundene Kostenfunktion betroffen sind. Bei diesem Ansatz werden die Ergebnisse nicht nur schneller, effizienter und genauer erzielt, sie finden auch in einem transparenten, nachweisbaren und designerfreundlichen Verfahren statt.

Summary

In the last decades, a number of developments have made global optimization of large multi-dimensional design option spaces possible. Such developments include the increase in computing power, emergence of sophisticated optimization algorithms, and new techniques for the derivation of computationally highly efficient meta-models. Along with their promise, such developments also involve a number of potential drawbacks. For one thing, meta-models occasionally fail to capture the behaviour of "non-conventional" and complex designs. Another critical problem pertains to the potentially opaque nature of large-scale global optimization exercises, which make them less amenable to provision of intuitively graspable support in a naturally iterative design process. In this context, this research explores the potential of a novel approach toward iterative global optimization of locally optimized attribute clusters of building design solutions. Thereby, clusters of design space attributes (i.e., sets of ontologically cognate aspects of designs) that are comprehensible to typical building designers as a compound yet coherent aspects of a design are made subject to multiple passes of local simulation-assisted optimization runs. Hence, instead of allocating an individual dimension to each and every variable of a complex design within the context of a single-pass global optimization campaign, multiple iterative optimization steps target coherent clusters of such attributes and pursue those until the overall design meets the expected performance (or until further performance improvement is not forthcoming). The dissertation reports on several tests of this approach, documenting the method’s advantages (i.e., use of

original simulation models instead of meta-models as well as iterative, transparent, and intuitive navigation of the design space). The performance of the implementations of the proposed approach via optimization case studies, which contained different system operation options are illustrated (e.g., random cycling between attribute clusters versus predefined sequences as well as different complexities of the buildings).

The proposed method delivers optimized solutions that are – as far as the values of the energy performance indicators and the associate cost function are concerned – virtually indistinguishable from those of a reference one-shot global optimization run. However, in this approach the results are not only obtained faster, more efficiently and more accurately, but also via a transparent, traceable, and designer-friendly process.

Acknowledgments

First of all, I would like to thank Prof. Ardeshir Mahdavi for his support, guidance, and advising me throughout my thesis writing process. He invested a lot of time in guiding me, giving me new suggestions and reviewing my work. He opened new windows to both the profession and life for me. I will be grateful for his kindness and patience till the last moment of my life.

Furthermore, I am grateful to Ms. Elisabeth Finz and the members of the Department of Building Physics and Building Ecology for their support.

Finally, I want to send my gratitude to my lovely family and friends for the endless love.

Parts of the text in this dissertation are adopted from the papers, written in relation to the research study, and co-authored with Prof. Ardeshir Mahdavi.

For my mother and the soul of my father

Contents

Chapter 1

1. Introduction

1.1. Motivation

The three world primary economic sectors of energy use are industry, transportation, and buildings (Al-Homoud, 2001). Buildings have a substantial share of the energy consumption all over the world, which deserves to be looked at carefully for the efficient operation of such facilities. Many possible approaches are expected to appear in order to meet future energy developments related to each of these sectors of energy use. The importance of heating and cooling in total building energy use is very diverse, with this share varying between 18% and 73% (Ürge-Vorsatz, et al., 2015). Buildings are the sector with the greatest potential and lowest cost for carbon reductions.

Building enclosure plays an essential role with regard to the overall performance of the buildings. Specifically, energy performance, thermal comfort, and lighting conditions are significantly affected by the quality of enclosure design and its operational status.

However, the design of sustainable buildings is not straight-forward. Most buildings are unique, and there are fewer prototypes. Designs must achieve high levels of performance for the lowest possible cost. Many physical processes lead to conflicting objectives. The design space of possible solutions is very large. These challenges have made it advantageous to apply computational methods of design optimization.

1.2. Objective

Recent progress in computer science and the stringent requirements of the design of “greener” buildings has accelerated the research and applications of simulation-based optimization methods in the building sector. Energy simulation methods allow designers to predict the energy demand of the buildings in different circumstances. This involves thermal, solar, airflow modeling and concerns the geometry, materials, control, and systems of the building. However, the large number of the design variables and conflicts which arise the solutions make optimization methods very complicated, especially for the architects.

The primary objective of this study was the implementation of a system, which supports architects to design energy efficient buildings in the early stages of the design. Furthermore, since the prevalent optimization methods are like a black box, mostly in the event that the number of variables is high, the user does not have the chance to understand the advantages and disadvantages of changing each parameter of the design for the final solution. Moreover, handling a large number of variables according to the normal methods could be very time consuming and the chance of convergence is reduced (depends on the optimization algorithms which have been used). Therefore, this research study aims to achieve a simulation-based optimization method which is fast, transparent, traceable, accurate and more designer-friendly.

1.3. Background

The applications of numerical optimization have been considered since the 1980s and 1990s based on great advances of computational science and mathematical optimization methods.

However, most studies in building engineering which combined a building energy simulation program with an algorithmic optimization 'engine' have been published in the late 2000s, although the first efforts were found much earlier (Nguyen, et al., 2014). Moreover, managing a large number of independent variables in the optimization process still represents a challenge (Wetter, 2016).

The primary idea of this project was inspired by another PhD project, which was presented at December 2002 by Prechaya Mahattanatawe at the School of Architecture, Carnegie Mellon University, Pittsburgh PA under the supervision of professor Ardeshir Mahdavi. It presents a computational environment for performance-based integrated building enclosure design and control support. The key concepts and features of this environment include virtual enclosure, construction mapping, and shading device recommendation. Optimization methods are adapted and dynamically applied to derive the basic properties of a "virtual" enclosure for a given set of indoor climate requirements. These values are then mapped to a construction database to identify an actual building enclosure construction (Mahdavi & Mahattanatawe, 2003).

The current generation of energy simulation software simulates building's multi-aspect nonlinear performances based on numerical methods of particle integrations. The building models are discontinuous with respect to some facility parameters, for simulation software, such as EnergyPlus and TRSYS, using adaptive algorithms and condition logic. Moreover, some of the parameters are discrete. Because of the tolerance to discontinue functions, stochastic optimization algorithms, such as GA (genetic algorithm) and PSO (Particle Swarm Optimization), are widely used to solve optimization problems with such building simulation models (Yang, et al., 2014). However, stochastic algorithms frequently require hundreds and thousands of runs of the simulation in order to reach the optimal result. Thus, a large amount of time, sometimes days and weeks on ends is consumed to solve the simulation-based building optimization problems.

Improved building energy control and operation strategies provide great opportunities to reduce building energy consumption. Therefore, an optimum operation strategy of the building's dynamic elements such as window openings, dynamic shading devices would also result in a significant reduction in the building's energy consumption. Natural ventilation systems use the freely available resources of wind and solar energy and, with proper design, they could represent an alternative technique for reducing the energy consumption in buildings and for creating thermal comfort and healthy indoor conditions. Typically, the energy cost of a naturally ventilated building is 40% less than that of an air-conditioned building (Stavrakakis, et al., 2012).

Chapter 2

2. Research methodology

Simulation-based optimization is an efficient method for finding optimum values of design/decision variables in the building envelope, heating/cooling systems, and energy generation systems (Hasan, et al., 2015). While an exhaustive search method will require a significant number of simulations to find optimal solutions, the optimization algorithm will need reasonable time and a number of simulations to find comparable solutions. Within a specific set of constraints (climate conditions, building use and occupancy, availability of materials and technologies on the market…), the effective design of sustainable buildings results from an accurate optimization process of all the variables that are involved and interrelated in meeting all the sustainability goals in the field of energy, indoor environmental quality, water management, and sustainable materials. In this field, the application of the principles of the integrative design could help in effectively managing and optimizing synergies between the complex set of technical and living systems associated with design.

In this chapter, the approach of the research will be explained step by step. The main aspects include the building energy performance simulation, optimization and algorithms, variables and indicators and finally the specific idea of this study based on the previous findings.

2.1. Energy simulation in buildings

Using computer programs for building energy analysis is not new. Since the late 1960s, the number of computer programs in both the public and private sectors has proliferated (Fiske & Bhonde, 2015). As Computer-Aided Design (CAD) systems possessing the drafting function proliferate, the demand for advanced performance appraisal software grows. Designers will then come to rely on simulation as the means to test alternative design hypotheses throughout the design process as well as post occupancy.

Many simulation engines have been tested during this study. In the end, the main energy simulation engine is EnergyPlus. EnergyPlus is a whole building energy simulation program that engineers, architects, and researchers use to model both energy consumption—for heating, cooling, ventilation, lighting and plug and process loads—and water use in buildings (Anon., 2018). EnergyPlus is a console-based program that reads input and writes output as text files. It ships with numerous utilities including IDF-Editor for creating input files using a simple spreadsheet-like interface, EP-Launch for managing input and output files and performing batch simulations, and EP-Compare for graphically comparing the results of two or more simulations. Several comprehensive graphical interfaces for EnergyPlus are also available. For developing the first platform of this study, the text file produced by EP-Launch was used. Ladybug and Honeybee were used for the energy simulation of the platform, which was developed in Rhino-Grasshopper. Ladybug Tools is a collection of free computer applications that support environmental design and education. Of all of the available environmental design software packages, Ladybug Tools is among the most comprehensive, connecting 3D Computer-Aided Design (CAD) interfaces to a host of validated simulation engines (Sadeghipour Roudsari & Mackey, 2018). Ladybug allows the user to import and analyze standard weather data in Grasshopper; draw diagrams such as Sun-path, wind-rose, radiation-rose, etc.; customize the diagrams in several ways; run radiation analysis, shadow studies, and view analysis (Figure 2-1). Honeybee supports detailed daylighting and thermodynamic modeling that tends to be most relevant during the middle and later stages of design. To be more specific, it creates, runs and visualizes the results of daylight simulations using Radiance, the results of energy models using EnergyPlus/OpenStudio, and heat flow through construction details using Berkeley Lab Therm/Window (Sadeghipour Roudsari & Mackey, 2018).

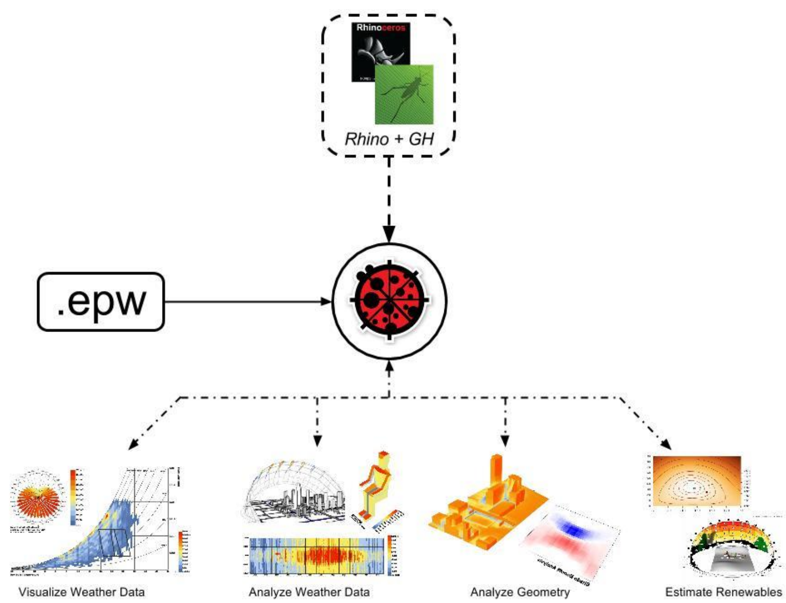


Figure 2-1 Demonstration of Ladybug's functionalities. (Sadeghipour Roudasri, 2018)

It accomplishes this by linking these simulation engines to CAD and visual scripting interfaces such as Grasshopper/Rhino and Dynamo/Revit plugins. It also serves as an object-oriented Application Programming Interface (API) for these engines. For this reason, Honeybee is one of the most comprehensive plugins currently available for environmental design (Figure 2-2). Ladybug and honeybee along with the other tools of this family are working together closely (Figure 2-3).

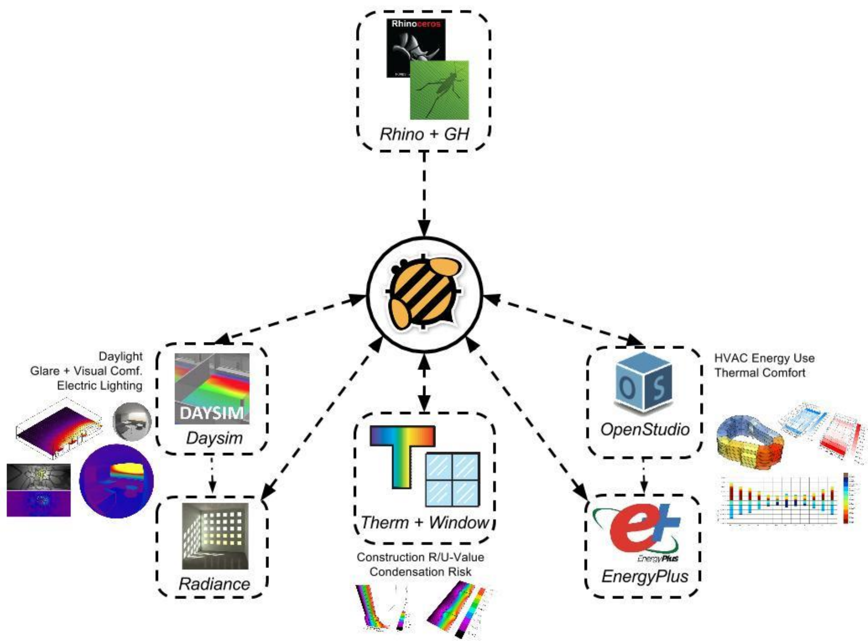


Figure 2-2 Demonstration of Honeybee functionalities (Sadeghipour Roudasri, 2018)

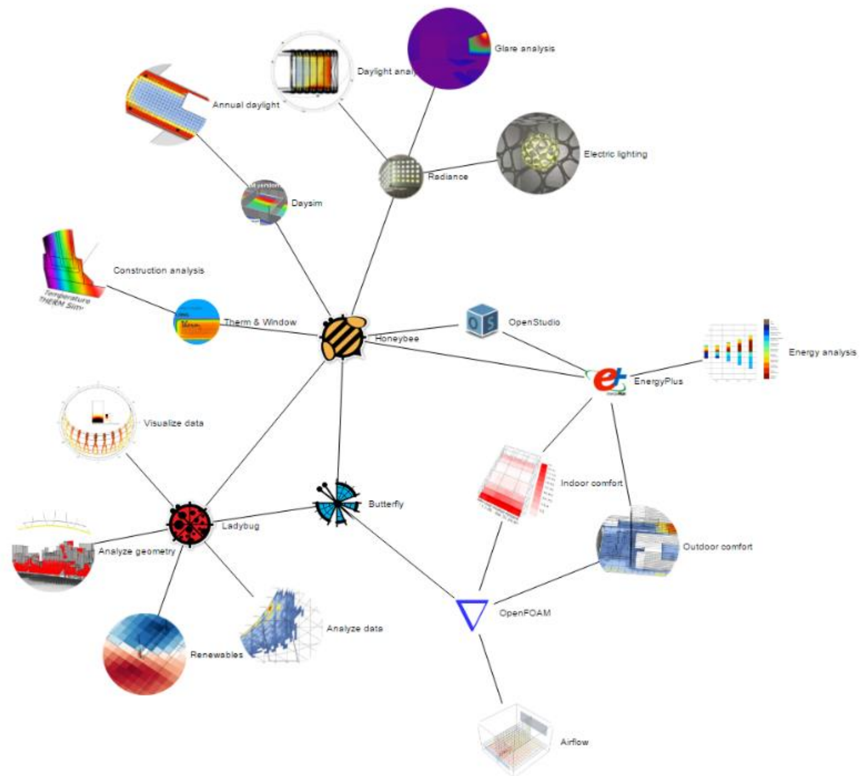


Figure 2-3 demonstration of the Ladybug tools and the simulation engines connections. (Sadeghipour Roudasri, 2018)

2.2. Optimizers and algorithms

To find energy-efficient building designs, building energy simulation is increasingly combined with black-box optimization methods. During this research, several algorithms have been considered in different platforms.

The first optimizer which has been used, is GenOpt. GenOpt is an optimization program for the minimization of a cost function that is evaluated by an external simulation program, such as EnergyPlus, TRNSYS, Dymola, IDA-ICE or DOE-2 (Figure 2-4). It has been developed for optimization problems where the cost function is computationally expensive, and its derivatives are not available or may not even exist (Wetter, 2016). GenOpt can be coupled to any simulation program that reads its input from text files and writes its output to text files. The independent variables can be continuous variables (possibly with lower and upper bounds), discrete variables, or both. Constraints on dependent variables can be implemented using penalty or barrier functions.

GenOpt has a library with local and global multi-dimensional and one-dimensional optimization algorithms, as well as algorithms for doing parametric runs (Wetter, 2016). GPSHookeJeeves and GPSPSOCCHJ, which are the different implementations of Generalized Pattern Search (GPS) algorithms from GenOpt library, have been employed for the optimizations of this study. GPSHookeJeeves does not deal with discrete variables but GPSPSOCCHJ do take the discrete variables into account.

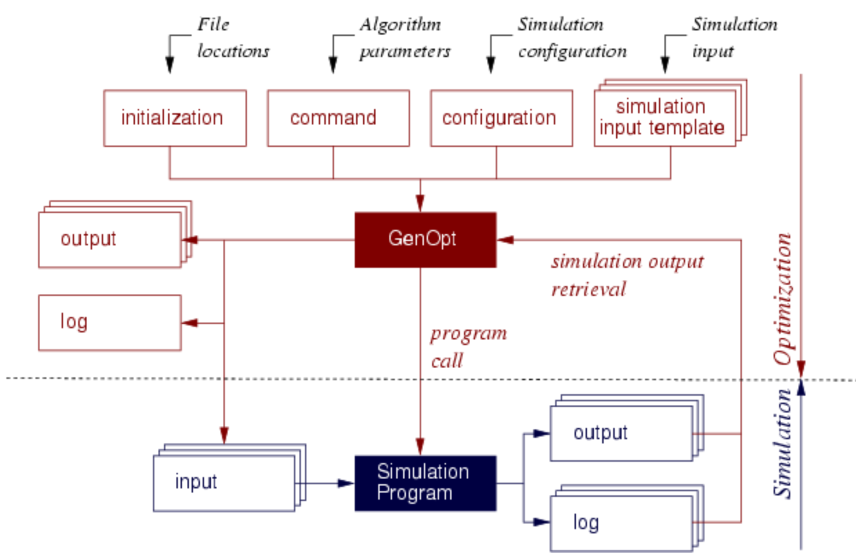


Figure 2-4 An overview of GenOpt's feature and functionality. (Wetter, 2016)

The other optimizer which has been used is Opossum. Opossum is a new optimization plug-in for Grasshopper. It is the first publicly available, model-based optimization tool aimed at architectural design optimization and is especially applicable to problems that involve time-intensive simulations of day-lighting and building energy for example (Wortmann, 2017). It uses advanced machine learning techniques to find appropriate solutions with a small number of iterations. This high speed of convergence is important for sustainable design problems such as daylighting and building energy, where a single simulation takes several minutes or hours to complete. In such cases, it is impractical to perform the thousands of simulations required by population-based metaheuristics such as genetic algorithms (GAs).

2.3. Variables and performance indicators

In short, this research considers multiple influential parameters of the building's envelope in regard to its energy performance. The main issue is how to cover and manage all of the variables in the most efficient way in terms of the time, accuracy and flexibility.

The primary categories of the variables are the design variables and the control variables. Design variables cover all of the geometric parameters as well as the material and construction properties of the building's envelope. The control variables consider the operation strategies of the dynamic elements of the building enclosure, such as venetian blinds control, windows opening control, etc.

The performance indicators are the annual heating energy demand, cooling energy demand and electric light energy demand of the building. Obviously, depending on the location of the building and priorities of the designers the combination of those indicators might lead to different objective functions in the optimization part in order to achieve the best design and operation solutions.

2.4. Idea of the clusters

Over the last decades, a number of developments have made global optimization of large multi-dimensional design option spaces possible. Such developments include the increase in computing power, emergence of sophisticated optimization algorithms, and new techniques for the derivation of computationally efficient meta-models. Along with their promise, such developments also involve a number of potential drawbacks. For one thing, meta-models occasionally fail to capture the behavior of "non-conventional" and complex designs. Another critical problem pertains to the potentially opaque nature of large-scale global optimization exercises, which make them less amenable to provision of intuitively graspable support in the –typically iterative – design process (Mahdavi, et al., 2016). In this context, the current research study explores the potential of a novel approach toward iterative global optimization of locally optimized attribute clusters of building design solutions. Thereby, clusters of design space attributes that are comprehensible to typical building designers as a compound yet coherent aspects of a design are made subject to multiple passes of local simulation-assisted optimizations. Hence, instead of allocating an individual dimension to each and every variable of a complex design within the context of a single-pass global optimization campaign, multiple iterative optimization steps target coherent clusters of such attributes and pursue those until the overall design meets the expected performance (or until further performance improvement is not forthcoming) (Mahdavi, et al., 2016). Moreover, the implementation targets scalability and flexibility: more specifically, users are to be provided with degrees of freedom in view of the selection of the clusters to be optimized. Besides, additional clusters can be defined and variable sets in each cluster can be manipulated, while still achieving convergence within reasonable temporal horizons. Therefore, different system operation options are imaginable, predefined sequences between attribute clusters (Figure 2-5) or random cycling between attribute clusters (Figure 2-6).

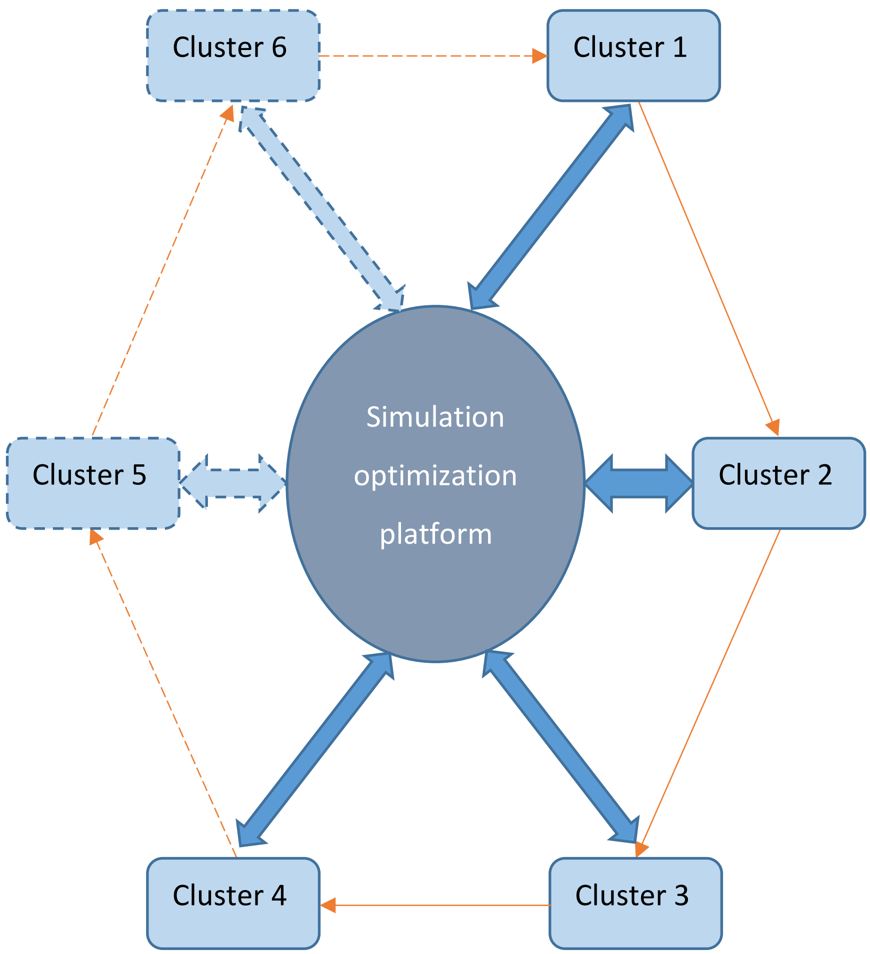


Figure 2-5 Illustration of predefined sequence between attribute clusters.

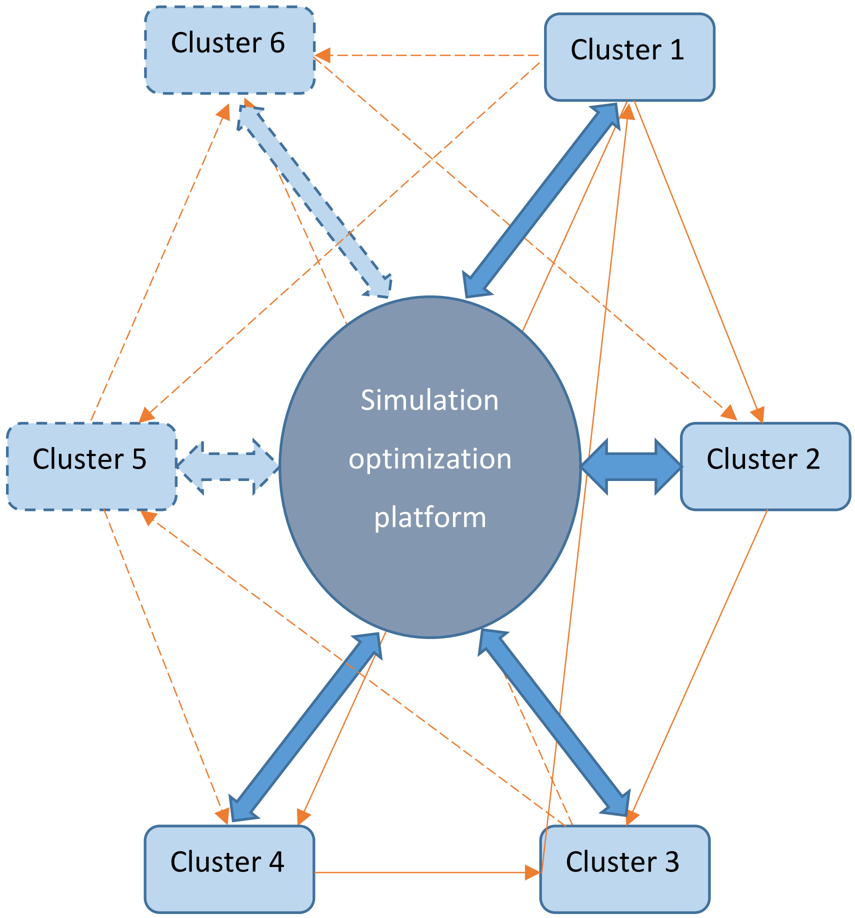


Figure 2-6 Illustration of random cycling between attribute clusters.

The effects of the building designer decisions begin since the first imaginary concept comes through her/his mind and continues even after the destruction. That could influence the form of the building, functionality, cultural meanings in the context, psychological effects on the occupants, environmental effects and etc. The variety of the parameters results in a huge amount of the data which needs to be well categorized and organized to achieve an acceptable solution in each stage (design, construction, operation, etc.). The idea of the clusters makes the designer able to consider all those variables separately but connected together in an efficient way. Monitoring the variation of each variable during the optimization and its effect on the predefined goal (goals) is a great opportunity for the designer to make the decisions clearer and sagely. Since the sustainability is a multi-dimensional procedure which is related to all the aforementioned branches of the building performance, a well-defined system of the attribute clusters for the cycling between the optimization runs would make a smooth, transparent and accurate way to achieve an optimum solution for the building design and operation. Some examples of the clusters which could be defined by the designer come below.

Building Geometry (BG)

The geometry of the building is one of the first issues an architect involves at the early stages of the design procedure. The range of the variables is very broad. It could cover the orientation, footprint, the aspect ratio of the building, the glazing to wall ratio, dimensions of the elements to be annexed to the building, and so forth.

Material properties / Constructions (MC)

When achieving energy efficiency is the goal of the optimization, selecting the proper material and constructions is a vital factor to be considered at the design stage. The material properties could be taken into account layer by layer or as a package of predefined constructions for the building elements from the market or other sources. These options could be the variables of this cluster.

Moveable Shading Device properties (MSD)

Moveable shading devices can be very effective on the building performance from different aspects such as energy consumption, visual comfort, etc. but they need to be selected and designed correctly in terms of the geometry and material of the elements. The slat width, the distance between the slats, the distance between the shading device and the window, and the material properties such as the reflectance of the slats, are some examples of the variables for this cluster.

Functions Organizations (FO)

Organizing the internal functions of the building is very important for the architects from different aspects. For instance, in terms of energy efficiency, different functions which are the different thermal zones should be located in the right position in the building. Therefore, it is necessary to find the optimum zoning system for the building. It could be considered with the other criteria such as internal connections between the functions, then it will be a multi-objective optimization problem to be solved.

Landscape Design (LD)

One part of the energy and comfort performance of the buildings is related to the position of the building in the site and effects of the natural and artificial obstacles to the building performance. In many projects designing the landscape is a part of the problem which needs to be solved. That also can be a cluster of the variables.

Renewable Energy Applications (REA)

There is a broad range of technical opportunities, means, and methods for incorporating renewable energy technologies into building designs and operations. Depends on the local environmental and climatic conditions, there is always a high chance of using renewable energy sources and available technologies successfully to offset building electrical and thermal energy loads.

Renewable energy resources commonly used for building applications include solar, wind, geothermal, and biomass (Hayter & Kandt, 2011). Therefore, many design factors of the solar electric or photovoltaic (PV) systems, solar thermal including solar hot water and solar ventilation air preheating, geothermal heat pump, wind turbines, biomass systems could be a cluster of the variables.

Building Management System (BMS)

In this context, it could be the control system of the mechanical and electrical systems or the moveable elements of the building. Finding optimum set points to set the equipment or elements on or off as well as finding a good combination of controlling different sectors, will be a very useful cluster of variables to reduce energy consumption besides operation cost of the building.

Prefabricated Elements (PE)

In prefabricated buildings, finding the optimum combination of the different prefabricated elements according to the location of the building, designer ideas, energy, material properties, transportation, and etc. will be an important cluster of the variables to achieve the optimum design solution.

Building Schedules (BS)

There are many schedules for the building operation that are planned based on the building use, climatic conditions, and so forth. For instance, occupancy, lighting, heating, cooling schedules will need to be well designed and run to get the optimum energy performance or any other goal from the building. This could be also a very effective cluster of variables.

As it has been shown, variety of the clusters are imaginable to be defined and assigned to the optimization platform; what came above were just some examples.

In Figure 2-7 to 2-10 some combinations of the attribute clusters will be graphically illustrated.

