A simulation-based framework to optimize occupant-centric controls given stochastic occupant behaviour

Zeinab Khorasani Zadeh

A Thesis
In the Department
of
Building, Civil and Environmental Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science (Building Engineering) at
Concordia University
Montreal, Quebec, Canada

March2022

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Zeinab Khorasani Zadeh

Entitled: A simulation-based framework to optimize occupant-centric controls given stochastic occupant behaviour

And submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Building Engineering)

Complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final Examining Committee:

______________________________ Chair
Dr. Fuzhan Nasiri

______________________________ Examiner
Dr. Fuzhan Nasiri

______________________________ Examiner
Dr. Ursula Eicker

______________________________ Thesis Supervisor
Dr. Mohamed M. Ouf

Approved by

____
Dr. Mazdak Nik-Bakht, Graduate Program Director

March 26, 2022

____
Dr. Mourad Debbabi, Dean, Gina Cody School of Engineering and Computer Science

Abstract

A simulation-based framework to optimize occupant-centric controls given stochastic occupant behaviour

Zeinab Khorasani Zadeh

Occupant-centric control (OCC) strategies represent a novel approach to indoor climate control in which occupancy patterns and occupant preferences are embedded within control sequences. They aim to improve occupant comfort and energy efficiency by learning and predicting occupant behaviour (OB), then optimizing building operations accordingly. Previous studies estimate OCC can increase energy savings by up to 60% while improving occupant comfort. However, their performance is subject to several factors, including uncertainty due to OB, OCC configurational settings, as well as building design parameters. To this end, testing OCCs and adjusting their configurational settings before implementation are critical to ensure optimal performance. Furthermore, identifying building design alternatives that can optimize such performance is an important step that faces logistical constraints during field implementations. This research presents a framework to optimize OCC performance in a simulation environment, which entails coupling synthetic OB models with OCCs that learn their preferences. The framework features a parallel processing structure to obviate the computational burden and enhance optimization efficiency. A sensitivity analysis is conducted to identify the most influential variables on OCC performance in terms of increasing energy efficiency and occupant comfort. A two-step multi-objective optimization is then developed to identify the configurational settings and design parameters that minimize energy consumption and maximize occupant comfort. Results revealed significant improvement in OCC performance when they were customized with the identified optimal settings for different occupants. The proposed framework aims to improve OCC performance in actual buildings and avoid discomfort issues that may arise during its initial implementation.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor Dr. Mohamed Ouf for his continuous intellectual and personal support, patient guidance, encouragement, and advice. I am very grateful for the opportunity to work under his supervision and for his insightful guidance and inspiring feedback and suggestions.

My grateful thanks are also extended to my examiners for their time in reviewing my thesis and for their constructive comments.

This research has been supported by grants from the Natural Sciences and Engineering Research Council of Canada (NSERC). These supports are greatly appreciated.

I would like to give my special thanks to my beloved parents, Malihe and Rasoul, for their endless love, spiritual support, and encouragement. Without them, I could never have reached success in all stages of my life.

My sincere thanks go to my close friend and my roommate Miss Vida Fathi who was always there for me with her positivity, which gave me enough energy to walk towards my goals.

I must also acknowledge the support and encouragement I got from my research group in the Intelligent Buildings and Cities lab (IBCL) at Concordia University.

Dedicated to my lovely parents: Malihe and Rasoul

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABM	Agent-Based Modelling
BAS	Building Automation System
BDI	Belief Desire Information
BPS	Building Performance Simulation
CSP	Cooling Set-Point
EMS	Energy Management System
EUI	Energy Use Intensity
FMI	Functional Mock-up Interface
HPC	High-Performance Computing
HSP	Heating Set-Point
HVAC	Heating, Ventilation, and Air Conditioning
LR	Learning Rate
NECB	National Energy Code of Canada for Building
NSGA	Non-Dominated Sorting Algorithm
OB	Occupant Behaviour
OCC	Occupant-Centric Control
OPA	Occupant Presence and Action
P.O.P	Pareto-Optimal Point
RL	Reinforcement Learning
SD	Standard Deviation
TUS	Time Use Survey
VAV	Variable Air Volume

Chapter 1: Introduction

Global energy consumption will grow by nearly 50% between 2018 and 2050 [1]. Thirty percent of the total energy consumption is attributed to the global building sector (22% for residential and 8% for non-residential buildings) [2]. In the United States, residential and commercial buildings accounted for 22% and 18% (40% combined) of total U.S. energy consumption in 2020, respectively [1]. For any conventional building, building operations constitute 80%-90% of building life cycle energy use [3]. Therefore, building operation management plays a crucial role in reducing energy consumption in buildings [4]. More than 73% of total energy consumption in commercial buildings is attributed to space heating, cooling, and lighting (Figure 1) [5]. As a result, focusing on these three categories can significantly reduce energy usage in commercial or institutional buildings. The large discrepancy between estimated building energy usage in the design phase and the actual energy usage in the actual operation highlighted the role of occupants in building energy use [6] .The human factor, including occupant behaviour and preferences, significantly affects building operations and, consequently building energy usage and performance [2]. Energy consumption in commercial buildings showed a variation of 30% to 150% due to occupant behaviour [6]. Besides, without considering occupant behaviour and needs, control engineers must make conservative assumptions which lead to operating schedules that exceed occupied hours (energy waste), as well as temperature setpoints that result in cold complaints in the summer and hot complaints in the winter (Occupant discomfort) [7]. Providing occupant comfort is of great importance to the life quality of occupants, including health and productivity. This relationship between occupant behaviour and building energy usage, on the one hand, and the goal of improving building energy consumption while maintaining or increasing occupant comfort, on the other hand, highlights the necessity of an intelligent approach to maintain user expected comfort while decreasing energy use during the operation of buildings [8].

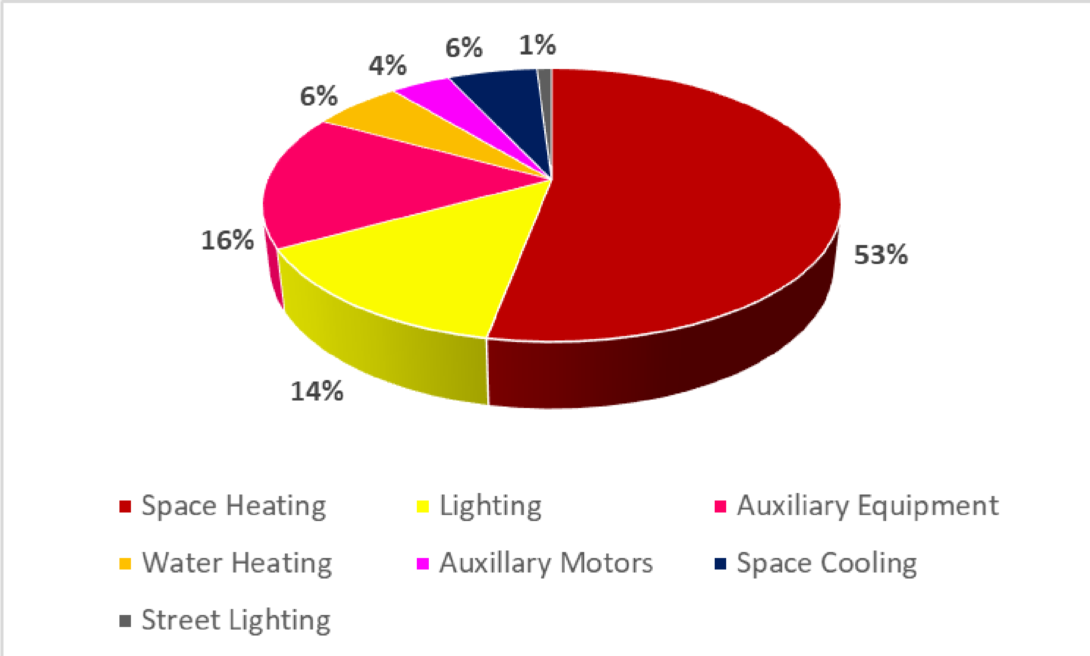


Figure 1: Commercial/institutional energy use by end-use 2018.

Given the recent developments in information and communication technology, Occupant-Centric building Control (OCC) strategies have been introduced in which a control system acquires various data from occupants, the indoor environment, and the outdoor climate. The occupant-related information is gained directly or indirectly through various sensors, occupant feedback from control interfaces, and mobile or wearable devices. The derived information is then used for building controls, e.g., room occupancy patterns and adaptive setpoints, to improve energy efficiency and occupant comfort [9], [10]. The newly introduced OCC algorithms, which can address the conflicts between the objectives of saving energy and ensuring occupant comfort, continue to be brought up to have better control of building operations [11].

Many OCC-related studies have indicated that implementing smart control strategies is proliferating to make buildings more energy-efficient while improving the level of occupant comfort. Although previous studies showed that energy efficiency and occupant comfort can be achieved by OCC, the actual performance of OCCs has a significant potential for further investigation and improvement. In addition, generalizing the performance of various OCC algorithms is known as one of the main limitations of deploying OCC in real buildings. As OCCs learn from occupant behaviour and their interactions with building systems, there is a three-way relationship between OCC configurations, the type of occupant subject to OCC, and building design parameters that needs to be optimized. OCC's performance can change while learning from different types of occupants' interactions within various building design alternatives. Therefore, fine-tuning OCC configurations for different building design parameters and various types of occupants' preferences is crucial in improving the performance of these control systems. In addition, there is an inverse relationship between minimizing building energy consumption and improving occupant comfort. Identifying a balance between these two important objectives is another important stride in upgrading OCC performance. Multi-objective optimization algorithms are required to investigate all the possible various combinations settings thoroughly. Regulating OCCs with these optimal solutions will enhance control of the indoor environment, which improves building energy performance while meeting occupants' needs at the same time. To the best of the authors' knowledge, there has not been any research on the multi-objective optimization of the investigated OCCs performance to customize their configuration when learning from various types of occupants.

1.1 Research objectives

This research aims to develop a simulation-based framework to optimize OCC configurational settings while working with various scenarios of occupant preferences before their implementation. The proposed simulation framework integrates stochastic occupant behaviour models as well as OCCs, which learn from occupants' interaction with building systems and then control building operations accordingly. To provide a proof-of-concept of the framework, a single office was modelled with different design parameters to represent the capability of the proposed framework to investigate OCC performance and identify their optimal configurational settings. The main objectives of this research are defined as follows:

1- Developing an energy model entailing stochastic occupant behaviour models as well as OCCs algorithms that learn from occupants' preferences into the building energy simulation program, EnergyPlus.
2- Integrating the energy simulation model with a sensitivity analysis algorithm to investigate the effects of building design variables and OCC configurational settings variables on energy consumption and occupant comfort.

3- Developing a two-step optimization framework to identify the optimal OCC configurational settings while working with various scenarios of occupant preferences before their implementation.
4- Presenting a case study as a proof-of-concept of the framework to represent the capability of the framework to investigate OCC performance and identify their optimal configurational settings.

1.2 Thesis Organization

The structure of this thesis is as follows: Chapter 1 provides an introduction, including background, problem statement, and a summary of the thesis objectives. Chapter 2 contains a review of the literature on (1) different occupancy and occupant behaviour modeling approaches; (2) OCCs and recent developments in these control strategies; and (3) recent studies to integrate occupant behaviour and OCCs in building simulation tools. Chapter 3 provides a detailed explanation of the methodology used to develop the proposed simulation-based optimization framework. In addition, the case study which is used as a proof of concept of the proposed framework, is described in this chapter. Chapter 4 presents the case study results implemented in the proposed framework, and the major findings are discussed. Finally, Chapter 5 provides a summary of the research and its contribution. In addition, the limitation of the current study and the recommendation for future studies are presented.

Chapter 2: Literature review

2.1 Introduction

This chapter aims to provide, firstly, a review of occupancy and occupant behaviour modeling approaches. This is then followed by providing an overview of recent developments in OCC strategies. Finally, recent studies on the integration of OB and OCC models in building simulation tools are reviewed.

2.2 Occupancy and occupant behaviour modeling

Occupants are widely recognized as an influential factor in building energy performance and cause a discrepancy between predicted and real energy consumption in buildings [12]. In reality, they are active recipients in building as opposed to passive recipients meaning that once they feel uncomfortable, they interact building systems and components to restore their comfort. Occupants can adapt building indoor environment to their comfort by switching lights on, changing thermostat setpoints, opening or closing operable windows, and opening or closing window shades. In addition, they can adapt themselves to indoor climates. For example, they can change their clothing level, activity level or drink hot or cold beverages [13], [14]. These types of behaviours are called adaptive behaviours [15]. On the other hand, non-adaptive behaviours are motivated by contextual reasons (non-physical factors affecting occupant behaviour) rather than physical discomfort [14]. They are not usually undertaken to restore comfort [13]. As an example, occupants are more likely to switch off the light or shut down their computers when they are going on vacation [16]. In general, there are four main occupant categories to model adaptive and non-adaptive behaviours (Table 1). These approaches are categorized as: 1- Static or Dynamic and 2- Deterministic or Stochastic (aka probabilistic) [17]. Static models are those in which occupants are considered as input and are not affected by the building, while dynamic models can model a two-way relationship between building and occupant. In other words, occupants can be affected and respond to changes in building and conditions. Deterministic approaches are fixed models, and simulation gives the same results after every run. In contrast to the deterministic model, Stochastic models are related to the randomness of behaviours [13]. The most popular approach is deterministic-static models in the form of fixed schedules. These models are easy to implement and have the advantage of simplicity, repeatability, and transparency [18]. However, they ignore the two-way relationship between occupant and building and can not reflect the uncertainty due to occupant behaviour. Stochastic-static models can provide insight on uncertainty; they cannot characterize the two-way relationship between occupants and building, though. As a result, they can be used to model non-adaptive OB such as occupancy or plug loads. Deterministic-dynamic models enable characterizing the two-way interactions of occupants and building but do so consistently. Stochastic-dynamic models which is the focus of this research are the state-of-the-art in OB modeling. They allow both the two-way interactions between occupant and building as well as presenting the stochastic aspect of OB [18]. The advantage and disadvantages of each of these four models are shown in Table 2.

Table 1: The four categories of modeling adaptive and non-adaptive OB.

Type	**Static**	**Dynamic**
Deterministic	A fixed schedule (e.g., lighting, occupancy, etc.)	Use a fixed threshold to determine if the value of schedule is 0 or 1 (e.g., light is turned on below a fixed indoor illuminance threshold).
Stochastic	Fixed schedule that multiplies by a randomly generated number.	A model that responds to the changes in conditions with a degree of randomness (e.g., a light use model that predicts the probability of light switches on with the value between zero and one over a wide range of indoor illuminances.

Table 2: Advantages and disadvantages of the four categories for OB modeling [17].

Model	**Advantages**	**Disadvantages**
static	Simple and suitable for non-adaptive behaviour (e.g., Occupancy).	Ignore the effect of building designs on occupant adaptive behaviour and vice versa.
Dynamic	Enables understanding the effect of building designs on occupant behaviour and vice versa.	Requires more details (e.g., room temperature or indoor illuminance).

Deterministic	Requires only a single simulation as it gives the same result for every simulation.	Do not imply building performance uncertainty.
Stochastic	Provides a range of feasible results and occupant diversity.	Requires approximately 50 to 100 simulations.

Thanks to the recent development in information and communication technologies, several mathematical methods have been developed to quantitively model occupants' presence and actions (OPA) in the building using the collected data during the occupants' monitoring period [19]. Advanced OB modeling has been developed based on a wide range of statistical methods to model stochastic OB. Logistic regression model is a common statistical model that has been widely used to model stomachic OB due to its capability to model binary dependent variables (e.g., if an action happened or not) as well as using non-normal distribution [13]. Another approach among advanced OB modeling is agent-based modeling (ABM). In ABM, occupant is treated as an individual who can make decisions and take actions and interact with the others [20]. In 2021, Zambrano et al. classified OB models based on their level of complexity (from the lowest to the highest level): fixed schedule, non-probabilistic models, stochastic (probabilistic models), and ABM [20]. This section provides a summary of the two most complicated OB modeling categories (stochastic and ABM) with more details on stochastic occupant behaviour modeling as the focus of this research.

2.2.1 Probabilistic/Stochastic models

As occupants behave in a random way, stochastic models are developed to model OB under various situations. These models estimate the probability of action based on historical or statistical data [21]. Three types of stochastic occupant behaviour modeling are commonly used: Markov chain models, Bernoulli process, and survival analysis [22], [23], [24]. Bernoulli models predict the state of a building system or component [13]. They can be used when the probability of a state or event is independent of previous states or events. They are sufficient for large-scale energy modeling but do not represent individual behaviour and the timing of occupant actions [23]. Survival models are usually used to estimate the duration when a state remains unchanged by the occupant [25]. As an example, Wang et al., (2005) proposed a survival model to predict the duration of time it takes when the occupant goes for a lunch or coffee break [26]. Markov chain models are widely used to predict OB among these three approaches. It is a time series of data in which all the states of the systems can be observed directly. The basic assumption of this approach is that the future state only depends on the present state while being independent of all past states. These models are developed using two types of data: Time use Survey data (TUS) [27] and sensor measured data [28]. The advantage of Markov chain models is to predict the probability of transition between two states rather than one state and therefore predicts behaviour patterns more realistically [14], [15]. Markov chain models are divided into two main categories to predict occupant behaviour: Discrete-time Markov model and discrete-event Markov model. Discrete-time Markov models predict the likelihood of an action by the occupant in the next time step [14]. Thus, the states can change only at fix discrete time intervals. To address this shortcoming, discrete-event Markov models are developed in which the probability of an action is linked to an event. For example, Reinhart developed a model for light switches using the discrete-event Markov approach. This model implies that occupant is more likely to switch light on (action) at arrival (event) and switches the light off (action) at departure (event) [24]. Despite discrete-time Markov models that only work with fixed time steps, discrete-event probabilistic models have been proved to be efficient since they can incorporate various time steps in the simulation. However, a key to the success of developing these stochastic models is to link an occupant action to a suitable event (as a substitute for time step) [14].

2.2.2 Agent-based models (ABM)

Agent-based models are computational methods comprised of multi-agents that can interact with each other and their environment under defined rules [12], [29]. Each agent can evaluate their situation and change their behaviours in response to other agents and the environment. Andrew et al. (2013) proposed a comprehensive framework to model occupant behaviour using agent-based modeling. The framework was comprised of two main sub-models: building performance sub-model and human agent sub-model. Building performance sub-model tracked and modified the state of the indoor environment. The human agent sub-model simulated occupant response and reaction to the changing environmental conditions. As for the occupant module, a procedurally oriented model of human decision-making called the Belief-Desire-Intention (BDI) framework was introduced, which was then enriched to an advanced version [30].

ABM has the flexibility of adding more agents to simulate multiple behaviours as well as the ability to extend for further levels such as building levels. In addition, these models are well suited to consider OB's social and psychological aspects. However, real-time communication between ABM and building energy simulation programs is challenging as these programs are usually unique in their coding language [12].

2.3 Occupant centric control strategies

OCC strategies represent a novel approach for indoor climate control in which occupancy patterns and occupant preferences are embedded within control sequences. They aim to improve both occupant comfort and energy efficiency by learning and predicting occupant behaviour then optimizing building operations accordingly. In recent years, applying OCC to improve the building design and operation through BAS has been subject to increased academic interest [6]. OCC can be divided into two main categories based on the type of occupant-related data it uses: 1) Occupancy-centric controls and 2) Occupant behaviour-centric controls. Occupancy-centric control adjusts setpoints and schedules based on presence or absence of occupant data or occupant count data. However, Occupant-behaviour-centric controls learn from occupant behaviour and preferences (e.g., through their interactions) and then adjust building system operations (e.g., heating and cooling setpoints) [9].

