https://doi.org/10.7275/17635856 https://scholarworks.umass.edu/dissertations_2/1961

This Open Access Dissertation is brought to you for free and open access by the Dissertations and Theses at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.
A FRAMEWORK FOR PERFORMANCE-BASED FACADE DESIGN:
APPROACH FOR AUTOMATED AND MULTI-OBJECTIVE SIMULATION
AND OPTIMIZATION

A Dissertation Presented

by

MAHSA MINAEI

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2020

Department of Environmental Conservation
A FRAMEWORK FOR PERFORMANCE-BASED FACADE DESIGN:
APPROACH FOR AUTOMATED AND MULTI-OBJECTIVE SIMULATION
AND OPTIMIZATION

A Dissertation Presented

by

MAHSA MINAEI

Approved as to style and content by:

____________________________________
Ajla Aksamija, Chair

____________________________________
L. Carl Fiocchi, Member

____________________________________
Ho-Sung Kim, Member

____________________________________
Mark Hamin, Outside Member

____________________________________
Curt Griffin, Department Head,
Environmental Conservation
ABSTRACT

A FRAMEWORK FOR PERFORMANCE-BASED FACADE DESIGN: APPROACH FOR AUTOMATED AND MULTI-OBJECTIVE SIMULATION AND OPTIMIZATION

MAY 2020

MAHSA MINAEI

Ph.D. UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Ajla Aksamija

Buildings have a considerable impact on the environment, and it is crucial to consider environmental and energy performance in building design. Buildings account for about 40% of the global energy consumption and contribute over 30% of the CO2 emissions. A large proportion of this energy is used for meeting occupants' thermal comfort in buildings, followed by lighting. The building facade forms a barrier between the exterior and interior environments; therefore, it has a crucial role in improving energy efficiency and building performance.

In this regard, decision-makers are required to establish an optimal solution, considering multi-objective problems that are usually competitive and nonlinear, such as energy consumption, financial costs, environmental performance, occupant comfort, etc. Sustainable building design requires considerations of a large number of design variables and multiple, often conflicting objectives, such as the initial construction cost, energy cost, energy consumption and occupant satisfaction. One approach to address these issues is the use of building performance simulations and optimization methods.
This research first investigates and highlights the key research methods, issues and tools associated with building performance simulations and the optimization methods. Then a novel method for improving building facade performance is presented, taking into consideration occupant comfort, energy consumption and energy costs. The dissertation discusses development of a framework, which is based on multi-objective optimization and uses a genetic algorithm in combination with building performance simulations. The framework utilizes EnergyPlus simulation engine and Python programming to implement optimization algorithm analysis and decision support. The framework enhances the process of performance-based facade design, couples simulation and optimization packages, and provides flexible and fast supplement in facade design process by rapid generation of design alternatives. The dissertation describes the components and functionality of this framework in detail, as well as two-step optimization technique which is a new technique that combines GA and machine learning. The dissertation also presents results and validation techniques and provides conclusions of the study.
TABLE OF CONTENTS

ABSTRACT .................................................................................................................... iv

LIST OF TABLES .......................................................................................................... ix

LIST OF FIGURES ....................................................................................................... xi

NOMENCLATURE ...................................................................................................... xv

CHAPTER

1. INTRODUCTION ...................................................................................................... 1

1.1 Research Background ............................................................................................ 1

1.2 Role of Facade Design in High Performance Building ........................................ 4

1.3 Role of Building Performance Simulation in Design Process ............................ 5

1.3.1 Building Simulation Model Classifications ....................................................... 8
1.3.2 Building Simulation Modeling Issues ............................................................... 10
1.3.3 Building Simulation Tools and Categories ....................................................... 12

1.4 Role of Optimization in Design Process ............................................................ 20

1.4.1 Classification of building optimization properties ........................................... 22
1.4.2 Overview on optimization algorithms in building energy simulation and analysis ... 24
1.4.3 Building optimization methods ......................................................................... 28

1.5 Purpose and Outline of the Dissertation ............................................................ 34

2. LITERATURE REVIEW .......................................................................................... 36

2.1 Simulation and Optimization for Performance-Based Design: Overview of Tools, Applications and Methods ................................................................. 36

2.2 Building Performance Simulation Studies in Design Process .............................. 47

2.2.1 Whole Building Simulation Models’ Literature ................................................ 47
2.2.2 Lighting and Daylight Simulation Literature .................................................... 49
2.3 Building Performance Optimization Studies in Design Process .........................50
2.4 Occupant Comfort (Thermal Comfort and Visual Comfort) .............................55
2.5 Summary of Literature .......................................................................................59

3. RESEARCH OBJECTIVES AND PROBLEM STATEMENT .............................68

3.1 Current Gaps in Knowledge (Research Problems) ..............................................68
3.2 Research Objectives & Questions ......................................................................70
   3.2.1 Research Questions for Objective 1 .................................................................71
   3.2.2 Research Questions for Objective 2 .................................................................71
3.3 Model Development and Overview of the Framework .......................................72

4. RESEARCH METHODOLOGY .............................................................................83

4.1 Overview of Research Methodology ....................................................................83
   4.1.1 Research Methods for Objective 1 .................................................................83
   4.1.2 Research Methods for Objective 2 .................................................................84
4.2 Step 1: Defining Goals, Performance Criteria and Facade Design Variables ......87
4.3 Step 2: Creating the Database (MYSQL) ............................................................96
4.4 Step 3: Coupling Python Script with Simulation Engine (EnergyPlus) ..............104
   4.4.1 Modeling in EnergyPlus (Simulation Engine) .................................................105
   4.4.2 Wall Assemblies (Cladding & Framing) ..........................................................105
   4.4.3 Insulation (Cavity & Continuous) .................................................................107
   4.4.4 Glazing Systems (Glass & Frame Specification) ..........................................109
   4.4.5 Solar Control (Shading) ................................................................................110
   4.4.6 Daylight Simulation ......................................................................................112
   4.4.7 Cost Simulation .............................................................................................114
4.5 Step 4: Filtering and Narrowing Down the Results by Implementing Python
   Script, Genetic Algorithm (GA) and Machine Learning Methods .....................115
   4.5.1 Optimization Algorithm - Genetic Algorithms ..........................................117
   4.5.2 Machine Learning & Artificial Neural Network (ANN) .............................119

5. RESULTS .............................................................................................................136
5.1 Results of the Study and Discussion ................................................................. 136

6. CONCLUSION & FUTURE WORK ....................................................................... 155

6.1 Conclusion ........................................................................................................ 155

6.2 Benefits of the Developed Data-Driven Framework .......................................... 156

6.3 Future Work .................................................................................................... 157

APPENDICES

A. SCRIPTS ............................................................................................................ 160

B. DATABASE ....................................................................................................... 184

C. OUTPUTS ......................................................................................................... 189

BIBLIOGRAPHY .................................................................................................. 204
<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Properties of building optimization problems (Adopted from Si 2019)</td>
<td>23</td>
</tr>
<tr>
<td>2. Comparison of software in terms of fulfilling the requirements of the proposed software framework (Adopted from Ostergard 2016)</td>
<td>39</td>
</tr>
<tr>
<td>3. ASHRAE standard recommendations (Djongyang 2010)</td>
<td>56</td>
</tr>
<tr>
<td>4. Predicted percentage of dissatisfied (PPD) based on the Predicted mean vote (PMV) (Djongyang 2010)</td>
<td>56</td>
</tr>
<tr>
<td>5. Key factors and major findings of reviewed research</td>
<td>59</td>
</tr>
<tr>
<td>6. Facade design variables used in this study, and specific parameters</td>
<td>90</td>
</tr>
<tr>
<td>7. List of facade elements, variables, properties and simulation tools used for design scenarios</td>
<td>92</td>
</tr>
<tr>
<td>8. List of fixed parameters</td>
<td>93</td>
</tr>
<tr>
<td>9. AHU features</td>
<td>94</td>
</tr>
<tr>
<td>10. Central plant features</td>
<td>95</td>
</tr>
<tr>
<td>11. Cladding material properties for EnergyPlus inputs</td>
<td>106</td>
</tr>
<tr>
<td>12. Equivalent insulation thickness and thermal properties</td>
<td>108</td>
</tr>
<tr>
<td>13. Glazing systems (glass &amp; frame specifications)</td>
<td>109</td>
</tr>
<tr>
<td>14. Shading control (exterior shading)</td>
<td>111</td>
</tr>
<tr>
<td>15. Illuminance level analysis</td>
<td>113</td>
</tr>
<tr>
<td>16. Daylight Glare Index analysis</td>
<td>113</td>
</tr>
<tr>
<td>17. Indicators, ANN input and weight values</td>
<td>122</td>
</tr>
</tbody>
</table>
18. Minimum and maximum value scenarios for indicators to dynamically update the
rang of results. ........................................................................................................130

19. Indicators, ANN input and weight values. ..............................................................138

20. The result of different weight bias and number of scenarios selected for each
type ......................................................................................................................138

21. The result of optimization showing the characteristic of optimum scenarios
(Cluster 1). ..............................................................................................................144

22. The result of optimization showing the characteristic of optimum scenarios
(Cluster 2). ..............................................................................................................145

23. The result of optimization showing the characteristic of optimum scenarios
(Cluster 3). ..............................................................................................................145

24. Range of 3 cluster outputs results for weight bias 25-35. ......................................146

25. The result of optimization and the characteristic of optimum scenarios
(cluster B). .............................................................................................................149

26. The result of optimization and the characteristic of optimum scenarios
(cluster C). .............................................................................................................149

27. The result of optimization showing the characteristic of optimum scenarios
(cluster 1). .............................................................................................................151

28. The result of optimization showing the characteristic of optimum scenarios
(cluster 2). .............................................................................................................152

29. The result of optimization showing the characteristic of optimum scenarios
(cluster 3). .............................................................................................................152

30. Range of 3 clusters outputs results for weight bias 30-30 ....................................153
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Energy consumption by sectors (Adapted from Attia 2013)</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>The standard approach to simulation</td>
<td>7</td>
</tr>
<tr>
<td>3.</td>
<td>Black, grey and white box models (Adopted from deWilde and Augenbroe 2018)</td>
<td>10</td>
</tr>
<tr>
<td>4.</td>
<td>Radiance and growing number of public- and private-sector tools</td>
<td>17</td>
</tr>
<tr>
<td>5.</td>
<td>General flowchart of the computational optimization (Adopted from Kheiri 2018)</td>
<td>29</td>
</tr>
<tr>
<td>6.</td>
<td>General flowchart of Genetic Algorithm process (Adopted from Kheiri 2018)</td>
<td>32</td>
</tr>
<tr>
<td>7.</td>
<td>Histogram of utilized optimization methods (Adopted from Kheiri 2018)</td>
<td>42</td>
</tr>
<tr>
<td>8.</td>
<td>Frequency of the use of building performance simulation tools for building performance research (Adopted from Kheiri 2018)</td>
<td>43</td>
</tr>
<tr>
<td>10.</td>
<td>Participants’ choice of optimization variables (Adapted from Attia 2013)</td>
<td>44</td>
</tr>
<tr>
<td>11.</td>
<td>Participants’ choice of objective functions (Adapted from Attia 2013)</td>
<td>45</td>
</tr>
<tr>
<td>12.</td>
<td>Participants’ choice of constraints (Adapted from Attia 2013)</td>
<td>45</td>
</tr>
<tr>
<td>13.</td>
<td>Adaptive Comfort Standard (ACS) for ASHRAE Std. 55, applicable for naturally ventilated buildings</td>
<td>57</td>
</tr>
<tr>
<td>14.</td>
<td>MVP diagram</td>
<td>73</td>
</tr>
<tr>
<td>15.</td>
<td>Data flow diagram, level 0</td>
<td>74</td>
</tr>
<tr>
<td>Number</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>16</td>
<td>Basic MVC architecture.</td>
<td>75</td>
</tr>
<tr>
<td>17</td>
<td>MVC flow chart.</td>
<td>77</td>
</tr>
<tr>
<td>18</td>
<td>Procedure for applying and coupling elements of the developed framework in this research</td>
<td>78</td>
</tr>
<tr>
<td>19</td>
<td>Web-application sign in page.</td>
<td>80</td>
</tr>
<tr>
<td>20</td>
<td>Web-application main page, which allows users to select specific parameters.</td>
<td>81</td>
</tr>
<tr>
<td>21</td>
<td>Web-application result page.</td>
<td>82</td>
</tr>
<tr>
<td>22</td>
<td>The Framework Back-End Diagram (Steps 1-4)</td>
<td>85</td>
</tr>
<tr>
<td>23</td>
<td>Major steps (1-5) and tasks in the study.</td>
<td>86</td>
</tr>
<tr>
<td>24</td>
<td>Conceptual diagram, showing components of the framework.</td>
<td>88</td>
</tr>
<tr>
<td>25</td>
<td>Basic procedure for applying the framework in this research.</td>
<td>89</td>
</tr>
<tr>
<td>26</td>
<td>Different visual configurations of test cell.</td>
<td>96</td>
</tr>
<tr>
<td>27</td>
<td>Data Flow Diagram (DFD) for the framework implementation.</td>
<td>99</td>
</tr>
<tr>
<td>28</td>
<td>Part of scripts to create and permute all scenarios table in MYSQL.</td>
<td>99</td>
</tr>
<tr>
<td>29</td>
<td>Part of scripts to create table for outputs in MYSQL.</td>
<td>100</td>
</tr>
<tr>
<td>30</td>
<td>Scripts used to couple database with EnergyPlus output results table, and to save selected output to the database.</td>
<td>101</td>
</tr>
<tr>
<td>31</td>
<td>Scripts used to save selected outputs from EnergyPlus report table to save into database table.</td>
<td>102</td>
</tr>
<tr>
<td>32</td>
<td>MYSQL-all scenarios table.</td>
<td>103</td>
</tr>
<tr>
<td>33</td>
<td>MYSQL-output table.</td>
<td>103</td>
</tr>
<tr>
<td>34</td>
<td>Scripts used to define WWR input and couple simulation model to the EnergyPlus engine.</td>
<td>107</td>
</tr>
</tbody>
</table>
35. Scripts used to couple glazing system input into the EnergyPlus engine. ..........110
36. Scripts used to define solar control and insulation inputs and coupling with
   EnergyPlus engine. .................................................................................................112
37. Artificial Neural Network diagram. .....................................................................120
38. Total EUI-Electricity, EUI-Gas, PMV and Energy Cost indicator scores.............124
39. Optimization flow chart, illustrating the application of GA and ANN.................126
40. Scripts used to calculate total points and decide whether to keep or skip the
   scenarios. ................................................................................................................127
41. Scripts used to calculate Energy Consumption, comfort and cost, and to assign
   weight bias..............................................................................................................129
42. Total Indicators vs. Scenario IDs (for 2,061 scenarios). ......................................132
43. Total Indicators vs. Scenario IDs (for 18,103 scenarios). ....................................132
44. Correlation matrix. ............................................................................................134
45. Results for all scenarios with applying batch normalization and flood field algorithm
   (Phase 1 optimization, flood fill+ Batch normalization). .......................................134
46. Results for all scenarios with applying correlation matrix and cluster eliminating
   (Phase 2 optimization)..........................................................................................135
47. Part of scripts to create web application as the front-end and interface...............137
48. Weight Bias (35-20) results for total indicator scores 70 and greater..................139
49. Weight Bias (35-25) results for total indicator scores 70 and greater..................139
50. Weight Bias (15-30) results for total indicator scores 70 and greater...............140
51. Weight Bias (30-30) results for total indicator scores 70 and greater...............140
52. Weight Bias (17-35) results for total indicator scores 70 and greater...............140
53. Weight Bias (20-35) results for total indicator scores 70 and greater....................141
54. Weight Bias (25-35) results for total indicator scores 70 and greater....................141
55. Selected scenarios for weight bias (25-35) that have total indicator scores 75 or
greater................................................................................................................................143
56. Selected scenarios for weight bias (25-35) that have total indicator scores 77 or
greater and illustration of 3 selected output clusters....................................................143
57. Selected scenarios for weight bias (25-35) that have total indicator 79 or greater (top
selected scenarios-Cluster 1)............................................................................................144
58. Weight Bias (30-30) results for total indicator scores 77 and greater....................148
59. Selected scenarios for weight bias (30-30) that have total indicator scores 75 or
greater................................................................................................................................150
60. Selected scenarios for weight bias (30-30) that have total indicator scores 77 or
greater and illustration of 3 selected output clusters....................................................150
61. Selected scenarios for weight bias (30-30) that have total indicator scores 79 or
greater (top selected scenarios-Cluster 1). ........................................................................151
62. Example of results and outputs, which can be exported as Excel files......................154
## NOMENCLATURE

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEC</td>
<td>Architecture, Engineering, and Construction</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>BIM</td>
<td>Building Information Modeling</td>
</tr>
<tr>
<td>BPA</td>
<td>Building Performance Analysis</td>
</tr>
<tr>
<td>BPS</td>
<td>Building Performance Simulation</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma-Separated Value</td>
</tr>
<tr>
<td>DOE</td>
<td>Department of Energy</td>
</tr>
<tr>
<td>DGI</td>
<td>Daylight Glare Index</td>
</tr>
<tr>
<td>EA</td>
<td>Evolutionary Algorithm</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>gbXML</td>
<td>Green Building eXtended Markup Language</td>
</tr>
<tr>
<td>HVAC</td>
<td>Heating, Ventilating, and Air Conditioning</td>
</tr>
<tr>
<td>IFC</td>
<td>Industry Foundation Classes</td>
</tr>
<tr>
<td>LEED</td>
<td>Leadership in Energy and Environmental Design</td>
</tr>
<tr>
<td>MOEA</td>
<td>Multi-Objective Evolutionary Algorithms</td>
</tr>
<tr>
<td>MOO</td>
<td>Multi-Objective Optimization</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>Non-dominated Sorting Genetic Algorithm-II</td>
</tr>
<tr>
<td>PMV</td>
<td>Predicted Mean Vote</td>
</tr>
<tr>
<td>PPD</td>
<td>Predicted Percentage of Dissatisfied</td>
</tr>
<tr>
<td>R</td>
<td>Thermal Resistance</td>
</tr>
<tr>
<td>VT</td>
<td>Visible Transmittance</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>U</td>
<td>Heat Transfer Coefficient U</td>
</tr>
<tr>
<td>VLT</td>
<td>Visible Light Transmittance</td>
</tr>
<tr>
<td>WWR</td>
<td>Window to Wall Ratio</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Research Background

Energy use of buildings and the release of carbon greenhouse gases associated with their operations are one of the largest contributors to climate change, which is considered as a major problem for the built environment. The buildings and construction sector combined are responsible for 36% of global final energy consumption and nearly 40% of total direct and indirect CO₂ emissions (IEA 2019). Energy demand for building operations continues to rise, driven by improved access to energy in developing countries, greater ownership and use of energy-consuming devices, and rapid growth in global buildings floor area, at nearly 3% per year (IEA 2019).

Climate change refers to a change in the state of the climate that can be identified (e.g. using statistical tests) by changes in the mean and/or the variability of its properties, and which persists for an extended period, typically decades or longer. It refers to any change in climate over time, whether due to natural variability or human activity. Global climate change, depletion of natural energy resources, building occupants’ needs and comfort expectations, increasing awareness of the relation between indoor environmental quality and health and wellbeing of occupants and their productivity, have increased the demand of integrated approach for high-performance design to manage all these aspects. The current trend of designing high-performance,
Energy-efficient buildings has increased the need for tools, frameworks and methods that can aid the decision-making process.

Energy consumption is usually divided into three main sectors: industry, transport, and other uses. Buildings’ energy consumption is considered as the third group. Growth in population, increasing demand for building services and comfort levels, together with time spent inside buildings, contribute to increasing energy demand in the building sector, which has raised building energy consumption beyond the level of transport and industry. For this reason, energy efficiency in buildings is currently a prime objective for energy policy at regional, national and international levels. Buildings represent a large portion of the world's energy consumption and associated CO₂ emissions. For example, based on recent studies the building sector represents 39% of the energy consumption and 38% of the CO₂ emissions in the U.S. (Amasyali 2018).

A large proportion of building energy consumption is used to regulate thermal conditions in buildings. Sources of consumption vary between residential and commercial buildings. Energy consumption for heating, lighting, and cooling are more than other sectors (Figure 1). The building envelope forms a barrier between the exterior and interior environments, and has a crucial role in improving energy efficiency and building performance. Improvement of facade design can reduce energy needs associated with cooling, heating and lighting. Therefore, this research focuses on performance-based facade design, appropriate simulation and optimization tools and methods for design analysis and support.
Figure 1: Energy consumption by sectors (Adapted from Attia 2013).

Building performance simulation (BPS) provides relevant design information by indicating potential (quantifiable) directions for design solutions. BPS tools and applications improve design decision-making by providing quantifiable data about building performance. BPS tools are an integral part of the design process for energy efficient and high-performance buildings, since they help in investigating design options and assess the environmental and energy impacts of design decisions (Attia 2013). The important aspect is that simulation does not generate design solutions, instead, it supports designers by providing feedback on performance results of different design scenarios.

Optimization is a method for finding a best scenario with highest achievable performance under certain constraints and variables. There are different methods for
optimization, requiring use of computational simulations to achieve an optimal solution, or sometimes requiring analysis or experimental methods to optimize building performance without performing mathematical optimization. But in BPS context, the term optimization generally indicates an automated process that is entirely based on numerical simulation and mathematical optimization (Nguyen 2014). Integrating BPS and optimization methods can form a process for selecting optimal solutions from a set of available alternatives for a given design problem, according to a set of performance criteria.

1.2 Role of Facade Design in High Performance Building

The facade is the primary barrier between an interior and exterior environment of any building, as such greatly influences energy efficiency, building performance and occupant comfort. Highest energy consumption for different types of buildings is consumed by the HVAC systems, followed by lighting systems. Space heating, space cooling, water heating and lighting account for close to 70% of site energy consumption (Harish 2016). Therefore, heating, cooling, and lighting are the main domain to focus for reducing energy consumption in buildings. The energy requirements for heating, cooling, and lighting of the building are strongly driven by the facade performance, especially the glazing part. Glazing systems for buildings and windows have always been a critical element in facade design, and have significant impact on occupants, energy use and energy costs. Glazing parts provide view but must control glare, they allow daylight to enter but must control solar transmittance and cooling loads. Glazing systems provide a degree of connection with the outdoors and impact psychological comfort, but they must also maintain physical comfort in terms of temperature and solar
extremes. They provide natural ventilation but admit undesired air leakage and can create drafts if not properly installed and insulated. Most glazing and windows must respond over a wide range of climate and use conditions. Integrated facade design for high-performance buildings must control and consider these features. With active and passive strategies, and the ability to respond to the changes for facade design, building performance can be improved and contribute to energy use and carbon released reduction for heating and cooling and enhancement of indoor environment quality (IEQ).

1.3 Role of Building Performance Simulation in Design Process

The role of simulations in design process has evolved, and simulation models are used in different design phases to predict energy consumption of buildings and occupants’ comfort levels. These methods are used at the conceptual, schematic and design development phases to optimize building performance, during the occupancy phase to monitor and control the performance and during the retrofit to decide about the benefits of different alternatives and interventions. Therefore, understanding the effects of design decisions and outlining a framework in which the simulation models should be used is crucial to achieve high levels of performance.

Simulation is an integral part of measuring and quantifying performance criteria. Defining the interface between physical building element and performance criteria plays an important role. For instance, the existing building or the reference building (i.e., in case of new construction) can be defined in BPS software programs, including thermal envelope and the HVAC systems, operation, schedules, material properties, etc. Then,
the parameters that most affect the energy performance can be identified as design variables, such as different materials, efficiencies of HVAC system, characteristics of thermal envelope, etc.

The biggest challenge of simulation in performance-based design is to provide a variety of normative calculations when an advanced simulation cannot provide a more accurate answer, either because of the presence of uncertainties, the lack of available information, or the context of decision that demands it (Hensen 2012).

Building simulations started to stand out as a separate discipline in the late 1970’s. This area of building science research matured since then into a field that offers unique expertise, methods, and tools for building performance evaluation. The early groundwork was done in the 1960’s and 1970’s, mainly in the energy performance field, followed by an expansion into other fields such as lighting, HVAC systems, air flow, and others. More recent additions relate to combined heat, air and moisture flow, heat transfer, mass transfer, acoustics, control systems, and various combinations with urban and microclimate simulations. Building simulations are the key tool to quantify performance criteria for decision making. Building simulation models predict the building behavior under specific scenario (Augenbroe 2019). Figure 2 shows relationships between the physical world (reality), the model used to represent the physical world, and simulations, which are used to investigate the behavior of the model under certain conditions. The results of simulations reveal a behavior related to performance under study. The words ‘modeling’ and ‘simulation’ are sometimes interchanged in discussions; generally, a model is an abstraction or description of reality where simulation is the execution of that model in the computer (Incose 2015).
Computational building performance modeling and simulations are multidisciplinary, problem-oriented and wide in scope. Simulations are one of the most powerful analysis methods for a variety of problems, but they do not provide solutions or answers. Instead, simulation results support user understanding of complex systems by providing (relatively) rapid feedback on the performance implications of designs (Clarke 2015).

BPS applications provide relevant design information by indicating potential (quantifiable) directions for design solutions. BPS tools and applications facilitate the process of design decision making by controlling the values and simulations. Building performance simulations are an integral part of the design process for energy efficient and high-performance buildings since they help in investigating design options and assess the environmental and energy impacts of design decisions (Augenbroe 2019). There are many available tools, and they can evaluate different aspects of building performance, such as capital and operating costs; energy performance and demand; human comfort, health and productivity; illumination; electrical flows; water and waste; acoustic design; renewable energy; and atmospheric emissions. Because the number of
simulation tools and methods is large, this research will focus on human factors (thermal and visual comfort), energy use and energy cost.

### 1.3.1 Building Simulation Model Classifications

Simulation models have wide spectrum of capability and come in different form. They can run on the basis of a short list of inputs or require substantial modeling effort; some are useful for research and academia while other are provided as commercial tools with support and training provisions. The underlying mathematical models may be linear or nonlinear, static or dynamic, discrete or continuous and deterministic or stochastics. Simulation models may cover different physical process such as heat, mass transfer, acoustics, light or structural behavior. They may represent small part of the building, such as one-dimensional section through a wall or complex combination of geometry, materials and systems for a full building.

Other way of classifying building simulation models is by their temporal dimension and differences between stationary, semi-dynamic (quasi-stationary) and transient models. For instance, in thermal models a stationary model of heat transport through a wall requires the mathematical solution of one set of equations, in semi-dynamic model one may for instance use 12 sets of conditions to represent the 12 months of the year, which means the same equations need to be solved twelve times. By moving to transient conditions, often studied by looking at hourly values, this increases to 8760 steps per year (de Wilde 2018).

There are other ways to categorize simulation models, one of them is related to system identification and distinguishes between black box, grey box and white box
models. These models link a series of inputs with outputs. In black box model, machine learning is the ration between input and outputs, which captures the correlation by learning from correlated data pairs. The internal working of the system that cause this correlation, is unknown so the typical techniques used are regression analysis and neural networks. In grey box model, there is certain insight in the internal working system that allows to do predictions; however, these models still have some unknown properties and relationships that need to be estimated. In the white box model, all properties and relations are known and allow to clear and in detail modeling of the relation between inputs and outputs. Sometimes these types of models are known as glass models and represent the state of art in certain domain. Figures below show these models and their workflows. Further discussion can be found in Zhao and Magoules (2012), Fumo (2014), and de Wilde and Augenbroe (2018).
1.3.2 Building Simulation Modeling Issues

The process of modeling simplifies the real-world problems by applying variables of interest and enhances management of complex situations when dealing with full complexity in decision making process. This aspect of modeling applies to building and building systems, and complex patterns of external and internal actions in building. But sometimes using these models is not appropriate. The list below includes several cases that would be better without using modeling process (Basmadjian 2003):
1) If an answer is needed within hours only, leaving no time for a serious
modelling effort.

2) Where a simple and inexpensive experiment might provide the answer.

3) If the client is suspicious of theory and can be better convinced by physical
evidence.

4) In cases where the system is too complex for meaningful modelling.

5) Whenever the answer is self-evident.

6) Where modelling will fail to provide any new or meaningful insights.

Beside all advantages of building simulations, there are some inherent issues and
challenges, which are described in more detail below.

- **Uncertainty**: The concept of simulation involves a combination of certainty
and uncertainty. This can be related to the complexity of the system that is
simulated. In building design, there are many external aspects that are
subject to uncertainties. The subjects that have most influence on the
uncertainties are weather and climate, properties of material of building
elements and behavior of occupants. There are methods that have been
developed to estimate the extent of these factors in building assessment.

- **Performance gap**: Performance gap is the gap between predicted and
measured performance. There is a range of underlying factors, including
fundamental model uncertainties, tool deficiencies and training issues that
aggravated by complexity of the building design, construction and operation
process. Calibration techniques and training techniques are important steps
to decrease this gap.
• **Occupant behavior:** Predicting occupant presence and activities is a complex matter. A mismatch between modeled and actual occupant behavior leads to divergences as large as 40% (Duarte 2015). International Energy Agency (IEA) Annex 66 tried to explore some of these issues (Annex 66 IEA 2016). In brief, Annex 66 proposed scientific approaches to reduce the gap between the simulated and measured values for building energy performance by representing occupant behavior in a standardized quantitative way, and went further by integrating them with current building performance simulation programs, which could have important impacts on the industry from various perspectives. (Annex 66 IEA 2016)

• **Exchange data between different digital environment:** Sharing data between simulation domain and wider building information modeling (BIM) is still a challenge. Creating interoperable models that exchanges data does not contain the specific information needed to fully define appropriate digital experiments (Augenbrore 2015).

• **Evolution of buildings and systems:** Unusual shapes, complex shading systems, radiant barriers, complex HVAC systems and innovative technologies may push the borders of simulation and may even require the development of new models (Efficiency Evaluation Organization 2014).

### 1.3.3 Building Simulation Tools and Categories

This section explains simulation techniques according to physical aspects that are simulated. Furthermore, there are number of categories that do not fit the ordering
by physical aspect; but are related to this research, coupled simulation that combines domains and the area of tool interfaces, shells and environments. The building energy software tools directory (IBPSA-USA 2016) listed information on a wide range of building performance simulation tools.