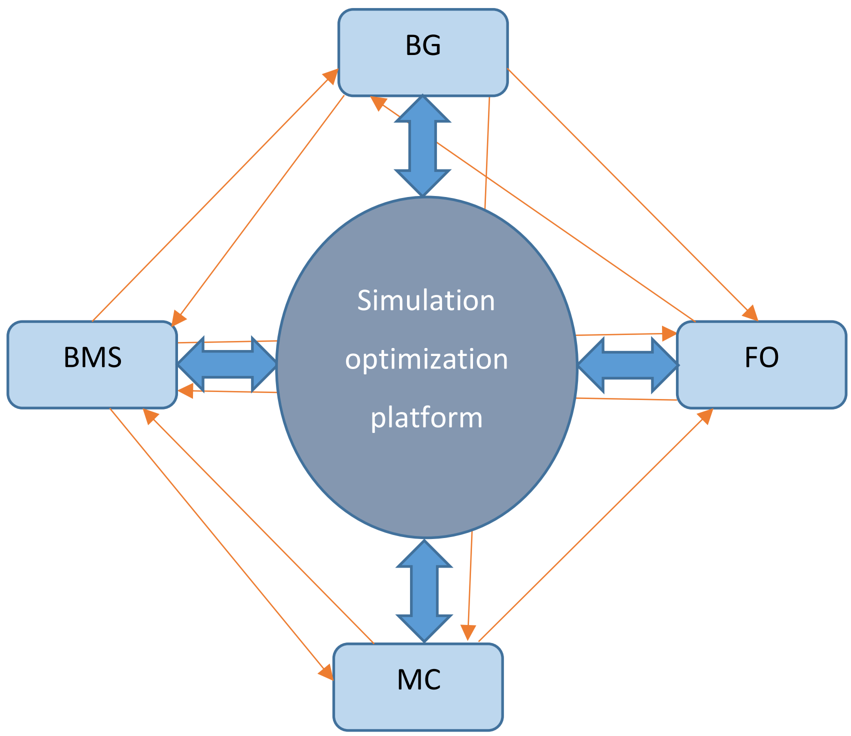


Figure 2-7 Combination of four clusters in a random cycling between them.

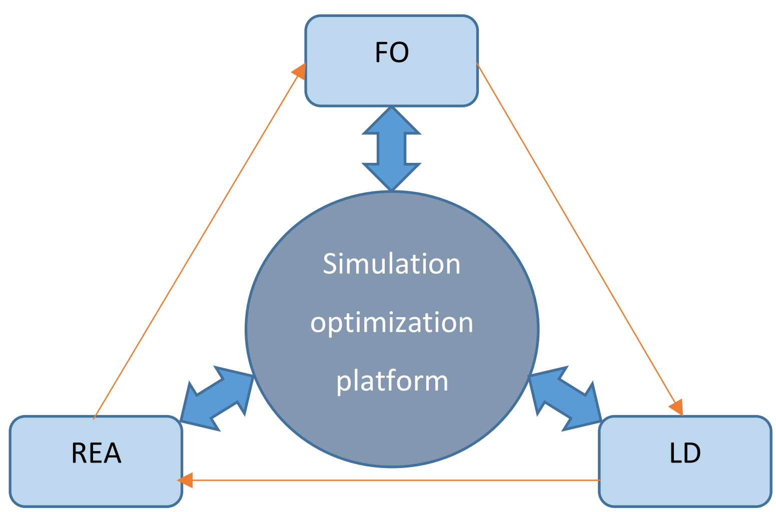


Figure 2-8 Illustration of predefined sequence cycling between three attribute clusters of variables.

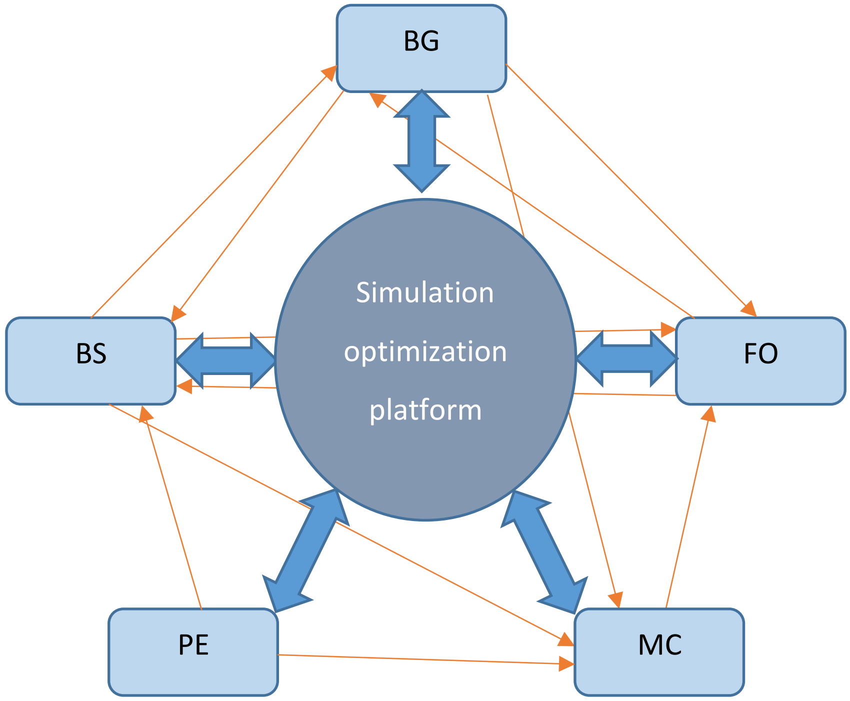


Figure 2-9 Random cycling between five clusters of variables.

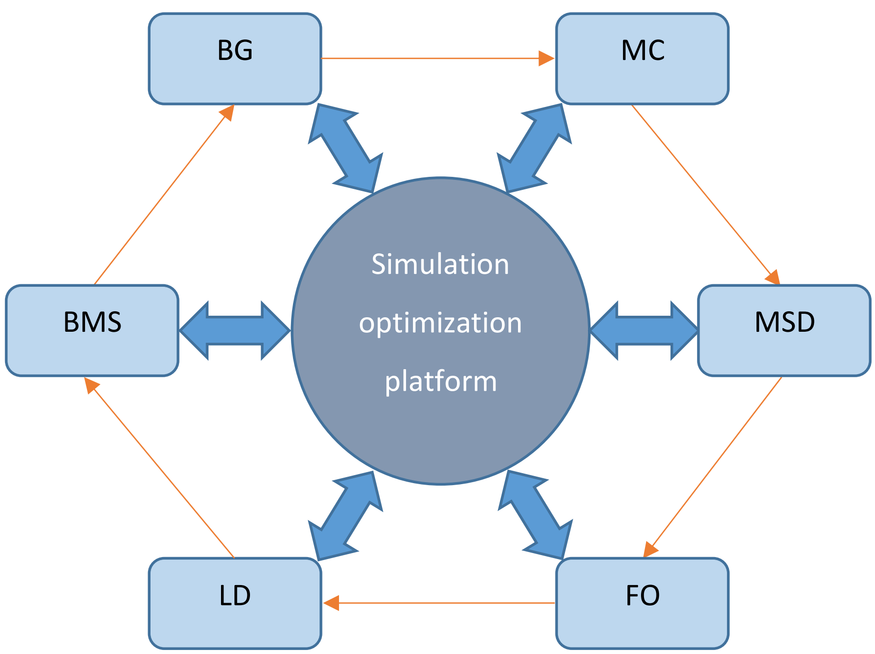


Figure 2-10 Illustration of cycling between six attribute clusters of variables.

Chapter 3

3. Implementation of the system

In this chapter, two implementations of the system will be presented and discussed. It is necessary to develop a platform for coupling simulation engine and the optimizer for this specific way of optimization.

In general, inputs of the system are building initial information and the proper objective function which the designer has in mind. Building initial information covers all of the information of the building which is necessary for the energy simulation such as geometry, building use, the location, the weather data, etc.

3.1. Architecture of the system

Figure 3-1 shows the general architecture of the system which is going to be discussed in more detail in the next pages.

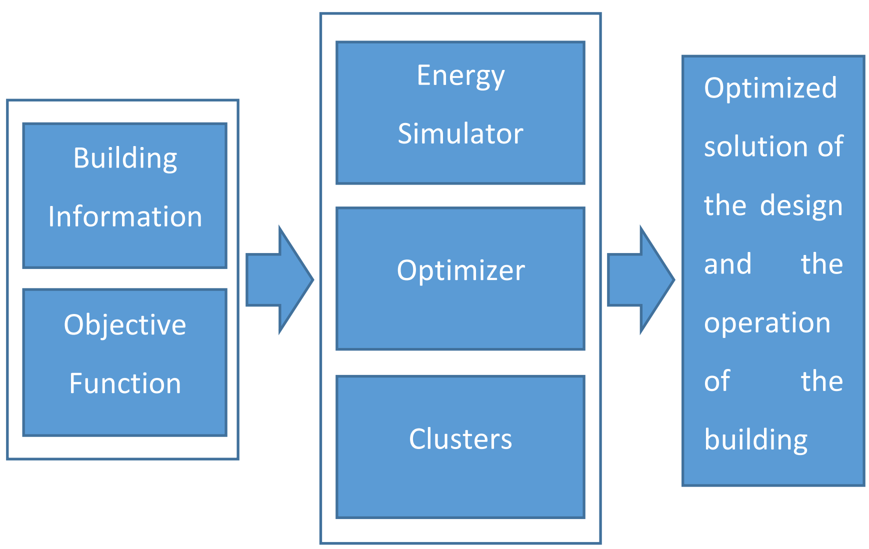


Figure 3-1 General architecture of the system.

3.2. Platform one: Java-based (EnergyPlus and GenOpt)

In this approach, the platform was developed in Java which takes the idf file produced by EnergyPlus and uses it as the GenOpt template file. This file contains the energy model and the placeholders for the variables that are going to be optimized. Also, the other necessary files to run GenOpt which are GenOpt initialization file; this file provides the important data for Genopt to run, such as, the objective function of the optimization, and the the location of the other necessary files for GenOpt. Command file which clarifies the variables and one specific algorithm from the GenOpt library to use. Additionally, the configuration file which is mainly for GenOpt to find the simulation engine to run, in this case EnergyPlus.

In this platform, one Java class has been created for each cluster of variables. This class goes to the idf template file, finds the parameters related to the variables of the cluster and replaces the proper placeholders for GenOpt to read. Then, runs GenOpt from the command prompt; obviously the other necessary files for running GenOpt which are the initialization file, command file, and the configuration file, must be created and addressed in this class. When the optimization of the cluster is completed, it takes the optimized values and replaces them in the idf template file for the next optimization. The program also adds the optimized values of the variables and the other important data from the optimization, such as the best value of the objective function and of the energy demand of the building after the optimization, zone-by-zone and in total, to an excel file which had been previously created by the program. Figure 3-2 illustrates the general architecture of the platform.

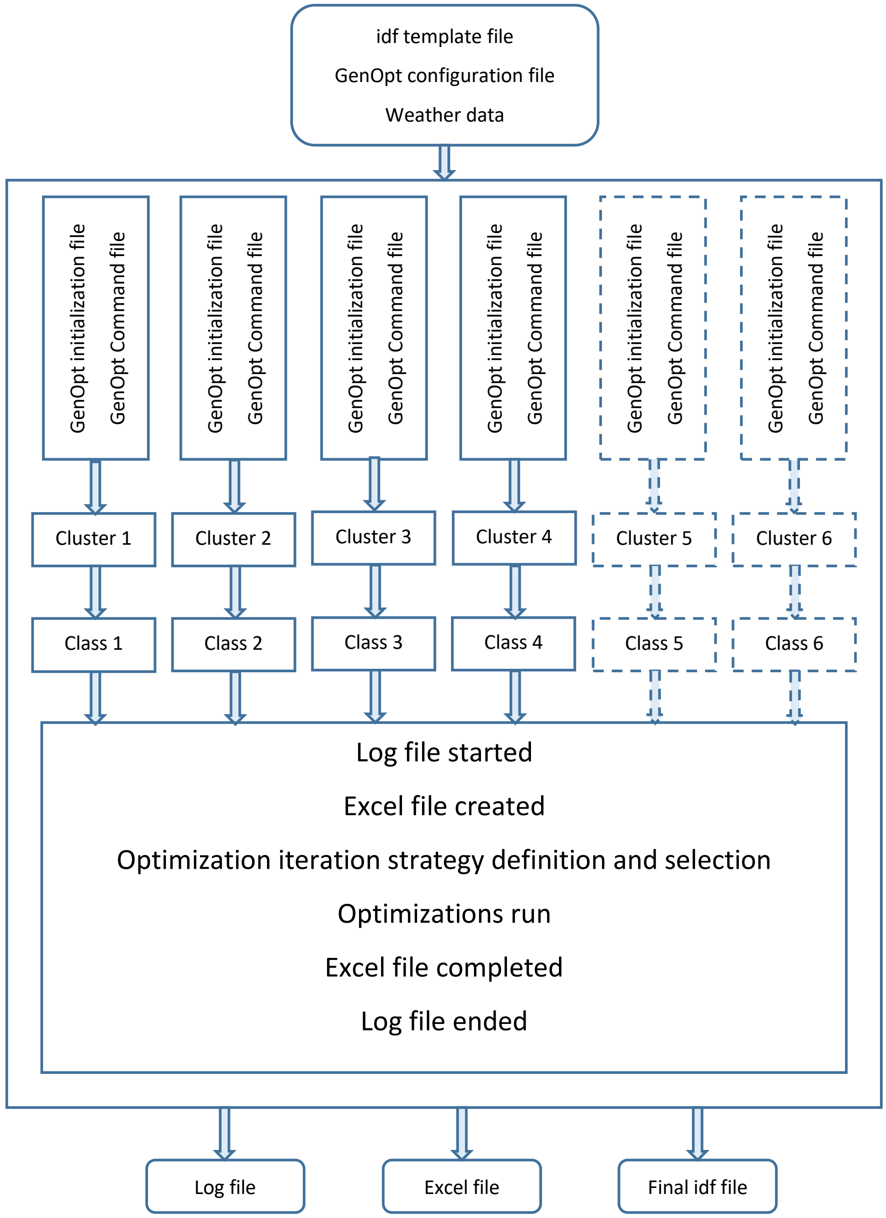


Figure 3-2 Java-based platform general configuration.

Above those classes, there is a main function which creates the aforementioned excel file, and specifies the number of optimization iterations as well as the strategy of iteration in regard to their sequencing, which is either predefined or random. Moreover, a log file will be produced which records all details from the beginning of the procedure to the end.

When dealing with more clusters, it is necessary to create the proper similar classes and identify them in the main function.

3.3. Platform two: Rhino-Grasshopper

(Ladybug, Honeybee, and Opossum)

Ladybug and Honeybee were used for energy simulation in Grasshopper. Opossum was the optimizer in this platform. The building geometry model can be created in Grasshopper itself or in Rhino. If the geometry is not a Brep (Boundary Representation), then it first needs to be converted to that in Grasshopper before it can be used to create the thermal zones. Obviously, the strategy of defining the thermal zones is dependent on the designer. As an example, it could be a combination of a core and perimeter zones (Figure 3-3).

The next necessary step is to define the building use program. All the predefined EnergyPlus building programs are available in Honeybee (Figure 3-4).

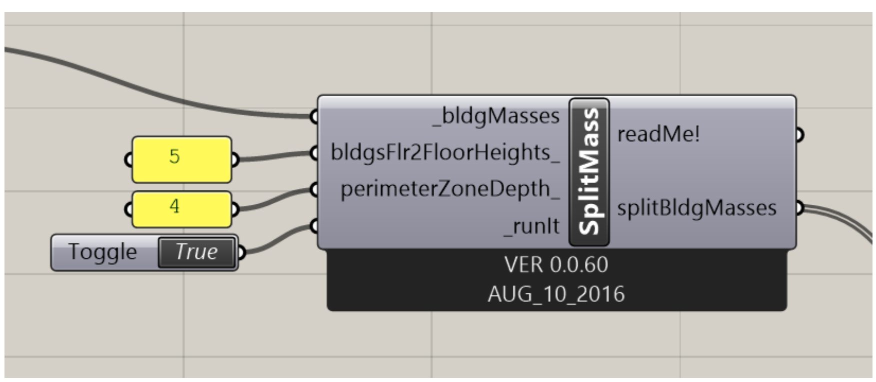


Figure 3-3 Splitting the building mass to a core and perimeter parts to create the thermal zones.

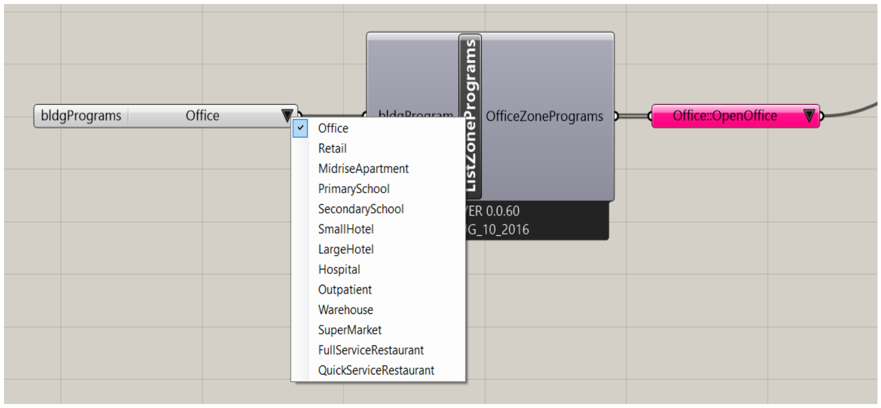


Figure 3-4 Building program selector.

To import the weather data, it is possible to use an existing EnergyPlus weather file or directly download it from the website (Figure 3-5).

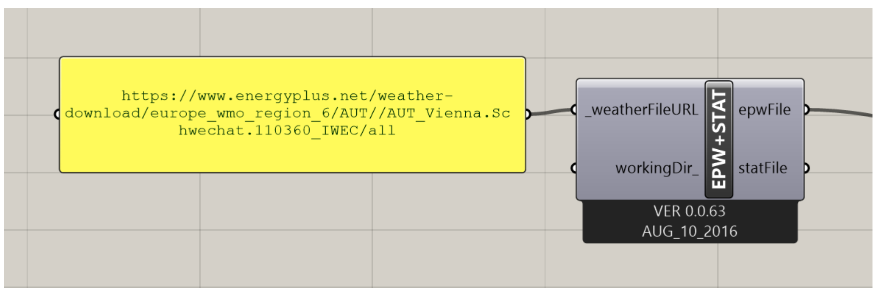


Figure 3-5 Defining the weather data for the energy simulation.

There is another component to access the website, find the proper weather station, and get the URL of the weather file needed for the energy simulation; this can be easily connected to the component as a text in Grasshopper (Figure 3-6).

Defining the objective function for the optimization depends on the designer opinion and goal. Since the consideration of this research is on the energy, the relevant outputs have been defined and used for calculating the objective function. The outputs are the annual energy demands of heating, cooling and electric lights (Figure 3-7).

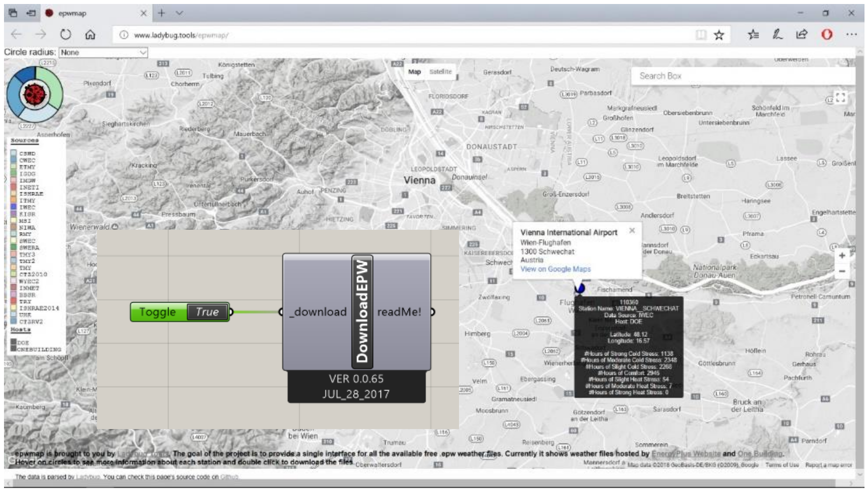


Figure 3-6 Finding the proper weather station to get the necessary weather data.

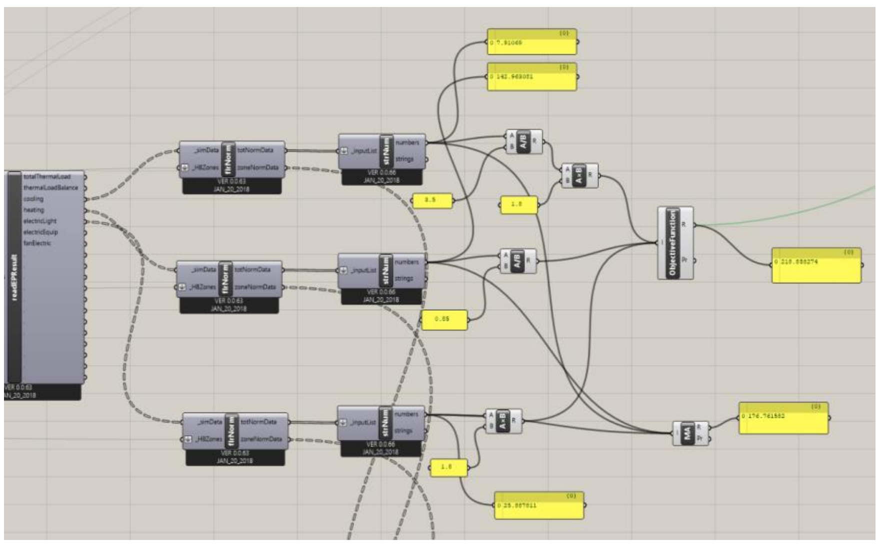


Figure 3-7 Defining the objective function.

For each cluster one optimizer (Opossum) is employed (Figure 3-8). Each variable must be a Grasshopper Number Slider (Integer, Double, etc.) and is connected to the optimizer. In the case of discrete variables, it is necessary to put the values in a list and then select the items from the list. The values will be taken from the list by their index. The other necessary input of the optimizer is the Objective Function of the optimization. It must be a number and is connected to the other port of the optimizer. In the case of the random iteration of the clusters, a new component has been created to generate a list of random numbers for the optimizations. Adding a new cluster is simply possible by adding a new optimizer and connecting the relevant variables and the objective function to it.

The general architecture of the platform in Rhino-Grasshopper is illustrated below (Figure 3-9).

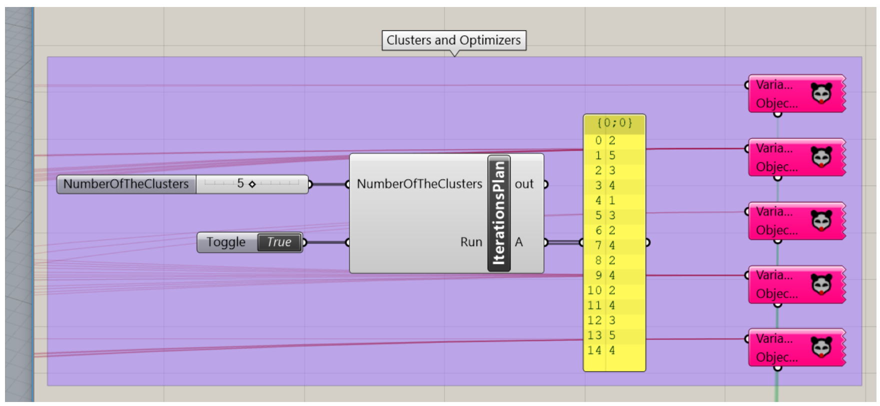


Figure 3-8 Defining the Clusters and the strategy of the iterations.

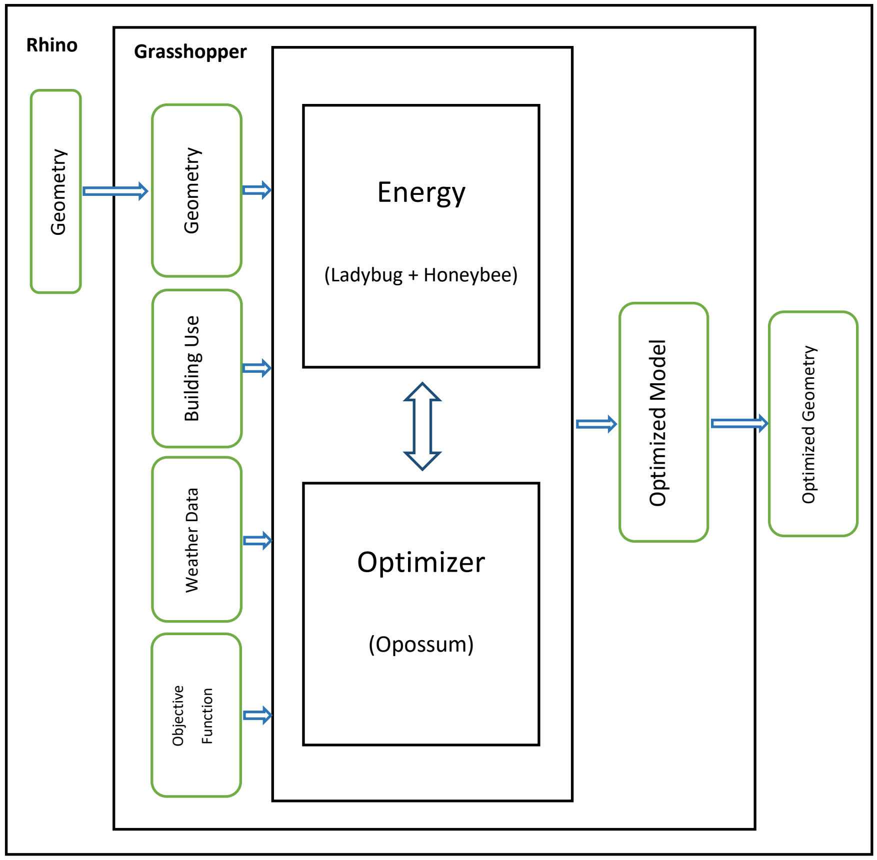


Figure 3-9 Architecture of the system in Rhino-Grasshopper.

3.3.1. The energy simulator

Energy simulator of the platform has several sections. The first step after receiving the geometry is the thermal zones and building program definition. This includes conversion of the mass to Honeybee zones and solving the adjacency. The other part of the platform is the glazing creator which defines the glazing for the energy model (Figure 3-10). The value of the glazing to opaque ratio is one of the variables of the Geometry cluster.

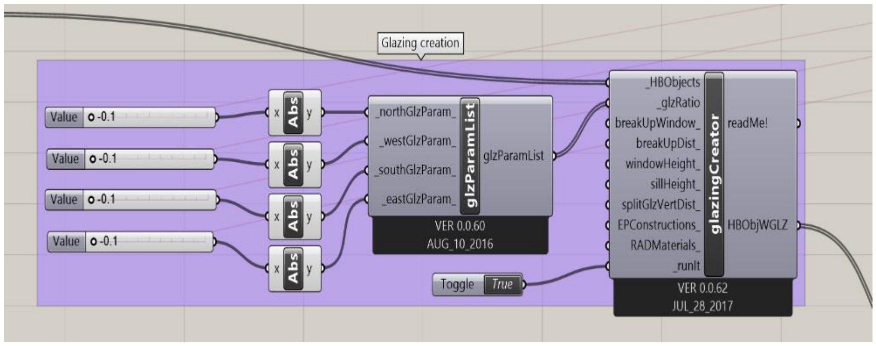


Figure 3-10 Defining the glazing for the energy model.

The other vital part of the energy simulator is the construction definer for the different surfaces of the building (Figure 3-11).

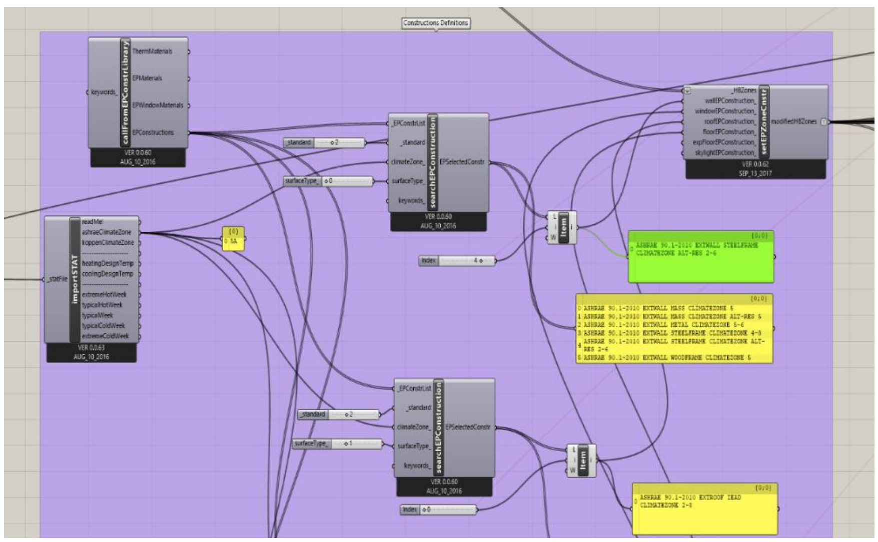


Figure 3-11 Construction definition for the energy model

Similar to the model in Energyplus, it is possible here to create your own material and then constructions, or use the material of the Energyplus material library and create the constructions based on them. It is also possible to use the appropriate predefined constructions of the Energyplus database for each type of the surfaces. There are some ways to narrow down the search space of material or constructions from the relevant libraries. One example is to extract the ASHRAE climate zone from the weather file which is based on the location of the building. This reduces the number of the available items dramatically.

There are also several thresholds which are defined in the platform (Figure 3-12) any of them could be a variable in a cluster. For example, the outdoor minimum and maximum temperature can be used to control the opening of the windows for natural ventilation.

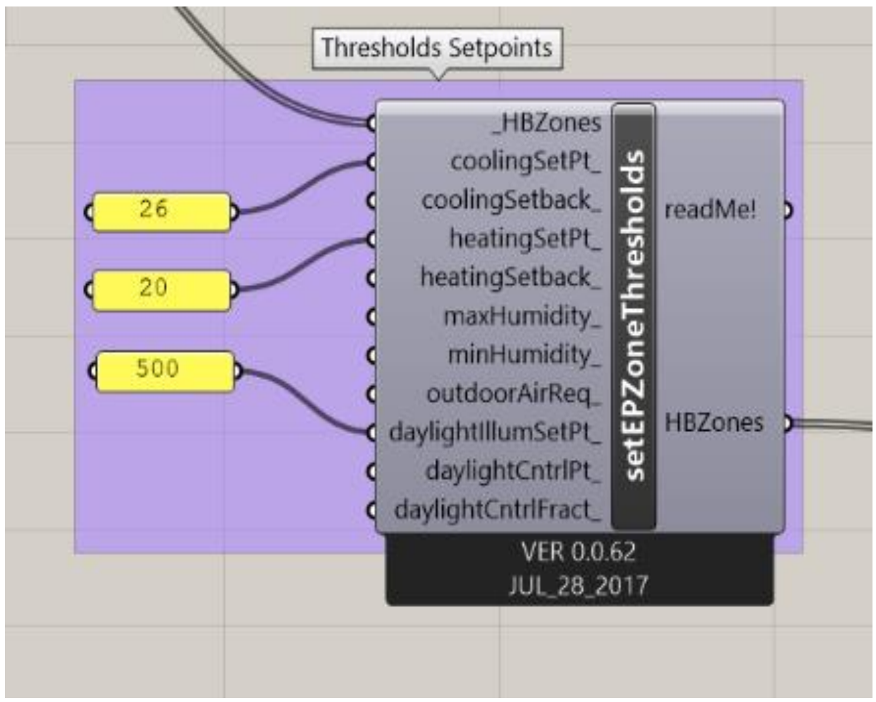


Figure 3-12 defining thresholds for the energy model.