The past decade showed a significant development of OCC algorithms [31]. For example, an adaptive lighting and blind control algorithm were developed based on analyzing occupant behaviour in ten private offices [32]. The photosensor setpoints to switch off the lights and open the blinds were derived from an algorithm that learns from occupants' preference towards light switch on and blind closing behaviour. The results indicated that the control algorithm could considerably reduce light electricity consumption (25%) without affecting occupant comfort negatively. A reinforcement learning (RL)-based OCC for thermostats was also developed by [33]. The agent learned occupant behaviour and indoor environment while monitoring indoor air temperature, occupancy, and thermal comfort and determined the thermostat setpoint to reach a balance between energy consumption and occupant comfort. Jung et al. (2021) proposed a novel OCC algorithm to control indoor temperature setpoint using a deep learning algorithm [34]. Occupant activity recognition was performed using a one-dimensional convolutional neural network model, and an RL-based model was developed to control indoor temperature. The results showed that the developed RL model decreased thermal discomfort by 50.3% compared to the RL-based model without considering occupant activities. In addition, the proposed control method reduced occupant thermal discomfort by 10.9% more than manual control. Ye et al. (2021) evaluated the energy-saving potential of deploying two advanced OCCs, including presence-based and counting-based strategies in primary schools in the US [35]. The results revealed that the energy-saving potential of counting-based OCC was up to 12% and 10% for presence-based OCC. Although literature witnessed a major effort for the development of OCC algorithms, the implementation of these control systems in real buildings is very limited due to some logistical constraints. Researchers employed simulation-based methodologies to investigate the performance of OCCs by modeling the target building. Bakker et al. (2017) highlighted that many aspects of these control strategies could be further developed using simulation to increase energy-saving and occupant comfort before real-world applications [36].

2.4 Integrating OB and OCC in building performance simulation (BPS)

Traditionally, deterministic schedules of BPS were used to represent occupant-related input. Over the past decade, advanced mathematical models have been developed to predict occupancy profile and occupant behaviour more accurately. Generally, to integrate OB models in BPS, these stochastic mathematical equations should be formulated using custom functions to overwrite default controls [37]. Ouf et al. (2018) provided an overview of various methods for incorporating OB models into BPS [37]. They classified all the methods of integrating OB in BPS into two main approaches: 1) direct input and 2) user modifications. Direct input includes defining schedules for the operation of different building components such as HVAC system, specifying changes in occupant density and the use of light and equipment, and designating specific rules for using lights, operable windows and shading devices in common BPS tools. Although these methods can be beneficial for predicting thermal loads, they are incapable of representing occupant adaptive and non-adoptive behaviours. The user modifications approach, however, enables integrating more comprehensive occupant adaptive and non-adaptive behviour, including human-building interactions. The user modification approach is divided into three sub-categories, including 1) co-simulation, 2) custom function, and 3) user-modified source code [37].

Co-simulation is a method that allows different simulation tools to run at the same time and simulate different components while switching information in a combined routine [38]. As an example, Hong et al. (2016) presented a new OB modeling tool with an occupant behaviour Functional Mock-up Unit (obFMU) [39]. It allows co-simulation of OB with BEM programs based on the Functional Mock-up Interface (FMI), which does not restrict users to the specific simulation program. This tool allows interoperability of OB models during simulation, obviating some of the shortcomings of generic co-simulation limitations, such as a virtual testbed for building control. In spite of all the advantages offered by co-simulation, it still needs advanced user experience [37]. In the custom function approach, the user can write functions or custom code as an input file of the building energy model to overwrite existing or default controls [38]. As an example, EnergyPlus has an energy management system (EMS) feature to implement such functionality [38]. Gunay et al. (2016) implemented OB models for lighting, blind, occupancy, and clothing using the EMS feature of EnergyPlus [40]. This method has high flexibility by allowing users to change the building energy model in BPS without recompiling the BPS programs' source code [38].

As OCCs learn from human-building interactions, simulation-based OCC research must be accompanied by comprehensive occupant behaviour models, including both adaptive and non-adaptive behaviour. In 2021, Hobson et al. developed a library of OCC in R to provide a practical workflow to evaluate OCC in building simulation [41]. This approach was broken down into two main phases: 1) the offline learning phase and 2) the simulation phase. In the first phase, five different occupancy-centric and occupant behaviour-centric control metrics (e.g., presence or absence time) were used from measured Building Automation System (BAS) data of 29 private offices. To this end, R functions were developed to extract occupant-centric metric-related data from different types of archived BAS data. In the second phase, these occupant-centric metrics were integrated into EnergyPlus to test several OCC strategies and investigate their impact on energy use and thermal discomfort. Pang et al., (2020) used a hybrid simulation to quantify the energy-saving potential of occupant-centric HVAC controls in office buildings [11]. Multiple advanced OCCs for multi-zone VAV systems were integrated into the simulation. The EMS module of EnergyPlus was used to implement OCC strategies, and Python scripts were used to generate EMS programs with implemented OCC strategies. In another study in 2021, the U.S Department of Energy (DOE) Asset Scoring Tool and EnergyPlus were used to investigate the energy-saving potential in large hotels due to occupancy sensors and OCC implementation. This tool was used to evaluate the physical characteristics of the building regardless of occupancy and operational choices. The authors used EnergyPlus built-in EMS module to implement the OCC algorithms to read the real-time value of the indoor variables during execution [42]. Ouf et al. (2020) implemented different OCC algorithms and OB models using the EMS object of EnergyPlus to test and finetuning their settings. They highlighted the necessity of optimizing these control systems before field implementation [7].

Chapter 3: Methodology

This chapter is divided into two main subsections. First, it begins by presenting the proposed framework to perform multi-objective optimization of OCC performance using a simulation approach. The first subsection describes occupancy and stochastic occupant behaviour models used in the proposed framework. Subsequently, the method used to implement occupant behaviour models and OCC algorithms in the simulation framework is explained. Finally, it explains the genetic algorithm-based optimization model used to optimize the performance of OCC.

The second subsection provides a detailed description of the case study demonstrated as the proof of concept of the framework, starting with describing the office building model. Afterwards, the occupant-related parameters and OCC algorithms mainly used in the case study are described. Finally, more details about the sensitivity analysis method and applied optimization algorithm that were particularly used in the case study are provided.

3.1 Simulation-based framework

An overview of the proposed framework to optimize OCC performance is shown in Figure 2. In this framework, stochastic and dynamic occupancy and occupant behaviour (OB) models are integrated into the building simulation program. OCC algorithms are then implemented to learn simultaneously from occupant behaviours and their interactions with the building and control the building operations. Coupling the simulation model with a sensitivity analysis algorithm, the influence of each design variable and OCC configurational settings variable on building energy consumption and occupant comfort are identified. The influential variables are then selected and defined as decision variables for the optimization problem.

This framework conducts optimization for each type of occupant in two main steps. First, a multi-objective optimization algorithm determines the optimal design variables using the simulation model with the OB and OCC information. In the second step, the building is modelled using the optimal design alternative from the first step optimization, then the optimization algorithm identifies the optimal OCC configurational settings. In both steps, optimization aims to minimize building energy consumption while improving occupant comfort, which is represented by the number of occupant interactions with various systems. The proposed framework enables defining various occupant behaviour assumptions. Moreover, several OCC algorithms can be investigated using this framework, and their optimal performance can be identified while learning from different occupant preferences. It is worth mentioning that the sensitivity analysis and optimization process were coded to work in parallel computing mode to take advantage of multi-core processors rather than the traditional simulation-based framework.

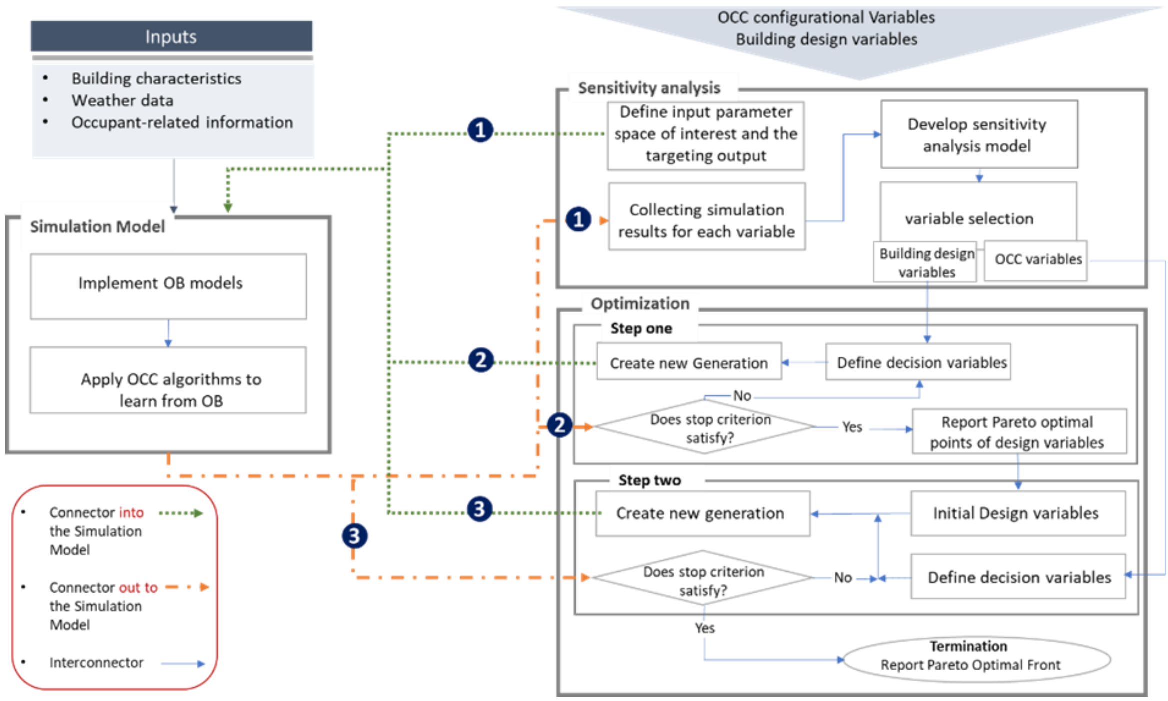


Figure 2: Overview of the proposed method

3.1.1 OB models

Four OB models were used to simulate stochastic occupant behaviour in this framework. Since more than 73% of total energy consumption in commercial buildings is attributed to space heating, cooling, and lighting [5], focusing on these categories can significantly reduce energy usage in this sector. However, other behaviours can be implemented in the framework for further studies to optimize building's operational energy consumption more realistically. This section summarizes these four models, including occupancy, lighting, heating and cooling setpoints, and blinds.

A probabilistic model to predict occupancy in offices was implemented based on the model proposed by [26]. At the beginning of each day, five event times, including arrival, two coffee breaks, lunch break, and departure, were sampled randomly from a pre-defined normal distribution (Table 3). The duration of coffee and lunch breaks was calculated through an exponential probability distribution with the time constant of 15 minutes for a coffee break and 1 hour for a lunch break. Office status was considered vacant during weekends and holidays.

Table 3: Occupancy events and time

Event	Time (Mean ± SD)
Arrival	9 AM ± 15 min
Coffee Break 1	10:30 AM ± 15 min
Lunch Break	12 ±15 min
Coffee Break 2	3 PM ± 15 min
Departure	5 PM ± 15 min

A probabilistic logistic regression equation was used to model occupant behaviour towards light and thermostat use (Equation (1)). These models were adapted from literature [43], [44]. However, the coefficients (β_0 and β_1) were modified to represent different occupant behaviours which are explained in more detail in Section 3.2.1. For light switch behaviour, this logistic regression model predicts the probability of switching lights on (P(x)) based on indoor illuminance (x) upon arrival and during the presence of occupant. The probability of light switching off upon departure was calculated based on the predicted length of absence. As the duration of expected absence increased, the probability of switching off the lights increased accordingly. Similarly, to predict the probability of increasing and decreasing the heating and cooling setpoints (P(x)), the indoor temperature in each time step (x) was used as the predictor as shown in Equation (1).

$$P(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{1}$$

Where P(x) is the probability of action happened by the occupant, β_0 and β_1 are the coefficients of the logistic regression model, and x is the predictor of the model.

Occupants' interaction with blinds was predicted using the model presented by [45]. Either lowering or raising blind action was predicted based on indoor illuminance and the current unshaded fraction. If an action was predicted, the probability of fully opening or closing the blind was calculated with a logistic regression model with solar radiation as the predictor. Alternatively, if a partial lowering or closing action was predicted, the related counterpart was drawn from the Weibull distribution. In the proposed framework, if the predicted partial for blind closing was more than 0.5, it was assumed that the blind was fully closed. Likewise, if it was predicted less than 0.5, the blind was assumed to be fully open.

3.1.2 Integrating OB models and OCC algorithms in building energy simulation

In this framework, the EMS (EnergyManagementSystem) object of EnergyPlus is used to implement occupant behaviour models and OCC algorithms in which occupant and OCC can affect the operation of building systems using EMS actuators at every time step. Indoor environmental sensors to measure indoor temperature, indoor illuminance, and solar radiation are defined as EMS sensors. The probabilistic occupancy and OB models are integrated into the simulation-based framework using custom scripts in EMS. Similarly, OCC algorithms are implemented using EMS and learned from OB in each time step (Figure 3). Occupants and OCC interact with building operations by changing the value of the actuators and consequently the value of pre-defined schedules for occupancy, lighting, blinds, and heating and cooling setpoints. More details on the implementing methods can be found in [40], [7].

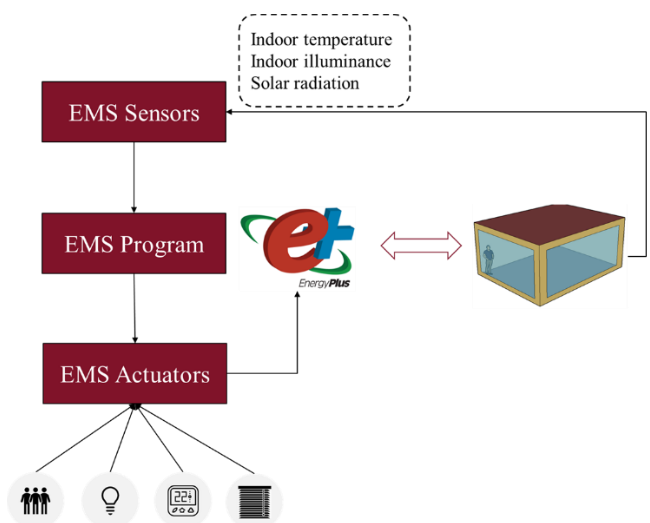


Figure 3: Exchanging data between OB and OCC algorithms and building energy model through EMS object of EnergyPlus.

3.1.3 Optimization

Two types of decision variables for the optimization problem are defined: building designs variables and OCC configurational variables. A sensitivity analysis is conducted before optimization to investigate the effects of each variable on energy consumption and occupant comfort. Based on the sensitivity analysis results, the influential variables are then selected as decision variables for the optimization problem.

The genetic algorithm (GA) optimization method is used to optimize OCC performance when learning from various occupant behaviours. The main objective of the optimization is to minimize annual energy consumption while maximizing occupant comfort (visual and thermal comfort). The objective functions are formulated as follows: EUI (Energy use intensity) and the annual number of thermostat keypress and light switch on actions by occupant as the proxy of occupant comfort. The NSGA-II algorithm (Non-dominated Sorting Genetic Algorithm) proposed by Deb et al. (2002) is used to identify the best trade-off solutions between EUI and comfort [46]. It is an evolutionary algorithm that uses the concept of elitism and non-dominated sorting to improve the convergence of multi-objective optimization problems. According to Wang (2016), NSGA-II is a mature algorithm for multi-objective optimization with the high capability of solving a wide range of problems with considerable complexity [47]. In this algorithm, once the population of the first generation is produced, the value of objective functions is evaluated and sorted based on non-domination ranking and crowd distance. Then, the offspring population is produced using genetic operators such as Cross over and Mutation. In the next step, parents and offspring population combined and sorted based on the best ranking. The best candidates are then used to produce the next generation and offspring population. Once the stop criteria are fulfilled, the algorithm stops and reports the Pareto front optimal points. Readers can refer to Deb et al. for more details on how the algorithm finds Pareto-optimal solutions.

A two-step optimization process is then defined. In the first step (step one), the multi-objective optimization algorithm is run, and only building design parameters are defined as the optimization decision variables. In this step, the simulation is performed with OCCs using the original configurational settings, and the optimization algorithm determines the optimal design variables. The goal is to optimize building design parameters based on the type of occupant. In the second step (step two), the optimal design variables are used, and the optimization algorithm is run using

OCC configurational settings as the optimization decision variables. In this step, the optimization algorithm determines the optimal OCC configurational settings while learning from different occupant types. Finally, the optimization algorithm identifies the optimal OCC configurational settings to minimize EUI and the number of occupant interactions with building systems. In this research, optimization is also conducted using all the optimization variables (building design variables and OCC configurational variables) in one step at once. The goal is to compare the performance of one-step and two-step optimization mathematically in this context.

3.2 Case study

To provide a proof-of-concept of the proposed framework, a single office with a floor area of 16 m2 (4m x 4m) and a height of 3 m was created in EnergyPlus (Figure 4) and simulated using the EnergyPlus weather data file (EPW) for Montreal. Internal heat gains from occupants were set to 130 W per person, which is based on the activity level of an occupant in the office. It was assumed that the office is occupied by one person. For the initial design, 35% WWR was installed on the south side of the building. The other three walls of the building and the roof and floor of the building were assumed to be adiabatic (i.e., attached to spaces with the same thermal condition). An interior shading device was defined as a blind, and a daylight reference point was located at the height of 0.8 m, almost equivalent to desk height. An ideal load air-based HVAC system was designed with a heating and cooling capacity of 2000 W. The initial selected design parameters were derived from previous studies, and they meet the requirements specified by NECB 2017 (Table 4).

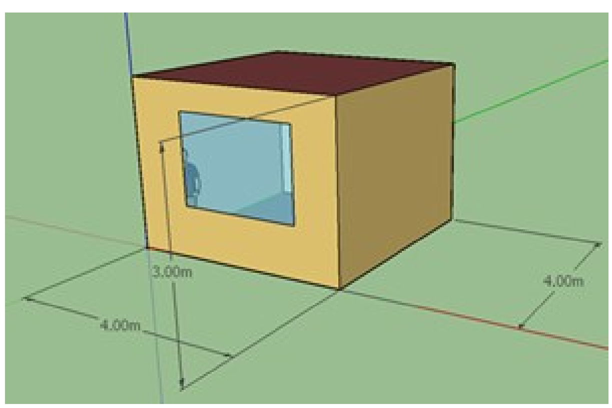


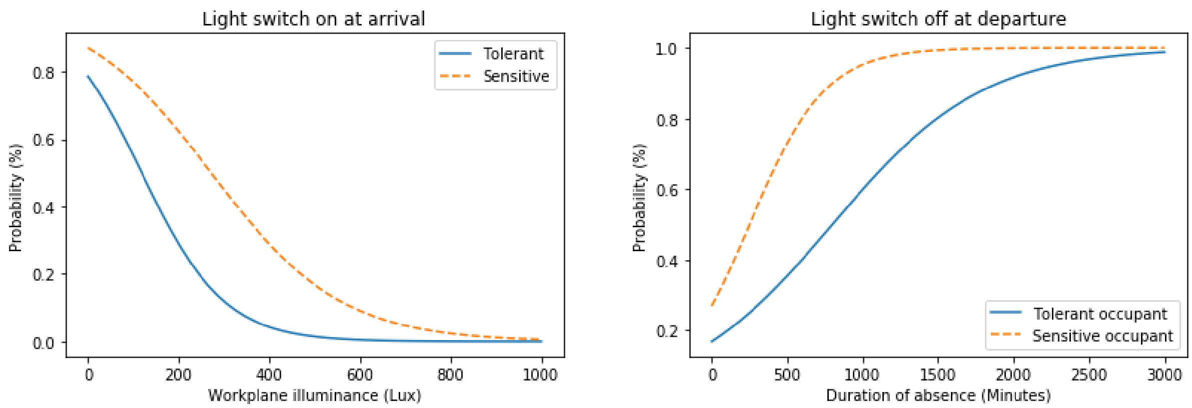
Figure 4: Single office building modeled as a case study.

