Decision Support System (DSS) for selecting sustainable insulation material using Pareto Search and Novel fuzzy modified TOPSIS approach

by

Sungyi Kim

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Construction Engineering and Management

Department of Civil and Environmental Engineering
University of Alberta

Abstract

This research developed a Decision Support System (DSS) to aid decision-makers in selecting the most sustainable insulation materials and their thickness among commercially available alternatives. The developed DSS ranks available alternatives according to individual project cases by incorporating project information, material information, and the decision makers' preferences.

The methodology developed in this study utilizes the TOPSIS technique with Pareto search technique for multi-objective optimization. By limiting the alternatives to the 'Pareto front' for Life Cycle Assessment (LCA) and Life Cycle Cost (LCC), the study attempted to reduce subjectivity in the Multi-Criteria Decision-Making (MCDM) process. The suggested method is demonstrated in Excel and programmed with Python to implement a user interface for data input and output results.

In addition, this study uses a product-specific Environmental Product Declaration (EPD) of material to calculate embodied energy in the Life Cycle Assessment (LCA). Also, the research provides understanding of the decision-making criteria for the sustainable selection of insulation materials, based on literature review and structured interviews with industry experts.

The developed DSS has been validated by industry experts and tested with different inputs.

Acknowledgments

I would like to thank my graduate studies supervisor, Dr. Ahmed Hammad, who gave me the opportunity to continue my thesis by giving me courage and guidance throughout this research. I would like to extend my gratitude to Dr. Mohamed Al-Hussein for giving me the chance to be in the MSc program. Special thanks to my friend, Jason Mulvaney who proofread my thesis.

My extended gratitude to Mr. Justin Phill and Mr. Kelly Hopkin for sharing their time and advice.

Finally, I thank my loving husband and parents for their support, encouragement, and for always being on my side.

Table of Contents

List of Table

List of Figure

List of Abbreviations

AHP Analytic Hierarchy Process

BIM Building Information Modeling

DSS Decision Support System

EPD Environmental Product Declaration

FU Functional Unit

GHG Green House Gas

HVAC Heating, Ventilation, and Air Conditioning

ISO International Standards Organization

LCA Life Cycle Assessment

LCC Life Cycle Cost(ing)

LCI Life Cycle Inventory

M&E Mechanical & Electrical

MCDM Multi-Criteria Decision Making

MOO Multi-Objective Optimization

NZE House Net Zero Energy House

PCR Product category rules

PHI Passive House Institute

PWF Present Worth Factor

WLCC Whole Life Cycle Cost(ing)

Chapter 1 **Introduction**

1.1 **Rationale for Research**

The Canadian government introduced “The Pan-Canadian Framework (PCF)” in 2016 in response to the Paris Agreement. It sets a target of reducing the country’s carbon emissions to 30 percent below its 2005 levels by 2030 (Canada.Ca, n.d.). The construction industry cannot hide its head in the sand as building and construction are responsible for 39% of all carbon emissions globally where operational emissions contribute 28% and embodied carbon emissions 11% (World Green Building Council, 2019).

In 2017, Canada's total GHG emissions, including electricity, were 488.6. (Mt of CO2e). Among the total emissions, the construction industry accounted for 7.7 (Mt of CO2e), without including material transportation and manufacturing emissions (Natural Resources Canada, n.d.). Manufacturing accounted for 19% of global GHG emissions, while transportation accounted for 13% (Stiel et al., 2016). In an effort to reduce carbon emissions, Passive Houses and Net Zero Houses (hereafter NZE) are being built. Some aspects of these houses are stipulated in building codes or policies of many countries. Since 2017, Canada has promoted the “Build Smart” strategy nationwide as part of a swathe of other policies and incentive schemes. However, Passive Houses and Net Zero Houses are primarily concerned with operational energy during the building's use stage.

According to Thormark (2002), low-energy buildings contain higher embodied energy than conventional energy. Embodied energy means the energy used from the raw material extraction, transportation, and production. This raises the question of what constitutes true sustainability in the construction industry and how can construction management assist in reducing carbon emissions, both operational and embodied. Chen et al. (2019) has stated that sustainable building material selection is diffusely regarded as the simplest and most essential way of achieving sustainability. A solution could be to select a material that has the least environmental impact during procurement.

Most construction projects are budget-constrained, and sustainable options are known to require more initial investment, in other words, environmental concerns have not been a strong decision driver in material procurement.

If the decision makers are concerned with sustainable material selection, life cycle costing (LCC) is a popular approach that focuses on the financial element from the initial investment to the future energy bills. Elsewhere, there is the less frequently used Life Cycle Assessment (LCA) concept for selecting building components. Even though LCA is less common, it focuses on actual emissions not only during the operational stage but also from the raw material extraction and it measures emissions across six categories, including CO2. If used in the procurement decision-making process, LCA could contribute to reducing CO2 emissions.

Hence, this research aims to build a decision support system for sustainable building material procurement that incorporates LCA while including other traditional aspects which can help decision-makers find the best option for their project.

As for the material, the study focuses on building insulation because insulation is the main performer for energy conservation during the operation of a building.

1.2 **Research Objectives**

This research attempted to develop a decision support framework and software that users can easily utilize to find a sustainable insulation product and thickness among all those available on the market. The research objective was accomplished by achieving the following:

1. From the literature review, understand what drives major decisions, how sustainable the insulation materials are and how decisions are made.

2. Collect the data required to calculate LCA (Life Cycle Assessment) and LCC (Life Cycle Cost) and apply a Pareto search for minimum LCA and LCC for individual material options.
3. Find decision-making criteria, apply TOPSIS techniques, and build a decision support system that helps find the best insulation product and best thickness among the available products.
4. Build a software prototype that can comply with all the above for user applications.

1.3 **Expected Contributions**

This thesis intends to contribute to the following:

1. Application of operational energy and embodied energy concepts in the decision-making process
2. Integration of Pareto front and TOPSIS technique
3. Development of a selection tool that can be used in real market options
4. Development of a Project-specific selection tool

1.4 **Thesis Organization**

The thesis comprises of six chapters.

Chapter 1 introduces the topic’s background, defines the objectives of the study, and outlines the structure of the research.

Chapter 2 is the literature review. First, it ascertains common decision-making drivers in sustainable material selection before investigating previous sustainable material selection studies and their methodologies. It also covers life cycle environmental impact-related subjects, including Environmental Product Declaration and its definition by ISO and related calculation techniques.

Chapter 3 introduces the methodology adopted in this research. It explains the design of the framework, material selection methodologies, calculation assumptions, required data, and data collection.

Chapter 4 demonstrates the application of the methodology using a case study.

Chapter 5 verifies the method from various angles and validates the method by experts.

Chapter 6 includes the final summary, contribution, limitation of the study, and proposed future research.

Chapter 2 **Literature Review**

2.1 **Introduction**

A review of previous articles and research is necessary to establish the analytical foundations of this research. First, section 2.2 discusses what the main drivers of material selection are before investigating some previous studies, methodologies on material selection, as well as the technical understanding of sustainable buildings and insulation, such as how the thermal resistance of insulation can affect the building’s emissions and costs over its life cycle. A look at the ISO and EN standards for life cycle assessment is also required to build knowledge about the subject. This literature review is broken down as follows:

- Sustainable construction decision drivers and barriers
- Sustainable material selection making
- Sustainable insulation material and Embodied Energy
- Environmental Life cycle assessment (LCA)
- LCA application, Software, and Database
- Measuring Specific Data

2.2 **Sustainable construction decision drivers and barriers**

Many international environmental agreements have been signed between countries in an effort to stop global warming. The aim and purpose of such environmental policies seem straightforward and clear at the government level, but there is more to consider except goodwill when it comes to practice. This section discusses the forces pushing industries to green construction.

2.2.1 Regulation and LEED

The government's environmental policies are the guidance for green buildings, and designers and suppliers are dependent on governmental incentives for green innovation (Fu et al., 2020). Gbadebo and Ajibike (2019) also, found that regulations are a strong driver of environmentally sustainable construction among large construction firms. However, relying on only policies and regulations has its limitations. The limitation of regulatory-driven policy is that the line is drawn at the minimum. Therefore, these legislative measures of assessment cannot be the decision-making tool for sustainable building design (Burke & Kristen, 2008).

While compulsory government policies are the fundamental driver of sustainable construction, voluntary green building certificates promote sustainability further to a challenging level. There are a number of certifications for building sustainability. Among those, Leadership in Energy and Environmental Design (LEED) is the most widely used environmental assessment rating system in the construction industry today. As of 2020, there are more than 100,000 buildings and spaces participating in LEED across 176 countries and territories (Stanley, 2020). LEED acts as a green building market driver internationally. LEED requirements for credits are in line with the USA's Green Building Codes and the International Green Construction Code (IgCC). Internationally, LEED is flexible enough to adopt local regulations to their point system. USGBC issues four levels of LEED certificates in four levels: platinum, gold, silver, and certified. This makes the certification more onerous than the government's regulations. After fulfilling each category's necessary prerequisites, points from each credit are added up for each certification level. However, Burke & Parrish (2018) pointed out the limitation saying it is a measurement tool and not a design tool. Despite the criticism by Burke & Parrish (2018), it is true that sustainable building certification is pushing the sustainable choices further than the government's regulation.

2.2.2 *Economic aspect*

While government-driven regulations work as a sustainable construction driver to a certain extent, economic aspects of sustainability work as a barrier. The benefits of building sustainably may not be so readily perceived since most sustainable materials can be pricier than conventional ones. As a consequence, owners might hesitate to integrate sustainability strategies because of increased initial costs (Wao, 2017). Kats (2013, as cited in Li et al. 2020) claimed that it is approximately $ 3/ft2 to $ 9/ft2 more expensive to construct green buildings than conventional buildings. Research shows that the key barriers to sustainable construction are linked to initial costs. Wao (2017) reiterates that most believe sustainability has higher initial costs compared to ordinary buildings. The conventional, finance-based toolkits such as life cycle cost analysis, capital cost, and discounted cash flow continue to be the most frequently used tools in project investment decision-making practice. (Higham & Fortune, 2016)

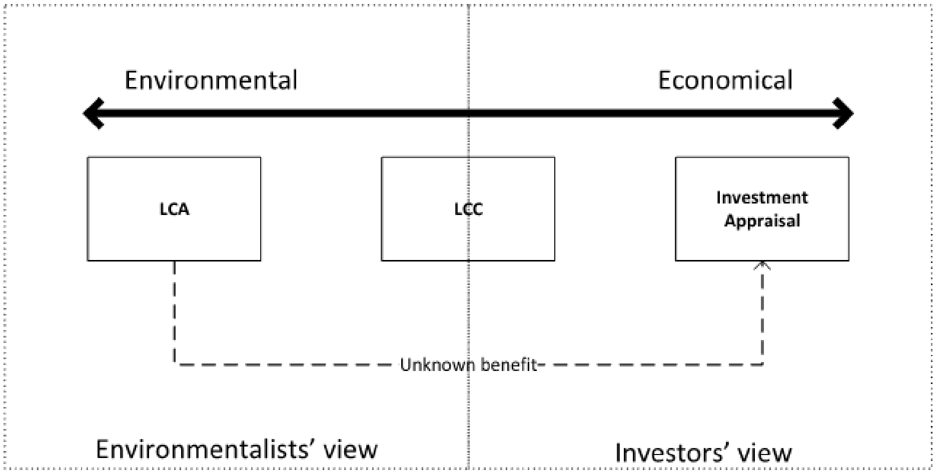


Figure 1 Sustainable Material Decision Drivers

2.2.3 Life Cycle Assessment (LCA) [1]

Masanet and Chang (2014) surveyed some 900 manufacturing and transport operations practitioners on their use of LCA as a decision support tool. Around 33% of the respondents intended to use LCA for decision-making in the context of manufacturing and transport operations. On the other hand, according to Stiel et al. (2016) some practitioners argue that they cannot incorporate LCA within their decision support systems due to the time and resources required as well as due to the current LCA software that is partially unsatisfactory.

2.3 **Sustainable material selection**

This section reviews previous studies for sustainable material selection. In the literature, various methodologies are employed, and four main streams can be found: 1) Multi-Criteria Decision-Making Techniques, 2) Multi-objective optimization, and 3) quantitative comparison of materials, and 4) integration of LCA and LCC. While introducing the various methodologies, some are examined in more detail for later application in Chapter 3.

2.3.1 Multi-Criteria Decision-Making Techniques

Multi-Criteria Decision-Making Techniques (MCDM) are a range of methods that evaluate multiple conflicting criteria to place alternatives in order, rank by weighting. MCDM can assess not only quantitative criteria but also qualitative criteria in combination with the Fuzzy set theory.

Burke & Parrish (2018) studied an effective way for integrating environmental product declarations (EPD) in sustainable material selection by using message sequence charts (MSC) a graphical language that visualizes communications between systems or entities. The MSC

---

[1] Section 2.5 elaborates further on LCA.

captures stakeholders' thoughts. This study's focus group showed that it would be practical to integrate EPD into the early design process.

Nofal & Hammad (2020) applied Fuzzy TOPSIS (a technique for order of preference by similarity to ideal solution) as an MCDM technique for selecting the optimal sustainable wall building material. The technique incorporated the linguistic preferences of experts who are suppliers, consultants, and contractors.

Chen et al. (2019) insisted the processes of LCA-based sustainable material selection methods are expensive, whereas MCDM methods are less expensive. The author did, however, point out that the MCDM methodology itself becomes the determining factor of its analysis. Santoyo-Castelazo & Azapagic (2014) also mentioned that MCDM is subjective but straightforward.

2.3.2 TOPSIS and Fuzzy TOPSIS

Jaini & Utyuzhnikov (2017) identified that there is no unique solution to a conflicting multi-criteria problem. Instead, the conflicting multi-criteria problem gives a set of Pareto solutions. There are many techniques related to the ranking of available alternatives presented by the Pareto solutions. Among those, TOPSIS has been widely used in MCDM due to its simplicity (Jaini & Utyuzhnikov, 2017).

2.3.2.a TOPSIS:

The TOPSIS method was developed by Hwang and Yoon (1981). This method is based on the concept that the chosen alternative should have the shortest Euclidean distance from the ideal solution and the farthest from the negative ideal solution. The ideal solution is a hypothetical solution for which all attribute values correspond to the maximum attribute values in the database comprising the satisfying solutions; the negative ideal solution is the hypothetical solution for which all attribute values correspond to the minimum attribute values in the database (Rao, 2007).

Suppose an MCDM problem with m alternatives, A1, ..., Am and n decision criteria, C1, ..., Cn. Each alternative is assessed with respect to the n criteria. All the performance ratings assigned to the alternatives with respect to each criterion form a decision matrix denoted by $X = (x_{ij})_{m\times n}$. Let $W = (w_1, w_2, ..., w_n)$ be the relative weight vector about the criteria, satisfying $\sum_{j=1}^{n} W_j = 1$. Then, the TOPSIS method can be summarized as follows (Vahdani et al., 2011)

Step 1: Normalize the decision matrix $X = (x_{ij})_{m\times n}$*. This can be represented as*

$$r_{ij} = x_{ij}/[\sum_{j=i}^{m} x^2{}_{ij}]^{1/2}, \text{ i= 1, 2…, m; j=1, 2…n,} \quad 1)$$

Where r_{ij}*is the normalized criteria rating.*

Step 2: Calculate the weighted normalized decision matrix $V = (vij)m\times n$*.*

$$v_{ij} = w_j r_{ij},\ i = 1, 2 \dots, m;\ j = 1, 2 \dots n, \quad 2)$$

Where w_{ij}*is the relative weight of the jth criterion and* $\sum_{j=i}^{n} w_j = 1$..

Categorization of attributes for cost/ benefit criteria

Step 3: Determine the ideal and negative-ideal solutions.

$$A^* = v_1^*, \dots, v_m^* = \{(max_j v_{ij} | j \in \Omega_b), (min_j v_{ij} | j \in \Omega_c)\},$$

$$A^- = v_1^-, \dots, v_m^- = \{(min_j v_{ij} | j \in \Omega_b), (max_j v_{ij} | j \in \Omega_c)\}, \quad 3)$$

where Ω_b *and* Ω_c *are the sets of benefit criteria and cost criteria, respectively.*

Step 4: Calculate the Euclidean distances of each alternative from the positive ideal solution and the negative ideal solution, respectively.

$$D^+ = \{(\sum_{j=1}^{n} (v_{ij} - v_j{}^*)^2\}^{0.5},\ i = 1,2, \dots, m,$$

$$D^- = \{(\sum_{j=1}^{n} (v_{ij} - v_j{}^-)^2\}^{0.5},\ i = 1,2, \dots, m, \quad 4)$$

Step 5: Calculate the relative closeness of each alternative to the ideal solution. The relative closeness of the alternative A_i *with respect to* A^* *is defined*

$$RC_i = D_i^{*}/(D_i^{*} + D_i^{-}), \quad i = 1,2,\ldots,m, \qquad 5)$$

Step 6. Rank the alternatives according to the relative closeness to the ideal solution. The bigger the RC_i, the more desirable the alternative A_i will be. The best alternative is the one with the greatest relative closeness to the ideal solution.