The model can have two types of shading devices, venetian blinds (Figure 3-13) or fixed shading devices (Figure 3-14) (overhangs and fins) or both at the same time. With the Boolean switches (True/False), the user can enable or disable any of the shading device creators.

In case there is some context shadings on the building, the user can define the geometries, the other relevant settings of them, and connect it to the simulation run part of the simulator.

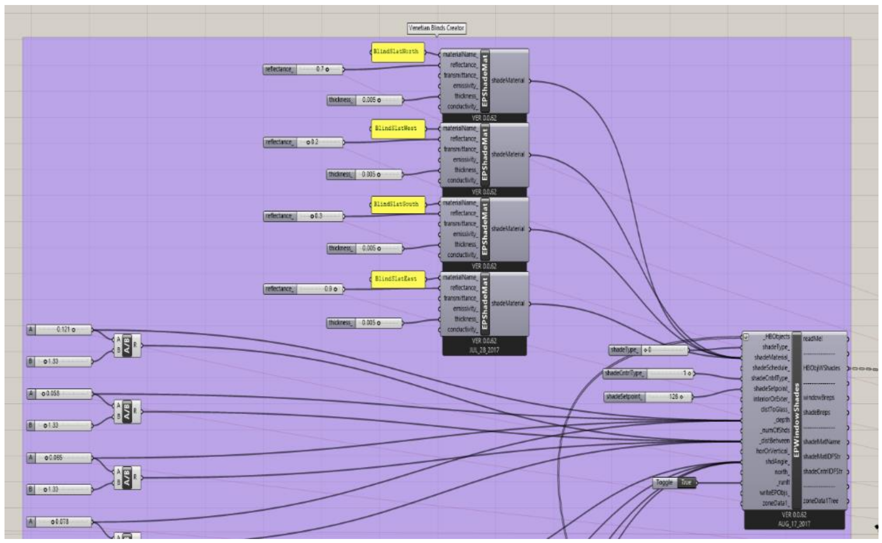


Figure 3-13 Venetian Blinds creator.

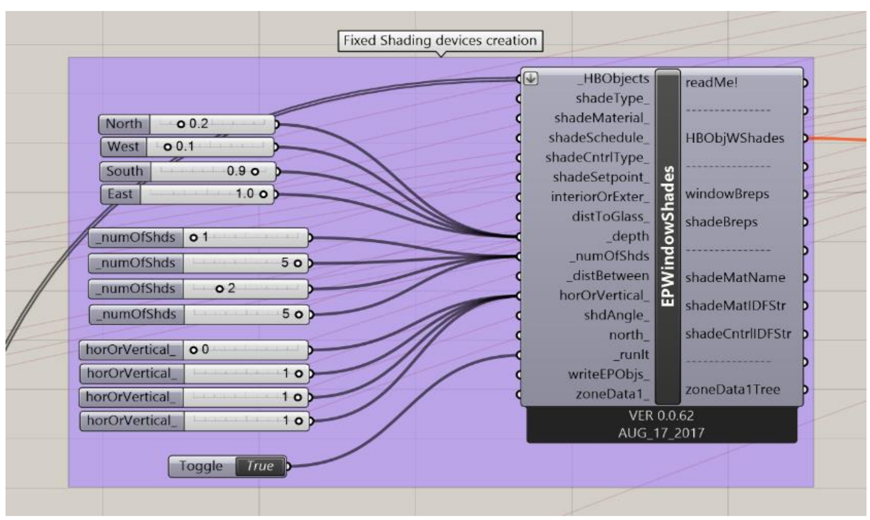


Figure 3-14 Shading devices creator.

Chapter 4

4. Demonstrative examples

4.1. Examples in Java-based platform

4.1.1. Example one

i. Model

The base model for the experiments is a modified version of a standard ASHRAE small office building retrieved from Commercial Prototype Building Models supported by the U.S. Department of Energy (DOE 2016). The office's dimensions are approximately 27.7m length, 18.5m width and 3.1m height. Five thermal zones have been defined, namely one core zone and four perimeter zones (Figure 4-1).

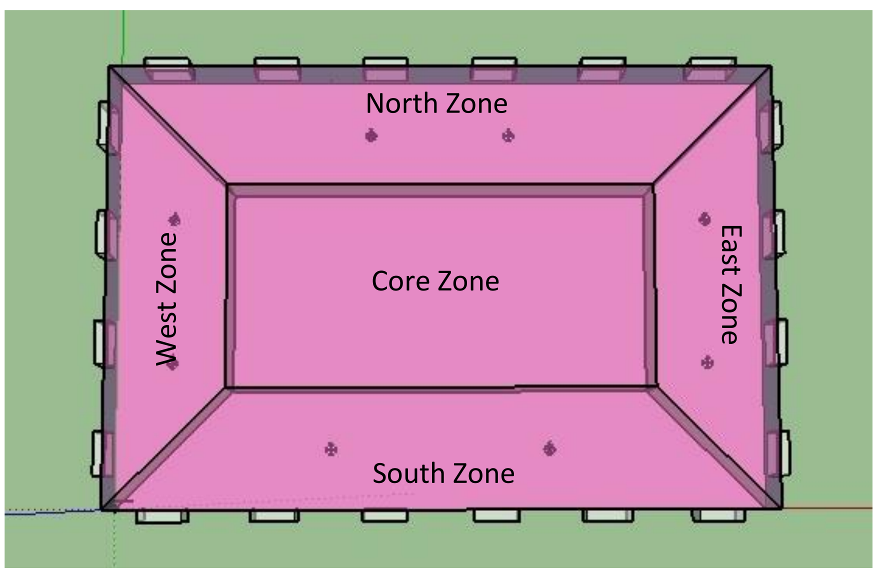


Figure 4-1 Schematic plan of the building model.

The assumed location of the building is Vienna, Austria. It has 20 similarly sized (1.8m by 1.8m) windows (6 windows on the south and north facades and 4 windows on the east and west facades). Each window has a 0.5m deep overhang and 2 lateral 0.5m deep fins (Figure 4-2).

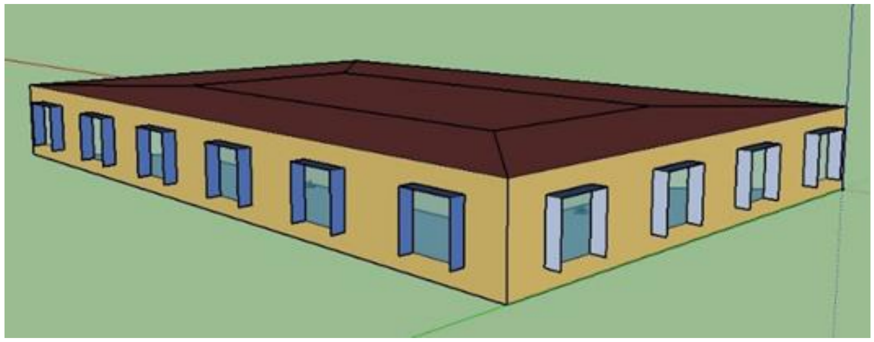


Figure 4-2 View of the building model.

The opaque part of the exterior walls is modelled in terms of two thermally distinct (inner and outer) layers (Figure 4-3).

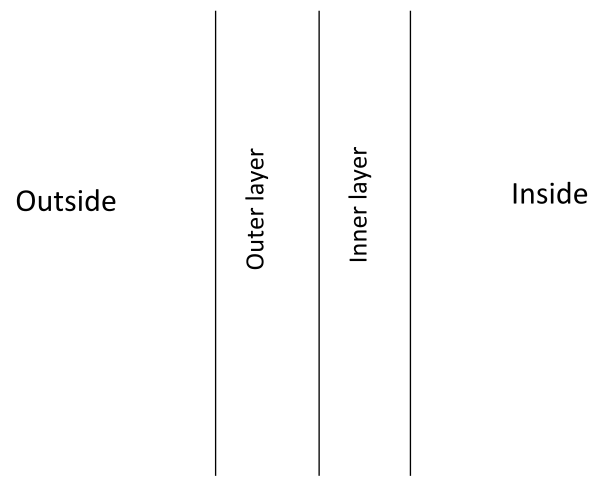


Figure 4-3 Configuration of the opaque part.

The office is assumed to be occupied by a total of 31 inhabitants (9 in the core zone, 7 each in the south and north zones, and 4 each in the east and west zones). Daylighting control sensor points are assigned to perimeter zones (half-way on the middle axis between the front and back walls, 0.8 m above the floor). Each window is assumed to have an exterior blind (movable shading device).

Natural ventilation at the perimeter zones is controlled based on the outside air temperature.

ii. Definition of the clusters

In this example, the optimization variables have been categorized in terms of four clusters. Three clusters entail design parameters and one pertains to control variables.

Façade Geometry (FG)

This cluster includes the dimension of the windows as well as overhangs and depth of the fins (Figure 4-4).

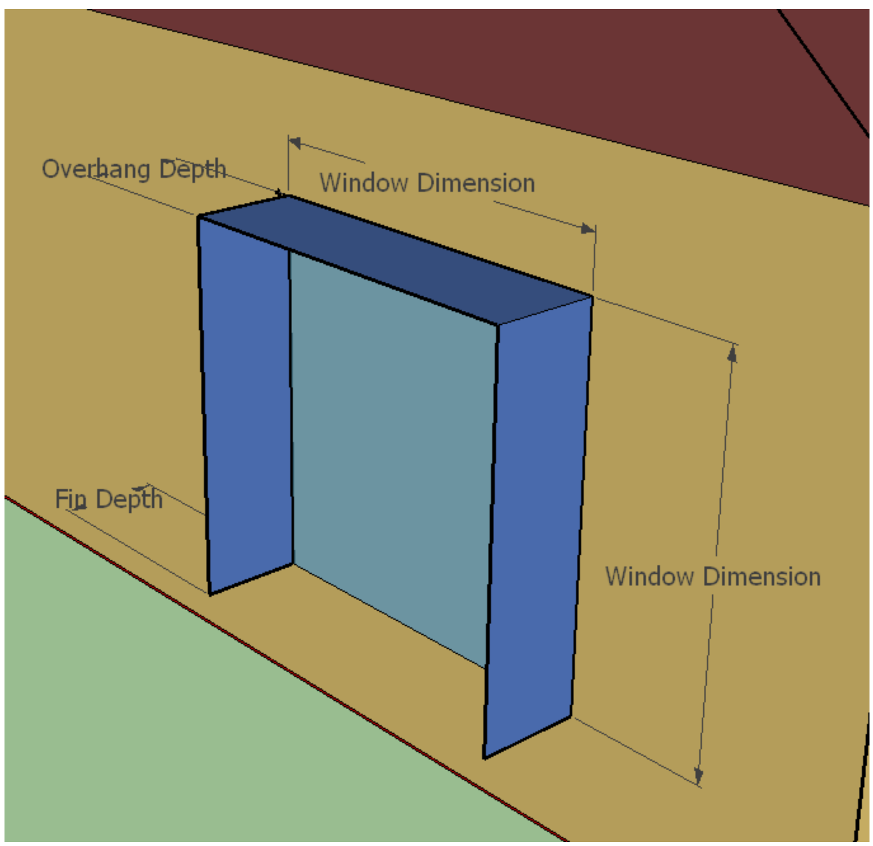


Figure 4-4 Variables of the cluster FG (Facade Geometry).

For each façade, the dimensions of the windows and the depth of overhangs/fins vary simultaneously. As a whole, this cluster has 12 variables.

Material Properties (MP)

The façade is modelled in terms of opaque and transparent (glazing) components. The opaque part consists of an outer layer (with thermal conductivity as the pertinent variable) and an inner layer (with density as the pertinent variable). The pertinent variables of the glazing are assumed to be the thermal transmittance (U-value) and the visible transmittance. It is important to note that for the purposes of the present demonstration, the glazing g-value is obtained as a function of its visible transmittance. Overall, this cluster has four variables.

Blind (movable shading device) Properties (BP)

This cluster comprises the pertinent variables of the movable blind, namely Slat Width, Slat Angle, and Slat Solar Reflectance (Figure 4-5). Slat separation is obtained as a function of the slat width. Given the independence of the four facades in view of blind operation, this cluster entails 12 variables.

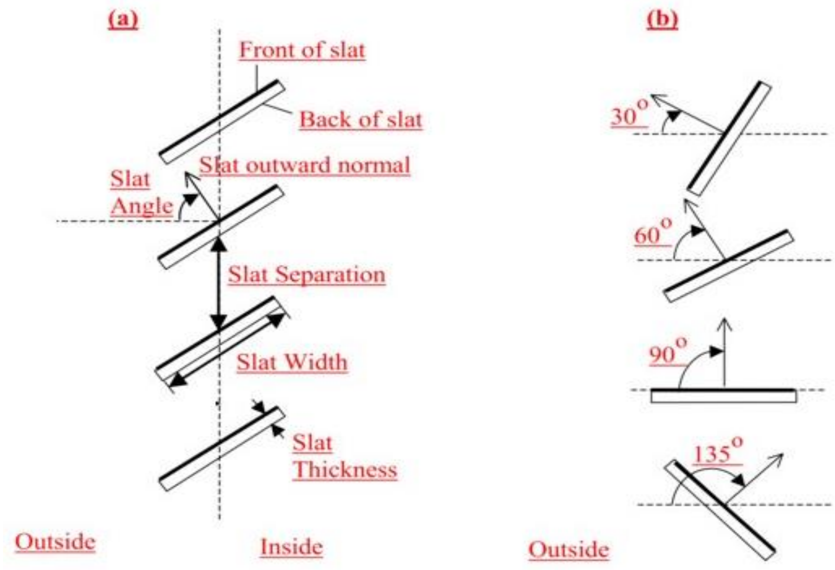


Figure 4-5 Venetian blind properties based on the instruction of EnergyPlus.

Control (CO)

The final cluster is concerned with operation of blinds and windows openings during the year. For blinds, the variable subjected to optimization is a threshold value for incident solar irradiance, above which the blinds are closed. Regarding ventilation, a base mechanical ventilation system is assumed that delivers a prescribed fresh air supply rate of 7 l/s per person. However, given the appropriate conditions (expressed in terms of an outdoor ambient temperature band), an additional magnitude of fresh air flow via window operation is supplied. In this case, the variables subject to optimization are threshold values for outdoor temperature which serve to keep the windows closed when the temperature is below or above the given threshold values. This cluster includes thus 12 variables (4 for blind and 8 for window operation).

Table 4-1 shows an overview of the clusters and variables.

Table 4-1 Summary table of the clusters and variables.

Clusters	variables	Lower band	Upper band	Units	Sum	
Façade Geometry (FG)	Windows dimension	0.20	2.65	m	12 variables	40 independent variables
	Overhang depth	0	0.90	m		
	Fin Depth	0	0.90	m		
Material Properties (MP)	Thermal conductivity	0.025	2.3	W/m.K	4 variables	
	Density	10	2950	Kg/m³		
	Thermal transmittance (U-value)	0.90	5.80	W/m².K		
	Visible transmittance	0.30	0.90	-		
Blind Properties (BP)	Slat width	0.00	0.15	m	12 variables	
	Slat angle *	15	135	°		
	Slat solar reflectance	0.10	0.90	-		
Control (CO)	Incident solar irradiance	10	250	W/m²	12 variables	
	Minimum outdoor temperature	16	22	°c		
	Maximum outdoor temperature	22	26	°c		

*This is a discrete variable and the values are multiples of 15.

iii. Tools and platform

A Java-based platform was developed for the purpose of integrating the simulation and optimization tools as well as automating the procedure. The energy analysis and thermal load simulation program EnergyPlus (see 2.1) was coupled with GenOpt (see 2.2). In this case, the optimization algorithm GPSPSOCCHJ was used (a Hybrid Generalized Pattern Search Algorithm with Particle Swarm Optimization Algorithm for Continuous and Discrete Variables).

iv. Approach

The approach of optimization iteration in this example is the random cycling between attribute clusters (Figure 4-6).

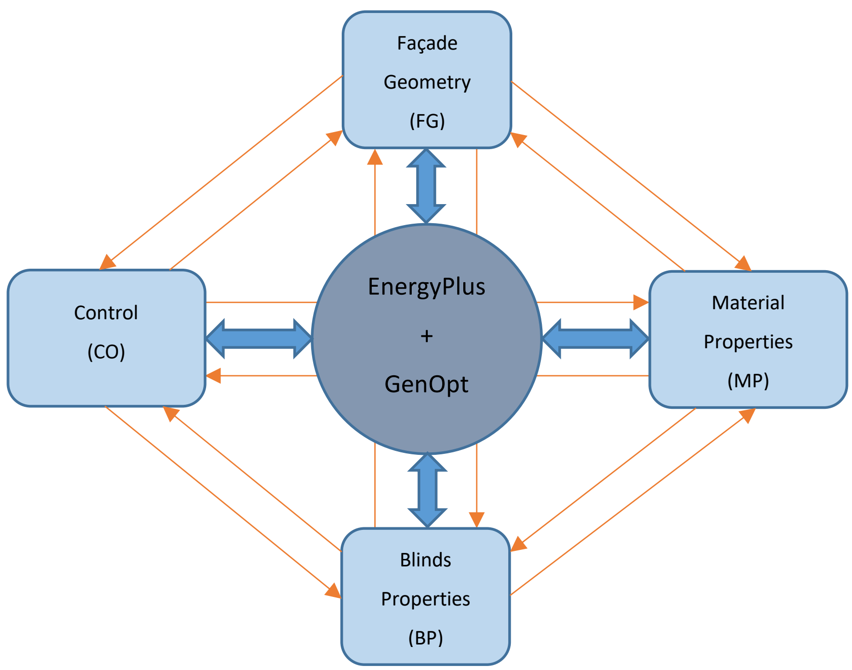


Figure 4-6 Illustration of random cycling between attribute clusters.

v. Performance indicators and cost function

The selected performance indicators in the present experiment (and the basis for the definition of the cost function (U)) include the building's annual heating (H), cooling (C), and lighting (L) energy demands. To establish the cost function, it is assumed that the building's heating system uses natural gas and has an 85% efficiency. The cooling system was assumed to be electrically driven with a COP of 3.5. Furthermore, it was assumed that, for the same energy content, the electricity price is 1.8 times the price of natural gas. This results in the following formulation for the cost function (U) (Equation 4-1). Therefore, H, C, and L are in units of J and U in units of MWh.

$$U = \frac{\frac{H}{0.85} + \left(1.8 \times \frac{C}{3.5}\right) + (1.8 \times L)}{3.6 \times 10^9}$$

Equation 4-1 Objective Function of the optimizations.

vi. Results

In this section, results of the optimization will be presented. The results include the charts of the objective function and energy demand values after each iteration, and the table of the variable values during the optimization procedure.

Abbreviations BM and GO refer to "Base Model" and "Global Optimization" respectively.

Figure 4-7 illustrates the evolution of the cost function together with the total energy demand during 12 random iterations between attribute clusters.

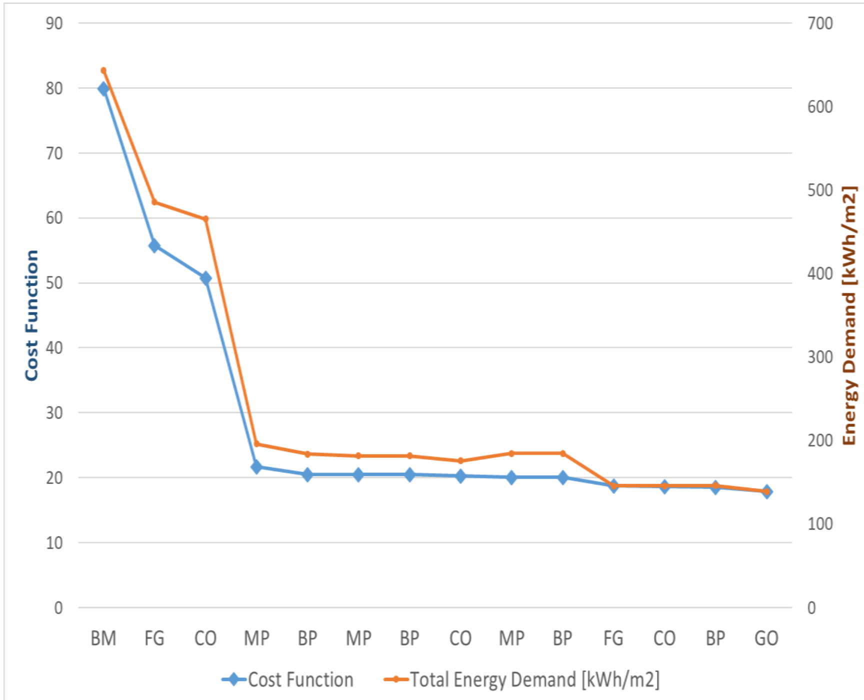


Figure 4-7 Cost function versus total energy demand.

Figure 4-8 shows the energy demand (heating, lighting, cooling, and total) evolution for the same iterations.

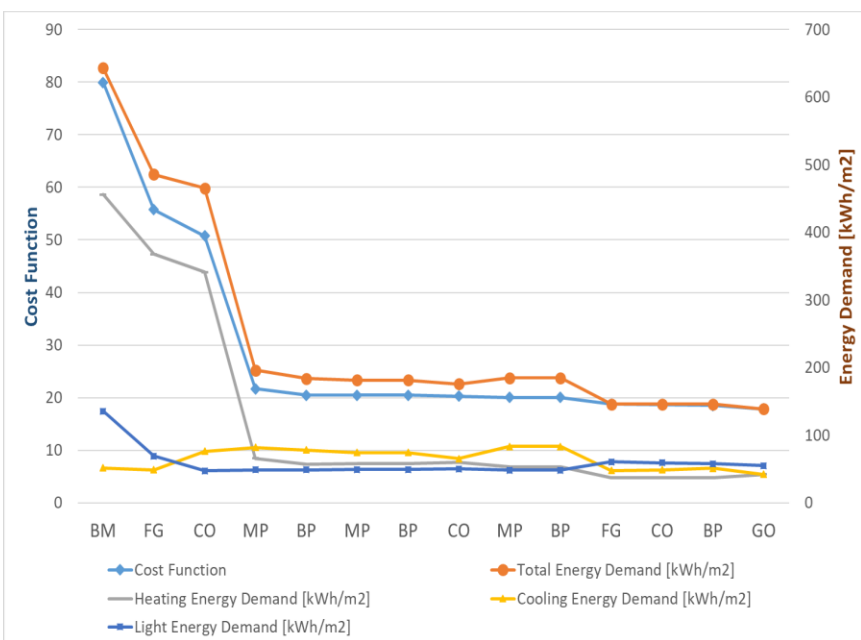


Figure 4-8 Energy demand evolution.

Figure 4-9 provides further details regarding the energy demand at the zone level.

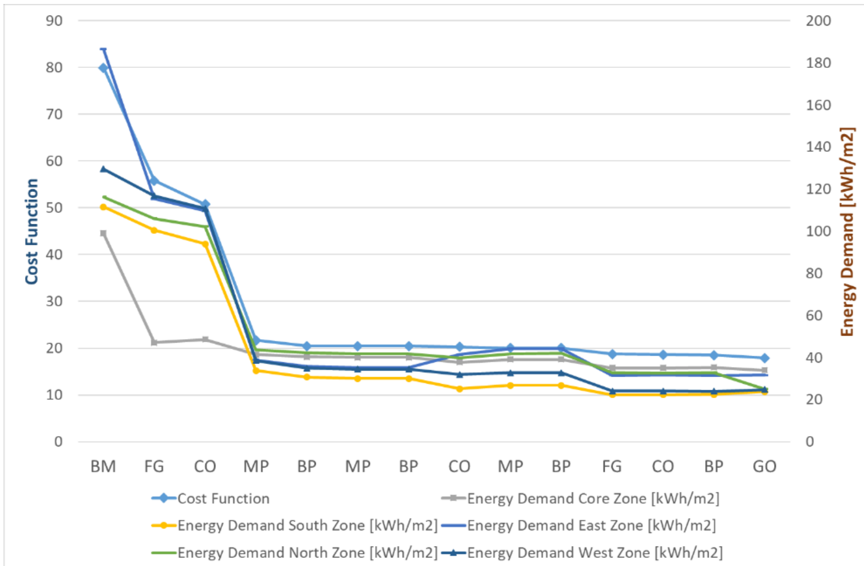


Figure 4-9 Energy demand evolution for each zone.

Figure 4-10 shows the core zone energy demand variation during the iterations.

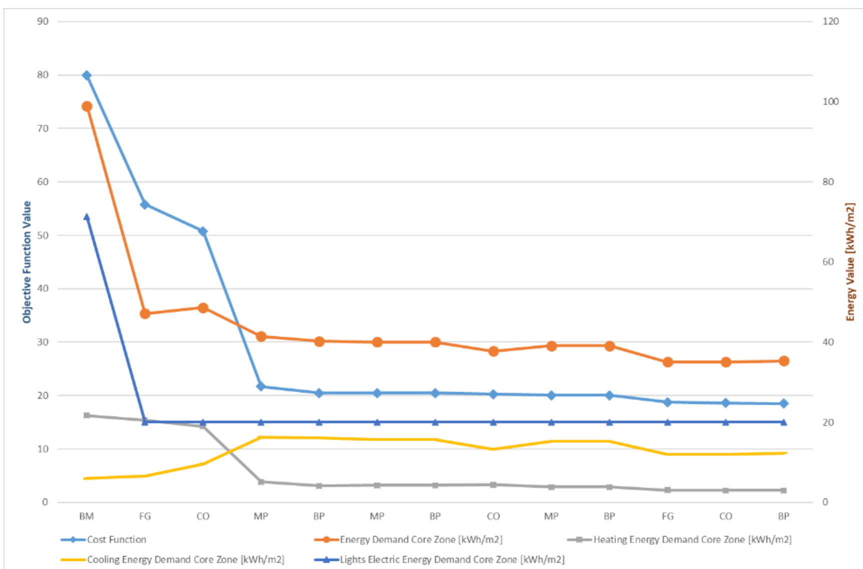


Figure 4-10 Details of the core zone energy demand evolution.

Figure 4-11 shows the south zone energy demand variation during the iterations.

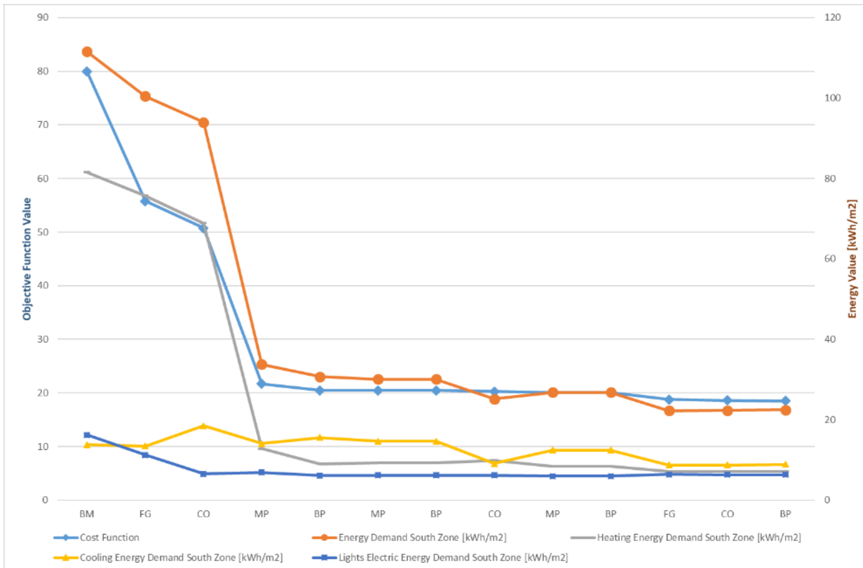


Figure 4-11 Details of the south zone energy demand evolution.

Figure 4-12 shows the east zone energy demand variation during the iterations.

Figure 4-13 shows the north zone energy demand variation during the iterations.

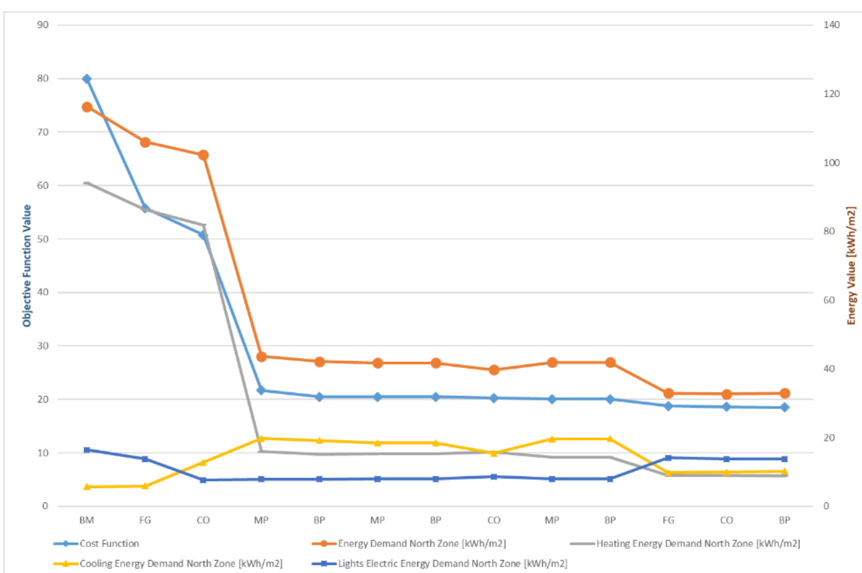


Figure 4-12 Details of the north zone energy demand evolution.

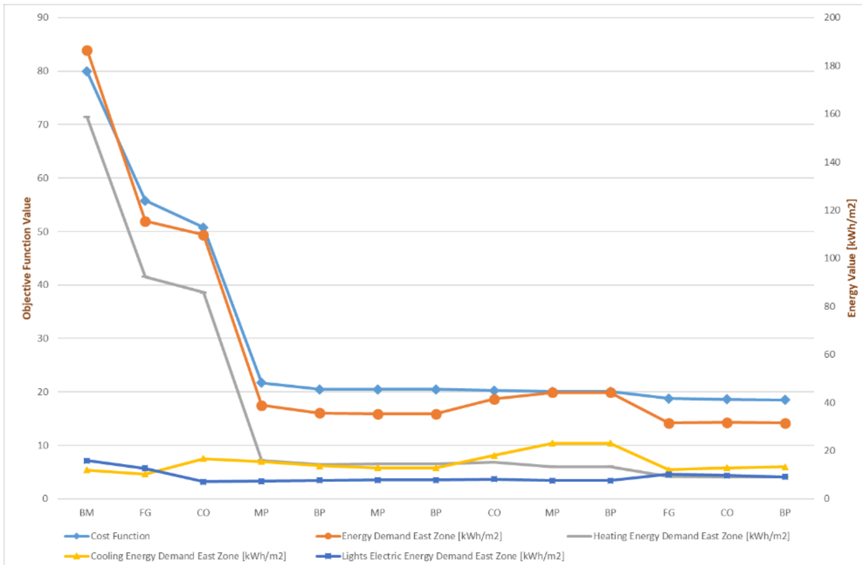


Figure 4-13 Details of the east zone energy demand evolution.