1.3.3.1 Whole building thermal simulation

Thermal performance of whole building is an important area of building performance simulation. This domain is the main tool for thermal comfort condition analysis and studies related to human health and well-being. Sometimes the whole building thermal simulation is named building energy simulation in research and literature. Thermal models are typically well developed and have strong interaction with other building domains such as occupant behavior, building materials, HVAC systems and control system. Three main heat transfer principles (conduction, convection and radiation) are taken into account by heat and mass balance equations in thermal building models. These mechanisms occur in different combinations within the five main energy flow paths in buildings:

- Transmission (heat flow through walls, windows, doors, floors and roof)
- Ventilation (heat flow related to movement of air)
- Solar radiation (heat entering the building windows or glazed surfaces)
- Energy storage (where heat is added or removed from building parts such as concrete walls and floors)
• Internal gain (heat produced by occupants, lights, appliances and building services).

Modeling buildings for thermal analysis usually starts with defining building geometry, considering zoning decisions (such as individual rooms being modelled as one larger space). Walls, facades, roofs and floors are modeled in order of layers, layer thickness and material properties of each layer such as conductivity, density and thermal capacity. For glazing parts, reflectance, absorption and transmission values are applied to the model. Building systems can be modeled at different level of details but the simplest form is as instantaneous heat or cooling loads, and it is possible to include detailed models of boilers, furnaces, distribution systems or operation details. Modeling occupants has also various level of detail, but must represent heat emitted by human bodies in the room and the required comfort conditions. Weather conditions is assigned as weather data, which includes outdoor temperature, relative humidity, wind direction and speed and solar irradiance. Most of the weather files have an hourly format and cover a duration of one year. Modelling has to include the definition of heat transfer coefficient, air flow and boundary conditions (adiabatic boundary condition is often applied to walls, floors that have similar temperatures at both sides like internal walls). Sometime, spatial discretization (such as sub-layers to represent building fabric) and temporal discretization (time step) need to be selected for the model.

The most well-known whole building thermal simulation tools today are EnergyPlus, ESP-r, IDA-ICE and TRANSYS. Other tools, including some legacy applications, are BLAST, BSim, DOE-2, Capsol, COMFIE, DeST, HAP, PowerDomus, Sunrel and TAS (de Wilde 2018).
1.3.3.2 Lighting simulation

The subject of lighting is an important domain in simulation. Daylighting, because of the changes to outdoor conditions caused by weather, can be challenging to simulate. The complexity of lighting simulation is due to the interaction between outdoor and indoor environment, occupant and system behavior and some performance aspects that are related to psychology and perception, such as visual performance and comfort. There are criteria for daylighting such as illuminance level of task areas and surrounding, uniformity of light, color rendition, contrast, absence of glare and flickering that influence indoor environmental quality, occupant’s well-being and building assessment.

One of the main lighting metrics is daylight factor (DF), the horizontal illuminance level and the horizontal to vertical illuminance ratio. And two main underlying methods for lighting simulation are raytracing and radiosity. Raytracing follow the path of light, and there are different subcategories for it. Forward raytracing starts at the light source; backward raytracing starts from the point of measurement or observation then follows the path towards the light source. Bidirectional raytracing combines forward and backward raytracing.

Radiosity method work similar to thermal models of radiation exchange. They split the surface of entities of interest into “patches” and then account for light exchange between these patches (de Wilde 2018). In both ray tracing and radiosity, managing specular and diffuse reflections is important. Raytracing is a good approach for cases
with direct illuminance, shadows, specular reflection and refraction through transparent materials, but it is complicated when dealing with diffuse interreflections.

In lighting simulation, first the geometry model is required and surfaces in the model need to be assigned with relevant properties such as reflectance and transmittance. Artificial light sources, the properties of light sources and illuminance and the operational parameters need to be positioned and assigned to the model. In relation to daylight, blinds and shutters need to be included. Selection of sky model is another important step in modeling. The references are CIE standard sky models (CIE standard overcast sky, CIE intermediate sky and CIE clear sky), the Perez all weather sky and the Matsuura intermediate sky. In some cases, measured lighting data may also be used.

There are different performance metrics in lighting simulation results that are explained briefly. Daylight autonomy (DA) captures the percentages of occupied hours when a certain illuminance threshold is met by daylight. Spatial daylight autonomy (sDA) represents the percentage of area that meets a minimum daylight illumination criterion for specific fraction of operating hours. Useful daylight illuminance (UDI) uses thresholds of 100 and 2000 lux to identify when there is too little, enough or too much daylight. Unified glare rating (UGR), Daylight glare index (DGI) and Daylight glare probity (DGP) are performance metrics in simulation results that predict glare or glare avoidance. Illuminance level can be used to develop contour maps and extended towards temporal maps which indicate areas that receive certain amount of light for specific time period.
There are many lighting simulation tools, such as AGi32, Adeline, DElight, DIALux, DIVA, Inspirer, Lightscape, Lightsolve, Lightswitch, Mental Ray, Radiance, Relux, Zemax OpticStudio, 3ds Max and Velux Daylight Visualizer. Among these tools, Radiance is the most widely used engine in lighting, daylighting, and solar control design and is embedded in several free and commercial architectural-engineering design applications, including IES Virtual Environment, DesignBuilder, Sefaira Architecture and etc.

Figure 4: Radiance and growing number of public- and private-sector tools (Adopted from DOE).

1.3.3.3 Heat and Moisture models (HAM)

Sometime the constructions are at the risk of surface or forming condensation or where moisture may cause significant change in thermal properties of the fabric. In these situations, consequences include mold growth, rot, mildew, cracks, swelling and
increased energy loss. Hygrothermal or HAM models consider both thermal and moisture conditions, based on solving the fundamental heat and mass equation. Simulation models of heat, moisture and air transport may be difficult due to the different time scales that each of these occur.

Some of the HAM simulation tools are Delphin, HAM-Tools, HAM-VIE, WUFI. The HAM model combines various domains, so application of generic engineering tools such as Matlab, Simulink and COMSOL would be helpful.

1.3.3.4 Acoustic simulation

Assessments of sound inside and outside the building, and unwanted sounds (noise) are subject of acoustic modeling and simulation. The main underlying acoustical simulation approach includes geometrical models, wave-base models and diffusion equation models. Geometrical models are based on ray acoustics; the concept of sound travelling along lines or rays. The approach in this category includes image source model, particle tracing, ray tracing, pyramid tracing, cone tracing and radiosity models. Wave-based models use partial differential equation to describe wave travelling through any medium. And diffusion equation models describe how the sound energy spread. Acoustical software in building simulations are CATT-Acoustic, Enhanced Acoustic Simulator for Engineers (EASE), Odeon, and LMS Virtual Ray Acoustic.

1.3.3.5 Computational Fluid Dynamics (CFD)

In building simulations, CFD is mainly used to study the airflow. CFD techniques are suitable for issues such as airflow patterns, temperature distributions and
dispersion of contamination. The modeling requires the definition of geometry, which can be a room or series of rooms inside the building or group of buildings and even full urban environment subject to wind, HVAC systems and other system components.

Some of the CFD software packages used in building simulation are ADREA-HF, FloVENT, PHOENICS, STAR-CCM+ and OpenFoam. The generic engineering analysis software ANSYS includes dedicated CFD tools, of which CFX and Fluent are often used for building simulation (de Wilde 2018).

### 1.3.3.6 Pedestrian movement models (evacuation)

The movement of humans inside the building (occupants) and outside (pedestrians) is an important domain in building simulation. Pedestrian movement has two modes: normal movement and emergency situation, which is usually studied for building evacuation. Normal pedestrian movement can be used in studies related to comfort, efficiency and risk assessment of pedestrian safety and evacuation behavior may be studied to reduce certain risks associated with movement.

### 1.3.3.7 Tool interfaces

Some tools provide Graphical User Interface (GUI) for simulations engines. IES-VE, eQuest are examples of these tools. DesignBuilder, OpenStudio and Sefaira are examples of interfaces for EnergyPlus as simulation engine.
1.4 Role of Optimization in Design Process

The role of simulations in the decision process has evolved and simulation models are used to predict energy consumption and comfort level of buildings. Therefore, understanding the effects of design decisions and outlining a framework in which the simulation models should be used is crucial to achieving high level of performance. This framework should ensure the effectiveness of decision making to obtain the highest performance for the lowest possible cost. For this process, there are many conflicting objectives to consider and numbers of possible solutions, which require coupling of design optimization methods with building performance simulations.

Decreasing energy consumption and improving indoor comfort levels are two fundamental, yet conflicting objectives of energy-efficient building design. Moreover, considerations for costs add additional complexity to decision making process. Finding the design solutions that consider energy efficiency objectives, while satisfying both indoor comfort level and cost optimality, is challenging. There are numbers of parameters and strategies involved, and it is necessary to use optimization strategies to find the best solutions. Conventional rules of thumb or trial-and-error processes are extremely unlikely to achieve near-optimal designs. To significantly increase energy efficiency and indoor comfort level, optimization techniques can be used. Multi-objective optimization is more relevant to single objective approach in many cases in solving real world problems.

Building performance optimization, within which optimization is coupled to BPS tools, is a process that aims at the selection of the optimal solutions from a set of
available alternatives for a given design or control problem, according to a set of performance criteria. Such criteria are expressed as mathematical functions, called objective functions. Simulation-based optimization has become an efficient design approach to satisfy several requirements in high performance buildings (Nguyen 2014).

In conventional optimization study, the process is usually automated by coupling a building simulation program and an optimization engine, which may consist of one or more optimization algorithms or strategies (Attia 2013). Genetic Algorithms (GA) are well suited to solve multi-objective optimization problems. GA-based multi-objective optimization methods that are frequently used in building research are Multi-Objective Genetic Algorithm (MOGA) and Niched Pareto Genetic Algorithm (NPGA) (Nguyen 2014). All these methods aim to produce a subset of the Pareto optimal set from which decision-makers can achieve the most appropriate solution for the problem at hand. The GA is particularly suitable for solving multi-objective optimization problems, because it deals with a set of solutions simultaneously, enabling a series of Pareto-optimal solutions in a single run, which is superior to classical methods. It can provide a continuous and iterative improvement of the building model, with the final aim of identifying a set of solutions that optimize both energy performance and thermal comfort.

The concept of multi objective optimization is an appropriate solution, which is a set of trade-off optimal solution. This approach is referred to “multi-objective optimization” or “Pareto optimization”. Pareto optimization provides only some elements of the Pareto set rather than the entire one. Due to complexity, researchers often use up to two objective functions, but some exceptions with three or more
functions have been observed. Three objective functions can be indoor environment quality, the carbon payback period and cash payback period time (Jin Q 2012). As a result, the aim of multi-objective optimization is to locate the Pareto-optimal set of solutions. The final goal of the multi-objective programming problem is the achievement of the Pareto front, which represents the set of non-dominated (i.e., “the best”) solutions. There are many termination criteria that are mostly dependent on optimization algorithms. Some of the most frequently used criteria in BPS are maximum optimization time, maximum number of generations, maximum number of equal cost function evaluations and acceptable objective functions (Nguyen 2014). An optimization may have more than one termination criteria, and when at least one of the criteria is met, the process ends. Some optimization studies divide this phase into two steps: an initial optimization and a detailed optimization to investigate various design solutions (Hamdy et al. 2011) and various model responses (Nguyen 2014). The multi-objective programming problem is solved by means of proper setting and running of the optimization program. This is a very delicate phase because it affects both reliability and accuracy of results.

1.4.1 Classification of building optimization properties

To select the correct optimization algorithm for a specific problem, we need to investigate properties of the problem. The properties of building optimization problems are various and can be generally classified into three basic schemes, namely, objective functions, design variables, and constraints. Understanding these classifications helps us to find a proper approach and algorithm for different aspects in building energy related problems.
<table>
<thead>
<tr>
<th>Design Variables</th>
<th>Objective Functions (Energy/Cost/Comfort)</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-dimensional</td>
<td>Single-objective</td>
<td>Inequality</td>
</tr>
<tr>
<td>Multi-dimensional</td>
<td>Multi-objective</td>
<td>Equality</td>
</tr>
<tr>
<td>Static</td>
<td>Quantitative</td>
<td>Linear</td>
</tr>
<tr>
<td>Dynamic</td>
<td>Qualitative</td>
<td>Non-linear</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Computationally-expensive</td>
<td>Separable</td>
</tr>
<tr>
<td>Probabilistic</td>
<td>Computationally-inexpensive</td>
<td>Inseparable</td>
</tr>
<tr>
<td>Discrete</td>
<td>Linear</td>
<td>Convex</td>
</tr>
<tr>
<td>Continuous</td>
<td>Non-linear</td>
<td>Non-convex</td>
</tr>
<tr>
<td>Independent-variables</td>
<td>Uni-model</td>
<td>Differentiable</td>
</tr>
<tr>
<td>Dependent-variables</td>
<td>Multi-model</td>
<td>Non-differentiable</td>
</tr>
<tr>
<td></td>
<td>Continuous</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Discontinuous</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Convex</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Non-convex</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Differentiable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Non-differentiable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Separable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inseparable</td>
<td></td>
</tr>
</tbody>
</table>

There are several methods that can be used to improve building performance, and to achieve an optimal solution to a problem. For example, computer building models can be created by a repetitive method, constructing infinitive sequences of progressively better approximations to a solution. These methods are known as “numerical optimization” or “simulation-based optimization” (Gosavi 2015). For example, one study focused on optimizing building engineering systems, where the direct search method in optimizing HVAC systems was used (Haung 2016).
Application of multi-objective optimization algorithms (MOOAs) identify the Pareto optimum trade-off between conflicting design objectives. To find optimal solutions without spending much time for simulation, MOOAs can be applied. There are seven evolutionary optimization algorithms in solving multi-objective optimization problems by using simulation-based optimization approach.

1.4.2 Overview on optimization algorithms in building energy simulation and analysis

There are large amount of design variables for building energy models that are discrete, non-linear and highly constrained, so the simulation results can be discontinuous and generate noise in building optimization problems (BOPs). The performance of optimization algorithms in solving BOPs are essential and need to be investigated.

Different optimization algorithms have been applied on single objectives BOPs in recent studies. These include coordinated search algorithm, Hooke-Jeeves algorithm, particle swarm optimization (PSO), PSO that searches on a mesh, hybrid PSO-Hooke-Jeeves algorithm, simple GA, simplex algorithm of Nelder and Mead, and discrete Armijo gradient algorithm, used in solving simple and complex building models and performance criteria. Many studies compare the performance of these algorithms on different design variables, which is be discussed in more detail in literature review chapter. (Si et al. 2016, Hamdy et al 2016, Junghans and Dande 2015, Kheiri 2018), Most of them found that GA consistently provide results closer to best solutions.
1.4.2.1 Hooke-Jeeves algorithm

This algorithm is a Generalized Pattern Search algorithm. It is initiated from a given starting solution and subsequently performs exploration and pattern search alternatively in the solution space. The exploration searches along the direction of each coordinate from a base solution using a predefined step size for a better performing solution. The pattern search strategy accelerates the searches along the improved objective value direction. The algorithm terminates when the maximum number of step reduction or the predefined convergence precision is achieved.

1.4.2.2 Particle swarm optimization (PSO)

The PSO algorithm is a member of the population-based stochastic algorithms. It is inspired by the social behavior of a shoal of fish or a flock of birds. Each individual, such as a “fish” in the shoal, is called a “particle”. For each particle, the PSO algorithm tracks the best position of the particle (which models cognitive behavior), as well as the best position of the population (which models social behavior) in terms of the objective function.

1.4.2.3 Hybrid PSO-Hooke-Jeeves algorithm

The hybrid algorithm first runs the PSO to achieve a near-optimal solution. This step is performed based on a pre-defined number of generations. When the PSO finishes, its optimal solution is utilized as the initial solution for the Hooke-Jeeves. This hybrid algorithm aims to locate the optimal area of the solution space by running the
PSO first and subsequently to carry out a refined search in that area by running the Hooke-Jeeves (Futrell 2015).

1.4.2.4 Simple GA

The genetic algorithm is a method for solving both constrained and unconstrained optimization problems, which is based on natural selection and the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution. It can be applied to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non differentiable, stochastic, or highly nonlinear. The genetic algorithm can address problems of mixed integer programming, where some components are restricted to be integer-valued.

1.4.2.5 Simplex algorithm of Nelder and Mead

This method is a commonly applied numerical method used to find the minimum or maximum of an objective function in a multidimensional space. It is a direct search method (based on function comparison) and is often applied to nonlinear optimization problems for which derivatives may not be known. However, the Nelder–Mead technique is a heuristic search method that can converge to non-stationary points on problems that can be solved by alternative methods.
1.4.2.6 Discrete Armijo gradient algorithm

This algorithm is affiliated with the family of gradient-based algorithms and can be applied to minimize continuously differentiable functions. This algorithm approximates the gradients using finite differences, and the optimization progress is driven by the difference increment reduced. The Armijo step-size rule is used to perform line searches.

The multi-objective optimization algorithms require more research-based evidence and studies to be understood in building domain. But, some of the popular MOOAs are: NSGA-II (non-dominated sorting genetic algorithm), MOPSO (multi-objective particle swarm optimization), PR-GA (two phase optimization GA), evMOGA (multi-objective evolutionary algorithm based on the concept of epsilon dominance), ENSEN (elitist non-dominated sorting evolution strategy), spMODE-II (multi-objective differential evolution algorithm) and MODA (multi-objective dragonfly algorithm). These algorithms have been used from 2002 to 2016, and implemented into MATLAB optimization toolbox. (Hamdy et al. 2016)

The performance of optimization algorithms can be evaluated based on these following aspects and corresponding indicators (Hamdy 2016):

- Execution time of the algorithm
- Convergence to the benchmarking optimal set, indicated by normalized generational distance (GDn)
- Normalized inversed generational distance (IGDn) used to quantify both the convergence and the diversity
• Diversity of solutions in the Pareto-optimal set, indicated by the normalized diversity metric, denoted by DMn

• Number of solutions on the Pareto-optimal set, denoted by NoPsolution

• Contribution of the algorithm’s solutions to the best Pareto front.

1.4.3 Building optimization methods

The selection of optimization method should be based on the nature of problem and the features of the optimization method. Each optimization problem should be formulated and defined with the goals and constraints. The next step is selection of solutions based on the goals and problem decision maker (DM) preferences. Figure 5 shows a general flowchart in building optimization process. There are many types of optimization methods, such as single and multi-objective methods, or based on the type of variables. The section below briefly discusses application of optimization methods in facade design. Chapter 2 provides an in-depth discussion and literature review.
1.4.3.1 Calculus-based optimization

This method of optimization is basic form of nonlinear programming and includes the first and second derivative test, and their higher-dimensional generalizations to find maximum and minimum values of a function subject to constraint of them. In building performance domain, this method is applied to find optimum building envelope configurations, optimal shape and number of floors to minimize conduction heat losses.
1.4.3.2 Direct search

This optimization method does not explicitly use derivatives. Therefore, the
direct search can optimize a function without recourse to its derivatives when explicit
information about the function is unavailable or untrustworthy. It is defined as a
method that sequentially compares trial solutions with the “best” solution at the time
while having a strategy for defining the next trial solutions (Kheiri 2018).

1.4.3.3 Simulated annealing

This is a meta-heuristic optimization technique that can be used in searches for the
global optimum. It is often used when search space is discrete and for problems where
finding an approximate optimum is more important than finding a precise local
optimum in a fixed amount of time. In building performance assessments, this method
(SA) compared it against GA and the GA performed slightly better than SA. But SA
applied in optimizing timber building for both LCA and structural configuration or
finding optimum LCC of two facade design with considering some facade related
parameters. (Varma and Bhattacharjee 2015).

1.4.3.4 Dynamic programming

This method is based on dividing a complex problem into linked sub-problems,
which deal with a few variables. Dynamic programming is efficient in time when
optimization problems deals with large number of variables. In this method we need to
ensure that the problem can be divided into sub-systems while maintaining the integrity,
conditions of monotonicity and separability should be met with two conditions. First, it
should be possible to divide the objective function. Second, as the output of each step will be the input of next step. This method has been used for thermal analysis in buildings, minimizing the thermal discomfort and heating and cooling loads in residential and office space. It has also been used for minimizing annual source energy utilization level, as well as heating and cooling peak loads for office building in different climate zones with respect to the envelope parameters. (Pomberio et al. 2017)

1.4.3.5 Genetic Algorithm (GA)

Genetic Algorithm was inspired by the idea of natural evolution and survival of the fittest principle of Darwin. In general, GA is a population-based algorithm, classified as a global-optimum finding algorithm that includes operations to search for the optimum solution globally as well as the operations that try to improve the solutions locally. Figure 6 shows the general process in application of GA algorithms. The iterative process of GA will converge to better solutions based on the breeding of the parents with higher performance. The multi-objective GA method is used to optimize building energy cost and zone thermal comfort. There are many studies using this method for energy-efficient design by optimizing the location or size of windows, building form, size of thermal zone and tilting the roof, facade configurations to reducing lighting, heating and cooling loads.
1.4.3.6 Particle swarm optimization (PSO)

PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution regarding a given measure of quality. The roots of the PSO is in the artificial life in general, and in bird flocking, fish schooling, and swarming theory. PSO, which has many similarities with GA, is one of the simplest methods that can be applied to discrete problems. This method is applied in research to optimize building envelope and passive strategy I residential buildings or minimize the total energy demand for heating, cooling and lighting of a classroom.
1.4.3.7 Harmony search (HS)

HS is a random meta-heuristic algorithm used to search for the optimum solution and inspired by harmony in music by mimicking the use of improvisation in music to seek better harmony (Geem 2001). Harmony Memory (HM) stores the solutions. The probability of selecting a component from the HM vectors is determined as Harmony Memory Considering Rate (HMCR). Pitch Adjusting Rate (PAR) is a mutation parameter that allows some small modifications to the selected vector from HM. The degree of modification can be defined as a small discrete or continuous value for discrete and continuous parameters, respectively. This method has been used to optimize building envelope and find alternatives to minimize energy consumption, cost and environmental impacts (Fesanghary et al. 2012). A different study optimized energy consumption, lighting and thermal comfort by using shading device in residential building. (Khoroshiltseva et al. 2016)

1.4.3.8 Ant colony optimization (ACO)

ACO is a probabilistic technique used to solve computational problems, which can be deduced as finding the best routes through graphs. Inspired by the foraging behavior of real ants, ACO is characterized as a distributed, stochastic search method based on the indirect communication of a colony of artificial ants (Dorigo 2006).

In this method, artificial ants will choose possible paths in a graph to reach the destination point. Pheromone trails, evaporation of pheromone, and other defined parameters direct route findings. Attractiveness, distance, tabu paths etc. will yield the probability of choosing the best routes by artificial ants. This method has been applied
to find the panel configurations that optimize lighting performance and cost (Evins 2013).

After discussing different algorithms and methods for optimization of building performance and envelope design, it should be mentioned that all methods can be classified into local and global optimum finding methods. A local optimum is defined as the relative best solutions within a neighbor solution set. A global optimum is a point where the function value is optimum, considering the value of all other feasible points. There is only one global max/min, but there can be more than one local max/min. The main difference is the way that algorithms search for the better solution. However, some of the methods, with some modifications, can be used for both global and local optimums.

In general, different optimization methods are implemented differently to find the optimum solutions. The efficiency of the optimization methods in each problem may be different based on problem characteristics.

1.5 Purpose and Outline of the Dissertation

This research first focuses on identifying the role of BPS and design optimization methods and outlines potential challenges and obstacles in performance-based facade design. This section is primarily based on literature review. Then, a new framework for performance-based facade design is presented. This framework considers occupant comfort, energy consumption and cost optimality, and implements BPS and relevant optimization methods to achieve a proper process for performance-based facade design. The components and development of the framework are discussed in detail, followed by testing and validation. The last part of the dissertation offers
conclusions.

The main contribution of this research is development of a new framework to support decision making in facade design process, considering real world applications and conflicting objectives. The research focuses on facade design and parameters related to this building system; however, the framework can be used a back-end for developing tools and interfaces that couple building simulation and optimization.

Chapter 1 introduces the main subject of this research, and discusses the role of facade, building simulation and optimization methods in building design.

Chapter 2 provides a literature review focusing on simulation-based optimization methods used for building envelope design. This chapter also provides an overview of applications, tools and methods for understanding occupants’ comfort.

Chapter 3 presents the research objectives, problem statement and describes the research questions and approach. This chapter also discusses contributions of this research.

Chapter 4 describes the methodology for this research and procedures for developing the framework.

Chapter 5 discusses application of the framework, results of the study and validation. This chapter discusses research findings in detail.

Chapter 6 presents the conclusion and proposes directions for future studies.
CHAPTER 2
LITERATURE REVIEW

There are many existing studies that provide literature reviews about whole building performance simulations and optimization methods. In this research, building facade was selected because of its influence on energy consumption, thermal and visual comfort of occupants. The literature review focuses on the role of BPS, optimization and tools, applications and methods in facade design, and then reviews studies on thermal and daylight modeling for facades and occupants’ comfort. High performance buildings require an efficient performance-based design process that integrates optimization methods into building performance simulations. Coupling simulation tools and optimization algorithms are aimed at removing the existing barriers between optimization and building simulations. The literature review has been conducted to identify studies on simulation and optimization for building energy performance, with special focus on facade systems, thermal and visual comfort.

2.1 Simulation and Optimization for Performance-Based Design: Overview of Tools, Applications and Methods

Providing an overview of BPS tools and the methods to quantify the objectives (performance criteria) in design process is important, since designers need to choose appropriate and efficient methods among several number of available approaches. The core tools in the building energy field are the whole-building energy simulation programs, which provide users with key building performance indicators, such as energy.
BPS tools have essential role in the process of building design to improve energy performance, environmental impacts, costs, etc. These simulation tools are available for simulating the performance of buildings and building subsystems, and are capable of whole building simulation, model input calibration and building auditing. Among whole-building simulation software tools, it is important to make a distinction between simulation engines and interfaces, modeler applications and Graphical User Interfaces (GUI). So, BPS tools can be classified into three main categories below:

- Applications with an integrated simulation engine (e.g. EnergyPlus, ESP-r, IES-VE, IDA ICE)
- Software that docks to a certain engine (e.g. DesignBuilder, eQuest, Sefaira, OpenStudio)
- Plugins for other software applications, enabling certain performance analysis (e.g. DIVA for Rhino, Ladybug, Honeybee, Autodesk’s Green Building Studio).

Most modeler applications support the user with a GUI to make data input easier. The modeler creates an input file for the simulation engine to solve. The engine returns output data to the modeler application or another visualization tool, which in turn presents the results to the user. For some software packages, the calculation engine and the interface may be the same product. Table 2 provides an overview of BPS software programs and commonly used simulation engines, plugins and glues modeler applications for BPS (Ostergard 2016). Glues refer to software programs that enable links between BPS and geometrical modeling through graphic programming. Important properties of the reviewed software are:

A. Users: Is the software primarily intended for architects, engineers, or both?
B. Design stage: In which design stages is the software typically used?

C. Interoperability: How does the BPS software connect to CAD environment and other software packages?

D. Level of complexity of the core algorithms: The complexity set the constraints of design options that the software enables to investigate and to what level of detail.

E. Objectives: Important, interdependent objectives must be evaluated to ensure holistic design.

F. Parametric: Ability to run global parametric calculations and to perform Uncertainty Analysis (UA) and Sensitivity Analysis (SA) – either by using integrated features or by configuring input text files and accessing output text files.
Table 2: Comparison of software in terms of fulfilling the requirements of the proposed software framework (Adopted from Ostergard 2016).

<table>
<thead>
<tr>
<th>Glue</th>
<th>Plugin</th>
<th>BPS (external engine)</th>
<th>BPS (own engine)</th>
<th>Software</th>
<th>Users</th>
<th>Stage</th>
<th>Interoperability</th>
<th>Core Complexity</th>
<th>Objectives</th>
<th>Parametric sim.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamo</td>
<td>AE</td>
<td>E</td>
<td>E</td>
<td>A</td>
<td>E</td>
<td>E</td>
<td>Conceptual</td>
<td>Preliminary</td>
<td>AE</td>
<td>AE</td>
</tr>
<tr>
<td>Grasshopper(GH)</td>
<td>AE</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>Detailed</td>
<td>Management</td>
<td>AE</td>
<td>AE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>Standalone</td>
<td>Standalone</td>
<td>Standalone</td>
<td>Standalone</td>
</tr>
<tr>
<td></td>
<td>Energy</td>
<td>Thermal Daylight LCA LCC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Glue</td>
<td>Dynamo, Grasshopper(GH)</td>
<td>BPS (external engine)</td>
<td>BPS (own engine)</td>
<td>Software</td>
<td>Users</td>
<td>Stage</td>
<td>Interoperability</td>
<td>Core Complexity</td>
<td>Objectives</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>AE</td>
<td>AE</td>
<td>AE</td>
<td>AE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>AE</td>
<td>AE</td>
<td>AE</td>
<td>AE</td>
</tr>
<tr>
<td></td>
<td>Energy</td>
<td>Thermal Daylight LCA LCC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Previous related research studies focusing on building envelope and simulation-based optimization are discussed below. Application of computational optimization methods in sustainable building design, with focus on residential retrofits and building envelope, was reviewed in detail by Envis (2013). Key aspects included construction and form, configuration and control of building systems, renewable energy generation and holistic optimizations of several areas (Envis 2013). Simulation-based optimization methods applied to building with focus on handling discontinuous multi-model building optimization problem was discussed by Nguyen (2014). Building design optimization techniques for real-world design challenge and the advances and obstacles in this field are described (Nguyen 2014). Optimization algorithms in building design was studied by Machairas (2014). This study reviewed the methods and tools used for building design optimization to explore the reasoning behind specific selections to present their abilities and performance issues and to identify key characteristics of their future versions (Machairas 2014). Building energy efficient design optimization from the architectural perspective was reviewed by Shi (2016). Analyzed subjects included the general procedure, the origin and development, the classification, design objectives and variables, the energy simulation engines, the optimization algorithm and the applications (Shi 2016). The optimization of building envelope design for the last 30 years was reviewed by Huang and Niu (2016). Numerous studies on optimization of building envelope design were assembled and reviewed, and targeted objectives were collected and summarized (Huang and Niu 2016). Building simulation and computational techniques were discussed by Wang and Zhai (2016). This study focused specially on six different topics, including ventilation performance predictions, building
energy and thermal load simulation, lighting and daylight modeling, building
information modeling, indoor acoustic simulation, and life cycle analysis of buildings. Major advances in each were highlighted, as well as trends for development and application (Wang and Zhai 2016). Ostergard reviewed the building simulation in early building design by covering development in both academia and commercial software industry that target the challenges, such as time-consuming modeling, rapid change of the design, conflicting requirements, input uncertainties and large design variability (Ostergard 2016). Building energy modeling for control and operation was reviewed by Li and Wen (2016). An effort was aimed to develop a zero-energy building design support tool (ZEBO) that facilitates the use of building performance simulation in the early design stage in hot climate (Attia 2013). This study reviewed challenges and opportunities for integrating optimization tools in net zero energy building design. Key factors and major findings of reviewed research in optimization of simulated performance-based building envelope design are summarized and listed at the end of this chapter in the final table.