The conventional MCDM solutions assume all values are crisp numbers. In reality, the values can be crisp, fuzzy, or linguistic. Therefore, in fuzzy MCDM, the weights of the criteria and the performance of the alternative are converted to linguistic variables to tackle the problem. (Jaini & Utyuzhnikov, 2017)

One way to solve fuzzy MCDM problems is to reduce the fuzzy MCDM problem to defuzzification and solve it with a conventional MCDM method. The defuzzification process converts the fuzzy numbers into crisp values. The defuzzification process is essential in both ways since the MCDM solution must provide a crisp result. (Jaini & Utyuzhnikov, 2017)

2.3.2.b Fuzzy numbers:

The three most common types of fuzzy membership functions are monotone, triangle, and trapezoidal. However, the triangular fuzzy number is more convenient in application due to its simple calculation (H. Li et al., 2020). The following are the basic definitions and notations of fuzzy sets and fuzzy numbers summarized by Vahdani et al. (2011) from his literature review of Gupta (1991) and Ross (2004).

Definition. 1. *A fuzzy set $\tilde{A}$ in a universe of discourse, X is characterized by a membership function $\mu\tilde{A}(x)$ which associates with each element x in X a real number in the interval [0, 1]. The function value $\mu\tilde{A}(x)$ is termed the grade of membership of x in $\tilde{A}$.*

Definition. 2. *The triangular fuzzy numbers can be denoted as $\tilde{A} = (a_1, a_2, a_3)$, the membership function of the fuzzy number $\tilde{A}$ is defined as follows:*

$$\mu\tilde{A}(x) = \begin{cases} 0 & x < a_1, \\ (x - a_1)/(a_2 - a_1) & a_1 \le x \le a_2, \\ (a_3 - x)/(a_3 - a_2) & a_2 \le x \le a_3, \\ 0 & x > a_3. \end{cases} \quad 6)$$

Definition. 3. *A non-fuzzy number r can be expressed as (r, r, r). The fuzzy sum ⊕ and fuzzy subtraction ⊖ of any two triangular fuzzy numbers are also triangular fuzzy numbers; however, the multiplication ⊗ of any two triangular fuzzy numbers is only an approximate triangular fuzzy number. Given any two positive triangular fuzzy numbers,* $\tilde{A} = (a_1, a_2, a_3)$*,* $\tilde{B} = (b_1, b_2, b_3)$ *and a positive real number r, some main operations of fuzzy numbers* $\tilde{A}$ *and* $\tilde{B}$ *can be expressed as follows:*

$$\tilde{A} \oplus \tilde{B} = (a_1 + b_1, a_2 + b_2, a_3 + b_3),$$

$$\tilde{A} \ominus \tilde{B} = (a_1 - b_3, a_2 - b_2, a_3 - b_1),$$

$$\tilde{A} \otimes r = (a_1 r, a_2 r, a_3 r), \quad 7)$$

$$\tilde{A} \otimes \tilde{B} = (a_1 - b_3, a_2 - b_2, a_3 - b_1),$$

2.3.2.c Defuzzification of fuzzy numbers:

There are various defuzzification methods, and the following introduces the 2nd Weighted average method.

2nd Weighted Average Method.

For the triangular fuzzy number $\tilde{A} = (a_1, a_2, a_3)$*, the second weighted average method is a less computationally intensive method. The defuzzified value is defined as:(Jaini & Utyuzhnikov, 2017)*

$$x^* = \frac{a_1 + 2a_2 + a_3}{4} \quad 8)$$

2.3.3 Novel fuzzy modified TOPSIS

Vahdani et al. (2011) introduced a novel fuzzy modified TOPSIS method. The method finds the best alternative by considering both conflicting quantitative and qualitative evaluation criteria in real-life application with multi-judges and multi-criteria in a fuzzy environment.

In the paper, Vahdani et al. (2011) converted numbers for objective criteria into fuzzy triangular numbers set by multiplying the aggregate relative importance of the criteria which was initially expressed in linguistic terms. Once all criteria are weighted with the fuzzy set, then the values are defuzzified. Lastly, the paper applied TOPSIS skills to solve the rest.

2.3.3.a TOPSIS calculation:

Step1 : The aggregate the fuzzy ratings and weights are represented as $\tilde{x}_{ij}$*,* $\widetilde{w}_{ij}$

$$\tilde{x}_{ij} = (a_{ij}, b_{ij}, c_{ij}),\ \mathrm{k} = 1, 2, \dots, \mathrm{m},$$

$$\widetilde{w}_j = (w_{j1}, w_{j2}, w_{j3}),\ \mathrm{k} = 1, 2, \dots, \mathrm{m},$$

9)

$$a_{ij} = \frac{1}{k}\sum_{k=1}^{k} a_{ijk}, \quad b_{ij} = \frac{1}{k}\sum_{k=1}^{k} b_{ijk}, \quad c_{ij} = \frac{1}{k}\sum_{k=1}^{k} c_{ijk}$$

$$w_{j1} = \frac{1}{k}\sum_{k=1}^{k} w_{jk1}, \quad w_{j2} = \frac{1}{k}\sum_{k=1}^{k} w_{jk2}, \quad w_{j3} = \frac{1}{k}\sum_{k=1}^{k} w_{jk3}$$

Where K *is the number of subjective criteria*

Step 2: Compute the normalized decision matrix for r_{ij} *and* $\tilde{r}_{ij}$ *(for objective and subjective ratings)*

$$r_{ij} = x_{ij}/[\textstyle\sum_{j=i}^{m} x^2{}_{ij}]^{1/2},\ i = 1, 2 \dots, m;\ j = 1, 2 \dots n,$$

10)

$$\tilde{r}_{ij} = \left(\frac{a_{ij}}{e_j^*}, \frac{b_{ij}}{e_j^*}, \frac{c_{ij}}{e_j^*}\right),\ i = 1, 2 \dots, m;\ j = 1, 2 \dots n,$$

$$e_j^* = \sqrt{\sum_{i=1}^{m} c_{ij}^2}$$

Step 3: Calculate the fuzzy weighted normalized decision matrix $\widetilde{V} = [\tilde{v}_{ij}]_{m\times n}$. *The fuzzy weighted normalized decision matrix is calculated by multiplying each column of the matrix by the fuzzy weight* $(\widetilde{w}_j)$, *which uses the equation* $\widetilde{w} \otimes \tilde{r}$ *and* $\widetilde{w} \otimes r$ *Thus,*

$$\tilde{v}_{ij} = \widetilde{w}_j \tilde{r}_{ij},\ i = 1, 2 \ldots, m;\ j = 1, 2 \ldots k-1,$$

11)

$$\tilde{v}_{ij} = \widetilde{w}_j r_{ij},\ i = 1, 2 \ldots, m;\ j = k, \ldots, n,$$

Step 4. Defuzzify fuzzy numbers.

Step 5. From the defuzzified values, determine the ideal and negative-ideal solutions by using equation 3 (see section 2.3.2.a) and calculate the Euclidean distances of each alternative from the positive ideal solution and the negative ideal solution with equation 4) above. Finally, calculate the relative closeness of each alternative to the ideal solution for ranking.

2.3.4 Criteria selection

Table 1 summarizes the previously used criteria in other sustainable wall material selection studies. The criteria can mainly be categorized into the following: cost-related, sustainability-related, and technical performance-related.

Selection Objective	**EXTERNAL WALL INSULATION MATERIAL** (Ruzgys et al., 2014)	**BUILDING WALL MATERIAL** (Nofal & Hammad, 2020)	**BUILDING ENCLOSURE MATERIAL** (Mahmoudkelaye et al., 2018)
Criteria1	Price with VAT	Potential for recycling and reuse	Material & Construction Cost
Criteria2	Duration of works	Amount of waste during use	Transportation Cost
Criteria3	Payback period	Cost per m2	Service & Maintenance Cost
Criteria4	Energy losses	Labor productivity	Overhead Cost

Criteria5	Water vapor diffusion	Fire resistance	Energy Cost (during operation)
Criteria6		Energy-saving and thermal insulation	Market value
Criteria7			Weight
Criteria8			Chemical Resistant
Criteria9			Water Resistant
Criteria10			Fire Resistance Strength
Criteria11			Life Expectancy
Criteria12			Embodied Energy
Criteria13			Loss Factor
Criteria14			Energy Saving & Thermal Insulation
Criteria15			Water Use
Criteria16			Safety During Construction
Criteria17			Indoor Air Quality Human Health
Criteria18			Fire Immunity
Criteria19			Shock Immunity
Criteria20			Global Warming
Criteria21			Ozone Depletion
Criteria22			Acidification
Criteria23			Photo-Chemical
Criteria24			Smog
Criteria25			Eutrophication
Criteria26			Land Occupation
Criteria27			Recycling/Reusing
Criteria28			Potential Air pollutants
Criteria29			Ecological Toxicity
Criteria30			Social, Religious, And Cultural Identity
Criteria31			Aesthetics
Criteria32			Labor Availability
Criteria33			Designer’s Knowledge

Table 1 Criteria for sustainable wall selection

2.3.5 Optimization

While MCDM finds the best solutions among the candidate materials, optimization methods can help find a material’s optimum design features. The following are from previous studies.

Aleixo et al. (2018) developed a tool that uses multidisciplinary and multiobjective optimization methodology for the design of sustainable aircraft structures by optimizing the trade-offs between technical, economic, and environmental performance indicators. The indicators for economic and environmental performance were the life cycle cost and life cycle CO2 emissions. For technical indicators, aircraft parts were analyzed for functionality.

Leite et al. (2015) used Direct MultiSearch (DMS) optimization for material selection considering manufacturing costs and weight reduction with structural isoperformance. According to Leite et al. (2015), due to the discrete and combinatorial nature of the problem (commercially available materials), the optimization algorithm has to deal with non-differentiable objective functions and constraints (e.g., minimization of conflicting objectives subjected to some technological constraints). To solve this problem, the authors adapted the DMS solver for multiobjective optimization problems, which works with real variables as discrete variables.

Castro-lacouture et al. (2009) studied the most efficient way to achieve a Leadership in Energy and Environmental Design (LEED) score with the lowest budget when selecting the best material. The study identified all LEED points related to the material selection problem and maximized the points awarded with a mixed-integer model. The research shows that budget can be a determining factor for a building's level of environmental sustainability.

Florez & Castro-lacouture (2013) continued the Castro-lacouture et al. (2009) research and added one more dimension to the preceding research by considering visual perceptions. They have suggested a method to quantify visual perceptions based on the creativity measurement instrument developed by Horn and Salvendy and other authors. The decision-making process first seeks out the optimal solution for environmental requirements, budget constraints, and LEED requirements; then, the decision-maker determines if the optimal solution is satisfactory or not. If not satisfied, the quantified perceptions of sustainability for the construction scores are converted to loads. Then the loads are multiplied by each material—this information is fed back into the system to find the optimal solution.

2.3.5.a Multi-objective optimization (MOO):

Single-objective optimization problems may have a unique optimal solution; however, real-life problems often involve multiple objectives. The multi-objective problems (as a rule) present a potentially uncountable set of solutions. A decision-maker has to choose one or more solutions by selecting one or more vectors. The decision-maker usually selects an acceptable solution belonging to the Pareto front (Chiandussi et al., 2012).

Custódio et al., 2011 introduced a constrained MOO problem as follows.

$$find\ x = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_s \end{Bmatrix} \qquad 12)$$

which minimizes:

$$min\ \ F(x) = (f_1(x), f_2(x), \dots\ f_k(x))^T \qquad 13)$$

Subject to:

$$\begin{aligned} g_{l_1}(x) &\leq 0, \quad l_1 = \{1,2,\cdots,m_1\} \\ h_{l_2}(x) &= 0, \quad l_2 = \{1,2,\cdots,m_2\} \end{aligned} \qquad 14)$$

where s is the number of design variables, k is the number of objective functions to be minimized, and $m1$ $m2$ are the number of constraint equations.

The concept of Pareto dominance is crucial for comparing any two points (Custódio et al., 2011)

$$\mathbb{R}^m_+ = \{z \in \mathbb{R}^m : z \geq\ 0\}, \qquad 15)$$

defined by $F(x) \prec_F F(y) \Longleftrightarrow F(y) - F(x) \in \mathbb{R}^m_+ \setminus \{0\}$

Given two points x, y in Ω, we say that $x \prec y$ (x dominates y) when $F(x) \prec_F F(y)$. We will also say that a set of points in Ω is nondominated (or indifferent) when no point is dominated by another in the set. The Pareto front is the set of points in Ω nondominated by any other one in Ω. (Custódio et al., 2011)

2.3.6 Quantitative comparison of candidate materials

Unlike the above methods, this method directly compares the candidate materials' environmental criteria through life cycle tools. Due to its quantitative nature, technical calculations are carried out within a specific case.

Hafner and Storck (2019) and Takano et al. (2015) carried out the life cycle assessment for exterior envelope and partition wall materials in a building. They found that the external components such as sheathing, exterior cladding, and thermal insulation have relatively greater environmental emissions than the inner components categories. Ozturk et al. (2019) did a life cycle cost (LCC) analysis based on cooling degree-day (CDD) for refrigerated warehouses to determine the best insulation thickness.

Like the above two cases, when it comes to quantification, as Giorgi et al. (2019) pointed out, Life Cycle tools are often applied in a 'downstream approach' (construction and demolition waste management studies) rather than 'upstream approach' (design approaches).

2.3.6.a Integrating LCA and LCC

The following frameworks integrate LCA and LCC, developed as decision-making tools for broader subjects not limited to construction. The methods are 1) create an indicator: LCC divided by LCA, 2) convert LCA to monetary terms.

1) Eco-efficiency

Miah et al. (2017) investigated the integration of Life Cycle Assessment (LCA) and Life Cycle Costing (LCC) and suggested a hybrid framework in 4 stages. The hybrid framework is a mix of two parts. First, find the best alternative by combining TOPSIS and AHP. Second, optimize LCC and LCA through an index called 'Eco-efficiency' where the index is LCC divided by LCA (Life cycle environmental impact). The followings are the summaries of the four stages from Miah et al., (2017) 's paper.

> Stage 1 - Decision-making perspective and goal: Take perspectives from an investor's and non intestor’s viewpoint. Define the goal and scope of analysis for both perspectives and provide a route to navigate the framework.
>
> Stage 2 - Systems analysis: Choose a system between optimization and evaluation and carry out hybrid LCA and LCC.
>
> Stage 3 - System integration: Depending on whether different decision makers' preference needs to be taken into account, choose between the hybrid MCDA (best alternative) method or Energy Efficiency index (optimization).
>
> Stage 4 - Graphical interpretation and recommendations

2) Converting LCA to monetary terms

Kim et al. (2013) applied the AHP and the CO2 conversion method to integrate LCC and LCA. The CO2 conversion method is used when the analysis must yield an exact economic value including specific environmental pollution prices. Meanwhile, the use of AHP is recommended when perspectives on environmental pollution are counted. The most significant difference between the CO2 conversion method and AHP is that the former places an economic value on the environmental impact.

3) Integration of sustainability indicators via a multicriteria decision

Santoyo-Castelazo & Azapagic (2014) also suggested a framework in steps with a case demonstration that comprises scenario analysis, life cycle assessment, life cycle costing, social sustainability assessment, and multicriteria decision analysis. The demonstrated case is a selection of energy systems. For the indicators, LCA is used for assessing environmental sustainability, life cycle costing for the economics, and various social indicators for social sustainability. The social indicators are security and diversity of supply, public acceptability, health and safety, and intergenerational issues. The following are the steps:

1. selection of environmental, economic, and social indicators to be used for measuring sustainability
2. selection and specification of energy technologies
3. definition of scenarios and the time horizon
4. environmental, economic, and social assessment on a life cycle basis
5. integration of sustainability indicators via a multicriteria decision analysis to determine the most sustainable options for the future.

2.4 **Sustainable insulation material and Embodied Energy**

This section will discuss environmentally sustainable building, sustainable building insulation material, and insulation thickness.