Figure 4-14 shows the west zone energy demand variation during the iterations.

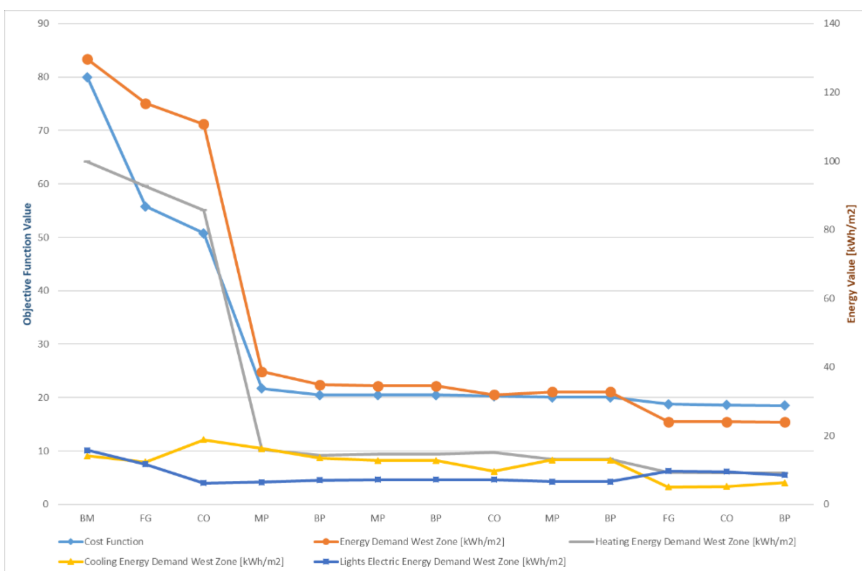


Figure 4-14 Details of the west zone energy demand evolution.

4.1.2. Example two

i. Approach

In this experiment, the model is the same, but the strategy of the iterations has been changed to a predefined sequence between the attribute clusters (Figure 4-15). The number of the iterations, clusters and variables as well as the objective function are the same as the previous example.

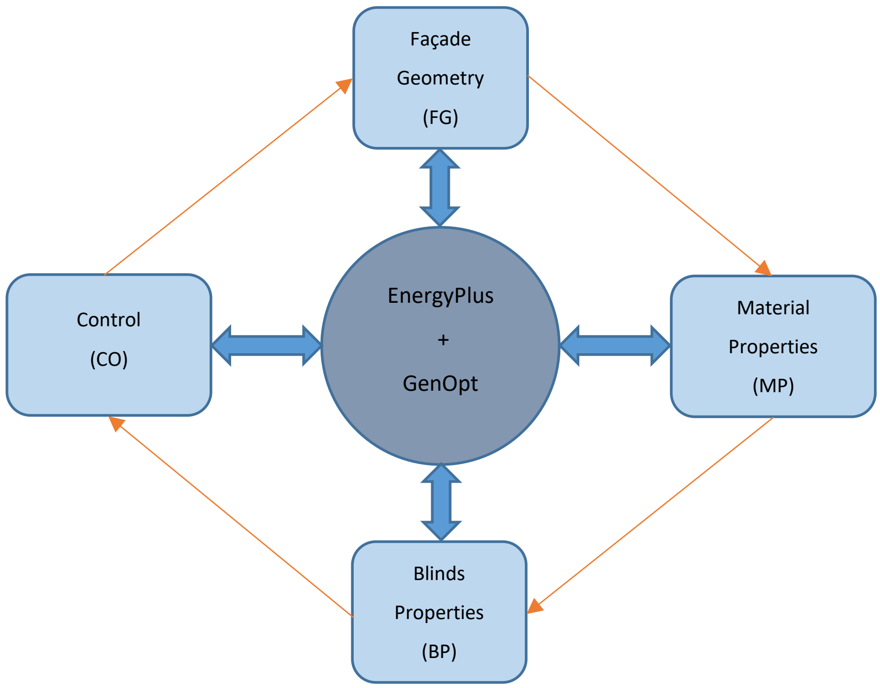


Figure 4-15 Illustration of predefined sequence between attribute clusters.

ii. Results

The results of the example come below.

Figure 4-16 demonstrates the evolution of the cost function and total energy demand during the iterations. Figure 4-17 shows the evolution of the different types of the energy demand during the optimization procedure. In the Figure 4-18, more information on the energy demands at the zones' level have been provided.

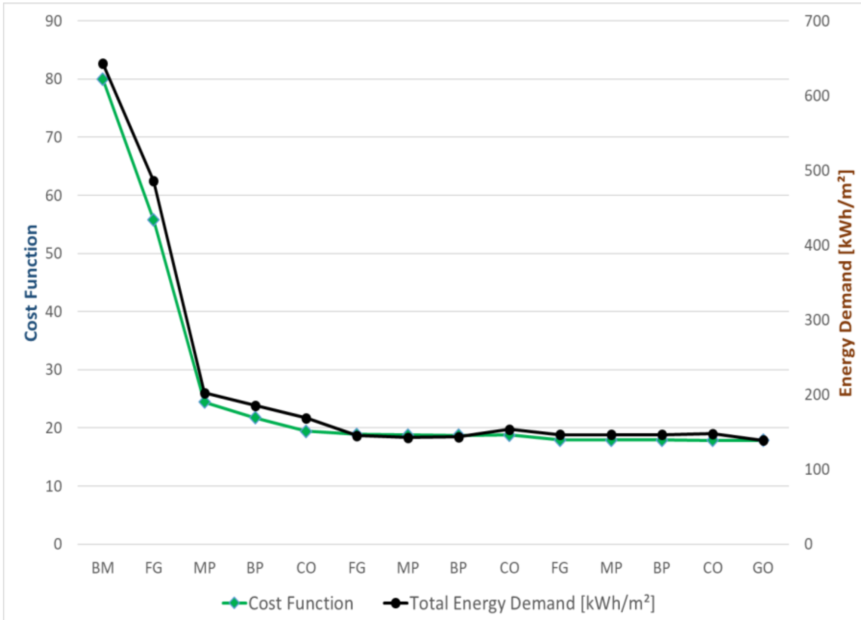


Figure 4-16 Cost function versus total energy demand.

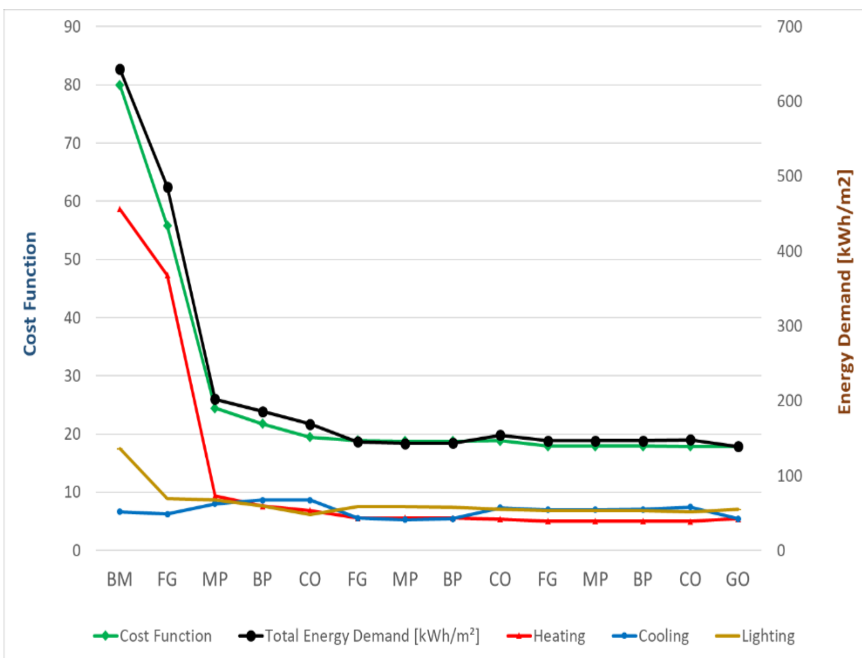


Figure 4-17 Energy demand evolution.

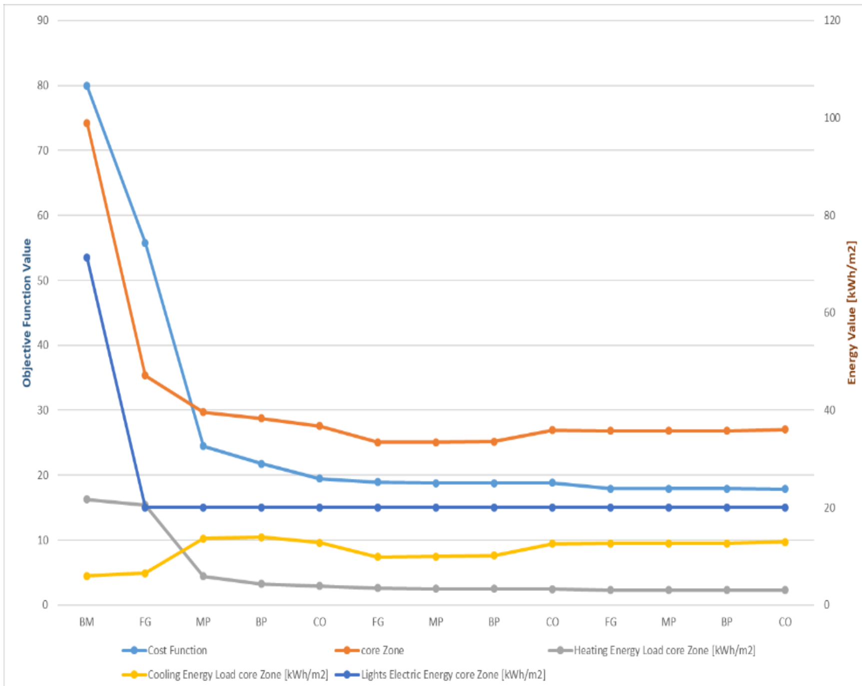


Figure 4-18 Details of the energy demand evolution at the core zone.

Figures 4-19 to 4-23 show more details of the energy demand evolution during the iterations for each zone.

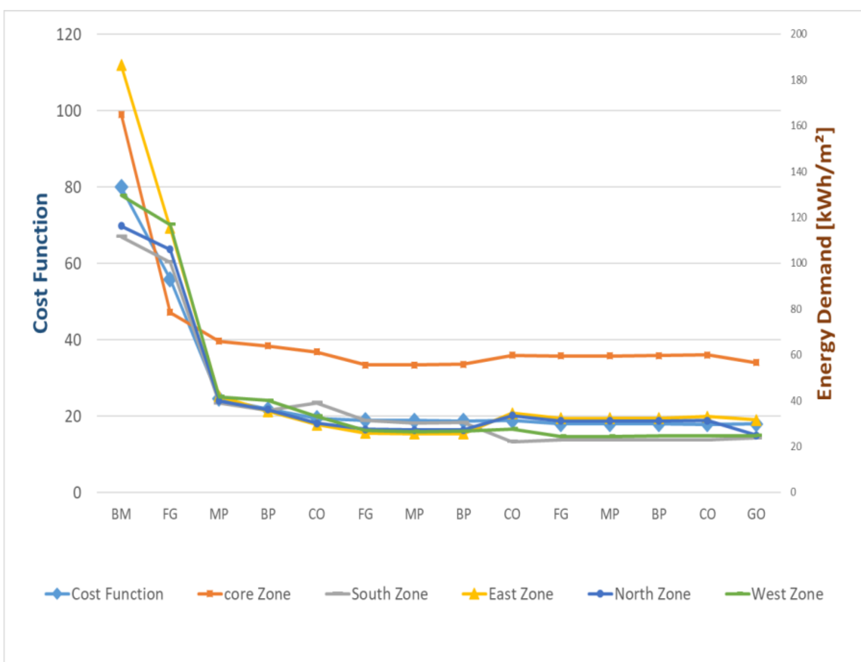


Figure 4-19 Energy demand evolution for each zone.

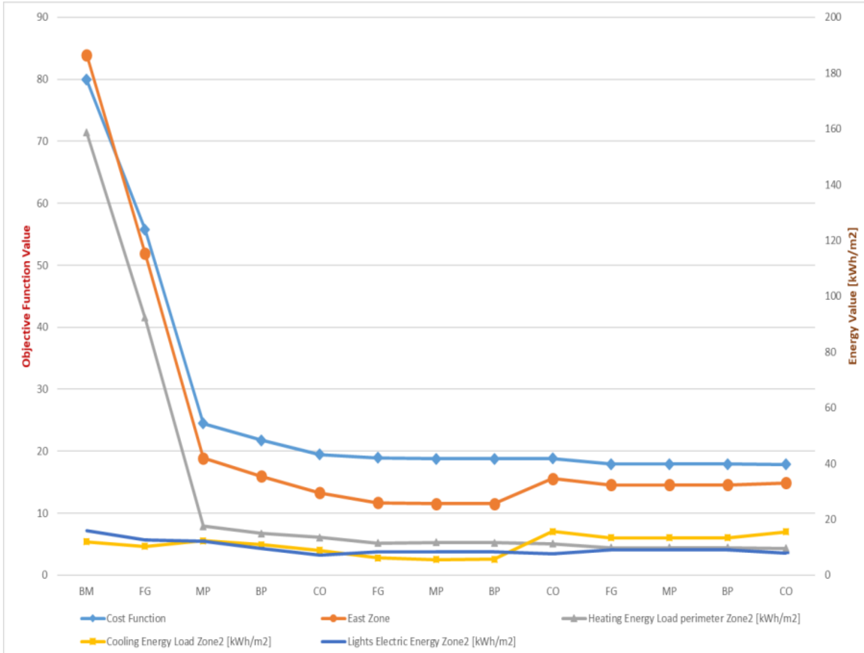


Figure 4-20 Details of the energy demand evolution at the east zone.

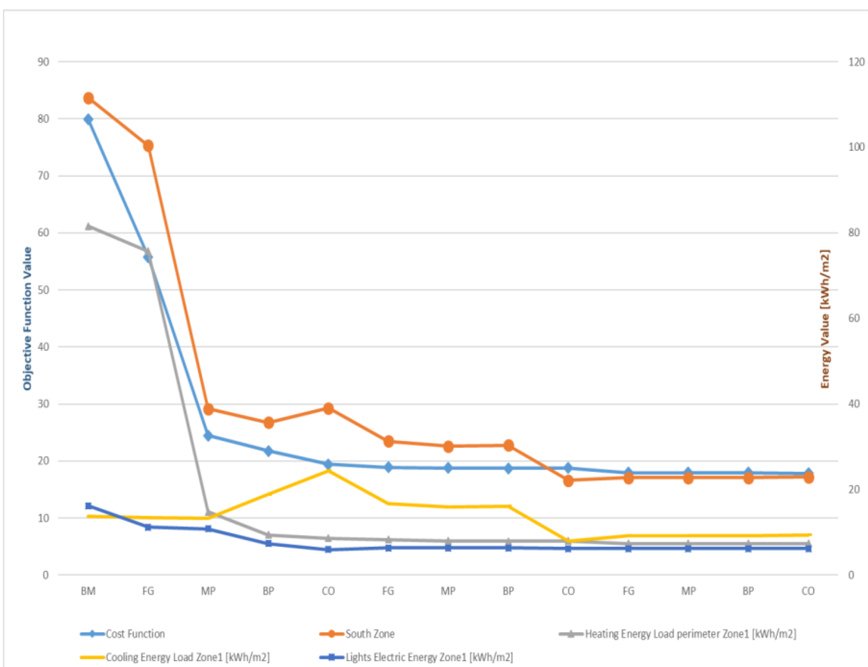


Figure 4-21 Details of the energy demand evolution at the south zone.

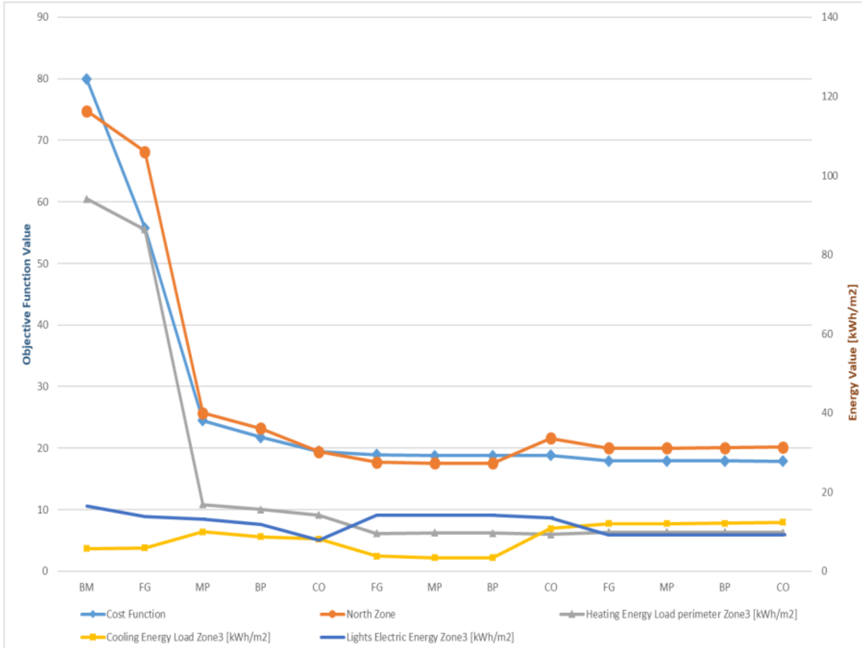


Figure 4-22 Details of the energy demand evolution at the north zone.

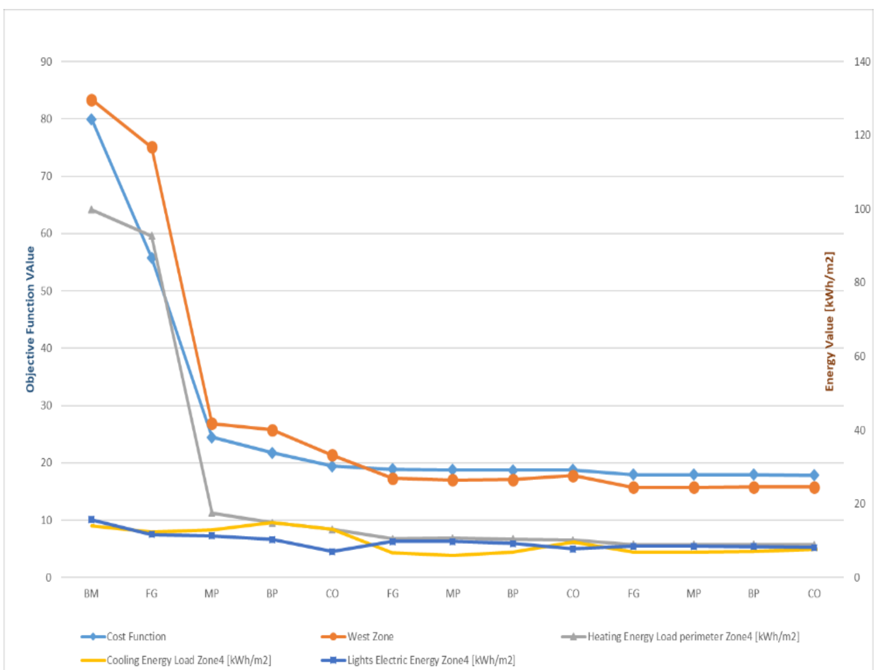


Figure 4-23 Details of the energy demand at the west zone.

4.1.3. Discussion

Both examined procedures (random and predefined cycling between clusters) display the rapid convergence toward optima, as indicated by the decrease in the value of energy use indicators as well as the cost function (see Figures 4-7 to 4-23). The bulk of optimization-based design improvement is in fact achieved during the first five iterations. Note that after some 10 iterations, no significant change in cost function values is observed. Hence, in the present demonstration, only the first 12 iterations are illustrated.

As such, the end values of the individual design variables are not necessarily identical with those in the global optimization scenario. However, after the 12th iteration, the values of the performance indicators (as well as the value of the cost function) are virtually indistinguishable from those obtained in the global optimization scenario. Interestingly, global optimization requires roughly twice as much computational time as the proposed intra-cluster cycling approach.

4.2. Examples in Rhino-Grasshopper platform

4.2.1. Example one

i. Model

The base model is an open-office building in Vienna. The building's dimensions are 20m (length), 15m (width) and 5m (height). Five thermal zones have been defined, namely one core zone and four perimeter zones (Figure 4-24).

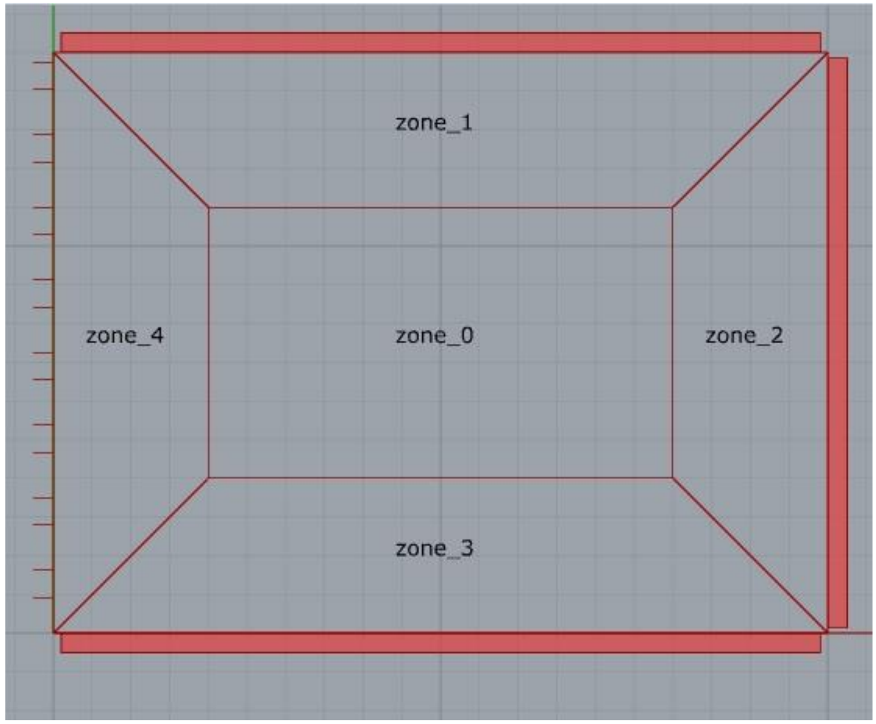


Figure 4-24 Thermal zones from the top view.

The glazing to wall ratio is 40% on the North façade, 30% on the West façade, 60% on the South façade and 50% on the East façade. For each window shading devices have been considered. Shadings on the North façade are horizontal, two on the window height; on the West façade are vertical, two on each window; on the South façade are horizontal, two on the window height; on the East façade are horizontal, two on the window height (Figure 4-25).

The construction for the opaque and glazing parts of the building's envelope have been selected from the EnergyPlus construction library in regard to the climate zone of the building.

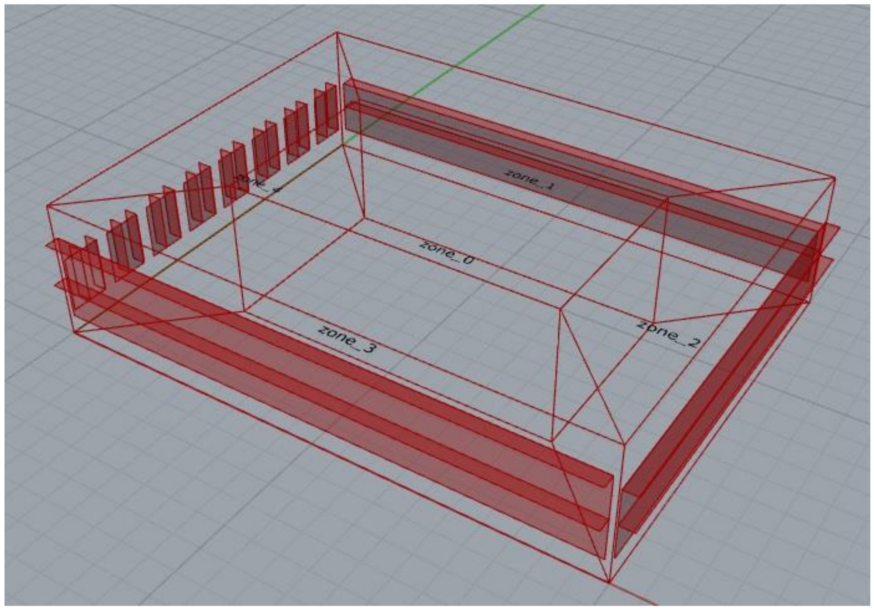


Figure 4-25 Glazing and shadings on the different facades of the building.

ii. Clusters and variables

Building's Geometry (BG)

Assuming the building's area is a fixed quantity (in this case 300m^2), the first variable of this cluster is the building's length which, can vary between 10m to 30m (Figure 4-26).

The other variable is the orientation of the building, which is the angle of the building and the north axis, and varying between -90° to 90°. This variable is a discrete variable and the values are seen in multiples of 5 (Figure 4-26).

In each façade, the window to the wall ratio varies between 10% and 90% (Figure 4-27).

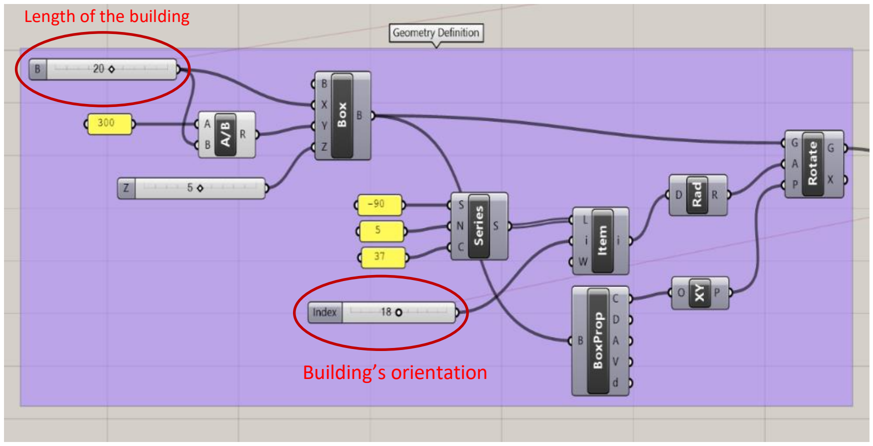


Figure 4-26 Building's footprint variables in Grasshopper.

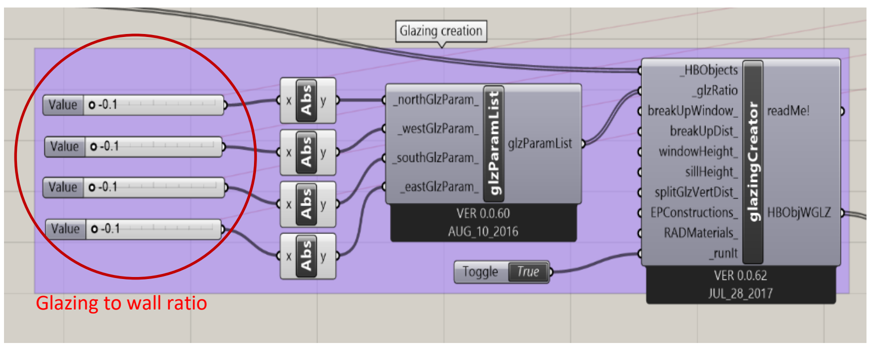


Figure 4-27 Window to wall ratio variables.

On each window it is possible to apply horizontal or vertical shading devices.

The depth of the overhangs and fins could vary between 0m to 0.60m on each façade (Figure 4-28).

The number of the fixed shading devices could be between 0 and 5 on each window (Figure 4-28).

Therefore, the total number of the independent variables in this cluster is 18.

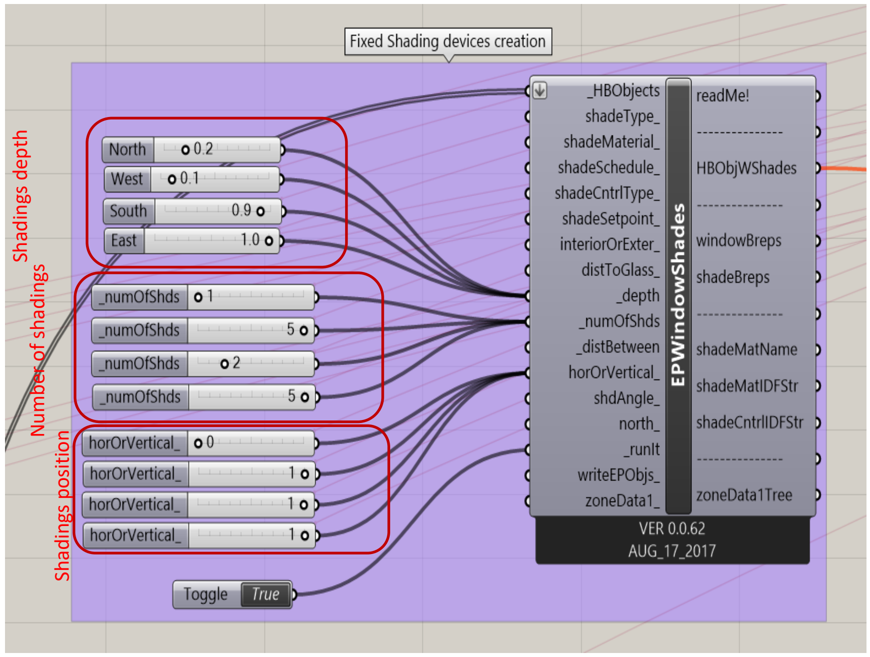


Figure 4-28 Fixed shading device variables.

Envelope Constructions (EC)

The variables of this cluster are external wall, windows, and roof constructions. All of the variables are discrete and have been extracted from the construction libraries of EnergyPlus (Figure 4-29).