Detailed frequency distribution of the utilization of different optimization methods for building envelope design is shown in Figure 7 along with the year that each specific optimization method was proposed (Kheiri 2018). The diagram shows the rapid increase in building envelope optimization studies in recent years. Three mostly used optimization methods are Genetic Algorithm (GA), followed by particle swarm optimization (PSO) and direct search.
Figure 7: Histogram of utilized optimization methods (Adopted from Kheiri 2018).

Among building performance simulation tools, EnergyPlus is used most frequently. EnergyPlus is followed by the calculus-based assessments, TRANSYS, Daysim, Radiance, DOE2, IDA ICF and other tools that are listed in Figure 8. The simulation tools for daylighting and thermal analysis are not divided in this list.
MATLAB is the most widely used platform for the optimization, followed by mathematical optimization, GenOpt, JEPlus, BeOpt, modeFRONTIER, ENEROPT, etc. The platforms include all the tools specifically developed for building design as well as optimization platforms for different disciplines. Figure 9 shows the optimization tools for building performance optimization studies.

Figure 8: Frequency of the use of building performance simulation tools for building performance research (Adopted from Kheiri 2018).

Figure 9: Frequency of the use of optimization tools for building performance research (Adopted from Kheiri 2018).
Optimization in building design has been used in different domains including space layout design, construction, structural design, acoustics, HVAC systems, and controls design. Furthermore, there are studies focusing on optimization of building components, such as wood-cement blocks, insulation, WWR, etc. to achieve lower energy consumption. There have also been different research studies conducted in the area of optimization for thermal and visual comfort in buildings (Kheiri 2018). In general, the optimization of an architectural design or retrofit can be done with different approaches. The difference can be in sampling the search space, methods to obtain optimum objectives, and the search algorithms.

According to existing surveys and interviews with professionals, users and participants for building performance optimization, findings reveal that Matlab toolbox and GenOpt are effective optimization tools, and the most used simulation engine is EnergyPlus (Attia 2013). Figures 10 to 12 show the results of that study about participants’ choice of variables, objective functions and constraints in building performance optimization subject.

Figure 10: Participants’ choice of optimization variables (Adapted from Attia 2013).
Computer building models that rely on repetitive method, construct infinitive sequences, of progressively better approximations to improve building performance and achieve the optimal solution to a problem can be used in energy efficient design. These methods are known as “numerical optimization” or simulation-based optimization. Wright pioneered the methods to optimize building engineering systems when he applied the direct search method in optimizing HVAC system (Gosavi 2015).

The term optimization refers to the procedures of making something (as a design, system or decision) as fully perfect, functional or effective as possible. In BPS, the term optimization generally indicates an automated process, which is entirely based
on numerical simulation and mathematical optimization (Attia 2012). In conventional optimization study, this process is usually automated by the coupling between a building simulation program and an optimization engine, which may consist of one or more optimization algorithms or strategies (Attia 2013). Building optimization refers to the methods that put forward optimal combination of simulation parameter setting in architecture design to find the optimum for the lowest total energy cost and simulation time.

One of the earliest studies used multi-objective optimization in building design and performed a Pareto optimization using dynamic programming (Gero 1983). Gero used multi-objective optimization in building design, using dynamic programming. Objectives included thermal load, daylighting, usable area and cost, and the variables covered massing, orientation and construction. Coley used a genetic algorithm to minimize energy use, and the method varied thermal conductance and thermal capacity for each zone in a model (2002). The novelty of this work was the combination of GA with human judgments. Presentation of both optimal and near optimal designs was achieved in a visual manner, enabling the user to choose based on the preference that need not be formalized as constraints or objectives. Mahdavi brought “virtual enclosure” concept that describes the building skin based on thermal and visual properties (2003). In this approach, multiple actual realizations map to a single virtual enclosure, allowing optimization algorithm to solve only the core underlying problem without conflicting information relating to its realization (Mahdavi 2003).

Wright optimized the placement of glazing using a genetic algorithm to minimize energy use based on hourly annual simulation conducted with EnergyPlus
This study described an approach in which a building facade is divided into a number of cells, each cell having one of two possible states, a solid wall construction, or a window. This method separated facade elements that could be glazed or solid, but the size of the design space increased due to very small windows. The solution was not practical because it required a constraint to limit the number of windows. Evins optimized the cost and energy use of a modular building for different climate types (2012). The variables included: construction (U-values and shading), HVAC and renewable energy sources (PV and solar thermal). Shading was optimized using a local search, which was embedded in the genetic algorithm used for all other variables. Then in 2015, Envis implemented multi-level optimization of building design, energy system sizing and operation. The optimization techniques used were a multi-objective genetic algorithm (design) and mixed-integer linear programming (operation). The objective functions used here were annual carbon emissions and initial capital cost for the multi-objective design problem and annual running costs for the single objective operational problem (Envis 2015).

2.2 Building Performance Simulation Studies in Design Process

Because the topic of this research is related to whole building simulation and daylighting simulation, the literature reviews are focused on these areas and recent related studies in these areas are discussed below.

2.2.1 Whole Building Simulation Models’ Literature

Thermal building simulation is arguably the most mature of the subfields of building performance simulation. Key information on this area is described in Chapter
There are many works and researches for energy simulation in building design and modeling methods for energy in buildings, two significant works were done by Clarke (2001) and Underwood (2014). In building performance simulation for design and operation, Hansen and Lambert studied thermal load and energy performance prediction and thermal analysis on climate data, occupant behavior, HVAC systems and building automation (2011). Crawley in an influential paper compared the features and capabilities of twenty building simulation tools for whole building simulation (2008). Oh and Haberl studied the history of building energy simulation tools in the United States, and included charts to trace some of these developments (2016).

Strachan discussed the history of validation of the ESP-r simulation program (2008). Dols studied work on TRANSYS simulation tool, focusing on improving the coupling of heat transfer, airflow and contaminated simulation (2015). Daum and Morel studied different blind control and lighting schedules and simulated the impact of these variables on energy use of an office space (2009). Zhou worked on general comparison of the HVAC system modeling in simulation tools and described some of the fundamental differences (2014). Peng linked EnergyPlus simulation, building control virtual test bed and actual energy management and control system to compare predicted and measured building performance in terms of power consumption in real time (2012). Rysanek developed an automated search for energy efficient building refurbishment option within Matlab environment (2012). Zhao and Magoul studied different ranges of building energy consumption prediction tools, including the traditional building simulation tools, statistical methods, neural networks, support vector machines and grey methods (2012). Foucquier studied building energy performance prediction and refined
physical models, statistical and hybrid methods (2013). This work blended the borders between thermal and CFD models.

2.2.2 Lighting and Daylight Simulation Literature

Based on literature reviews, Ochoa provided a deep discussion about daylighting and how it relates to light, form and people (2012). Reinhart gave a brief overview about this subject as well (2006). Mardaljevic provided an overview about daylight metrics (2009). Andersen studied interaction between daylighting and human needs, and investigated the impact of daylighting on health, visual comfort and perception of space (2015). Du and Shaples used Radiance to study the impact of different forms and geometries and reflectance distribution on vertical daylighting level (2010). Mahdavi studied the five options to generate sky models and how these options impact computed irradiance values for vertical surfaces (2010). De Keyser worked on modeling and simulation of lighting control systems to keep the illuminance level constant in the space (2010). Reinhart presented a review of dynamic daylighting simulation capabilities and suggested the use of daylighting in design of daylit spaces (2011). Kota discussed the integration between BIM and daylighting simulation tools (2014). This research linked Revit with Radiance and Daysim. Yu worked on energy assessment potential offered by use of daylighting in educational buildings (2014). Bueno developed a Radiance-based approach to evaluate complex fenestration system (2015). Amundadottir studied advanced performance aspects, such as visual interest and gaze behavior, using Radiance to predict ocular light exposure in educational buildings (2017). Konis investigated the relations between building design, daylighting and the circadian rhythm using simulations (2017).
2.3 Building Performance Optimization Studies in Design Process

As discussed before in previous section, there is a range of available BPS tools, including EnergyPlus, DOE-s, TRNSYS, IDA ICE, etc. Besides these simulation engines and tools, there are some optimization tools such as: GenOpt, JEPlus, MultiOpt, BeOpt, etc. These tools are used to couple the energy simulation programs with optimization algorithms. This section discusses some of these optimization methods, tools and objectives in building design process.

Bucking used BeOpt to measure the effect of economic incentive on net-zero energy building (2013). Delgarm studied building energy consumptions of a single room model by coupling JEPlus and EnergyPlus (2016). Particle swarm optimization was used in this study to investigate the effects of building architectural parameters on energy consumption, including building orientation, shading overhang specifications, window size, and the glazing and the wall material properties in four major climatic regions of Iran (Delgarm 2016). Chantrellea developed MultiOpt tool for optimizing the building renovation operations, including building envelope, heating and cooling loads and control strategies (2011). MultiOpt is based on existing assessment software and methods, using a genetic algorithm (NSGA-II) coupled to TRNSYS, and economic and environmental databases (Chantrellea 2011).

GenOpt is the most used optimization tool in research and literature. Futrell combined a light environment analysis software and GenOpt to optimize buildings daylighting design, and to minimize lighting loads (2015). The study used dynamic climate-based lighting simulations and compared four optimization algorithms as implemented in GenOpt: Simplex Algorithm of Nelder and Mead with the Extension of
O’Neill (SA), Hooke Jeeves (HJ), Particle Swarm Optimization using Inertia Weight (PSOIW), and a hybrid PSO Constriction/Hooke Jeeves (PSOC/HJ) algorithm (Futrell 2015). Karaguzel used EnergyPlus and GenOpt to minimize the life cycle cost of building materials and operational energy consumption for an office building (2014).

Machine learning techniques for the development of classification models are applied in many fields. Methods such as support vector machines (SVM) or artificial neural networks (ANN) have demonstrated their high reliability in the training of non-linear regression and classification models. The difference is mainly on how non-linear data is classified. The SVM utilizes nonlinear mapping to make the data linear separable, hence the kernel function is the key. However, ANN employs multi-layer connection and various activation functions to deal with nonlinear problems. Among data-driven methods (such as ANNs, support vector machine, etc.), ANN (Artificial Neural Network) is mostly used in research studies because of better performance and structure compared to other methods (Gossard 2016, Wei 2015, Asadi 2014). Asadi studied multi-objective optimization model that combined ANN with NSGA-II to quantitatively assess technology choices in a building retrofit project (2014). Objective functions were energy consumption, retrofit cost, and thermal discomfort hours. Wei used ANN models for building comfort and energy efficiency of residential building (2015). Based on the simulation data on energy consumption and indoor thermal comfort, the study also used a simulation based improved back-propagation (BP) network, which is optimized by a genetic algorithm (GA) to characterize building behavior, and then establishes a GA-BP network model for rapidly predicting the energy consumption and indoor thermal comfort status of residential buildings (Wei
Gossard used ANN and NSGA-II to find optimal value of a building envelope considering annual energy consumption and summer comfort degree (2016). In order to reduce the computation time without reducing the complexity of the problem, an artificial neural network was developed to provide fast and accurate evaluations of objective functions, which are used by a genetic algorithm. These studies also showed that using data-driven methods and models can speed up the optimization.

There are several studies on the performance of different optimization algorithms, for both single objective optimization and multi-objectives optimization. There are more studies on single-objective optimization algorithms and their performance than multi-objective. But here multi-objective optimization algorithms are discussed because they relate more to the dissertation topic and the real-world problems, and there is a need to improve our understanding of these methods. Below, the studies relating to the building domain and multi-objective optimization are discussed in more detail.

Hofpe (2009) compared performance of two multi-objective optimization algorithms. The NSGA-II and the SMS-EMOA were compared based on their performance and solutions on multi objective optimization. The NSGA-II was not satisfied in all tests, while the SMS EMOA gave more competitive results, but the number of simulations was much higher. Brownlee compared the performance of five multi-objective algorithms (IBEA, MOCell, NSGA-II, SPEA2 and PAES) to solve multi-objective problem regarding window placement (2011). The study found that the NSGA-II outperformed the others, both in terms of the size of hyper volume and the spread of optimal solutions.
Hamdy evaluated performance of three multi-objective algorithms, including the NSGA-II, aNSGA-II (NSGA-II with active archive) and pNSGA-II (NSGA-II with a passive archive strategy) to find the cost optimal and near-zero energy building solutions for a problem that has a large discrete solution-space (2012). The study reported that the aNSGA-II has a better repeatability in finding high-quality solutions close to the true Pareto front with fewer evaluations and high convergence than the original NSGA-II and pNSGA-II.

Bichiou developed a comprehensive energy simulation environment to optimally select both building envelope features, and heating and air conditioning system designs and operation settings for single family homes (2011). The study used three optimization algorithms considered in the simulation environment: GA, PSO and SS. Based on comparative conclusion the robustness and the effectiveness of these algorithms, SS technique has generally significantly higher computational efforts than both the PSO and the GA techniques. GA requires the least computational time compared to PSO and SS.

Junghans combined an evolutionary genetic algorithm with a modified simulated annealing algorithm (2015). The study shows that GA does not always provide solutions close to the global optimum and provided a building optimization method, which result a higher reliability than the GA alone. This study illustrated that the hybrid GA coupled with the modified SA provides solutions close to the global optimum in all of the test runs. The proposed algorithm provides more reliable results than the GA without the addition of the modified SA.
Hamdy compared performance of seven commonly used multi-objective evolutionary optimization algorithms in solving the design problem of a nearly zero energy building (2016). The compared algorithms include: pNSGA-II, MOPSO, PR-GA, ENSES, evMOGA, spMODE-II, and MODA. In most cases, the improvement of the quality of the obtained solutions has direct relation with the number of generations. The search by the PR-GA algorithm covered a large area of the solution space and included near optimal solutions with a good diversity. Similar characteristics were shown by pNSGA-II, evMOGA and spMODE-II after PR-GA. In most running cases, ENSES, MOPSO and MODA achieved uncompetitive results.

Si evaluated algorithms used for building energy efficient design optimization (2016). A set of performance indices, namely, stability, robustness, validity, speed, coverage and locality, were proposed to evaluate the overall performance of algorithms. Based on the comparative conclusion, the convergence of MOGA-II is faster than MOPSO. Both MOPSO and MOGA-II widely searched throughout the space globally, although MOPSO performed slightly better than MOGA-II, while HJ trapped in local optimum. Both MOGA-II and MOPSO showed robust performance when slightly changing the optimization parameters, although MOGA-II performed slightly better than MOPSO, while HJ results are highly sensitive to its control parameter settings.

Multi-objective evolutionary algorithms (MOEA) are used for several purposes, such as cost-effective design, environmentally friendly buildings, energy efficient and high comfort building, net-zero energy buildings, etc. These algorithms also have great potential for these research subjects and combination with other algorithms to find optimal solutions for multi-objective problems.
2.4 Occupant Comfort (Thermal Comfort and Visual Comfort)

The American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) defines human thermal comfort as “the condition of the mind in which satisfaction is expressed with the thermal environment” (ANSI/ASHRAE Standard 55). There are two approaches to understanding and measuring thermal comfort: rational and adaptive. The rational or heat balance approach uses data from climate chamber to support its theory, while the adaptive approach uses data from field studies of people in buildings. Both have certain benefits and limits, as described below.

For thermal comfort criteria, the Predicted Mean Vote (PMV) is widely used and accepted for design and assessment of comfort conditions. PMV, suggested by Fanger (1970), relies on six factors for thermal comfort: air temperature, mean radiant temperature, air movement, relative humidity, clothing levels and activity levels. With these factors, the comfort zone is on average close to condition of 79°F (26°C) and 50% relative humidity. The Predicted Percentage of Dissatisfied (PPD) predicts the percentage of people who would feel more than slightly warm or slightly cold. Three kinds of comfort zone can be defined for PPD based on PMV. PMV is an index corresponding to the occupant's assessment of thermal conditions in an environment, where indices of -3 (cold) and +3 (hot) are the extremes of the seven-point scale. The standard recommends that PMV values for a particular space be within a -0.5 to +0.5 range values, which correspond to a 10% PPD, as seen in Tables 3 and 4.
Table 3: ASHRAE standard recommendations (Djongyang 2010).

<table>
<thead>
<tr>
<th>Operative temperature</th>
<th>Acceptable range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summer</td>
<td>72°F (22°C)</td>
</tr>
<tr>
<td>Winter</td>
<td>76°F (24.5°C)</td>
</tr>
<tr>
<td></td>
<td>68-73.5 °F (20-23°C)</td>
</tr>
<tr>
<td></td>
<td>73.5-79 °F (23-26°C)</td>
</tr>
</tbody>
</table>

Table 4: Predicted percentage of dissatisfied (PPD) based on the Predicted mean vote (PMV) (Djongyang 2010).

<table>
<thead>
<tr>
<th>Comfort</th>
<th>PPD</th>
<th>Range of PMV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;6</td>
<td>-0.2 &lt; PMV &lt; 0.2</td>
</tr>
<tr>
<td>2</td>
<td>&lt;10</td>
<td>-0.5 &lt; PMV &lt; 0.5</td>
</tr>
<tr>
<td>3</td>
<td>&lt;15</td>
<td>-0.7 &lt; PMV &lt; 0.7</td>
</tr>
</tbody>
</table>

An adaptive approach is derived from field studies, with the purpose to analyze the real acceptability of thermal environment, depending on the context, the behavior of occupants and their expectations. Therefore, the adaptive approach allows for analysis of other factors than those that can be simulated, such as everyday clothing and behaving without additional restrictions. In thermal experience, there is a very complex interaction between occupants and their environments. The adaptive comfort zone allows for the change in operating conditions depending on prevailing mean outdoor temperature, as seen in Figure 13 (ASHRAE 55-2010).
Figure 13: Adaptive Comfort Standard (ACS) for ASHRAE Std. 55, applicable for naturally ventilated buildings.

Illuminating Society of North America (IESNA) defines visual performance as the quantitative assessment of the performance of visual task, taking into consideration speed and accuracy (IESNA, 2001). It is a measure of how well one can perform a visual task. Visual comfort relates to the illuminance level. Visual performance depends on the following criteria: task illuminance, luminance ratio, uniformity, daylighting glare index and visual comfort probability. Selected visual comfort criteria for daylight quality and quantity for facade designs are task illuminance, uniformity, and luminance of window surface. Lighting Handbook, Chartered Institute of Building Service Engineers (CIBSE), Code of Interior Lighting, Commission Internationale de L’Eclairage (CIE) provides guidelines for task illuminance values, which can be used during the design process (IES lighting handbook, CIBSE, CIE Guideline). The
recommended value for general offices based on CIBSE is 46.5 foot-candles (500 lux). For performance of visual tasks of medium contrast or small size, 46.5-70-93 fc (500-750-1000 lux), based on IESNA. Spaces with the medium visual requirements, such as medium machining, offices, control rooms etc., require 28-70-93 fc (300-500-750 lux) based on CIE. Uniformity is the ratio between minimum and maximum illuminance or between the minimum and average illuminance at a work plane. Based on CIE and CIBSE standards, for the whole room, the ratio of the minimum and average illuminance should be greater or equal to 0.8.

Moeck proposed an average illuminance of window surface instead of luminance ratio, daylighting glare index and visual comfort probability, which is more appropriate (1998). There is no clear-cut standard for limiting luminance on window surface, but IESNA RP-1 1993 limits the luminance of any room surface to 850 cd/m$^2$. With a perfectly diffuse surface with the reflectance of 0.8, this is illuminance of 310 fc (3340 lux).

Another area of current research interest involves dynamic daylight performance metrics. While the widely-used daylight factor (DF) is calculated based on the ratio of the internal illuminance at a given point to the unshaded external horizontal illuminance under a CIE overcast sky, the dynamic daylight metrics are calculated based on annual climate data, and thus account for variations in both climate and solar position (Reinhart et al., 2006; Nabil and Mardaljevic, 2006). The calculation of these metrics is more complicated than the daylight factor method and requires the use of special software, such as DAYSIM, a Radiance-based annual daylight simulation software.
2.5 Summary of Literature

Key factors and major findings of reviewed research studies relating to optimization of simulated performance-based building envelope design are summarized in Table 5.

Table 5: Key factors and major findings of reviewed research.

<table>
<thead>
<tr>
<th>Author &amp; Year</th>
<th>Objectives</th>
<th>Optimized parameter</th>
<th>Tool &amp; Algorithms</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ucar and Balo in 2009 &amp; 2010</td>
<td>Payback Period</td>
<td>Thickness of insulation materials</td>
<td>Direct Search</td>
<td>A functional relationship between heat loss and the thickness of insulation materials. Reduced cost, the space heating and cooling energy requirement of a new house in Sydney by up to 94% compared to BASIX requirement.</td>
</tr>
<tr>
<td>Bambrook et al. in 2011</td>
<td>Life Cycle Cost</td>
<td>Thickness of insulation materials, ventilation rate, thermal mass, window type, area, orientation and shading</td>
<td>Direct Search</td>
<td></td>
</tr>
<tr>
<td>Albatici and Passerini in 2011</td>
<td>Heating requirement</td>
<td>South exposure coefficient</td>
<td>Direct Search</td>
<td>Both the shape coefficient and the south exposure coefficient were almost linear in relation to the heating energy consumption. A function for connecting the building internal envelope’s specific heat and the indoor space</td>
</tr>
<tr>
<td>Jiang et al. in 2012, Cheng et al. in 2014</td>
<td>Room temperature</td>
<td>Building internal envelope’s specification heat</td>
<td>Direct Search</td>
<td></td>
</tr>
<tr>
<td>Author(s)</td>
<td>Research Topic</td>
<td>Key Parameters</td>
<td>Methodology</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Asadi et al. in 2014</td>
<td>Energy consumption</td>
<td>17 popular building design parameters</td>
<td>Direct Search</td>
<td></td>
</tr>
<tr>
<td>Futrell et al. in 2014</td>
<td>Energy consumption and visual comfort</td>
<td>Window’s size, location and optical properties</td>
<td>Hybrid GPS Hooke-Jeeves/PSO algorithm</td>
<td></td>
</tr>
<tr>
<td>Stazi et al. in 2012</td>
<td>Life cycle cost of the solar wall system</td>
<td>Material, thickness of the wall, type of the window</td>
<td>Factorial plan technique</td>
<td></td>
</tr>
<tr>
<td>Gong et al. in 2012</td>
<td>Minimum thermal load</td>
<td>Thickness of insulation materials, window type, area, orientation and shading</td>
<td>Orthogonal method</td>
<td></td>
</tr>
</tbody>
</table>

Temperature was built.

Regression analysis based on the gradient-deterministic search methodology was still an acceptable approach in building design for the optimization of energy conservation. For the south orientation, thermal and the visual performance did not significantly conflict, but for the north orientation the values had the greatest conflict. Method was fast, simple and intuitive. However, it could only be applied for simple systems. An orthogonal method did not require a deep knowledge of computer programming, and thus was suitable for architects.
<table>
<thead>
<tr>
<th>Authors</th>
<th>Year</th>
<th>Focus</th>
<th>Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ruiza et al.</td>
<td>2014</td>
<td>Energy consumption and life cycle cost</td>
<td>Tabu Search</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thickness of insulation materials, window type and area</td>
<td></td>
</tr>
<tr>
<td>Wright and Moursched</td>
<td>2009</td>
<td>Energy consumption</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Window to wall ratio</td>
<td></td>
</tr>
<tr>
<td>Tuhus-Dubrow and Krarti</td>
<td>2010</td>
<td>Energy consumption</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Envelope shape</td>
<td></td>
</tr>
<tr>
<td>Sahu et al.</td>
<td>2012</td>
<td>Energy consumption</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The shape, orientation and materials of the building envelope</td>
<td></td>
</tr>
</tbody>
</table>

The methodology is designed to estimate the annual energy demand, life-cycle cost and the energy rating. In this paper, 48 different scenarios were analyzed with SEDICA software. The result from the GA showed a stochastic behavior, the reliability was satisfactory. There were some differences in energy consumption among different building shapes, these deviations were within 0.5%. The climate influenced the accuracy of the optimization. If the space load situation was complex, with a widely variable combination of heating, cooling and humidity conditions, the degree of error would be larger.
<table>
<thead>
<tr>
<th>Authors and Year</th>
<th>Topic</th>
<th>Parameters Considered</th>
<th>Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ioannou and Itard 2015</td>
<td>Energy consumption</td>
<td>U-value for window, wall, roof and floor, ventilation, occupant, thermostat</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>Kampf et al. 2010</td>
<td>Energy absorbed by corresponding surface</td>
<td>Building shape with consistent building volume</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>Gagne and Andersen 2011</td>
<td>Illumination level and glare index</td>
<td>Size, location, transmittance of the window, window-to-wall ratio</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>Rakha and Nassar 2011</td>
<td>Illuminance level</td>
<td>Geometric forms of ceiling area</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>Yi and Kim 2015</td>
<td>Sunlight exposure time</td>
<td>Distribution of buildings</td>
<td>Genetic Algorithm</td>
</tr>
</tbody>
</table>

Under heating conditions, the window thermal properties were the most critical parameters among all variables. Urban scale optimization by using GA was possible. The micro-GA-based optimization required much more computing time and a relatively large population size. Proposed method was more precise. The relative errors for the Pareto optimal solution and simulated results in EnergyPlus are relatively small, illustrating that the energy consumption and indoor thermal comfort performance can be accurately predicted by the building design multi-objective optimization model, and the model can be used in actual building design.
<table>
<thead>
<tr>
<th>Authors</th>
<th>Focus</th>
<th>Parameters</th>
<th>Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hamdy et al. in 2011</td>
<td>Energy consumption and thermal comfort</td>
<td>Size and U-value of window, size of shading system</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>Yu et al. in 2015</td>
<td>Energy consumption and thermal comfort</td>
<td>Area, orientation, heat transfer coefficient of wall and window</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>Stavrakakis et al. in 2012</td>
<td>Thermal comfort</td>
<td>Position and size of window-opening</td>
<td>Genetic algorithm based on meta-model construction</td>
</tr>
<tr>
<td>Palonen et al. in 2009</td>
<td>Life cycle cost</td>
<td>Insulation thickness, U-value of window</td>
<td>Genetic Algorithm (Applying an omni-optimizer)</td>
</tr>
<tr>
<td>Kampf and Robinson in 2009</td>
<td>Energy consumption</td>
<td>Insulation thickness, U-value of window</td>
<td>Genetic Algorithm (Hybrid covariance matrix adaptation)</td>
</tr>
</tbody>
</table>

An optimal solution for both thermal comfort and energy saving would require 10 kWh/(m²a) more than an optimal solution for energy saving alone. Under the same level of energy consumption, the indoor thermal comfort status did not change significantly. Proposed methodology provides the optimal window designs, which correspond to the best objective variables for both single and several activity levels. The stopping criterion definition was very important for the reduction of computing time. The covariance matrix was used for decorrelation of design variables. Although the calculation of the...
<table>
<thead>
<tr>
<th>Reference</th>
<th>Energy Consumption</th>
<th>Design Variables</th>
<th>Optimization Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ramallo-Gonzalez and Coley in 2014</td>
<td>Energy consumption</td>
<td>Window to wall ratio, ventilation, size of shading systems</td>
<td>Genetic Algorithm (evolutionary strategy and HDE algorithm)</td>
</tr>
<tr>
<td>Eisenhower et al. in 2012, Geyer and Schluter in 2014</td>
<td>Energy consumption and thermal comfort</td>
<td>Insulation thickness, orientation, U-value of window, window-to-wall ratio</td>
<td>Applying multiple algorithms for building envelope optimization.</td>
</tr>
<tr>
<td>Bucking et al. in 2013</td>
<td>Energy consumption</td>
<td>Insulation thickness, orientation, U-value of window, window to wall ratio</td>
<td>Genetic Algorithm</td>
</tr>
</tbody>
</table>

covariance matrix was time-consuming (compared with the computing time used for annual building simulation), it was acceptable. The results show that this method can reach the same optimal design, with substantially lower computational time than the optimization methods found in the literature. The proposed approach could save around 80% of the computing time with a negligible change in the degree of error. Considering mutual information (dependency between variables) during the evolution of the design cases. With the consideration of mutual information, the data mining with genetic algorithm optimization could save 40%
<table>
<thead>
<tr>
<th>Author(s) and Year</th>
<th>Objective</th>
<th>Design Parameters</th>
<th>Optimization Algorithm</th>
<th>Accuracy Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Junghans and Darde in 2015</td>
<td>Life cycle cost</td>
<td>U-value of wall, U-value and transmittance of window, U-value of shading system</td>
<td>Genetic Algorithm (Combined with simulated annealing Algorithm)</td>
<td>At least 5%</td>
</tr>
<tr>
<td>Fesanghary et al. in 2012</td>
<td>Life cycle cost and CO₂ emission</td>
<td>Wall material and glazing type</td>
<td>Harmony Search Algorithm</td>
<td>Relatively long computing time, stable and trustworthy results</td>
</tr>
<tr>
<td>Magnier and Haghighat in 2010</td>
<td>Energy consumption and thermal comfort</td>
<td>Window to wall ratio, wall thickness</td>
<td>Genetic algorithm coupled with artificial neural networks</td>
<td>More quickly and reliably</td>
</tr>
<tr>
<td>Zemella et al. in 2011</td>
<td>Energy consumption and annual carbon emission</td>
<td>Popular building envelope design parameters</td>
<td>Evolutionary artificial neural networks</td>
<td>More accurate and less time-consuming</td>
</tr>
<tr>
<td>Gossard et al. in 2013</td>
<td>Energy consumption and thermal comfort</td>
<td>Thermophysical properties of wall and roof</td>
<td>Genetic algorithm coupled with artificial neural network</td>
<td>Comparison shows the importance of performing an aggregative method and a constraint problem in GenOpt.</td>
</tr>
<tr>
<td>Authors</td>
<td>Focus Areas</td>
<td>Parameters</td>
<td>Optimization Method</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------------</td>
<td>---------------------</td>
<td></td>
</tr>
<tr>
<td>Manzan and Pinto in 2009</td>
<td>Energy consumption and visual comfort</td>
<td>Positions and sizes of the external shading equipment</td>
<td>Genetic Algorithm</td>
<td></td>
</tr>
<tr>
<td>Hamdy et al. in 2010, Ferrara et al. in 2014</td>
<td>Energy consumption and life cycle cost.</td>
<td>Insulation thickness, type of glazing and shading systems.</td>
<td>Genetic Algorithm</td>
<td></td>
</tr>
<tr>
<td>Karmellos et al. in 2015</td>
<td>Energy consumption and life cycle cost.</td>
<td>Popular building envelope design parameters</td>
<td>Genetic Algorithm</td>
<td></td>
</tr>
<tr>
<td>Ferraraa et al. in 2015</td>
<td>Energy consumption, Thermal and Visual comfort</td>
<td>Popular building envelope design parameters</td>
<td>Particle swarm optimization (PSO) algorithm</td>
<td></td>
</tr>
</tbody>
</table>

The results showed that the impact of shading devices on buildings energy needs to consider the electrical energy absorbed by the lighting system. A series of reference data for residential house designs in Finland and Italy was proposed. A multi-objective optimization should consist of two parts: an optimization process and a decision-making process. A decision-making software tool was developed. Methodology for designers to identify the most suitable set of passive solutions to guarantee a comfortable indoor environment and hence to minimize the total energy needs.
<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Topic</th>
<th>Parameters</th>
<th>Method</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ferrara et al. in 2014 &amp; 2018</td>
<td>Energy consumption and life cycle cost</td>
<td>Popular building envelope design parameters</td>
<td>Particle swarm optimization (PSO) algorithm</td>
<td>The cost-optimal approach results as an effective method for delineating the future of NZEB design.</td>
</tr>
<tr>
<td>Azari et al. in 2016</td>
<td>Energy consumption and life cycle cost</td>
<td>Insulation material, window type, window frame material, wall thermal resistance, window to wall ratio</td>
<td>Genetic Algorithm coupled with hybrid artificial neural network</td>
<td>Find the optimum design scenario for low-rise office building in Seattle.</td>
</tr>
<tr>
<td>Delgarm et al. in 2016</td>
<td>Energy consumption</td>
<td>building orientation, shading overhang specifications, window size, and glazing and wall material properties</td>
<td>Particle swarm optimization (PSO) algorithm</td>
<td>Proposed optimization method shows a powerful and useful tool that can save time while searching for the optimal solutions with conflicting objective functions; therefore, facilitate decision making in early phases of a building design in order to enhance its energy efficiency.</td>
</tr>
</tbody>
</table>
CHAPTER 3

RESEARCH OBJECTIVES AND PROBLEM STATEMENT

3.1 Current Gaps in Knowledge (Research Problems)

Based on the literature review, it was concluded that some studies limited research objectives to building performance simulations relating to heating, cooling, airflow, lighting, and some investigations considered optimization of the building shape, HVAC systems, control systems, etc. A limited number of studies focused on developing a method for the design of building facades that take into account different performance criteria (energy, thermal and visual comfort). There is also a lack of the workable framework that implements both simulation analysis and optimization methods during facade design based on multiple performance criteria that are related to this building system.