Table 4: Initial design parameter for the office building model.

Design parameters	Values	Unit
Window glazing U-Factor	1.6	$W/(m^2K)$
Window glazing VT	0.6	-
Window glazing SHGC	0.3	-
Blind VT	0.2	-
Blind solar transmittance	0.05	-
Wall R-Value	3.07	$(m^2K)/W$

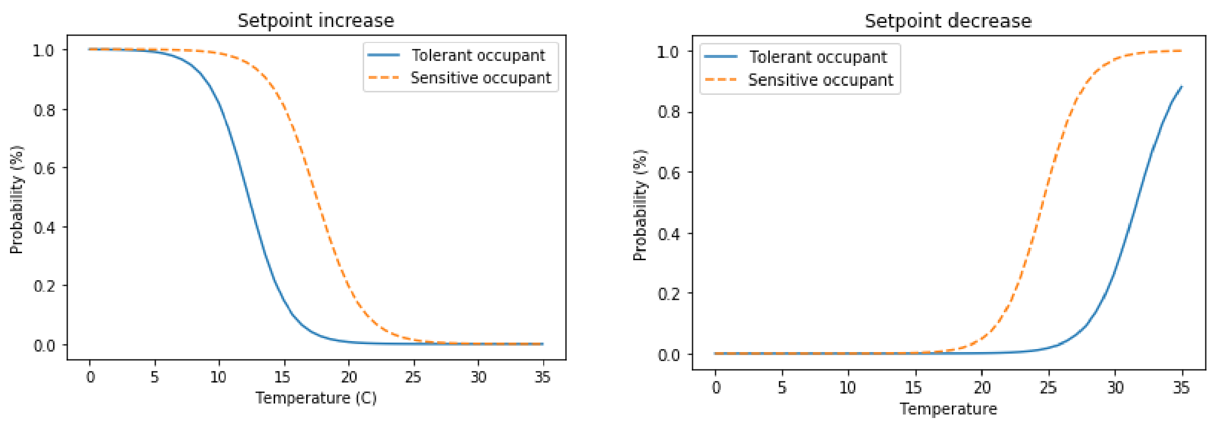
3.2.1 Occupant-related modifications

In this case study, two different types of occupants were defined to represent two extreme occupant behaviours, namely tolerant and sensitive. Overall, under the same environmental condition, sensitive occupant interacts with building operations more than tolerant occupant behaviour. In other words, tolerant occupant has more adaptability to the changes in the environmental condition. As mentioned in Section 3.1.1, the occupant behaviour models towards light and thermostat were developed based on a logistic regression equation. The authors modified the coefficients of these logistic regression models to define sensitive and tolerant occupant preferences. For example, for the lighting model, a sensitive occupant has a higher probability of interacting with the building than tolerant occupant with the same indoor illuminance (Figure 5). Similarly, the logistic regression coefficients of the implemented thermostat model were modified so that in summer, sensitive occupant has a higher probability of decreasing the cooling setpoint with the same indoor temperature than tolerant occupant. Likewise, in the winter, the probability of increasing the heating setpoint is greater for the sensitive occupant than the tolerant occupant type given the same indoor temperature (Figure 6).



(a) (b)

Figure 5: The probability of light a) switches on, b) switches off for two types of occupant.



(a) (b)

Figure 6: The probability of setpoint a) increase, b) decrease for two types of occupants.

3.2.2 OCCs

In this study, three OCCs for controlling lights, heating, and cooling setpoints were investigated from the literature. In the following, the way that each of these OCCs works is explained.

3.2.2.1 Lighting OCC #1

The first implemented OCC (lighting OCC #1) works with a threshold for light switching off [32]. This OCC switches lights off either when the indoor illuminance exceeds this threshold or in the occupant's absence. This threshold decreased gradually until an occupant light switch on action was observed, at which point the threshold slightly increased. The rate of increasing or decreasing the threshold relied on quadratic deviation (Equations 2-7):

$$L_{(Th)Off} = \frac{(-D_1 - 2)}{D_2} \tag{2}$$

$$P_{occ} = \frac{1}{1 + e^{-(D_1 + D_2 * Lux)}} \tag{3}$$

Where $L_{(Th)Off}$ is a light switch off threshold and P_{occ} is the probability of light switch on by occupant, which is predicted by OCC based on a logistic regression probability with indoor illuminance (Lux) as a predictor and D_1 and D_2 are the coefficients of the logistic regression.

If the light switch on action by occupant is not observed in 30 minutes, these two coefficients get updated based on the following formula (4-5), which causes a slight increase in the threshold for light switch off:

$$D_1{}' = D_1 - ((P_{occ}{}^2)(1 - P_{occ}{}^2)(LR)) \tag{4}$$

$$D_2{}' = D_2 - ((P_{occ}{}^2)(1 - P_{occ}{}^2)(LR)(Lux) \tag{5}$$

Where $D_1{}'$and $D_2{}'$ are the new values of the logistic regression coefficients, LR is the learning rate, which is defined as 0.001 in the original configuration, and Lux is indoor illuminance in foot-candles. The initial value of D_1 and D_2were 1 and -0.01, respectively.

Once occupant switches the light on, these two thresholds get updated based on the following equations (6-7)

$$D_1{}' = D_1 + ((P_{occ})(1 - P_{occ})^2(LR)) \tag{6}$$

$$D_2{}' = D_2 + ((P_{occ})(1 - P_{occ})^2(LR)(\mathrm{Lux})) \tag{7}$$

In the original configuration of this OCC, the light switches off threshold can only decrease to 200 lux and increase to 500 lux, which becomes constant afterward

3.2.2.2 Lighting OCC #2

The second implemented OCC (lighting OCC #2) works with a threshold for switching lights on and off [48]. The switching on threshold is derived dynamically based on the average of indoor illuminance when the occupant turns on the light (Equation 8).

$$L_{(Th)ON} = \frac{\sum_{n=1}^{N} I_n}{N} \tag{8}$$

Where $L_{(Th)ON}$ is a light switch on threshold, I_n is indoor illuminance when occupant switches the light on, and N is the number of times that occupant switches the light on within one month.

This average is taken every month, which means that the threshold gets updated monthly. The light switching off threshold is calculated by adding the light's illuminance in the room at night (e.g. 450 Lux) to the threshold for Light switch on.

3.2.2.3 Thermostat OCC

The implemented thermostat OCC works with heating setpoint in the winter and cooling setpoint in the summer [49]. The heating setpoint decreases gradually till it is interrupted by occupant thermostat keypress action, after which it increases slightly. Similarly, the cooling setpoint increases gradually as long as no interaction with thermostats by the occupant is observed. Once the occupant adjusts the thermostat setpoint, the cooling setpoint decreases slightly. The rate of increasing or decreasing setpoints relied on a quadratic deviation (Equations 9- 14):

$$H_{SP} = \frac{(-a_H - 2)}{b_H} \tag{9}$$

$$P_{occc} = \frac{1}{1 + e^{-(a_H + b_H * T_{in})}} \tag{10}$$

Where H_{SP} is the heating setpoint determined by OCC, and P_{occ} is the probability of increasing heating setpoint by occupant. This probability is predicted by OCC based on a logistic regression probability with indoor temperature (T_{in}) as a predictor and a_H and b_H are the coefficients of the logistic regression.

In the winter, if occupant does not increase the HSP, these two thresholds get updated every 30 minutes based on the following formulas, which causes a slight decrease in the HSP:

$$a_H{}' = a_H - ((P_{occ}{}^2)(1 - P_{occ}{}^2)(LR)) \tag{11}$$

$$b_H{}' = b_H - ((P_{occ}{}^2)(1 - P_{occ}{}^2)(LR)(T_{in}) \tag{12}$$

Where $a_H{}'$and $b_H{}'$ are the new values of the logistic regression coefficients, LR is the learning rate which is defined as 0.001 in the original configuration of this OCC, and T_{in} is indoor temperature. The initial value of a_H and b_H were 20 and -1, respectively.

If the occupant increases the HSP, these two thresholds get updated using the following equations:

$$a_H{}' = a_H + ((P_{occ})(1 - P_{occ})^2(LR)) \tag{13}$$

$$b_H{}' = b_H + ((P_{occ})(1 - P_{occ})^2(LR)(\,T_{in})) \tag{14}$$

Similarly, for the cooling setpoint in the summer:

$$C_{SP} = \frac{(-a_C - 2)}{b_C} \tag{15}$$

$$P_{occ} = \frac{1}{1 + e^{-(a_C + b_C * T_{in})}} \tag{16}$$

Where C_{SP} is the cooling setpoint which is defined by OCC, and P_{occ} is the probability of reducing cooling setpoint by the occupant, which is predicted by OCC based on a logistic regression model with indoor temperature (T_{in}) as a predictor and a_C and b_C are the coefficients of the logistic regression. In the summer, if occupant does not decrease the CSP, these two thresholds get updated every 30 minutes using the following equation, which results in a slight increase in the CSP:

$$a_C' = a_C - ((P_{occ}{}^2)(1 - P_{occ}{}^2)(LR)) \tag{17}$$

$$b_C' = b_C - ((P_{occ}{}^2)(1 - P_{occ}{}^2)(LR)(T_{in}) \tag{18}$$

In which a_C' and b_C' are the new values of the logistic regression coefficients, LR is the learning rate which is defined as 0.001 in the original configuration of this OCC, and T_{in} is indoor temperature. The initial value of a_C and b_C were -25 and 1, respectively.

Once occupant decreases the heating setpoint, these two coefficients are updated based on the following equations:

$$a_C' = a_C + ((P_{occ})(1 - P_{occ})^2(LR)) \tag{19}$$

$$b_C{}' = b_C + ((P_{occ})(1 - P_{occ})^2(LR)(\,T_{in})) \quad (20)$$

The original configuration of this OCC has the conservative boundaries of [20,22] ℃ for heating setpoint and [22, 25] ℃ for cooling setpoint. In other words, if the setpoints exceed these boundaries, it becomes constant at the boundaries' point till it backs to the defined range.

3.2.3 Sensitivity analysis

In this case study, a sensitivity analysis was conducted to identify the most influential variables on EUI and the number of occupant interactions with building systems. Overall, two main scenarios for sensitivity analysis were defined: 1- variables selection for the most influential parameters on EUI by considering tolerant behaviour scenario, 2- variable selection based on the most influential variables on the number of occupant interactions regarding sensitive behaviour scenario. Since sensitive occupant interacts with building systems more frequently, this type of occupant is more appropriate to show how the number of interactions is affected by the changes in defined variables. Sensitivity analysis was conducted separately for the three OCCs (Light #1, Light #2, and Thermostat). Therefore, a total of 6 cases for sensitivity analysis were defined. To quantify the effect of changes in each variable on the target variable (EUI or comfort), each variable changed in the range with the defined increment in Table 5 and Table 6, while other variables were kept constant in their original values.

To investigate the results of sensitivity analysis, the slope of $\frac{\partial y_i}{\partial x_i}$ is calculated using the finite differences method, and then the mean (μ) and standard deviation (SD) were calculated. ∂y_i indicates the change in target variable to the corresponding change in variable x. In this method, a larger value of μ implies that the variable has a larger overall influence on the target variable. The larger value of SD indicates that the influence of that variable is highly nonlinear. This method allows a qualitative assessment of the importance of each variable while dealing with nonlinear responses. The sensitivity analysis results were normalized before calculating μ and SD to make the variables comparable as their scales were not the same.

Table 5: Range of design variables for the three OCCs.

Variable	Unit	Min	Max	Increment	OCC
VT (glazing)	-	0.3	0.8	0.05	Lighting
Wall visible reflectance	-	0	1	0.1	Lighting
Roof visible reflectance	-	0	1	0.1	Lighting
Floor visible reflectance	-	0	1	0.1	Lighting
North axis	Degree	0	270	90	Lighting, Thermostat
Blinds visible transmittance	-	0.05	0.2	0.05	Lighting
WWR	-	20	70	5	Lighting, Thermostat
U-Factor (glazing)	W/m2K	1.4	2.2	0.1	Thermostat
SHGC	-	0.3	0.6	0.05	Thermostat
WWR	-	20	70	5	Lighting, Thermostat
Blind solar transmittance	-	0.05	0.2	0.05	Thermostat
Wall R-value	$m^2K/_W$	4	11	1	Thermostat

Table 6: Range of OCC configurational settings variables.

variables	OCC	Min	Max	Unit	Increment
Upper boundary for heating setpoint	Thermostat	22	30	℃	0.5
Lower boundary for heating setpoint	Thermostat	15	20	℃	0.5
Upper boundary for cooling setpoint	Thermostat	12	30	℃	0.5
Lower boundary for cooling setpoint	Thermostat	15	20	℃	0.5
Learning rate	Thermostat	0.0005	0.01	-	0.0005
Time interval	Thermostat	10	240	Minute	-
Upper boundary for light switches off threshold	Lighting #1	300	700	Lux	20
Lower boundary for light switches off threshold	Lighting #1	0	300	Lux	20
Learning rate	Lighting #1	0.0005	0.1	-	0.0005
Time interval	Lighting #1	10	240	Minute	-
Period of updating the threshold	Lighting# 2	1	120	Day	1
Threshold for the first period	Lighting# 2	0	300	Lux	20

3.2.4 Optimization

In this case study, four optimization cases were defined (Table 7). The GA was used to optimize the performance of OCCs, and optimization was conducted using Pymoo Python packages. In addition, a number of Python packages such as EPPY and GEOMEPPY were used as an interface to modify (EMS) object's scripts in EnergyPlus through Python programming language to change the decision variables of the optimization problem in the simulation environment. GA parameters such as population size, mutation probability, crossover probability, elite ratio, which are a function of problem space and available computational capacity, are shown in Table 8. These parameters were selected based on a practical approach by evaluating the optimum result improvement and the computational time. For this study, the maximum number of iterations (40) is considered for termination criteria.

Table 7: The list of optimization cases

	OCC thermostat and lighting #1		OCC thermostat and lighting #2	
Occupant type	Tolerant	Sensitive	Tolerant	Sensitive
Case #	1	2	3	4

Table 8: GA optimization parameters

Optimization parameters	Value
Population size	15
Cross over probability	0.7
Mutation probability	0.35
Elite ratio	0.05
Maximum number of generations	40

The main objective of optimization was to minimize EUI while maximizing occupant comfort. A two-objective and a three-objective optimization problems were defined. For the former, two objective functions were formulated: the first one was annual energy consumption per square meter (EUI), and the second objective function was defined as the total annual number of thermostat keypresses and light switches by the occupant (used as a comfort proxy). Since the annual number of light switches is naturally much larger than the number of thermostat keypresses, the second objective function was normalized. To this end, the number of interactions was divided by the maximum observed number of interactions. For two-objective optimization, it was assumed that there is no difference between interaction with light switch and interaction with thermostat keypress in terms of the level of comfort. As a result, the goal is to minimize the number of interactions in total (summation of interactions with light switch and thermostat keypress) regardless of the type of interaction. As to three-objective optimization, three objective functions were defined as follows: 1- EUI, 2- Annual number of light switches, 3- Annual number of thermostat keypresses. The logic behind three-optimization was to distinguish between two types of interactions with light switches and thermostat keypress from the point of comfort view.

As the simulation relied on stochastic models, EUI and the number of occupant interactions with building systems was different after each run with a set of identical variables. Therefore, simulations had to be repeated multiple times to get an average of EUI and the number of interactions. To this end, the initial model was run 200 times, and the expanding mean for EUI and the number of interactions were calculated. It was found that after 32 runs, the changes in expanding mean for EUI were less than 2%. The change in expanding mean for the number of interactions by sensitive occupant with light, setpoint increase, and setpoint decrease were less than 2% after 25, 32, and 110 runs, respectively (Figure 7). Therefore, to calculate the objective functions for each set of candidate parameters, the model was executed 110 times, and the average value of target variables was calculated over 100 runs using a high-performance computer (HPC) with multi-core processing. Parallel processing allows to break down a complicated, time-consuming problem into different parts and distribute them among a group of cores. Each core executes the assigned task independently. At the end of the process, the final results are pooled. In this study, a 32-core processer was used, which significantly lowered the computational time required for executing the simulation-based multi-objective optimization procedure.

As mentioned in Section 3.1.3, optimization was conducted using two different steps for both two-objective and three-objective optimization. In the following, each step is explained in more detail.

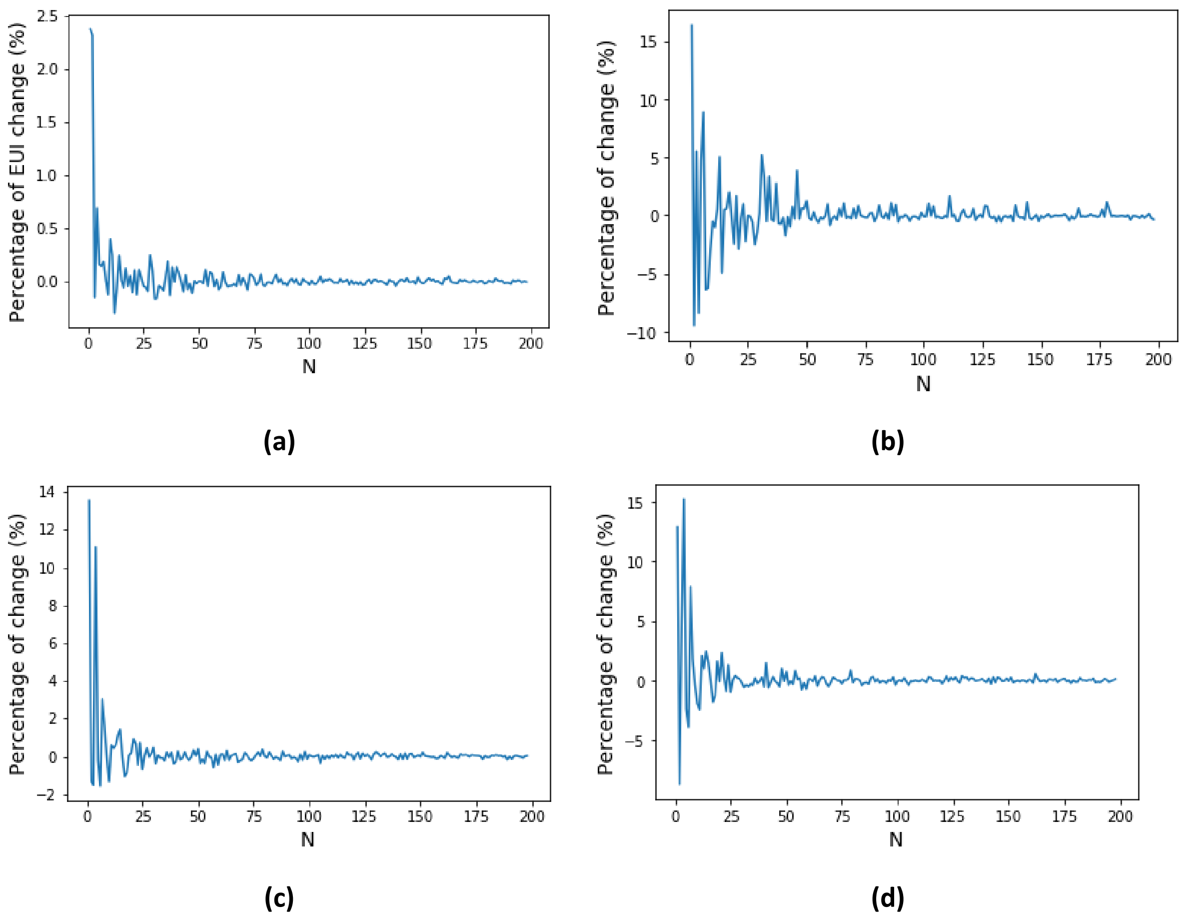


Figure 7: Change in expanding mean over 200 runs for a) EUI, b) light switches on, c) setpoint increase, and d) setpoint decrease.