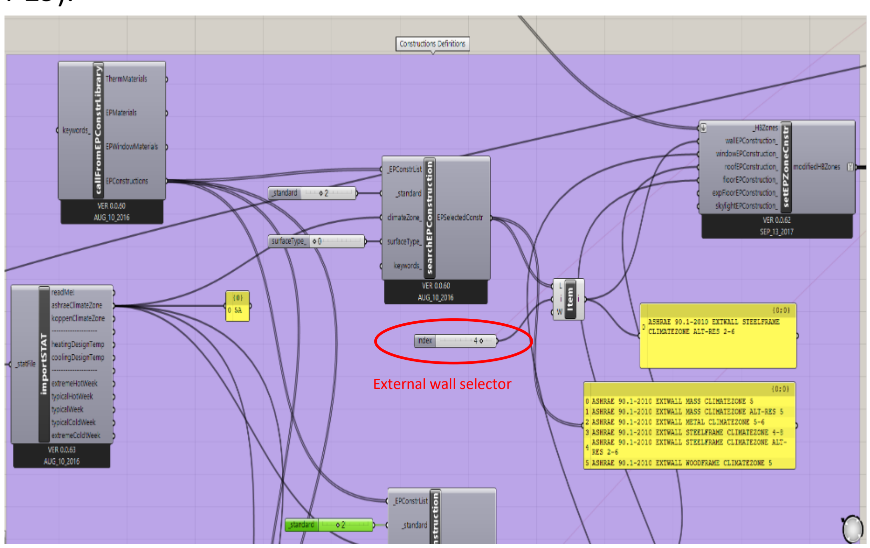


Figure 4-29 Envelope construction definition in Grasshopper.

To make the search space smaller and therefore reduce the optimization time, the construction libraries filtered by the ASHRAE climate zone of the building adheres to the EnergyPlus weather file (Table 4-2). In total, there are 3 variables in this cluster.

Table 4-2 the list of retrieved constructions from the construction library of EnergyPlus.

List of the constructions based on EnergyPlus construction library.
0 – ASHRAE 90.1-2010 EXTWALL MASS CLIMATEZONE 5 1 – ASHRAE 90.1-2010 EXTWALL MASS CLIMATEZONE ALT-RES 5 2 - ASHRAE 90.1-2010 EXTWALL METAL CLIMATEZONE 5-6 3 - ASHRAE 90.1-2010 EXTWALL STEELFRAME CLIMATEZONE 4-8 4 - ASHRAE 90.1-2010 EXTWALL STEELFRAME CLIMATEZONE ALT-RES 2-6 5 – V ASHRAE 90.1-2010 EXTWALL WOODFRAME CLIMATEZONE 5
0 - ASHRAE 90.1-2010 EXTROOF IEAD CLIMATEZONE 2-8 1 - ASHRAE 90.1-2010 EXTROOF METAL CLIMATEZONE 2-5
0 - ASHRAE 90.1-2010 EXTWINDOW METAL CLIMATEZONE 4-6 1 - ASHRAE 90.1-2010 EXTWINDOW NONMETAL CLIMATEZONE 5-6

Control (CO)

This cluster targeted the operation of the shadings and window openings during the year on each façade. To control the shadings, the angle of the shadings (varies between 0° to 90°) and the solar incidence (varies between 10 W/m^2 and 150 W/m^2) on the window have been considered (Figure 4-30). To have the optimum, natural ventilation and controlling the window operation, the outdoor temperature was considered, the threshold which above and below of them the windows stay open can vary respectively 18°C to 22°C and 22°C to 26°C (Figure 4-31). Altogether, there are 4 variables in this cluster.

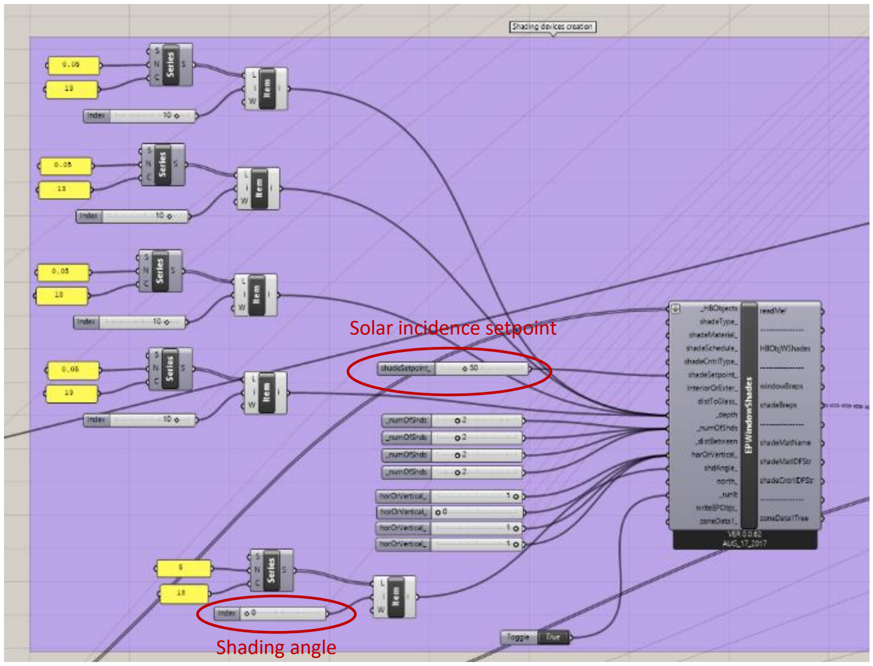


Figure 4-30 Shading control variables.

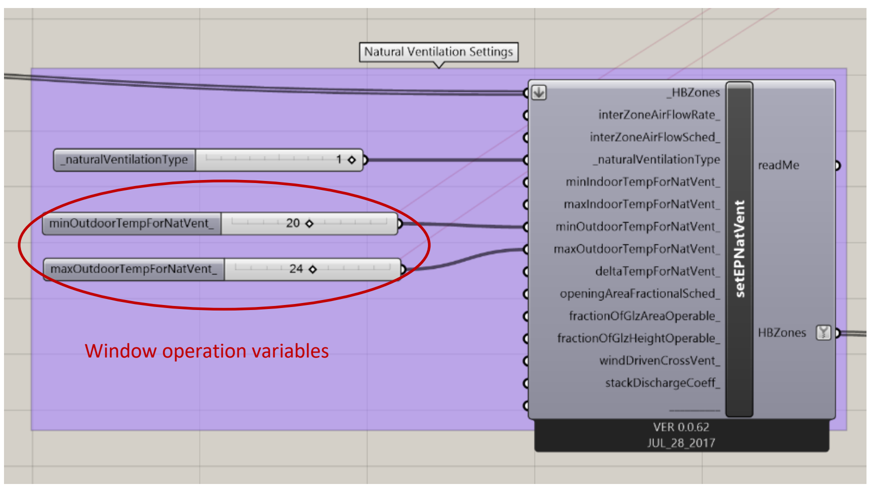


Figure 4-31 Control items of the natural ventilation.

Table 4-3 shows an overview of the clusters and variables belongs to the current example.

Table 4-3 Summary table of the clusters and variables of the current example.

Clusters	Variables	Lower band	Upper band	Units	Sum	
Building's Geometry (BG)	Building's length	10	30	m	18 variables	25 independent variables
	Building's orientation *	-90	90	°		
	Window to wall ratio	10	90	%		
	Shading device position **	-	-	-		
	Shading device depth	0	0.60	m		
	Number of shading devices	0	5	-		
Envelope Constructions (EP)	External walls ***	-	-	-	3 variables	
	Windows ***	-	-	-		
	Roofs ***	-	-	-		
Control (CO)	Incident solar irradiance	10	150	W/m^2	4 variables	
	Shading device angle	0	90	°		
	Minimum outdoor temperature	18	22	°c		
	Minimum outdoor temperature	22	26	°c		

*This is a discrete variable and the values are multiples of 5.

**This is a Boolean variable and can be horizontal or vertical.

***These variables are also discrete and the items have been shown in Table 4-2.

iii. Tools and platform

The platform has been created in Grasshopper, which is a visual programming language and is tightly integrated with Rhino. Ladybug and Honeybee are the tools used for the energy simulation and Opossum is the optimizer of the system (see 3.3).

iv. Approach

The approach of optimization iterations is a predefined cycling between attribute clusters (Figure 4-32).

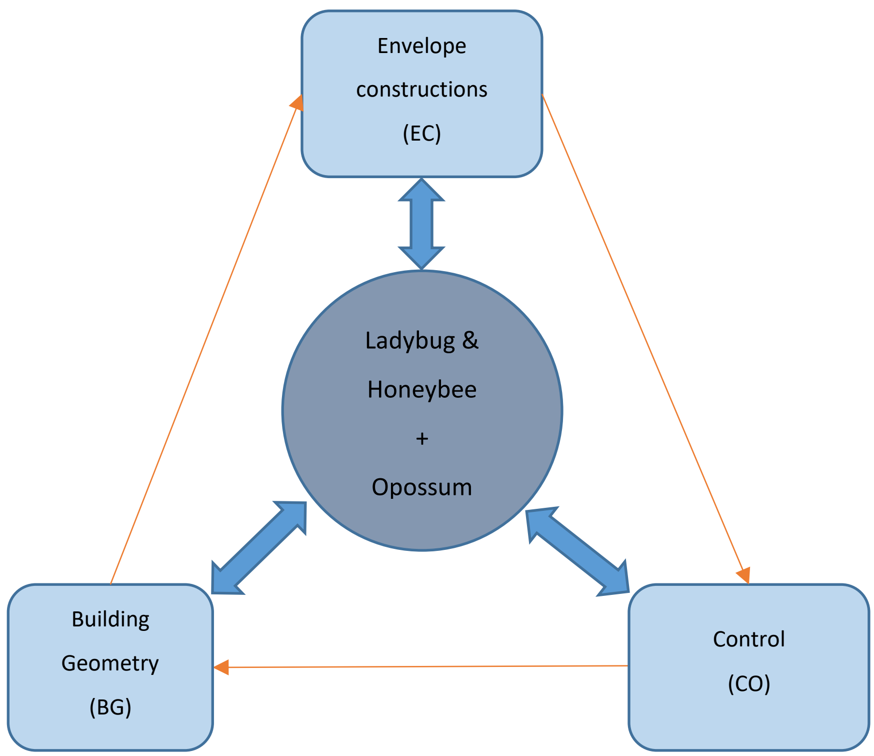


Figure 4-32 Illustration of predefined cycling between attribute clusters in Grasshopper.

v. Performance indicators and cost function

Performance indicators are the same as they were for the previous examples (see Performance indicators and cost function). Based on the previously mentioned indicators, the cost function (U) is as it is shown in Equation 4-2:

Equation 4-2 The cost function of the optimizations in Rhino-Grasshopper.

$$U = \left(\frac{H}{0.85}\right) + \left(\frac{C}{3.5} \times 1.8\right) + (L \times 1.8)$$

Where:

H is the normalized heating demand of the whole building.

C is the normalized cooling demand of the whole building.

L is the normalized electric light demand of the whole building.

vi. Results

This section includes results (and variable values and their evolution) from optimization iterations between the clusters. Thereby, abbreviations BM and GO refer to "Base Model" and "Global Optimization" respectively. Figure 4-33 illustrates the evolution of the cost function together with the total energy demand in the course of 6 iterations between attribute clusters. Respectively, in Table 4-4 related values are shown.

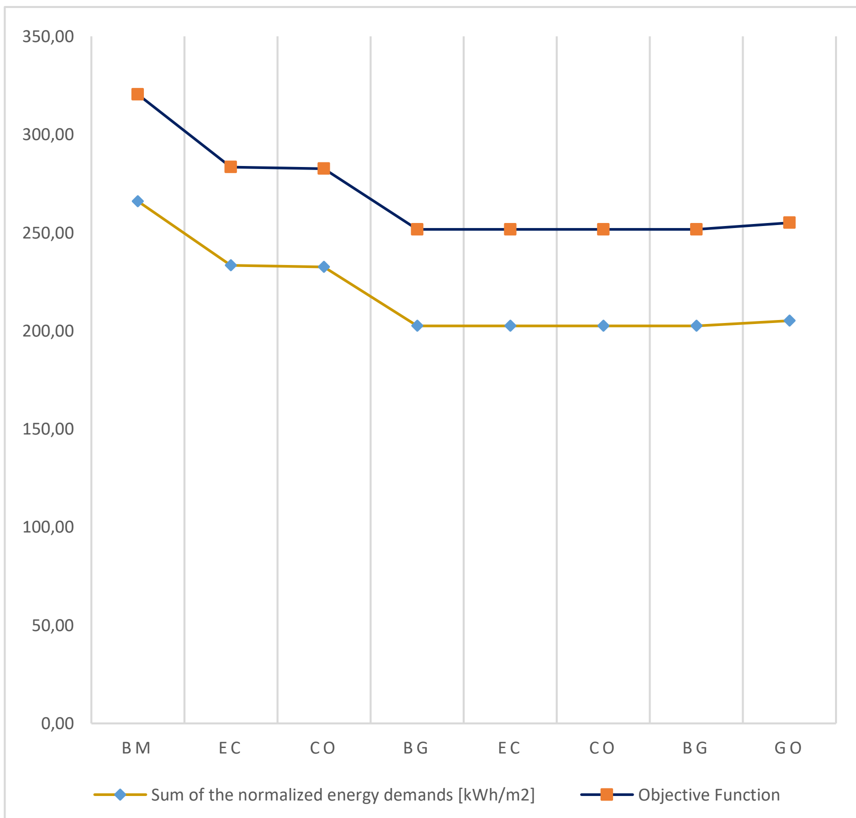


Figure 4-33 Objective function versus Energy demand.

Table 4-4 Related values of Figure 4-33.

	BM	EC	CO	BG	EC	CO	BG	GO
Sum of the normalized energy demands [kWh/m2]	265.98	233.37	232.51	202.48	202.48	202.48	202.48	205.06
Objective Function	320.41	283.34	282.60	251.65	251.65	251.65	251.65	255.05

Figure 4-34 shows the evolution of cooling, heating, and electric light energy demand normalized per area of the building as well as the total energy demand and objective function for the same iterations. The related values are demonstrated in Table 4-5.

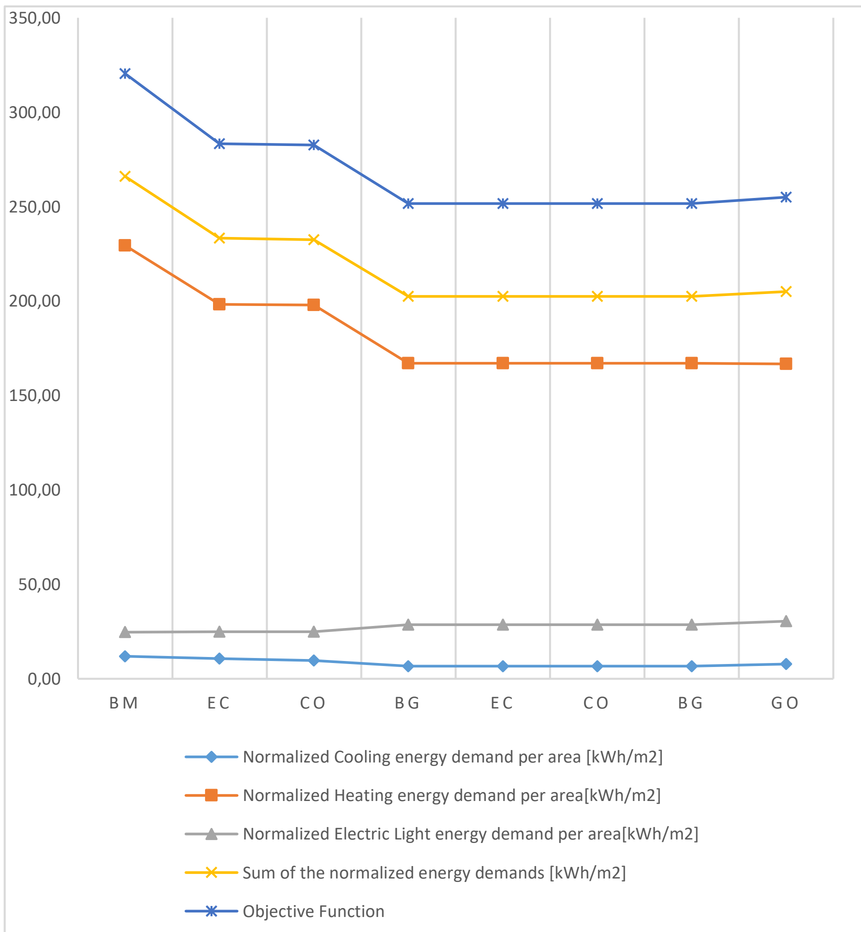


Figure 4-34 Energy demand evolution.

Table 4-5 Values of Figure 4-34.

	BM	EC	CO	BG	EC	CO	BG	GO
Normalized Cooling energy demand per area [kWh/m2]	11.92	10.65	9.72	6.70	6.70	6.70	6.70	7.87
Normalized Heating energy demand per area[kWh/m2]	229.39	198.17	197.95	167.11	167.11	167.11	167.11	166.71
Normalized Electric Light energy demand per area[kWh/m2]	24.67	24.84	24.84	28.67	28.67	28.67	28.67	30.49
Sum of the normalized energy demands [kWh/m2]	265.98	233.37	232.51	202.48	202.48	202.48	202.48	205.06
Objective Function	320.41	283.34	282.60	251.65	251.65	251.65	251.65	255.05

By looking at the chart in Figure 4-35, the designer is able to study the cooling energy demand evolution in each zone. Respectively, the numerical values of the Figure 4-35 are demonstrated in Table 4-6.

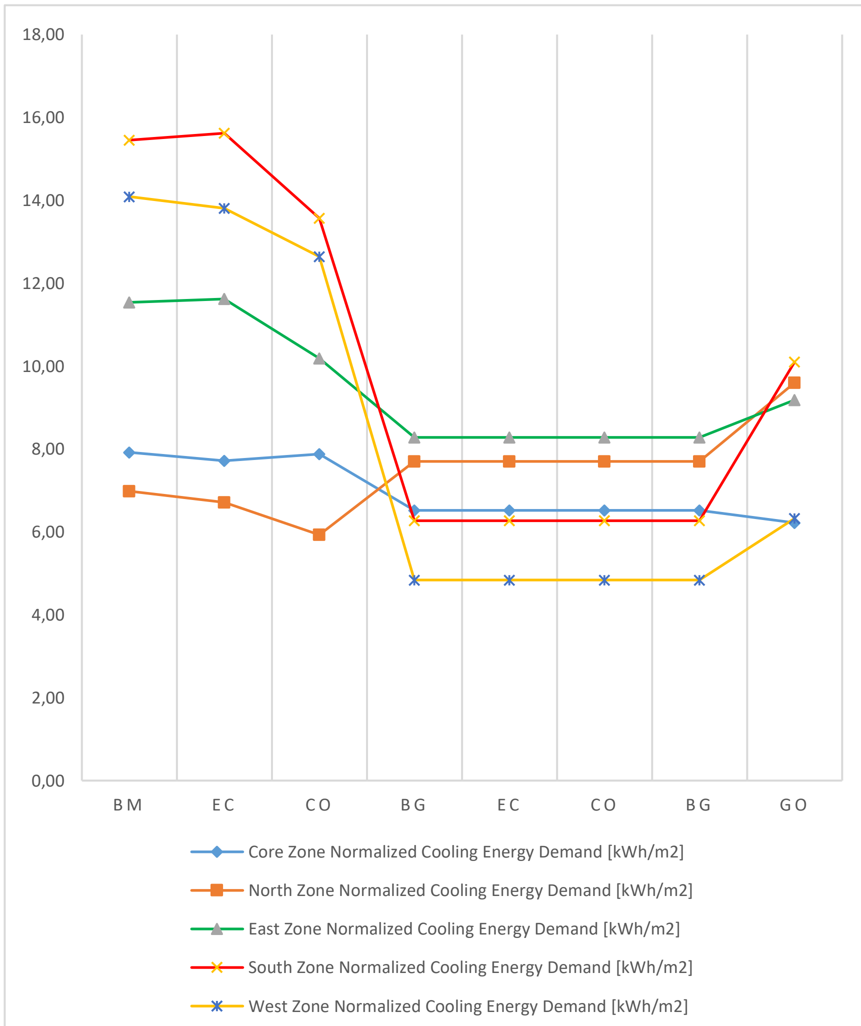


Figure 4-35 Zones cooling energy demand evolution.

Table 4-6 Numerical values of Figure 4-35.

	BM	EC	CO	BG	EC	CO	BG	GO
Core Zone Normalized Cooling Energy Demand [kWh/m2]	7.92	7.72	7.88	6.52	6.52	6.52	6.52	6.22
North Zone Normalized Cooling Energy Demand [kWh/m2]	6.98	6.71	5.93	7.70	7.70	7.70	7.70	9.60
East Zone Normalized Cooling Energy Demand [kWh/m2]	11.54	11.62	10.19	8.28	8.28	8.28	8.28	9.18
South Zone Normalized Cooling Energy Demand [kWh/m2]	15.45	15.62	13.57	6.27	6.27	6.27	6.27	10.10
West Zone Normalized Cooling Energy Demand [kWh/m2]	14.09	13.81	12.64	4.84	4.84	4.84	4.84	6.33

Figure 4-36 demonstrates the heating energy demand variation per zone during the optimization iterations between the attribute clusters. In Table 4-7 related values are shown.

Figure 4-36 Zones heating energy demand evolution.

Table 4-7 Numerical values of Figure 4-36.

	BM	EC	CO	BG	EC	CO	BG	GO
Core Zone Normalized Heating Energy Demand [kWh/m2]	153.68	143.39	143.23	137.90	137.90	137.90	137.90	150.13
North Zone Normalized Heating Energy Demand [kWh/m2]	260.02	223.95	223.72	176.83	176.83	176.83	176.83	191.43
East Zone Normalized Heating Energy Demand [kWh/m2]	273.91	229.19	228.98	173.65	173.65	173.65	173.65	165.91
South Zone Normalized Heating Energy Demand [kWh/m2]	260.58	215.72	215.47	182.54	182.54	182.54	182.54	192.85
West Zone Normalized Heating Energy Demand [kWh/m2]	242.66	208.69	208.45	183.04	183.04	183.04	183.04	162.78

In Figure 4-37 the electric light energy demand during the optimization clusters have been illustrated according to each zone. Following, in Table 4-8 the numerical values related to the chart are included.

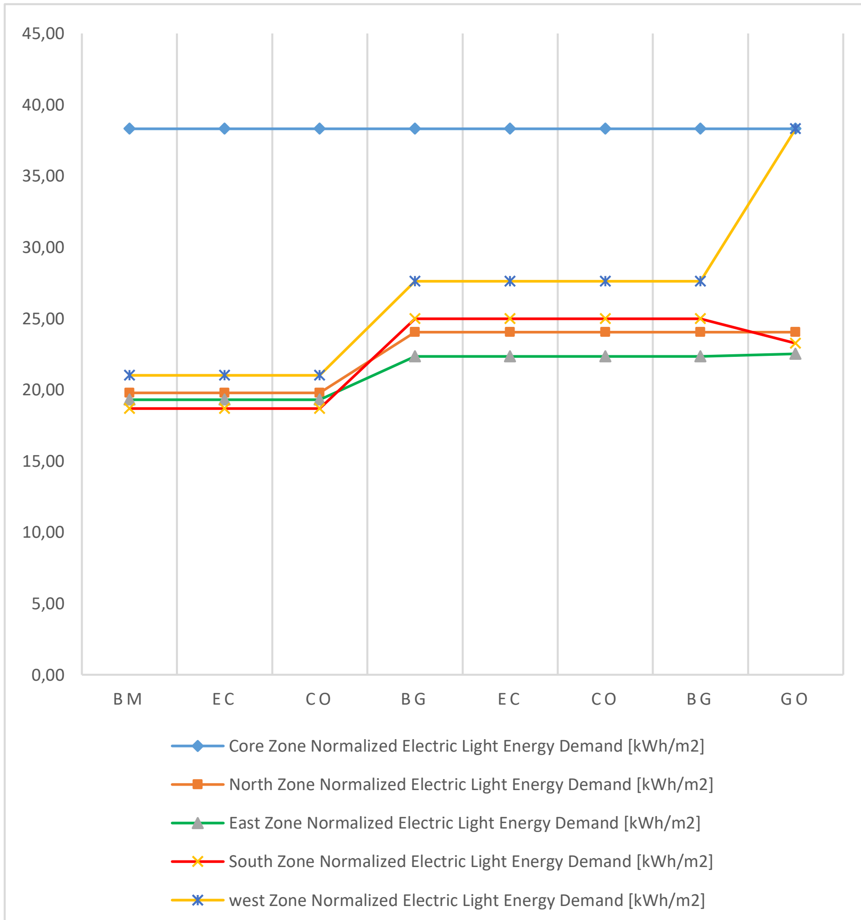


Figure 4-37 Zones electric light energy demand evolution.

Table 4-8 Numerical values of Figure 4-37.

	BM	EC	CO	BG	EC	CO	BG	GO
Core Zone Normalized Electric Light Energy Demand [kWh/m2]	38.31	38.31	38.31	38.31	38.31	38.31	38.31	38.31
North Zone Normalized Electric Light Energy Demand [kWh/m2]	19.78	19.78	19.78	24.04	24.04	24.04	24.04	24.04
East Zone Normalized Electric Light Energy Demand [kWh/m2]	19.30	19.30	19.30	22.33	22.33	22.33	22.33	22.52
South Zone Normalized Electric Light Energy Demand [kWh/m2]	18.68	18.68	18.68	24.98	24.98	24.98	24.98	23.25
west Zone Normalized Electric Light Energy Demand [kWh/m2]	21.01	21.01	21.01	27.61	27.61	27.61	27.61	38.31

In Table 4-9 the evolution of the variable values during the optimization iterations are demonstrated.

Table 4-9 Variables values evolution in the current experiment.

	BM	EC	CO	BG	EC	CO	BG	GO
Building Length [m]	20			17			17	10
Building Orientation [degree from the North]	0			65			65	90
Window to wall ratio, North [%]	40			10			10	10
Window to wall ratio, East [%]	30			10			10	10
Window to wall ratio, South [%]	60			10			10	10
Window to wall ratio, West [%]	50			10			10	10
Position of the Shading, North (Horizontal/Vertical)	0.5			0.45			0.45	0.6
Position of the Shading, East (Horizontal/Vertical)	0.5			0			0	0
Position of the Shading, South (Horizontal/Vertical)	0.5			0			0	0
Position of the Shading, West (Horizontal/Vertical)	0.5			0			0	0
Shading slat length, North [m]	2			1			1	5
Shading slat length, East [m]	2			5			4	1
Shading slat length, South [m]	2			1			5	5
Shading slat length, West [m]	2			5			1	1
Number of the Shadings on the Windows, North	H			H			H	V
Number of the Shadings on the Windows, East	V			V			V	H
Number of the Shadings on the Windows, South	H			H			H	H
Number of the Fixed Shadings on the Windows West	H			H			H	H
External Wall Construction	2	5			3			5
Roof Construction	1	0			0			0
Windows Construction	0	1			1			1
Minumum Outdoor Temperature for Natural Ventilation [°C]	20		21			21		22
Maximum Outdoor Temperature for Natural Ventilation [°C]	24		26			26		22
Solar Incidence Set point for shading control [w/m2]	50		31			104		131
Shading Angle on the window [°]	0		0			0		0

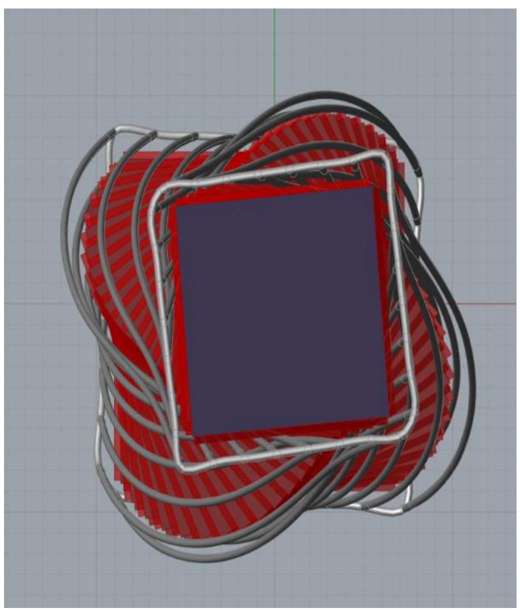
vii. Discussion

The examined model above displayed a rapid convergence toward optima, as indicated by a decrease in the value of the cost function (Figure 4-33). The bulk of optimization-based design improvement is in fact achieved during the first three iterations. Note that after the third iteration, no change in cost function and (no significant change in) variable values is observed. As such, the end values of the individual design variables are not necessarily identical with those in the global optimization scenario. Interestingly, as it was shown in the graphs and tables, the proposed intra-cluster cycling approach resulted in a better design solution in connection to achieving the minimum value of the cost function comparing the one-shot global optimization. The transparency of the current approach enables the designer to study the effect of each cluster of the variables on the objective of the optimizations separately and therefore achieve a more suitable strategy for the architectural design. The details of the resulted data also help to consider every single variable during the iterations and change the categories if deemed necessary.

4.2.2. Example Two

i. Model

The base model of the second case study is a tower office building in Vienna consisting of 20 floors (Figure 4-38 and 44-39). Each floor



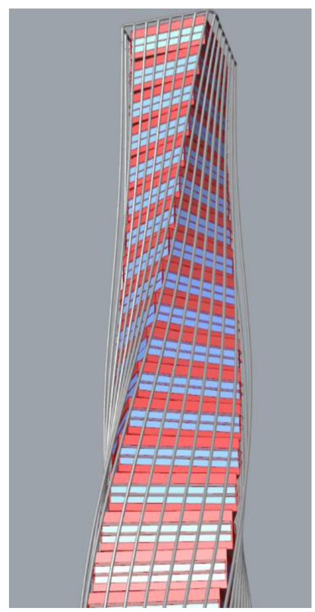


Figure 4-38 Top view and perspective of the model in Rhino.

is an open office and a thermal zone. Therefore, the model has 20 thermal zones. Each floor has a four-sided polygon shape. The radius of the peripheral circle of the first-floor polygon (the distance from the center to the tip) is 5 meters. The height of each floor is 4 meters. The scale of the floors changes in a sinusoid pattern from the bottom to the top of the tower. Each floor rotates 5° more counterclockwise than the floor below.

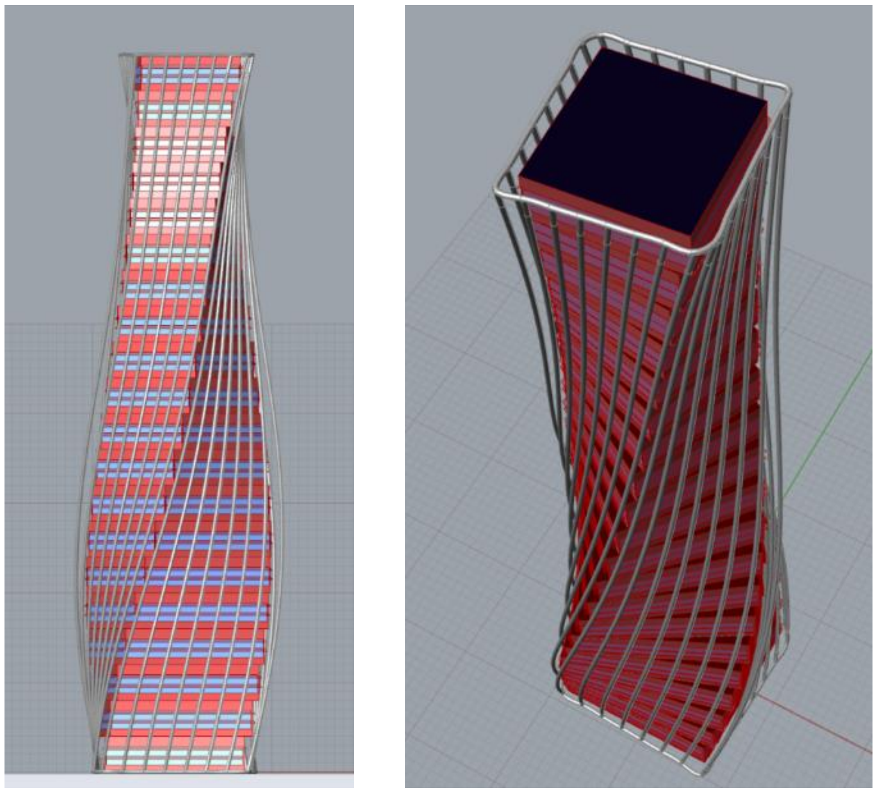


Figure 4-39 Front view and bird view perspective of the model.