Building simulation tools have evolved from equation solvers to validated software programs with a large user base. But there are weaknesses and gaps in the functionality of these tools. It is expected that building simulation tools will be developed to better integrate simulations during all phases of the building process. Past research efforts focused more on the data aspects of design analysis interaction rather than the design process. Enabling data exchange between applications and analysis applications enable designers and decision makers to properly deal with data exchange and control the design process. Discussions are no longer about software features but on the integration and use of simulation in the building process. So, a new approach for
design integration requires a conceptual framework that improves the analysis applications in the design process.

The new developments in building simulations should consider integration of different physical domains and provide dynamic control of design settings. But the problem of many facade design-related studies is that they are investigating the impacts of different variables separately. Therefore, this research tackled this gap in knowledge and addressed fenestration, glazing system, wall assembly and shading control systems, taking into consideration their combined impacts on energy consumption (heating, cooling, and lighting), thermal and visual comfort and then optimizing their operation by simulation and analysis of different design scenarios. The main problem in current integrated building simulation software programs is that they are usually used to evaluate the overall energy performance in the design development phase when the building form is already determined, as well as building systems. Defining different combination of facade elements, evaluating these scenarios and selection of final design solutions concerning subjective factors mentioned above are the goals of this study.

Moreover, energy modeling and simulations in design process are usually limited to analysis of few different scenarios. It is time consuming to simulate and analyze all possible design scenarios. In summary, a limited number of studies have focused on the performance-based design process for building facades which integrate simulations and optimization methods. There is a lack of a workable framework that implements both simulation analysis and optimization methods for facade design, taking into account performance criteria specific to this building system. Discussions are now longer about the software and tools’ features, but about the integration and
increased use of simulations in the design process. The future performance-based design approaches and simulation tools for facades should increase effectiveness, speed, quality, assurance and users’ productivity. Therefore, this research focused on developing a framework that couples simulation and optimization methods, and allows multiple design scenarios to be tested rapidly. The framework was implemented by coupling Python scripting with EnergyPlus simulation engine, enabling users to consider more variables during the design process. Moreover, a web-based application (PerforPy) was developed, which enables easier interaction between simulation program and optimization algorithms through a Graphical User Interface (GUI).

3.2 Research Objectives & Questions

The objectives of this research can be divided into two parts. The first part (Objective 1) focuses on the fundamental concepts of understanding building performance simulations, optimization methods, and tools for quantifying and making design decisions. The research categorized simulation and optimization problems, analyzed the state-of-the-art building simulation software and procedures for performance-based design. This objective is covered in Chapters 1 and 2, and the primary research methods included literature reviews and analysis of previous studies. The second part (Objective 2) focuses on developing the design framework for building facades, taking into account facade design variables to optimize energy consumption, occupant comfort and energy cost. Then, to test the framework and its effectiveness, a test cell was used to investigate functionality of the framework, its implementation and effects on design decision making.
3.2.1 Research Questions for Objective 1

The main research questions associated with Objective 1 are:

1. How do simulations support performance-based building design through different stages of the design process?

2. What performance criteria are used in the design process, how to quantify the required level of performance and choose adequate measurement methods?

3. What are the appropriate building performance simulation tools and optimization engines for building facade design?

4. How do thermal and daylight modeling contribute to decrease in energy use and enhance occupant comfort?

5. What are challenges for coupling building simulation programs and optimization methods in building performance analysis and decision making?

3.2.2 Research Questions for Objective 2

The main questions that are associated with Objective 2 are:

1. What are the main methods in the performance-based approach of facade design in this research?

2. What are the design variables for performance-based facade design, which can reduce energy consumption and meet occupants' thermal and visual comfort?
3. What kind of simulations should be performed to quantify each particular design variable? What data is needed and what is available?

4. How can different design scenarios be created and controlled based on specific variables and performance ranges?

5. How can simulations and optimization methods be integrated and conducted to meet the performance criteria requirements in the framework?

6. How will the optimization methods be implemented to select optimal scenarios? How will the coupling between optimization and simulations be performed?

3.3 Model Development and Overview of the Framework

This research focused on simulation-based optimization methods, with the primary goal to develop a framework for facade design that integrates simulation engine with optimization algorithms, in an automated manner. This framework can be used as a back-end for web or mobile applications (or any user interfaces) and eventually after collecting enough data, the deep learning can be implemented to predict the configurations related to outputs.

Figure 14 below represents the whole model development, and each part will be explained in more detail in next chapter, where the step-by-step methodology is discussed in detail.

The main focus of this work is to develop the Minimum Viable Product (MVP), which is the core feature to effectively deploy the product to the users. A MVP is that version of a new product which allows a team to collect the maximum amount of
validated learning about customers with the least effort (Ries, 2011). MVPs are
designed to test hypotheses and assumptions, but they are not the final product. They
rely on the concept of validated learning.

MVP can be part of the process directed toward making and marketing this
product to the users (Raddof, 2014). It works as an object in the iterative process of
generation, presentation, data collection and analysis and learning. Creating MVP will
allow collection of maximum amounts of validated learning data from users.

Figure 14: MVP diagram.

In this research, MVP is a web application product developed to test and collect
data for the next stage of development, which requires big data for deep learning. This
study mainly focuses on developing a framework as a back-end and simple front-end
for users. Figure 15 shows the level 0 data flow diagram (DFD), which is known as
context diagram and shows data system as whole and the way it interacts with external
entities.
In order to develop the product based on this research and test the iteration and collect the results and outputs, we need a user interface as the front-end that is connected with developed back-end. For this propose, Model, View, Controller (MVC) method is applied, which stands for three separated interconnected elements. MVC is a software and application design pattern used for developing interfaces. MVC is a lightweight highly testable framework as compared to traditional ASP.NET web forms. This method separates content from presentation and data processing. In other words, design pattern keeps the display and data separate to allow each to change without affecting the other and enable full control over the rendered HTML. The main
advantages of MVC method are providing clean separation of concern (SoC) and enabling test driven development (TDD). The basic interaction of MVC architecture is illustrated in Figure 16.

![Figure 16: Basic MVC architecture.](image)

Model represents the shape of data of the application. It is a central component that directly manages the data, logics and rules of application and is independent of the user interface. It receives user input from the controller.

View is a user interface that displays data, allowing users to modify the data. It actually works as the front-end in this case. All result representations, such as charts, diagrams or tables, can be displayed here.
Controller handles the user request and renders the appropriate view with the model data as a response. In other words, it accepts inputs and data, and converts them to commands for the model or view.

Figures 16 and 17 illustrate MVC architecture and the flow of users’ requests and responses in this research, and the interaction between front-end and back-end with simulation engine and creating the configurations. Figure 18 shows the coupling of simulation and optimization procedures in this study. Next chapter describes the development of the framework, coupling, and implementation in detail.
Figure 17: MVC flow chart.
The web application was developed by utilizing Laravel, a Hypertext Pre-processor (PHP) web development framework. Laravel is an open-source PHP framework that is particularly known for developing high-functional web applications built with MVC pattern. Besides the open-access, the advantages of using Laravel for developing this web application are: robust code foundation, integration with other tools.
to make web application faster, accreditation, authorization and access control, configuration error and handling tasks and integration with e-mail services and many other important tasks.

MVC structure in Laravel integrated the logic and presentation in framework and provides light-weight template for implementation. Therefore, Laravel was used to build a dynamic web-application. Figures 19, 20 and 21 illustrate the main component of the developed web-application. At the sign-in page, user entered their email contact information and sign into the website. On the main page, users can modify the parameters based on their design goals. The main page also allows the users to select the range of parameters relating the energy consumption, occupant’s comfort and the energy cost. This is meant to improve the decision-making process. After setting the parameter, web application sends the request to the back-end to run the simulation, and find the optimum scenarios based on set parameters. After completing this process on the server, selected scenarios are saved in the MYSQL database table and send to controller. The results are displayed on the web-application in a table format. In this step, the application sends a notification to the users to notify them that the results are ready to be viewed.
Figure 19: Web-application sign in page.
Figure 20: Web-application main page, which allows users to select specific parameters.
<table>
<thead>
<tr>
<th>Scenario ID</th>
<th>timeGamp</th>
<th>Energy Consumption</th>
<th>Occupant Comfort</th>
<th>Total Cost</th>
<th>EUI-Electricity for Internal Lighting</th>
<th>EUI-Electricity for Heating</th>
<th>EUI-Gas for Heating</th>
<th>PPM &amp; PPD</th>
<th>Daylight Illuminance</th>
<th>Daylight Glare</th>
<th>Total Electricity Cost</th>
<th>Total Gas Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>06</td>
<td>2019-11-13 21:53:24</td>
<td>33.59</td>
<td>14.26</td>
<td>23.45</td>
<td>41.81</td>
<td>71.62</td>
<td>0.0</td>
<td>0.52599</td>
<td>306.13</td>
<td>15.10</td>
<td>1412.72</td>
<td>1.28</td>
</tr>
<tr>
<td>87</td>
<td>2019-11-13 21:53:34</td>
<td>31.02</td>
<td>21.13</td>
<td>23.52</td>
<td>45.07</td>
<td>37.75</td>
<td>1.23</td>
<td>0.59907</td>
<td>78.35</td>
<td>3.76</td>
<td>1410.85</td>
<td>1.98</td>
</tr>
<tr>
<td>88</td>
<td>2019-11-13 21:54:24</td>
<td>30.66</td>
<td>18.86</td>
<td>20.3</td>
<td>50.64</td>
<td>64.02</td>
<td>0.73</td>
<td>0.45247</td>
<td>115.12</td>
<td>5.15</td>
<td>1433.21</td>
<td>1.77</td>
</tr>
<tr>
<td>91</td>
<td>2019-11-18 21:54:35</td>
<td>33.52</td>
<td>17.8</td>
<td>28.29</td>
<td>43.83</td>
<td>64.59</td>
<td>2.86</td>
<td>0.38768</td>
<td>300.87</td>
<td>14.81</td>
<td>1375.36</td>
<td>4.38</td>
</tr>
<tr>
<td>92</td>
<td>2019-11-13 21:53:26</td>
<td>29.69</td>
<td>23.21</td>
<td>25.17</td>
<td>66.21</td>
<td>54.36</td>
<td>3.34</td>
<td>0.25482</td>
<td>70.17</td>
<td>3.39</td>
<td>1393.36</td>
<td>5.38</td>
</tr>
<tr>
<td>93</td>
<td>2019-11-13 21:53:17</td>
<td>30.21</td>
<td>20.87</td>
<td>22.88</td>
<td>62.24</td>
<td>58.84</td>
<td>2.18</td>
<td>0.32079</td>
<td>132.23</td>
<td>4.78</td>
<td>1411.54</td>
<td>3.52</td>
</tr>
<tr>
<td>94</td>
<td>2019-11-13 21:52:27</td>
<td>29.0</td>
<td>22.35</td>
<td>22.35</td>
<td>67.45</td>
<td>56.83</td>
<td>2.88</td>
<td>0.26044</td>
<td>60.44</td>
<td>3.72</td>
<td>1413.13</td>
<td>4.61</td>
</tr>
<tr>
<td>95</td>
<td>2019-11-13 21:52:59</td>
<td>30.01</td>
<td>20.94</td>
<td>21.75</td>
<td>65.13</td>
<td>61.38</td>
<td>2.2</td>
<td>0.34156</td>
<td>115.07</td>
<td>5.03</td>
<td>1419.17</td>
<td>3.55</td>
</tr>
<tr>
<td>96</td>
<td>2019-11-13 21:53:30</td>
<td>34.16</td>
<td>15.8</td>
<td>25.75</td>
<td>43.63</td>
<td>68.19</td>
<td>0.84</td>
<td>0.46025</td>
<td>300.87</td>
<td>14.81</td>
<td>1397.65</td>
<td>1.36</td>
</tr>
<tr>
<td>97</td>
<td>2019-11-13 22:09:01</td>
<td>31.06</td>
<td>21.21</td>
<td>24.04</td>
<td>58.21</td>
<td>58.51</td>
<td>1.27</td>
<td>0.34197</td>
<td>70.17</td>
<td>3.39</td>
<td>1407.38</td>
<td>2.05</td>
</tr>
<tr>
<td>98</td>
<td>2019-11-13 22:09:32</td>
<td>30.77</td>
<td>18.67</td>
<td>21.09</td>
<td>52.24</td>
<td>62.46</td>
<td>0.76</td>
<td>0.49496</td>
<td>102.53</td>
<td>4.78</td>
<td>1428.11</td>
<td>1.22</td>
</tr>
<tr>
<td>99</td>
<td>2019-11-13 22:01:03</td>
<td>30.06</td>
<td>20.3</td>
<td>21.0</td>
<td>67.45</td>
<td>59.14</td>
<td>1.95</td>
<td>0.27904</td>
<td>60.44</td>
<td>3.72</td>
<td>1427.70</td>
<td>1.69</td>
</tr>
<tr>
<td>100</td>
<td>2019-11-13 22:01:36</td>
<td>30.59</td>
<td>18.94</td>
<td>16.97</td>
<td>61.13</td>
<td>63.88</td>
<td>0.77</td>
<td>0.49934</td>
<td>115.07</td>
<td>5.03</td>
<td>1435.35</td>
<td>1.24</td>
</tr>
</tbody>
</table>

Figure 21: Web-application result page
CHAPTER 4

RESEARCH METHODOLOGY

4.1 Overview of Research Methodology

The research process for this dissertation included multiple methods. The first step focused on identifying building simulation programs, design optimization methods and outlining potential challenges and obstacles in performance-based design techniques. The primary research method included literature reviews, addressing research questions posed in Objective 1. The results will help to obtain the clear understanding of the latest simulation programs and optimization methods for facade design, and thermal and daylight modeling in facade design process that contribute to decrease energy use and meet the occupant comfort needs.

The next step is the main part of the dissertation, and it focused on developing a framework for optimum facade design, considering energy usage, energy costs and occupants’ comfort. Energy modeling is associated with improvement of the design process, while the optimization is applied during the decision-making process. Specific research methods for both parts are described in more detail in the following sections.

4.1.1 Research Methods for Objective 1

For the first objective, research methods included literature reviews of publications relating to BPS, BPO, and existing optimization and simulation tools, followed by determination of how building performance simulation tools are currently used, and what major limitations and problems currently exist. Then, occupant comfort
criteria and measurement methods were investigated. Research problems within the BPS community were identified, and the comprehensive literature review was conducted to define the role of simulation in performance-based design, methods and tools. This review and analysis identified issues and problems with the current methods and tools used in Architectural, Engineering and Construction (AEC) industry. Providing an overview of building simulation programs and classification of both optimization problems and algorithms is an important basis for developing a new framework and new strategies.

4.1.2 Research Methods for Objective 2

The new framework for performance-based design approach was developed as part of this research. The main goal is to minimize building energy consumption and energy costs associated with facade design, while considering occupants’ comfort. This is a modular framework, consisting of independent scripts that represent modules, steps and functions of application under test. Figure 23 shows the major steps and tasks of the study. The modules are used in a hierarchical fashion to apply the framework, consisting of four steps for the main framework as backend and the fifth step is the additional step for the front end:

1) Defining goals, performance criteria, facade design variables and their properties, acceptable ranges for high-performance facade design

2) Generating the database that includes all possible design scenarios based on the variables with permutation in Python and selected outputs after simulation in EnergyPlus. This is module 1.
3) Coupling Python script with simulation engine (EnergyPlus) to automatically perform simulations for scenarios from a database (measurements methods) to quantify variables and generate the needed outputs. This is module 2.

4) Filtering and narrowing down the results by implementing Python script, genetic algorithm (GA) and Batch Normalization to evaluate outputs and find the optimal scenarios. This is module 3. Figure 22 illustrates these modules as the main part of this framework (back end).

5) Developing a web-based front-end interface for users to test the framework and collect the data for next step, which is implementation of deep learning after collecting enough data.

The next sections discuss the model development, components of the framework and its implementation in detail.

Figure 22: The Framework Back-End Diagram (Steps 1-4).
Figure 23: Major steps (1-5) and tasks in the study.
4.2 Step 1: Defining Goals, Performance Criteria and Facade Design Variables

Figure 24 shows the components of the framework. Performance-based facade design requires a holistic approach, considering performance indicators, such as energy performance and human comfort. The goals (optimization objectives) for this framework are to aid the design decision making process, where energy consumption and cost are minimized, and occupants' comfort (thermal and visual) is maximized. The energy requirements for heating, cooling, and lighting of buildings are strongly driven by the performance of the facade, especially glazing parts. The objectives for reducing energy consumption are to reduce heating, cooling and lighting loads. Performance requirements (design variables) to meet this objective include window to wall ratio (WWR), wall assembly, insulation, solar control, and glazing system. These design variables are considered in this research. Performance-based facade design objectives that relate to human factors and contribute to occupants' comfort and satisfaction include thermal comfort and visual comfort. The variables that relate to facade design include air temperature, mean radiant temperature, air movement, relative humidity, clothing levels and activity levels. The Predictive Mean Vote (PMV), suggested by Fanger (1970), predicts the effects of these six factors on thermal comfort. Predicted Percentage of Dissatisfied (PPD) persons predicts the percentage of people who would feel discomfort with certain thermal conditions.
In summary, this research investigates how optimization objectives (energy consumption and cost, and occupant comfort) are treated and set up in different phases, and what approach is more desirable for this multi-objective problem.

Figure 24: Conceptual diagram, showing components of the framework.

After defining all design requirements, performance criteria and variables, design scenarios were developed and prepared for the next phase of research, which included simulations and analysis. To investigate all these strategies, the basic procedures included: defining design information, building simulation models,
developing design scenarios, and simulating scenarios using different tools, as shown in Figure 25.

![Diagram of a framework with four layers: Design Information, Building Models, Scenarios, and Tools & Application.]

Figure 25: Basic procedure for applying the framework in this research.

The building design information relies on general data design requirements, such as weather, location, orientation and geometry, occupancy, etc. The building model layer contains analysis models that can be used for specific analysis or performance aspect. The scenario layer is based on different facade design scenarios, specifically looking into related performance criteria (variables) that will be simulated. The last layer will employ building performance analysis tools to perform the analysis.

Climate and environmental factors are the first aspects that should be considered for high-performance facade design. There are three general climate types: heating-dominated, cooling-dominated and mixed climates. Heating-dominated climates benefit from the collection of solar radiation, passive heating, heat storage, improved insulation to reduce heating demand, and the use of daylighting to reduce lighting demand. In cooling-dominated climates, protection from sun and direct solar radiation become
more important. In mixed climates, combined strategies that balance solar exposure and access to daylight must be implemented (Aksamija 2015).

The design scenarios were differentiated based on the following variables and related facade elements, properties, measurements and simulation tools, as listed in Tables 3 and 4. All variables in this research are discrete, which means that they can only take certain numerical values and are counted:

1. **Facade wall type assembly**: Different construction types were considered, including metal and wood framing with different cladding materials: Aluminum, Zinc, Stainless Steel, Cement, Terracotta, Brick and Stone. Table 11 shows the properties of these material as EnergyPlus inputs.

2. **Insulation**: Different insulation types, thicknesses, and placements within the wall assembles were considered. The type of insulation materials and equivalent insulation thickness considered in this research are listed in Table 12.

3. **Transparency**: Varying WWR (20 to 60%) and glazing system types were considered. Glazing system properties and specification are listed in Table 13.

4. **Solar control**: Different exterior blinds were considered (horizontal and vertical blinds).

Table 6: Facade design variables used in this study, and specific parameters.

<table>
<thead>
<tr>
<th>Design Variables</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window to Wall Ratio (WWR)</td>
<td>20%</td>
</tr>
<tr>
<td>(3 Types)</td>
<td>40%</td>
</tr>
<tr>
<td>(3 Types)</td>
<td>60%</td>
</tr>
<tr>
<td>Glazing System (21 Types)</td>
<td>1A: Sgl-Clr-Alum</td>
</tr>
<tr>
<td></td>
<td>1B: Sgl-Clr-AlumTB</td>
</tr>
<tr>
<td></td>
<td>1C: Sgl-Clr-W/F</td>
</tr>
<tr>
<td></td>
<td>2A: Dbl-Clr-Air-Alum</td>
</tr>
<tr>
<td></td>
<td>2B: Dbl-Clr-Air-AlumTB</td>
</tr>
</tbody>
</table>
| Wall Assembly (3 Types) | W1: Wood Framing  
W2: Steel Framing |
| Cladding Material (8 Types) | C1: Aluminum (AL)  
C2: Zinc  
C3: Stainless Steel (SS)  
C4: Terracota (TR)  
C5: Brick (BR)  
C6: Stone (ST)  
C7: Cement (CM) |
| Insulation (12 Types) | I1: 2x6 wood @16” +3” XPS Styrofoam  
I2: 2x6 wood @16” +4” XPS Styrofoam  
I3: 2x6 wood @16” +1” Aerogel  
I4: 2x6 wood @24” +3” XPS Styrofoam  
I5: 2x6 wood @24” +4” XPS Styrofoam  
I6: 2x6 wood @24” +1” Aerogel  
I7: 2x6 metal @16” +3” XPS Styrofoam  
I8: 2x6 metal @16” +4” XPS Styrofoam  
I9: 2x6 metal @16” +1” Aerogel  
I10: 2x6 metal @24” +3” XPS Styrofoam  
I11: 2x6 metal @24” +4” XPS Styrofoam  
I12: 2x6 metal @24” +1” Aerogel |
| Solar Control (5 Types) | S1: No Shading  
S2: Alum-H-45  
S3: Alum-H-90  
S4: Alum-V-45  
S5: Alum-V-90 |
Table 7: List of facade elements, variables, properties and simulation tools used for design scenarios.

<table>
<thead>
<tr>
<th>Facade Elements</th>
<th>Scenario Variables</th>
<th>Main Functionality</th>
<th>Properties</th>
<th>Simulation Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wall Assembly</td>
<td>Metal &amp; Wood Framing with Aluminum, Zinc, Stainless Steel, Terracotta, Brick, Stone, Cement</td>
<td>Energy gain</td>
<td>Heat capacity</td>
<td>EnergyPlus</td>
</tr>
<tr>
<td>Insulation</td>
<td>Metal &amp; Wood Framed @16” &amp; @24” 3” &amp; 4” XPS Styrofoam, 1” Aerogel</td>
<td>Energy gain</td>
<td>Heat capacity</td>
<td>EnergyPlus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Energy loss</td>
<td>Thermal conductivity</td>
<td></td>
</tr>
<tr>
<td>Transparency</td>
<td>Window to Wall Ratio 20%-40%-60%</td>
<td>Energy transfer</td>
<td>Flow rate</td>
<td>EnergyPlus</td>
</tr>
<tr>
<td>Solar Control</td>
<td>Glazing System Clear Glass-Double Clear Glass-Triple with Air, Argon Low e-Double with Argon Low e -Triple with Argon No Shading Aluminum Horizontal-Vertical 45 &amp; 90 degree</td>
<td>Energy transfer</td>
<td>SHGC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lighting</td>
<td>U-value</td>
<td>VT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Energy transfer</td>
<td>VT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R-value</td>
<td>VT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lighting</td>
<td>VT</td>
<td></td>
</tr>
</tbody>
</table>
The fixed parameters that are used in this study include geometry and size of the space used for simulations (40'x40'x10' H test cell), building type (office), HVAC equipment, operating schedule, equipment loads, lighting loads and system ranges, as shown in Table 8. Different visual configurations of the test cell are shown in Figure 26. This figure reflects the design variables, such as window to wall ratio and shading control options considered in this research.

Table 8: List of fixed parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Range</td>
<td>Heating temperatures set points: upper &amp; lower band 68-77 °F (20-25 °C)</td>
</tr>
<tr>
<td></td>
<td>PMV range: -0.5 to +0.5</td>
</tr>
<tr>
<td></td>
<td>10% PPD</td>
</tr>
<tr>
<td></td>
<td>Operating hours for analysis: 9 AM-5 PM</td>
</tr>
<tr>
<td></td>
<td>Num. of people per area</td>
</tr>
<tr>
<td></td>
<td>Lighting Loads: 10 W/m2 Plug &amp; Process Loads: 25 W/m2 People: 10 m2 per person</td>
</tr>
<tr>
<td>Fixed Parameters</td>
<td>Type: Medium office</td>
</tr>
<tr>
<td></td>
<td>Floor area: 1600 ft²</td>
</tr>
<tr>
<td></td>
<td>Floor U-value: 0.10 Btu/ h ft² F</td>
</tr>
<tr>
<td></td>
<td>Zones program: Open office</td>
</tr>
<tr>
<td></td>
<td>Relative humidity set points: upper band 60 % RH lower band 30 % RH</td>
</tr>
<tr>
<td></td>
<td>Operating hours for analysis: 9 AM-5 PM</td>
</tr>
<tr>
<td></td>
<td>Num. of people per area</td>
</tr>
<tr>
<td></td>
<td>Lighting Loads: 10 W/m2 Plug &amp; Process Loads: 25 W/m2 People: 10 m2 per person</td>
</tr>
</tbody>
</table>

HVAC system for this research is packaged rooftop VAV with reheat. Variable air volume (VAV) system Return Air Package system incorporates one supply duct that, in cooling mode, distribute supply air at constant temperature of approximately 55°F (13 °C). Because the supply air temperature is constant, the air flow rate must vary to meet the rising and falling heat gains or losses within the thermal zone served. Table 9 describes the key properties of the Air Handling Unit (AHU) for this system.
Table 9: AHU features.

<table>
<thead>
<tr>
<th>AHU Feature</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply Air Temperature</td>
<td>Constant (13C/55F)</td>
</tr>
<tr>
<td>Supply Air Temperature Reset Controls</td>
<td>None</td>
</tr>
<tr>
<td>Supply Air Pressure Reset Controls</td>
<td>None</td>
</tr>
<tr>
<td>Peak Fan Power</td>
<td>3.5 W/(L/s) or 1.65 W/ cfm</td>
</tr>
<tr>
<td>Fan Curve</td>
<td>Standard ASHRAE</td>
</tr>
<tr>
<td>Economizer Cycle</td>
<td>Dry-bulb Control with 19C threshold</td>
</tr>
<tr>
<td>Minimum Outside Air</td>
<td>Sum of all zones minimum ventilation air</td>
</tr>
<tr>
<td>Heating Coil</td>
<td>Hot water from gas boiler</td>
</tr>
<tr>
<td>Heat Recovery</td>
<td>Via return air only (not indirect)</td>
</tr>
<tr>
<td>Cooling Source</td>
<td>Chilled Water</td>
</tr>
</tbody>
</table>

This multi-zone VAV system served by a rooftop AHU provides ventilation and cooling. Each zone contains a VAV box which varies the flow of air for zone-level cooling and/or heats the air to provide heating. There is a central chilled water plant that provides cooling to the air handling unit. The central plant properties are listed in Table 10.
Table 10: Central plant features.