3.2.4.1 Optimization step one

Two types of decision variables, specifically building design variables and OCC configurational settings variables, were defined. In the optimization step one, optimization was conducted using building design variables as the optimization decision variables. In this step, simulation was run while OCC configurational settings were set to their original values. Table 5 indicates the solution space of design variables for three investigated OCCs. These ranges of design parameters were selected in a way that meets the requirements by NECB 2017.

3.2.4.2 Optimization step two

In optimization step two, the optimal points obtained from optimization step one were used as the value of building design parameters, and the optimization was conducted using OCC configurational variables as the optimization variables. The decision variables for three investigated OCCs were formulated as follows:

Regarding the original configuration of lighting OCC #1, the light switch of threshold can only decrease to 200 lux and increase to 500 lux, which becomes constant afterward. In this study, the boundaries of (0, 300) and (300, 700) with the increment of 20 Lux were considered for minimum and maximum limits for threshold, respectively, to identify the optimized constraints for OCC configuration based on each occupant type. Moreover, a boundary of (0.0005, 0.01) was considered as an optimization boundary for learning rate, which affects the intensity of the increasing and decreasing rate. The time interval when OCC increases the threshold is every 30 minutes in the original configuration. The optimization problem defined the following time intervals to find the optimal time span for this OCC configuration: 10, 20, 30, 60, 90, 120, 150, 180, 210 and 240 minutes.

The decision variables for lighting OCC #2 were considered as follows: boundaries of (1,120) days are considered for the duration which the average take over for calculating light switch on threshold to investigate different duration from one day to almost one season. This average is taken every one month in the original configuration. In addition, a boundary of (0, 300) Lux was defined as a boundary for the initial value of the threshold when OCC still does not learn from occupant behaviour.

For thermostat OCC, a range of [15, 20] for the lower boundary of the heating setpoint and [22, 30] for the upper boundary of the heating setpoint were considered. Likewise, a range of [15, 20] for the lower boundary and [25, 30] for the upper boundary of the cooling setpoint were considered to estimate the optimal boundaries. In addition, a boundary of (0.0005, 0.01) was defined for learning rate, which affects the intensity of the increasing and decreasing rate. As mentioned earlier, the time interval when OCC increases the cooling setpoint and decreases the heating setpoint is every 30 minutes in the original configuration. The solution space for this variable was defined as 10, 20, 30, 60, 90, 120, 150, 180, 210, and 240 minutes for optimization problems to determine the best time interval for each optimization case.

Table 6 shows the solution space of OCC configurational variables defined for the optimization problems. Overall, the solution spaces of 1.43×10^{21} for the combination of OCC thermostat and OCC light #1, and 74.02×10^{19} for the combination of OCC thermostat and OCC light #2 were defined.

Chapter 4: Results and discussion

This section presents the results of the case study, including the sensitivity analysis to investigate the effect of design variables and OCC configurational settings on annual energy consumption and the annual number of occupant interactions. This is followed by presenting the results of multi-objective optimization to minimize EUI and the number of occupant interactions for both tolerant and sensitive occupant scenarios. Finally, the major findings for each OCCs are discussed in the last subsection.

4.1 Sensitivity analysis results

Figure *8* depicts the results of sensitivity analysis for OCC thermostat in which EUI was defined as the dependent variable. Overall, there are two main areas in the plot which show the influential and linear variables (under the line of μ=SD), and influential but nonlinear (above the line plot for μ=SD). As the values of μ get closer to 0, the less influential the variables are. Figure 9 illustrates the sensitivity analysis results for OCC thermostat, considering the number of occupant interactions as the target variable. The analysis indicated all the variables, including OCC variables and design parameters, were influential on EUI and the number of occupant interactions.

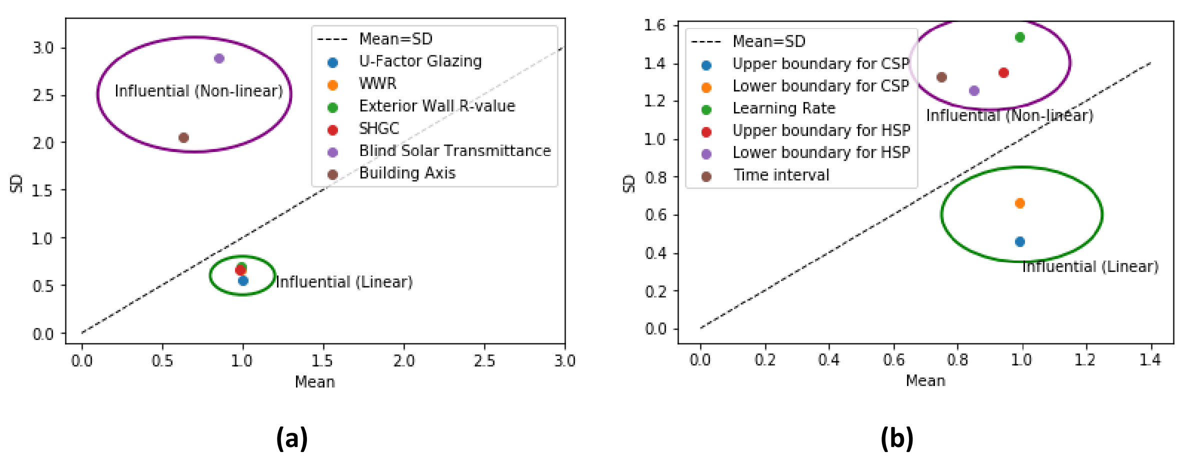
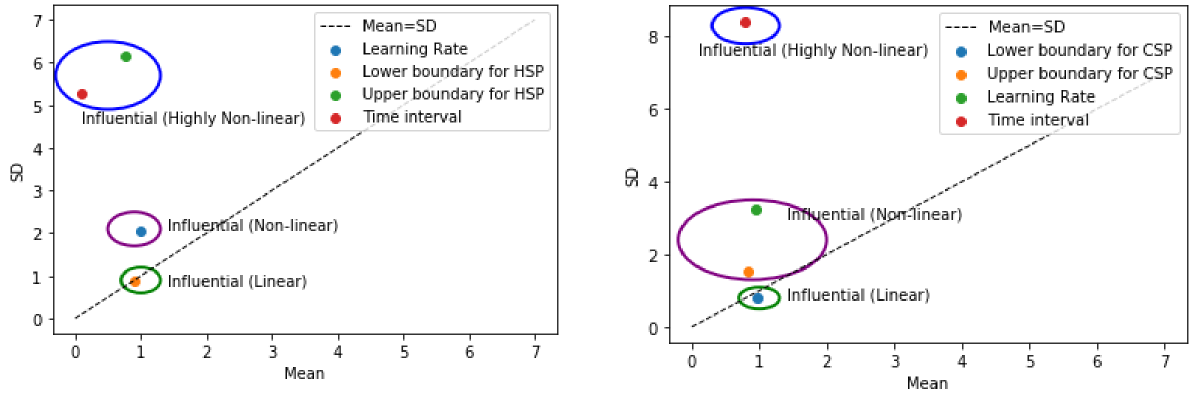


Figure 8: Sensitivity analysis results for OCC thermostat in which EUI defined as the dependent variable a) design variables, b) OCC configurational settings variables



(a) **(b)**

Figure 9: Sensitivity analysis results for OCC thermostat configurational settings variables for the number of a) SP increase, b) SP decrease.

According to the sensitivity analysis results for thermostat OCC using building design variables (see Appendix), glazing U-factor, WWR, exterior wall R-value and solar heat gain coefficient showed strong highly linear effects on EUI. Building axis, however, had a highly nonlinear but influential effect on EUI. Among OCC configurational settings variables for OCC thermostat, upper boundary, and lower boundary for CSP had a strong linear impact on EUI, while the other four OCC variables were recognized as influential but nonlinear parameters. Among OCC variables, learning rate was found to be a highly influential variable for both EUI (nonlinear) and the number of interactions (almost linear) for this OCC.

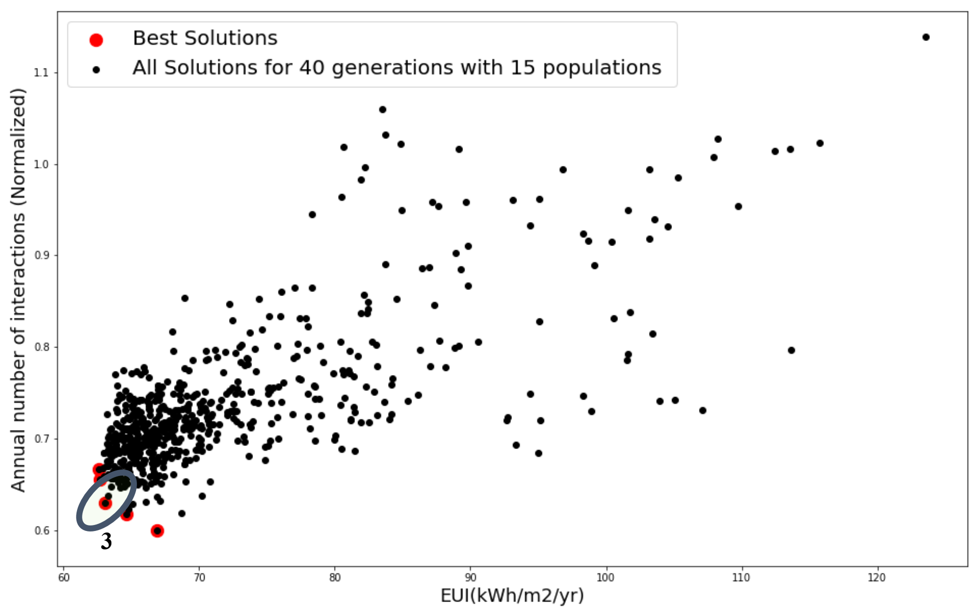
A similar sensitivity analysis was conducted for OCC lighting #1 and OCC lighting #2 (results can be found in the Appendix). Based on the results, among design variables, ceiling visible absorptance, blind visible transmittance, WWR, and wall visible absorptance were identified as influential linear variables in changing EUI for both OCC light #1 and light #2. All the investigated OCC configurational settings variables for both OCC light #1 and light #2 were found as influential variables for EUI and the number of interactions. Overall, all the investigated OCC variables were included in the multi-objective optimization problem since they showed considerable effects on at last one of the objective functions based on the sensitivity analysis results.

4.2 Multi-objective optimization results

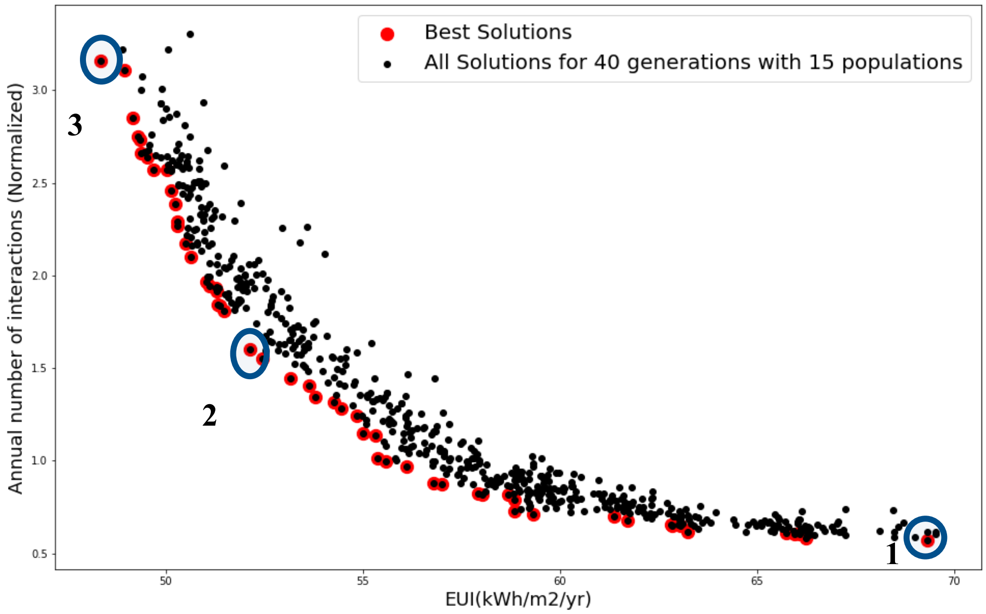
This section shows the results of multi-objective optimization to identify the optimal Pareto-Front alternatives for design parameters and OCC configurational settings variables. Optimization was conducted for four cases shown in Table 7. For brevity, the results of optimization for case #1 are presented and the results of optimization for the other three cases can be found in Appendix. In the following, the results of two-objective optimization and three-objective optimization are presented respectively.

4.2.1 Two-objective optimization results

By conducting step one optimization, in which design variables were defined as the optimization decision variables, the algorithm introduced 5 Pareto-optimal alternatives (Figure 10_a). Point #3 was selected as the best trade-off between the two objectives. In the second step, the optimization was performed using the optimal design variables of the selected best solution (Point #3) to find the Pareto-optimal alternatives for OCC configurational settings. The result of optimization for step two is shown in Figure 10-b.



(a)



(b)

Figure 10: Results of multi-objective optimization for case #1: a) Optimization-step one, b) Optimization-Step two

Optimization was also conducted using a one-step optimization problem explained in Section 3.1.3. Figure 11 shows a comparison between the results of one-step optimization versus two-step optimization. It can be seen how the two-step optimization method was capable of further minimizing the defined objective functions than one-step optimization. These two steps of optimization were also performed for the other three cases shown in Table 7. The values of optimal design parameters introduced by the first optimization step for the four optimization cases are shown in Table 9.

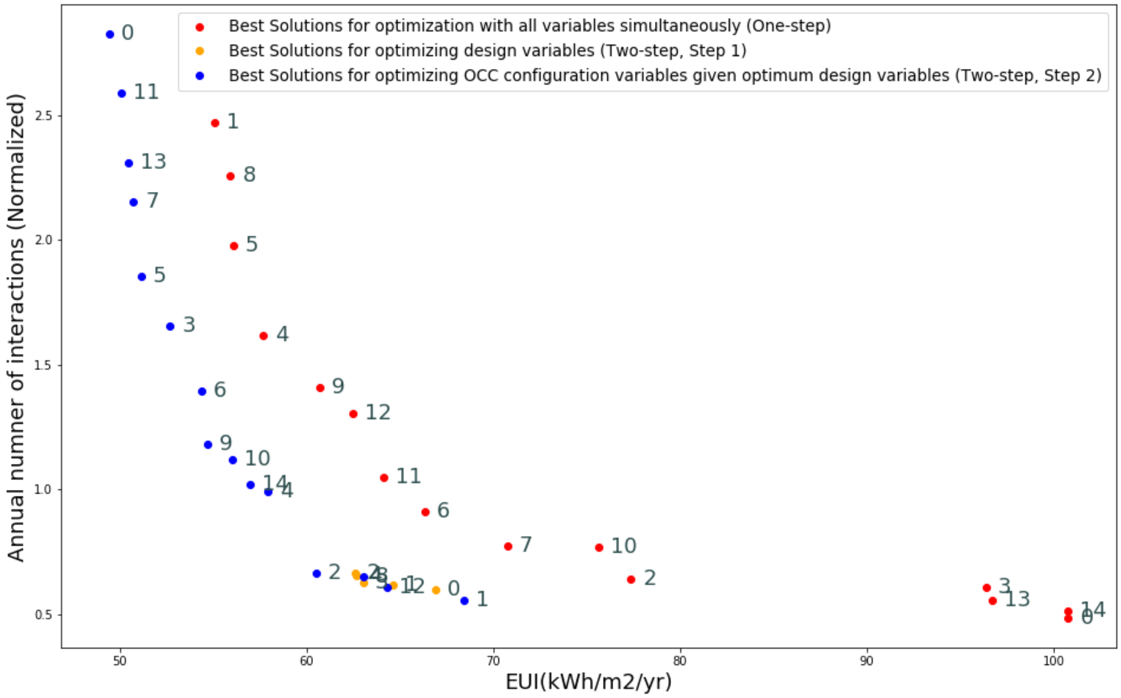


Figure 11: Comparison of one-step optimization with the two-step optimization method.

Table 9: The building design value of selected Pareto-optimal alternatives for all Optimization cases.

Case #	U-Factor (glazing)	SHGC	North axis	WWR	Blind ST	Wall R-value	Window VT	Wall visible reflectance	Ceiling visible Absorptance	Floor visible Absorptance	Blind VT
1	1.4	0.6	0	0.2	0.15	11	0.75	0.0	0.0	0	0.2
2	1.4	0.45	0	0.2	0.15	11	0.8	0.0	0.0	0	0.2
3	1.4	0.6	0	0.2	0.05	10	0.7	0.0	0.0	0.3	0.2
4	1.4	0.45	0	0.2	0.2	10	0.8	0.0	0.1	0	0.15

In analyzing the optimization results, three types of Pareto optimal points (P.O.P) were selected (Figure 10), where P.O.P #1 has the highest EUI but the lowest number of interactions (comfort-side), P.O.P #3 has the lowest EUI but the highest number of interactions (energy-side), and P.O.P #2 as the best trade-off between two objectives (balanced-solution). The values of these OCC configurational alternatives for optimization case #1 and case #2 are shown in Table 10, while case #3 and case #4 are shown in Table 11.

Table 10: Value of selected Pareto-optimal alternatives for optimization case #1 and case #2.

OCC Thermostat configurations							
OCC Variable	**Unit**	**Pareto-optimal solutions**					
		(Case #1_Tolerant)			**(Case #2_Sensitive)**		
		P.O.P #1	P.O.P #2	P.O.P #3	P.O.P #1	P.O.P #2	P.O.P #3
Learning rate	-	0.002	0.0085	0.008	0.0005	0.0075	0.0085
Time interval	min	60	10	30	240	30	30
Maximum amount for threshold (Heating)	℃	26	24	24.5	27	26	26.5
Minimum amount for threshold (Heating)	℃	17	16	15	19	17	16.5
Maximum amount for threshold (Cooling)	℃	27	29	30	25	25.5	30
Minimum amount for threshold (Cooling)	℃	17	17.5	18	16	17	18
OCC Light #1 configurations							
OCC Variable	**Unit**	**Pareto-optimal solutions**					
		(Case #1_Tolerant)			**(Case #2_Sensitive)**		
		P.O.P #1	P.O.P #2	P.O.P #3	P.O.P #1	P.O.P #2	P.O.P #3
Learning rate	-	0.004	0.004	0.004	0.005	0.0055	0.0055
Time interval	min	150	10	10	180	30	30
Upper boundary for the light threshold	Lux	340	520	340	440	480	540
Lower boundary for the light threshold	Lux	180	40	40	200	280	0

Table 11: Value of selected Pareto-optimal alternatives for optimization case #3 and case #4.