There is a pipe-wire skin with the distance of 1 meter from the building's envelope. Fifty percent of each façade (of each zone) is the glazing part. Half of the glazing area is operable on each side. There are two horizontal shading devices on each window. Each shading device has 0.5 meter depth with 5 centimeter thickness, while the angle of the slats is 45° (Figure 4-40). The material of each shading device has 50% reflectance and 80% emissivity. Constructions of the opaque and glazing parts have been retrieved from the construction library of EnergyPlus based on the ASHRAE climate zone of Vienna. Natural ventilation of each zone is controlled based on the outside air temperature. And the shading devices are controlled based on the solar incidence on the windows.

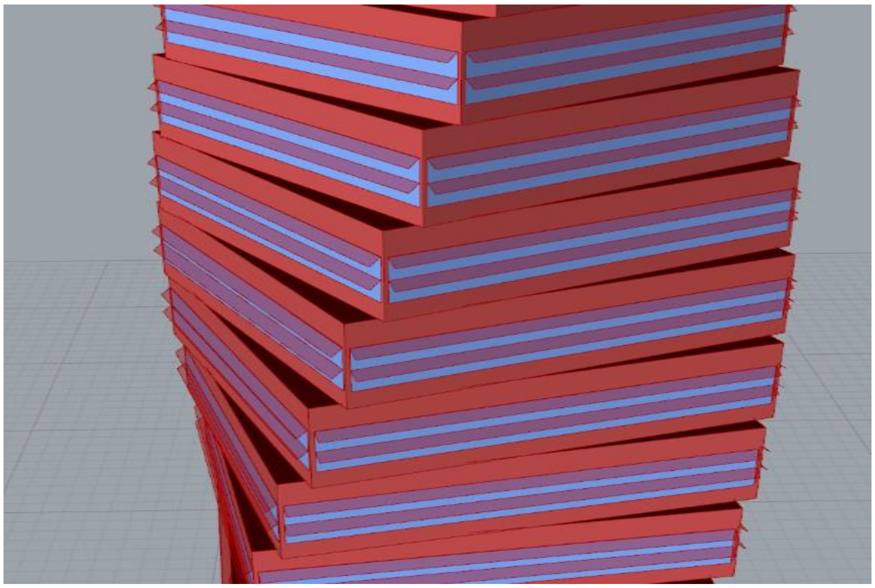


Figure 4-40 Closer view of the glazing and shading devices on the base model.

ii. Clusters and variables

Building's geometry (BG)

This cluster includes the footprint's geometry of the tower, which is the number of the polygon's edges, and varies from 3 to 6. The rotation of the successive floors can vary between 0° to 10° (Figure 4-41).

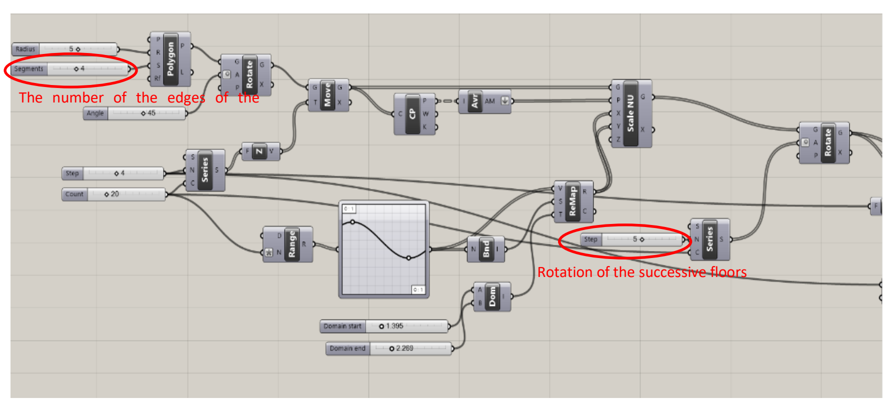


Figure 4-41 New geometry variables in this case study.

The window to wall ratio of each façade of each zone is another variable and varies between 10% and 90%. The length of the shading slat varies from 0 to 0.6 meter. The number of the shadings on each window can be between 0 to 5. The horizontal or vertical position of the shadings was also considered as a variable. Overall, this cluster has 6 independent variables.

Material properties and constructions (MC)

The reflectance and emissivity of the shadings (each varying between 5% and 95%) (Figure 4-42), along with external walls, roof, and window constructions make up 5 variables of this cluster. The source for selecting the constructions is the same as the example one (Table 4-2).

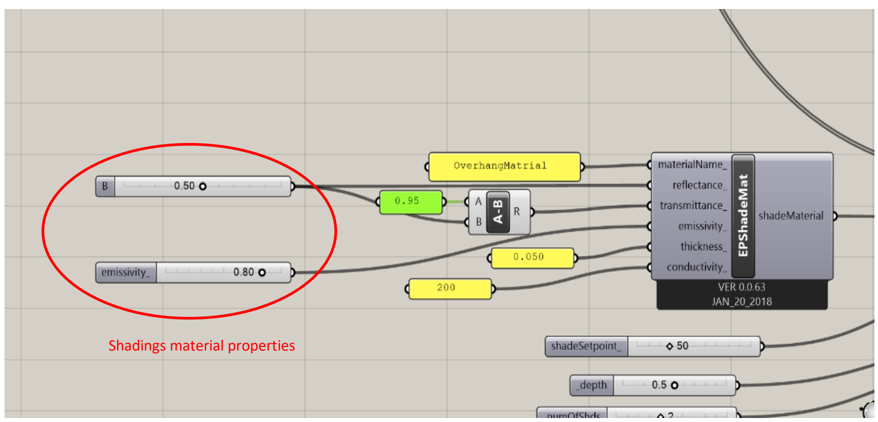


Figure 4-42 Material properties variables.

Control (CO)

The last cluster is concerned with the operation of window openings and shadings during the year. For shadings, the variables subjected to optimization are the threshold value for incident solar irradiance, above which the shadings are on, and the angle of the shading slats. Regarding ventilation, a base mechanical ventilation system is assumed that delivers a prescribed fresh air supply. However, given appropriate conditions (expressed in terms of an outdoor ambient temperature band), an additional magnitude of fresh air flow via window operation is supplied. The variables subjected to optimization are, in this case, the threshold values for outdoor temperature below which and above which windows are closed. This cluster includes thus 4 variables.

The platform and pattern of the attribute clusters iterations are the same as the previous example (Figure 4-32).

Also, the performance indicators and cost function are the ones which have been used for the example one (Equation 4-2).

Table 4-10 shows a summary of the clusters and variables belongs to the example two implemented in the Rhino-Grasshopper platform.

Table 4-10 An overview of the clusters and variables of the example two.

Clusters	Variables	Lower band	Upper band	Units	Sum	
Building's Geometry (BG)	Number of polygon edges	3	6	-	6 variables	15 independent variables
	rotation of the successive floors	0	10	°		
	Window to wall ratio	10	90	%		
	Shading device position *	-	-	-		
	Shading device depth	0	0.60	m		
	Number of shading devices	0	5	-		
Material properties and Constructions (MC)	External walls **	-	-	-	5 variables	
	Windows **	-	-	-		
	Roofs **	-	-	-		
	Shading slat reflectance	5	95	%		
	Shading slat emissivity	5	95	%		
Control (CO)	Incident solar irradiance	10	150	W/m^2	4 variables	
	Shading device angle	0	90	°		
	Minimum outdoor temperature	18	22	°c		
	Minimum outdoor temperature	22	26	°c		

*This is a Boolean variable and can be horizontal or vertical.

**These variables are discrete and the items have been shown in Table 4-2.

iii. Results

This section includes results from optimization via cycling between the attribute clusters. Figure 4-43 illustrates the evolution of the cost function together with the total energy demand during 6 iterations between attribute clusters. Following that, Table 4-11 contains the related numerical values of it.

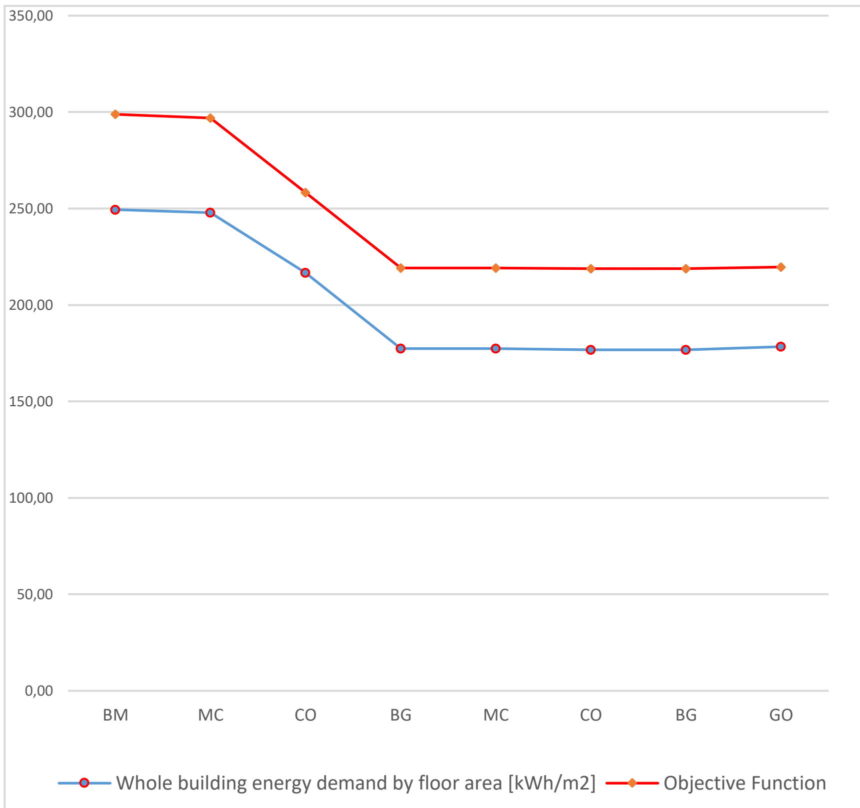


Figure 4-43 Cost function versus total normalized energy demand.

Table 4-11 Numerical values of Figure 4-43.

	BM	MC	CO	BG	MC	CO	BG	GO
Whole building energy demand by floor area [kWh/m2]	249.43	247.79	216.72	177.36	177.36	176.76	176.76	178.43
Objective Function	298.85	296.87	258.36	219.13	219.13	218.88	218.86	219.66

Figure 4-44 shows the energy demand (heating, lighting, cooling, and total) evolution for the same iterations. Following that Table 4-12 shows the related numerical values.

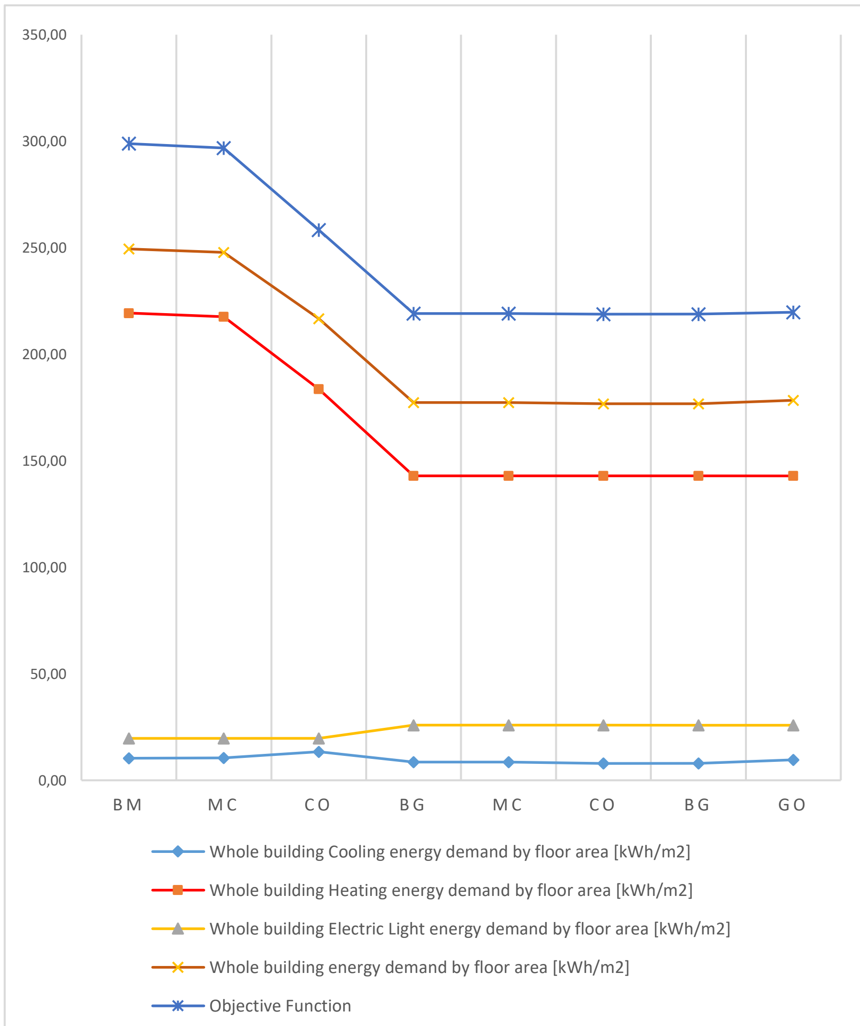


Figure 4-44 Cost function and energy demand evolution.

Table 4-12 Related numerical values of Figure 4-44.

	BM	MC	CO	BG	MC	CO	BG	GO
Whole building Cooling energy demand by floor area [kWh/m2]	10.39	10.50	13.41	8.58	8.58	7.90	7.91	9.66
Whole building Heating energy demand by floor area [kWh/m2]	219.33	217.57	183.62	142.88	142.88	142.96	142.96	142.87
Whole building Electric Light energy demand by floor area [kWh/m2]	19.70	19.73	19.69	25.90	25.90	25.90	25.89	25.89
Whole building energy demand by floor area [kWh/m2]	249.43	247.79	216.72	177.36	177.36	176.76	176.76	178.43
Objective Function	298.85	296.87	258.36	219.13	219.13	218.88	218.86	219.66

Figure 4-45 demonstrates the evolution of the normalized cooling energy demand per area for each zone. Table 4-13 shows the related numerical values respectively.

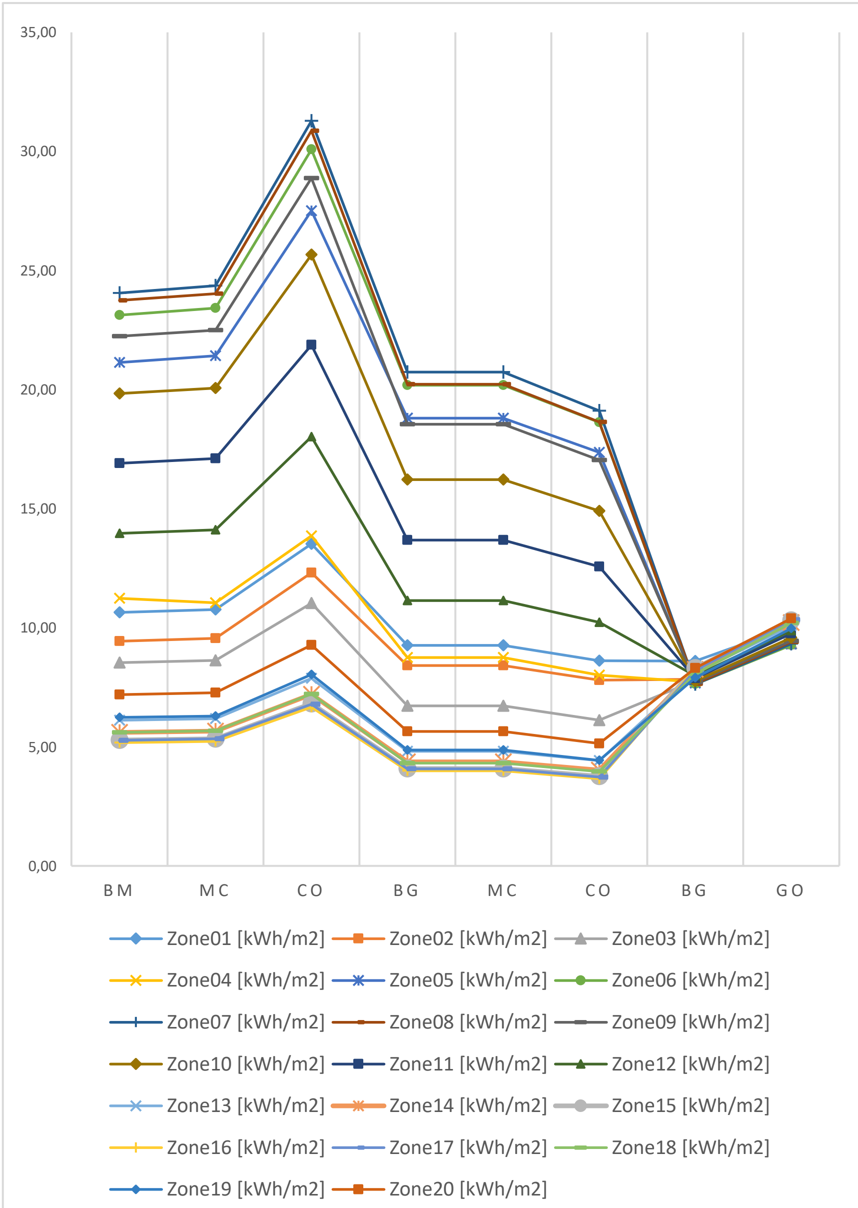


Figure 4-45 Zones normalized cooling energy demand evolution during the optimizations iterations.

Table 4-13 Related numerical values of Figure 4-45.

	BM	MC	CO	BG	MC	CO	BG	GO
Zone01 [kWh/m2]	10.65	10.77	13.52	9.26	9.26	8.61	8.60	10.04
Zone02 [kWh/m2]	9.44	9.56	12.31	8.41	8.41	7.80	7.83	9.31
Zone03 [kWh/m2]	8.53	8.63	11.03	6.72	6.72	6.12	7.78	9.34
Zone04 [kWh/m2]	11.24	11.05	13.86	8.75	8.75	8.01	7.72	9.33
Zone05 [kWh/m2]	21.14	21.42	27.51	18.80	18.80	17.36	7.67	9.31
Zone06 [kWh/m2]	23.13	23.42	30.08	20.18	20.18	18.62	7.63	9.27
Zone07 [kWh/m2]	24.06	24.36	31.28	20.74	20.74	19.11	7.63	9.33
Zone08 [kWh/m2]	23.74	24.03	30.87	20.22	20.22	18.64	7.65	9.41
Zone09 [kWh/m2]	22.24	22.50	28.88	18.54	18.54	17.04	7.72	9.45
Zone10 [kWh/m2]	19.83	20.06	25.66	16.22	16.22	14.90	7.76	9.58
Zone11 [kWh/m2]	16.91	17.10	21.87	13.68	13.68	12.56	7.88	9.76
Zone12 [kWh/m2]	13.96	14.11	18.02	11.14	11.14	10.23	7.98	9.88
Zone13 [kWh/m2]	6.11	6.18	7.87	4.81	4.81	4.42	8.11	10.09
Zone14 [kWh/m2]	5.60	5.66	7.21	4.37	4.37	4.02	8.25	10.22
Zone15 [kWh/m2]	5.28	5.34	6.81	4.09	4.09	3.76	8.31	10.31
Zone16 [kWh/m2]	5.17	5.23	6.66	3.99	3.99	3.66	8.28	10.32
Zone17 [kWh/m2]	5.27	5.33	6.79	4.07	4.07	3.73	8.21	10.33
Zone18 [kWh/m2]	5.61	5.67	7.22	4.32	4.32	3.96	8.07	10.20
Zone19 [kWh/m2]	6.23	6.29	8.04	4.86	4.86	4.43	7.90	9.97
Zone20 [kWh/m2]	7.19	7.27	9.27	5.64	5.64	5.14	8.31	10.39

Table 4-14 shows the numerical values of the normalized heating energy demand variation during the optimization iterations for each zone which have been illustrated in a graph, shown in Figure 4-46.

Table 4-14 Related numerical values of Figure 4-46.

	BM	MC	CO	BG	MC	CO	BG	GO
Zone01 [kWh/m2]	187.01	185.51	172.73	138.49	138.49	138.54	138.58	138.51
Zone02 [kWh/m2]	190.92	189.41	170.47	136.45	136.45	136.53	136.52	136.43
Zone03 [kWh/m2]	200.17	198.46	157.46	118.32	118.32	118.39	136.16	136.11
Zone04 [kWh/m2]	263.49	258.90	204.43	158.49	158.49	158.58	136.38	136.22
Zone05 [kWh/m2]	429.85	426.69	379.26	303.51	303.51	303.69	136.87	136.72
Zone06 [kWh/m2]	474.48	470.99	413.08	329.96	329.96	330.16	137.61	137.56
Zone07 [kWh/m2]	495.26	491.66	429.16	341.97	341.97	342.17	138.76	138.60
Zone08 [kWh/m2]	494.91	491.4	422.74	335.57	335.57	335.78	140.07	139.95
Zone09 [kWh/m2]	466.25	462.94	394.44	311.39	311.39	311.58	141.72	141.57
Zone10 [kWh/m2]	417.76	414.75	349.82	274.5	274.50	274.67	143.57	143.51
Zone11 [kWh/m2]	356.31	353.70	297.27	231.40	231.40	231.55	145.65	145.58
Zone12 [kWh/m2]	296.68	294.44	244.79	188.98	188.98	189.10	147.77	147.71
Zone13 [kWh/m2]	130.95	129.98	106.83	81.71	81.71	81.76	149.77	149.72
Zone14 [kWh/m2]	121.22	120.28	97.67	74.05	74.05	74.09	151.54	151.52
Zone15 [kWh/m2]	114.28	113.37	92.21	69.28	69.28	69.33	152.85	152.73
Zone16 [kWh/m2]	113.83	112.92	90.44	67.53	67.53	67.57	153.52	153.38
Zone17 [kWh/m2]	117.11	116.16	92.69	68.85	68.85	68.90	153.40	153.30
Zone18 [kWh/m2]	127.24	126.21	99.41	73.67	73.67	73.71	152.80	152.73
Zone19 [kWh/m2]	141.89	140.75	111.54	82.72	82.72	82.77	153.06	152.95
Zone20 [kWh/m2]	166.24	164.92	130.20	96.94	96.94	97.00	164.61	164.61

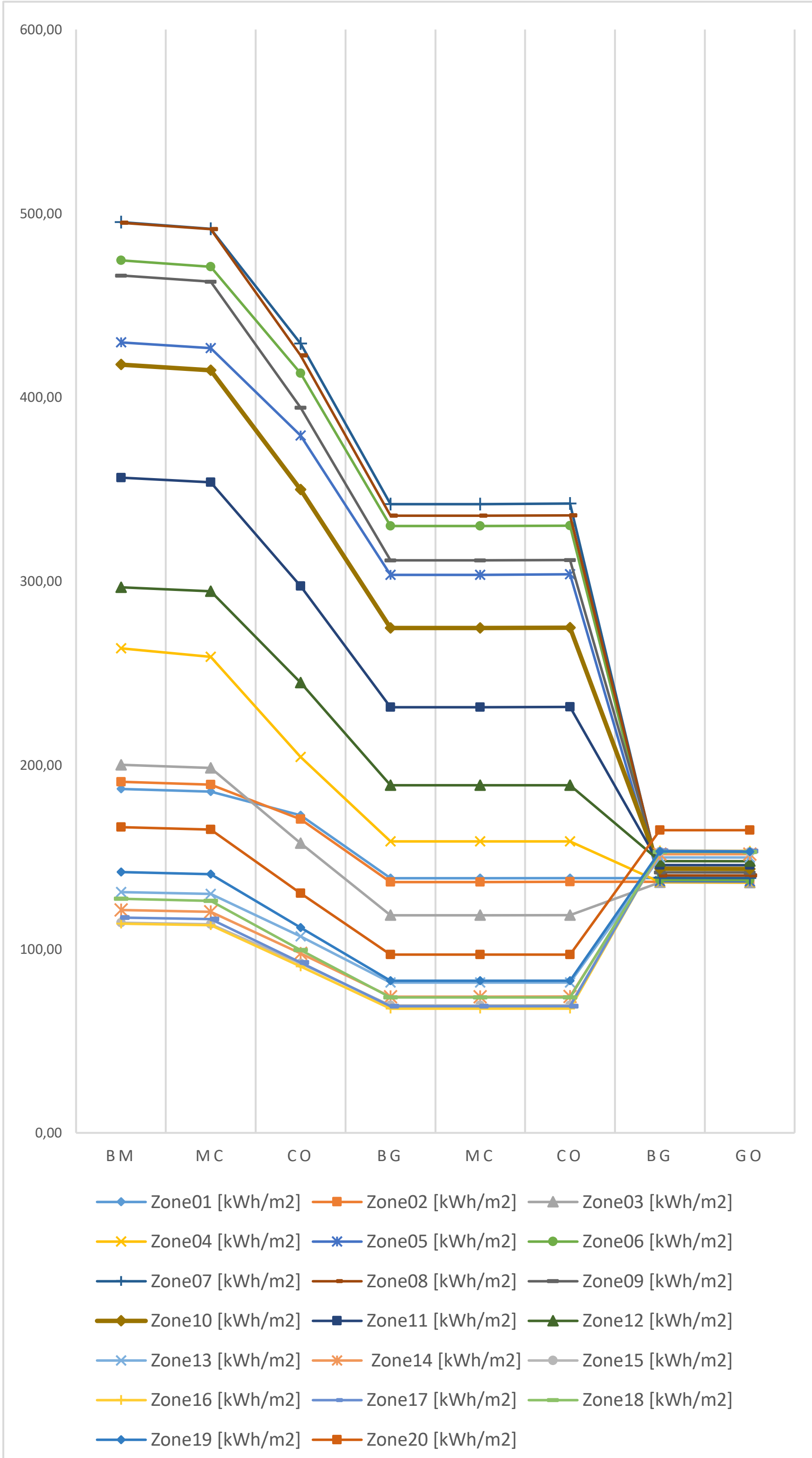


Figure 4-46 Zones normalized heating energy demand evolution during the optimizations iterations.

Figure 4-47 and Table 4-15 demonstrate the electric light energy demand evolution normalized by floor area, per zone.

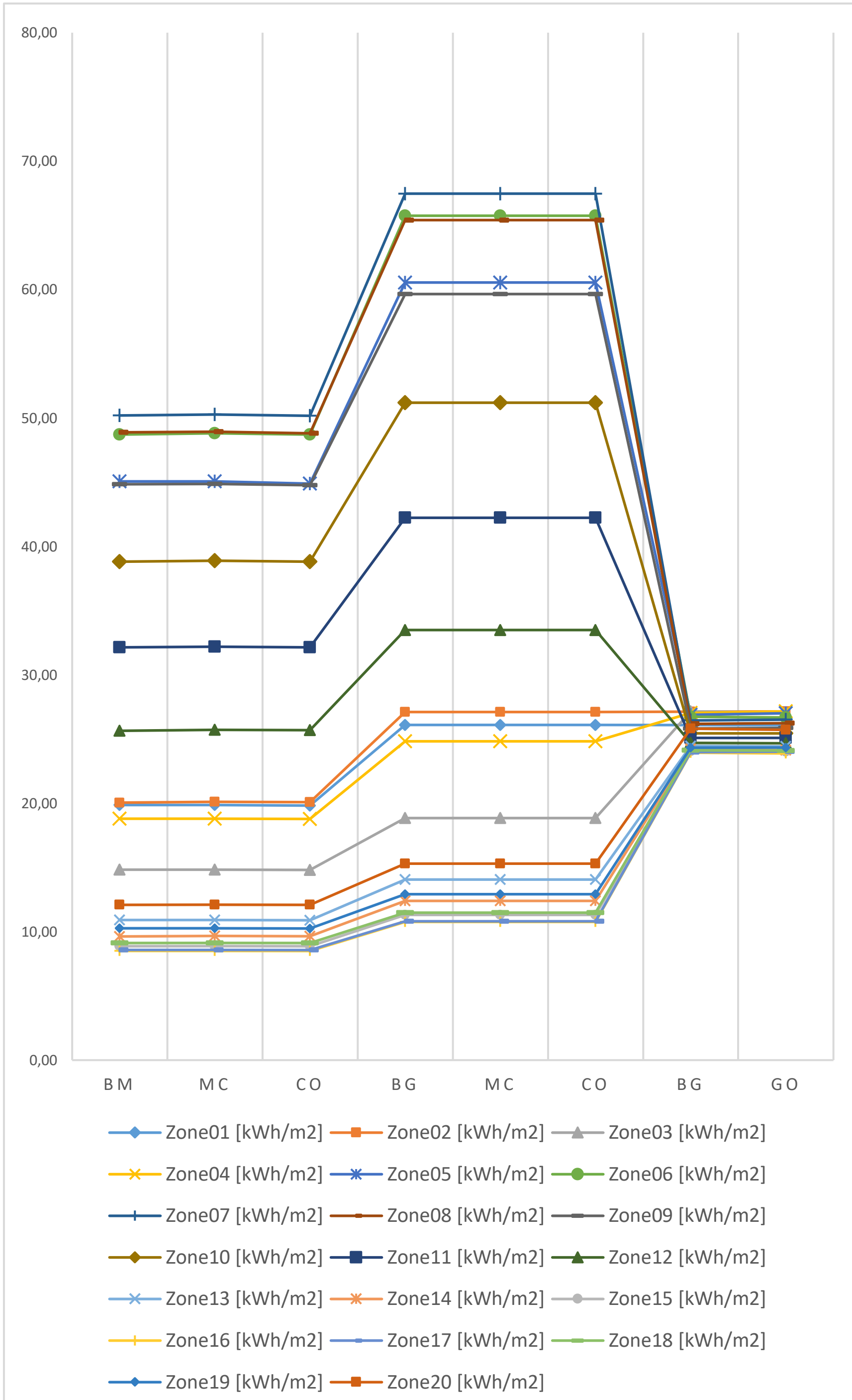


Figure 4-47 Zones normalized electric light energy demand evolution during the optimizations iterations.