<table>
<thead>
<tr>
<th>Central Plant Feature</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heating Hot Water Source</td>
<td>Natural Gas</td>
</tr>
<tr>
<td>Boiler Peak Efficiency</td>
<td>Assumed to be 0.9</td>
</tr>
<tr>
<td>Supply and Return Temperatures-hot water</td>
<td>80/70 C or 176/158 F</td>
</tr>
<tr>
<td>System losses</td>
<td>Constant efficiency of 15%</td>
</tr>
<tr>
<td>Peak Pumping Energy</td>
<td>1.32 W/(L/s) or 21 W/gpm</td>
</tr>
<tr>
<td>Chilled Water Source</td>
<td>Water-cooled Chiller</td>
</tr>
<tr>
<td>Chiller Peak Load Efficiency</td>
<td>5.50</td>
</tr>
<tr>
<td>Supply and Return Temperatures- chilled water</td>
<td>7/12 C or 44.6/53.6 F</td>
</tr>
<tr>
<td>Peak Pumping Energy</td>
<td>1.32 W/(L/s) or 21 W/gpm</td>
</tr>
<tr>
<td>Pump Efficiency Controls</td>
<td>Variable Speed Drives</td>
</tr>
<tr>
<td>Condenser Water Source</td>
<td>Cooling Tower</td>
</tr>
<tr>
<td>Cooling Tower Efficiency</td>
<td>245.7 W/(L/s) or 15.5 W/gpm</td>
</tr>
<tr>
<td>Supply and Return Temperatures-condenser water</td>
<td>29/35 C or 84.2/95 F</td>
</tr>
<tr>
<td>System losses</td>
<td>Constant efficiency at 95%</td>
</tr>
<tr>
<td>Peak Pumping Energy</td>
<td>0.44 W/(L/s) or 7 W/gpm</td>
</tr>
</tbody>
</table>
4.3 Step 2: Creating the Database (MYSQL)

After setting all variables and parameters for facade design, all possible design scenarios are generated using Python programming. With permutation in Python script, design scenarios are generated and added to database with specific scenario ID. In this study, we have 26,460 scenarios to investigate for the test cell, described in the next
After running simulations in EnergyPlus, all outputs are populated in this database with identical scenario ID. EnergyPlus provides a wide range of outputs, but for this purpose, the following results are obtained:

- Cooling, heating, and lighting loads
- Energy Use Intensity (EUI) for electricity and gas
- PMV and PPD
- Daylight Illuminance and glare index, and
- Total energy costs for electricity and gas.

Module 1 is responsible for generating all scenarios with defined variables, and for populating these scenarios within the database. Module 2 is responsible for automatically sending these scenarios to the simulation engine and for populating the selected outputs in the database. Data Flow Diagram (DFD) in Figure 27 shows the overview of the framework system, its implementation, and the flow of data through this process. Figures 28 to 31 show some part of the scripts for this framework that created the database. These figures show the scripts to create all scenarios and collecting output to create a database in MYSQL. Figures 32 and 33 show this database in MYSQL representation.

The MYSQL database manages all scenario inputs and outputs. MYSQL is an open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) for adding, accessing, and managing content in the database.
MYSQL database is free and easy to set up and use. The advantages of this type of database for the purpose of this research is the quick processing time, proven reliability, open source, ease and flexibility of use.

Furthermore, the interface allows for easy integration of different types of optimization algorithms that solve the constrained nonlinear optimization problems. Developing an object-oriented framework is beneficial in this case because it can easily be reused in other programs and parallelize the function evaluations. This method is rarely used in early design process, consequently, optimization with this method will be integrated with the design process. The benefits of this kind of framework are:

- Rapid generation and evaluation of designs alternatives and scenarios

- Support of different types of decision-making process while providing all possible scenarios and narrowing down optimized solutions based on objective functions

- Supporting designers’ ability to solve nonlinear and multi-criteria problems

- Providing more flexible and faster supplement in facade design process while facilitating simulations of multiple design scenarios.
Figure 27: Data Flow Diagram (DFD) for the framework implementation.

Figure 28: Part of scripts to create and permute all scenarios table in MYSQL.
Figure 29: Part of scripts to create table for outputs in MYSQL.
Figure 30: Scripts used to couple database with EnrgyPlus output results table, and to save selected output to the database.
Figure 31: Scripts used to save selected outputs from EnergyPlus report table to save into database table.
### Figure 32: MYSQL-all scenarios table.

<table>
<thead>
<tr>
<th>ID</th>
<th>Scenario</th>
<th>WWR</th>
<th>Wall_Cladding</th>
<th>Wall_Assembly</th>
<th>Insulation</th>
<th>Glazing_System</th>
<th>Solar_Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-Alum</td>
<td>No Shading</td>
</tr>
<tr>
<td>2</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-Alum</td>
<td>Alum-H-45</td>
</tr>
<tr>
<td>3</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-Alum</td>
<td>Alum-H-30</td>
</tr>
<tr>
<td>4</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-Alum</td>
<td>Alum-V-45</td>
</tr>
<tr>
<td>5</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-Alum</td>
<td>Alum-V-90</td>
</tr>
<tr>
<td>6</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-Alum</td>
<td>No Shading</td>
</tr>
<tr>
<td>7</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-Alum</td>
<td>Alum-H-45</td>
</tr>
<tr>
<td>8</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-Alum</td>
<td>Alum-H-90</td>
</tr>
<tr>
<td>9</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-Alum</td>
<td>Alum-V-45</td>
</tr>
<tr>
<td>10</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-Alum</td>
<td>Alum-V-90</td>
</tr>
<tr>
<td>11</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-WF</td>
<td>No Shading</td>
</tr>
<tr>
<td>12</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-WF</td>
<td>Alum-H-45</td>
</tr>
<tr>
<td>13</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-WF</td>
<td>Alum-H-90</td>
</tr>
<tr>
<td>14</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-WF</td>
<td>Alum-V-45</td>
</tr>
<tr>
<td>15</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Sgl-Cl-WF</td>
<td>Alum-V-90</td>
</tr>
<tr>
<td>16</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Dbl-Cl-Air-4in</td>
<td>No Shading</td>
</tr>
<tr>
<td>17</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Dbl-Cl-Air-4in</td>
<td>Alum-H-45</td>
</tr>
<tr>
<td>18</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Dbl-Cl-Air-4in</td>
<td>Alum-H-90</td>
</tr>
<tr>
<td>19</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Dbl-Cl-Air-4in</td>
<td>Alum-V-45</td>
</tr>
<tr>
<td>20</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Dbl-Cl-Air-4in</td>
<td>Alum-V-90</td>
</tr>
<tr>
<td>21</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Dbl-Cl-Air-4in</td>
<td>Alum-V-45</td>
</tr>
<tr>
<td>22</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Dbl-Cl-Air-4in</td>
<td>Alum-H-45</td>
</tr>
<tr>
<td>23</td>
<td>SteelFramed</td>
<td>20%</td>
<td>AL</td>
<td>Steel-Framed</td>
<td>2x6 metal @16 + 3 XPS Styrofoam</td>
<td>Dbl-Cl-Air-4in</td>
<td>Alum-H-90</td>
</tr>
</tbody>
</table>

### Figure 33: MYSQL-output table.

- **Scenario**: SteelFramed
- **WWR**: 20%
- **Wall_Cladding**: AL
- **Wall_Assembly**: Steel-Framed
- **Insulation**: 2x6 metal @16 + 3 XPS Styrofoam
- **Glazing_System**: Sgl-Cl-Alum
- **Solar_Control**: No Shading
4.4 Step 3: Coupling Python Script with Simulation Engine (EnergyPlus)

EnergyPlus 8.5 is used in this framework as an energy modeling engine and simulator. EnergyPlus has been chosen as BPS tool for two main reasons: (a) this program allows reliable modeling of both building and HVAC systems, and (b) it works with text-based inputs and outputs, which enable the interaction with Python scripts.

EnergyPlus can investigate discussed variables as inputs and simulate envelope related outputs in the study. Thermal comfort is calculated based on PMV and PPD. The formulas for both PMV and PPD are built into EnergyPlus and their values can be obtained directly from the simulation output file.

A simulation test cell was used to investigate the functionality and application of this framework. The test cell is an office space (40’ x 40’ x 10’), located in Atlanta, Georgia (Climate zone 3A). The south-facing facade under different scenarios was modeled in EnergyPlus 8.5. Impact of different facade types, WWR, glazing systems, and shading control was investigated in a simulation study. All related parameters for the design variables were specified in the IDF editor in EnergyPlus. However, they were specified in a certain way so that Python script can call these variables and send it to EnergyPlus engine to automatically simulate all these scenarios in the database. Then, the determined output is stored in the database for filtering step.

Verification and finding related parameters as inputs, and setting data needed as outputs are the key elements for connecting the design scenario in database with the simulation engine. Python script works as the interface to call scenarios from the database and sends them to the simulator. Each parameter must identify a well-defined
relation with discussed variables, which reveals facade behavior in relation to
performance aspects being analyzed.

Using a correct simulation methodology and the appropriate level of modeling
resolution are two important elements that should be paid attention to when using
computer simulations in the context of building design. In this study, Python scripts
connect directly to EnergyPlus because other current simulation tools do not provide
proper Application Programming Interface (API) to connect these scripts. Next section
explains the modeling process in EnergyPlus.

4.4.1 Modeling in EnergyPlus (Simulation Engine)

This section discusses the difference facade variables/parameters (wall
assembly, insulation, solar control, and glazing system) are modeled in the EnergyPlus.

Important features include layer-by-layer input of custom glazing, cladding and
insulation, ability to accept different glass optical properties, setting up material thermal
properties, incidence angle-dependent solar and visible transmission and reflection,
iterative heat balance solution to determine glass surface temperatures, calculation of
frame and divider heat transfer, and modeling of movable interior or exterior shading
devices with user-specified controls. Setting up these variables in EnergyPlus for the
test cell are explained in more details in this section.

4.4.2 Wall Assemblies (Cladding & Framing)

Definition of cladding material for facade should be defined with four main
thermal properties (thickness, conductivity, density, and specific heat) of the material.
This syntax is used to describe opaque construction elements only, especially for
building surfaces. The building surface models are for normal applications to building energy efficiency where the main focus is on assemblies with some thermal resistance. Extremely thin should be neglected from the construction rather than included because they will not contribute to the assembly’s overall thermal resistance or heat capacity. For some cases, thin and/or highly conductive materials are a serious problem for the heat transfer modeling and the values for thickness, conductivity, density and specific heat are checked for appropriateness. Table 11 presents the properties and variables for cladding materials that were used in the study.

Table 11: Cladding material properties for EnergyPlus inputs.

<table>
<thead>
<tr>
<th>Material (outside layer)</th>
<th>Name (obj)</th>
<th>Roughness</th>
<th>Thickness</th>
<th>Thermal Conductivity</th>
<th>Density</th>
<th>Specific Heat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aluminum (outside layer)</td>
<td>AL</td>
<td>Smooth</td>
<td>5 mm (0.005 m) (0.197 in)</td>
<td>200 (W/m K) 0.385 (Btu./In/s. ft². F)</td>
<td>2700 (k/m³) 168.5 (lb/ft³)</td>
<td>860 (J/Kg K) 0.2 (Btu/lb. F)</td>
</tr>
<tr>
<td>Zinc (outside layer)</td>
<td>Zinc</td>
<td>Medium Smooth</td>
<td>0.8 mm (0.0008 m) (0.032 in)</td>
<td>110 (W/m k) 0.212 (Btu./In/s. ft². F)</td>
<td>7100 (K/m³) 443.2 (lb/ft³)</td>
<td>350 (J/kg K) 0.08 (Btu/lb. F)</td>
</tr>
<tr>
<td>Stainless Steel (outside layer)</td>
<td>SS</td>
<td>Smooth</td>
<td>2 mm (0.02 m) (0.079 in)</td>
<td>40 (W/m k) 0.078 (Btu./In/s. ft². F)</td>
<td>7500 (k/m³) 468.2 (lb/ft³)</td>
<td>480 (J/kg k) 0.11 (Btu/lb. F)</td>
</tr>
<tr>
<td>Terracotta (outside layer)</td>
<td>TR</td>
<td>Rough</td>
<td>30 mm (0.03 m) (1.18 in)</td>
<td>0.60 (w/m k) 0.0011 (Btu./In/s. ft². F)</td>
<td>1120 (k/m³) 69.9 (lb/ft³)</td>
<td>830 (J/kg k) 0.198 (Btu/lb. F)</td>
</tr>
<tr>
<td>Brick (outside layer)</td>
<td>BR</td>
<td>Rough</td>
<td>100 mm (0.1 m) (3.94 in)</td>
<td>0.80 (w/m k) 0.0015 (Btu./In/s. ft². F)</td>
<td>1700 (k/m³) 106.1 (lb/ft³)</td>
<td>800 (J/kg k) 0.191 (Btu/lb. F)</td>
</tr>
<tr>
<td>Stone (outside layer)</td>
<td>ST</td>
<td>Very Rough</td>
<td>75 mm (0.075)</td>
<td>1.8 (w/m k) 0.0035 (Btu./In/s. ft². F)</td>
<td>2100 (k/m³)</td>
<td>850 (J/kg k)</td>
</tr>
</tbody>
</table>
### Insulation (Cavity & Continuous)

The insulation included both continuous and cavity insulation, as well as equivalent insulation thickness for each variable. Table 12 represents specifications of the insulation variables for this research.

**Table 12: Specifications of Insulation Variables**

<table>
<thead>
<tr>
<th>Material</th>
<th>Type</th>
<th>Thickness</th>
<th>Thermal Resistance</th>
<th>Density</th>
<th>Specific Heat</th>
<th>Thermal Conductivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cement (outside layer)</td>
<td>CM</td>
<td>Very Rough</td>
<td>6 mm (0.006 m)</td>
<td>0.35 (w/ m k)</td>
<td>0.0007 (Btu. In/s. ft². F)</td>
<td>1350 (k/m³)</td>
</tr>
<tr>
<td>Concrete Exposed (Inside Layer)</td>
<td>Con</td>
<td>Rough</td>
<td>150 mm (0.15 m)</td>
<td>1.7 w/ m k</td>
<td>0.0033 (Btu. In/s. ft². F)</td>
<td>2100 (k/m³)</td>
</tr>
</tbody>
</table>

**Figure 34:** Scripts used to define WWR input and couple simulation model to the EnergyPlus engine.

```python
def updateData(alldata):
    print alldata
    #--------------------------------- Window
    window = idf.idfobjects['window'].upper()()[0]

    # print window['length']
    # print window['height']

    length = 9
    height = { '20K' : 0.6, '40K' : 1.2, '60K' : 1.8, '80K' : 2.45, '90K' : 2.75 }
    print window

    window['length'] = length
    print (calldata)
    window['Height'] = height[calldata[1]]

    #--------------------------------- Material
    Construction = idf.idfobjects['Construction'].upper()()[9]

    Construction['Outside_Layer'] = alldata[2]
    Construction['Layer_3'] = 'Conc 30mm'

    print (Construction)
```
Continuous insulation is insulation that runs continuously over structural members and is free of significant thermal bridging, such as rigid foam insulation above the ceiling deck. It is installed on the interior, exterior, or is integral to any opaque surface of the building envelope. Cavity insulation is the insulation installed between structural members such as wood studs and metal framing.

And equivalent insulation thickness shows the thickness of Fiberglass Batt insulation that equals to a unit of each specific insulation that listed in the table. In this way, we can have equivalent thickness for both cavity and continuous and add both as composite assembly U-value to define it EnergyPlus.

Table 12: Equivalent insulation thickness and thermal properties.

<table>
<thead>
<tr>
<th>Thermal Bridge Assemblies</th>
<th>R-Value</th>
<th>U-Value</th>
<th>Equivalent Fiberglass Batt Thickness</th>
<th>R-Value</th>
<th>U-Value</th>
<th>Equivalent Fiberglass Batt Thickness</th>
</tr>
</thead>
<tbody>
<tr>
<td>2x6 wood @16”</td>
<td>14.9</td>
<td>0.07</td>
<td>4.66</td>
<td>2.63</td>
<td>0.38</td>
<td>118</td>
</tr>
<tr>
<td>2x6 wood @24”</td>
<td>15.4</td>
<td>0.07</td>
<td>4.81</td>
<td>2.71</td>
<td>0.37</td>
<td>122</td>
</tr>
<tr>
<td>2x6 metal @16”</td>
<td>7.4</td>
<td>0.14</td>
<td>2.31</td>
<td>1.30</td>
<td>0.77</td>
<td>59</td>
</tr>
<tr>
<td>2x6 metal @24”</td>
<td>9.0</td>
<td>0.11</td>
<td>2.81</td>
<td>1.58</td>
<td>0.63</td>
<td>71</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Continuous Material</th>
<th>R-Value</th>
<th>U-Value</th>
<th>Equivalent Fiberglass Batt Thickness</th>
<th>R-Value</th>
<th>U-Value</th>
<th>Equivalent Fiberglass Batt Thickness</th>
</tr>
</thead>
<tbody>
<tr>
<td>3” XPS Styrofoam</td>
<td>15.0</td>
<td>0.07</td>
<td>4.69</td>
<td>2.64</td>
<td>0.38</td>
<td>119</td>
</tr>
<tr>
<td>4” XPS Styrofoam</td>
<td>20.0</td>
<td>0.05</td>
<td>6.25</td>
<td>3.52</td>
<td>0.28</td>
<td>159</td>
</tr>
<tr>
<td>1” Aerogel</td>
<td>10.0</td>
<td>0.10</td>
<td>3.13</td>
<td>1.76</td>
<td>0.57</td>
<td>79</td>
</tr>
<tr>
<td>Fiberglass Batt</td>
<td>3.20</td>
<td>0.31</td>
<td>0.56</td>
<td>1.78</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.4.4 Glazing Systems (Glass & Frame Specification)

Glazing systems are combination of different glass types, air gap between the glass lites and different frame types, such as aluminum, thermally broken and wood or fiberglass framing. Table 13 shows different glazing systems and their specifications that were considered in this study, including visual transmission, solar heat gain coefficient and U-values. Figure 35 shows the part of scripts to define this variable for EnergyPlus.

Table 13: Glazing systems (glass & frame specifications).

<table>
<thead>
<tr>
<th>Glass &amp; Frame Specification</th>
<th>Glass &amp; Frame</th>
<th>Glass &amp; Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Aluminum</td>
<td>Thermally</td>
</tr>
<tr>
<td>Glass</td>
<td></td>
<td>Broken Aluminum</td>
</tr>
<tr>
<td>Single Clear</td>
<td>0.66</td>
<td>7.39</td>
</tr>
<tr>
<td>Double Clear (Air)</td>
<td>0.59</td>
<td>4.60</td>
</tr>
<tr>
<td>Double Low-e Clear (Air)</td>
<td>0.52</td>
<td>3.81</td>
</tr>
<tr>
<td>Double Low-e Clear (Argon)</td>
<td>0.52</td>
<td>3.64</td>
</tr>
<tr>
<td>Triple Low-e Clear (Argon)</td>
<td>0.5</td>
<td>3.01</td>
</tr>
<tr>
<td>Double Clear, High-SHGC Low-e (Air)</td>
<td>0.58</td>
<td>3.81</td>
</tr>
<tr>
<td>Triple Clear, High-SHGC Low-e (Argon)</td>
<td>0.52</td>
<td>3.01</td>
</tr>
</tbody>
</table>
4.4.5 Solar Control (Shading)

EnergyPlus was used to determine the incident direct and diffuse radiation on a window surface with and without external shading. Window shading with coverings like drapes, blinds, screens or pull-down shades can be used to reduce the amount of solar radiation entering the window or reduce glare. It can also be used to reduce heat loss through the window (movable insulation). Leaving the window covering open in the winter can maximize solar heat gain and thereby reduce heating loads.
In EnergyPlus with WindowProperty:ShadingControl, which is referenced by windows and glass doors (FenestrationSurface:Detailed with Type = Window or GlassDoor), we can specify the type and location of the shading device, what variable or combination of variables controls deployment of the shading device, and what the control set point is. If the shading device is a blind, also it is possible to specify how the slat angle is controlled. Shading device can be inside the window (Shading Type = InteriorShade or InteriorBlind), outside the window (Shading Type = ExteriorShade or ExteriorBlind), or between panes of glass (ShadingType=BetweenGlassShade or BetweenGlassBlind). The exception is window screens, which can only be outside the window (Shading Type = ExteriorScreen). In this study, exterior blinds with different slat angles were considered for the solar control variables. Table 14 shows the parametric variations for the shading devices in this research. Figure 36 shows part of the scripts that define and send the inputs for the EnergyPlus engine.

Table 14: Shading control (exterior shading).

<table>
<thead>
<tr>
<th>Material</th>
<th>Slat angle</th>
<th>Distance between</th>
<th>Slat depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Shading</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alum-H-45</td>
<td>Aluminum</td>
<td>45</td>
<td>9 (cm)-3.5 (in)</td>
</tr>
<tr>
<td>Alum-H-90</td>
<td>Aluminum</td>
<td>90</td>
<td>9 (cm)-3.5 (in)</td>
</tr>
<tr>
<td>Alum-V-45</td>
<td>Aluminum</td>
<td>45</td>
<td>9 (cm)-3.5 (in)</td>
</tr>
<tr>
<td>Alum-V-90</td>
<td>Aluminum</td>
<td>90</td>
<td>9 (cm)-3.5 (in)</td>
</tr>
</tbody>
</table>
4.4.6 Daylight Simulation

For evaluating daylight in this study, the reference point was set close to back of the test cell to calculate illuminance and glare index. The basic approach is to divide the window into small rectangular elements and find the daylight reaching the reference point directly from each element taking into account the luminance of the sky, the angle of incidence of light on the element, and the visible transmittance of the glazing at this angle. Considering all of these elements, the total direct illuminance at the reference point is calculated and the illuminance due to light that reaches the reference point after reflecting from room surfaces. The ratio of interior illuminance to exterior horizontal...
illuminance gives daylight factors for hourly sun positions along the same solar paths for which beam solar shadowing is calculated.

The daylight factors are determined separately for four different sky types: clear, clear turbid, intermediate (partly cloudy) and overcast. In the time-step calculation, the daylight factors are interpolated for the actual sun position, and then weighted according to the fraction of each sky type (determined from beam normal and total horizontal solar from the weather file) that is present at that time step. Daylight illuminance and glare index are calculated in this study to evaluate occupants’ visual comfort. Table 15 show the illuminance level applied for analysis in this research. Table 16 shows the Daylight Glare Index (DGI) range for analysis in this research.

Table 15: Illuminance level analysis.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illuminance &lt; 300 Lux</td>
<td>Insufficient</td>
</tr>
<tr>
<td>Illuminance 300 to 1000 Lux</td>
<td>Suitable</td>
</tr>
<tr>
<td>Illuminance 1000 to 2000 Lux</td>
<td>Great</td>
</tr>
<tr>
<td>Illuminance 2000 to 3000 Lux</td>
<td>Admissible</td>
</tr>
</tbody>
</table>

Table 16: Daylight Glare Index analysis.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Below 16</td>
<td>Imperceptible</td>
</tr>
<tr>
<td>16-20</td>
<td>Perceptible</td>
</tr>
<tr>
<td>20-22</td>
<td>Acceptable</td>
</tr>
<tr>
<td>22-28</td>
<td>Unconfutable</td>
</tr>
<tr>
<td>Above 28</td>
<td>Intolerable</td>
</tr>
</tbody>
</table>
4.4.7 Cost Simulation

EnergyPlus contains several inputs that enable users to perform economic analysis. When these inputs are included in the model, new data output reports are automatically generated under economic section. The inputs for cost estimations are:

- Component cost: line item
- Component cost: adjustment
- Component cost: references

These inputs are related to determining the design and construction cost and other costs related to permitting, insurance and commissioning fees and etc. Life cycle analysis cost can also be performed but is not in the scope of this research.

The other part of cost estimation in EnergyPlus relates to computation of estimated utility bills, which can provide an understanding of energy cost saving. These cost types are included in this research. The goals of the UtilityCost input objects of EnergyPlus are to make simple tariffs easy to model and complex tariffs possible to model. The inputs used to calculate monthly and annual utility bills and are listed below:

- UtilityCost:Tariff: sets the parameters for the overall tariff such as associated output meter, conversion factors, demand computation, monthly charges, and minimum charges, and schedule names for the time of use periods, seasons, billing cycle, and real-time pricing.
• UtilityCost:Qualify: provides a method to quantify if a tariff applies to the building being modeled since utilities provide limits usually based on peak demand or energy consumption for each tariff.

• UtilityCost:Charge:Simple: defines cost per unit value for a charge based on energy or demand or other charges for a specific season.

• UtilityCost:Charge:Block: defines a more complicated but common type of charge for energy or demand where the costs are defined in the tariff with blocks of charges (i.e., the first 1000 kWh is one cost per kWh, the next 5000 kWh is a second cost, and remaining kWh is a third cost).

• UtilityCost:Ratchet: some utilities charge not only for the current month’s demand but also compensate for the highest demand in recent months.

This research implemented UtilityCost:Tariff for electricity and gas for simulation of energy cost.

4.5 Step 4: Filtering and Narrowing Down the Results by Implementing Python Script, Genetic Algorithm (GA) and Machine Learning Methods

The optimization method used in this study is a combination of GA and Machine Learning. The GA, in combination with flood fill algorithms and Artificial Neural Network (ANN), create a new technique to identify the relations between the outputs, to assign weights and dynamically adjust the target position.

The GA is a method for solving both constrained and unconstrained optimization problems, based on natural selection and bio-inspired operators such as
mutation, crossover and selection. The GA repeatedly modifies a population of individual solutions. The GA contains a genetic representation of the solution domain. In this research, the solution domain includes all design scenarios and a fitness function to evaluate defined ranges and standards for high-performance facade.

Reinforcement Learning is an area of Machine Learning that focuses on what to do and how to map situations to actions. The end result is to maximize the numerical reward signal. The learner is not told which action to take but instead must discover which action will yield the maximum reward. Reinforcement Learning belongs to a bigger class of machine learning algorithms. Because of reducing process time, it can enhance the optimization process. In this research, it narrows down the scenarios to optimized results by sending scenarios to simulator and evaluating the outputs.

The optimization procedure was implemented by coupling Python and EnergyPlus, and by applying a reinforcement learning and genetic algorithm. Python sees EnergyPlus as a generator of hidden functions and this implies the use of heuristic and iterative optimization algorithms, such as the GAs and MLs. This allows the evaluation of feasible energy efficiency measures applied to buildings. Then, the energy cost of the building is calculated to identify the cost-optimal solution. The optimization engine is responsible for searching for design solutions that are as close to the objective function and as diversified as possible.

This phase is significant because this concerns the boundaries between building science and mathematical optimization, by requiring a satisfactory expertise in both the fields. Since the optimization algorithm is implemented in Python and the evaluation of the objectives needs the use of EnergyPlus engine, a communication between these two
programs is required. Therefore, coupling the simulation program and optimization algorithm is important in this phase. A coupling function discussed in previous step is written in Python environment, in order to convert decision variables into an EnergyPlus input file (.idf) and also for converting an output file of EnergyPlus (.csv) into the objectives. In this way, the communication is achieved, and the optimization problem can be solved. Next sections will explain more about the GA and ML techniques implemented in phase two of the optimization process in this research. Evaluation and validation of the final results are discussed in the next chapter.

4.5.1 Optimization Algorithm - Genetic Algorithms

There are many optimization algorithms and mainly fall into two categories: iterative methods (mostly gradient-base) and heuristics (mainly direct-search methods and stochastic). Besides, there are hybrid methods that combine two or more techniques. Each of these groups and algorithms contain different methods for constrained, unconstrained or single and multi-objective problems.

Based on the problem that we are trying to solve in this research, evolutionary algorithms were selected to be implemented because of their robustness. Between multi-objectives genetic algorithms (MOGAs), NSGA II is selected which is a non-dominated sorting genetic algorithm and used to find optimal solutions for this research objectives. NSGA-II is a fast and multi-objective method that provides a good tradeoff between well converged and distributed solutions. This algorithm uses a specific population sorting based first on dominance, and then on crowding distance computed for each individual scenario. This selection process makes both convergence and spreading of the solution more ensured, without requiring the use of external population. This is the
base algorithm, which is coupled with other machine learning techniques to improve the framework performance. The parameters of this algorithms are explained below:

- Population size (the number of scenarios to be evaluated in each iteration): More design variables require larger population size. The number of processor cores available for running the simulations also influence the population size.
- Max generations (the number of iterations we want the optimization to run): A large enough number of design scenarios can be assigned, and the process can be terminated manually if enough solutions have been found.
- Crossover rate (when the new solutions are created by merging features of existing solutions): A high value (towards 100%, or 1.0) is desirable.
- Mutation rate (how often random changes happen to the new solutions): A lower value is preferred; otherwise the algorithm may behave like a random trial and error.
- Tournament selection size (new solutions in algorithm are created from selected existing solutions): The selection process is stochastic but influenced by the 'fitness' of the existing solutions. Tournament selection picks two (or more) solutions randomly from the existing population, and keeps the better (best) one based on its fitness value. The larger the tournament size is, the harder the algorithm is pushing towards the desired objectives. A tournament size of 2 is normally a good balance.
In building optimization studies, using GA generally reduces the computational time by two methods. The first method uses very simplified model instead of a complete simulation. However, this simplification can cause inaccurate modeling of a building. The second method is based on selecting a very small size for GA population or relatively small number of generations. But one efficient solution is to reduce the computational time associated with GA algorithm by employing machine learning techniques to reduce time and increase the accuracy of the results. The machine learning method used in this research is a combination of Batch normalization, which is an Artificial Neural Network (ANN) technique, and flood fill algorithm. ANNs are effective methods that imitate the complex relationship of the network to solve multi-objective and non-linear problems. ANNs resemble the biological neural system, composed of layers of parallel neurons and weighted links. They learn the relationship between the input and output variables by studying previously recorded data.

The neurons are connected by a large number of weighted links, over which signals or information can pass. Basically, a neuron receives inputs over its incoming connections, combines the inputs, performs generally a non-linear operation, and then outputs the final results. The most known, simple and used network arrangement is the feedforward model. In this model, the neurons are placed in several layers. The first one is the input layer, which receives inputs from outside. The last layer, called output layer, supplies the result evaluated by the network. Between these two layers, a network can have none, one or more intermediate layers called hidden layers. ANN diagram is illustrated in Figure 37, and shows these following components:
• **Input layer:** First layer in neural network takes inputs (values) and passes them to the next layer. There is no weight or bias associated in this level. In our network our input values are shown in Table 17.

• **Hidden Layer:** Hidden layers have neurons (nodes) that apply different transformations to the input data. Hidden layers can be collection of stacked neurons that are connected to each other, so they are fully connected hidden layers.