OCC Thermostat configurations							
OCC Variable	**Unit**	**Pareto-optimal solutions**					
		(Case #3_Tolerant)			**(Case #4_Sensitive)**		
		P.O.P #1	P.O.P #2	P.O.P #3	P.O.P #1	P.O.P #2	P.O.P #3
Learning rate	-	0.005	0.0085	0.009	0.0005	0.0085	0.009
Time interval	min	30	20	20	30	30	20
Maximum amount for threshold (Heating)	℃	26.5	25	24.5	27.5	26.5	26
Minimum amount for threshold (Heating)	℃	16.5	16	15	18.5	17	16.5
Maximum amount for threshold (Cooling)	℃	25.5	27	29.5	24	25	25.5
Minimum amount for threshold (Cooling)	℃	18	18.5	20	17	17	18
OCC Light #2 configurations							
OCC Variable	**Unit**	**Pareto-optimal solutions**					

		(Case #3_Tolerant)			(Case #4_Sensitive)		
		P.O.P #1	P.O.P #2	P.O.P #3	P.O.P #1	P.O.P #2	P.O.P #3
Period of updating the threshold	Day	4	60	118	1	6	87
Threshold for the first period	Lux	160	60	0	280	280	260

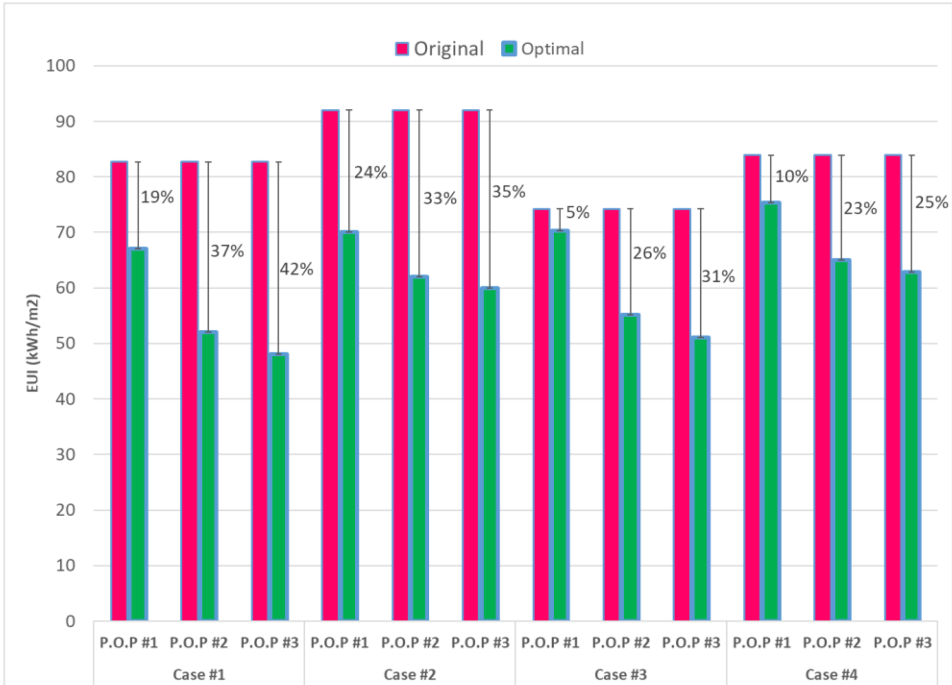
Baseline scenarios for all four optimization cases were defined when they worked using initial design parameters and original values of OCC configurational settings (Table 12). The baseline scenarios provided an environment where different OCCs can be compared under similar conditions. Based on the results, before optimization, lighting OCC#1 used 12.77 kWh/m2/yr of electricity more than lighting OCC #2. However, the difference between EUI was 8.52 kWh/m2/yr. As lighting OCC #1 used more electricity for lighting energy than lighting OCC #2, thermostat OCC needed less heating energy when working with the combination of this OCC than lighting OCC #2. In other words, a small part of the need for heating energy was produced by lights. Figure 12 indicates the percentage of changes in the optimal value of the two objective functions relative to the baseline scenarios for the three selected Pareto-optimal alternatives for each of the optimization cases. As mentioned above, the results of baseline scenarios revealed that lighting OCC #2 showed a better performance in terms of reducing light electricity consumption. The annual electricity consumption for OCC lighting #1 was more than twice that of OCC lighting #2 for both occupant types. Furthermore, the type of occupant affected the performance of OCC light #1 more than OCC light #2 as the range of differences in EUI, and the number of interactions were more significant for OCC light #1 than OCC light #2 by changing the type of occupant. However, after optimizing the performance of these two OCCs, the results indicated that OCC lighting# 1 had a better performance in terms of energy consumption than OCC lighting #2, which highlighted the necessity of customizing OCC configurations based on the type of occupant.

Generally, the optimization algorithm was able to reduce both objective functions for all optimization cases to a great extent. The optimization results for the combination of OCC light #1 and OCC thermostat showed that the optimization algorithm could minimize EUI by 42% and 35% for tolerant and sensitive occupants, respectively. It also introduced optimal alternatives that improve occupant comfort by reducing the number of interactions by 50% and 42% for tolerant and sensitive occupants, respectively. Likewise, using the optimal configurational setting for OCC light #2 and OCC thermostat, energy usage was reduced by 31% and 25% for the tolerant and sensitive occupant, respectively.

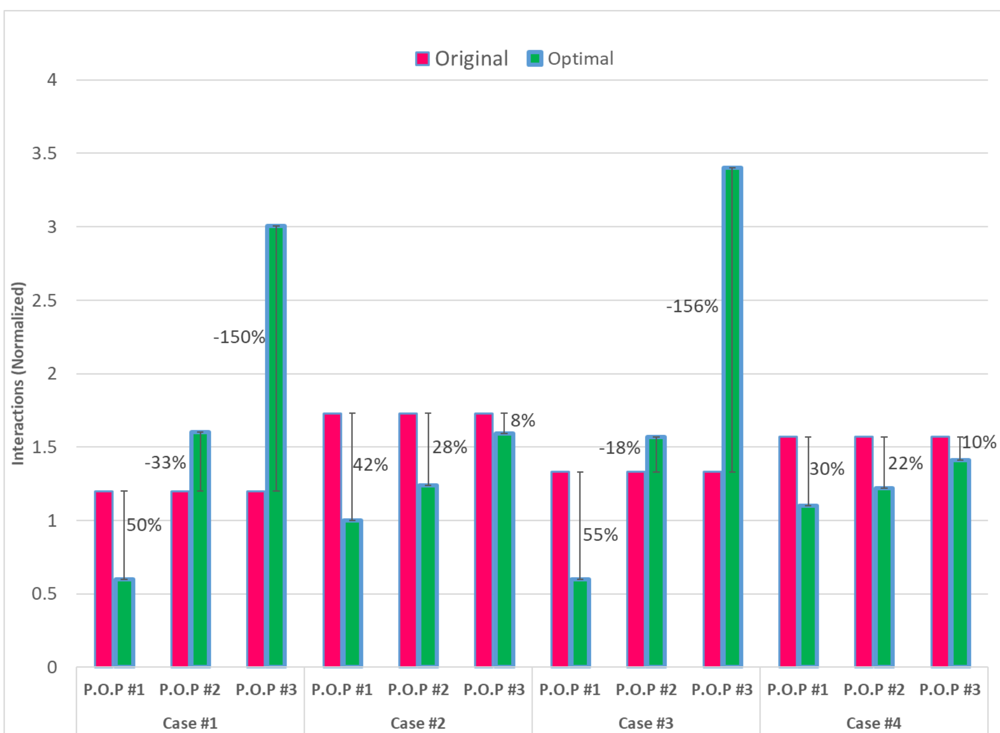
A number of alternatives can improve occupant comfort to 55% and 30% for tolerant and sensitive occupant types, respectively. These three types of optimal solutions provide various options for each type of occupant. Based on the priority of the occupant, which can be either energy-saving priority or comfort priority, the OCC configuration can be customized. As an example, if the priority of a sensitive occupant is energy-saving than comfort, the optimization algorithm proposed a set of optimal configurations that can reduce energy consumption to 35%. However, occupant comfort yet can be improved to 8%. On the other hand, a sensitive occupant with a comfort priority can reach a 42% increase in comfort while reducing energy consumption to 24%. In contrast, tolerant occupant with energy priority allowed to increase the number of interactions by about 1.5 times more than the baseline scenario to reach about 40% energy-saving. It does not happen for sensitive occupant which is aligned with the fact that the level of tolerance is higher for tolerant occupant than sensitive occupant. In other words, for sensitive occupant, there is no trade-off between energy-saving and interactions if they reach this high number of interactions (e.g., 1.5 times more than baseline). It would happen in the points that there is no energy-saving in exchange for lowering comfort to this extent.

Table 12: The results of baseline scenarios for all four cases.

Case #	**1**	**2**	**3**	**4**
Average light electricity use (kW/m2)	20.31	21.50	7.54	8.54
Average annual number of interactions (Normalized)	1.2	1.73	1.33	1.57
Average of EUI (kWh/m2/yr)	82.72	92.02	74.20	83.92



(a)

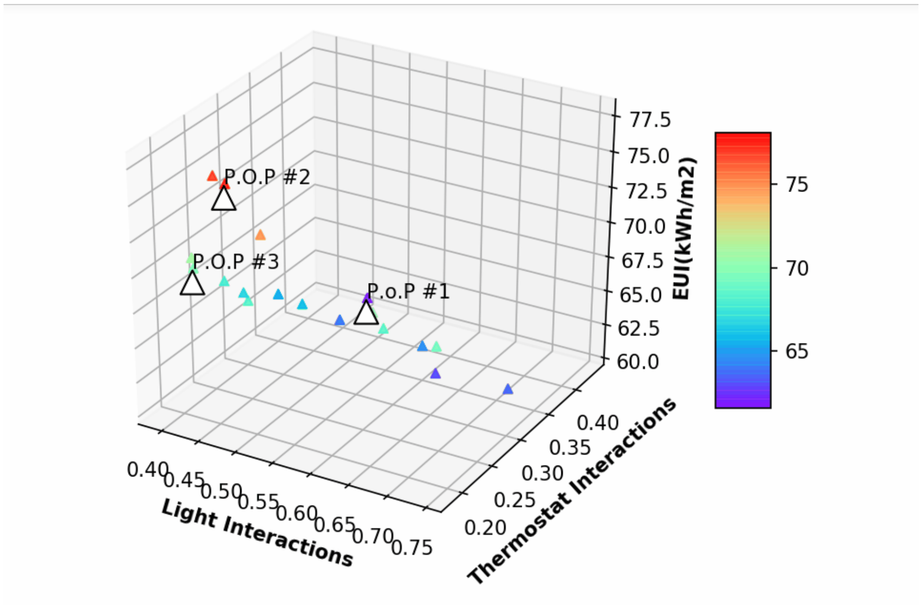


(b)

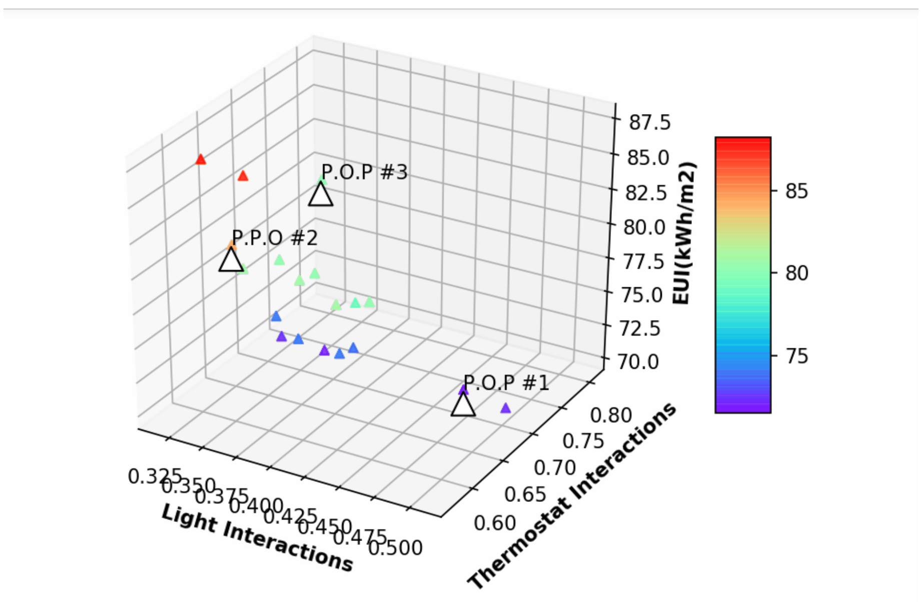
Figure 12: Percentage of changes in the optimal values of a) EUI, and b) Interactions, relative to the baseline scenarios.

4.2.2 Three-objective optimization results

The results of optimization for the three-objective optimization problem for tolerant and sensitive occupants are shown in Figure 13. Each axis is dedicated to each defined objective function. The rainbow color bar shows the intensity of EUI in order of the colors, from blue corresponding to the lowest to red corresponding to the highest EUI. Three-optimization presented several optimal points on the Pareto plot. Each point can be designated as the optimum state of the OCC configurational settings. Selecting any point depends on the decision-maker criteria. Three points were distinguished on the Pareto plot, where P.O.P #1 has the lowest EUI, P.O.P #2 has the lowest number of thermostat keypress, and P.O.P #3 has the lowest number of light switches. The value of this Pareto-optimal points is shown in Table 13 and Table 14. As expected, EUI changed inversely with the number of interactions, either thermostat keypress or light switches. In other words, decreasing EUI increases the number of interactions. In addition, optimization introduced some points with equal EUI (same color tone) but different interactions with thermostat keypress and light switches. One of the main benefits of three-objective optimization is providing various configurations that can meet different occupant needs without affecting EUI inversely. For example, if an occupant complains about a high number of interactions with light switches but not thermostat, the issue can be resolved by switching to another appropriate Pareto-optimal alternative without increasing energy consumption. Based on the results, it can be done by decreasing learning rate for OCC lighting (0.002) and decreasing the frequency of updating light threshold for light switching off (every four hours) while increasing the corresponding value for OCC thermostat (e.g., increasing learning rate to 0.0015 and the frequency of updating the setpoints to every 150 minutes). Although two-objective optimization was able to minimize energy usage more than three objective-optimization, three-objective optimization gives more freedom over controlling the number of interactions based on their type, which can be interpreted as granting privilege to the comfort.



(a)



(b)

Figure 13:Three-objective optimization results for 2) Case #1 and b) Case #2.

Table 13: Value of selected Pareto-optimal alternatives for three-optimization case #1 and case #2.

OCC Thermostat configurations							
OCC Variable	**Unit**	**Pareto-optimal solutions**					
		(Case #1_Tolerant)			**(Case #2_Sensitive)**		
		P.O.P #1	P.O.P #2	P.O.P #3	P.O.P #1	P.O.P #2	P.O.P #3
Learning rate	-	0.0085	0.005	0.0015	0.0065	0.0005	0.0075
Time interval	min	60	240	150	150	240	240
Maximum amount for threshold (Heating)	°C	22	26	22.5	23	26.15	23.5
Minimum amount for threshold (Heating)	°C	15	19	17	18	20	17.5
Maximum amount for threshold (Cooling)	°C	29	26	28	25.5	25.5	27
Minimum amount for threshold (Cooling)	°C	20.5	19	19	19.5	16.5	18
OCC Light #1 configurations							
OCC Variable	**Unit**	**Pareto-optimal solutions**					
		(Case #1_Tolerant)			**(Case #2_Sensitive)**		
		P.O.P #1	P.O.P #2	P.O.P #3	P.O.P #1	P.O.P #2	P.O.P #3
Learning rate	-	0.006	0.0045	0.002	0.004	0.004	0.0005
Time interval	min	60	150	240	120	240	240
Upper boundary for the light threshold	Lux	300	320	420	320	340	440
Lower boundary for the light threshold	Lux	20	200	260	140	260	280

Table 14: Value of selected Pareto-optimal alternatives for three-optimization case #3 and case #4.

OCC Thermostat configurations							
OCC Variable	**Unit**	**Pareto-optimal solutions**					
		(Case #3_Tolerant)			**(Case #4_Sensitive)**		
		P.O.P #1	P.O.P #2	P.O.P #3	P.O.P #1	P.O.P #2	P.O.P #3
Learning rate	-	0.009	0.0055	0.0085	0.0055	0.001	0.005
Time interval	min	10	240	210	20	240	240
Maximum amount for threshold (Heating)	°C	22.5	25.5	23.5	24	26.5	24.5
Minimum amount for threshold (Heating)	°C	15.5	20	17.5	17.5	20.5	17
Maximum amount for threshold (Cooling)	°C	28.5	26	28	26.5	24.5	28
Minimum amount for threshold (Cooling)	°C	21	19.5	20.5	17.5	18	17.5
OCC Light #2 configurations							
OCC Variable	**Unit**	**Pareto-optimal solutions**					
		(Case #3_Tolerant)			**(Case #4_Sensitive)**		

		P.O.P #1	P.O.P #2	P.O.P #3	P.O.P #1	P.O.P #2	P.O.P #3
Period of updating the threshold	Day	85	12	14	60	60	8
Threshold for the first period	Lux	120	80	260	160	160	280

4.3 Discussion

In this section, the major findings of the results of multi-objective optimization are discussed. The first subsection's discussion focuses on the results of step one optimization, and the rest of the subsections are dedicated to the results of optimization for each investigated OCC.

4.3.1 Building design variables

According to the optimization results for building design parameters (Table 5), some optimal points were in the same range for all cases regardless of the type of occupants. To name the main one: the south direction of the building was identified as the optimal axis to minimize EUI as well as the number of interactions for both occupant types. Since the simulation was conducted using the cold climate of the Montreal weather file, high solar heat gain in this direction causes a reduction in electricity consumption. In addition, WWR of 20% was estimated as the optimal window to wall ratio for both occupant types. One of the notable differences between the design parameters for different types of occupant was related to SHGC, which were gained as 0.6 and 0.45 for tolerant and sensitive occupant, respectively. Given that the optimization algorithm chose the south direction for the office building model as the optimal axis, the higher SHGC for tolerant occupant decreases energy consumption needs in the winter while not disturbing occupant comfort in the summer. However, the lower value of SHGC for sensitive occupant was selected as the optimal point. It can be said that, although higher SHGC would cause to decrease in energy usage in the winter, it disturbs sensitive occupant in the summer and results in a higher number of interactions with thermostat keypress. Moreover, window visible transmittance was identified as another feature that its optimal value depends on the type of occupant for both light OCC #1 and OCC #2. The difference between them was quite slight, though.

4.3.2 Lighting OCC #1

For lighting #1 OCC, the optimal lower boundary of a threshold for switching lights off was 40 lux and 280 lux for tolerant and sensitive occupant, respectively. It can be concluded that 200 lux as the original lower boundary of the threshold for this OCC was too conservative for tolerant occupant. However, the upper boundary of 500 lux in the original configuration of this OCC was known as a bit strict; since the optimal value of some Pareto-optimal alternatives exceeds the original value of this OCC to ensure occupant comfort (e.g., 540 lux). In the original configuration, this OCC reduced the threshold every 30 minutes. Based on the optimal alternatives introduced by the optimization algorithm, this frequency can be reduced to every 10 minutes for tolerant occupant. If the goal is to improve the comfort for sensitive occupant, the algorithm suggested regulating time interval every 3 hours to reduce the switch-off threshold.

4.3.3 Lighting OCC #2

Regarding the optimization results for lighting OCC #2, the frequency of updating the light switching on threshold (P.O.P #2) was identified as almost every week and every two months for the sensitive and tolerant occupant, respectively. It can be said that this period was reduced to one week for sensitive occupants as they are naturally more vulnerable to changes in indoor illuminance. As a result, OCC should update the threshold more frequently to track their reaction to illuminance changes by passing days. Moreover, the fact that this threshold should be updated more frequently to skew to the comfort side of the plot makes sense as it is totally aligned with the results of sensitivity analysis results.

4.3.4 Thermostat OCC

Some logical patterns can be observed when comparing the optimal values of OCC variables for the three selected alternatives (Table 6). For example, optimal points for the learning rate of the OCC thermostat were introduced as 0.005, 0.0075, and 0.0085 for alternatives 1,2 and 3 for sensitive occupant, respectively. Learning rate was growing as the optimal alternatives moved from the comfort-side (right) side of the Pareto-Front plot to the lowest EUI (left) side of it. Similarly, it happened for the tolerant occupant; based on sensitivity analysis results, a higher learning rate reduced energy consumption while disrupting occupant comfort. In addition, in general, the optimal value of the learning rate for tolerant occupant was higher than the related counterpart for sensitive occupant. As the learning rate acts as a rate of increase or decrease of the new setpoint by OCC, there is a need to slow down this rate for sensitive occupant to avoid disturbing occupant comfort. Likewise, time interval for OCC thermostat, which shows the frequency of reducing HSP in the winter and increasing CSP in the summer by OCC, were higher for tolerant than sensitive occupant. It can be concluded that, although a smaller value of time interval can minimize energy usage and the number of interactions for tolerant occupant, the longer time interval would be necessary for sensitive occupant, since it can guarantee comfort for this type of occupant.

The original configuration of OCC thermostat defined the boundary of [22, 25] °C for cooling setpoint in the summer. The Pareto-optimal alternatives introduced these boundaries so that they can reduce to 16°C or reach 27°C in the summer. On the other hand, by comparing EUI and the number of interactions for these alternatives with the baseline scenario, it was found that the two objectives improved. In other words, the optimization algorithm was able to find the optimal points that can improve occupant comfort to a great extent while minimizing energy consumption. However, the optimal points for the boundaries were not as strict as the original configuration. Correspondingly, the heating setpoint can change with the boundary of [20, 22] °C in the original configuration. Nevertheless, the Pareto-front optimal solutions of the upper boundary for heating setpoint exceeded 27°C and 26°C for sensitive and tolerant occupant, respectively. Heating setpoint can also decrease to 16.5°C and 15°C for sensitive and tolerant occupant, respectively. It is worth noting that although expanding these boundaries gives more freedom to OCC to change the setpoint, it does not necessarily mean that it chooses a higher HSP or lower CSP. In fact, by expanding these boundaries, the OCC can learn from occupant interactions efficiently without increasing discomfort that can cause an increase in these occupant interactions with buildings.

Chapter 5: Conclusion

The operation of building systems is highly dependent on occupant behaviour and their interactions with building systems. OCCs are recognized as promising strategies to enhance energy management of buildings. Hence, optimizing the performance of such control systems and customizing their configurations based on the type of occupants and building is an important step to promote user satisfaction as well as energy efficiency. This study proposed a simulation-based framework to optimize the performance of OCC algorithms when learning from stochastic occupant behaviour. The main contribution of this research is integrating a simulation model with a multi-objective optimization algorithm to optimize the performance of OCCs in office buildings based on stochastic and dynamic occupant behaviour models. The simulation-based framework is also equipped with parallel processing capability, which enables robust assessment of optimization problems by investigating a larger domain of solutions while reducing computational time. The framework enables customizing OCC configurational settings while learning from behaviours of different occupant types. It also provided an environment in which different OCCs can be compared under similar conditions, which may not be easily feasible in real buildings.

Conducting a sensitivity analysis on OCC configurational variables and design parameters gives an in-depth insight to decision-makers to identify which available options would significantly improve building energy efficiency and comfort for each occupant type. The two-step optimization framework provides a general approach to robustly optimize these variables for different ranges of occupant types. The first optimization step determines how building design parameters can be influenced by the type of occupant and the choice of OCC. The second optimization step enables improving OCC performance while learning from different ranges of occupant preferences. A single office building was modeled as a proof of concept of the proposed framework, which indicated that the original configuration of OCCs does not suit various types of occupants, which can cause occupant dissatisfaction and energy waste. It was found that the performance of OCCs can improve significantly when their configurational settings are fine-tuned to occupant preferences. In addition, a number of optimal building design parameters were influenced by the type of OCCs when learning from different occupant types. As a result, the choice of OCC and its parameters should be informed by design decisions.

The proposed framework will enable building operations to perform efficiently by working in optimal configurational settings for a broad range of occupant preferences and building types. Therefore, performing simulation using the presented framework for other cities' weather data files would allow building designers to localize the OCC to optimize energy consumption and occupant satisfaction in that context. Furthermore, building designers can use this framework to identify the most compatible combination of OCC configurational settings with design alternatives as well as the potential limitation of employing OCCs in existing buildings before field implementation. They can also use this framework to investigate how design parameters can be affected by considering the uncertainty due to occupant behaviour and the choice of OCC. In addition, it allows building operators to analyze and optimize OCC configurational settings based on the type of occupant and improve the performance of the building in terms of energy consumption and the level of comfort in the post-occupancy stage.

5.1 Limitations and future work

Notwithstanding all the benefits of the proposed framework, it should also be mentioned that the case study has some limitations that must be acknowledged. Since the presented simulation framework works with stochastic occupant behaviour, the performance of OCCs was optimized based on the assumptions of these probabilistic models. Although all the occupant behaviour models used in this study were previously published and developed and validated using real-world data (e.g. [24], [26], [43]), future work should consider integrating more comprehensive occupant behaviour models that represent other behaviours. For example, discomfort glare index (DGI) and the position of the occupant desk with respect to the window and light of the office could be included to simulate occupants' interactions with lights and blinds more precisely. Furthermore, the correlation between different types of occupant behaviour can be investigated.

The assumptions that were made to introduce sensitive and tolerant OB were arbitrary to represent two extreme occupant scenarios since there is generally no information on occupant type beforehand. To resolve this limitation, future work can investigate a more extensive range of occupant assumptions to introduce various occupant types and preferences. Although the results of step one optimization showed that optimal building design variables changed by the type of occupant and OCCs, applying these changes in building designs is quite challenging, specifically if the choice of OCCs or the type of occupant is identified after the design stage. To resolve this issue, building designers would use this framework to identify the optimal designs variables for the average preferences of different occupant types, given uncertainty due to occupant behaviour. Subsequently, building operators can optimize OCC configurational settings for each occupant type using optimization step two of the framework.

Future work will also focus on implementing and optimizing other OCCs that target different buildings systems using alternative approaches. Although the proof-of-concept case study was based on an individual office building model, the proposed framework can be extended to various building types, behaviours, and OCCs, which enables building operators to explore the potential benefits and appropriate settings for implementing OCC strategies.

5.2 Reproducibility

This section explains the overall steps of Python code, which can be found in the Appendix.

At first, the required packages were installed and called from their sources. Then, the working directory is defined, and the EnergyPlus directory with the executable file is provided to the code. Based on the Pymoo package instruction, the optimization problem has to be structured like a class. One of the class components initializes the optimization criteria such as number of variables, number of objectives, the boundaries of variables, etc. The second component of the class is the simulation model developed for this paper's purpose. The path to "idf" file and weather file (epw) is passed to the code at this stage. After that, the variables (building design variables/ OCC configurational variables) which are selected based on sensitivity analysis results have to be defined with their applicable ranges. The code replaces the variables in the "idf" file with the assigned value from the optimization algorithm selection process at each simulation loop. Then, the code creates multiple identical idfs and passes them through the simulation program (EP) along with weather files. The multi-core processing is implemented to speed up the simulation process. The idf and weather files split based on the number of cores passed to the model. In this model, high-performance computing (HPC) is implemented to handle multicore processing. The number of cores passes to the code by the "Ncores=int(sys.argv[1])" command. Results of the simulation are stored in multiple arrays. At the end of one loop of the process, the average of each array will be passed to the optimization algorithm. According to the termination criteria, the algorithm will decide to terminate the process to continue to create a new set of variables. The last part of the code is the genetic algorithm setting section. At this stage, the parameters and characteristics of the selected GA method are passed to the GA model to run the optimization algorithm.

References

[1] U. Energy Information Administration, “International Energy Outlook 2019,” 2019, Accessed: Jan. 03, 2022. [Online]. Available: www.eia.gov/ieo.

[2] UNEP, “owards a zero-emission, efficient, and resilient buildings and construction sector,” 2017. Accessed: Jan. 04, 2022. [Online]. Available: www.globalabc.org.

[3] K. I. Praseeda, B. V. Venkatarama Reddy, and M. Mani, “Life-Cycle Energy Assessment in Buildings: Framework, Approaches, and Case Studies,” *Encycl. Sustain. Technol.*, pp. 113–136, Jan. 2017, doi: 10.1016/B978-0-12-409548-9.10188-5.

[4] S. S. Korsavi, R. V. Jones, and A. Fuertes, “The gap between automated building management system and office occupants’ manual window operations: Towards personalised algorithms,” *Autom. Constr.*, vol. 132, p. 103960, Dec. 2021, doi: 10.1016/J.AUTCON.2021.103960.

[5] NRCan, “Energy Use in the Commercial/Institutional Building Sector in Canada,” 2020.

[6] S. Naylor, M. Gillott, and T. Lau, “A review of occupant-centric building control strategies to reduce building energy use,” *Renew. Sustain. Energy Rev.*, vol. 96, no. June, pp. 1–10, 2018, doi: 10.1016/j.rser.2018.07.019.

[7] M. M. Ouf, J. Y. Park, and H. B. Gunay, “A simulation-based method to investigate occupant-centric controls,” *Build. Simul. 2020 144*, vol. 14, no. 4, pp. 1017–1030, Oct. 2020, doi: 10.1007/S12273-020-0726-Y.

[8] W. Jin, I. Ullah, S. Ahmad, and D. Kim, “Occupant Comfort Management Based on Energy Optimization Using an Environment Prediction Model in Smart Homes,” *Sustain. 2019, Vol. 11, Page 997*, vol. 11, no. 4, p. 997, Feb. 2019, doi: 10.3390/SU11040997.

[9] J. Y. Park *et al.*, “A critical review of field implementations of occupant-centric building controls,” *Build. Environ.*, vol. 165, no. May, p. 106351, 2019, doi: 10.1016/j.buildenv.2019.106351.

[10] W. O’Brien *et al.*, “Introducing IEA EBC annex 79: Key challenges and opportunities in the field of occupant-centric building design and operation,” *Build. Environ.*, vol. 178, no. December 2019, p. 106738, 2020, doi: 10.1016/j.buildenv.2020.106738.

[11] Z. Pang, Y. Chen, J. Zhang, Z. O’Neill, H. Cheng, and B. Dong, “Nationwide HVAC energy-saving potential quantification for office buildings with occupant-centric controls in various climates,” *Appl. Energy*, vol. 279, no. September, 2020, doi: 10.1016/j.apenergy.2020.115727.

[12] J. Li, Z. (Jerry) Yu, F. Haghighat, and G. Zhang, “Development and improvement of occupant behavior models towards realistic building performance simulation: A review,” *Sustain. Cities Soc.*, vol. 50, p. 101685, Oct. 2019, doi: 10.1016/J.SCS.2019.101685.

[13] S. Gilani and W. O'brien, "Best Practices Guidebook on Advanced Occupant Modelling."

[14] S. D'Oca, H. B. Gunay, S. Gilani, and W. O'Brien, "Critical review and illustrative examples of office occupant modelling formalisms:," *https://doi.org/10.1177/0143624419827468*, vol. 40, no. 6, pp. 732–757, Feb. 2019, doi: 10.1177/0143624419827468.

[15] H. B. Gunay, W. O'Brien, and I. Beausoleil-Morrison, "A critical review of observation studies, modeling, and simulation of adaptive occupant behaviors in offices," *Build. Environ.*, vol. 70, pp. 31–47, Dec. 2013, doi: 10.1016/J.BUILDENV.2013.07.020.

[16] H. Burak Gunay, W. O'brien, I. Beausoleil-Morrison, and S. Gilani, "Modeling plug-in equipment load patterns in private office spaces," *Energy Build.*, vol. 121, pp. 234–249, 2016, doi: 10.1016/j.enbuild.2016.03.001.

[17] S. G. H. Burak Gunaya, William O'Briena, Ian Beausoleil-Morrisonb, "Roadmap for Occupant Modelling in Building Codes and Standards," in *the 10th conference of IBPSA-Canada Montréal, QC, Canada, May 9-10, 2018*.

[18] "Occupant modelling for building design and energy codes: roadmap, feasibility study, best practices guidebook, and tested case study," 2020.

[19] S. Carlucci *et al.*, "Modeling occupant behavior in buildings," *Build. Environ.*, vol. 174, p. 106768, May 2020, doi: 10.1016/J.BUILDENV.2020.106768.

[20] J. Mahecha Zambrano, U. Filippi Oberegger, and G. Salvalai, "Towards integrating occupant behaviour modelling in simulation-aided building design: Reasons, challenges and solutions," *Energy Build.*, vol. 253, p. 111498, Dec. 2021, doi: 10.1016/J.ENBUILD.2021.111498.

[21] M. Jia and R. S. Srinivasan, "Occupant behavior modeling for smart buildings: A critical review of data acquisition technologies and modeling methodologies," *Proc. - Winter Simul. Conf.*, vol. 2016-February, pp. 3345–3355, Feb. 2016, doi: 10.1109/WSC.2015.7408496.

[22] P. D. Andersen, A. Iversen, H. Madsen, and C. Rode, "Dynamic modeling of presence of occupants using inhomogeneous Markov chains," *Energy Build.*, vol. 69, pp. 213–223, Feb. 2014, doi: 10.1016/J.ENBUILD.2013.10.001.

[23] D. Yan *et al.*, "Occupant behavior modeling for building performance simulation: Current state and future challenges," *Energy Build.*, vol. 107, pp. 264–278, Nov. 2015, doi: 10.1016/J.ENBUILD.2015.08.032.

[24] C. F. Reinhart, "Lightswitch-2002: A model for manual and automated control of electric lighting and blinds," *Sol. Energy*, vol. 77, no. 1, pp. 15–28, 2004, doi: 10.1016/j.solener.2004.04.003.

[25] Y. Zhang, X. Bai, F. P. Mills, and J. C. V. Pezzey, "Rethinking the role of occupant behavior in building energy performance: A review," *Energy Build.*, vol. 172, pp. 279–294, Aug. 2018, doi: 10.1016/J.ENBUILD.2018.05.017.

[26] D. Wang, C. C. Federspiel, and F. Rubinstein, "Modeling occupancy in single person offices," *Energy Build.*, vol. 37, no. 2, pp. 121–126, 2005, doi: 10.1016/j.enbuild.2004.06.015.

[27] J. Widén, A. Molin, and K. Ellegård, "Models of domestic occupancy, activities and energy use based on time-use data: deterministic and stochastic approaches with application to various building-related simulations," *undefined*, vol. 5, no. 1, pp. 27–44, Jan. 2012, doi: 10.1080/19401493.2010.532569.

[28] G. Y. Yun, P. Tuohy, and K. Steemers, "Thermal performance of a naturally ventilated building using a combined algorithm of probabilistic occupant behaviour and deterministic heat and mass balance models," *Energy Build.*, vol. 41, no. 5, pp. 489–499, May 2009, doi: 10.1016/J.ENBUILD.2008.11.013.

[29] T. Labeodan, K. Aduda, G. Boxem, and W. Zeiler, "On the application of multi-agent systems in buildings for improved building operations, performance and smart grid interaction – A survey," *Renew. Sustain. Energy Rev.*, vol. 50, pp. 1405–1414, Oct. 2015, doi: 10.1016/J.RSER.2015.05.081.

[30] C. J. Andrews, H. C. Putra, and C. Brennan, "Simulation Modeling of Occupant Behavior in Commercial Buildings 2013. Simulation Modeling of Occupant Behavior in Commercial Buildings," 2013.

[31] A. Wagner, W. O'Brien, and B. Dong, "Exploring occupant behavior in buildings: Methods and challenges," *Explor. Occupant Behav. Build. Methods Challenges*, pp. 1–310, Jan. 2017, doi: 10.1007/978-3-319-61464-9.

[32] H. B. Gunay, W. O'Brien, I. Beausoleil-Morrison, and S. Gilani, "Development and implementation of an adaptive lighting and blinds control algorithm," *Build. Environ.*, vol. 113, pp. 185–199, 2017, doi: 10.1016/j.buildenv.2016.08.027.

[33] J. Y. Park and Z. Nagy, "HVACLearn: A reinforcement learning based occupant-centric control for thermostat set-points," *e-Energy 2020 - Proc. 11th ACM Int. Conf. Futur. Energy Syst.*, pp. 434–437, 2020, doi: 10.1145/3396851.3402364.

[34] S. Jung, J. Jeoung, and T. Hong, "Occupant-centered real-time control of indoor temperature using deep learning algorithms," *Build. Environ.*, p. 108633, Nov. 2021, doi: 10.1016/J.BUILDENV.2021.108633.

[35] Y. Ye *et al.*, "Energy-saving potential evaluation for primary schools with occupant-centric controls," *Appl. Energy*, vol. 293, p. 116854, Jul. 2021, doi: 10.1016/J.APENERGY.2021.116854.

[36] C. de Bakker, M. Aries, H. Kort, and A. Rosemann, "Occupancy-based lighting control in open-plan office spaces: A state-of-the-art review," *Build. Environ.*, vol. 112, pp. 308–321, Feb. 2017, doi: 10.1016/J.BUILDENV.2016.11.042.

[37] M. M. Ouf, W. O'Brien, and H. B. Gunay, "Improving occupant-related features in building performance simulation tools," *Build. Simul. 2018 114*, vol. 11, no. 4, pp. 803–817, Apr. 2018, doi: 10.1007/S12273-018-0443-Y.

[38] T. Hong, Y. Chen, Z. Belafi, and S. D'Oca, "Occupant behavior models: A critical review of implementation and representation approaches in building performance simulation programs," *Build. Simul.*, vol. 11, no. 1, Feb. 2018, doi: 10.1007/S12273-017-0396-6.

[39] T. Hong, H. Sun, Y. Chen, S. C. Taylor-Lange, and D. Yan, "An occupant behavior modeling tool for co-simulation," *Energy Build.*, vol. 117, pp. 272–281, Apr. 2016, doi: 10.1016/J.ENBUILD.2015.10.033.

[40] H. B. Gunay, W. O'Brien, and I. Beausoleil-Morrison, "Implementation and comparison of existing occupant behaviour models in EnergyPlus," *J. Build. Perform. Simul.*, vol. 9, no. 6, pp. 567–588, Nov. 2016, doi: 10.1080/19401493.2015.1102969/SUPPL_FILE/TBPS_A_1102969_SM1650.TXT.

[41] B. W. Hobson, B. Huchuk, and H. Burak Gunay, "A workflow for evaluating occupant-centric controls using building simulation," 2021, doi: 10.1080/19401493.2021.1894485.

[42] Z. Pang, Y. Chen, J. Zhang, Z. O'Neill, H. Cheng, and B. Dong, "Quantifying the nationwide HVAC energy savings in large hotels: the role of occupant-centric controls," *https://doi.org/10.1080/19401493.2021.1994650*, vol. 14, no. 6, pp. 749–769, 2021, doi: 10.1080/19401493.2021.1994650.