Table 4-15 Related numerical values of Figure 4-47.

	BM	MC	CO	BG	MC	CO	BG	GO
Zone01 [kWh/m2]	19.87	19.87	19.82	26.10	26.10	26.10	26.08	26.08
Zone02 [kWh/m2]	20.05	20.12	20.10	27.10	27.10	27.10	27.13	27.15
Zone03 [kWh/m2]	14.84	14.84	14.81	18.84	18.84	18.84	27.15	27.07
Zone04 [kWh/m2]	18.79	18.81	18.78	24.82	24.82	24.82	27.03	27.15
Zone05 [kWh/m2]	45.06	45.06	44.89	60.55	60.55	60.55	26.89	27.01
Zone06 [kWh/m2]	48.71	48.80	48.70	65.75	65.75	65.75	26.74	26.68
Zone07 [kWh/m2]	50.19	50.28	50.18	67.47	67.47	67.47	26.45	26.51
Zone08 [kWh/m2]	48.89	48.93	48.82	65.41	65.41	65.41	26.18	26.23
Zone09 [kWh/m2]	44.83	44.87	44.77	59.65	59.65	59.65	25.82	25.86
Zone10 [kWh/m2]	38.82	38.89	38.81	51.19	51.19	51.19	25.45	25.44
Zone11 [kWh/m2]	32.14	32.20	32.14	42.23	42.23	42.23	25.09	25.09
Zone12 [kWh/m2]	25.65	25.72	25.70	33.49	33.49	33.49	24.71	24.67
Zone13 [kWh/m2]	10.92	10.92	10.89	14.07	14.07	14.07	24.46	24.44
Zone14 [kWh/m2]	9.63	9.66	9.65	12.39	12.39	12.39	24.24	24.13
Zone15 [kWh/m2]	8.87	8.88	8.87	11.31	11.31	11.31	24.03	24.01
Zone16 [kWh/m2]	8.51	8.51	8.50	10.76	10.76	10.76	23.94	23.88
Zone17 [kWh/m2]	8.58	8.58	8.57	10.81	10.81	10.81	23.99	24.02
Zone18 [kWh/m2]	9.12	9.13	9.12	11.47	11.47	11.47	24.12	24.11
Zone19 [kWh/m2]	10.26	10.27	10.25	12.92	12.92	12.92	24.33	24.34
Zone20 [kWh/m2]	12.09	12.11	12.09	15.30	15.30	15.30	25.81	25.71

Table 4-16 shows the variations of the cluster variables during the optimization iterations between the attribute clusters.

Table 4-16 Variable values evolution during the optimization iterations.

	BM	MC	CO	BG	MC	CO	BG	GO
Plan geometry (number of the polygon edges)	4			6			6	6
Rotations of successive floors [°]	5			10			1	2
Window to wall ratio [%]	50			10			10	10
Shading slat length [m]	0.50			0			0	0
Number of the shadings on the windows	2			0			0	4
Position of the shadings (Horizontal/Vertical)	H			H			H	V
External Wall Construction	2	4			5			3
Roof Construction	1	0			0			0
Windows Construction	1	1			1			1
Shading's material Reflectance [%]	50	27			23			94
Shading's material Emissivity [%]	80	39			46			81
Minumum Outdoor Temperature for Natural Ventilation [°C]	20		22			21		22
Maximum Outdoor Temperature for Natural Ventilation [°C]	24		24			24		23
Shading control setpoint (Solar incident on the windows) [W/m2]	50		15			77		81
Shading angle [°]	45		0			20		60

iv. Discussion

The second example also displayed a rapid convergence toward optima, as indicated by a decrease in the value of the cost function (Figure 4-43). The bulk of optimization-based design improvement is in fact achieved during the first three iterations. Note that after the third iteration, no change in cost function and (no significant change in) variable values is observed. As observed in example one, as it was shown in the graphs and tables, the proposed intra-cluster cycling approach resulted in a better design solution in relation to achieving the minimum value of the cost function compare to the one-shot global optimization. Since, this case study was more complex in terms of geometry and has more zones, the benefits of transparency are easier to comprehend. The energy performance of the zones varied dramatically during the iterations and interestingly converged at the end of the procedure (Figures 4-45 to 4-47). Therefore, the proposed approach helps the designer to consider every single zone’s performance in correlation to the different attribute clusters and make the decisions wisely.

Chapter 5

5. Conclusion

5.1. Contributions

In this research study a novel approach toward iterative global optimization of locally optimized attribute clusters of building design solutions has been introduced. If well-structured, such clusters of design space attributes can be easily comprehensible to typical building designers as a compound yet coherent aspects of a design (e.g., building enclosure, building materials, building geometry, building systems, and control systems). Thus, grouped, clusters can be made subject to multiple passes of local simulation-assisted optimizations, instead of a single-pass black-box global optimization step. Evidence of the concept of the approach was provided via different implementations using existing simulation and optimization tools.

The implementation provides the users with degrees of freedom in regard to the view of the selection of the clusters to be optimized. Moreover, additional clusters can be defined and variable sets in each cluster can be manipulated, while still achieving convergence within reasonable temporal horizons.

The performance of the implementations of the proposed approach via optimization case studies, which contained different system operation options, was illustrated (e.g., random cycling between attribute clusters versus predefined sequences and also different complexities of the buildings).

The proposed method delivers optimized solutions that are – as far as the values of the energy performance indicators and the associate cost function are concerned – virtually indistinguishable from those of a reference, one-shot global optimization run and in some cases are even better. However, in this approach the results are not only obtained faster, more efficiently and more accurately, but also via a transparent, traceable, and designer-friendly process.

5.2. Future research

The proposed approach has the capacity to be a potential concept of a new plug-in for Grasshopper. Achieving this goal is challenging, as it will require a significant amount of programming as well as sophisticated integrations between different software and tools. Combining different programming languages effectively will be another challenge.

It might be necessary to develop a new optimizer which works tightly and more efficiently with this system; in this case it would be a good opportunity to test different optimization algorithms under the approach proposed in this study.

In this research project, EnergyPlus was considered as the main simulation engine to calculate the energy demand of the building. The other option would be to use different simulation engines for daylight, CFD, etc. and probably implement the system with multi-objective optimizations.

5.3. Related publications

H. Shirdel, A. Mahdavi, "Convergence toward optimal building designs via multiple iterative local optimization steps applied to attribute clusters of design variants", BauSIM2018, Karlsruhe, Germany, September 2018.

A. Mahdavi, H. Shirdel, F. Tahmasebi, “A novel approach to building performance optimization via iterative operations on attribute clusters of designs options”, The 11th European Conference on Product & Process Modelling, Limassol, Cyprus, September 2016.

6. References

6.1. Literature

Adamski, M., 2007. Optimmization of the form of a building on an oval base. *Building and environment,* Volume 42.

Al-homoud, M., 2005. A systematic approach for the thermal design optimization of building envelopes. *Building physics,* Volume 29.

Al-Homoud, M. S., 2001. Computer-aided building energy analysis techniques. *Building and Environment,* Volume 36, pp. 421-433.

Anon., 2018. *EnergyPlus.* [Online]
Available at: https://energyplus.net/

Bambrook, S., Sproul, A. & Jacob, D., 2011. Design optimisation for low energy home in Sydney. *Energy and buildings,* Volume 43.

Coley, D. A. & Schukat, S., 2002. Low-energy design: combining computer-based optimizaition and human judgment. *Building and environment,* Volume 37.

Diakaki, C., Grigoroudis, E. & Klokotsa, D., 2008. Toward a multi-objective optimization approach for improving energy efficieny in buildings. *Energy and buildings,* Volume 40.

Evins, R., 2013. A review of computational optimization methods applied to sustainable building design. *Elsevier Ltd..*

Ferrara, M., Filippi, M., Sirombo, E. & Cravino, V., 2015. *A simulation-based optimization method for the integrative design of the building envelope.* Torino, s.n., pp. 2608-2613.

Fiske, V. P. & Bhonde, C. B. K., 2015. Energy Simulation in Buildings. *International Journal of Scientific & Engineering Research,* April.6(4).

Gama Caldas, L. & Norford, L. K., 2002. A design optimization tool based on a genetic algorithm. *Automation in construction,* Volume 11.

Hasan, A., Palonen, M. & Hamdy, M., 2015. Simulation-Based Optimization for Energy and Buildings. In: A. Sayigh, ed. *Renewable Energy in the Service of Mankind.* s.l.:Springer, Cham, pp. 503-513.

Hayter , S. J. & Kandt, A., 2011. *Renrewable energy applications for existing buildings.* Baveno-Lago Maggiore, Italy, s.n.

Holst, J. N., 2003. *Using whole building simulation models and optimizing procedures to optimize building envelope design with respect to energy consumption and indoor environment.* Eindhoven, Netherlands, Eighth International IBPSA conference.

Kampf, J. H. et al., 2010. Optimisation of buildings' solar irradiation availability. *Solar energy,* Volume 84.

Kreyszig, E., 2011. *Advanced engineering mathematics.* Tenth ed. Ohio: Laurie Rosatone.

Kyu Yi, Y. & Malkawi, A. M., 2009. Optimizing building form for energy performance based on hierarchical geometry relation. *Automation in construction,* Volume 18.

Leskovar, V. z. & Premrov, M., 2011. An approach in architectural design of energy-efficient timber buildings with a focus on the optimal glazing size in the south-oriented facade. *Elsevier B.V..*

Magnier, L. & Haghighat, F., 2010. Multiobjective optimization of building design using TRANSYS simulations, genetic algorithm, and artificial neural network. *Building and environment,* Volume 45.

Mahdavi A., Mahattanatawe P., 2003. *Enclosure systems design and control support via dynamic simulation-assisted optimization.* Eindhoven, s.n., pp. 785-792.

Mahdavi, A., Shirdel, H. & Tahmasebi, F., 2016. *A novel approach to building performance optimization via iterative operations on* *attribute clusters of designs options.* Limassol, eWork and eBusiness in Architecture, Engineering and Construction.

Manzan, M. & Pinto, F., 2009. *Genetic optimization of external shading devices.* Glasgow, Scotland, Eleventh international IBPSA conference.

Marsh, A., 20003. *Computer-optimized shading design.* Eindhoven, Netherlands, Eighth international IBPSA conference.

McNeel, R., 2013. *Essential mathematics for computational design.* Third ed. s.l.:Creative commons attribution-share.

Nguyen A., Reiter S., Rigo P., 2014. A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy,* Volume 113, pp. 1043-1058.

Ordonez, J. & Modi, V., 2011. Optimizing CO2 emissions from heating and cooling and from the materials used in residential buildings, depending on their geometric characteristics. *Building and environment,* Volume 46.

Sadeghipour Roudasri, M., 2018. *Food4Rhino.* [Online]
Available at: http://www.food4rhino.com/app/ladybug-tools

Sadeghipour Roudsari, M. & Mackey, C., 2018. *Ladybug Tools.* [Online]
Available at: http://www.ladybug.tools/

Sassi, P., 2006. *Strategies for sustainable architecture.* First ed. s.l.:Taylor & Francis Group.

Sivanandam, S. N. & Deepa, S. N., 2008. *Introduction to genetic algorithms.* Berlin: Springer-Verlag.

standard, I., 2007. *Building materials and products.* s.l.:ISO.

Stavrakakis, G., Zervas, P., Sarimveis, H. & Markatos, N., 2012. Optimization of window-openings design for thermal comfort in naturally ventilated buildings. *Applied Mathematical Modelling,* 36(1), pp. 193-211.

Szokolay, S., 2008. *Introduction to architectural science.* Second ed. s.l.:Elsevier Ltd..

Tuhus-Dubrow, D. & Krarti, M., 2010. Genetic-algorithm based approach to optimize building envelope design for residential buildings. *Buildingand environment,* Volume 45.

U.S. Department of Energy, 2017. *EnergyPlus Input Output Reference.* EnergyPlus Version 8.8.0 Documentation ed. s.l.:The University of Illinois.

Urge-Vorsatz D., Cabeza L., Serrano S., Barreneche C., Petrichenko K., 2015. Heating and cooling energy trends and drivers in buildings. *Renewable and Sustainable Energy Reviews,* Volume 41, pp. 85-98.

Wang, W., Rivard, H. & Zmeureanu, R., 2006. Floor shape optimization for green building design. *Advanced engineering informatics,* Volume 20.

Wang, W., Zmeureanu, R. & Rivard, H., 2005. Applying multi-objective genetic algorithms in green building design optimizaiton. *Building and environment ,* Volume 40.

Wetter, M., 2004. *Simulation-based building energy optimizaiton.* Berkeley: University of California.

Wetter, M., 2016. *GenOpt, Generic optimization program,* s.l.: Lawrence Berkeley National Laboratory.

Wetter, M., 2016. *GenOpt, Generic Optimization Program.* [Online] Available at: https://simulationresearch.lbl.gov/GO/

Wetter, M. & Polak, E., 2005. Building design optimization using a convergent pattern search algorithm with adaptive precicion simulations. *Energy and building ,* Volume 37.

Wortmann, T., 2017. *Opossum: Introducing and Evaluating a Model-based Optimization Tool for Grasshopper.* Hong Kong, s.n.

Wright, J. & Mourshed, M., 2009. *Geometric optimization of fenestration.* Glasgow, Scotland, Eleventh international IBPSA conference.

Yang, C. et al., 2014. High throughput computing based distributed genetic algorithm for building energy consumption optimization. *Energy and Buildings,* Volume 76, pp. 92-101.

Zemella, G., De march, D., Borroti, M. & Poli, I., 2011. Optimised design of energy efficient building facades via evolutionary neural networks. *Energy and buildings,* Volume 43.

6.2. Tables

6.3. Figures

6.4. Equations

Appendices

The information in this chapter will help those who want to implement the proposed approach by coupling EnergyPlus and GenOpt. In the next two sections the sample codes of the necessary Genopt files as well as one Java class which belongs to a cluster will come.

i. GenOpt files

This code belongs to the cluster Façade Geometry.

Command file

```
/* GenOpt example command file
   MWetter@lbl.gov, 06/18/2003
*/

// perimeter Zone1 South Façade
Vary{
  Parameter{   //
    Name    = WindowDimensionZn1;
    Min     =  0.2;
    Ini     =  1;
    Max     =  2.65;
    Step    =  0.2;
  }
  Function{    //
    Name    = Lzn1;
    Function="multiply(%WindowDimensionZn1%, -0.5)";
  }
  Function{    // x Coordinate of the Start point of
the windowsZN1
    Name    = Window1StartXZn1;
    Function="add(%Lzn1%, 2.475)";
  }
    Function{    //
    Name    = Window2StartXZn1;
    Function="add(%Lzn1%, 7.025)";
  }
      Function{    //
    Name    = Window3StartXZn1;
    Function="add(%Lzn1%, 11.575)";
  }
      Function{    //
    Name    = Window4StartXZn1;
    Function="add(%Lzn1%, 16.125)";
  }
```

```
      Function{    //
    Name    = Window5StartXZn1;
    Function="add(%Lzn1%, 20.675)";
  }
      Function{    //
    Name    = Window6StartXZn1;
    Function="add(%Lzn1%, 25.225)";
  }
  Function{    //
    Name    = Hzn1;
    Function="multiply(%WindowDimensionZn1%, -0.5)";
  }

    Function{    // z Coordinate of the Start point
of windowsZN1
    Name    = WindowsStartZzn1;
    Function="add(%Hzn1%, 1.525)";
  }



  Parameter{   //
    Name    = WindowDimensionZn2;
    Min     =  0.2;
    Ini     =  1;
    Max     =  2.65;
    Step    =  0.2;
  }
  Function{    //
    Name    = Lzn2;
    Function="multiply(%WindowDimensionZn2%, -0.5)";
  }
  Function{    // y Coordinate of the Start point of
the windowsZN2
    Name    = Window1StartYZn2;
    Function="add(%Lzn2%, 2.4575)";
  }
    Function{    //
    Name    = Window2StartYZn2;
    Function="add(%Lzn2%, 6.9725)";
  }
      Function{    //
    Name    = Window3StartYZn2;
    Function="add(%Lzn2%, 11.4875)";
  }
      Function{    //
    Name    = Window4StartYZn2;
    Function="add(%Lzn2%, 16.0025)";
  }
    Function{    //
    Name    = Hzn2;
    Function="multiply(%WindowDimensionZn2%, -0.5)";
```

```
  }

    Function{    // z Coordinate of the Start point
of windowsZN2
    Name    = WindowsStartZzn2;
    Function="add(%Hzn2%, 1.525)";
  }


  Parameter{   //
    Name    = WindowDimensionZn3;
    Min     =  0.2;
    Ini     =  1;
    Max     =  2.65;
    Step    =  0.2;
  }
  Function{    //
    Name    = Lzn3;
    Function="multiply(%WindowDimensionZn3%, -0.5)";
  }
  Function{    // x Coordinate of the Start point of
the windowsZN3
    Name    = Window1StartXZn3;
    Function="add(%Lzn3%, 2.475)";
  }
    Function{    //
    Name    = Window2StartXZn3;
    Function="add(%Lzn3%, 7.025)";
  }
      Function{    //
    Name    = Window3StartXZn3;
    Function="add(%Lzn3%, 11.575)";
  }
      Function{    //
    Name    = Window4StartXZn3;
    Function="add(%Lzn3%, 16.125)";
  }
      Function{    //
    Name    = Window5StartXZn3;
    Function="add(%Lzn3%, 20.675)";
  }
      Function{    //
    Name    = Window6StartXZn3;
    Function="add(%Lzn3%, 25.225)";
  }
  Function{    //
    Name    = Hzn3;
    Function="multiply(%WindowDimensionZn3%, -0.5)";
  }

    Function{    // z Coordinate of the Start point
of windowsZN3
```

```
    Name    = WindowsStartZzn3;
    Function="add(%Hzn3%, 1.525)";
  }

  Parameter{   //
    Name    = WindowDimensionZn4;
    Min     =  0.2;
    Ini     =  1;
    Max     =  2.65;
    Step    =  0.2;
  }
  Function{    //
    Name    = Lzn4;
    Function="multiply(%WindowDimensionZn4%, -0.5)";
  }
  Function{    // y Coordinate of the Start point of
the windowsZN4
    Name    = Window1StartYZn4;
    Function="add(%Lzn4%, 2.4575)";
  }
    Function{    //
    Name    = Window2StartYZn4;
    Function="add(%Lzn4%, 6.9725)";
  }
      Function{    //
    Name    = Window3StartYZn4;
    Function="add(%Lzn4%, 11.4875)";
  }
      Function{    //
    Name    = Window4StartYZn4;
    Function="add(%Lzn4%, 16.0025)";
  }
    Function{    //
    Name    = Hzn4;
    Function="multiply(%WindowDimensionZn4%, -0.5)";
  }

    Function{    // z Coordinate of the Start point
of windowsZN4
    Name    = WindowsStartZzn4;
    Function="add(%Hzn4%, 1.525)";
  }


  Parameter{   //
    Name    = OverhangDepthZN1;
    Min     =  0.01;
    Ini     =  0.2;
    Max     =  0.9;
    Step    =  0.1;
  }
```

```
 Parameter{   //
   Name    = OverhangDepthZN2;
   Min     =  0.01;
   Ini     =  0.2;
   Max     =  0.9;
   Step    =  0.1;
 }

 Parameter{   //
   Name    = OverhangDepthZN3;
   Min     =  0.01;
   Ini     =  0.2;
   Max     =  0.9;
   Step    =  0.1;
 }

 Parameter{   //
   Name    = OverhangDepthZN4;
   Min     =  0.01;
   Ini     =  0.2;
   Max     =  0.9;
   Step    =  0.1;
 }

Parameter{   //
   Name    =  FinDepthZN1;
   Min     =  0.01;
   Ini     =  0.2;
   Max     =  0.9;
   Step    =  0.1;
 }

  Parameter{   //
   Name    =  FinDepthZN2;
   Min     =  0.01;
   Ini     =  0.2;
   Max     =  0.9;
   Step    =  0.1;
 }

  Parameter{   //
   Name    =  FinDepthZN3;
   Min     =  0.01;
   Ini     =  0.2;
   Max     =  0.9;
   Step    =  0.1;
 }

  Parameter{   //
   Name    =  FinDepthZN4;
   Min     =  0.01;
   Ini     =  0.2;
```

```
    Max     =  0.9;
    Step    =  0.1;
  }
}

OptimizationSettings{
  MaxIte = 2000;
  MaxEqualResults = 100;
  WriteStepNumber = false;
  UnitsOfExecution = 0;
}

Algorithm{
  Main = GPSPSOCCHJ;
  NeighborhoodTopology = vonNeumann;
  NeighborhoodSize = 5;
  NumberOfParticle = 10;
  NumberOfGeneration = 10;
  Seed = 1;
  CognitiveAcceleration = 2.8;
  SocialAcceleration = 1.3;
  MaxVelocityGainContinuous = 0.5;
  MaxVelocityDiscrete = 4;
  ConstrictionGain = 0.5;
  MeshSizeDivider = 2;
  InitialMeshSizeExponent = 0;
  MeshSizeExponentIncrement = 1;
  NumberOfStepReduction = 4;
}
```

Initialization file

```
/* GenOpt example initialization file for EnergyPlus
   Operating system: Windows 10

*/

Simulation {
  Files {
    Template {
      File1 = FcdeGeome_template.idf;
    }
  Input {
      File1 = FcdeGeome.idf;
    }
    Log {
```

```
      File1 = FcdeGeome.err;
    }
    Output {
      File1 = FcdeGeome.eso;
    }
    Configuration {
      File1 = "EnergyPlus-8-1-0-Win7.cfg";
    }
  }


  CallParameter { // optional section
    // The weather file without extension
    Suffix = AUT_Vienna.Schwechat.110360_IWEC;
  }


  ObjectiveFunctionLocation
  {

      Name1      = ObjectiveFunction;
      Function1      =   "add(   %FinHeat%,   %FinCool%,
%FinElectric%)";

        Name2      = FinHeat;
      Function2     =   "divide(   %TotalHeatingEnergy%,
0.85)";
        Name3      = FinCool;
      Function3  = "multiply( %COPCool%, 1.8)";
        Name4      = COPCool;
      Function4     =   "divide(   %TotalCoolingEnergy%,
3.5)";
        Name5      = FinElectric;
      Function5                  =          "multiply(
%TotalLightsElectricEnergy%, 1.8)";
```

```
        Name6      = TotalEnergyLoad;
      Function6      =   "add(   %TotalHeatingEnergy%,
%TotalCoolingEnergy%, %TotalLightsElectricEnergy%)";

        Name7      = TotalHeatingEnergy;
      Function7        =    "add(    %HECoreZNandZN1%,
%HEZN2andZN3andZN4%)";
        Name8      = HECoreZNandZN1;
      Function8      =   "add(   %HeatingEnergyCoreZN%,
%HeatingEnergyZN1%)";
        Name9      = HEZN2andZN3andZN4;
      Function9       =    "add(    %HeatingEnergyZN2%,
%HeatingEnergyZN3%, %HeatingEnergyZN4%)";

        Name10      = TotalCoolingEnergy;
      Function10        =    "add(    %CECoreZNandZN1%,
%CEZN2andZN3andZN4%)";
        Name11      = CECoreZNandZN1;
      Function11      =   "add(   %CoolingEnergyCoreZN%,
%CoolingEnergyZN1%)";
        Name12      = CEZN2andZN3andZN4;
      Function12       =    "add(    %CoolingEnergyZN2%,
%CoolingEnergyZN3%, %CoolingEnergyZN4%)";

        Name13      = TotalLightsElectricEnergy;
      Function13        =    "add(    %LEECoreZNandZN1%,
%LEEZN2andZN3andZN4%)";
        Name14      = LEECoreZNandZN1;
      Function14  = "add( %LightsElectricEnergyCoreZN%,
%LightsElectricEnergyZN1%)";
        Name15      = LEEZN2andZN3andZN4;
      Function15   =  "add(  %LightsElectricEnergyZN2%,
%LightsElectricEnergyZN3%,
%LightsElectricEnergyZN4%)";

        Name16      = TotalEnergyLoadCoreZN;
```

```
      Function16      =   "add(   %HeatingEnergyCoreZN%,
%CoolingEnergyCoreZN%, %LightsElectricEnergyCoreZN%)";
        Name17      = TotalEnergyLoadZN1;
      Function17       =    "add(    %HeatingEnergyZN1%,
%CoolingEnergyZN1%, %LightsElectricEnergyZN1%)";
        Name18      = TotalEnergyLoadZN2;
      Function18       =    "add(    %HeatingEnergyZN2%,
%CoolingEnergyZN2%, %LightsElectricEnergyZN2%)";
        Name19      = TotalEnergyLoadZN3;
      Function19       =    "add(    %HeatingEnergyZN3%,
%CoolingEnergyZN3%, %LightsElectricEnergyZN3%)";
        Name20      = TotalEnergyLoadZN4;
      Function20       =    "add(    %HeatingEnergyZN4%,
%CoolingEnergyZN4%, %LightsElectricEnergyZN4%)";

      Name21             = HeatingEnergyCoreZN;
      Delimiter21        = "495,";
      FirstCharacterAt21 = 1;
      Name22             = CoolingEnergyCoreZN;
      Delimiter22        = "527,";
      FirstCharacterAt22 = 1;
      Name23             = LightsElectricEnergyCoreZN;
      Delimiter23        = "67,";
      FirstCharacterAt23 = 1;

        Name24             = HeatingEnergyZN1;
      Delimiter24        = "559,";
      FirstCharacterAt24 = 1;
      Name25             = CoolingEnergyZN1;
      Delimiter25        = "561,";
      FirstCharacterAt25 = 1;
      Name26             = LightsElectricEnergyZN1;
      Delimiter26        = "89,";
      FirstCharacterAt26 = 1;

        Name27             = HeatingEnergyZN2;
```

```
Delimiter27        = "563,";
FirstCharacterAt27 = 1;
Name28             = CoolingEnergyZN2;
Delimiter28        = "565,";
FirstCharacterAt28 = 1;
Name29             = LightsElectricEnergyZN2;
Delimiter29        = "111,";
FirstCharacterAt29 = 1;

  Name30             = HeatingEnergyZN3;
Delimiter30        = "567,";
FirstCharacterAt30 = 1;
Name31             = CoolingEnergyZN3;
Delimiter31        = "569,";
FirstCharacterAt31 = 1;
Name32             = LightsElectricEnergyZN3;
Delimiter32        = "133,";
FirstCharacterAt32 = 1;

  Name33             = HeatingEnergyZN4;
Delimiter33        = "571,";
FirstCharacterAt33 = 1;
Name34             = CoolingEnergyZN4;
Delimiter34        = "573,";
FirstCharacterAt34 = 1;
Name35             = LightsElectricEnergyZN4;
Delimiter35        = "155,";
FirstCharacterAt35 = 1;

  Name36      = ZN1X1;
Function36  = %Window1StartXZn1%;
Name37      = ZN1X2;
Function37  = %Window2StartXZn1%;
Name38      = ZN1X3;
Function38  = %Window3StartXZn1%;
  Name39      = ZN1X4;
```

```
Function39  = %Window4StartXZn1%;
Name40      = ZN1X5;
Function40  = %Window5StartXZn1%;
Name41      = ZN1X6;
Function41  = %Window6StartXZn1%;

  Name42      = ZN1Z;
Function42  = %WindowsStartZzn1%;


  Name43      = ZN2Y1;
Function43  = %Window1StartYZn2%;
Name44      = ZN2Y2;
Function44  = %Window2StartYZn2%;
Name45      = ZN2Y3;
Function45  = %Window3StartYZn2%;
  Name46      = ZN2Y4;
Function46  = %Window4StartYZn2%;

  Name47      = ZN2Z;
Function47  = %WindowsStartZzn2%;


  Name48      = ZN3X1;
Function48  = %Window1StartXZn3%;
Name49      = ZN3X2;
Function49  = %Window2StartXZn3%;
Name50      = ZN3X3;
Function50  = %Window3StartXZn3%;
  Name51      = ZN3X4;
Function51  = %Window4StartXZn3%;
Name52      = ZN3X5;
Function52  = %Window5StartXZn3%;
Name53      = ZN3X6;
Function53  = %Window6StartXZn3%;
```

```
        Name54      = ZN3Z;
      Function54  = %WindowsStartZzn3%;


        Name55      = ZN4Y1;
      Function55  = %Window1StartYZn4%;
      Name56      = ZN4Y2;
      Function56  = %Window2StartYZn4%;
      Name57      = ZN4Y3;
      Function57  = %Window3StartYZn4%;
        Name58      = ZN4Y4;
      Function58  = %Window4StartYZn4%;

        Name59      = ZN4Z;
      Function59  = %WindowsStartZzn4%;
  }

 }// end of section Simulation

Optimization {
  Files {
    Command {
      File1 = command.txt;
    }
  }
} // end of configuration file
```

Configuration file

```
/* GenOpt configuration file for
   EnergyPlus on Windows 10

*/

// Error messages of the simulation program.
SimulationError
```

```
{
    ErrorMessage = "**  Fatal  **";
    ErrorMessage = "** EnergyPlus Terminated--Error(s)
Detected";
}

// Number format for writing the simulation input files.
IO
{
    NumberFormat = Double;
}

/*  Specifying how to start the simulation program.
    In "Command", only those words in %xx% are
    replaced (possibly with empty Strings).
*/
SimulationStart
{
    // The command line below calls RunEPlus.bat.
    Command   =   "cmd   /C   \"\"C:\\EnergyPlusV8-1-
0\\RunEPlus.bat\"   \"%Simulation.Files.Input.File1%\"
\"%Simulation.CallParameter.Suffix%\"\"";
    WriteInputFileExtension = false;
}
```

ii. Sample code of a Java class

This class covers the Façade Geometry cluster

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
```

```
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.logging.Level;
import java.util.logging.Logger;

public class FacadeGeometry {
      public   void   mainFacadeGeo(String   mainFolder,
double areaCoreZone,
                  double      areaPerZone1,      double
areaPerZone2, double areaPerZone3,
                  double      areaPerZone4)      throws
IOException {

            System.out
                        .println("The  optimization  of
the cluster 'Facade Geometry' is starting.");

            String   currentFoder   =   mainFolder   +
"\\FacadeGeometry";
            String     file     =     mainFolder     +
"\\file_template.idf";
            String    fileOut    =    currentFoder    +
"\\FcdeGeome_template.idf";

            String  Xword  =  "!-  Starting  X  Coordinate
{m}";
            String Win1Xzn1 = "    %Window1StartXZn1%";
            String Win2Xzn1 = "    %Window2StartXZn1%";
            String Win3Xzn1 = "    %Window3StartXZn1%";
            String Win4Xzn1 = "    %Window4StartXZn1%";
            String Win5Xzn1 = "    %Window5StartXZn1%";
            String Win6Xzn1 = "    %Window6StartXZn1%";
            String Win4Yzn2 = "    %Window4StartYZn2%";
            String Win3Yzn2 = "    %Window3StartYZn2%";
```

```
            String Win2Yzn2 = "    %Window2StartYZn2%";
            String Win1Yzn2 = "    %Window1StartYZn2%";
            String Win1Xzn3 = "    %Window1StartXZn3%";
            String Win2Xzn3 = "    %Window2StartXZn3%";
            String Win3Xzn3 = "    %Window3StartXZn3%";
            String Win4Xzn3 = "    %Window4StartXZn3%";
            String Win5Xzn3 = "    %Window5StartXZn3%";
            String Win6Xzn3 = "    %Window6StartXZn3%";
            String Win4Yzn4 = "    %Window4StartYZn4%";
            String Win3Yzn4 = "    %Window3StartYZn4%";
            String Win2Yzn4 = "    %Window2StartYZn4%";
            String Win1Yzn4 = "    %Window1StartYZn4%";

            FacadeGeometry.wordReplacer(file,  fileOut,
Xword, Win1Xzn1, Win2Xzn1,
                        Win3Xzn1,  Win4Xzn1,  Win5Xzn1,
Win6Xzn1, Win4Yzn2, Win3Yzn2,
                        Win2Yzn2,  Win1Yzn2,  Win1Xzn3,
Win2Xzn3, Win3Xzn3, Win4Xzn3,
                        Win5Xzn3,  Win6Xzn3,  Win4Yzn4,
Win3Yzn4, Win2Yzn4, Win1Yzn4);

            String  Zword  =  "!-  Starting  Z  Coordinate
{m}";
            String WinZzn1 = "    %WindowsStartZzn1%";
            String WinZzn2 = "    %WindowsStartZzn2%";
            String WinZzn3 = "    %WindowsStartZzn3%";
            String WinZzn4 = "    %WindowsStartZzn4%";
            FacadeGeometry.wordReplacer(fileOut,
fileOut, Zword, WinZzn1, WinZzn1,
                        WinZzn1,   WinZzn1,   WinZzn1,
WinZzn1, WinZzn2, WinZzn2, WinZzn2,
                        WinZzn2,   WinZzn3,   WinZzn3,
WinZzn3, WinZzn3, WinZzn3, WinZzn3,
                        WinZzn4,   WinZzn4,   WinZzn4,
WinZzn4);
```

```
            String LengthWord = "!- Length {m}";
            String       WinLengthZN1       =       "
%WindowDimensionZn1%";
            String       WinLengthZN2       =       "
%WindowDimensionZn2%";
            String       WinLengthZN3       =       "
%WindowDimensionZn3%";
            String       WinLengthZN4       =       "
%WindowDimensionZn4%";
            FacadeGeometry.wordReplacer(fileOut,
fileOut, LengthWord, WinLengthZN1,
                        WinLengthZN1,     WinLengthZN1,
WinLengthZN1, WinLengthZN1,
                        WinLengthZN1,     WinLengthZN2,
WinLengthZN2, WinLengthZN2,
                        WinLengthZN2,     WinLengthZN3,
WinLengthZN3, WinLengthZN3,
                        WinLengthZN3,     WinLengthZN3,
WinLengthZN3, WinLengthZN4,
                        WinLengthZN4,     WinLengthZN4,
WinLengthZN4);

            String HeightWord = "!- Height {m}";
            String       WinHeightZN1       =       "
%WindowDimensionZn1%";
            String       WinHeightZN2       =       "
%WindowDimensionZn2%";
            String       WinHeightZN3       =       "
%WindowDimensionZn3%";
            String       WinHeightZN4       =       "
%WindowDimensionZn4%";
            FacadeGeometry.wordReplacer(fileOut,
fileOut, HeightWord, WinHeightZN1,
                        WinHeightZN1,     WinHeightZN1,
WinHeightZN1, WinHeightZN1,
```

```
                            WinHeightZN1,     WinHeightZN2,
WinHeightZN2, WinHeightZN2,
                            WinHeightZN2,     WinHeightZN3,
WinHeightZN3, WinHeightZN3,
                            WinHeightZN3,     WinHeightZN3,
WinHeightZN3, WinHeightZN4,
                            WinHeightZN4,     WinHeightZN4,
WinHeightZN4);

              String OverDepthWord = "!- Depth {m}";
              String        OverDepthZN1        =        "
%OverhangDepthZN1%";
              String        OverDepthZN2        =        "
%OverhangDepthZN2%";
              String        OverDepthZN3        =        "
%OverhangDepthZN3%";
              String        OverDepthZN4        =        "
%OverhangDepthZN4%";
              FacadeGeometry.wordReplacer(fileOut,
fileOut, OverDepthWord,
                            OverDepthZN1,     OverDepthZN1,
OverDepthZN1, OverDepthZN1,
                            OverDepthZN1,     OverDepthZN1,
OverDepthZN2, OverDepthZN2,
                            OverDepthZN2,     OverDepthZN2,
OverDepthZN3, OverDepthZN3,
                            OverDepthZN3,     OverDepthZN3,
OverDepthZN3, OverDepthZN3,
                            OverDepthZN4,     OverDepthZN4,
OverDepthZN4, OverDepthZN4);

              String  LeftFinDepthWord  =  "!-  Left  Depth
{m}";
              String       LeftFinDepthZN1       =       "
%FinDepthZN1%";
```

```
            String      LeftFinDepthZN2      =      "
%FinDepthZN2%";
            String      LeftFinDepthZN3      =      "
%FinDepthZN3%";
            String      LeftFinDepthZN4      =      "
%FinDepthZN4%";
            FacadeGeometry.wordReplacer(fileOut,
fileOut, LeftFinDepthWord,
                        LeftFinDepthZN1,
LeftFinDepthZN1, LeftFinDepthZN1,
                        LeftFinDepthZN1,
LeftFinDepthZN1, LeftFinDepthZN1,
                        LeftFinDepthZN2,
LeftFinDepthZN2, LeftFinDepthZN2,
                        LeftFinDepthZN2,
LeftFinDepthZN3, LeftFinDepthZN3,
                        LeftFinDepthZN3,
LeftFinDepthZN3, LeftFinDepthZN3,
                        LeftFinDepthZN3,
LeftFinDepthZN4, LeftFinDepthZN4,
                        LeftFinDepthZN4,
LeftFinDepthZN4);

            String RightFinDepthWord = "!- Right Depth
{m}";
            String      RightFinDepthZN1      =      "
%FinDepthZN1%";
            String      RightFinDepthZN2      =      "
%FinDepthZN2%";
            String      RightFinDepthZN3      =      "
%FinDepthZN3%";
            String      RightFinDepthZN4      =      "
%FinDepthZN4%";
            FacadeGeometry.wordReplacer(fileOut,
fileOut, RightFinDepthWord,
```

```
                        RightFinDepthZN1,
RightFinDepthZN1, RightFinDepthZN1,
                        RightFinDepthZN1,
RightFinDepthZN1, RightFinDepthZN1,
                        RightFinDepthZN2,
RightFinDepthZN2, RightFinDepthZN2,
                        RightFinDepthZN2,
RightFinDepthZN3, RightFinDepthZN3,
                        RightFinDepthZN3,
RightFinDepthZN3, RightFinDepthZN3,
                        RightFinDepthZN3,
RightFinDepthZN4, RightFinDepthZN4,
                        RightFinDepthZN4,
RightFinDepthZN4);

            System.out.println("Variable         names
replacement is done.");

            String Command = "cmd /c cd C:\\Program
Files\\genopt   &&   java   -classpath   genopt.jar
genopt.GenOpt "
                        +        currentFoder        +
"\\optWin7.ini";
            FacadeGeometry.GenOptRun(Command);

            String   FileName1   =   currentFoder   +
"\\OutputListingMain.txt";
            String   FileName2   =   mainFolder   +
"\\BriefOutput.txt";
            FacadeGeometry.NewBriefFile(FileName1,
FileName2, areaCoreZone,
                        areaPerZone1,    areaPerZone2,
areaPerZone3, areaPerZone4);

            FacadeGeometry.wordReplacer(file,    file,
Xword, Win1Xzone1, Win2Xzone1,
```

```
                        Win3Xzone1,         Win4Xzone1,
Win5Xzone1, Win6Xzone1, Win4Yzone2,
                        Win3Yzone2,         Win2Yzone2,
Win1Yzone2, Win1Xzone3, Win2Xzone3,
                        Win3Xzone3,         Win4Xzone3,
Win5Xzone3, Win6Xzone3, Win4Yzone4,
                        Win3Yzone4,         Win2Yzone4,
Win1Yzone4);

            FacadeGeometry.wordReplacer(file,     file,
Zword, WinZzone1, WinZzone1,
                        WinZzone1,           WinZzone1,
WinZzone1, WinZzone1, WinZzone2,
                        WinZzone2,           WinZzone2,
WinZzone2, WinZzone3, WinZzone3,
                        WinZzone3,           WinZzone3,
WinZzone3, WinZzone3, WinZzone4,
                        WinZzone4,           WinZzone4,
WinZzone4);

            FacadeGeometry.wordReplacer(file,     file,
LengthWord, WinDimZone1,
                        WinDimZone1,       WinDimZone1,
WinDimZone1, WinDimZone1,
                        WinDimZone1,       WinDimZone2,
WinDimZone2, WinDimZone2,
                        WinDimZone2,       WinDimZone3,
WinDimZone3, WinDimZone3,
                        WinDimZone3,       WinDimZone3,
WinDimZone3, WinDimZone4,
                        WinDimZone4,       WinDimZone4,
WinDimZone4);

            FacadeGeometry.wordReplacer(file,     file,
HeightWord, WinDimZone1,
```

```
                        WinDimZone1,       WinDimZone1,
WinDimZone1, WinDimZone1,
                        WinDimZone1,       WinDimZone2,
WinDimZone2, WinDimZone2,
                        WinDimZone2,       WinDimZone3,
WinDimZone3, WinDimZone3,
                        WinDimZone3,       WinDimZone3,
WinDimZone3, WinDimZone4,
                        WinDimZone4,       WinDimZone4,
WinDimZone4);

            FacadeGeometry.wordReplacer(file,     file,
OverDepthWord,
                        OverhangDepthZN1,
OverhangDepthZN1, OverhangDepthZN1,
                        OverhangDepthZN1,
OverhangDepthZN1, OverhangDepthZN1,
                        OverhangDepthZN2,
OverhangDepthZN2, OverhangDepthZN2,
                        OverhangDepthZN2,
OverhangDepthZN3, OverhangDepthZN3,
                        OverhangDepthZN3,
OverhangDepthZN3, OverhangDepthZN3,
                        OverhangDepthZN3,
OverhangDepthZN4, OverhangDepthZN4,
                        OverhangDepthZN4,
OverhangDepthZN4);
            FacadeGeometry.wordReplacer(file,     file,
LeftFinDepthWord,
                        FinDepthZone1,   FinDepthZone1,
FinDepthZone1, FinDepthZone1,
                        FinDepthZone1,   FinDepthZone1,
FinDepthZone2, FinDepthZone2,
                        FinDepthZone2,   FinDepthZone2,
FinDepthZone3, FinDepthZone3,
```

```
                        FinDepthZone3,   FinDepthZone3,
FinDepthZone3, FinDepthZone3,
                        FinDepthZone4,   FinDepthZone4,
FinDepthZone4, FinDepthZone4);
            FacadeGeometry.wordReplacer(file,     file,
RightFinDepthWord,
                        FinDepthZone1,   FinDepthZone1,
FinDepthZone1, FinDepthZone1,
                        FinDepthZone1,   FinDepthZone1,
FinDepthZone2, FinDepthZone2,
                        FinDepthZone2,   FinDepthZone2,
FinDepthZone3, FinDepthZone3,
                        FinDepthZone3,   FinDepthZone3,
FinDepthZone3, FinDepthZone3,
                        FinDepthZone4,   FinDepthZone4,
FinDepthZone4, FinDepthZone4);

            System.out
                        .println("New Brief Report and
Template File are ready for the next iteration.");
      }

      public static void wordReplacer(String InputFile,
String OutputFile,
                  String Word, String... AuxWrd) throws
FileNotFoundException,
                  IOException {
            BufferedReader bf = new BufferedReader(new
FileReader(InputFile));
            String input = "";
            String line;
            String oldChar;
            String AuxWord = "    %TheAuxWord%";
            int linecount = 1;
            while ((line = bf.readLine()) != null) {
                  int indexfound = line.indexOf(Word);
```

```
                    if (indexfound > -1) {
                          System.out.println("The Word '"
+ Word
                                            + "' is at the
position " + indexfound
                                            + " on the line " +
linecount);
                          String    tokens[]    =
line.split(",|\\;");
                          oldChar = tokens[0];
                          System.out.println("The  first
word of the line is " + oldChar);
                          line  =  line.replace(oldChar,
AuxWord);
                          input    +=    line    +
System.lineSeparator();
                    } else {
                          input    +=    line    +
System.lineSeparator();
                    }
                    linecount++;
             }

             for (String AuxWrd1 : AuxWrd) {
                    input  =  input.replaceFirst(AuxWord,
AuxWrd1);
             }

             FileOutputStream    os    =    new
FileOutputStream(OutputFile);
             os.write(input.getBytes());
             os.close();
             bf.close();
       }

       public static void GenOptRun(String command) {
```

```
            try {
                  String line;
                  Process              p              =
Runtime.getRuntime().exec(command);
                  BufferedReader     bri     =     new
BufferedReader(new InputStreamReader(
                              p.getInputStream()));
                  BufferedReader     bre     =     new
BufferedReader(new InputStreamReader(
                              p.getErrorStream()));
                  while  ((line  =  bri.readLine())  !=
null) {
                        System.out.println(line);
                  }
                  bri.close();
                  while  ((line  =  bre.readLine())  !=
null) {
                        System.out.println(line);
                  }
                  bre.close();
                  p.waitFor();
                  System.out.println("Optimization   of
this cluster is done!");
            } catch (Exception err) {
                  err.printStackTrace();
            }
      }

      static String Win1Xzone1;
      static String Win2Xzone1;
      static String Win3Xzone1;
      static String Win4Xzone1;
      static String Win5Xzone1;
      static String Win6Xzone1;
      static String WinZzone1;
      static String Win1Yzone2;
```

```
static String Win2Yzone2;
static String Win3Yzone2;
static String Win4Yzone2;
static String WinZzone2;
static String Win1Xzone3;
static String Win2Xzone3;
static String Win3Xzone3;
static String Win4Xzone3;
static String Win5Xzone3;
static String Win6Xzone3;
static String WinZzone3;
static String Win1Yzone4;
static String Win2Yzone4;
static String Win3Yzone4;
static String Win4Yzone4;
static String WinZzone4;

static String WinDimZone1;
static String WinDimZone2;
static String WinDimZone3;
static String WinDimZone4;

static String OverhangDepthZN1;
static String OverhangDepthZN2;
static String OverhangDepthZN3;
static String OverhangDepthZN4;

static String FinDepthZone1;
static String FinDepthZone2;
static String FinDepthZone3;
static String FinDepthZone4;

public static void NewBriefFile(String fileIn,
String fileOut,
            double coreZNarea, double ZN1area,
double ZN2area, double ZN3area,
```

```
                double ZN4area) throws IOException {

            FileInputStream       in       =       new
FileInputStream(fileIn);
            BufferedReader br = new BufferedReader(new
InputStreamReader(in));

            String strLine = null, tmp;
            while ((tmp = br.readLine()) != null) {
                  strLine = tmp;
            }
            String lastLine = strLine;
            ArrayList<String>      items      =      new
ArrayList<>(Arrays.asList(lastLine
                        .split("\\t")));
            System.out.println(items);

            String ObjectiveFunction = items.get(3);

            String EnergyLoadCoreZN = items.get(18);
            String EnergyLoadZN1 = items.get(19);
            String EnergyLoadZN2 = items.get(20);
            String EnergyLoadZN3 = items.get(21);
            String EnergyLoadZN4 = items.get(22);

            String HeatingEnergyCoreZN = items.get(23);
            String CoolingEnergyCoreZN = items.get(24);
            String     LightsElectricEnergyCoreZN     =
items.get(25);
            String HeatingEnergyZN1 = items.get(26);
            String CoolingEnergyZN1 = items.get(27);
            String      LightsElectricEnergyZN1       =
items.get(28);
            String HeatingEnergyZN2 = items.get(29);
            String CoolingEnergyZN2 = items.get(30);
```

```
            String      LightsElectricEnergyZN2      =
items.get(31);
            String HeatingEnergyZN3 = items.get(32);
            String CoolingEnergyZN3 = items.get(33);
            String      LightsElectricEnergyZN3      =
items.get(34);
            String HeatingEnergyZN4 = items.get(35);
            String CoolingEnergyZN4 = items.get(36);
            String      LightsElectricEnergyZN4      =
items.get(37);

            double         HeaEnerCoreZN_J         =
Double.parseDouble(HeatingEnergyCoreZN);
            double         CoolEnerCoreZN_J         =
Double.parseDouble(CoolingEnergyCoreZN);
            double            LEECoreZN_J            =
Double.parseDouble(LightsElectricEnergyCoreZN);
            double           HeaEnerZN1_J           =
Double.parseDouble(HeatingEnergyZN1);
            double           CoolEnerZN1_J           =
Double.parseDouble(CoolingEnergyZN1);
            double              LEEZN1_J              =
Double.parseDouble(LightsElectricEnergyZN1);
            double           HeaEnerZN2_J           =
Double.parseDouble(HeatingEnergyZN2);
            double           CoolEnerZN2_J           =
Double.parseDouble(CoolingEnergyZN2);
            double              LEEZN2_J              =
Double.parseDouble(LightsElectricEnergyZN2);
            double           HeaEnerZN3_J           =
Double.parseDouble(HeatingEnergyZN3);
            double           CoolEnerZN3_J           =
Double.parseDouble(CoolingEnergyZN3);
            double              LEEZN3_J              =
Double.parseDouble(LightsElectricEnergyZN3);
```

```
            double          HeaEnerZN4_J             =
Double.parseDouble(HeatingEnergyZN4);
            double          CoolEnerZN4_J            =
Double.parseDouble(CoolingEnergyZN4);
            double             LEEZN4_J              =
Double.parseDouble(LightsElectricEnergyZN4);

            Win1Xzone1 = items.get(38);
            Win2Xzone1 = items.get(39);
            Win3Xzone1 = items.get(40);
            Win4Xzone1 = items.get(41);
            Win5Xzone1 = items.get(42);
            Win6Xzone1 = items.get(43);
            WinZzone1 = items.get(44);
            Win1Yzone2 = items.get(45);
            Win2Yzone2 = items.get(46);
            Win3Yzone2 = items.get(47);
            Win4Yzone2 = items.get(48);
            WinZzone2 = items.get(49);
            Win1Xzone3 = items.get(50);
            Win2Xzone3 = items.get(51);
            Win3Xzone3 = items.get(52);
            Win4Xzone3 = items.get(53);
            Win5Xzone3 = items.get(54);
            Win6Xzone3 = items.get(55);
            WinZzone3 = items.get(56);
            Win1Yzone4 = items.get(57);
            Win2Yzone4 = items.get(58);
            Win3Yzone4 = items.get(59);
            Win4Yzone4 = items.get(60);
            WinZzone4 = items.get(61);

            WinDimZone1 = items.get(62);
            WinDimZone2 = items.get(63);
            WinDimZone3 = items.get(64);
            WinDimZone4 = items.get(65);
```

```
            OverhangDepthZN1 = items.get(66);
            OverhangDepthZN2 = items.get(67);
            OverhangDepthZN3 = items.get(68);
            OverhangDepthZN4 = items.get(69);

            FinDepthZone1 = items.get(70);
            FinDepthZone2 = items.get(71);
            FinDepthZone3 = items.get(72);
            FinDepthZone4 = items.get(73);

            double              value6              =
Double.parseDouble(EnergyLoadCoreZN);
            double TCoreZN_KWh = value6 / 3600000;
            double   TCoreZN_KWhm2   =   TCoreZN_KWh   /
coreZNarea;
            double              value7              =
Double.parseDouble(EnergyLoadZN1);
            double TZN1_KWh = value7 / 3600000;
            double TZN1_KWhm2 = TZN1_KWh / ZN1area;
            double              value8              =
Double.parseDouble(EnergyLoadZN2);
            double TZN2_KWh = value8 / 3600000;
            double TZN2_KWhm2 = TZN2_KWh / ZN2area;
            double              value9              =
Double.parseDouble(EnergyLoadZN3);
            double TZN3_KWh = value9 / 3600000;
            double TZN3_KWhm2 = TZN3_KWh / ZN3area;
            double             value10              =
Double.parseDouble(EnergyLoadZN4);
            double TZN4_KWh = value10 / 3600000;
            double TZN4_KWhm2 = TZN4_KWh / ZN4area;

            double HeaEnerCoreZN_kWh = HeaEnerCoreZN_J
/ 3600000;
```

```
            double CoolEnerCoreZN_kWh = CoolEnerCoreZN_J / 3600000;
            double LEECoreZN_kWh = LEECoreZN_J / 3600000;
            double HeaEnerZN1_kWh = HeaEnerZN1_J / 3600000;
            double CoolEnerZN1_kWh = CoolEnerZN1_J / 3600000;
            double LEEZN1_kWh = LEEZN1_J / 3600000;
            double HeaEnerZN2_kWh = HeaEnerZN2_J / 3600000;
            double CoolEnerZN2_kWh = CoolEnerZN2_J / 3600000;
            double LEEZN2_kWh = LEEZN2_J / 3600000;
            double HeaEnerZN3_kWh = HeaEnerZN3_J / 3600000;
            double CoolEnerZN3_kWh = CoolEnerZN3_J / 3600000;
            double LEEZN3_kWh = LEEZN3_J / 3600000;
            double HeaEnerZN4_kWh = HeaEnerZN4_J / 3600000;
            double CoolEnerZN4_kWh = CoolEnerZN4_J / 3600000;
            double LEEZN4_kWh = LEEZN4_J / 3600000;

            double HeaEnerCoreZN_kWhm2 = HeaEnerCoreZN_kWh / coreZNarea;
            double CoolEnerCoreZN_kWhm2 = CoolEnerCoreZN_kWh / coreZNarea;
            double LEECoreZN_kWhm2 = LEECoreZN_kWh / coreZNarea;
            double HeaEnerZN1_kWhm2 = HeaEnerZN1_kWh / ZN1area;
            double CoolEnerZN1_kWhm2 = CoolEnerZN1_kWh / ZN1area;
            double LEEZN1_kWhm2 = LEEZN1_kWh / ZN1area;
```

```
            double HeaEnerZN2_kWhm2 = HeaEnerZN2_kWh /
ZN2area;
            double CoolEnerZN2_kWhm2 = CoolEnerZN2_kWh
/ ZN2area;
            double LEEZN2_kWhm2 = LEEZN2_kWh / ZN2area;
            double HeaEnerZN3_kWhm2 = HeaEnerZN3_kWh /
ZN3area;
            double CoolEnerZN3_kWhm2 = CoolEnerZN3_kWh
/ ZN3area;
            double LEEZN3_kWhm2 = LEEZN3_kWh / ZN3area;
            double HeaEnerZN4_kWhm2 = HeaEnerZN4_kWh /
ZN4area;
            double CoolEnerZN4_kWhm2 = CoolEnerZN4_kWh
/ ZN4area;
            double LEEZN4_kWhm2 = LEEZN4_kWh / ZN4area;

            double THL_KWhm2 = HeaEnerCoreZN_kWhm2 +
HeaEnerZN1_kWhm2
                        +       HeaEnerZN2_kWhm2       +
HeaEnerZN3_kWhm2 + HeaEnerZN4_kWhm2;
            double TCL_KWhm2 = CoolEnerCoreZN_kWhm2 +
CoolEnerZN1_kWhm2
                        +      CoolEnerZN2_kWhm2      +
CoolEnerZN3_kWhm2 + CoolEnerZN4_kWhm2;
            double   TLEE_KWhm2   =   LEECoreZN_kWhm2   +
LEEZN1_kWhm2 + LEEZN2_kWhm2
                        + LEEZN3_kWhm2 + LEEZN4_kWhm2;

            double TEL_KWhm2 = THL_KWhm2 + TCL_KWhm2 +
TLEE_KWhm2;

            System.out.println("Total Energy Load is "
+ TEL_KWhm2 + "[KWh/m2].");
            System.out.println("Total   Heating   Energy
Load is " + THL_KWhm2
                        + "[KWh/m2].");
```

```
            System.out.println("Total  Cooling  Energy
Load is " + TCL_KWhm2
                        + "[KWh/m2].");
            System.out.println("Total  Lights  Electric
Load is " + TLEE_KWhm2
                        + "[KWh/m2].");

            System.out.println("Total  Energy  Load  of
the Core Zone is "
                        + TCoreZN_KWhm2 + "[KWh/m2].");
            System.out.println("Total  Energy  Load  of
the Perimeter Zone1 is "
                        + TZN1_KWhm2 + "[KWh/m2].");
            System.out.println("Total  Energy  Load  of
the Perimeter Zone2 is "
                        + TZN2_KWhm2 + "[KWh/m2].");
            System.out.println("Total  Energy  Load  of
the Perimeter Zone3 is "
                        + TZN3_KWhm2 + "[KWh/m2].");
            System.out.println("Total  Energy  Load  of
the Perimeter Zone4 is "
                        + TZN4_KWhm2 + "[KWh/m2].");

            System.out
                        .println("The  optimized  value
for the Windows Dimension of the Perimeter Zone1 is "
                                    +   WinDimZone1   +
"[m].");
            System.out
                        .println("The  optimized  value
for the Windows Dimension of the Perimeter Zone2 is "
                                    +   WinDimZone2   +
"[m].");
            System.out
                        .println("The  optimized  value
for the Windows Dimension of the Perimeter Zone3 is "
```

```
                                   +   WinDimZone3   +
"[m].");
            System.out
                        .println("The  optimized  value
for the Windows Dimension of the Perimeter Zone4 is "
                                   +   WinDimZone4   +
"[m].");

            System.out
                        .println("The  optimized  value
for the Overhangs Depth of the Perimeter Zone1 is "
                                   +  OverhangDepthZN1
+ "[m].");
            System.out
                        .println("The  optimized  value
for the Overhangs Depth of the Perimeter Zone2 is "
                                   +  OverhangDepthZN2
+ "[m].");
            System.out
                        .println("The  optimized  value
for the Overhangs Depth of the Perimeter Zone3 is "
                                   +  OverhangDepthZN3
+ "[m].");
            System.out
                        .println("The  optimized  value
for the Overhangs Depth of the Perimeter Zone4 is "
                                   +  OverhangDepthZN4
+ "[m].");

            System.out
                        .println("The  optimized  value
for the Fins Depth of the Perimeter Zone1 is "
                                   +  FinDepthZone1  +
"[m].");

            System.out
```

```
                        .println("The  optimized  value
for the Fins Depth of the Perimeter Zone2 is "
                                    +  FinDepthZone2  +
"[m].");

            System.out
                        .println("The  optimized  value
for the Fins Depth of the Perimeter Zone3 is "
                                    +  FinDepthZone3  +
"[m].");

            System.out
                        .println("The  optimized  value
for the Fins Depth of the Perimeter Zone4 is "
                                    +  FinDepthZone4  +
"[m].");

            in.close();

            try {
                  FileWriter      fwr      =      new
FileWriter(fileOut, true);
                  BufferedWriter     bfr     =     new
BufferedWriter(fwr);
                  bfr.newLine();
                  PrintWriter      pwr      =      new
PrintWriter(bfr);
                  pwr.print("  ;Facade  Geometry;"  +
ObjectiveFunction + ";"
                              +  TEL_KWhm2  +  ";"  +
THL_KWhm2 + ";" + TCL_KWhm2 + ";"
                              +  TLEE_KWhm2  +  ";"  +
TCoreZN_KWhm2 + ";" + TZN1_KWhm2 + ";"
                              +  TZN2_KWhm2  +  ";"  +
TZN3_KWhm2 + ";" + TZN4_KWhm2 + ";"
```

```
                              +  HeaEnerCoreZN_kWhm2  +
";" + CoolEnerCoreZN_kWhm2 + ";"
                              + LEECoreZN_kWhm2 + ";" +
HeaEnerZN1_kWhm2 + ";"
                              + CoolEnerZN1_kWhm2 + ";"
+ LEEZN1_kWhm2 + ";"
                              +  HeaEnerZN2_kWhm2  +  ";"
+ CoolEnerZN2_kWhm2 + ";"
                              +  LEEZN2_kWhm2  +  ";"  +
HeaEnerZN3_kWhm2 + ";"
                              + CoolEnerZN3_kWhm2 + ";"
+ LEEZN3_kWhm2 + ";"
                              +  HeaEnerZN4_kWhm2  +  ";"
+ CoolEnerZN4_kWhm2 + ";"
                              +  LEEZN4_kWhm2  +  ";"  +
WinDimZone1 + ";" + WinDimZone2
                              + ";" + WinDimZone3 + ";"
+ WinDimZone4 + ";"
                              +  OverhangDepthZN1  +  ";"
+ OverhangDepthZN2 + ";"
                              +  OverhangDepthZN3  +  ";"
+ OverhangDepthZN4 + ";"
                              +  FinDepthZone1  +  ";"  +
FinDepthZone2 + ";" + FinDepthZone3
                              +  ";"  +  FinDepthZone4  +
";");
                  pwr.close();
            } catch (IOException ex) {

      Logger.getLogger(FacadeGeometry.class.getName())
.log(Level.SEVERE,
                              null, ex);
            }
      }
}
```

Curriculum Vitae

Educations

2012 – 2018 **Ph.D.:** Architecture-Building physics, Vienna University of Technology, Vienna, Austria.
Dissertation: “A Computational Environment for Building Enclosure Systems Design and Control Support via Dynamic Simulation-Assisted Optimization”.
Supervisor: Univ. Prof. DI. Dr. tech. Ardeshir Mahdavi

1997 - 2004 **Bachelor and Master:** Architecture, Imam Khomeini International University, Qazvin, Iran.
Master thesis: “Design of Culture and Art Center of Zahedan City”
Advisor: Univ. Prof. Dr. Abbasali Izadi

1993 - 1997 **Secondary Education Diploma:** Mathematics and Physics, Sistan & Balouchestan University High School, Zahedan, Iran.

Work Experiences

2006 - 2011 Faculty member of Dept. of Architecture, University of Zabol, Zabol, Iran.

2005 - 2011 Guest lecturer at Zahedan Azad University, Zahedan, Iran.

2006 Guest lecturer at Official Management Education Centre, Zahedan, Iran.

2006 Guest lecturer at Zahedan Payam Nour University, Zahedan, Iran.

2005 - 2006 Faculty member of Dept. of Architecture, Tehran Azad University, Tehran, Iran.

2007 - 2011	Head of the department of Architecture, University of Zabol.
2008 - 2011	Member of the Research Group for Usage of new energy methods, University of Zabol.
2008 - 2011	Head of Architecture Research Group, Archeology Research Center, University of Zabol.
2007 - 2010	Administration Manager of University of Zabol.
2008 - 2010	Member of the Non-Academic Human Capital committee, University of Zabol.

Language Skills

Persian (mother tongue): Fluent in speaking, reading and writing.

English: Fluent in speaking, reading and writing.

German: intermediate.