• **Output layer:** This is the last layer in the network where we can get desired number of values or ranges.

![Artificial Neural Network diagram](image.png)

Figure 37: Artificial Neural Network diagram.
The normalization process in this study was conducted on input and output data. The normalization process minimizes the influence of number of scenarios and variations as input variables and trains the outputs to find optimum parameters and scenarios. ANN implements weight and bias to find the optimums within a divided subset of data. This study used 75% of data for normalization and training, and 25% of data (that was not considered in training) was used for validation and testing process. Then activation function is utilized to scale the outputs and optimize the ANN. Activation function decides whether a scenario should be selected or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a scenario.

After defining the activation function, we need to apply appropriate algorithms to train the ANN. In this framework, performance of ANN is evaluated by calculation of the mean for selected outputs. These values are dynamically calculated as average of minimum and maximum for selected scenarios and are unique for this framework. A computer script (Python) is developed for adjusting and scaling the weight and bias dynamically for the indicators.

GA was implemented to find better generation of scenarios among all scenario variation. This is a different technique from implementing GA into the optimization phase. In regular GA, implementation attempt is to generate better solution in each step of generation by predefined fitness function. But in this study the goal is select and find better solutions among all variations, and the fitness of each generated batch is calculated by recalling the trained ANN of the first optimization step.
Therefore, the optimization method in this study is a combination of GA and ANN. The GA in combination with flood fill algorithm and batch normalization creates a new technique to find a relation between the outputs, to assign weights and dynamically adjust the target position to find optimal scenarios. For this framework, three indices were defined for energy consumption, occupants’ comfort and costs as indicators. Indicators are combined values that are used to measure performance, achievement or the impact of changes. The values that were used in this study to define each indicator are shown Table 17 and Figure 38, representing a sample for scoring total EUI electricity indicator.

Table 17: Indicators, ANN input and weight values.

<table>
<thead>
<tr>
<th>Indicators</th>
<th>Output (Input Layer for ANN)</th>
<th>Weight Values</th>
<th>Total Indicator Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Energy Consumption</strong></td>
<td>EUI-Electricity for Int. Lighting</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>EUI-Electricity for Cooling</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EUI-Gas for Heating</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td><strong>Occupant Comfort</strong></td>
<td>PMV &amp; PPD</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>Daylight-illuminance</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Daylight-Glare index</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td><strong>Energy Cost</strong></td>
<td>Total Electricity Cost (Int. lighting &amp; Cooling)</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Total Gas Cost (Heating)</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

Batch normalization technique is used for the first phase of optimization. This is a technique for improving speed, performance and stability of ANN by adjusting and scaling the activations. The batch normalization was introduced in 2015 by Loffè. The intention behind batch normalization is to optimize network training. Several benefits of this methods include faster training, higher learning rates, reduced sensitivity, easier
methods to initialize and produce better results. This technique, combined with flood field algorithm, facilitates the optimization by sorting the highest indicators and decides which scenarios must be simulated.

The flood fill algorithm takes three parameters: start node, target and replacement, and determines the area connected to our target. This algorithm facilitates the optimization by sorting the highest indicators and decides which scenarios must be simulated, based on the specific scenario ID. Using this algorithm decreases the process time, because it is not necessary to simulate all scenarios—rather, only scenarios that are closer to the target proceed to simulation. The comparison is based on the assigned indicator value. In dynamic system, it is necessary to scale indicators to represent the impact of the indicators, and to configure following tasks and converge the results to the goal based on these scores. Figure 38 shows a sample for scoring total EUI electricity, EUI gas, PMV and energy cost indicators. These indicators work as fitness functions in genetic algorithms. These indicators or fitness functions must correlate closely to the goals and must be computed quickly because they need to be iterated many times in order to produce usable results.
The initial population is generated randomly, based on the range of possible design scenarios. It is sent to the simulator to run the initial calculations, and then results are returned to the database to compare with the goals and standards. Then, design scenarios that have results closer to the goals are kept, and others are removed. In this framework, the goal is a summation of three indicators, for energy consumption, comfort and cost. The indicators are dynamically updated based on the range of results. The minimum and maximum of each of these indicators are shown in Table 18.
values of this table are calculated and fed into the Python script to automatically weight the results and score the indicators for the optimization phase. The weight bias is part of ANN that creates a hidden layer to adjust and scale the outputs and results. The fitness of each generated output is calculated by recalling the trained ANN in the first step of optimization. Each of the outputs is defined by fitness values of indicators.

Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. This method accelerates the simulation process and the results give us pattern or clusters of optimized scenarios for analysis in next phase of optimization. The whole optimization process is illustrated in Figure 39.

Figures 40 and 41 show part of decision-making scripts that implement the flood fill and ANN for optimization phases.
Figure 39: Optimization flow chart, illustrating the application of GA and ANN.
Figure 40: Scripts used to calculate total points and decide whether to keep or skip the scenarios.
```python
# getSkipVal(skipPorom, id):

cur.execute('SELECT * from allSec where isDone = '1' and ID = '' + str(id) + '')
res = cur.fetchone()

print res
if skipPorom == 'consumption': # Glazing, insulation int
glz = res['Glazing_System']
im1 = res['Insulation']
fx = cur.execute(update allSec set isDone = '2' where isDone = 'B' and (Glazing_System = '' + glz + ' ' and Insulation = '' + im1 + ' '))
db.commit();
cur.execute('SELECT ID from allSec where isDone = 'B' AND (Glazing_System != '' + glz + ' ' or Insulation != '' + im1 + ' ') ORDER BY 'ID' ASC limit 1')
res = cur.fetchone()
res = res['ID']
cur.execute('UPDATE allSec set isDone = '2' where ID < ' + str(res) + ' and isDone = 'B''
db.commit();
return res

if skipPorom == 'comfort': # MNR, gazing system,
MNR = res['MNR']
glz = res['Glazing_System']
fx = cur.execute(update allSec set isDone = '2' where isDone = 'B' and (MNR != '' + MNR + ' ' and Glazing_System = '' + glz + ' '))
db.commit();
cur.execute('SELECT ID FROM allSec where isDone = 'B' AND (MNR != '' + MNR + ' ' or Glazing_System != '' + glz + ' ')) ORDER BY 'ID' ASC limit 1')
res = cur.fetchone()
res = res['ID']
cur.execute('UPDATE allSec set isDone = '2' where ID < ' + str(res) + ' and isDone = 'B''
db.commit();
return res

if skipPorom == 'cost' # Insulation, cladding, glazing system
ins1 = res['Insulation']
wllC = res['Wll_Cladding']
fx = cur.execute(update allSec set isDone = '2' where isDone = 'B' and (Insulation = '' + ins1 + ' ' and Wll_Cladding = '' + wllC + ' '))
db.commit();
cur.execute('SELECT ID FROM allSec where isDone = 'B' AND (Insulation != '' + ins1 + ' ' or Wll_Cladding != '' + wllC + ' ') ORDER BY 'ID' ASC limit 1')
res = cur.fetchone()
res = res['ID']
cur.execute('UPDATE allSec set isDone = '2' where ID < ' + str(res) + ' and isDone = 'B''
db.commit();
return res

#-------------------------------

def valConsumption(runID):
cur.execute('SELECT ID, EUI_Elec_IntLight, EUI_Elec_Cooling, EUI_Gas_Total FROM `output` where runID = '' + str(runID) + '')
res = cur.fetchone()

print res, runID
point = 0
for scenario in res:
    x = float(scenario['EUI_Elec_IntLight'])
    minMax = cur.execute('SELECT min(EUI_Elec_IntLight), max(EUI_Elec_IntLight) from outputInit WHERE name = ''EUI_Elec_IntLight'''
    minMax = cur.fetchone()
    d = float(minMax['max(EUI_Elec_IntLight)']) - float(minMax['min(EUI_Elec_IntLight)'])
    point = point + (20 - (x - float(minMax['min(EUI_Elec_IntLight)']) / d))
    print(point, minMax, '--------')

    x = float(scenario['EUI_Elec_Cooling'])
    minMax = cur.execute('SELECT min(EUI_Elec_Cooling), max(EUI_Elec_Cooling) from outputInit WHERE name = ''EUI_Elec_Cooling''
    minMax = cur.fetchone()
    d = float(minMax['max(EUI_Elec_Cooling)']) - float(minMax['min(EUI_Elec_Cooling)'])
    point = point + (20 - (x - float(minMax['min(EUI_Elec_Cooling)']) / d))
    print(point, '--------')

    x = float(scenario['EUI_Gas_Total'])
    minMax = cur.execute('SELECT min(EUI_Gas_Total), max(EUI_Gas_Total) from outputInit WHERE name = ''EUI_Gas_HotWater''
    minMax = cur.fetchone()
    d = float(minMax['max(EUI_Gas_Total)']) - float(minMax['min(EUI_Gas_Total)'])
    point = point + (20 - (x - float(minMax['min(EUI_Gas_Total)']) / d))
    print(point, '--------')
updatePoints(scenario['ID'], 'consumption', round(point, 2))
# print 'consumptionInit', scenario['EUI_Elec_Total'], '\"', scenario['EUI_Gas_Total'], '\"', point
point = 0

db.commit();
```

Figure 41: Scripts used to calculate Energy Consumption, comfort and cost, and to assign weight bias.
Table 18: Minimum and maximum value scenarios for indicators to dynamically update the range of results.

<table>
<thead>
<tr>
<th>Output-Indicator</th>
<th>Min Value-Scenario</th>
<th>Max Value-Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EUI Electric-Int. Lighting</strong></td>
<td>60% No shading AL 2x6 metal @16”+ 1” Aerogel Hi VT (single clear-AL, TBAL or wood)</td>
<td>20% shading V-45 or H-45 CM 2x6 wood @24”+ 4” XPS Styrofoam Trip le Low-e Clear (Argon)</td>
</tr>
<tr>
<td><strong>EUI Electric-Cooling</strong></td>
<td>20% shading V-45 OR H-45 CM 2x6 wood @24”+ 4” XPS Styrofoam Trip le Low-e Clear (Argon)/wood fiberglass</td>
<td>60% No Shading AL 2x6 metal @16”+ 1” Aerogel single clear-AL Sgl-Clr-Air-W/F</td>
</tr>
<tr>
<td><strong>EUI Gas-Heating</strong></td>
<td>20% shading V-45 or H-45 AL 2x6 wood @24”+ 4” XPS Styrofoam Triple Clear, High–SHGC low-e (Argon)/Wood, fiberglass (Trip-Clr-HiSHGC-LowE-Argon-W/F)</td>
<td>60% No shading CM 2x6 metal @16”+ 1” Aerogel Single clear, AL or Double Low-e Clear (Air), AL</td>
</tr>
<tr>
<td><strong>PMV-PPD</strong></td>
<td>Range -0.5 and +5 Min -0.55 or +0.55</td>
<td>0-PMV 10-PPD</td>
</tr>
<tr>
<td><strong>Energy Cost- Elec Total</strong></td>
<td>Min Enduse int. lighting +Min Enduse Cooling EnergyCost_Elec_Total</td>
<td>Max Enduse int. lighting +Max Enduse Cooling</td>
</tr>
<tr>
<td></td>
<td>60% No shading AL 2x6 metal @16”+ 1” Aerogel Hi VT (single clear-AL, TBAL or wood) + 20% shading V-45 OR H-45 CM 2x6 wood @24”+ 4” XPS Styrofoam Trip le Low-e Clear (Argon)/wood fiberglass</td>
<td>20% shading V-45 or H-45 CM 2x6 wood @24”+ 4” XPS Styrofoam Trip le Low-e Clear (Argon) + 60% No Shading AL 2x6 metal @16”+ 1” Aerogel single clear-AL</td>
</tr>
<tr>
<td>Energy Cost- Gas Total</td>
<td>20% shading V-45 or H-45 AL 2x6 wood @24”+ 4” XPS Styrofoam Triple Clear, High–SHGC low-e (Argon)/Wood, fiberglass</td>
<td>60% No shading CM 2x6 metal @16”+ 1” Aerogel Single clear, AL or Double Low-e Clear (Air), AL</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Daylight (Daylight reference point 1 illuminance)</td>
<td>20% shading V-45 or H-45 CM 2x6 wood @24”+ 4” XPS Styrofoam Trip le Low-e Clear (Argon)</td>
<td>60% No shading AL 2x6 metal @16”+ 1” Aerogel single clear-AL, TBAL or wood Sgl-Clr-Air-AL 60% No shading AL 2x6 metal @16”+ 1” Aerogel single clear-AL, TBAL or wood</td>
</tr>
<tr>
<td>Daylight (Glare index)</td>
<td>20% shading V-45 or H-45 CM 2x6 wood @24”+ 4” XPS Styrofoam Trip le Low-e Clear (Argon)</td>
<td>60% No shading AL 2x6 metal @16”+ 1” Aerogel single clear-AL, TBAL or wood Sgl-Clr-Air-AL 60% No shading AL 2x6 metal @16”+ 1” Aerogel single clear-AL, TBAL or wood</td>
</tr>
</tbody>
</table>

Figures 42 to 46 are demonstrating the initial efforts to improve the performance of this framework. The final results are discussed in detail the next chapter, but these figures show some preliminary data to illustrate optimization techniques. Figures 42 and 43 show how optimization algorithm selects and sorts the fitted results for this framework. Figure 42 shows the results before applying optimization for 2,061 design scenarios and Figure 43 shows the result of 18,103 scenarios with assigning the first step of optimization. In this case, we have 1,627 scenarios that scored 20 and more than 20 (1,591 scenarios at 20 and 36 more than 20). Using this process decreases the process time, because it is not necessary to simulate all scenarios-rather, only scenarios that are closer to the target. After running all scenarios (26460 ID) with applying batch normalization technique, 3,164 scenarios are selected. Next step focuses on comparison of results.
The next step of optimization is applying dropout technique, then comparing the results. This is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data. Dropout refers to dropping out
units (hidden or visible) in neural network. In machine learning, correlation clustering provides a method for clustering a set of objects into optimum number of clusters based on the similarity. So, in correlation matrix, the relationship between the objects (variables) are known instead of the actual representations of the objects. Figure 44 shows the correlation matrix based on output data, integrated with optimization method to sort the results. This correlation matrix will be automated later when we are collecting data from users, so it will be populated based on those data. Figures 45 and 46 show the final results of all scenarios with both techniques implemented. Figure 45 represent the first phase of optimization, and Figure 46 shows second phase after applying correlation matrix. Results show that process time, performance and accuracy are improved by using this method.
Figure 44: Correlation matrix.

Figure 45: Results for all scenarios with applying batch normalization and flood field algorithm (Phase 1 optimization, flood fill + Batch normalization).
In summary, this framework utilizes GA and ANN optimization techniques by employing custom Python scripts, developed as part of this research to automate simulation and optimization process.
CHAPTER 5
RESULTS

5.1 Results of the Study and Discussion

For final results, different weights and scores for indicators were tested to train the ANN, and then to evaluate and compare the effect of the main objective functions on final results. There are many ways to represent the results. However, in this research three different methods were used to represent the results:

- Line Charts: The line charts show the convergence of each objective and performance criteria with goals. It can be graphed by selecting the minimum, maximum, standard or individual benchmarks and deviation of objectives.

- Scatter Plots: The scatter plot can show all explored and simulated scenarios and selected scenarios with different colors based on different phase of optimization.

- Histograms: The distribution histograms of design variables are shown in these types of charts. These provide give a quick view of how scenarios are clustered and how objective functions are related to design variables.

The developed user interface allows used to visualize results in tabs, using these three different methods. Figure 47 shows part of scripts for preparing the web application that works as a front-end for the research framework.
Different weight bias values for ANN were tested to compare the results. In neural network, each input can be associated with a weight. This means that weight decides how the activation function will trigger the bias and affect the outputs. In this framework, different weights were associated with indicators, some of these weighted energy consumption and energy cost more and other weighted the occupant comfort. Table 19 shows the weights assigned to each output variables for the indicators (objectives), where the total indicator values are 100 and the optimization total points are tested for 70-80 to evaluate and find relations between results and validate the training process.
Table 20 shows different weights associated with ANN in optimization phase for the investigated test cell. Each of these indicators is a combination of related outputs from the EnergyPlus simulation, discussed in Chapter 4. Figures 48 to 54 show histograms for the investigated test cell. These figures illustrate the total scenarios that are sent to simulation engine to calculate the outputs and send back to optimization modules for decision making. It means that based on GA and ANN coupled in optimization module, it is automatically decided if the framework wants to keep or skip the scenario. The different pattern for different weight bias can also be seen in these figures.

Table 19: Indicators, ANN input and weight values.

<table>
<thead>
<tr>
<th>Indicators</th>
<th>Output</th>
<th>Weight Values</th>
<th>Total Indicator Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Energy Consumption</strong></td>
<td>EUI-Electricity for Int. Lighting</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>EUI-Electricity for Cooling</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EUI-Gas for Heating</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PMV &amp; PPD</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td><strong>Occupant Comfort</strong></td>
<td>Daylight-illuminance</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Daylight-Glare index</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td><strong>Energy Cost</strong></td>
<td>Total Electricity Cost (Int. lighting &amp; Cooling)</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Total Gas Cost (Heating)</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

Table 20: The result of different weight bias and number of scenarios selected for each type.

<table>
<thead>
<tr>
<th>Energy Consumption</th>
<th>Comfort</th>
<th>Selected Scenarios</th>
<th>Total Points &gt;=70</th>
<th>Total Points &gt;=73</th>
<th>Total Points &gt;=75</th>
<th>Total Points &gt;=77</th>
<th>Total Points &gt;=79</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>30</td>
<td></td>
<td>1770</td>
<td>1689</td>
<td>1088</td>
<td>673</td>
<td>388</td>
</tr>
<tr>
<td>17</td>
<td>35</td>
<td></td>
<td>1734</td>
<td>1653</td>
<td>1063</td>
<td>659</td>
<td>373</td>
</tr>
<tr>
<td>20</td>
<td>35</td>
<td></td>
<td>1317</td>
<td>1216</td>
<td>779</td>
<td>485</td>
<td>275</td>
</tr>
</tbody>
</table>
Figure 48: Weight Bias (35-20) results for total indicator scores 70 and greater.

Figure 49: Weight Bias (35-25) results for total indicator scores 70 and greater.
Figure 50: Weight Bias (15-30) results for total indicator scores 70 and greater.

Figure 51: Weight Bias (30-30) results for total indicator scores 70 and greater.

Figure 52: Weight Bias (17-35) results for total indicator scores 70 and greater.
Based on the above results, we can observe different patterns emerging for different weights and bias. This is because the ANN is trained differently based on GA algorithms and weight bias.

One of the general observations related to the range of scenarios that are sent to EnergyPlus and are simulated and selected is discussed here. When the energy consumption indicators are increased, the selected and calculated scenarios do not increase beyond 8,000 scenarios. However, when the comfort indicators are increased,
the network is selecting and calculating the wider range of scenarios (nearly all scenarios, which is 24,000).

For further evaluation of results, Figures 55 to 57 illustrate the closer results to optimum and higher total points for indicators. These figures illustrate the results in different steps for better observation and evaluation of the results.

The selected clusters can also be observed in the results in figures below. These clusters are separated by total indicator points, and show that when considering all objectives in this study (energy consumption, occupant comfort and energy cost), there is different level of results clusters with different characteristics. These clusters of selected outputs have some similar characteristics that are listed in Table 21 to 23. Table 24 lists the range of indicators for three top clusters. It illustrates the range of EUI for gas and electricity for cooling, heating and lighting, the range of daylight illuminance, glare index and PMV PPD, and cost indicators (for electricity and gas). All figures clearly show the sorted scenarios, based on the total indicator score. Figure 57 shows the top scored scenarios with associated scenario IDs.
Figure 55: Selected scenarios for weight bias (25-35) that have total indicator scores 75 or greater.

Figure 56: Selected scenarios for weight bias (25-35) that have total indicator scores 77 or greater and illustration of 3 selected output clusters.
Figure 57: Selected scenarios for weight bias (25-35) that have total indicator 79 or greater (top selected scenarios-Cluster 1).

Table 21: The result of optimization showing the characteristic of optimum scenarios (Cluster 1).

<table>
<thead>
<tr>
<th>Design Variables</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window to Wall Ratio</td>
<td>20%</td>
</tr>
<tr>
<td>Glazing System</td>
<td>Trip-Clr-HiSHGC-LowE-Argon-Alum</td>
</tr>
<tr>
<td>Wall Assembly</td>
<td>Wood Framed</td>
</tr>
<tr>
<td>Cladding Material</td>
<td>Terracotta (TR)</td>
</tr>
<tr>
<td></td>
<td>Stone (ST)</td>
</tr>
<tr>
<td></td>
<td>Aluminum (AL)</td>
</tr>
<tr>
<td></td>
<td>Zinc</td>
</tr>
<tr>
<td>Insulation</td>
<td>2x6 @24” +4” XPS Styrofoam</td>
</tr>
<tr>
<td></td>
<td>2x6 @16” +4” XPS Styrofoam</td>
</tr>
<tr>
<td>Solar Control</td>
<td>No Shading</td>
</tr>
</tbody>
</table>
Table 22: The result of optimization showing the characteristic of optimum scenarios (Cluster 2).

<table>
<thead>
<tr>
<th>Design Variables</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window to Wall Ratio</td>
<td>20%</td>
</tr>
<tr>
<td>Glazing System</td>
<td>Trip-Clr-HiSHGC-LowE-Argon-Alum</td>
</tr>
<tr>
<td>Wall Assembly</td>
<td>Wood Framed</td>
</tr>
<tr>
<td>Cladding Material</td>
<td>Terracotta (TR)</td>
</tr>
<tr>
<td></td>
<td>Stone (ST)</td>
</tr>
<tr>
<td>Insulation</td>
<td>2x6 @24” +4” XPS Styrofoam</td>
</tr>
<tr>
<td></td>
<td>2x6 @24” +3” XPS Styrofoam</td>
</tr>
<tr>
<td></td>
<td>2x6 @16” +4” XPS Styrofoam</td>
</tr>
<tr>
<td></td>
<td>2x6 @16” +3” XPS Styrofoam</td>
</tr>
<tr>
<td>Solar Control</td>
<td>Alum-H-45</td>
</tr>
</tbody>
</table>

Table 23: The result of optimization showing the characteristic of optimum scenarios (Cluster 3).

<table>
<thead>
<tr>
<th>Design Variables</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window to Wall Ratio</td>
<td>20%</td>
</tr>
<tr>
<td>Glazing System</td>
<td>Dbl-Clr-Air-AlumTB</td>
</tr>
<tr>
<td>Wall Assembly</td>
<td>Wood Framed</td>
</tr>
<tr>
<td>Cladding Material</td>
<td>Terracotta (TR)</td>
</tr>
<tr>
<td></td>
<td>Stone (ST)</td>
</tr>
<tr>
<td></td>
<td>Zinc</td>
</tr>
<tr>
<td>Insulation</td>
<td>2x6 @24” +4” XPS Styrofoam</td>
</tr>
<tr>
<td></td>
<td>2x6 @16” +4” XPS Styrofoam</td>
</tr>
</tbody>
</table>
Table 24: Range of 3 cluster outputs results for weight bias 25-35.

<table>
<thead>
<tr>
<th>Outputs (25-35)</th>
<th>Cluster 1 Range</th>
<th>Cluster 2 Range</th>
<th>Cluster 3 Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUI-Elec. For Lighting (MJ/m²)</td>
<td>43.83</td>
<td>65.07-66.21</td>
<td>41.23-64.71</td>
</tr>
<tr>
<td>EUI-Elec. For Cooling (MJ/m²)</td>
<td>64.38-64.67</td>
<td>54.32-56.06</td>
<td>55.14-68.04</td>
</tr>
<tr>
<td>EUI-Gas. For Heating (MJ/m²)</td>
<td>2.5-2.79</td>
<td>2.6-3.2</td>
<td>3.57-4.06</td>
</tr>
<tr>
<td>PMV &amp; PPD (%)</td>
<td>0.362-0.372</td>
<td>0.228-0.276</td>
<td>0.210-0.361</td>
</tr>
<tr>
<td>Daylight illuminance-Annual Sum or Average (Lux)</td>
<td>300.87</td>
<td>70.17-79.35</td>
<td>82.2-350.26</td>
</tr>
<tr>
<td>Daylight Glare index</td>
<td>14.81</td>
<td>3.39-3.76</td>
<td>3.89-15.31</td>
</tr>
<tr>
<td>Total Elec. Cost (Annual -$)</td>
<td>1378.63-1379.72</td>
<td>1398.34-1404.32</td>
<td>1392.11-1399.51</td>
</tr>
<tr>
<td>Total Gas Cost (Annual -$)</td>
<td>4.03-4.44</td>
<td>4.19-5.16</td>
<td>5.75-6.54</td>
</tr>
</tbody>
</table>

As the tables show, the optimum design scenario that yields the optimum low energy consumption is the one that incorporates triple-glazed windows with aluminum frame on south elevation, with 20% WWR and wood framed wall assembly.

Interestingly, this optimum design scenario does not contain the highest R-value. This is in line with diminishing return law, where increasing the R-value above a certain optimum point does not have a major impact on improving the building performance.

The window to wall ratio for all optimum results are 20% and 2x6 walls, while other parameters changed in other clusters. This observation verified that the impact of WWR and wall assembly compare to other parameters in facade design.
The effect of wall insulation is obvious in optimum results, and played an essential role in reducing the energy consumption. Similarly, high performance glazing (particularly, triple glazing with low-e coating) was present in most of the selected scenarios.

For the test cell case (Atlanta weather), lower WWRs had better energy performance, as savings from the use of shading and high-performance glazing could not match the increase in heating load due to the high WWRs.

Regarding the framing wall assemblies, structural framing would diminish the effectiveness of the thermal barrier. Whether steel or wood structural elements are used, there will be an impact on thermal performance. All building materials conduct heat at different rates. There is not a threshold beyond which a material becomes categorized as "insulation". However, both wood and steel framing will conduct heat more than cavity insulation products. Steel framing conducts more heat than wood, because it has higher thermal conductivity. Steel framing members also create significant thermal bridging problems if continuous exterior insulation is not used in the facade assembly. Wood framing also induces thermal bridging, but it is not as problematic as metal studs. Although less conductive than steel, wood will still diminish the effective R-value of batt insulation within the framing cavity. Therefore, another observation regarding the optimum selected scenarios indicates that even without the continuous insulation, the wood wall surpasses the efficiency of the steel wall framing with continuous insulation.

In conclusion, it became apparent that the use of multi-objective optimization for complicated design problems can aid at reaching performative solution and clarify
the relation between the design variables and objectives. The effect of the different variables on each objective can be separately and thoroughly analyzed, while trade-offs.

![Diagram](image_url)

Figure 58: Weight Bias (30-30) results for total indicator scores 77 and greater.

Results also indicate that when we have greater occupant comfort weight and bias, there are three other clusters for all scenarios. However, the total indicator scores are lower than for the first cluster, but it is important to analyze and evaluate these selected scenarios for further development of this study. Figure 58 illustrates clusters B and C which are clusters in range of 40% and 60% window to wall ratio. Tables 25 and 26 shows the characteristics of clusters B and C.

Figure 59 to 61 represent the outputs of cluster A and Table 27, 28 and 29 illustrate the characteristics of these clusters. Table 30 lists the range of indicators for three top clusters.
Table 25: The result of optimization and the characteristic of optimum scenarios (cluster B).

<table>
<thead>
<tr>
<th>Design Variables</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window to Wall Ratio</td>
<td>40%</td>
</tr>
<tr>
<td>Glazing System</td>
<td>Trip-Clr-HiSHGC-LowE-Argon-Alum</td>
</tr>
<tr>
<td>Wall Assembly</td>
<td>Wood Framed</td>
</tr>
<tr>
<td>Cladding Material</td>
<td>Terracotta (TR)</td>
</tr>
<tr>
<td></td>
<td>Stone (ST)</td>
</tr>
<tr>
<td></td>
<td>Aluminum (AL)</td>
</tr>
<tr>
<td>Insulation</td>
<td>2x6 @24” +4” XPS Styrofoam</td>
</tr>
<tr>
<td></td>
<td>2x6 @16” +4” XPS Styrofoam</td>
</tr>
<tr>
<td>Solar Control</td>
<td>No Shading</td>
</tr>
</tbody>
</table>

Table 26: The result of optimization and the characteristic of optimum scenarios (cluster C).

<table>
<thead>
<tr>
<th>Design Variables</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window to Wall Ratio</td>
<td>60%</td>
</tr>
<tr>
<td>Glazing System</td>
<td>Dbl-Clr-Air-AlumTB</td>
</tr>
<tr>
<td>Wall Assembly</td>
<td>Wood Framed</td>
</tr>
<tr>
<td>Cladding Material</td>
<td>Terracotta (TR)</td>
</tr>
<tr>
<td>Insulation</td>
<td>2x6 @24” +4” XPS Styrofoam</td>
</tr>
<tr>
<td></td>
<td>2x6 @16” +4” XPS Styrofoam</td>
</tr>
<tr>
<td>Solar Control</td>
<td>Alum-H-45</td>
</tr>
</tbody>
</table>
Figure 59: Selected scenarios for weight bias (30-30) that have total indicator scores 75 or greater.

Figure 60: Selected scenarios for weight bias (30-30) that have total indicator scores 77 or greater and illustration of 3 selected output clusters.
Figure 61: Selected scenarios for weight bias (30-30) that have total indicator scores 79 or greater (top selected scenarios-Cluster 1).

Table 27: The result of optimization showing the characteristic of optimum scenarios (cluster 1).

<table>
<thead>
<tr>
<th>Design Variables</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window to Wall Ratio</td>
<td>20%</td>
</tr>
<tr>
<td>Glazing System</td>
<td>Trip-Clr-HiSHGC-LowE-Argon-Alum</td>
</tr>
<tr>
<td>Wall Assembly</td>
<td>Wood Framed</td>
</tr>
<tr>
<td>Cladding Material</td>
<td>CM, SS, Aluminum (AL)</td>
</tr>
<tr>
<td>Insulation</td>
<td>2x6 @24” +4” XPS Styrofoam</td>
</tr>
<tr>
<td>Solar Control</td>
<td>No Shading</td>
</tr>
</tbody>
</table>
Table 28: The result of optimization showing the characteristic of optimum scenarios (cluster 2).