[43] H. B. Gunay, W. Shen, G. Newsham, and A. Ashouri, "Modelling and analysis of unsolicited temperature setpoint change requests in office buildings," *Build. Environ.*, vol. 133, no. December 2017, pp. 203–212, 2018, doi: 10.1016/j.buildenv.2018.02.025.

[44] C. F. Reinhart, "Lightswitch-2002: a model for manual and automated control of electric lighting and blinds," *Sol. Energy*, vol. 77, no. 1, pp. 15–28, Jan. 2004, doi: 10.1016/J.SOLENER.2004.04.003.

[45] F. Haldi and D. Robinson, "The impact of occupants' behaviour on building energy demand," *J. Build. Perform. Simul.*, vol. 4, no. 4, pp. 323–338, 2011, doi: 10.1080/19401493.2011.558213.

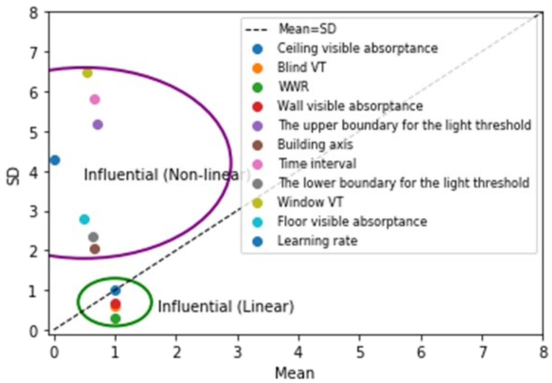
[46] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[47] R. Wang, "An Improved Nondominated Sorting Genetic Algorithm for Multiobjective Problem," *Math. Probl. Eng.*, vol. 2016, 2016, doi: 10.1155/2016/1519542.

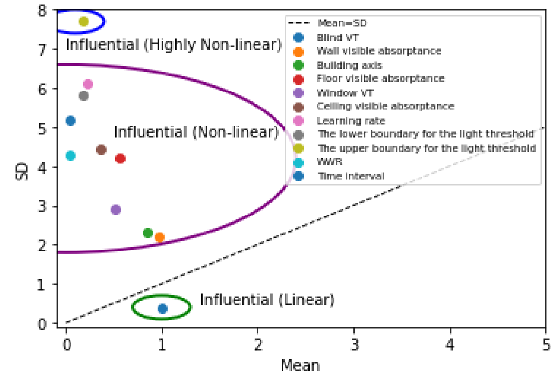
[48] Z. Nagy, F. Y. Yong, M. Frei, and A. Schlueter, "Occupant centered lighting control for comfort and energy efficient building operation," *Energy Build.*, vol. 94, pp. 100–108, 2015, doi: 10.1016/j.enbuild.2015.02.053.

[49] H. B. Gunay, W. O'Brien, I. Beausoleil-Morrison, and J. Bursill, "Development and implementation of a thermostat learning algorithm," *Sci. Technol. Built Environ.*, vol. 24, no. 1, pp. 43–56, 2018, doi: 10.1080/23744731.2017.1328956.

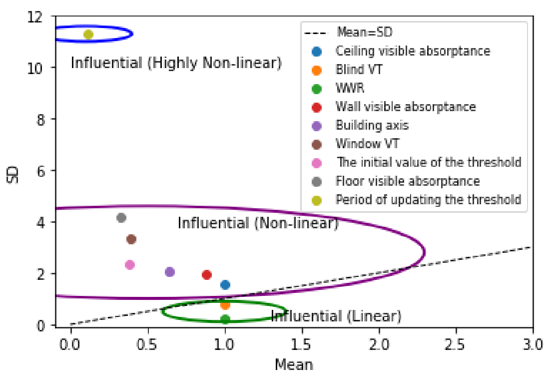
Appendix A. The results of sensitivity analysis for OCC light #2 and OCC light #2.



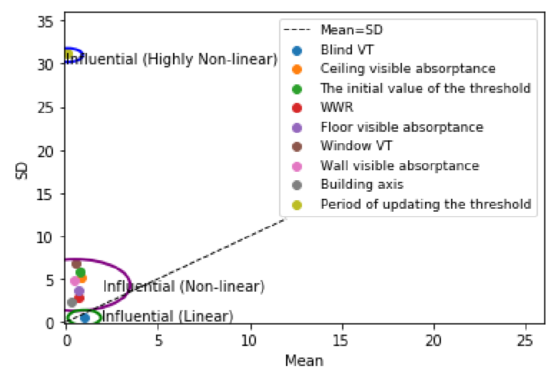
OCC light #1- Tolerant



OCC light #1- Sensitive



OCC light #2- Tolerant



OCC light #2- Sensitive

Appendix B. Results of finite differences method for all cases.

Result of sensitivity analysis for OCC thermostat design variables based on sensitive occupant scenario and the number of interactions as the target variable.

OCC Thermostat _ Sensitive occupant_ Design variables					
Design Variables		SP increase		SP decrease	
		Mean	SD	Mean	SD
1	Exterior Wall R-value	1	1.88	0.88	3.48
2	SHGC	0.99	2.03	0.99	1.05
3	WWR	0.86	1.39	0.97	0.81
4	U-factor Glazing	0.81	1.74	0.88	1.34
5	Building Axis	0.38	1.96	0.30	2.09
6	Blind Solar Transmittance	0.20	2.41	0.27	2.12

Result of sensitivity analysis for OCC thermostat variables based on sensitive occupant scenario and the number of interactions as the target variable

OCC Thermostat _ Sensitive occupant					
OCC Variables		SP increase		SP decrease	
		Mean	SD	Mean	SD
1	Lower boundary for CSP	-	-	0.97	0.81
2	Upper boundary for CSP	-	-	0.84	1.55
3	Learning Rate	0.99	2.06	0.94	3.24
4	Lower boundary for HSP	0.90	0.87	-	-
5	Upper boundary for HSP	0.76	6.17	-	-
6	Time interval	0.10	5.29	0.79	8.40

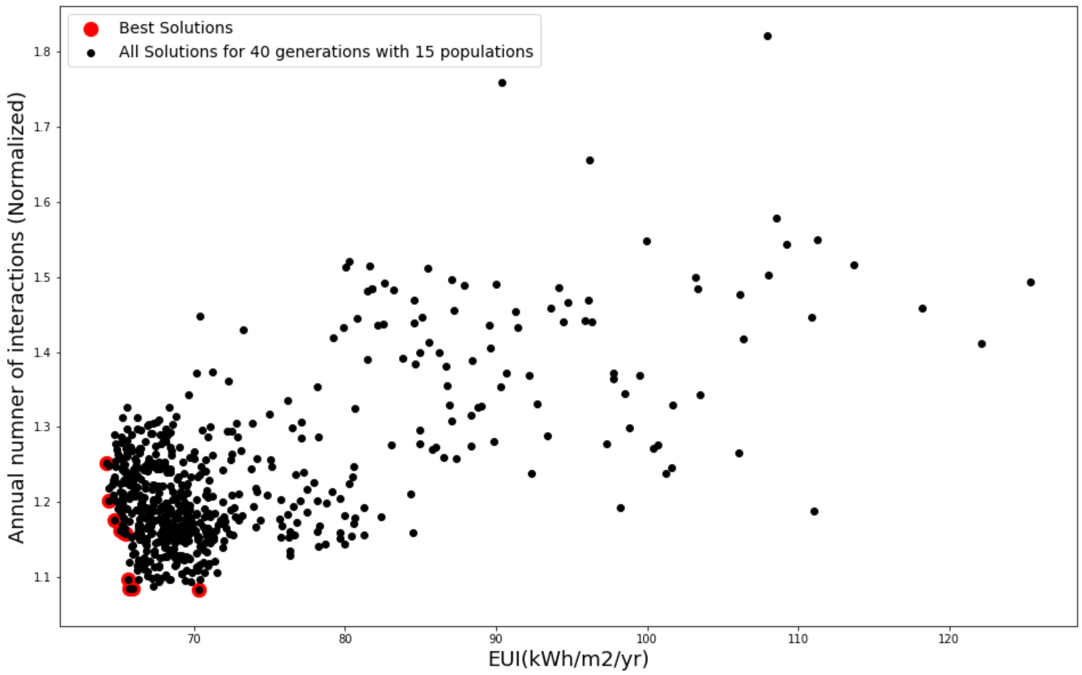
Result of sensitivity analysis for OCC light #2 based on sensitive occupant scenario and the number of light switches on as the target variable.

OCC Light #1 _ Sensitive occupant			
	OCC Variables	**Mean (μ)**	**SD (∂)**
1	**Blind VT**	1	0.38
2	**Wall visible absorptance**	0.98	2.22
3	**Building axis**	0.85	2.33
4	**Floor visible absorptance**	0.56	4.23
5	**Window VT**	0.51	2.91
6	**Ceiling visible absorptance**	0.36	4.45
7	**Learning rate**	0.23	6.13
8	**The lower boundary for the light threshold**	0.18	5.80
9	**The upper boundary for the light threshold**	0.18	7.70
10	**WWR**	0.05	4.30
11	**Time interval**	0.04	5.18

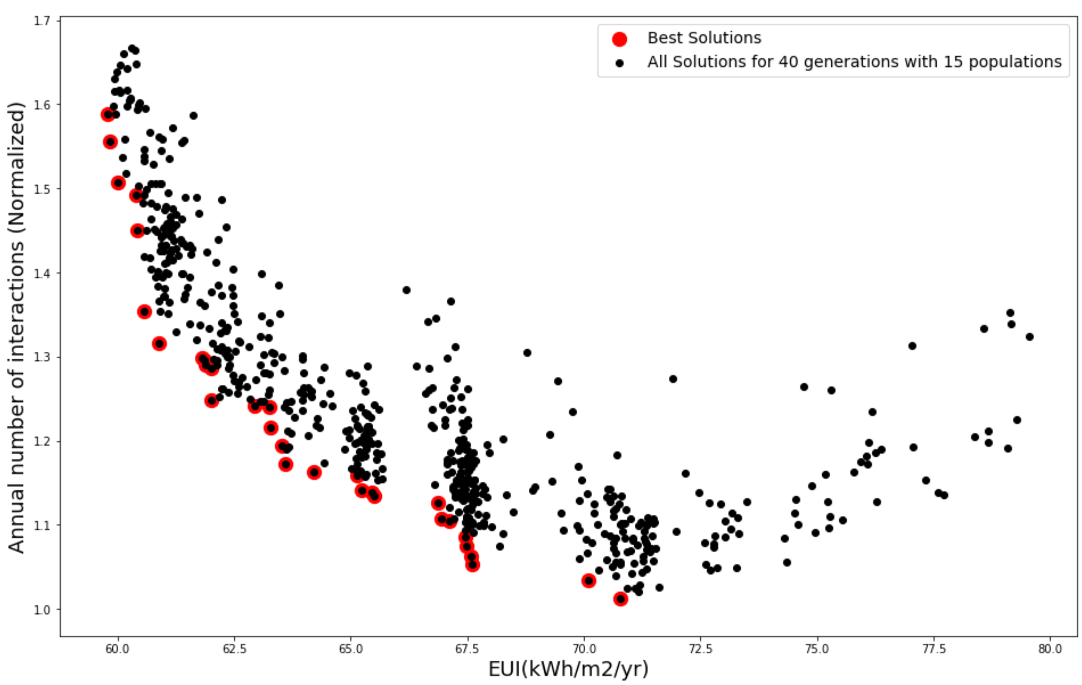
Result of sensitivity analysis for OCC light #2 based on sensitive occupant scenario and the number of light switches on as the target variable.

OCC Light #2 _ Sensitive occupant			
	OCC Variables	**Mean (μ)**	**SD (∂)**
1	**Blind VT**	0.99	0.58
2	**Ceiling visible absorptance**	0.85	5.15
3	**The initial value of the threshold**	0.76	5.83
4	**WWR**	0.69	2.86
5	**Floor visible absorptance**	0.65	3.67
6	**Window VT**	0.54	6.82
7	**Wall visible absorptance**	0.48	4.91
8	**Building axis**	0.32	2.32
9	**Period of updating the threshold**	0.07	31.30

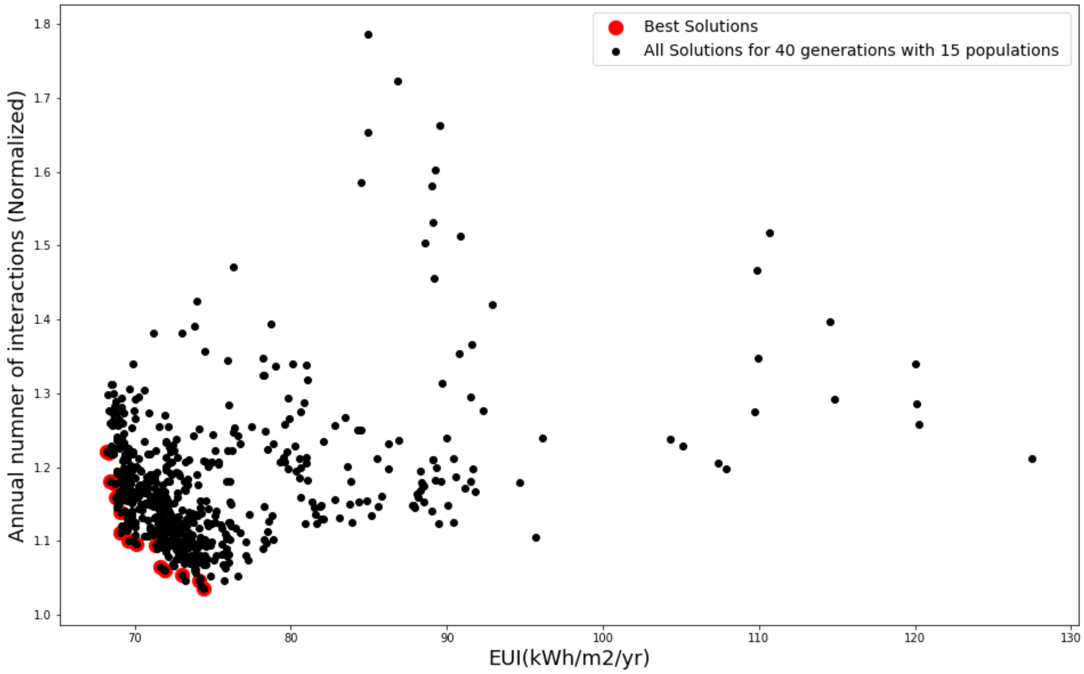
Appendix C. Two-objective optimization results.



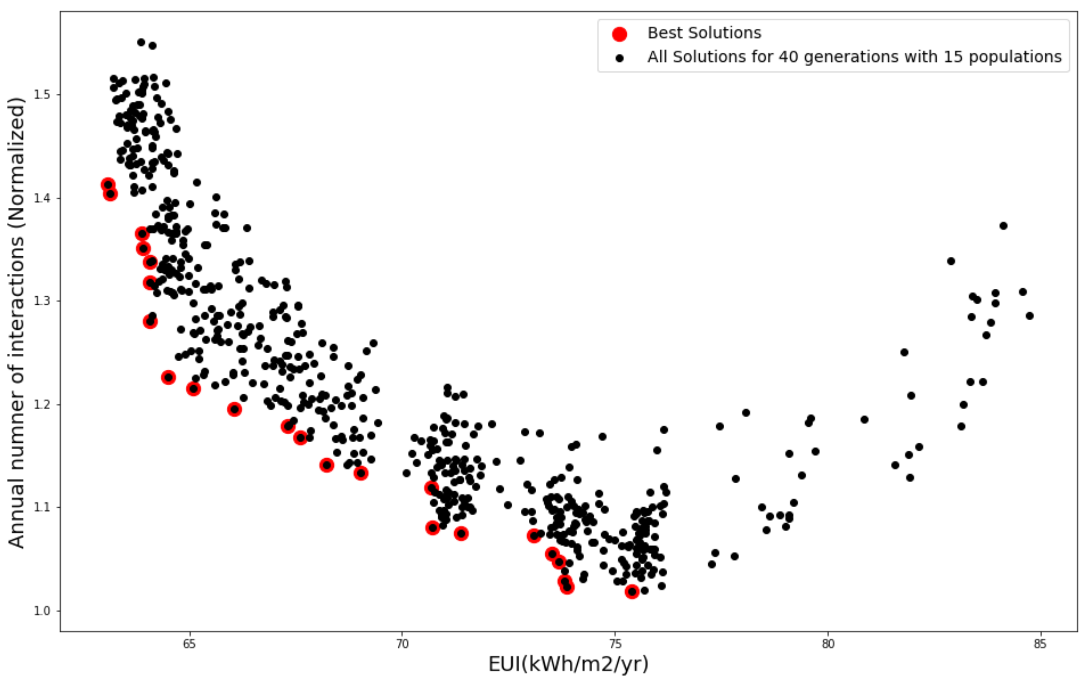
Case #2- Step 1 (design variables)



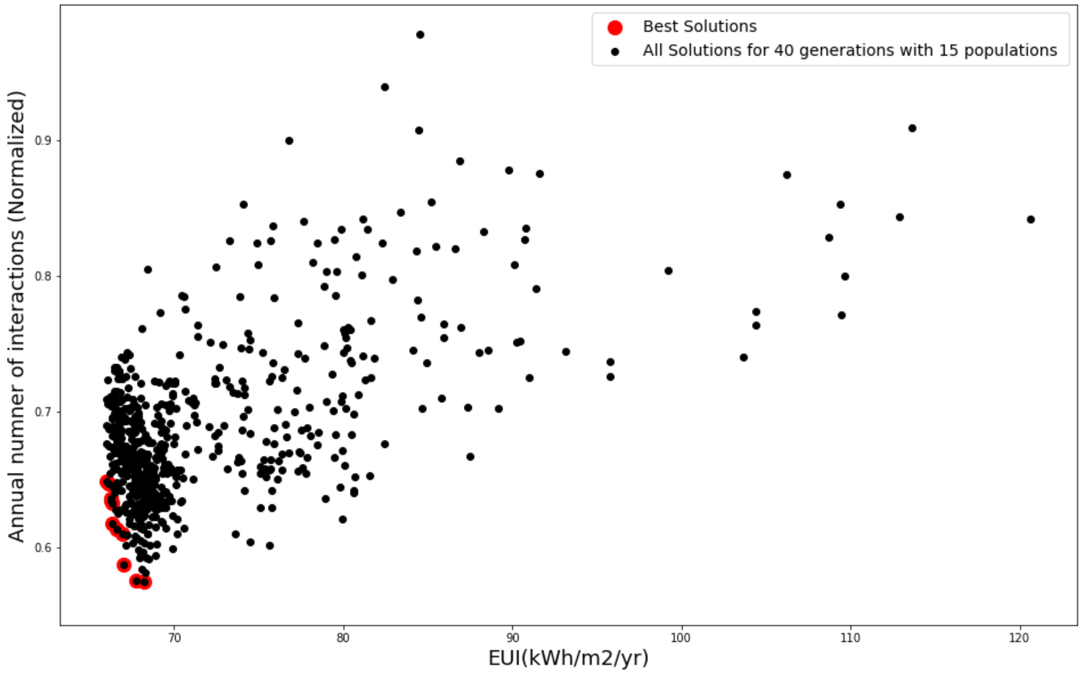
Case #2- Step 2 (OCC configurational variables)