<table>
<thead>
<tr>
<th>Design Variables</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window to Wall Ratio</td>
<td>20%</td>
</tr>
<tr>
<td>Glazing System</td>
<td>Trip-Clr-HiSHGC-LowE-Argon-Alum</td>
</tr>
<tr>
<td>Wall Assembly</td>
<td>Wood Framed</td>
</tr>
<tr>
<td>Cladding Material</td>
<td>SS</td>
</tr>
<tr>
<td></td>
<td>Aluminum (AL)</td>
</tr>
<tr>
<td></td>
<td>Zinc</td>
</tr>
<tr>
<td>Insulation</td>
<td>2x6 @24” +4” XPS Styrofoam</td>
</tr>
<tr>
<td></td>
<td>2x6 @16” +4” XPS Styrofoam</td>
</tr>
<tr>
<td>Solar Control</td>
<td>Alum-H-45</td>
</tr>
</tbody>
</table>

Table 29: The result of optimization showing the characteristic of optimum scenarios (cluster 3).

<table>
<thead>
<tr>
<th>Design Variables</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window to Wall Ratio</td>
<td>20%</td>
</tr>
<tr>
<td>Glazing System</td>
<td>Dbl-Clr-Air-AlumTB</td>
</tr>
<tr>
<td>Wall Assembly</td>
<td>Wood Framed</td>
</tr>
<tr>
<td>Cladding Material</td>
<td>Aluminum (AL)</td>
</tr>
<tr>
<td></td>
<td>Stainless Steel (SS)</td>
</tr>
<tr>
<td></td>
<td>Zinc</td>
</tr>
<tr>
<td>Insulation</td>
<td>2x6 @24” +4” XPS Styrofoam</td>
</tr>
<tr>
<td></td>
<td>2x6 @16” +4” XPS Styrofoam</td>
</tr>
<tr>
<td></td>
<td>2x6 @24” +3” XPS Styrofoam</td>
</tr>
<tr>
<td>Solar Control</td>
<td>Alum-H-45</td>
</tr>
</tbody>
</table>
Table 30: Range of 3 clusters outputs results for weight bias 30-30.

<table>
<thead>
<tr>
<th>Outputs (30-30)</th>
<th>Cluster 1 Range</th>
<th>Cluster 2 Range</th>
<th>Cluster 3 Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUI-Elec. For Lighting (MJ/m²)</td>
<td>43.83</td>
<td>65.03-66.21</td>
<td>64.71</td>
</tr>
<tr>
<td>EUI-Elec. For Cooling (MJ/m²)</td>
<td>64.5-64.7</td>
<td>54.32-54.85</td>
<td>55.17-55.21</td>
</tr>
<tr>
<td>EUI-Gas. For Heating (MJ/m²)</td>
<td>2.53-2.76</td>
<td>3.17-4.91</td>
<td>4.03-4.13</td>
</tr>
<tr>
<td>PMV &amp; PPD (%)</td>
<td>0.364-0.372</td>
<td>0.181-0.231</td>
<td>0.210-0.213</td>
</tr>
<tr>
<td>Daylight illuminance-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Annual Sum or Average (Lux)</td>
<td>300.87</td>
<td>70.17-79.63</td>
<td>82.2</td>
</tr>
<tr>
<td>Daylight Glare index</td>
<td>14.81</td>
<td>3.39-3.76</td>
<td>3.89</td>
</tr>
<tr>
<td>Total Elec. Cost (Annual -$)</td>
<td>1374.9-1375.92</td>
<td>1393.15-1393.73</td>
<td>1392.97-1393.19</td>
</tr>
<tr>
<td>Total Gas Cost (Annual -$)</td>
<td>4.07-4.44</td>
<td>5.1-7.91</td>
<td>6.49-6.63</td>
</tr>
</tbody>
</table>

Validation

Cross validation occurs in the decision-making process when scoring the indicators. Cross-validation or rotation estimation is a model validation technique for assessing a certain data set. It is used in setting for prediction or estimating the accuracy of a predictive model. The purpose of cross-validation is to test the framework ability to predict new data that was not used in estimating it to find problems, like overfitting and selection bias. Therefore, cross-validation combines measures of fitness and indicators to predict outcomes more accurately. This study used 75% of data for normalization, training and learning, while 25% of data that was not considered in training was used for validation and testing process.
In order to check the accuracy of the ANN to predict EUI and PMV, 25 samples were randomly selected. The corresponding EUI and PMV from EnergyPlus were compared to data issued from the ANN. The maximum deviations for EUI and PMV are 1.86% and 0.22%. These low values confirm the accuracy of the ANN.

All results and data saved in the MYSQL database are included as Appendices at the end of this document. Figure 62 shows the screenshot of these tables that can be exported as Excel files.

Figure 62: Example of results and outputs, which can be exported as Excel files.
CHAPTER 6

CONCLUSION & FUTURE WORK

6.1 Conclusion

This research investigated the role of simulations and optimization in design decision-making process for high-performance facade design. A novel performance-based facade design framework was developed, where different performance criteria have been defined for achieving energy efficiency, occupant comfort and cost optimality. The framework was implemented by coupling EnergyPlus as a simulation engine, and custom scripts using Python programming language. Then, a user interface was developed with MVC method as a front-end application to test and collect data.

The dissertation described the components and functionality of this framework in detail, as well as two-step optimization technique. A case study for a test cell was presented, illustrating how the framework is used to test a variety of design possibilities.

In the presented research, various characteristics of building facade design were considered. Quantitative and simulation-based techniques were used to explore combinations of these different variables and characteristics that yield optimum design scenarios, considering energy consumption, occupant comfort and energy cost.

The results of this research address the architecture and construction research community’s need to develop methods and facade design guidelines that reduce the energy consumption and energy cost and increase the occupant comfort impacts simultaneously.

For this purpose, the methodology that is used in this research can be followed to develop design guidelines. More than its results, the significance of this research is in
the methodology that is applied for coupling simulation and optimization phases and find optimum scenarios. For collecting more data and feedback from the professional users, the developed framework and MVP will be released to the public as an open-source platform.

It is important to note that the inputs can be all design variables with regards to their significance in design decision-making and impacts on environmental performance or occupants’ comfort. In real-life situations, however, architects may assign more weight to some design inputs or environmental impacts because of design priorities, or in order to overcome the conflicting effects of inputs or impacts, so future methodologies and tools should follow this research capability to assign weight to factors in their decision-making support process.

A future direction of the research could be to develop a software tool that can help conduct the entire process (simulation and optimization) within one tool. Such tools need to include all design variables and construction inputs beyond what is studied in this research. Different variables, such as building systems and other components of the building, could be incorporated into the decision-making process.

6.2 Benefits of the Developed Data-Driven Framework

The basic characteristics that differentiate the developed framework and improve decision-making process can be summarized as:

- Automation and Speed: The framework enables users to automatically send the design scenarios to a simulator and gather the outputs, and then screen out and sort these outputs to find optimized results. The advantages of this automated
process are efficient testing methodology, consistency, reliability and increase in the number of possible design scenarios. Also, by implementing this framework, simulation time will be decreased for thousands of design scenarios.

- Variety of variables (multi-objective variables): This framework enables users to test multiple variables at the same time during the design process.

- Modularity: The framework is designed in multiple modules, which work independently. The key benefits of modularity in this framework are distinct functionality and manageability. Each module provides a distinct function and can be combined to provide an entirely new collective function. The separate modules make it easier to test and implement this framework in design process or detect the errors.

- Usability: The ease of use and learnability of this framework for new users is other important benefit of this framework by adding the front end to this framework.

### 6.3 Future Work

The PerforPy framework was developed based on open-source applications. It is available to the public and can be improved by users or developers. Based on the completed research, the following items were identified as potential improvements and would benefit further development:

1. Expand the variables and develop the user interface: Future work for this research could focus on application for various facade designs in different climates and developing the user interface. In addition to developing the user
interface and web application, the aim is to develop a method where any input IDF files can be used, and to enable users to choose their variables for optimization. The rest of the process will be automated, and results will be represented.

2. AI and Deep Learning implementation: Other important development for this research, after releasing the open source application, is collecting the data to implement deep learning techniques utilizing outputs from different iterations to enable the correlation matrix to generate automatically.

3. Expand the scope of this study: Future research can also expand the scope of this study and examine more design characteristics or other building components and systems that can impact the energy consumption and occupant comfort in the building.

4. Developing the Life Cycle Analysis: Significant research has been conducted in this dissertation on the impact of building facade design on energy performance of buildings and occupants’ comfort. However, more research is needed to address Life Cycle Analysis (LCA) assessments. Future research could use other simulation tools and combine LCA software programs, such as SimaPro, GaBi and openLCA, to cross-examine the results of the present research in the context of other geographical locations and environmental data. Indeed, different locations have different climates, which would affect energy performance of buildings, energy cost and occupants’ comfort. Different locations also incorporate different manufacturing practices, electricity generation and distribution,
transportation and logistics systems, and consuming patterns, which all would affect the environmental life-cycle impacts of buildings.

5. **Interface update:** As it is mentioned before, there is a need for improving PerforPy interface and adding new features to better serve users’ needs in the process of performance optimization. Future research should also focus on development of user-friendly software and tools that can work as plug-ins for modeling simulation tools to optimize energy, comfort and life cycle and environmental impacts of design.

6. **Visualizing the optimization results:** Future work should also aim to improve visualization of results within the platform to help users navigate through the results quickly and make design decisions in a single design platform.

7. **Exploring more real projects and test cases:** Future goal is also to test the usefulness of the framework with larger projects and wider ranges of variables within design studio classes and in the industry.
APPENDIX A

SCRIPTS

(These are samples to illustrate the scripts)
#!/usr/bin/env python
import itertools
import MySQLdb

WWR = ['20%', '40%', '60%']
Wall_Assembly = ['Steel-Framed', 'Wood-Framed']
Wall_Cladding = ['AL', 'Zinc', 'SS', 'TR', 'BR', 'ST', 'CM']

Insulation = ['2x6 wood @16 +3 XPS Styrofoam',
              '2x6 wood @16 +4 XPS Styrofoam',
              '2x6 wood @16 +1 Aerogel',
              '2x6 wood @24 +3 XPS Styrofoam',
              '2x6 wood @24 +4 XPS Styrofoam',
              '2x6 wood @24 +1 Aerogel',
              '2x6 metal @16 +3 XPS Styrofoam',
              '2x6 metal @16 +4 XPS Styrofoam',
              '2x6 metal @16 +1 Aerogel',
              '2x6 metal @24 +3 XPS Styrofoam',
              '2x6 metal @24 +4 XPS Styrofoam',
              '2x6 metal @24 +1 Aerogel']

Glazing_System = ['Sgl-Clr-Alum',
                    'Sgl-Clr-AlumTB',
                    'Sgl-Clr-W/F',
                    'Db1-Clr-Alum',
                    'Db1-Clr-AlumTB',
                    'Db1-Clr-AlumW/F',
                    'Db1-Clr-LowE-Alum',
                    'Db1-Clr-LowE-AlumTB',
                    'Db1-Clr-LowE-AlumW/F',
                    'Db1-Clr-LowE-Argon-Alum',
                    'Db1-Clr-LowE-Argon-AlumTB',
                    'Db1-Clr-LowE-Argon-AlumW/F',
                    'Trip-LowE-Argon-Alum',
                    'Trip-LowE-Argon-AlumTB',
                    'Trip-LowE-Argon-AlumW/F',
                    'Db1-Clr-HiSHGC-LowE-Alum',
                    'Db1-Clr-HiSHGC-LowE-AlumTB',
                    'Db1-Clr-HiSHGC-LowE-AlumW/F',
                    'Trip-C1r-HiSHGC-LowE-Argon-Alum',
                    'Trip-C1r-HiSHGC-LowE-Argon-AlumTB',
                    'Trip-C1r-HiSHGC-LowE-Argon-AlumW/F']

Solar_Control = ['No Shading',
                 'Alum-H-45',
                 'Alum-H-90',
                 'Alum-V-45',
                 'Alum-V-90']
```
db = MySQLdb.connect(host="127.0.0.1",
user="mahsa",
passwd="1234r",
db="mahsa")

cur = db.cursor()

def generateAll():
    res = [','.join(x) for x in itertools.product(WWR, Wall_Cladding, Wall_Assembly, Insulation, Glazing_System, Solar_Control)]
    print (len(res), '" scenario generated !!! :D")
    return res

#------------------------------------------

def saveAll(res):
    cur.execute("TRUNCATE TABLE `allSce`")
    db.commit()
    for x in res:
        scenario = x.split(',
        ")")
        print(scenario)
        db.commit();

#------------------------------------------

def clearDatabase():
    # res = cur.execute("delete FROM `allSce` WHERE WWR = '80%' and (Wall_Assembly = 'Wood-Framed' or Wall_Assembly = 'ICF')")
    res = cur.execute("delete FROM `allSce` WHERE Insulation LIKE '%wood%' and (Wall_Assembly = 'Steel-Framed')")
    print (res, 'Rows deleted')
    res = cur.execute("delete FROM `allSce` WHERE Insulation LIKE '%Metal%' and (Wall_Assembly = 'Wood-Framed')")
    print (res, 'Rows deleted')
    db.commit();

#------------------------------------------

res = generateAll()
saveAll(res)
clearDatabase()
```

#!/usr/bin/env python
import MySQLdb
import os

def initMaxMin():
    for x in range(16):
        r = os.system("python bil2Init.py " + str(x))
        print r

    r = os.system("python bil2Init.py " + str(16))
# initMaxMin()
#!/usr/bin/env python

import sys
from eppy.modeleditor import IDF
from shutil import copyfile
import os
import MySQLdb
from eppy.results import readhtml
import pprint

iddfile = ""
idfname = ""
epwfile = ""
idf = ""
db = ""
cur = ""

def initEPP():
    global iddfile
    global idfname
    global epwfile
    global idf

    # iddfile = "/usr/local/EnergyPlus-8-5-0/Energy+.idd"
    iddfile = "/usr/local/EnergyPlus-8-5-0/PreProcess/IDFVersionUpdater/V8-5-0-Energy+.idd"
    IDF.setiddname(iddfile)

    idfname = "TestCell.idf"
    epwfile = "USA_GA_Atlanta.722190_TMY2.epw"
    idf = IDF(idfname, epwfile)

#--------------------------------------------------

def init():
    global db
    global cur

    # db = MySQLdb.connect(host="127.0.0.1",
    # user="mahsa",
    # passwd="mahsadbpass001",
    # db="mahsa")

    db = MySQLdb.connect(host="127.0.0.1",
    user="mahsa",
    passwd="1234r",
    db="mahsa")

    # cur = db.cursor(MySQLdb.cursors.DictCursor)
cur = db.cursor()

#--------------------------------------------------

def checkBuilding():
    building = idf.idfobjects['Material'].upper()[0]

    # for fieldname in building.fieldnames:
        # print "%s = %s" % (fieldname, building[fieldname])
    print (building['Thickness'])

#--------------------------------------------------

def copy():
    copyfile("TestCell-10.idf", "TestCell.idf")
    #
copyfile("/run/user/1000/gvfs/smb-share:server=192.168.1.201,share=auto/Mahsa/TestCell-5.idf", "TestCell.idf")

# delFile() 
def delFile():
    os.remove("TestCell.idf")

# updateData(alldata):
def updateData(alldata):
    print(alldata)

    # Window
    window = idf.idfobjects['window'.upper()][0]

    # print window['Length']
    # print window['Height']

    length = 9
    height = {'20%': 0.6, '40%': 1.2, '60%': 1.8, '80%': 2.45, '90%': 2.75}
    print(window)

    window['Length'] = length
    print(alldata)
    window['Height'] = height[alldata[1]]

    # Material
    Construction = idf.idfobjects['Construction'.upper()][9]
    Construction['Outside_Layer'] = alldata[2]
    Construction['Layer_3'] = 'Conc 30mm'
    print(Construction)

    # Glazing System
    GlazingSystem = {
        'Sgl-Clr-Alum': '1A',
        'Sgl-Clr-AlumTB': '1B',
        'Sgl-Clr-W/F': '1C',
        'Db1-Clr-Alum': '2A',
        'Db1-Clr-AlumTB': '2B',
        'Db1-Clr-Alum-W/F': '2C',
        'Db1-Clr-LowE-Air-Alum': '3A',
        'Db1-Clr-LowE-Air-AlumTB': '3B',
        'Db1-Clr-LowE-Air-W/F': '3C',
        'Db1-Clr-LowE-Argon-Alum': '4A',
        'Db1-Clr-LowE-Argon-AlumTB': '4B',
        'Db1-Clr-LowE-Argon-W/F': '4C',
        'Trip-LowE-Argon-Alum': '5A',
        'Trip-LowE-Argon-AlumTB': '5B',
        'Trip-LowE-Argon-W/F': '5C',
        'Db1-Clr-HiSHGC-LowE-Air-Alum': '6A',
        'Db1-Clr-HiSHGC-LowE-Air-AlumTB': '6B',
        'Db1-Clr-HiSHGC-LowE-Air-W/F': '6C',
        'Trip-Clr-HiSHGC-LowE-Argon-Alum': '7A',
        'Trip-Clr-HiSHGC-LowE-Argon-AlumTB': '7B',
        'Trip-Clr-HiSHGC-LowE-Argon-W/F': '7C'
    }
windowMaterial['Outside_Layer'] = GlazingSystem[alldata[5]]

windowMaterial = idf.idfobjects['Construction'.upper()][10]
windowMaterial['Layer_2'] = GlazingSystem[alldata[5]]

print(windowMaterial)

#--------------------------------------------- Solar Control
if (alldata[6] != 'No Shading'):
    windowMaterial = idf.idfobjects['Construction'.upper()][10]

    solar = {
        'Alum-V-45' : 'Ext. Blind-V-45',
        'Alum-V-90' : 'Ext. Blind-V-90'
    }

    windowMaterial['Outside_Layer'] = solar[alldata[6]]
    print(windowMaterial)

#--------------------------------------------- Insulation
windowMaterial = idf.idfobjects['Construction'.upper()][9]
Insulation = {
    '2x6 wood @16 +3 XPS Styrofoam' : 'Insulation W16/Batt+3"xps',
    '2x6 wood @16 +4 XPS Styrofoam' : 'Insulation W16/Batt+4"xps',
    '2x6 wood @16 +1 Aerogel' : 'Insulation W16/Batt+1"aerogel',
    '2x6 wood @24 +3 XPS Styrofoam' : 'Insulation W24/Batt+3"xps',
    '2x6 wood @24 +4 XPS Styrofoam' : 'Insulation W24/Batt+4"xps',
    '2x6 wood @24 +1 Aerogel' : 'Insulation W24/Batt+1"aerogel',
    '2x6 metal @16 +3 XPS Styrofoam' : 'Insulation M16/Batt+3"xps',
    '2x6 metal @16 +4 XPS Styrofoam' : 'Insulation M16/Batt+4"xps',
    '2x6 metal @16 +1 Aerogel' : 'Insulation M16/Batt+1"aerogel',
    '2x6 metal @24 +3 XPS Styrofoam' : 'Insulation M24/Batt+3"xps',
    '2x6 metal @24 +4 XPS Styrofoam' : 'Insulation M24/Batt+4"xps',
    '2x6 metal @24 +1 Aerogel' : 'Insulation M24/Batt+1"aerogel'
}

windowMaterial['Layer_2'] = Insulation[alldata[4]]

if (alldata[6] == 'No Shading'):
    windowMaterial = idf.idfobjects['FenestrationSurface:Detailed'.upper()][0]

    windowMaterial['Shading_Control_Name'] = ''

    # idf.save()

#---------------------------------------------
def runEP():
    idf.run()

#---------------------------------------------
def getAllData():
    res = cur.execute("select ID, WWR, Wall_Cladding, Wall_Assembly, Insulation, Glazing_System, Solar_Control from allSce where isDone = 0 and ID mod 200 = 0 and ID > 90000")
    res = cur.fetchall()
for x in res:
    delFile()
    copyf()
    updateData(x)
    # break;
    runEP()
    saveData(x[0])
    # break;

print ("Done")

#------------------------------------------------------------------------------
def saveData (ID):
    fname = "eplustbl.htm"
    filehandle = open(fname, 'r').read()
    htables = readhtml.titletable(filehandle)

    rData = []
    rData.append(readData(htables,-9,1,1))
    rData.append(readData(htables,-9,2,1))
    rData.append(readData(htables,-9,3,1))
    rData.append(readData(htables,-9,1,2))
    rData.append(readData(htables,-9,16,1))

    rData.append(readData(htables,5,2,1))
    rData.append(readData(htables,5,3,1))
    rData.append(readData(htables,5,4,1))
    rData.append(readData(htables,5,17,1))
    rData.append(readData(htables,5,17,2))

    rData.append(readData(htables,12,1,2))
    rData.append(readData(htables,12,1,3))
    rData.append(readData(htables,12,1,4))
    rData.append(readData(htables,13,1,7))
    rData.append(readData(htables,13,1,8))
    rData.append(readData(htables,13,1,9))

    rData.append(readData(htables,18,1,4))
    rData.append(readData(htables,43,1,1))
    rData.append(readData(htables,43,2,1))
    rData.append(readData(htables,43,4,1))
    rData.append(readData(htables,43,9,1))
    rData.append(readData(htables,43,27,1))

    rData.append(readData(htables,44,3,1))
    rData.append(readData(htables,67,1,2))
    rData.append(readData(htables,67,2,2))
    rData.append(readData(htables,67,3,2))
    rData.append(readData(htables,68,1,2))
    rData.append(readData(htables,68,2,2))
    rData.append(readData(htables,68,4,2))
    rData.append(readData(htables,70,1,1))
    rData.append(readData(htables,70,3,1))
    rData.append(readData(htables,70,8,1))

    rData.append(readData(htables,71,4,1))
    rData.append(readData(htables,101,14,1))
    rData.append(readData(htables,101,14,4))
    rData.append(readData(htables,104,14,4))
rData.append(readData(htable[0, 104, 14, 5]))

rData.append(readData(htable[1, 109, 14, 5]))

rData.append(readData(htable[2, 109, 14, 6]))

print(len(rData))
print(len(rData))

def saveToDB(ID, rData):
    #---------------------------------------------------------------
    # def saveToDB(ID, rData):
    #---------------------------------------------------------------
    initTabel = 11 + 1

    firstitem = htable[tableNum + initTabel]  # table Number
    firstitem_table = firstitem[1]  #
    x = firstitem_table[rowNum]  # row
    # print (x[columnNum])
    return x[columnNum]

    #---------------------------------------------------------------
    # def saveToDB(ID, rData):
    #---------------------------------------------------------------
    print rData

    x = 'replace into output (seeID, timeStmp, Enduse_Htng_Elec, Enduse_Cldg_Elec
    ,Enduse_IntLighting_Elec, Enduse_Htng_Nrg_Gs, total_enduse
    ,Demand_Enduse_Htng_Elec, Demand_Enduse_Cldg_Elec, Demand_Enduse_Lghting_Elec
    ,Demand_Enduse_Tot_Elec, Demand_Enduse_Tot_Gs
    ,Envelope_Opqrk_Reflectn ,Envelope_Opqrk_Ufactor_w_film
    ,Envelope_Opqrk_Ufactor_w_ofilm, ExtWin_Glass_Ufactor, ExtWin_Glass_SHGC
    ,ExtWin_Glass_Vt, IntLighting_TotPower, Energymtr_Annual_Elec_Fcl
    ,Energymtr_Annual_Elec_IntEqpt ,Energymtr_Annual_Elec_Cldg
    ,Energymtr_Annual_Elec_IntEqpt ,Energymtr_Annual_Elec_Cldg
    ,Enduse_Total_Gs, Enduse_Total ,EnergyCost_Elec_Total ,EnergyCost_Gas_Total
    ,EUI_Elec_Total ,EUI_Gs_Total ,ZoneElec_Mnthly_Lghts_Avg ,ZoneElec_Mnthly_Eqpt_Avg
    ,OccupntCmfort_Trmal_PPD_Avg, Daylight_Illuminance, Daylight_Glre) VALUES("\n    , + str(ID) + ", now(), \
    " + str(rData[0]) + ",
    " + str(rData[1]) + ",
    " + str(rData[2]) + ",
    " + str(rData[3]) + ",
    " + str(rData[4]) + ",
    " + str(rData[5]) + ",
    " + str(rData[6]) + ",
    " + str(rData[7]) + ",
    " + str(rData[8]) + ",
    " + str(rData[9]) + ",
    " + str(rData[10]) + ",
    " + str(rData[11]) + ",
    " + str(rData[12]) + ",
    " + str(rData[13]) + ",
    " + str(rData[14]) + ",
    " + str(rData[15]) + ",
    " + str(rData[16]) + ",
    " + str(rData[17]) + ",
    " + str(rData[18]) + ",
    " + str(rData[19]) + ",
    " + str(rData[20]) + ",
    " + str(rData[21]) + ",
    " + str(rData[22]) + ",
    " + str(rData[23]) + ",
    " + str(rData[24]) + ",
print x
for z in rData:
    print z
res = cur.execute(x)
res = cur.execute('update allSce set isDone = 1 where ID = '' + str(ID) + '''')

# res = cur.execute('update initMaxMin set resWindowsRM ='' + str(rData[0]) + ''',
resComponentCES ='' + str(rData[1]) + ''',resOccupantCDS =''
# + str(rData[2]) + ''' ,resZoneESM ='' + str(rData[3]) + ''' , resZoneHSM ='' + str(rData[4]) + '''
# , resZoneCSM ='' + str(rData[5]) +\n# '' , resDaylightingRM ='' + str(rData[6]) + ''' where ID ='' + str(ID) + '''')

db.commit();
#----------------------------------------------------------------------------------
def getOne (id):
    res = cur.execute('select ID, WWR, Wall_Cladding, Wall_Assembly, Insulation, Glazing_System,
Solar_Control from\nallSce where ID ='' + str(id) + '''')
    res = cur.fetchall()
    for x in res:
        delFile()
        copyf()
        updateData(x)
        # break;
        runEP()
        saveData(x[0])
        # break;
    print ("Done")
#----------------------------------------------------------------------------------

id = sys.argv[1]
copyf()
initEPPY()
initdb()
# getAllData()
getOne(id)
#!/usr/bin/env python

import sys
from eppy.modeleditor import IDF
import os
import MySQLdb
from eppy.results import readhtml
import pprint

iddfile = ""
idfname = ""
epwfile = ""
idf = ""
db = ""
cur = ""

# iddfile = "/usr/local/EnergyPlus-8-5-0/Energy+.idd"
iddfile = "/usr/local/EnergyPlus-8-5-0/PreProcess/IDFVersionUpdater/V8-5-0-Energy+.idd"
IDF.setiddname(iddfile)
idfname = "TestCell.idf"
epwfile = "USA_GA_Atlanta.722190_TMY2.epw"
idf = IDF(idfname, epwfile)

def initEPP():
    global iddfile
    global idfname
    global epwfile
    global idf

    # iddfile = "/usr/local/EnergyPlus-8-5-0/Energy+.idd"
    iddfile = "/usr/local/EnergyPlus-8-5-0/PreProcess/IDFVersionUpdater/V8-5-0-Energy+.idd"
    IDF.setiddname(iddfile)
    idfname = "TestCell.idf"
    epwfile = "USA_GA_Atlanta.722190_TMY2.epw"
    idf = IDF(idfname, epwfile)

    #--------
    def initDB():
        global db
        global cur

        # db = MySQLdb.connect(host="127.0.0.1",
        # user="mahsa",
        # passwd="mahsadbpass001",
        # db="mahsa")

        db = MySQLdb.connect(host="127.0.0.1",
                             user="mahsa",
                             passwd="1234r",
                             db="mahsa")

        # cur = db.cursor(MySQLdb.cursors.DictCursor)
        cur = db.cursor()

    #--------
    def checkBuilding():
        building = idf.idfobjects["Material"].upper()[0]
        # for fieldname in building.fieldnames:
        #    print "%s = %s" % (fieldname, building[fieldname])
        print (building["Thickness"])

    #--------
    def copyf():
        copyfile("TestCell-10.idf", "TestCell.idf")
    #
copyfile("/run/user/1000/gvfs/smb-share:server=192.168.1.201,share=auto/Mahsa/TestCell-5.idf", "TestCell.idf")

def delFile():
    os.remove("TestCell.idf")

#------------------------------------------ Window
window = idf.idfobjects['window'.upper()][0]

length = 9
height = {'20%': 0.6, '40%': 1.2, '60%': 1.8, '80%': 2.45, '90%': 2.75}
print window

window['Length'] = length
print (alldata)
window['Height'] = height[alldata[1]]

#------------------------------------------ Material
Construction = idf.idfobjects['Construction'.upper()][9]

Construction['Outside_Layer'] = alldata[2]
Construction['Layer_3'] = 'Conc 30mm'

print (Construction)

#------------------------------------------ Glazing_System
GlazingSystem = {
    'Sgl-Clr-Alum' : '1A',
    'Sgl-Clr-AlumTB' : '1B',
    'Sgl-Clr-W/F' : '1C',
    'Dbl-Clr-Air-Alum' : '2A',
    'Dbl-Clr-Air-AlumTB' : '2B',
    'Dbl-Clr-Air-W/F' : '2C',
    'Dbl-Clr-LowE-Air-Alum' : '3A',
    'Dbl-Clr-LowE-Air-AlumTB' : '3B',
    'Dbl-Clr-LowE-Air-W/F' : '3C',
    'Dbl-Clr-LowE-Argon-Alum' : '4A',
    'Dbl-Clr-LowE-Argon-AlumTB' : '4B',
    'Dbl-Clr-LowE-Argon-W/F' : '4C',
    'Trip-LowE-Argon-Alum' : '5A',
    'Trip-LowE-Argon-AlumTB' : '5B',
    'Trip-LowE-Argon-W/F' : '5C',
    'Dbl-Clr-HiSHGC-LowE-Air-Alum' : '6A',
    'Dbl-Clr-HiSHGC-LowE-Air-AlumTB' : '6B',
    'Dbl-Clr-HiSHGC-LowE-Air-W/F' : '6C',
    'Trip-Clr-HiSHGC-LowE-Argon-Alum' : '7A',
    'Trip-Clr-HiSHGC-LowE-Argon-AlumTB' : '7B',
    'Trip-Clr-HiSHGC-LowE-Argon-W/F' : '7C'
}
windowMaterial['Outside_Layer'] = GlazingSystem[alldata[5]]

windowMaterial = idf.idfobjects['Construction'.upper()][10]
windowMaterial['Layer_2'] = GlazingSystem[alldata[5]]

print(windowMaterial)

#----------------------------------- Solar Control
if (alldata[6] != 'No Shading'):
    windowMaterial = idf.idfobjects['Construction'.upper()][10]

    solar = {
        'Alum-V-45': 'Ext. Blind-V-45',
        'Alum-V-90': 'Ext. Blind-V-90'
    }

    windowMaterial['Outside_Layer'] = solar[alldata[6]]
    print(windowMaterial)

#----------------------------------- Insulation
windowMaterial = idf.idfobjects['Construction'.upper()][9]

Insulation = {
    '2x6 wood @16 +3 XPS Styrofoam': 'Insulation W16/Batt+3"xps',
    '2x6 wood @16 +4 XPS Styrofoam': 'Insulation W16/Batt+4"xps',
    '2x6 wood @16 +1 Aerogel': 'Insulation W16/Batt+1"aerogel',
    '2x6 wood @24 +3 XPS Styrofoam': 'Insulation W24/Batt+3"xps',
    '2x6 wood @24 +4 XPS Styrofoam': 'Insulation W24/Batt+4"xps',
    '2x6 wood @24 +1 Aerogel': 'Insulation W24/Batt+1"aerogel',
    '2x6 metal @16 +3 XPS Styrofoam': 'Insulation M16/Batt+3"xps',
    '2x6 metal @16 +4 XPS Styrofoam': 'Insulation M16/Batt+4"xps',
    '2x6 metal @16 +1 Aerogel': 'Insulation M16/Batt+1"aerogel',
    '2x6 metal @24 +3 XPS Styrofoam': 'Insulation M24/Batt+3"xps',
    '2x6 metal @24 +4 XPS Styrofoam': 'Insulation M24/Batt+4"xps',
    '2x6 metal @24 +1 Aerogel': 'Insulation M24/Batt+1"aerogel'
}

windowMaterial['Layer_2'] = Insulation[alldata[4]]

if (alldata[6] == 'No Shading'):
    windowMaterial = idf.idfobjects['FenestrationSurface:Detailed'.upper()][0]

    windowMaterial['Shading_Control_Name'] = ''

    # idf.save()

#-----------------------------------
def runEP():
    idf.run()

#-----------------------------------
def getAllData():
    res = cur.execute("select ID, WWR, Wall_Cladding, Wall_Assembly, Insulation, Glazing_System, Solar_Control from\initMaxMin where isDone = 0 and ID mod 200 = 0 and ID > 90000")
    res = cur.fetchall()
for x in res:
    delFile()
    copyf()
    updateData(x)
    # break;
    runEP()
    saveData(x[0])
    # break;

print("Done")

#---#---
def saveData(ID):
    fname = "eplustbl.htm"
    filehandle = open(fname, 'r').read()
    htables = readhtml.titletable(filehandle)

    rData = []
    rData.append(readData(htables,-9,1,1))
    rData.append(readData(htables,-9,2,1))
    rData.append(readData(htables,-9,3,1))
    rData.append(readData(htables,-9,1,2))
    rData.append(readData(htables,-9,16,1))

    rData.append(readData(htables,5,2,1))
    rData.append(readData(htables,5,3,1))
    rData.append(readData(htables,5,4,1))
    rData.append(readData(htables,5,17,1))
    rData.append(readData(htables,5,17,2))

    rData.append(readData(htables,12,1,2))
    rData.append(readData(htables,12,1,3))
    rData.append(readData(htables,12,1,4))

    rData.append(readData(htables,13,1,7))
    rData.append(readData(htables,13,1,8))
    rData.append(readData(htables,13,1,9))

    rData.append(readData(htables,18,1,4))
    rData.append(readData(htables,43,1,1))
    rData.append(readData(htables,43,2,1))
    rData.append(readData(htables,43,4,1))
    rData.append(readData(htables,43,9,1))
    rData.append(readData(htables,43,27,1))

    rData.append(readData(htables,44,3,1))
    rData.append(readData(htables,67,1,2))
    rData.append(readData(htables,67,2,2))
    rData.append(readData(htables,67,3,2))

    rData.append(readData(htables,68,1,2))
    rData.append(readData(htables,68,2,2))
    rData.append(readData(htables,68,4,2))

    rData.append(readData(htables,70,1,1))
    rData.append(readData(htables,70,3,1))
    rData.append(readData(htables,70,8,1))

    rData.append(readData(htables,71,4,1))
    rData.append(readData(htables,101,14,1))
    rData.append(readData(htables,101,14,4))
    rData.append(readData(htables,104,14,4))
rData.append(readData(htables, 104, 14, 5))
rData.append(readData(htables, 109, 14, 5))
rData.append(readData(htables, 109, 14, 6))

print(len(rData))
print(rData)

savetoDB(ID, rData)

#--- readdata ---#
def readData (htables, tableNum, rowNum, columnNum):
    initTabel = 11 + 1

    firstitem = htables[tableNum + initTabel] # table Number
    firstitem_table = firstitem[1] #
    x = firstitem_table[rowNum] # row
    # print (x[columnNum])

    return x[columnNum]

#--- savetodb ---#
def savetoDB (ID, rData):
    print rData

    res = cur.execute('select name from initMaxMin where ID = "" + str(ID) + '""

    res = cur.fetchone()

    x = 'replace into outputInit (seeID, timeStamp, name, Enduse Heating Elec
    ,Enduse Cooling Elec ,Enduse Intlighting Elec ,Enduse Heating Natural Gas ,Total Enduse
    ,Demand Enduse Heating Elec ,Demand Enduse Cooling Elec ,Demand Enduse Lighting Elec
    ,Demand Enduse Tot Elec ,Demand Enduse Tot Gas
    ,Envelope Opaque Reflectance ,Envelope Opaque Ufactor w ofil'm
    ,Envelope Opaque Ufactor w ofilm ,ExtWin Glass Ufactor ,ExtWin Glass SHGC
    ,ExtWin Glass VT ,IntLighting TotPower ,Energymeter Annual Elec Facil
    ,Energymeter Annual Elec Bldg ,Energymeter Annual Elec Intlight
    ,Energymeter Annual Elec IntEquipment ,Energymeter Annual Elec Cooling
    ,Energymeter Annual Gas Heating ,Enduse Total Elec
    ,Enduse Total Gas ,Enduse Total ,EnergyCost Elec Total ,EnergyCost Gas_Total
    ,EnergyCost Total ,EUI Elec Intlight ,EUI Elec Cooling
    ,EUI Elec Total ,EUI Gas Total ,ZoneElec Monthly Lights Avg ,ZoneElec Monthly Equipment Avg
    ,OccupantComfort Thermal PMV Avg
    ,OccupantComfort Thermal PPD Avg, Daylight_Illuminance, Daylight_Glare) VALUES("\n    '" + str(ID) + '"
    
    '" + str(res[0]) + '"
    
    '" + str(rData[0]) + '"
    
    '" + str(rData[1]) + '"
    
    '" + str(rData[2]) + '"
    
    '" + str(rData[3]) + '"
    
    '" + str(rData[4]) + '"
    
    '" + str(rData[5]) + '"
    
    '" + str(rData[6]) + '"
    
    '" + str(rData[7]) + '"
    
    '" + str(rData[8]) + '"
    
    '" + str(rData[9]) + '"
    
    '" + str(rData[10]) + '"
    
    '" + str(rData[11]) + '"
    
    '" + str(rData[12]) + '"
    
    '" + str(rData[13]) + '"
    
    '" + str(rData[14]) + '"
    
    '" + str(rData[15]) + '"
    
    '" + str(rData[16]) + '"
    
    '" + str(rData[17]) + '"
    
    '" + str(rData[18]) + '"
```python
'' + str(rData[19]) + ''
'' + str(rData[20]) + ''
'' + str(rData[21]) + ''
'' + str(rData[22]) + ''
'' + str(rData[23]) + ''
'' + str(rData[24]) + ''
'' + str(rData[25]) + ''
'' + str(rData[26]) + ''
'' + str(rData[27]) + ''
'' + str(rData[28]) + ''
'' + str(rData[29]) + ''
'' + str(rData[30]) + ''
'' + str(rData[31]) + ''
'' + str(rData[32]) + ''
'' + str(rData[33]) + ''
'' + str(rData[34]) + ''
'' + str(rData[35]) + ''
'' + str(rData[36]) + ''
'' + str(rData[37]) + ''
'' + str(rData[38]) + ''

print x
for z in rData:
    print z

res = cur.execute(x)

res = cur.execute('update initMaxMin set isDone = 1 where ID = '' + str(ID) + ''
# res = cur.execute('update outputInit set resWindowsRM ='' + str(rData[0]) + '',
# resComponentCES ='' + str(rData[1]) + '', resOccupantCDS =''
# + str(rData[2]) + '', resZoneESM ='' + str(rData[3]) + '', resZoneHSM ='' + str(rData[4]) + 
# , resZoneCSM ='' + str(rData[5]) + 
# '', resDaylightingRM ='' + str(rData[6]) + '' where ID ='' + str(ID) + ''

db.commit();
#--------------------------------------------------------------
def getOne (id):
    res = cur.execute('select ID, WWR, Wall_Cladding, Wall_Assembly, Insulation, Glazing_System,
    Solar_Control from initMaxMin where ID = '' + str(id) + ''
    res = cur.fetchall()
    for x in res:
        delFile()
        copyf()
        updateData(x)
        # break;
        runEP()
        saveData(x[0])
        # break;
        
        print ("Done")
    #--------------------------------------------------------------

id = sys.argv[1]

    copyf()
    initEPPY()
    initdb()
    # getAllData()
```
getOne(id)
#!/usr/bin/env python

import sys
from shutil import copyfile
import os
import MySQLdb
import pprint

#------------------------------------------------------------------------------
def initdb():
    global db
    global cur

    db = MySQLdb.connect(host="127.0.0.1",
                         user="mahsa",
                         passwd="1234r",
                         db="mahsa")

    cur = db.cursor(MySQLdb.cursors.DictCursor)

#------------------------------------------------------------------------------
def calcConsumption(senID):
    cur.execute('SELECT ID, EUI_Elec_IntLight, EUI_Elec_Cooling, EUI_Gas_Total FROM `output`
                 where seeID = "' + str(senID) + '")
    res = cur.fetchall()

    print res, senID
    point = 0
    for scenario in res:
        x = float(scenario['EUI_Elec_IntLight'])

        minMax = cur.execute('select min(EUI_Elec_IntLight), max(EUI_Elec_IntLight),
                              avg(EUI_Elec_IntLight) from outputInit WHERE name = "EUI_Elec_IntLight"')
        minMax = cur.fetchone()

        d = float(minMax['max(EUI_Elec_IntLight)']) - float(minMax['min(EUI_Elec_IntLight)'])
        d = d / 10
        point = point + (10 - ((x - float(minMax['min(EUI_Elec_IntLight)']) / d ))
        print(point, minMax, d, x, "----------------")

        x = float(scenario['EUI_Elec_Cooling'])

        minMax = cur.execute('select min(EUI_Elec_Cooling), max(EUI_Elec_Cooling),
                              avg(EUI_Elec_Cooling) from outputInit WHERE name = "EUI_Elec_Cooling"')
        minMax = cur.fetchone()

        d = float(minMax['max(EUI_Elec_Cooling)']) - float(minMax['min(EUI_Elec_Cooling)'])
        d = d / 10
        point = point + (10 - ((x - float(minMax['min(EUI_Elec_Cooling)']) / d ))
        print(point, "----------------")

        x = float(scenario['EUI_Gas_Total'])

        minMax = cur.execute('select min(EUI_Gas_Total), max(EUI_Gas_Total), avg(EUI_Gas_Total)
                              from outputInit WHERE name = "EUI_Gas_Heating"')
        minMax = cur.fetchone()

        d = float(minMax['max(EUI_Gas_Total)']) - float(minMax['min(EUI_Gas_Total)'])
        d = d / 10
point = point + (10 - ((x - float(minMax['min(EUI_Gas_Total)'])) / d))

print(point, "-----------------")
updatePoints(scenario['ID'], "consumption", round(point, 2))
# print "consumption\t", scenario['EUI_Elec_Total'], "\t", scenario['EUI_Gas_Total'], "\t -----> ", point
point = 0

db.commit();

#----------------------------------------------------------

def calcComfort(senID):
    cur.execute('SELECT ID, OccupantComfort_Thermal_PMV_Avg, Daylight_Illuminance, Daylight_Glare
FROM `output` where seeID ="" + str(senID) + "")
res = cur.fetchall()
print(res)
point = 0
for scenario in res:
    x = abs(float(scenario['OccupantComfort_Thermal_PMV_Avg']))
    if (x == 0):
        point = point + 20
    if (x > 0 and x <= 0.1):
        point = point + 18
    if (x > 0.1 and x <= 0.2):
        point = point + 16
    if (x > 0.2 and x <= 0.3):
        point = point + 14
    if (x > 0.3 and x <= 0.4):
        point = point + 12
    if (x > 0.4 and x <= 0.5):
        point = point + 10
    if (x > 0.5 and x <= 0.55):
        point = point + 8
    if (x > 0.55 and x <= 0.65):
        point = point + 4
    if (x > 0.65 and x <= 0.75):
        point = point + 2
    if (x > 0.75):
        point = point + 0

if(float(scenario['Daylight_Illuminance']) >= 300):
    x = float(scenario['Daylight_Illuminance'])
    minMax = cur.execute('select min(Daylight_Illuminance), max(Daylight_Illuminance),
    avg(Daylight_Illuminance) from\n    outputInit WHERE name = "Daylight_Illuminance"')
    minMax = cur.fetchone()
    d = float(minMax['max(Daylight_Illuminance)']) -
float(minMax['min(Daylight_Illuminance)'])
d = d / 15
point = point + ((x - float(minMax['min(Daylight_Illuminance)']))) / d )

if (scenario['Daylight_Glare'] <= 22):
x = float(scenario['Daylight_Glare'])

minMax = cur.execute('select min(Daylight_Glare), max(Daylight_Glare),
                        avg(Daylight_Glare) from outputInit WHERE name = "Daylight_Glare"')
minMax = cur.fetchone()
d = float(minMax['max(Daylight_Glare)']) - float(minMax['min(Daylight_Glare)'])
d = d / 15
point = point + (15 - ((x - float(minMax['min(Daylight_Glare)']))) / d ))

updatePoints(scenario['ID'], "comfort", round(point, 2))
# print "comfort\t", scenario["OccupantComfort_Thermal_PMV_Avg"], "\t------> ", point
point = 0
db.commit();
#-------------------------------------------------------------
def calcCost (senID):
cur.execute('SELECT ID, EnergyCost_Elec_Total, EnergyCost_Gas_Total  FROM `output` where

seeID = "' + str(senID) + '"')
res = cur.fetchall()

point = 0
for scenario in res:

x = float(scenario['EnergyCost_Elec_Total'])

minMax = cur.execute('select min(EnergyCost_Elec_Total), max(EnergyCost_Elec_Total),
avg(EnergyCost_Elec_Total) from outputInit WHERE name = "EnergyCost_Elec_Total"')
minMax = cur.fetchone()
d = float(minMax['max(EnergyCost_Elec_Total)']) - float(minMax['min(EnergyCost_Elec_Total)'])
d = d / 10
point = point + (10 - ((x - float(minMax['min(EnergyCost_Elec_Total)']))) / d ))

print(minMax)
print(point,x, "------------------")

x = float(scenario['EnergyCost_Gas_Total'])

minMax = cur.execute('select min(EnergyCost_Gas_Total), max(EnergyCost_Gas_Total),
avg(EnergyCost_Gas_Total) from outputInit WHERE name = "EnergyCost_Gas_Total"')
minMax = cur.fetchone()

print(minMax)
d = float(minMax['max(EnergyCost_Gas_Total)']) - float(minMax['min(EnergyCost_Gas_Total)'])
d = d / 10
point = point + (10 - ((x - float(minMax['min(EnergyCost_Gas_Total)']))) / d ))

print(point,x, "------------------")
updatePoints(scenario['ID'], "cost", round(point, 2))
# print "cost\t", scenario['EnergyCost_Total'], "\t-------> ", point
point = 0

db.commit();

# -----------------------------------

def updatePoints(outPutID, name, val):
    cur.execute('UPDATE output set ' + str(name) + '="' + str(val) + '" where ID = "' + str(outPutID) + '"')

# -----------------------------------

def main():
    # get first or next
    id = oneCycle(0, "")
    # id = 2633
    for i in range(2000):
        res = decisionMaker(id)
        print res
        if res == "nothing":
            id = oneCycle(0, "")
        else:
            id = oneCycle(1, getSkipVal(res, id))

    #make decision

    # loot
    # -----------------------------------

def oneCycle(command, id):
    if command == 0:
        res = cur.execute('SELECT seeID from output ORDER BY `ID` DESC limit 1')
        res = cur.fetchone()
        if res is None:
            lastID = 0
        else:
            lastID = int(res['seeID'])

        nextID = lastID + 1
        print """--------------------"", nextID
        cur.execute('SELECT ID from allSce where Solar_Control != "Blank" and isDone = "0" and ID = "' + str(nextID) + '" ORDER BY `ID` ASC limit 1')

    if command == 1:
        cur.execute('SELECT ID from allSce where Solar_Control != "Blank" and isDone = "0" and ID = "' + str(id) + '" ORDER BY `ID` ASC limit 1')

    print command, id
    res = cur.fetchone()
    print res
    if res is None:
        cur.execute('SELECT ID from allSce where Solar_Control != "Blank" and isDone = "0" and id > "' + str(id) + '" ORDER BY `ID` ASC limit 1')
        res = cur.fetchone()
id = res['ID']
print("---------------------", id)

os.system("python bil2.py " + str(id))

#calc
print id
id = str(id)
db.commit();
calcConsumption(int(id))
calcComfort(int(id))
calcCost(int(id))

return id

#-------------------------------------------------

def decisionMaker(id):
cur.execute('SELECT consumption, comfort, cost from output where seeID = "' + str(id) + '"')
res = cur.fetchone()

print res
if res is None:
cur.execute('update allSce set isDone = "3" where ID = "' + str(id) + '"')
db.commit();
return 'nothing-a'

consumption = float(res['consumption'])
comfort = float(res['comfort'])
cost = float(res['cost'])

x = consumption + comfort + cost
if (x < 70):
    if consumption < 25 :
        return 'consumption'
    if comfort < 50 :
        return 'comfort'
else :
    return 'cost'

return 'nothing'

#-------------------------------------------------

def getSkipVal (skipParam, id):
cur.execute('SELECT * from allSce where isDone = "1" and ID = "' + str(id) + '"')
res = cur.fetchone()

print res
if skipParam == "consumption": #Glazing, insulation int
glz = res['Glazing_System']
ingI = res ['Insulation']
fx = cur.execute('update allSce set isDone = "2" where isDone = "0" and (Glazing_System = "' + glz + '/' and Insulation ="' + insI + '/")')
print ("\/\/\/\/\/\//\//\//\//\//\//", fx)
db.commit();
cur.execute('SELECT ID from allSce where isDone = "0" AND (Glazing_System != "' + glz + /" or Insulation !="' + insI + /") ORDER BY `ID` ASC limit 1 ')
res = cur.fetchone()
res = res['ID']
cur.execute('UPDATE allSce set isDone ="2" where ID < ' + str(res) + ' and isDone = "0"')
db.commit();
return res
if skipParam == "comfort": # WWR, gazing system,
    WWR = res['WWR']
    glz = res ['Glazing_System']
    fx = cur.execute('update allSce set isDone ="2" where isDone = "0" and (WWR = "" + WWR + " and Glazing_System = "" + glz + ")'
    print ("\\", fx)
    db.commit();
    cur.execute('SELECT ID from allSce where isDone = "0" AND (WWR != "" + WWR + " or Glazing_System !="" + glz + ") ORDER BY `ID` ASC limit 1'
    res = cur.fetchone()
    cur.execute('UPDATE allSce set isDone ="2" where ID < ' + str(res) + ' and isDone = "0"
    return res

if skipParam == "cost":#insulation ext, cladding  glazing system
    insE = res['Insulation']
    wallC = res ['Wall_Cladding']
    fx = cur.execute('update allSce set isDone ="2" where isDone = "0" and (Insulation = "" + insE + " and Wall_Cladding ="" + wallC + ")'
    print ("\"", fx)
    db.commit();
    cur.execute('SELECT ID from allSce where isDone = "0" AND (Insulation != "" + insE + " or Wall_Cladding !="" + wallC + ") ORDER BY `ID` ASC limit 1'
    res = cur.fetchone()
    cur.execute('UPDATE allSce set isDone ="2" where ID < ' + str(res) + ' and isDone = "0"
    return res

# restart()
def restart():
    cur.execute('TRUNCATE TABLE `output`
    cur.execute('UPDATE allSce set isDone = "0"
    return res

initdb()

# restart()
# print oneCycle(1, getSkipVal("cost", 49))
# print getSkipVal("cost", 49)

main()
# calcCost(2199)

# restart()
APPENDIX B

DATABASE

(These are samples to illustrate the database)
### Table comments:

#### Space usage

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>3.5</td>
<td>MiB</td>
</tr>
<tr>
<td>Index</td>
<td>0</td>
<td>B</td>
</tr>
<tr>
<td>Total</td>
<td>3.5</td>
<td>MiB</td>
</tr>
</tbody>
</table>

#### Row statistics

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Compact</td>
</tr>
<tr>
<td>Collation</td>
<td>latin1_swedish_ci</td>
</tr>
<tr>
<td>Next autoindex</td>
<td>26,461</td>
</tr>
<tr>
<td>Creation</td>
<td>Dec 01, 2019 at 09:10 PM</td>
</tr>
</tbody>
</table>

---

### Partitions

No partitioning defined!
<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>int(11)</td>
<td></td>
<td>No</td>
<td>None</td>
<td>AUTO_INCREMENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>isDone</td>
<td>int(11)</td>
<td></td>
<td>No</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>name</td>
<td>varchar(120)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>WWR</td>
<td>varchar(3)</td>
<td>latin1_swedish_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Wall_Cladding</td>
<td>varchar(80)</td>
<td>latin1_swedish_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Wall_Assembly</td>
<td>varchar(80)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Insulation</td>
<td>varchar(80)</td>
<td>latin1_swedish_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Glazing_System</td>
<td>varchar(80)</td>
<td>latin1_swedish_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Solar_Control</td>
<td>varchar(80)</td>
<td>latin1_swedish_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Partitions**

No partitioning defined!

**Information**

Table comments:

<table>
<thead>
<tr>
<th>Space usage</th>
<th>Row statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data: 16 KiB</td>
<td>Format: Compact</td>
</tr>
<tr>
<td>Index: 0 B</td>
<td>Collation: latin1_swedish_ci</td>
</tr>
<tr>
<td>Total: 16 KiB</td>
<td>Next autoindex: 17</td>
</tr>
<tr>
<td></td>
<td>Creation: Oct 22, 2019 at 09:30 PM</td>
</tr>
<tr>
<td>#</td>
<td>Name</td>
</tr>
<tr>
<td>----</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>ID</td>
</tr>
<tr>
<td>2</td>
<td>seeID</td>
</tr>
<tr>
<td>3</td>
<td>timeStamp</td>
</tr>
<tr>
<td>4</td>
<td>consumption</td>
</tr>
<tr>
<td>5</td>
<td>comfort</td>
</tr>
<tr>
<td>6</td>
<td>cost</td>
</tr>
<tr>
<td>7</td>
<td>Enduse_Heating_Elec</td>
</tr>
<tr>
<td>8</td>
<td>Enduse_Cooling_Elec</td>
</tr>
<tr>
<td>9</td>
<td>Enduse_IntLighting_Elec</td>
</tr>
<tr>
<td>10</td>
<td>Enduse_Heating_Natural_Gas</td>
</tr>
<tr>
<td>11</td>
<td>Total_Enduse</td>
</tr>
<tr>
<td>12</td>
<td>Demand_Enduse_Heating_Elec</td>
</tr>
<tr>
<td>13</td>
<td>Demand_Enduse_Cooling_Elec</td>
</tr>
<tr>
<td>14</td>
<td>Demand_Enduse_Lighting_Elec</td>
</tr>
<tr>
<td>15</td>
<td>Demand_Enduse_Tot_Elec</td>
</tr>
<tr>
<td>16</td>
<td>Demand_Enduse_Tot_Gas</td>
</tr>
<tr>
<td>17</td>
<td>Envelope_Opaque_Returnance</td>
</tr>
<tr>
<td>18</td>
<td>Envelope_Opaque_Ufactor_w_ofilm</td>
</tr>
<tr>
<td>19</td>
<td>Envelope_Opaque_Ufactor_w_ofilm</td>
</tr>
<tr>
<td>20</td>
<td>ExtWin_Glass_Ufactor</td>
</tr>
<tr>
<td>21</td>
<td>ExtWin_Glass_SHGC</td>
</tr>
<tr>
<td>22</td>
<td>ExtWin_Glass_VT</td>
</tr>
<tr>
<td>23</td>
<td>IntLighting_TotPower</td>
</tr>
<tr>
<td>24</td>
<td>Energymeter_Annual_Elec_Facil</td>
</tr>
<tr>
<td>25</td>
<td>Energymeter_Annual_Elec_Bldg</td>
</tr>
<tr>
<td>26</td>
<td>Energymeter_Annual_Elec_IntLight</td>
</tr>
<tr>
<td>27</td>
<td>Energymeter_Annual_Elec_IntEquipment</td>
</tr>
<tr>
<td>28</td>
<td>Energymeter_Annual_Elec_Cooling</td>
</tr>
<tr>
<td>29</td>
<td>Energymeter_Annual_Elec_Gas_Heating</td>
</tr>
<tr>
<td>30</td>
<td>Enduse_Total_Elec</td>
</tr>
<tr>
<td>31</td>
<td>Enduse_Total_Gas</td>
</tr>
<tr>
<td>32</td>
<td>Enduse_Total</td>
</tr>
<tr>
<td>33</td>
<td>EnergyCost_Elec_Total</td>
</tr>
<tr>
<td>34</td>
<td>EnergyCost_Gas_Total</td>
</tr>
<tr>
<td>35</td>
<td>EnergyCost_Total</td>
</tr>
<tr>
<td>36</td>
<td>EUI_Elec_IntLight</td>
</tr>
<tr>
<td>37</td>
<td>EUI_Elec_Cooling</td>
</tr>
<tr>
<td>38</td>
<td>EUI_Elec_Total</td>
</tr>
<tr>
<td>39</td>
<td>EUI_Gas_Total</td>
</tr>
<tr>
<td>40</td>
<td>ZoneElec_Monthly_Lights_Avg</td>
</tr>
<tr>
<td>41</td>
<td>ZoneElec_Monthly_Equipment_Avg</td>
</tr>
<tr>
<td>42</td>
<td>OccupantComfort_Thermal_PMV_Avg</td>
</tr>
<tr>
<td>43</td>
<td>OccupantComfort_Thermal_PPD_Avg</td>
</tr>
<tr>
<td>44</td>
<td>Daylight_Illuminance</td>
</tr>
<tr>
<td>45</td>
<td>Daylight_Glare</td>
</tr>
</tbody>
</table>

187
### Table comments:

#### Space usage

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>112  KiB</td>
</tr>
<tr>
<td>Index</td>
<td>16   KiB</td>
</tr>
<tr>
<td>Total</td>
<td>128  KiB</td>
</tr>
</tbody>
</table>

#### Row statistics

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Compact</td>
</tr>
<tr>
<td>Collation</td>
<td>latin1_swedish_ci</td>
</tr>
<tr>
<td>Next autoindex</td>
<td>253</td>
</tr>
<tr>
<td>Creation</td>
<td>Nov 16, 2019 at 05:23 PM</td>
</tr>
</tbody>
</table>
APPENDIX C

OUTPUTS

(These are samples to illustrate the outputs and results)
<p>| ID | see | timeStamp | 267 | 515 | ID | see | timeStamp | 265 | 513 | ID | see | timeStamp | 263 | 511 | ID | see | timeStamp | 259 | 502 | ID | see | timeStamp | 258 | 501 | ID | see | timeStamp | 254 | 497 | ID | see | timeStamp | 253 | 496 | ID | see | timeStamp | 246 | 409 | ID | see | timeStamp | 238 | 396 | ID | see | timeStamp | 216 | 289 | ID | see | timeStamp | 210 | 233 | ID | see | timeStamp | 167 | 79 | ID | see | timeStamp | 145 | 20 | ID | see | timeStamp | 126 | 626 | ID | see | timeStamp | 120 | 596 | ID | see | timeStamp | 102 | 506 | ID | see | timeStamp | 74 | 366 | ID | see | timeStamp | 72 | 356 | ID | see | timeStamp | 69 | 341 | ID | see | timeStamp | 53 | 261 | ID | see | timeStamp | 41 | 201 | ID | see | timeStamp | 23 | 111 | ID | see | timeStamp |</p>
<table>
<thead>
<tr>
<th>Page</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>192</td>
<td>192</td>
</tr>
</tbody>
</table>

---

**Markdown Format:**

```markdown
<table>
<thead>
<tr>
<th>Page</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>192</td>
<td>192</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>Year</th>
<th>Value</th>
<th>Year</th>
<th>Value</th>
<th>Year</th>
<th>Value</th>
<th>Year</th>
<th>Value</th>
<th>Year</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1921</td>
<td>17.82</td>
<td>1932</td>
<td>17.81</td>
<td>1943</td>
<td>17.83</td>
<td>1954</td>
<td>17.84</td>
<td>1965</td>
<td>17.85</td>
</tr>
<tr>
<td>1923</td>
<td>18.28</td>
<td>1934</td>
<td>18.25</td>
<td>1945</td>
<td>18.27</td>
<td>1956</td>
<td>18.28</td>
<td>1967</td>
<td>18.29</td>
</tr>
<tr>
<td>1927</td>
<td>18.29</td>
<td>1938</td>
<td>18.27</td>
<td>1949</td>
<td>18.29</td>
<td>1960</td>
<td>18.30</td>
<td>1971</td>
<td>18.31</td>
</tr>
<tr>
<td>1929</td>
<td>18.32</td>
<td>1939</td>
<td>18.30</td>
<td>1951</td>
<td>18.32</td>
<td>1962</td>
<td>18.33</td>
<td>1973</td>
<td>18.34</td>
</tr>
<tr>
<td>1931</td>
<td>18.35</td>
<td>1940</td>
<td>18.33</td>
<td>1952</td>
<td>18.35</td>
<td>1964</td>
<td>18.36</td>
<td>1975</td>
<td>18.37</td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY