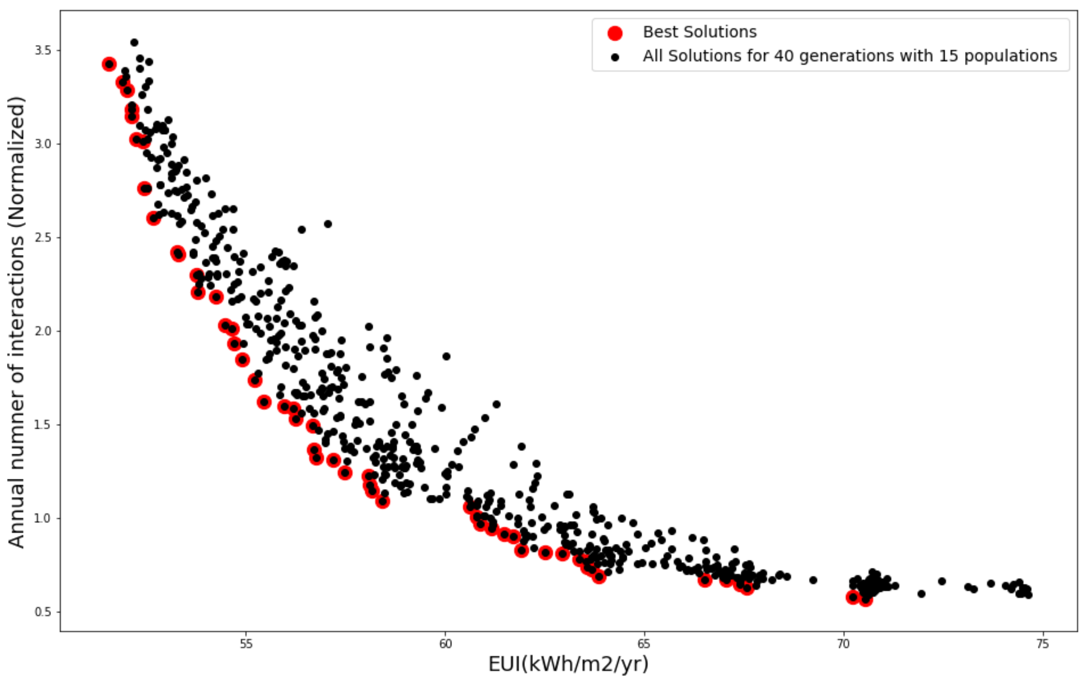
Case #3- Step 1 (design variables)



Case #3- Step 2 (OCC configurational variables)

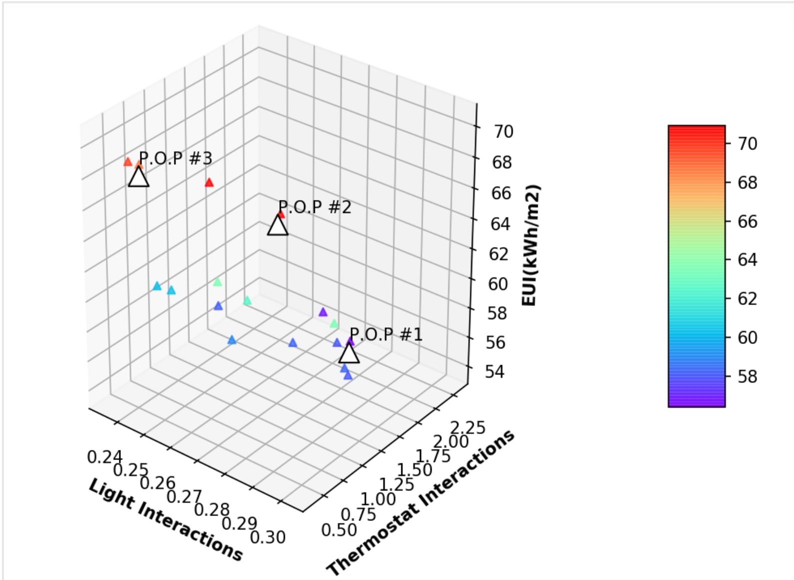


Case #4- Step 1 (design variables)

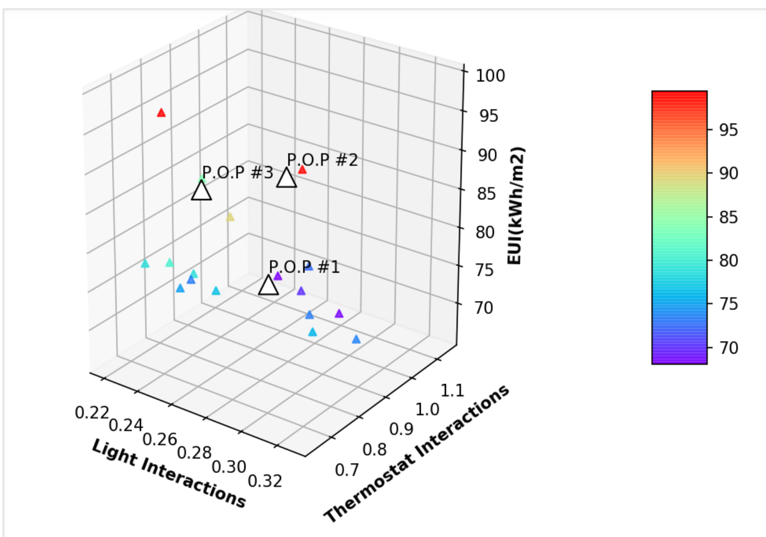


Case #4- Step 2 (OCC configurational variables)

Appendix D. Three-objective optimization results



Case #3- Tolerant



Case #4- Sensitive

Appendix E. PYTHON code of multi-objective optimization.

"""multiprocessing runs"""

```
#!/usr/bin/env python
# coding: utf-8
# %%
import pandas as pd
import os
import eppy as ep
from eppy import modeleditor
import sys
from eppy.modeleditor import IDF
import pandas as pd
import csv
from statistics import mean
import numpy as np
from geneticalgorithm import geneticalgorithm as ga
from eppy.pytest_helpers import do_integration_tests
from eppy.runner.run_functions import install_paths, EnergyPlusRunError
from eppy.runner.run_functions import multirunner
from eppy.runner.run_functions import run
from eppy.runner.run_functions import runIDFs
import pymoo
import autograd.numpy as anp

from pymoo.problems.util import load_pareto_front_from_file
from pymoo.model.problem import Problem
from pymoo.algorithms.nsga2 import NSGA2
from pymoo.optimize import minimize
from pymoo.visualization.scatter import Scatter
from pymoo.factory import get_algorithm, get_crossover, get_mutation,
get_sampling
from pymoo.model.population import Population
from pymoo.performance_indicator.hv import Hypervolume
from pymoo.util.nds.non_dominated_sorting import NonDominatedSorting
Ncores=int(sys.argv[1])


# %%
path = "/speed-scratch/z_khoras/"


# %%
iddfile = path+ '/EP-8-9/EnergyPlus-8-9-0/Energy+.idd'
IDF.setiddname(iddfile)
```

```
# %%
"""multiprocessing runs"""

# using generators instead of a list
# when you are running a 100 files you have to use generators

import os
from eppy.modeleditor import IDF
from eppy.runner.run_functions import runIDFs

class MultiNGBT(Problem):

    def __init__(self):
        super().__init__(n_var=7, n_obj=2, n_constr=0,
elementwise_evaluation=True, type_var=int)
        #self.args = args
        self.xl=anp.array([0,0,0,0,0,0,0])
        self.xu = anp.array([3,10,10,10,10,10,3])



    def _evaluate(self, X, out,*args, **kwargs):



        def make_options(idf):
            idfversion =
idf.idfobjects['version'][0].Version_Identifier.split('.')
            idfversion.extend([0] * (3 - len(idfversion)))
            idfversionstr = '-'.join([str(item) for item in idfversion])
            fname = idf.idfname
            options = {
                'ep_version':idfversionstr,
                'output_prefix':os.path.basename(fname).split('.')[0],
                'output_suffix':'C',
                'output_directory':os.path.dirname(fname),
                'readvars':True,
                'expandobjects':True
                }
            return options

        path = "/speed-scratch/z_khoras/"
        from eppy.modeleditor import IDF
        iddfile = path+ '/EP-8-9/EnergyPlus-8-9-0/Energy+.idd'
        IDF.setiddname(iddfile)
        epwfile = path+'/PMBTNG/CAN_PQ_Montreal.Intl.AP.716270_CWEC.epw'
        #mapping for heating maximum threshold
        #maxh=np.arange(25,30.1,0.5)
        a = 26.5
        #a = str(X[0])
```

```
# mapping for heating minimum threshold
#minhandc=np.arange(15,20.1,0.5)
b= 19

#mapping for cooling maximum threshold
#maxc=np.arange(25,30.1,0.5)
c=25.5
#c = str(X[1])
#c= maxc[int(X[1])]
#mapping for colling minimum threshold
d=16.5

#mapping for U-factor glazing
#e1=np.arange(1.4,2.2,0.1)
e2=1.4

#mapping for SHGC
#SHGC=np.arange(0.3,0.61,0.05)
f=0.5499

# mapping axis to an array
g1=[0,90,180,270]
g2=g1[int(X[0])]

#mapping for blind solar transmittance
#Blind=np.arange(0.05,0.2,0.05)
h=0.05

#mapping wwr
i1=np.arange(0.2,0.71,0.05)
#i2=0.2
i2 = i1[int(X[1])]
#mapping for roof R-value
#j1=np.arange(0.55,1.26,0.1)
j2=1.0499

#mapping for floor R-value
#k1=np.arange(0.13,0.54,0.1)
k2=0.43

#mapping for exterior wall R-value
#l1=np.arange(0.2,0.6,0.05)
l2=0.499

#Mapping for period
m = 30

# mapping for luxmean
#n1=np.arange(0,301,20)
n2= 0
```

```
        #mapping for window VT
        o1=np.arange(0.3,0.81,0.05)
        #o2=0.649
        o2 = o1[int(X[2])]

        #mapping visible reflectance for floor
        p1=np.arange(0,1.1,0.1)
        #p2=0
        p2 = p1[int(X[3])]

        #mapping visible reflectance for roof
        q1 = p1[int(X[4])]
        #q1=0.2

        #mapping visible reflectance for walls
        r1=np.arange(0,1.1,0.1)
        #r2=r1[int(X[1])]
        #r2=0
        r2 =r1[int(X[5])]

        #mapping for blind visible transmittance
        s1=np.arange(0.05,0.21,0.05)
        #s2=0.15
        s2 = s1[int(X[6])]

        fname1 = path +'/PMBTNG/BT_NG_S.idf'
        epwfile = path+'/PMBTNG/CAN_PQ_Montreal.Intl.AP.716270_CWEC.epw'

        idf = IDF(fname1,epwfile)
        occthermostatmodel =
idf.idfobjects['EnergyManagementsystem:program'][8]
        occthermostatmodel.Program_Line_73 ="IF HSP < " + str(b) #maybe we
can use directly "set x=" + str(X[0])
        occthermostatmodel.Program_Line_74 ="set HSP = " + str(b)
        # the second variable:
        occthermostatmodel.Program_Line_75="ELSEIF HSP >" + str(a)
        occthermostatmodel.Program_Line_76="set HSP = " + str(a)
        # the third variable:
        occthermostatmodel.Program_Line_79 ="IF CSP < " + str(d)
        occthermostatmodel.Program_Line_80 ="set CSP = " + str(d)
        # the forth variable
        occthermostatmodel.Program_Line_81 ="ELSEIF CSP > " + str(c)
        occthermostatmodel.Program_Line_82 ="set CSP = " + str(c)
        # u-factor glazing
        Windowmaterial =
idf.idfobjects['WindowMaterial:SimpleGlazingSystem'][0]
        Windowmaterial.UFactor=e2
        #SHGC
        Windowmaterial.Solar_Heat_Gain_Coefficient=f
        #the north axes
        office=idf.idfobjects['Building'][0]
        office.North_Axis=g2
```

```
        #blind solar Transmittance
        Blind_material=idf.idfobjects['WindowMaterial:Shade'][0]
        Blind_material.Solar_Transmittance=h
        #floor R-value
        F16_acoustic_tile_floor=idf.idfobjects['Material'][13]
        F16_acoustic_tile_floor.Thickness=k2
        #roof R-value
        F16_acoustic_tile_roof=idf.idfobjects['Material'][5]
        F16_acoustic_tile_roof.Thickness=j2
        # interior wall R-value
        #G01a_19mm_gypsum_board=idf.idfobjects['Material'][2]
        #G01a_19mm_gypsum_board.Thickness=l1
        #exterior wall R-value
        halfinch_gypsum=idf.idfobjects['Material'][6]
        halfinch_gypsum.Thickness=l2

        #Nagy OCC
        occlightingmodel =
idf.idfobjects['EnergyManagementsystem:program'][4]
        occlightingmodel.Program_Line_7 ="set x=" + str(m) #maybe we can use
directly "set x=" + str(X[0])
        # the second variable:
        occlightingmodel.Program_Line_3="set luxmean=" + str(n2)
        # the third variable:
        Windowmaterial =
idf.idfobjects['WindowMaterial:SimpleGlazingSystem'][0]
        Windowmaterial.Visible_Transmittance=o2
        #the forth variable (floor visible absorptance)
        F16_acoustic_tile_floor=idf.idfobjects['Material'][13]
        F16_acoustic_tile_floor.Visible_Absorptance=p2
        #the fifth variable (roof visible absorptance)
        F16_acoustic_tile_roof=idf.idfobjects['Material'][5]
        F16_acoustic_tile_roof.Visible_Absorptance=q1
        #wall visible absorptance
        G01a_19mm_gypsum_board=idf.idfobjects['Material'][2]
        G01a_19mm_gypsum_board.Visible_Absorptance=r2
        #the north axes
        #blind visible Transmittance
        Blind_material=idf.idfobjects['WindowMaterial:Shade'][0]
        Blind_material.Visible_Transmittance=s2

        idf.saveas(path +'/PMBTNG/BT_NG_S.idf')

        #WWr
        from geomeppy import IDF

        fname2 = path +'/PMBTNG/BT_NG_S.idf'
        idf1 = IDF(fname2,epwfile)
        idf1.set_wwr(wwr=0, wwr_map={180: i2}, force=True, construction=
"Exterior Window")
        idf1.saveas(path +'/PMBTNG/BT_NG_S.idf')
        #setting wshCTRL
```

```
from eppy.modeleditor import IDF
fname1 = path +'/PMBTNG/BT_NG_S.idf'
epwfile = path+'/PMBTNG/CAN_PQ_Montreal.Intl.AP.716270_CWEC.epw'
idf = IDF(fname1,epwfile)
sub_surface = idf.idfobjects['FenestrationSurface:Detailed'][0]
sub_surface.Shading_Control_Name="wshCTRL1"
idf.saveas(path +'/PMBTNG/BT_NG_S.idf')

fnames=[]
for i in range (1,33):
    fname1 = path +'/PMBTNG/BT_NG_S.idf'
    epwfile = path+'/PMBTNG/CAN_PQ_Montreal.Intl.AP.716270_CWEC.epw'
    idf = IDF(fname1,epwfile)
    idf.saveas(path +'/PMBTNG/BT_NG_S%d.i
from eppy.modeleditor import IDF
from eppy.runner.run_functions import runIDFs
idfs = (IDF(fname, epwfile) for fname in fnames)
runs = ((idf, make_options(idf) ) for idf in idfs)
num_CPUs = Ncores
runIDFs(runs, num_CPUs)

TCENERGY=[]
THENERGY=[]
TEUI=[]
TL = []
TINC=[]
TDCR=[]
TRELC=[]
TON=[]
TOFF=[]

for i in range (1,33):
    Data=pd.read_csv(path +'/PMBTNG/BT_NG_S%d.csv'%(i))

    CENERGY=Data['THERMAL ZONE 1 IDEAL LOADS AIR SYSTEM:Zone Ideal
Loads Zone Total Cooling Energy [J](TimeStep)'].sum()*2.78*10**(-7)
    HENERGY=Data['THERMAL ZONE 1 IDEAL LOADS AIR SYSTEM:Zone Ideal
Loads Zone Total Heating Energy [J](TimeStep)'].sum()*2.78*10**(-7)
    INC=Data['EMS:SP_Incoutput [](TimeStep)'].sum()
    DCR=Data['EMS:SP_Dcroutput [](TimeStep)'].sum()
    ELC=Data['LIGHT:Lights Electric Energy
[J](TimeStep)'].sum()*2.78*10**(-7)
    ON=Data['EMS:countonoutput [](TimeStep)'].iloc[-1]
    OFF=Data['EMS:countoffoutput [](TimeStep)'].iloc[-1]
    TRELC.append(ELC)
    TON.append(ON)
    TOFF.append(OFF)
    TCENERGY.append(CENERGY)
    THENERGY.append(HENERGY)
    TINC.append(INC)
    TDCR.append(DCR)
```

```
            file = path +'/PMBTNG/BT_NG_S%dTable.csv'%(i)
            f = open(file,'rt')
            reader = csv.reader(f)
            csv_list = []
            for l in reader:
                csv_list.append(l)
            f.close()
            df = pd.DataFrame(csv_list)
df'%(i))

            fnames.append(path +'/PMBTNG/BT_NG_S%d.idf'%(i))

            Lighting = df.iloc[51,2]
            EUI=df.iloc[14,3]
            TEUI.append(EUI)
            TL.append(Lighting)
        # change type of EUI array from string to float
        TEUI = np.array(TEUI, dtype=np.float32)
        TL = np.array(TL, dtype=np.float32)
        TEUI=TEUI*0.278
        TL = TL*278
        #print (np.average(TCENERGY))
        #print (np.average(THENERGY))
        #print (np.average(TINC))
        #print (np.average(TDCR))
        #print (np.average(TRELC))
        #print (np.average(TON))
        #print (np.average(TOFF))
        obj1 = np.average(TL)
        obj2 = np.average(TON)

        out["F"] = anp.column_stack([obj1, obj2])
        #return (np.average(TEUI),np.average(TOFF))

    #def _calc_pareto_front(self, *args, **kwargs):
    #    return load_pareto_front_from_file("MultiNGBT.pf")
    def _cal_pareto_front(self, *args, **kwargs):
        return func_pf(**kwargs)
    #def _calc_pareto_set(self, *args, **kwargs):
    #    return func_ps(**kwargs)
#vectorized_problem = MultiNGBT()

# %%
problem = MultiNGBT()


method = get_algorithm("nsga2",
                       pop_size=15,
                       sampling=get_sampling("int_random"),
                       crossover=get_crossover("int_sbx", prob=1.0, eta=3.0),
                       mutation=get_mutation("int_pm", eta=3.0),
                       eliminate_duplicates=True,
```

```
                              )


res = minimize(problem,
               method,
               ("n_gen", 30),
               verbose=True,
               save_history = True,
               seed=5)
print("Best solution found: %s" % res.X)
print("Function value: %s" % res.F)
print("design Space Value: %s" % res.X)
print("Algorithm Object: %s" % res.algorithm)
print("Final Population Object: %s" % res.pop)
print("History: %s" % res.history)

all_pop = Population()

for algorithm in res.history:
    all_pop = Population.merge(all_pop, algorithm.off)
df_Var = pd.DataFrame(all_pop.get("X"), columns=[f"X{i+1}" for i in
range(problem.n_var)])
df_Res = pd.DataFrame(all_pop.get("F"), columns=[f"F{i+1}" for i in
range(problem.n_obj)])
df_Var.to_csv('Variables.csv')
df_Res.to_csv('Results.csv')

#pf = problem.pareto_front(use_cache = False, flatten = False)
#ps = problem.pareto_set(use_cache = False, flatten = False)
#print(pf)
#print(ps)
#Val2 = [e.pop.get("F").max() for e in res.history]
Val = [e.pop.get("F").min(axis=0) for e in res.history]
print(Val)

#Performance
#non dominated sorting
#print(Val2)
#plot = Scatter()
#plot.add(res.F, color="red")
#plot.show()
```