2.4.1 Building sustainability

In general, there are two well-known types of sustainable building forms; the first is the 'Passive house,' the second is the 'Net Zero Energy house.' The passive house is defined by the Passive House Institute (PHI), which was established in 1996 in Germany. The PHI's suggested standard is that a building uses less than 1.5L of oil or 1.5 m3 of gas to heat one square meter (15KWH/m2) of living space for a year, which is 75% to 90% of energy savings compared to other buildings. The saving is achieved through efficient use of various elements such as the sun, internal heat sources, and heat recovery (Feist, n.d.). Therefore, the U-values (insulating performance) of external walls, floor slabs, and roof areas have critical importance.

On the other hand, the concept of the net-zero/low energy house (NZE house) is defined as "a home which produces enough renewable energy to meet its own annual energy consumption requirements, thereby reducing the use of non-renewable energy in the residential building sector” (Li, Gül, Yu, Awad, & Al-Hussein, 2016 as cited in H. X. Li et al., 2018). These two concepts are often used together in many sustainable buildings.

The above ‘passive’ and ‘NZE’ are concepts about minimizing operational energy. Meanwhile, the energy spent for the completion of a building is called embodied energy. Embodied energy accounts for up to 46% of the life cycle energy use (service life of 50 years) in low-energy buildings and up to 38% in conventional buildings (Sartori & Hestnes 2006 as cited in Takano et al., 2015). Ramesh et al. (2010) also insisted that ‘in the case of a self-sufficient house, though its operating energy is zero; its embodied energy is so high that it exceeds the life cycle energy of some of the low energy cases.’ In 2018, 11% of global energy-related CO2 emissions were attributed to manufacturing building materials and components (International Energy Agency, 2019 as cited in Opher et al., 2021). Therefore, to understand what is better for the environment, it is necessary to consider both sides: the embodied energy and operational energy.

2.4.2 Sustainable material for building

If only looking into the material life cycle rather than the building life cycle, Hafner and Storck (2019) said that the material production stage has a greater influence on environmental impact than the operation and maintenance stage. Once the material is installed in a building, it will last with the building except for replacement and maintenance.

The building’s operational energy is related to the building envelope’s thermal conductivity (the R-values) and the material’s thickness—especially if the buildings are located in the region where the numbers of Heating Degree Day and Cooling Degree Day are high.

Building codes have been geared towards increasing the R-values to gain better energy efficiency performance (Raouf & Al-Ghamdi, 2020). Along with the building code, LEED-Energy and Atmosphere (EA) category sets out its baseline requirements as the American

Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) 90.1 requirements as its baseline requirements (Raouf & Al-Ghamdi, 2020).

In ASHRAE 90.1 (2013), 'the rated R-value of insulation' is defined as the insulation's thermal resistance specified by the manufacturer in m2·K/W units at a mean temperature of 24°C'. For the exterior envelope, ASHRAE 90.1 stipulated the R-values required by the climate zones for conditioned spaces. The number range of the climate zone classifier is from 1 to 8; the lower the climate zone, the hotter the place is, where Alberta falls in zone 6 to 8 (BILD Alberta, n.d.).

2.4.3 Insulation thickness

Commonly, there are two ways to determine the insulation thickness. The first one finds the optimum economic thickness, and the other one references the requirements written in international or local codes. ASHRAE 90.1 provides tables of pre-calculated assembly U-factors for typical construction assemblies in its Appendix A, which specifies the maximum U-factor for each envelope element. Materials with lower thermal conductivity allow less thickness for the same R-value but are usually more expensive, extending the payback period.

Kallioğlu et al. (2020) introduced equations that calculate LCC for a building's unit external surface, which facilitates the finding of an insulation material's optimum economic thickness. The authors found the optimum economic thickness considering the LCC of a proposed building. For the LCC estimation of a building, the fuel's future cost is calculated with an estimated interest rate. The following are some basic equations for heat-loss-related calculations from Kallioğlu et al. (2020) 's research paper.

For a typical wall, U (W/m2·K) expressing the total heat transfer coefficient is calculated by equation 16.(Kallioğlu et al., 2020)

$$U = \frac{1}{R_i + R_w + R_{izo} + R_o} \quad 16)$$

The thermal conductivity is expressed as R-value (resistance to heat transfer) or U-value (heat transfer), where the relationship of two is 'U = 1/R and R = 1/U'.

R_i and R_o in Equation 16 indicate the thermal resistance of the inner and outer surfaces, respectively; R_w is the thermal resistance of the uninsulated wall layer; R_{izo} is the thermal resistance of the insulating material and is calculated by using equation 17. In equation 17, x is the thickness of the insulation material, and k is the insulation material's thermal conductivity coefficient. (Kallioğlu et al., 2020)

$$R_{izo} = \frac{x}{k} \quad 17)$$

The heat loss from the outer wall's unit surface is calculated using equation 18 below. (Kallioğlu et al., 2020)

$$q = U \cdot \Delta T \quad 18)$$

In equation 18), U (W/m2K) is the total heat transfer coefficient. ΔT (C°) is the temperature difference. The unit surface's annual heat loss is calculated using U and the number of degree days (NDD) in equation19. (Kallioğlu et al., 2020)

$$q_a = 86400 \cdot NDD \cdot U \quad 19)$$

Where q_a annual heat loss in the unit area (J/m2-year)

The annual energy requirement E_A (J/m2-year) required for heating is obtained from equation 20 in which the annual unit heat loss is divided by the system efficiency (Kallioğlu et al., 2020).

$$E_a = \frac{86400 \cdot NDD \cdot U}{\eta} = \frac{86400 \cdot NDD}{(R_{T.W.} + R_{insulation}) \cdot \eta} \quad 20)$$

Where η is heating system efficiency, $R_{T.W.}$ *is the sum of* R_i*,* R_w*,* R_o *(m2K/W),* $R_{insulation}$

The amount of fuel consumed per year m_{fA} (kg/m2-year) is calculated through equation 21. (Kallioğlu et al., 2020)

$$\mathrm{m_{fA}} = \frac{86400 \cdot \mathrm{NDD}}{(R_{T.W.} + R_{insulation}) \cdot \eta \cdot \mathrm{H_u}} \quad 21)$$

Where H_u *is the heating value of the fuel (J/kg; J/m3; J/kwh)*

The annual energy cost C_{AH} (\$/m2-year) used to heat the unit area is calculated using equation 22. (Kallioğlu et al., 2020)

$$C_{A.H} = \frac{86400 \cdot \mathrm{HDD} \cdot C_f}{(R_{T.W.} + R_{insulation}) \cdot \eta \cdot \mathrm{H_u}} \quad 22)$$

Where C_f *is the price of the fuel (\$/kg; \$/m3)*

2.4.4 Life Cycle Cost (LCC)

LCC is the economic assessment methodology for selecting the most cost-effective alternative over a particular time frame, considering its initial cost (construction), operational cost, and maintenance cost (Kang 2017 as cited in Giorgi et al., 2019). Compared to the conventional economic decision tool, making a decision based on LCC may mean paying more upfront(Perera et al., 2009). To reduce operational energy, LCC is used as a materials decision-making tool (Giorgi et al., 2019).

Kallioğlu et al. (2020) expressed the LCC of insulation material as follows.

$$C_T = C_A \cdot PWF + C_i \cdot x \quad 23)$$

Where C_T *is the total cost (\$),* C_A *is annual energy cost (\$/m2-year),* C_i *is the insulation cost in (\$/m3), x is the insulation material's thickness.*

The parameter (PWF: Present Worth Factor) is a uniformly distributed annual worth factor for a specified time term (N).

$$PWF = \frac{(1+r)^N - 1}{r \cdot (1+r)^N} \quad 24)$$

The actual interest rate (r) used in the calculation of the PWF price is calculated with respect to the two separate conditions, using equation 25, depending on the inflation rate (g) and the interest rate (i) (Kallioğlu et al., 2020)

$$if\ i > g\ then\ r = (i-g)/(1+g);\ if\ g > i\ then\ r = (g-i)/(1+i) \quad 25)$$

2.4.5 *Various insulation material and wall type*

Raouf & Al-Ghamdi (2020) study mentioned that 60% of the European market is dominated by inorganic fibrous materials such as stone wool (RW) and glass wool (GW), whereas oil derived foamy organic materials hold 30% of the market; these include extruded polystyrene (XPS), expanded polystyrene (EPS), and polyurethane (PUR). According to NAIMA Canada (2018), there are four main types of insulation products on the market: fiberglass, rock wool or slag wool, cellulose, and spray foam. These are supplemented by, extruded polystyrene (XPS), expanded polystyrene (EPS), polyisocyanurate (PIR), and polyurethane (PUR).

The different physical properties have different uses. For example, boards are commonly used in continuous sheathing and under slab applications, while blanket and spray type insulations are often used for external walls and roofs.

2.4.5.a Wall Type:

Steel and wooden stud frames are commonly used in North America as external wall frames. For the cavity of these frames, batt types of insulation are common. Some batt insulation may be rated as R-19; however, effective thermal resistance may be as much as 35% less than the rated cavity insulation due to the wood studs and other framing members. Therefore, adding insulation sheathing provides significant increases in the effective R-value without substantially increasing the wall thickness (Building Science Corporation, 2007). The following refers to insulation installation between steel and wooden frames introduced in the ASHRAE 90.1:

a.Standard framing: Steel stud framing at 400 mm on center with cavities filled with 400 mm wide insulation for both 89 mm deep and 152 mm deep wall cavities.

b.Advanced framing: Steel stud framing at 600 mm on center with cavities filled with 600 mm wide insulation for both 89mm deep and 152 mm deep wall cavities.

c.Standard framing: Wood framing at 400 mm on center with cavities filled with 368 mm wide insulation for both 89 mm deep and 140 mm deep wall cavities. Doubleheaders leave no cavity. Weighting factors are 75% insulated cavity, 21% studs, plates, sills, and 4% headers.

d.Advanced framing: Wood framing at 600 mm on center with cavities filled with 572 mm wide insulation for both 89 mm deep and 140 mm deep wall cavities. Doubleheaders leave uninsulated cavities. Weighting factors are 78% insulated cavity, 18% studs, plates, sills, and 4% headers.

e.Advanced framing with insulated headers: Wood framing at 600 mm on center with cavities filled with 572 mm wide insulation for both 89 mm deep and 140 mm deep wall cavities. Double header cavities are insulated. Weighting factors are 78% insulated cavity, 18% studs, plates, sills, and 4% headers.

Material	**Ranges R_{SI}/25.4mm(R/in.)**	**Design spec or average R_{SI}/25.4mm(R/in.)**
Polyurethane closed-cell spray foam	0.97 to 1.14 (R-5.5 to 6.5)	1.06 (R-6)
Polyurethane board	0.97 to 1.2 (R-5.5 to 6.8)	1.06 (R-6)
Extruded polystyrene board (XPS)	0.88 (R-5)	0.88 (R-5)
Polyisocyanurate spray foam	0.85 to 1.46 (R-4.8 to 8.3)	0.88 (R-5)
High-density glass fibre board	0.63 to 0.88 (R-3.6 to 5)	0.7 (R-4)
Expanded polystyrene board – Type I (EPS)	0.67 (R-3.8)	0.67 (R-3.8)
Expanded polystyrene board – Type II (EPS)	0.7 to 0.77 (R-4 to 4.4)	0.7 (R-4)
Glass fibre roof board	0.67 (R-3.8)	0.67 (R-3.8)
Cementitious foam	0.69 (R-3.9)	0.69 (R-3.9)
Cotton fibre batt	0.67 (R-3.8)	0.67 (R-3.8)
Cork	0.65 to 0.67 (R-3.7 to 3.8)	0.65 (R-3.7)
Polyurethane open-cell spray foam	0.63 to 0.67 (R-3.6 to 3.8)	0.63 (R-3.6)

Polyurethane open-cell foam, poured	0.7 (R-4)	0.7 (R-4)
Cellulose fibre, wet sprayed	0.53 to 0.67 (R-3 to 3.8)	0.63 (R-3.6)
Cellulose fibre, blown, settled thickness	0.53 to 0.67 (R-3 to 3.8)	0.63 (R-3.6)
Mineral fibre batt	0.53 to 0.7 (R-3 to 4)	0.6 (R-3.4)
Wood fibre	0.58 (R-3.3)	0.58 (R-3.3)
Mineral fibre, loose fill, poured	0.44 to 0.65 (R-2.5 to 3.7)	0.58 (R-3.3)
Glass fibre batt	0.55 to 0.76 (R-3.1 to 4.3)	0.56 (R-3.2)
Glass fibre, loose fill, poured	0.39 to 0.65 (R-2.2 to 3.7)	0.53 (R-3)
Mineral fibre, loose fill, blown	0.51 to 0.56 (R-3 to 3.8)	0.53 (R-3)
Glass fibre, loose fill, blown	0.48 to 0.63 (R-2.7 to 3.6)	0.51 (R-2.9)
Fibreboard (beaverboard)	0.41 (R-2.3)	0.41 (R-2.3)
Mineral aggregate board (Insulbrick)	0.41 to 0.7 (R-2.3 to 4)	0.46 (R-2.6)
Wood shavings	0.18 to 0.53 (R-1 to 3)	0.42 (R-2.4)
Vermiculite*	0.37 to 0.41 (R-2.1 to 2.3)	0.38 (R-2.2)
Compressed straw board	0.35 (R-2.0)	0.35 (R-2.0)
Eel grass (seaweed) batt	0.53 (R-3)	0.53 (R-3)
Cedar logs	0.18 (R-1)	0.18 (R-1)
Softwood logs (other than cedar)	0.18 to 0.25 (R-1 to 1.4)	0.22 (R-1.25)
Hardwood logs	0.12 (R-0.7)	0.22 (R-1.25)
Straw bale	0.23 to 0.28 (R-1.3 to 1.6)	0.26 (R-1.45)

Table 2 R_{SI}/ R-value of various materials (CleanBC Better Homes, n.d.)

2.5 **Environmental Life cycle assessment (LCA)**

This section introduces the guidelines for assessing material's sustainability for its life cycle. The applicable guidance rules for 'Life Cycle Assessment' by 'International Organization for Standardization (ISO)' and equivalent 'European Standards (EN)' are ISO 14040:2006, ISO 14044:2006, and EN 15804 and the recent additions of EN 15978 and ISO 21930.

2.5.1 *LCA introduction*

Environmental Life cycle assessment (LCA) is a standard methodology to evaluate the environmental impacts of products and services across the life cycle (Miah et al., 2017; ISO 14044:2006). Also, LCA is the cradle-to-grave quantification of potential environmental impacts of products or services. The LCA can be separated into three different life cycle stages for construction works and services, according to EN 15804 (BSI, 2014) .

- Upstream processes (from cradle-to-gate): module A1-A3

- Core processes (from gate-to-gate): module A4-A5
- Downstream processes (from gate-to-grave): B1B7, C1-C4

The difference in the LCA of construction from other products is that the product (building) has an operational period before its end of life. Ramesh et al. (2010), in their literature review, conclude that operating energy has a major share (80-90%) in life cycle energy use of buildings followed by embodied energy (10-20%). In contrast, demolition and other process and other process energy have negligible or little share in life cycle energy.

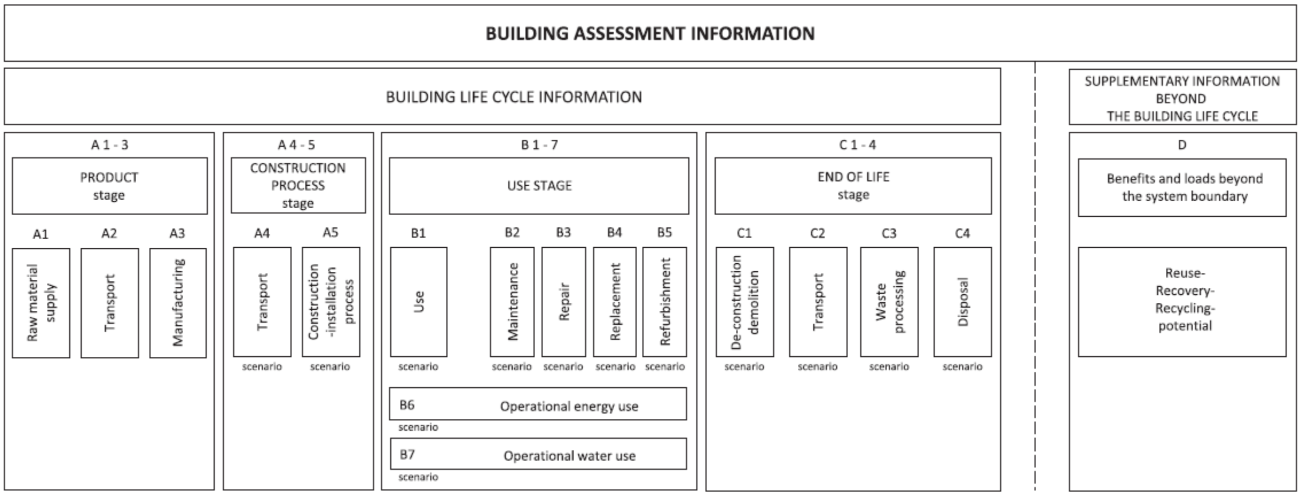


Figure 2 Building Life Cycle Stages (EN 15978)

2.5.2 LCA methodologies

There are three different LCA methods. They are (1) process LCA (P-LCA), (2) Economic-Input-Output LCA, and (3) hybrid LCA. (Miah et al., 2017). According to Crawford (2011), different LCA methods show different embodied energy levels for the same residential building.

Process-based LCA is the most detailed method and is usually applied to a particular process or industry (Hammond & Jones, 2008). The benefits of such approaches are that the Life Cycle Inventory (LCI) data are very accurate and specific. However, due to the high number of processes existing in a product life cycle, the practicality of accounting for all processes can be a time-consuming and resource-intensive procedure (Finnveden et al., 2009 as cited in Miah et al., 2017).

Next, Economic - input-output (EIO) LCA was originally developed by economists. Many countries periodically produce inter-industry tabular datasets depicting what each industrial category sells to and buys from other industries. Such tables can be converted from monetary values to yield data on an energy basis. The sum of direct energies for a particular industry then adds to the embodied energy in specific outputs (products) of that industry presented in terms of what is commonly known as 'energy intensities' (KJ/$ of product). (Hammond & Jones, 2008)

Miah et al. (2017) pointed out that an EIO-LCA method can offer fast analysis to identify environmental hotspots. However, Miah et al. (2017) also said the method is less accurate than the process-based LCA.

Lastly, there is the Hybrid LCA which is the most commonly used analysis method. It combines the Input-Output method and Process energy analysis to adjust the coverage and accuracy of the result. In a hybrid LCA method, the P-LCA methodology is combined with an EIO-LCA methodology to better reflect the true system boundary better while compensating for their respective limitations (Miah et al., 2017).

2.5.3 EPD and PCR

The ISO and EN established LCA standards for manufacturers, enabling them to publish LCA documents for their products.

2.5.3.a Environmental Product Declaration (EPD):

An Environmental Product Declaration (EPD) is an independently verified and registered document that communicates transparent and comparable information about a product's life-cycle environmental impact (EPD International AB, 2019). It is a voluntary declaration of the life-cycle environmental impact – having an EPD for a product does not imply that the declared product is environmentally superior to alternatives. (EPD International AB, 2019)

Under the 'General standard ISO 14020, Environmental Labels and Declarations-General Principles', the following three are the voluntary environmental labels.

- *Type I Standard ISO 14024, "Environmental Labels and Declarations-Type I Environmental Labelling- Principles and Procedures."*
- *Type II Standard ISO 14021, "Environmental Labels and Declarations-Type II Labelling -Self-declared Environmental Claims"*
- *Type III Standard ISO 14025, "Environmental Labels and Declarations- Type III Environmental Declarations-Principles and Procedures"*

Among the above, the Type III EDP is an independently verified and registered document. The Environmental Product Declaration (EPD) is defined by International Organization for Standardization (ISO) 14025 as a Product with third-party certification which "quantifies environmental information on the life cycle of a product to enable comparisons between products fulfilling the same function" (EPD International AB, 2019). An EPD is compiled based on a product category rule (PCR). A PCR is a set of rules, requirements, and guidelines for developing Type III declarations for products with similar functions. Program operators functioning under ISO 14025 publish PCRs for all kinds of products listed in the UN CPC - United Nations Product Category Classification.

Sustainable building certification systems such as LEED, Green Globes and BREEAM explicitly list or in some cases, require EPD, thereby encouraging many building materials manufacturers to pursue an EPD for their products. In LEED, the LCA-related points are for Materials and Resources (MR) credit: 'MR c1 Building Life Cycle Impact Reduction', and 'MR c2 Building Product Disclosure and Optimization-Environmental Product Declarations'. The point is awarded after filling the material EPDs according to requirements.

An EPD contains the following information.

- The product definition and information, including technical performance
- Information about the material and the material’s origin
- A description of the product’s manufacture
- Information on product processing
- Information about the in-use conditions
- LCA results
- Testing results and verifications

The PCR impact categories for EPDs are 'Global Warming Potential (kg CO2 eq.)', 'Acidification Potential (mole H+ eq.)', 'Eutrophication Potential (Kg N eq.)', 'Smog Creation Potential (kg O3 eq.)', 'Ozone Depletion Potential (Kg R11 eq.)', 'Primary Energy Demand (MJ)', 'Waste to Disposal (Kg)', 'Water Use(l)’, and ‘Waste to Energy (Kg)'.

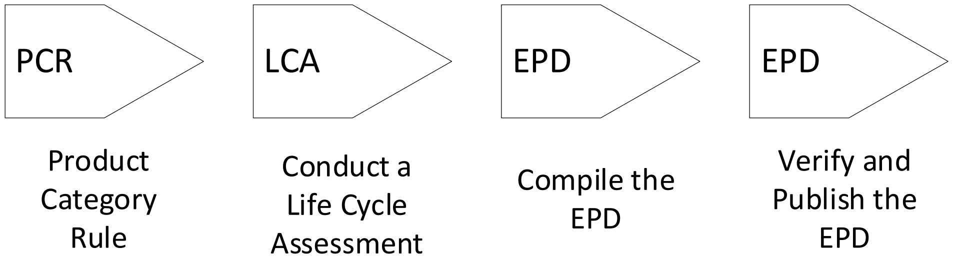


Figure 3 Publishing the EPD modified from ISO

2.5.3.b Construction Product category rule (PCR):

PCRs for building products are delineated in ISO 21930, refining those set out in ISO 14025 (GreenSpec, 2007). PCRs are developed openly and collaboratively, much like industry standards, and expire every three to five years leading to updates addressing relevant changes in the industry (UL, 2013). There are several repositories, and many PCRs have already been written for specific markets such as Europe or North America (GreenSpec, 2007).

According to the Construction Products PCR (EN-15804: A2), onsite construction LCA data is collected for each process stage.

- Upstream processes (from cradle-to-gate); manufacturer’s average or specific data [2] can be used, e.g., EPD, otherwise selected generic data, or other generic data.
- Core processes (from gate to gate); specific data gathered from the actual manufacturing plant where product-specific processes are carried out.
- Downstream processes (from gate-to-grave); selected generic data or proxy data.

As shown above, onsite data is called specific data. The following is regarding gathering specific data in the PCR.

Building PCR: specific data gathering

> ▪ *Transport to and from site*
>
> *Transports are calculated primarily from the construction products originating from the factory and secondarily from the supplier warehouse. The type of transport and transport distance should represent actual conditions on the market for which the EPD is valid. Therefore, transport of persons to and from the site shall not be included.*
>
> ▪ *construction and installation*
>
> *If possible, specific fuel consumption data and energy use on the site shall be collected in the life cycle inventory for construction and installation. Data on waste quantities and waste management at the construction site should also be collected and transports for waste.*

Even though there is a PCR for buildings due to the nature of the construction process’s uniqueness, EPDs are rare for buildings.

2.6 **LCA application, Software, and Database**

There can be two kinds of LCA databases: generic LCI databases and EPD databases. A few generic LCI databases are available in markets usually tailored to certain countries, while EPD

---

[2] *specific data (also referred to as "primary data" or "site-specific data") – data gathered from the actual manufacturing plant where product-specific processes are carried out, and data from other parts of the life cycle traced to the specific product (EN-15804: A2).

databases are still under development. This section discusses the traits and limitations of these databases and their applications with Building Information Modeling (BIM) technology.

2.6.1 *Generic database*

The usual criticism directed at generic databases is that these databases only contain an average of embodied energy and impact value for each material (Shadram et al., 2014). Shadram et al. (2014) also mentioned that while considering that each specific material is being manufactured in unique processes and different mechanisms in different factories, the outcome of the embodied energy and environmental impacts estimated by these LCA/LCEA tools is ambiguous. Another criticism is that using different LCA databases leads to the unreliability of the results (Gholizadeh et al., 2018).

Many commercial LCA software is designed for the whole building LCA process. Also, some BIM technologies are developed to adopt databases. These kinds of LCA-related technologies are mostly based on the generic dataset.

2.6.2 *EPD applications and databases*

Strömberg (2017) mentioned that EPDs for building materials and building parts had been increased; however, knowledge and experience of using such standardized climate calculations according to the EPD format is still very low. Although each EPD can be downloaded from a few repositories such as 'The International EPD System' website and the individual manufacturers, as pointed out by Shadram et al. (2014) EPDs are provided in HTML language or Excel/PDF data formats. EDP is not in a machine-readable format. Therefore, each product's EPD must be manually downloaded and read to compare. There is no EPD database ready for search and use. Integrating EPD with BIM has only recently started to be explored (Schwartz et al., 2016). For this problem, ISO is currently developing a database that unifies all EPDs, which can be utilized with BIM (ISO, 2020). There is still no general acceptance of EPDs as an industry-wide LCA specification (Strömberg, n.d.)

2.7 **Measuring Specific Data**

Some researchers have assumed that the construction process's environmental impacts are negligible. In contrast, others have indicated that the construction process's environmental impacts are underestimated (Guggemos and Horvath 2006, as cited in Fang et al., 2019). Whether environmental emissions from material logistics should be included in the construction process is still under discussion. Some researchers think the transportation of materials is part of the construction process because the energy used in transporting construction materials could take up around 20% of the transportation industry's total energy consumption (Smith et al. 2002 as cited in Fang et al., 2019).

On the other hand, according to EN-15804, environmental emissions during A4 to A5 (from gate to gate) should be gathered as specific data. Strömberg (2017) also mentioned the importance of A4 to A5 LCA, saying that it is not enough to calculate the climate impact only from the material production (Modules A1-A3), the use of environmental assessment only for A1-A3 in evaluating alternative contractor's designs may lead to sub-optimization (Strömberg, 2017b).

2.7.1 Material logistics CO2 emissions

The A4 stage is the transportation stage from the manufacturer to the site. The longer the shipping distance, the greater the CO2 emissions. Other than the sustainability concern, material shipping costs comprise 10%–20% of the total material cost. (P. H. Chen & Nguyen, 2019)

Chen and Nguyen (2019) developed a BIM plugin integrated with Google Maps or Apple Maps to calculate the distance between the manufacturer's place and the site to assess sustainability. The plugin takes information such as the delivery start date, truck type, truck capacity, number of trucks. The author’s tool can automatically identify and calculate the travel route of material transportation. This information is then used to calculate the transportation cost and expected material arrival time at the site and check compliance with the LEED Credit “MR” category (P. H. Chen & Nguyen, 2019).

2.7.2 Construction stage CO2 emissions

The A5 stage is the onsite construction stage. Due to each unique situation, the quantification of CO2 emissions at the construction stage cannot be standardized. However, Kawecki (2010) and Al-Hussein et al. (2009) observed the fuel used for equipment and site operation during the construction and converted it to CO2 emissions. CO2 emissions were calculated based on in-out methods; the total fuel consumption for equipment, electricity use, and gas consumption for the site’s operation were observed.

According to RS Means data’ 0721 Thermal Insulation’ data, among all insulation types, only ‘072129.10 Sprayed-On insulation’ needs equipment. Therefore, except for the transit within the site and general site operation, no special equipment is required to install insulation.

2.8 **Summary**

While government policy provides a minimum guideline and acts as a fundamental driver of sustainable building, voluntary green certificates, such as LEED, build on the government guideline and promote sustainable construction even further to a more rigorous level.

Conventional investment tools such as cost-benefit analysis and return period are still the preferred methods of many investors. Also, initial capital investment is still one of the most critical elements to consider for an investment appraisal. The operational, maintenance and disposal cost comes next, especially for short-term investors. Thus, even though some industry experts are interested in taking LCA into their decision-making process, the main barrier is the high cost.

For various sustainability related studies, including material selection, optimum design, or comparison, study methods are chosen between MCDM, optimization, quantification, or a combination. In all studies mentioned, the environmental criteria were the core criteria for all methods, and the economic criteria were the essential criteria in MCDM and optimization. Social/technical criteria were only in MCDM since the social/technical criteria are mostly subjective and qualitative. The most fundamental sustainability criterion was the environmental impact, and the most common indicator was GHG (Green House Gas)/GWP (Global Warming Potential, kg CO2 eq.). While some say that LCA is costly, the MCDM method is somewhat subjective. Currently, LCA is used more as a measurement tool rather than a design decision tool.

The core idea of the passive building is to reduce operational energy over the life cycle of the building. Although passive houses and the net-zero are more focused on operational energy, their embodied energy cannot be neglected as sustainable buildings could have higher embodied energy. Insulation thickness must be calculated to study the LCC and LCA of the insulation material. For insulation thickness, other than the required maximum U-factors method, economic thickness optimization methods can also be employed.

Most LCA quantification focuses on the building's operational energy; only a few LCA studies take a broader view including embodied emissions. The LCA's scope and stages are defined in ISO and EN. For material LCA, Material PCR is also available by ISO and EN, which instructs manufacturers to publish material specific EPDs. EPD helps in estimating a more accurate building LCA, as buildings encompass countless materials. Not all, but more and more manufacturers publish their product-specific EPD. Otherwise, generic LCI databases are available in some countries. The downside of using EPD is that EPDs are not readily usable in

BIM applications unless individually downloaded since there are currently no available databases.

Chapter 3 **Methodology**

This chapter has two parts. The first part explains the framework design. The second part contains a detailed explanation of the approach, including assumptions.

3.1 **Framework Design**

The framework is designed for user accessibility by having a user interface using Python Programming. After running the program, the software application should present the ranks of alternatives with informative graphics.

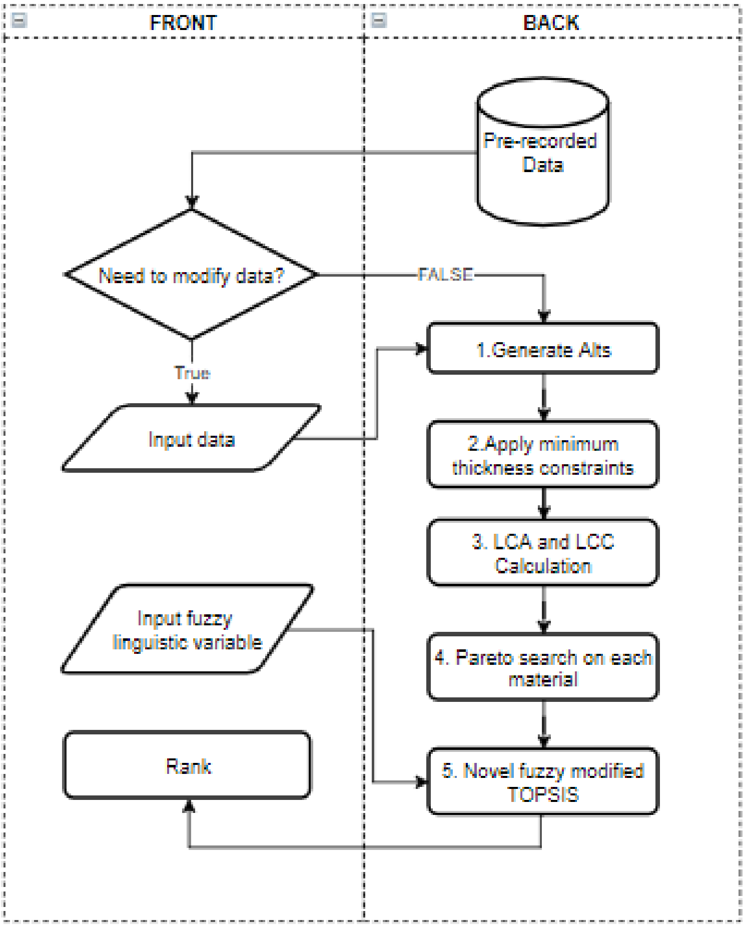


Figure 4 Framework Flow Chart

First, the project and material information are pre-stored for quick modification. The project information includes the following: Life Cycle Year (intended use years of the materials), Interest Rate & Inflation Rate (forecasted or current interest rate in trend), Heating Degree Day (of the region), the Heating value of fuel & Fuel Price & Fuel Emission (of specific fuel), Base wall structure R-value (of wall components except for the insulation). Also, each material's information is saved, including conductivity, price, distance (Km) from the factory, and GHG emission data by life cycle stage.

Second, the data is depicted on the user interface. Users can modify the input data if they want to edit, insert, or delete data. The program then generates alternatives based on available material thicknesses.

Third, by optimizing LCC and LCA, the program generates a nondominant Pareto front for each material.

Fourth, the program ranks the Pareto set based on TOPSIS linguistic preferences.

Finally, the program presents the result to the user.

In addition to developing the user interface calculation program, hand calculations are performed using Excel to verify the framework, the calculation is attached in Appendix A.

3.2 **Detailed method and assumption of calculation**

This section explains the details of the input data and assumptions (Section 3.2.1), the LCA and LCC calculation methodology (Section 3.2.2), pareto search technique (Section 3.2.3), and the novel fuzzy modified TOPSIS methodologies (Section 3.2.4).

To calculate LCA and LCC, two types of basic information are required: material information and project information. The material data is obtained from market research and EPDs published by manufacturing companies, and it includes the initial price for various thickness options as well as GHG emission quantities. The project data includes ASHRAE 90.1 (or local equivalent) fuel for heating, interest/discount rate, and base wall R-value. In addition, expert opinions are required later for TOPSIS.

Project Info:	Life Cycle Year, Interest rate, Inflation rate, Fuel Heating value, Fuel Price, Fuel Emission rate
	Base wall structure R-value
	Minimum R-value (maximum U-value) requirement from the ASHRAE 90.1 or local code (NECB).
Material Info:	Insulation material price and available thicknesses
	Distance from the manufacturer to the intended site base on EPD
Expert's Opinion:	Linguistic preference for fuzzy TOPSIS

Table 3 User Input and Knowledge Required for the Software

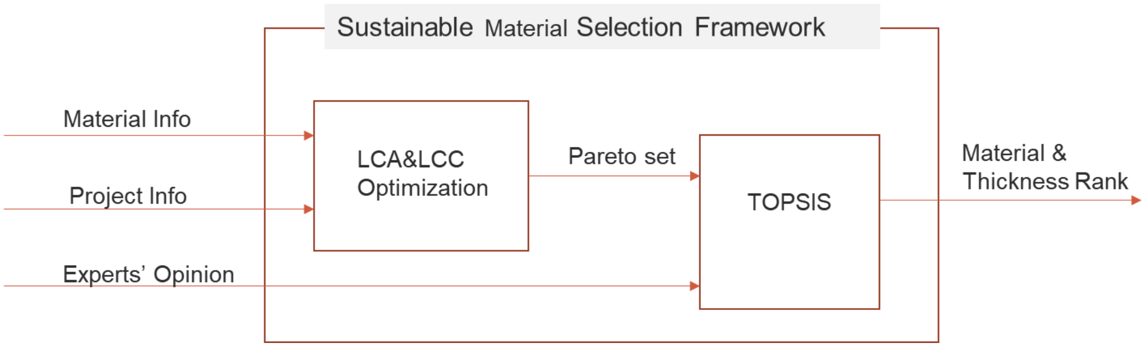


Figure 5 IDEF0 Diagram

3.2.1.a Base wall structure R-value:

The base wall structure can be made of steel, brick, or concrete, and the R-value of these base wall structures is required for LCA and LCC calculations. As shown in wall Figure 6, the study subject, insulation sheathing layer, is added on top of the base wall structure and under the external finishing. The total R-value of a complete wall can be calculated by adding the R-value of the insulation to the R-value of the rest of wall components.

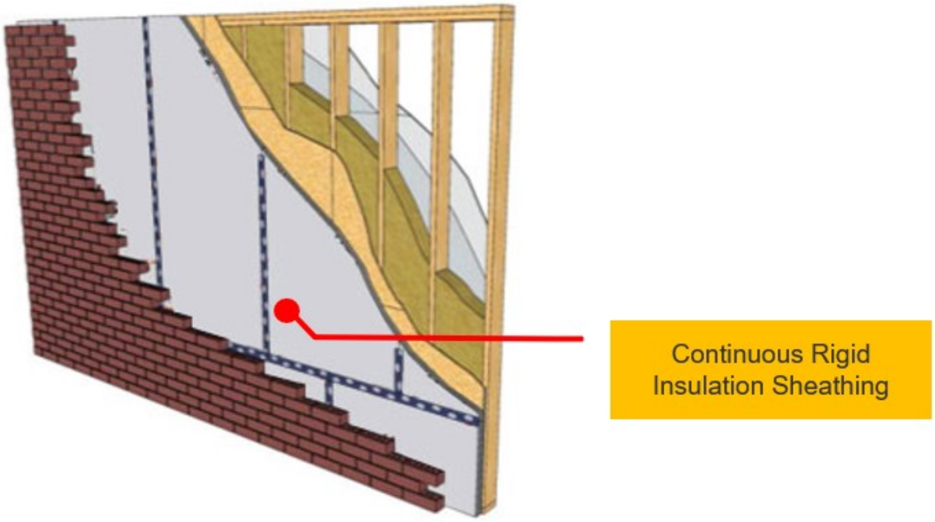


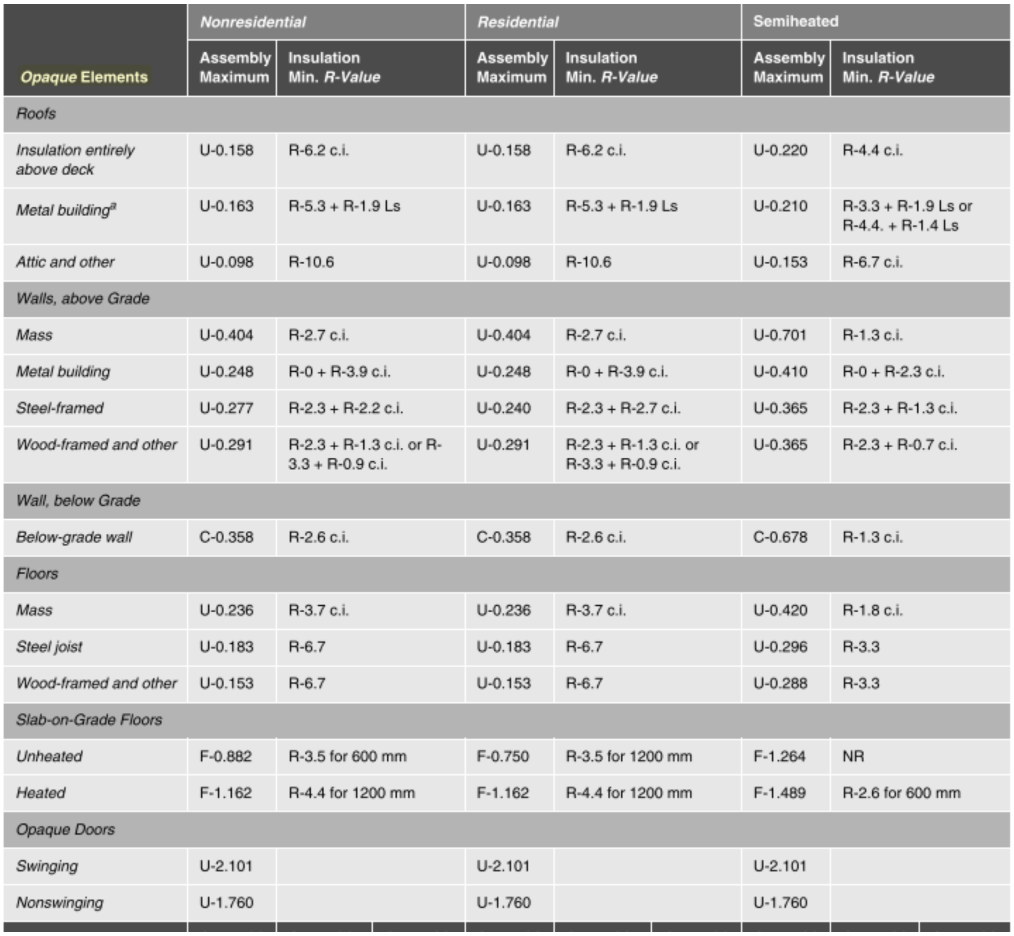
Figure 6 Wood Frame Wall Section(*Rigid Board Installation Builder's Guide*, 2018)

3.2.1.b EPD and price:

For sheathing insulation, board-type insulation is chosen as the study object. A material's price and available thicknesses can be obtained through market research. For the case of EPD, it can be downloaded from the manufacturer's website. For this study, only materials that are available on the North American market are considered as candidate materials.

3.2.1.c Location distance measure:

The distance between the manufacturer's factory and the construction site is measured in kilometres using Google Maps. If an EPD shows multiple manufacturing locations, the one closest to the intended construction site is selected. Most EPDs specify an assumed number of kilometres for the A4 impact declaration. Each EPD has a different default outbound distance for A4 calculation. Therefore, the A4 impact needs to be converted per kilometres, and then it can be multiplied by the actual distance. For example, suppose a material EPD shows A4 stage emission for Functional Unit thickness as 0.05 kg CO2 eq. and this is calculated based on 50 miles (80.4672 km). In that case, it is converted to 0.00062137 kg CO2 eq./km. This converted emission amount per kilometres then multiplied with the actual distance between the manufacturer and the intended site.



Opaque Elements	Nonresidential		Residential		Semiheated	
	Assembly Maximum	Insulation Min. *R-Value*	Assembly Maximum	Insulation Min. *R-Value*	Assembly Maximum	Insulation Min. *R-Value*
Roofs						
Insulation entirely above deck	U-0.158	R-6.2 c.i.	U-0.158	R-6.2 c.i.	U-0.220	R-4.4 c.i.
Metal building[a]	U-0.163	R-5.3 + R-1.9 Ls	U-0.163	R-5.3 + R-1.9 Ls	U-0.210	R-3.3 + R-1.9 Ls or R-4.4. + R-1.4 Ls
Attic and other	U-0.098	R-10.6	U-0.098	R-10.6	U-0.153	R-6.7 c.i.
Walls, above Grade						
Mass	U-0.404	R-2.7 c.i.	U-0.404	R-2.7 c.i.	U-0.701	R-1.3 c.i.
Metal building	U-0.248	R-0 + R-3.9 c.i.	U-0.248	R-0 + R-3.9 c.i.	U-0.410	R-0 + R-2.3 c.i.
Steel-framed	U-0.277	R-2.3 + R-2.2 c.i.	U-0.240	R-2.3 + R-2.7 c.i.	U-0.365	R-2.3 + R-1.3 c.i.
Wood-framed and other	U-0.291	R-2.3 + R-1.3 c.i. or R-3.3 + R-0.9 c.i.	U-0.291	R-2.3 + R-1.3 c.i. or R-3.3 + R-0.9 c.i.	U-0.365	R-2.3 + R-0.7 c.i.
Wall, below Grade						
Below-grade wall	C-0.358	R-2.6 c.i.	C-0.358	R-2.6 c.i.	C-0.678	R-1.3 c.i.
Floors						
Mass	U-0.236	R-3.7 c.i.	U-0.236	R-3.7 c.i.	U-0.420	R-1.8 c.i.
Steel joist	U-0.183	R-6.7	U-0.183	R-6.7	U-0.296	R-3.3
Wood-framed and other	U-0.153	R-6.7	U-0.153	R-6.7	U-0.288	R-3.3
Slab-on-Grade Floors						
Unheated	F-0.882	R-3.5 for 600 mm	F-0.750	R-3.5 for 1200 mm	F-1.264	NR
Heated	F-1.162	R-4.4 for 1200 mm	F-1.162	R-4.4 for 1200 mm	F-1.489	R-2.6 for 600 mm
Opaque Doors						
Swinging	U-2.101		U-2.101		U-2.101	
Nonswinging	U-1.760		U-1.760		U-1.760	

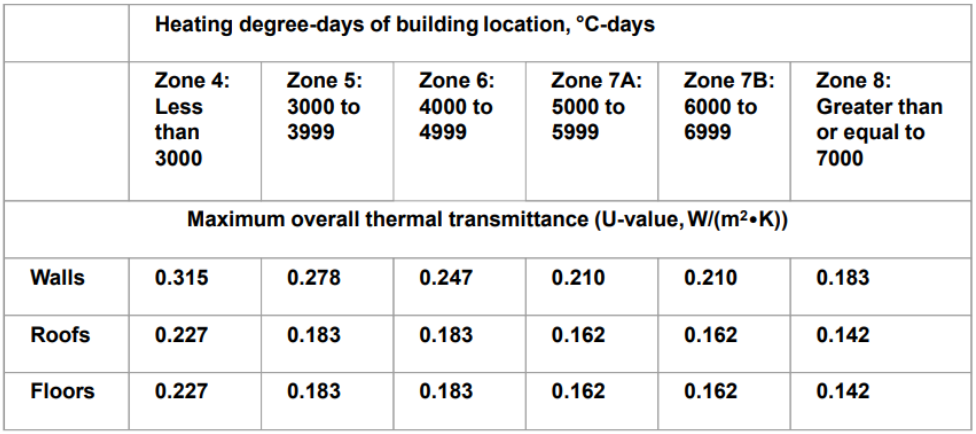
Table 4 Building Envelope Requirements for Climate Zone 7(ASHREA 90.1)

3.2.1.d Regulation:

The ASHRAE 90.1 specifies maximum U-values by building envelope parts (e.g., roof, wall, and floor) and by zones. Likewise, the Canadian National Energy Code for Buildings (NECB) specifies U-value requirement by zones and by parts.

Compared to the ASHRAE 90.1 standard, the NECB standard demands a higher insulation performance. For example, according to NECB, the maximum assembly U-value for a wall is 0.21 for any building type. However, according to ASHRAE 90.1, the maximum assembly U-value for a wall is 0.291 for a wood-framed residential building. So, more stringent NECB standard is used for this study.

The Heating degree day of study area is required in order to find out the climate zone defined by ASHRAE 90.1 and NECB. For example, the ten-year average Heating Degree Days in Edmonton is 5014.9 HDD (edmonton.weatherstats.ca, n.d.) which belongs to zone 7 according to ASHRAE 90.1. Table 4 is taken from ASHRAE 90.1, which shows the minimum R-value of insulation required for the type of building and the building parts. The Canadian National Energy Code for Buildings (NECB) divided zones in more detail. According to it, Edmonton belongs to zone 7A. The required U values for building envelopes for above-ground opaque building are shown in Table 5.



	Heating degree-days of building location, °C-days					
	Zone 4: Less than 3000	**Zone 5: 3000 to 3999**	**Zone 6: 4000 to 4999**	**Zone 7A: 5000 to 5999**	**Zone 7B: 6000 to 6999**	**Zone 8: Greater than or equal to 7000**
Maximum overall thermal transmittance (U-value, W/(m²•K))						
Walls	**0.315**	**0.278**	**0.247**	**0.210**	**0.210**	**0.183**
Roofs	**0.227**	**0.183**	**0.183**	**0.162**	**0.162**	**0.142**
Floors	**0.227**	**0.183**	**0.183**	**0.162**	**0.162**	**0.142**

Table 5 Building Envelope Requirements for Climate Zone 7A (NECB)

3.2.1.e Interest rate and inflation rate:

Interest rates and inflation rates are volatile and are determined by current market conditions. National and commercial banks provide forecasts of interest rate and inflation rate. For example, according to Statista (n.d.) the inflation rate averaged 1.6% in 2021, and is forecasted to be 1.8% in 2022, while the current (February 2021) interest rate is 0.25%.

3.2.1.f Fuel information:

To calculate the operational part of LCA and LCC, the fuel’s energy content, price, and CO2 emission rate are required. For example, $1m^3$ of natural gas produces 42.3 mol of CO2 or 1.86 kg CO2 eq., and natural gas has an average energy content of 37 MJ/m^3 (Willms, 2007).

According to the ‘Ontario Energy Board,' the price per m³ ranges from 10.35 to 13.51 dollars (Ontario Energy Board, 2020). According to the Alberta Utilities Commission, natural gas bills include energy charges, delivery charges (including fixed charge and variable charge), administration charges, municipal franchise fees, carbon levy, and GST (Goods and Services Tax).

3.3 **LCA and LCC calculation**

This section demonstrates the LCA and LCC calculation process. The life cycle system boundary for this study is from A1-4 plus B6. This includes manufacturing, transportation, and operation but excludes installation, maintenance, repair, replacement, refurbishment, and the end-of-life stage. The A5 installation stage is not included in the study due to its marginal importance and a lack of clear information on EPDs. Also, the B1 to B5 and B7 are excluded because they are unrelated to the insulation material.

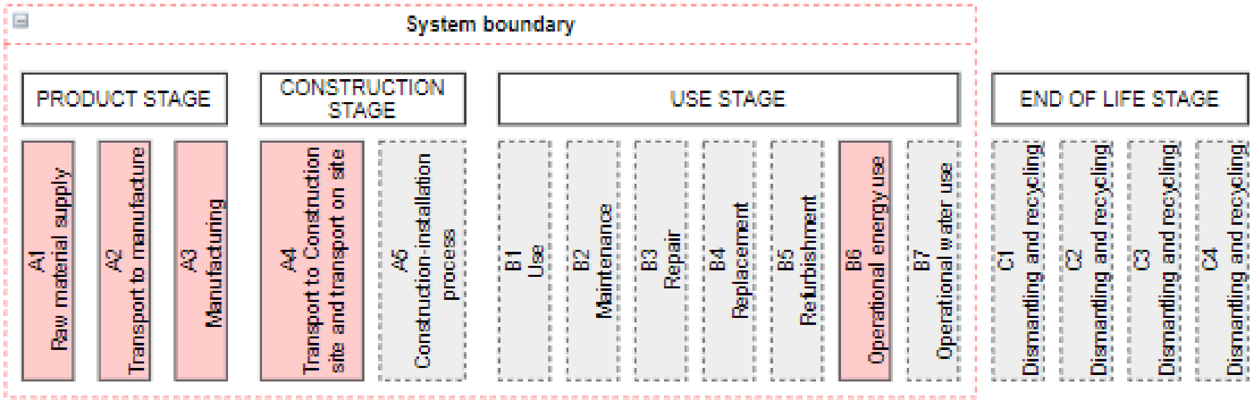


Figure 7 System Boundary of the Study

3.3.1.a Stage A1-A4 and C1 to C4 embodied energy calculation (LCA):

Based on a product EPD, the embodied CO2 from stage A1 to stage A4 can be calculated. Because the embodied CO2 data on an EPD are based on functional unit thickness per 1 m^2, alternatives' embodied CO2 can be quantified based on alternative thickness. As written on the PCR, the functional unit for thermal insulation is 1m² area with a thickness that gives an average thermal resistance of R_{SI} = 1k·m2/W (R_{IP}=5.68h·ft2 °F/Btu).

The thermal resistance may be expressed as R_{SI} = d/ λ, where d [m] is the insulation thickness and functional units may also be expressed as: $FU = R_{SI} \cdot \lambda \cdot \rho \cdot A\,[kg]$.

$R_{SI} = thermal\ resistance\ [m^2K/W]$

$\lambda = thermal\ conductivity\ [W/mK]$

$\rho = density\ of\ insulation\ product\ [kg/m^3]$

$A = Area\ [m^2]\ (here, 1\ m^2)$

To calculate the LCA for the A1 to A4 stage, convert the FU thickness of the A1 to A4 impact (kg CO2 eq.) to 1mm thickness, then multiply by the actual thickness of the subject material.

$$LCA_{a1\sim4} = \frac{(FUa_{1\sim3} + FUa_4 \times dist)x_i}{FUthk} \qquad 26)$$

x_i *: the thickness of material i*

$dist$*: distance between manufacturing factory to site location*

$FUa_{1\sim3}$*: Stage A1 to A3 GHG for Functional Unit*

FUa_4 *: Stage A4 GHG for Functional Unit*

$FUthk$ *: Functional Unit Thickness*

3.3.1.b Stage B6 operation energy calculation (LCA):

The EPD does not have information on the impact of B6 stage emissions because they are operational emission. The fuel consumption can be calculated utilizing equation 22 (see section 2.4.3) by calculating the heat loss per unit area (m^2).

$$LCA_{b6} = \frac{86400 \cdot HDD}{(R_{T.W.} + \frac{x_i}{k}) \cdot \eta \cdot \mathrm{H_u}} \times emis \times N \qquad 27)$$

$emis$: emission rate of the fuel

N: specified time term (years)

Combining equation 26 and 27, the following equation 28 represents LCA for this study.

$$LCA(x) = LCA_{a1\sim4} + LCA_{b6} \qquad 28)$$

3.3.1.c LCC :

For the B6 stage, the present value of future energy costs over the life cycle year is calculated utilizing equation 22 (see section 2.4.3) and the initial material price is added for a complete LCC calculation.

$$LCC = \frac{86400 \cdot HDD}{(R_{T.W.} + \frac{x_i}{k}) \cdot \eta \cdot \mathrm{H_u}} \times Fuel_{price} \times PWF + Initial_{price}(x_i) \qquad 29)$$

$Fuel_{price}$: price of the fuel

$Initial_{price}(x_i)$ = material price for the thickness of material i

3.4 **Pareto Search**

Many multi-objective optimization algorithms, including the Multi-Objectives Genetic Algorithm (MOGA) and the Non-Dominated Sorting Genetic Algorithm (NSGA-II), mutate alternatives to find the best ones. However, the alternatives for this framework must only be from the market's commercially available options. Because the available thickness of the specific product on the market is limited, double layering of a same material is considered to generate more alternatives. If there are n available thicknesses on the market, the number of options becomes 2n+n(n-1)/2. For example, if a material has 3 thickness options (0.75-

inch, 1 inch, 1.5 inch options), the available options, including double layer combinations, are 0.75 inch, 1 inch, 1.5 inch options, 1.5 (0.75+0.75) inch, 1.75 (1+0.75) inch, 2.25 (0.75+1.5) inch, 2 (1+1) inch, 2.5 (1+1.5) inch, and 3 (1.5+1.5) inch.

Following the creation of all alternatives, the multi-objective optimization process determines the non-dominance pareto optimum sets for each material's minimum LCA and minimum LCC. The optimization is expressed in equation 30, utilizing equations 28 (see section 3.3.1.b) and 29 (see section 3.3.1.c).

$$\min F(X) = \left(f1(x), f2(x)\right)$$

$$f1 = LCA(\mathrm{x}) \qquad 30)$$

$$f2 = LCC(\mathrm{x})$$

The following is the pareto search rule:

Given two points x, y in Ω, we say that $x \prec y$ (x dominates y) when $F(x) \prec_{\mathrm{F}} F(y)$ 31)

In this case, the upper limit constraint is the ‘up to double layers’, as implied already in the alternatives generation. Equation 32 expresses the lower limit constraint, while the maximum U value is specified in design requirements such as the local code or ASHRAE 90.1.

$$k\ (1/maxU - \alpha) - x \leq 0 \qquad 32)$$

maxU : the maximum U-value specified in the local code or ASHRAE 90.1

Where the base wall’s R-value is α, k is the insulation material’s thermal conductivity coefficient and the thickness x of the insulation material satisfies the maximum U-value.

3.5 **Novel fuzzy modified TOPSIS**

After identifying the pareto optimal material and thickness, the solutions are ranked using the TOPSIS method, with the expert's linguistic preference. As mentioned in the literature review, the ‘Novel fuzzy modified TOPSIS' method manages mixed criteria (objective and subjective), so it is used in this study.

First, TOPSIS criteria are selected. As shown in Table 1 (see section 2.4.3) from the literature review, criteria are generally classified into three types: cost-related, sustainability-related, and technical performance-related. Because cost and GHG are already considered in LCC and LCA, the focus of the linguistic evaluation criteria is on the material's technical performance. In addition, other than the LCC, the ‘initial cost’ is included as a cost criterion because the immediate cost is one of the primary concerns among decision makers. Hence, the objective criteria are the LCC, initial cost, and LCA.

For the subjective criteria, based on a literature review and expert opinion, technical performances such as work duration, material durability of construction, water vapour diffusion, fire resistance, and loss factor are chosen. In total, seven criteria are selected for the study, as shown in Table 6.

Because the study's alternatives are a pair of materials and a thickness combination. There are more alternatives than the number of materials. With the linguistic variables, each material can be rated but not the thicknesses. Therefore, regardless of thickness, the same type of material receives the same rating for subjective criteria. In contrast, all alternatives (material and thickness combination) should have different ratings for the objective criteria (LCA, LCC, and initial cost).

Subjective criteria	Objective criteria
Work duration (C1)	LCC (C5)
Material durability (C2)	Initial Cost (C6)
Water vapor diffusion (C3)	LCA (C7)
Fire resistance (C4)	

Table 6 Subjective and Objective Criteria

3.5.1.a Criteria weighting:

The fuzzy linguistic variables listed below are used to weight the criteria and rate the alternatives. Fuzzy number membership for weighting and rating is shown in Table 7 and Table 8, respectively.

Meaning of Linguistic Scale	Numerical Scale
Very low (VL)	(0, 0, 0.1)
Low (L)	(0, 0.1, 0.3)
Medium low (ML)	(0.1, 0.3, 0.5)
Medium (M)	(0.3, 0.5, 0.7)
Medium high (MH)	(0.5, 0.7, 0.9)
High (H)	(0.7, 0.9, 1.0)
Very high (VH)	(0.9, 1.0, 1.0)

Table 7 Linguistic Variables for the Importance of Each Criterion

Meaning of Linguistic Scale	Numerical Scale
Very poor (VP)	(0, 0, 1)
Poor (P)	(0, 1, 3)
Medium poor (MP)	(1, 3, 5)
Fair (F)	(3, 5, 7)
Medium good (MG)	(5, 7, 9)
Good (G)	(7, 9, 10)
Very good (VG)	(9, 10, 10)

Table 8 Linguistic Variables for the Alternative Performance

The ‘TOPSIS' process begins after decision-makers have completed the qualitative evaluation for criteria weighting and alternative ratings. Vahdani et al. (2011)’s novel fuzzy modified TOPSIS method is used for ranking because the criteria are mixed with subjective and objective criteria. First, the objective criteria’s crisp value is converted to a triangular fuzzy set in the form of $\tilde{A} \otimes r$ by multiplying normalized linguistically valued weight. Next, the fuzzy TOPSIS process begins, as described in 2.3.2 TOPSIS and Fuzzy TOPSIS. For defuzzification, the second weighted method is used, and Euclidean distance is used to compute the distance between the ideal point and each alternative.

Chapter 4 **Application of the method to an example case**

This section provides an example of how to choose a sustainable insulation material using the proposed method.

4.1 **Data input**

4.1.1.a Base wall R-value calculation:

The study assumes that the base wall is a wooden framed wall with 2”×6” stud framing at 400mm on the center and cavities filled with fiberglass batt insulation for the wall cavities. The stud wall is composed of an inner layer drywall, wood frame, insulation layer, and external finish layer. Studs are typically spaced at 16” (traditional framing) or 24” (advanced framing) according to ASHRAE 90.14-2019. If it is spaced at 16”, the wood frame takes about 25% of the wall, while the cavity takes 75% of the total area. So, the calculation of the 2×6 wood stud wall’s R-value is as shown in Table 9, which is expressed in equation 33 below. The R-values for each layer are obtained from ColoradoENERGY.Org. In this case, the calculated base wall R-value is 3.11 km2/W. (*ColoradoENERGY.Org - R-Value Table*, n.d.)

Component	R-Value Studs	R-Value Cavity	Assembly R-Value
Wall - Outside Air Film (winter)	0.03 K m²/W	0.03 K m²/W	
Siding - Wood Bevel	0.14 K m²/W	0.14 K m²/W	
Plywood Sheathing - 1/2”	0.11 K m²/W	0.11 K m²/W	
6” Fiberglass Batt		3.87 K m²/W	
6” Stud	1.19 K m²/W		
1/2” Drywall	0.08 K m²/W	0.08 K m²/W	
Inside Air Film	0.12 K m²/W	0.12 K m²/W	
Percent for 16" o.c. + Additional studs	25%	75%	
Total Wall Component R-Values	**1.67 K m²/W**	**4.35 K m²/W**	
Wall Component U-Values	**0.60 W/m²K**	**0.23 W/m²K**	
Total Wall Assembly R-Value			**3.11 K m²/W**

Table 9 R-Value of Sample Base Wall

$$\boldsymbol{Formula}: \boldsymbol{Assembly\ R\ value} = \mathbf{1} / (\boldsymbol{Assembly\ U - value}) = \mathbf{1} / (\boldsymbol{U - studs\ x}\,\% + \boldsymbol{U - cavity\ x}\,\%) \qquad 33)$$

4.1.1.b EPD and price:

Rigid insulation materials such as XPS, fibreglass, polyiso, and EPS board are chosen as study alternatives. These materials are from specific brands and have commercial names or numbers. In this study, they are referred as material A, B, C, and D. Each material comes in a range of thicknesses. Some have a wider range of thicknesses than others. Table 10 shows the researched material thickness options along with their prices.

Material Type	Board Size	Thickness (Inch)	Thickness (mm)	Price per Board (CAD)	Price per m2 (CAD)
A: XPS	4 ‘·8’ (2.97 m²)	0.75	19.05	14.85	5.00
		1	25.4	17.79	5.98
		1.5	38.1	27.58	9.28
		2	50.8	53.57	18.02
		2.5	63.5	40.75	13.71
		3	76.2	43.41	14.60
		4	101.6	80.63	27.12
B: Earthwool	2'·4' (0.74 m²)	1	25.4	23.33	31.39
		2	50.8	44.83	60.32
C: Polyiso	4'·8' (2.97 m²)	1	25.4	22.86	7.69
		1.55	39.37	44.49	14.97
		2	50.8	48.94	16.46
D: EPS	4'·8' (2.97 m²)	0.75	19.05	19.99	6.72
		1	25.4	24.99	8.41
		1.5	38.1	35.99	12.11
		2	50.8	43.99	14.80

Table 10 Candidate Material and Thickness Availability

4.1.1.c Location distance measure:

Table 11 summarizes the EPDs information for materials A-XPS, B-Earthwool, C-Polyiso, and D-EPS. The intended site is assumed to be in Edmonton in this study, and the distance from each manufacturer to the intended site is indicated in the third column. Figure 8 depicts the location of the manufacturing site. Product D-EPS has the shortest distance since the factory is in Edmonton. Overall, product A-XPS has a much higher embodied GHG (kg CO2 eq.).

MR	**FU Thickness (m)**	**Conductivity (W/mK)**	**Distance (Km)**	**A1 ~ A3 stage (kgCO2eq./m^2)**	**A4 stage (kgCO2eq./m^2/km)**	**C2&C4 stage (kgCO2eq./m^2)**
A	0.031	0.031	1,525.000	25.360	0.0009420	6.05
B	0.032	0.032	2,984.000	9.110	0.0007890	0.2528
C	0.022	0.022	2,041.250	4.100	0.0001970	0.09899
D	0.040	0.0401	6.600	2.627	0.0002693	0.038

Table 11 Summarized EPD of Candidate Material (1 m$_2$, R$_{SI}$=1)

Figure 8 Manufacturer Location

4.1.1.d Regulation:

The average Heating Degree Days in Edmonton over the last ten years, as explained in the literature review, is 5014.9 HDD (edmonton.weatherstats.ca, n.d.). According to the NECB standard, the maximum assembly U-value for an external wall is 0.21.

4.1.1.e Interest rate and inflation rate:

This study assumes an inflation rate of 1.8% (Canada - Inflation Rate 1986-2026 | Statista, n.d.) and an interest rate of 0.25% (Statista, n.d.).

4.1.1.f Fuel information:

Natural gas is the most commonly used energy source for heating in Canada. Therefore, this study assumes heating fuel is natural gas, which emits 1.86 kg CO2 equivalent per m^3. Natural gas has an average energy content of 37 MJ/m^3. Natural gas costs 13.4224 cents per m^3 according to Ontario Energy Board (July,2021).

4.2 **LCA and LCC calculation**

4.2.1.a Create options:

As explained in the methodology, to provide more alternatives for this study, the available thickness options are expanded by doubling layers. So, if there are 'n' thicknesses available on the market, then '2n+n(n-1)/2' thickness alternatives are generated for each material. As a result, for material A-XPS, B-Earthwool, C-Polyiso, and D-EPS, a total of 63 alternatives are created.

$$\text{A} : 2 \times 7 + 7(7-1)/2 = 35 \text{ EA}$$

$$\text{B} : 2 \times 2 + 2(2-1)/2 = 5 \text{ EA}$$

34)

$$\text{C} : 2 \times 3 + 3(3-1)/2 = 9 \text{ EA}$$

$$\text{D} : 2 \times 4 + 4(4-1)/2 = 14 \text{ EA}$$

4.2.1.b LCA calculation:

To demonstrate the LCA calculation technique, material A-XPS with a thickness of 82.55mm (19.05mm and 63.55mm.) is calculated as an example.

A1~4 stage: Calculate proportionate GHG based on thickness of 0.08255m against functional unit thickness by referring Table 11 (see section 4.1.1.c), then use equation 26 (see section 3.1.1.a). As a result, the GHG for A1~4 stage is 71.36 kg CO2 eq.

$$LCA_{a1\sim4} = \frac{(25.36 + 0.0009420 \times 1525) \times 0.08255}{0.031} = 71.36 \qquad 35)$$

B6 *stage*: The R-value is as follows for material A-XPS with a thickness of 0.08255m.

$$R_{izo} = \frac{x}{k} = \frac{0.08255}{0.031} = 2.6629 \qquad 36)$$

Edmonton heating degree day is 5014.9 and assume the heating system efficiency is 80%, then plug these numbers into equation 27 (see section 3.1.1.b)

$$\mathrm{m_{fA}} = \frac{86400 \cdot 5014.9}{(3.11+2.66) \cdot 0.8 \cdot 37000000} = 2.54\mathrm{m}^3/\mathrm{m}^2\text{-year} \qquad 37)$$

Where the average energy content of natural gas is 37 MJ/m3,

$$2.54\mathrm{m}^3/\mathrm{m}^2\text{-year} * 1.86\mathrm{kg}/\mathrm{m}^3 = 4.72\ \mathrm{kg}/\mathrm{m}^2\text{-year} \qquad 38)$$

Where the emission rate of natural gas is $1.86\mathrm{kg}/\mathrm{m}^3$,

Totalling up the environmental impact of stages A1~4 and B6:

71.63 kg CO2 eq.+4.72 kg CO2 eq.*70years

=401.5 kg CO2 eq. 39)

4.2.1.c LCC calculation:

The following is a continuation of the demonstration of material A-XPS's thickness 82.55mm for the LCC calculation. The first step is to calculate the actual interest rate (r) using equation 25 (see section 2.4.4).

$$if\ 1.8\% > 0.25\%\ then\ r\ =\ (1.8\% - 0.25\%)/(1 + 0.25\%) = 1.55\% \qquad 40)$$

Then, employing equation 24 (see section 2.4.4) calculate the Present Worth Factor(PWF) for 70 years using the actual interest rate (r).

$$PWF\ = \frac{(1 + 0.0155)^{70} - 1}{0.0155 \cdot (1 + 0.0155)^{70}} = 42.58 \qquad 41)$$

The annual energy cost $C_{A.H}$ (\$/m2-year) per unit area m2 is calculated using equation 22 (see section 2.4.3).

$$C_{A.H} = \frac{86400 \cdot 5014.9 * 0.13}{(3.11+2.66) \cdot 0.8 \cdot 37000000} = 0.33(\$/\mathrm{m}^2\text{-year}) \qquad 42)$$

If the price of the fuel is 0.13 \$/m^3,

The initial cost of 19.05mm and 63.5mm insulation of material A-XPS is 18.71 \$/m^2 (5\$/m^2+13.71\$/m^2). The outbound transportation cost is deemed included in the initial cost.

With the above annual energy cost, present worth factor, and initial cost, LCC can be calculated by using equation 23 (see section 2.4.4).

$$C_T = 0.33 \cdot 42.58 + 18.71 = 32.746\ (\$/\mathrm{m}^2) \qquad 43)$$

See the Appendix A for calculated LCC and LCA

4.3 **Pareto Search**

4.3.1.a Calculation of minimum thickness:

Because all materials have varying conductivities, different minimum thicknesses are required for each material to satisfy the design code.

Taking material A-XPS as an example, when the base wall's effective R-value is 3.11 K m²/W, the thickness x that satisfies the maximum 0.21 U value is caculated as in equation 44. Since we know the conductivity of the material from the Table 11 (see section 4.1.1.c) and the maximum allowed U value, the equation 44 is established by utilizing equation 16 and 17 (see section 2.4.3). The minimum thickness of all materials A-XPS, B-Earthwool, C-Polyiso, and D-EPS for regulation is calculated in Table 12.

$$\frac{1}{3.11+\frac{x}{0.031}} \leq 0.21 \qquad 44)$$

$$x = 0.051$$

MR	FU Thickness (m)	Minimum thickness required (m)
A	0.031	0.051
B	0.032	0.053
C	0.022	0.036
D	0.040	0.066

Table 12 Minimum Thickness for Materials

From all generated options, after removing unqualified options, the number of material A-XPS options drops to 28 from 35. Material B-Earthwool options are down to 2 from 5. Material C-Polyiso options diminish from 9 to 8. Finally, material D-EPS options fall from 14 to 5 as shown in Appendix A.

4.3.1.b Pareto search

The number of alternatives is further reduced while implementing the pareto search rule for non-dominated minimum LCC and LCA. The non-dominance pareto sets are shown in Table 13.

material A-XPS pareto sets (2 ,7, 12 ,17 ,23, 24,25,26)

Option 2 dominates option 1,4,5,9,10,14

Option 7 dominates option 5,6,10,11,14,15

Option 12 dominates option 3,5,10,15,16,19

Option 17 dominates option 3,5,10,15,19,20

Option 23 dominates option 3,8, 13, 15, 19, 20, 21

Option 24 dominates option 3,8, 13, 15, 18,19,20,21,28

Option 25 dominates option 22,27,28

Option 26 dominates option 3,8, 13, 15, 18,19,20,21,22,28

Material B-Earthwool pareto sets (1,2)

The two options are non dominance to each other

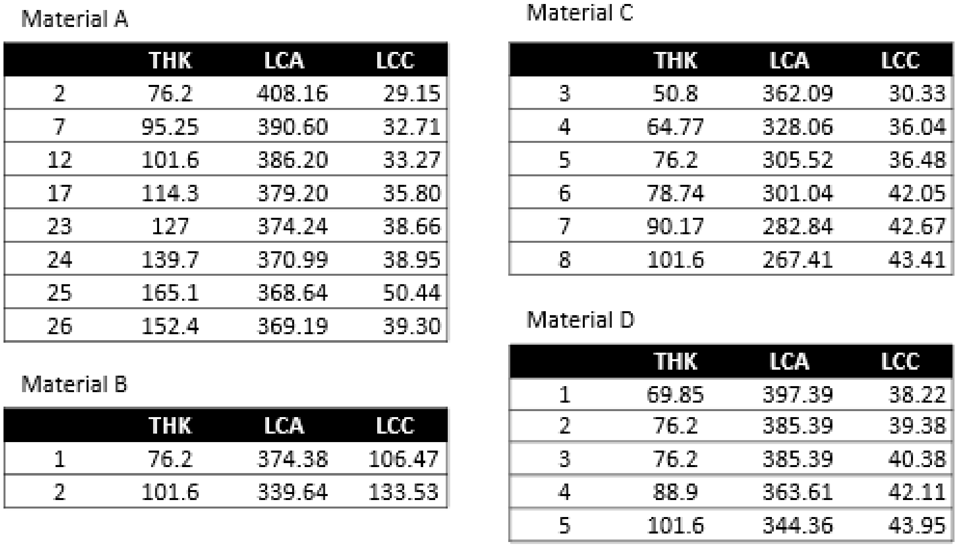
Material C-Polyiso pareto sets (3,4,5,6,7 and 8)

Option 3 dominates option 1,2

option 4,5,6,7, and 8 non dominance to each other

Material D-EPS (1,2 ,4, and 5)

non dominance to each other



Material A

	THK	LCA	LCC
2	76.2	408.16	29.15
7	95.25	390.60	32.71
12	101.6	386.20	33.27
17	114.3	379.20	35.80
23	127	374.24	38.66
24	139.7	370.99	38.95
25	165.1	368.64	50.44
26	152.4	369.19	39.30

Material B

	THK	LCA	LCC
1	76.2	374.38	106.47
2	101.6	339.64	133.53

Material C

	THK	LCA	LCC
3	50.8	362.09	30.33
4	64.77	328.06	36.04
5	76.2	305.52	36.48
6	78.74	301.04	42.05
7	90.17	282.84	42.67
8	101.6	267.41	43.41

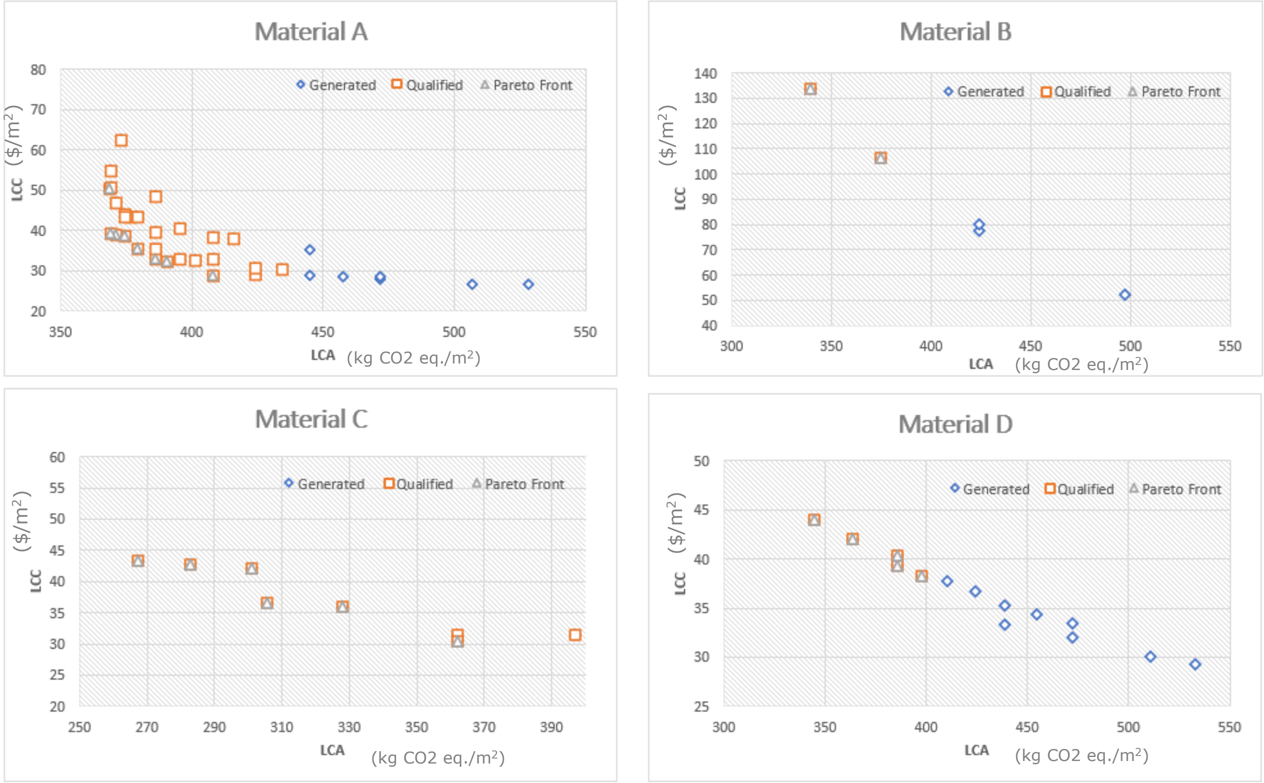
Material D

	THK	LCA	LCC
1	69.85	397.39	38.22
2	76.2	385.39	39.38
3	76.2	385.39	40.38
4	88.9	363.61	42.11
5	101.6	344.36	43.95

Table 13 Pareto Set of Materials

When searching for a pareto set in Excel, each alternative's LCA and LCC values are manually compared using the IF function. Whereas for python programming, the python paretoset 1.2.0 source code is used. The results were the same from Excel and software since the inputs were the same.

Figure 9 depicts the LCA and LCC of each material's various thickness alternatives in a scatter plot; all points represent all generated alternatives, while the blue points represent the ones that disqualified for the u-value requirements. The pareto front sets are grey points, and these are ready for TOPSIS. It is noticeable that the scatter plots are generally in a convex shape for the pareto front.


Material A

Material B

Material C

Material D

Figure 9 Pareto Set Scatter Plot

4.4 **TOPSIS**

Table 14 and Table 15 show the linguistic weighting for six criteria and the criteria rating for alternatives that are given by three decision-makers (DM1, DM2 and DM3).

	Criteria	**DM1**	**DM2**	**DM3**
C1	Work Duration	High	Very High	Very High
C2	Durability	Very High	High	Very High
C3	Water vapor diffusion	Medium	Low	Medium
C4	Fire Resistance	Medium	Medium	Medium
C5	LCC	Very High	Very High	High
C6	Initial Cost	Medium	Low	Medium
C7	LCA	High	Very High	Very High

Table 14 Linguistic Criteria Weighting

According to the “step 3” in 2.3.2.a, the cost criteria are 'LCC,' 'Initial Cost,' and 'LCA,' because smaller is better for these criteria, and the benefit criteria are 'work duration’, ‘material durability’, 'water vapour diffusion’, and 'fire resistance’ because they are evaluated linguistically. In linguistic scale the higher number scale is given for the better linguistic evaluation.

	C1				**C2**				**C3**				**C4**				**C5**			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
DM 1	F	F	G	G	G	VG	G	VG	F	G	G	G	F	F	G	G	F	F	G	G
DM 2	F	G	F	F	P	G	F	G	F	F	G	G	F	G	F	F	F	G	F	F
DM 3	G	F	VG	VG	G	VG	VG	VG	G	VG	G	G	G	F	VG	VG	G	F	VG	VG
VP - Very poor, P - Poor, MP - Medium poor, F - Fair, MG - Medium good, G - Good, VG - Very good																				

Table 15 Linguistic Criteria Ratings

After inputting the expert’s linguistic preference, the three decision makers’ linguistic variables are converted to numerical scales and those numerical scales are aggregated to one set of fuzzy numbers. Then for alternative ratings, subjective criteria ratings are normalized with equation 9 (see section 2.3.3.a) and objective criteria ratings normalized with equation 1 (see section 2.3.2.a). Next, the aggregated weightings above are applied to the normalized criteria ratings. Then the weighted normalized decision matrix is defuzzified. From the defuzzified decision, the matrix finds positive and negative ideal solutions to calculate each option’s distance. Finally, based on the Euclidean distance, ranks are determined. The Novel fuzzy modified TOPSIS is demonstrated in excel and can be found in Appendix A.

The following is the final ranking from the above procedure. The C 101.6mm (90.17mm + 78.74mm) are the best alternatives among the pareto set.

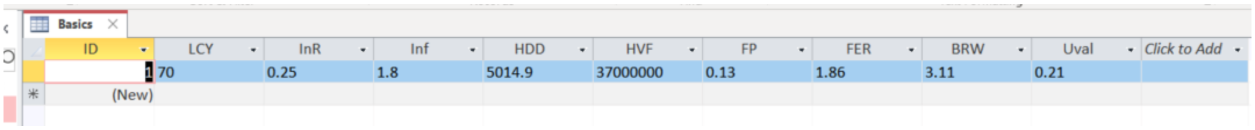
Alternatives		**D*+D-**	**Ci**	**Final Ranking**
A	**76.2**	178.47	0.72	14
A	**95.25**	175.24	0.78	11
A	**101.6**	174.61	0.79	10
A	**114.3**	173.23	0.81	9
A	**127**	172.11	0.83	7

A	**139.7**	171.79	0.84	6
A	**165.1**	170.29	0.83	8
A	**152.4**	171.58	0.84	5
B	**76.2**	189.76	0.17	21
B	**101.6**	176.54	0.23	20
C	**50.8**	181.29	0.65	16
C	**64.77**	173.81	0.77	12
C	**76.2**	171.40	0.86	4
C	**78.74**	170.36	0.87	3
C	**90.17**	170.94	0.92	2
C	**101.6**	175.83	0.94	1
D	**69.85**	181.39	0.58	19
D	**76.2**	178.37	0.62	17

Table 16 Final Score and Ranking

4.5 **Python code writing and interface**

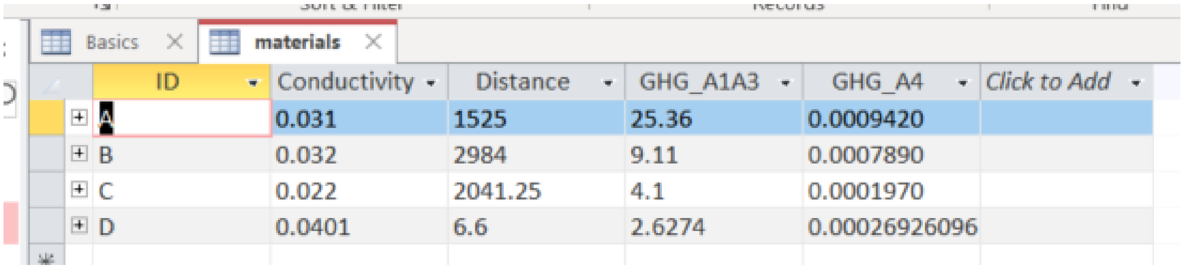
The required user data input is pre-recorded in MS Access for easier user operation. In MS Access, three categories of data are stored. The first category is the project data, which includes life cycle years, interest rate, inflation rate, regional average heating degree day, fuel price, base wall R-value and the maximum U value.



ID	LCY	InR	Inf	HDD	HVF	FP	FER	BRW	Uval
1	70	0.25	1.8	5014.9	37000000	0.13	1.86	3.11	0.21
(New)									

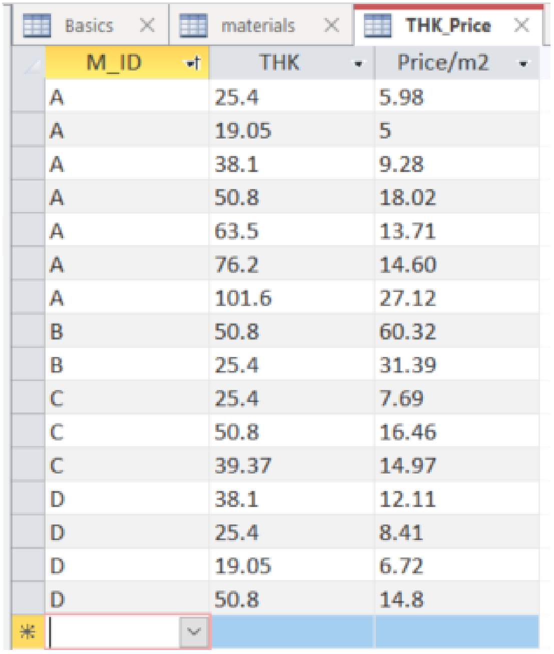
Figure 10 Project Base Information

The second category of data is the material EPD information including conductivity, GHG impact category for A1 to A4 stage and distance from the manufacturing factory to the intended site.



ID	Conductivity	Distance	GHG_A1A3	GHG_A4
A	0.031	1525	25.36	0.0009420
B	0.032	2984	9.11	0.0007890
C	0.022	2041.25	4.1	0.0001970
D	0.0401	6.6	2.6274	0.00026926096

Figure 11 Material EPD Information

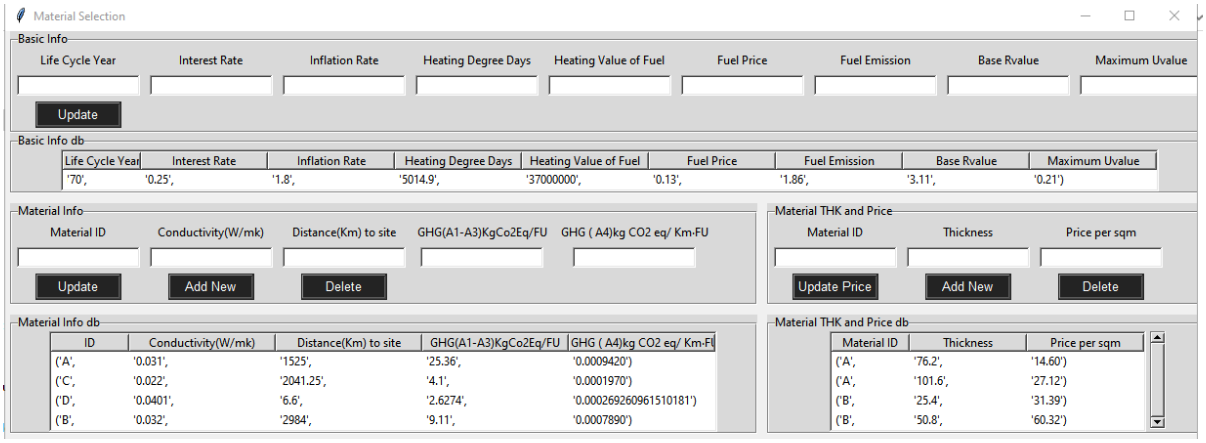


M_ID	THK	Price/m2
A	25.4	5.98
A	19.05	5
A	38.1	9.28
A	50.8	18.02
A	63.5	13.71
A	76.2	14.60
A	101.6	27.12
B	50.8	60.32
B	25.4	31.39
C	25.4	7.69
C	50.8	16.46
C	39.37	14.97
D	38.1	12.11
D	25.4	8.41
D	19.05	6.72
D	50.8	14.8

Figure 12 Available Material Thickness Information

Lastly, the third category of data stored is material commercial information, including thickness and the price of each option. These saved data are fed into the python programme via the pyodbc 4.0.31 (*Pyodbc,* n.d.) source code.

The python interface is designed with Tkinter package from the python library (*Tkinter*, n.d.). The interface includes three main compartments for pre-stored data, namely 'Basic Info,' 'Material Info,' and 'Material THK and Price.' With this user interface pre-recorded data can be modified by double-clicking the data and update button.



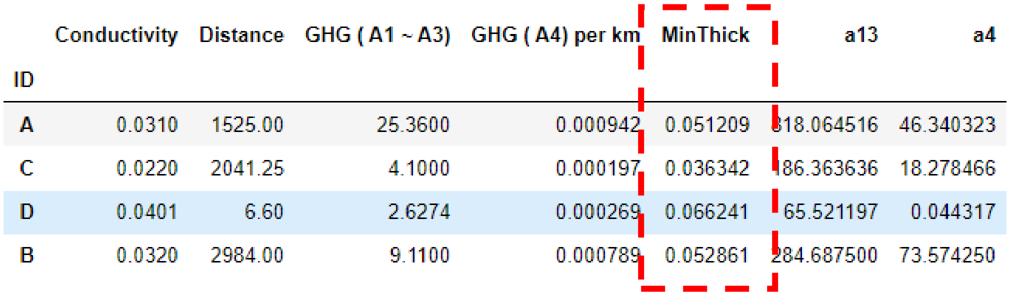
Life Cycle Year	Interest Rate	Inflation Rate	Heating Degree Days	Heating Value of Fuel	Fuel Price	Fuel Emission	Base Rvalue	Maximum Uvalue
'70',	'0.25',	'1.8',	'5014.9',	'37000000',	'0.13',	'1.86',	'3.11',	'0.21')

ID	Conductivity(W/mk)	Distance(Km) to site	GHG(A1-A3)KgCo2Eq/FU	GHG (A4)kg CO2 eq/ Km·FU
('A',	'0.031',	'1525',	'25.36',	'0.0009420')
('C',	'0.022',	'2041.25',	'4.1',	'0.0001970')
('D',	'0.0401',	'6.6',	'2.6274',	'0.000269260961510181')
('B',	'0.032',	'2984',	'9.11',	'0.0007890')

Material ID	Thickness	Price per sqm
('A',	'76.2',	'14.60')
('A',	'101.6',	'27.12')
('B',	'25.4',	'31.39')
('B',	'50.8',	'60.32')

Figure 13 Material Input Interface

The decision support system process begins by generating options for up to two layers, after which it calculates the LCA and LCC and discards the under-qualified thicknesses. Throughout the program, most of the calculations are done with the panda dataframe.



ID	Conductivity	Distance	GHG (A1 ~ A3)	GHG (A4) per km	MinThick	a13	a4
A	0.0310	1525.00	25.3600	0.000942	0.051209	318.064516	46.340323
C	0.0220	2041.25	4.1000	0.000197	0.036342	186.363636	18.278466
D	0.0401	6.60	2.6274	0.000269	0.066241	65.521197	0.044317
B	0.0320	2984.00	9.1100	0.000789	0.052861	284.687500	73.574250

Table 17 Calculated Minimum Thickness

It then performs the pareto search, only leaving the pareto front of each material for the non dominated minimum LCC and LCA. As shown in Table 18, items are reduced in steps through this process. To determine each material's pareto set, the paretoset 1.2.0 source code (Odland, 2021) is used in python programming.

	Number of generated	Number of Qualified	Number of Pareto set
Material A-XPS	35	28	8
Material B-Earthwool	5	2	2
Material C-Polyiso	9	8	6
Material D-EPS	14	5	5

Table 18 Number of Alternatives in Each Step

Figure 14 depicts the TOPSIS input interface. This program is designed to take three decision-makers' opinions. After selecting linguistic variables from the dropdown box, each opinion of a decision-maker is recorded by clicking the DC1, DC2, DC3 buttons. No linguistic variables are needed for alternative ranking for the objective criteria, but it is required for criteria weighting. The enlarged view of the input interface is attached in Appendix B. When linguistic preferences are entered in the program's backend, they are saved as a CSV file and brought back for TOPSIS calculation. There are few open code sources for TOPSIS available on the internet. However, those are ineligible for the novel modified TOPSIS steps. The open source TOPSIS codes are designed for crisp values input rather than fuzzy or mixed values. Hence, the code for TOPSIS is manually scripted using the pandas dataframe.

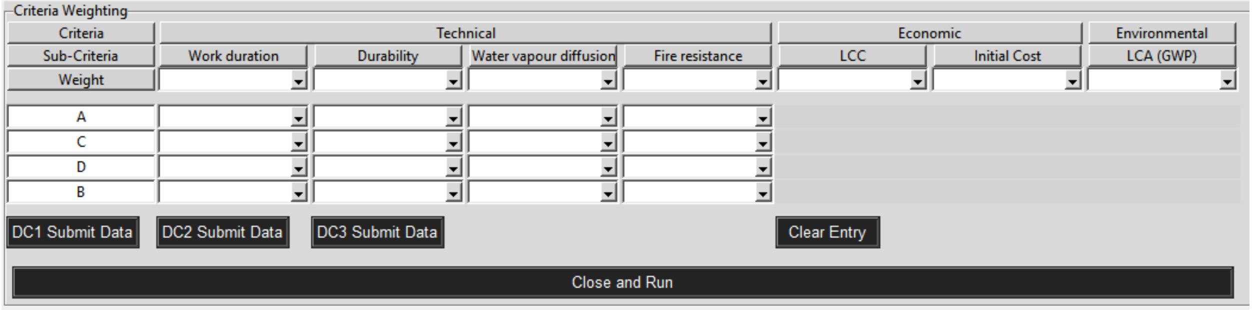


Figure 14 TOPSIS Linguistic Value Input Interface

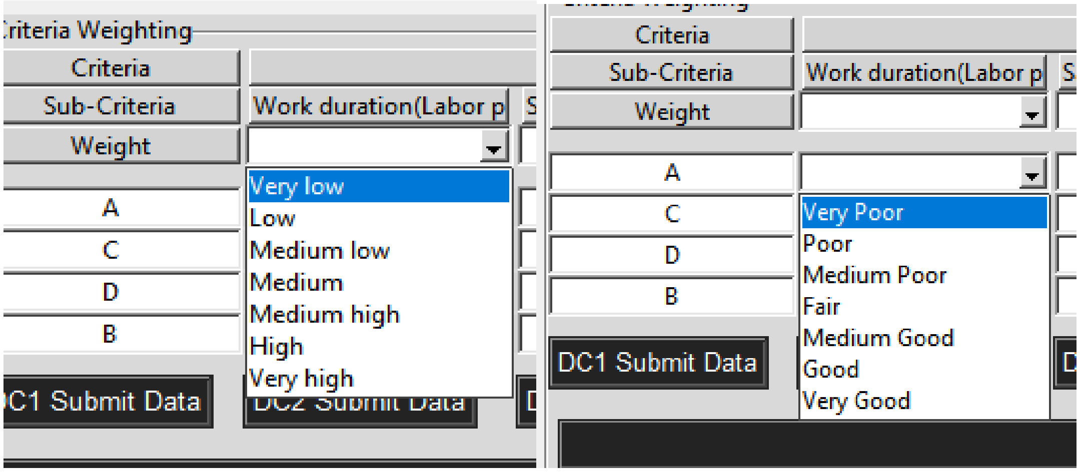
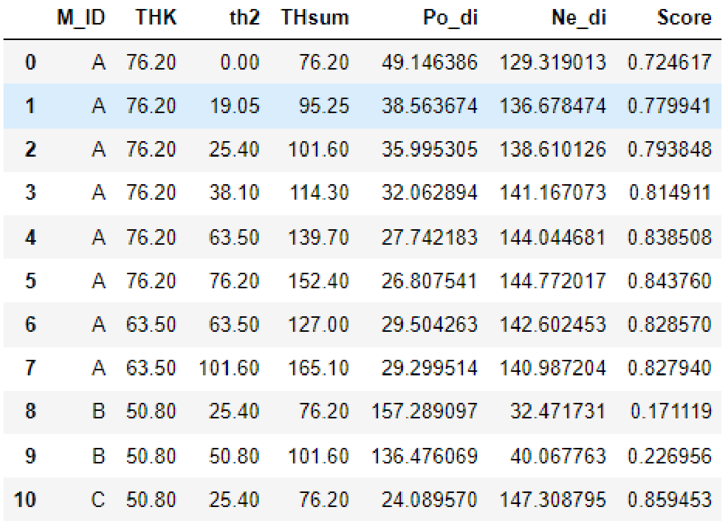


Figure 15 Linguistic Value Input Dropdown Box

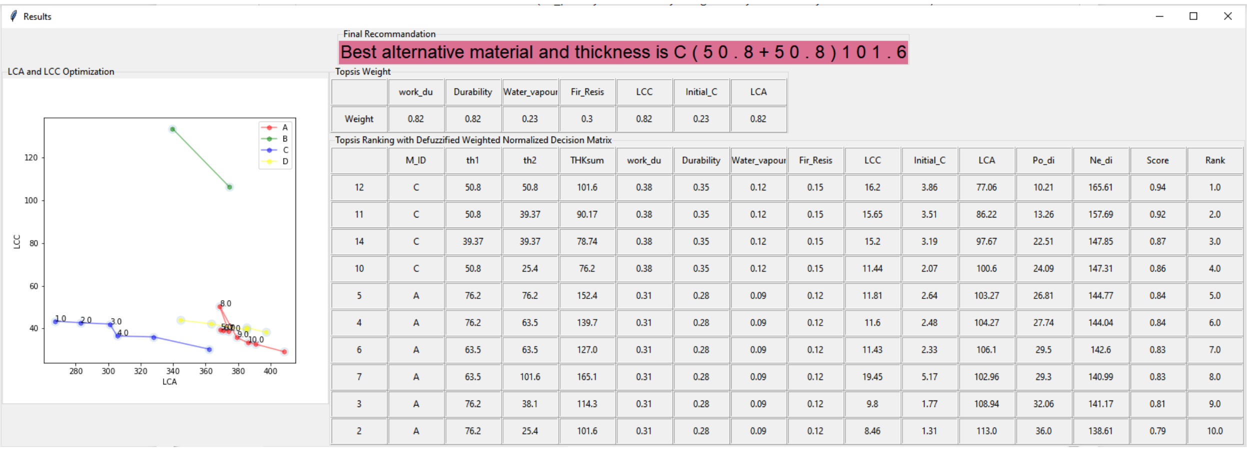
Once all three decision-makers input their linguistic value, it follows the Novel fuzzy modified TOPSIS method. The 'benefit criteria' and 'cost criteria' are indicated in the programme as 1 and -1. The ‘positive ideal solutions’, ‘negative ideal solutions’, and the ‘score’ are calculated as shown in Table 19 Alternatives' Top 10 Rank. Because all user inputs are the same, the result is the same as in the preceding Excel demonstration.



	M_ID	THK	th2	THsum	Po_di	Ne_di	Score
0	A	76.20	0.00	76.20	49.146386	129.319013	0.724617
1	A	76.20	19.05	95.25	38.563674	136.678474	0.779941
2	A	76.20	25.40	101.60	35.995305	138.610126	0.793848
3	A	76.20	38.10	114.30	32.062894	141.167073	0.814911
4	A	76.20	63.50	139.70	27.742183	144.044681	0.838508
5	A	76.20	76.20	152.40	26.807541	144.772017	0.843760
6	A	63.50	63.50	127.00	29.504263	142.602453	0.828570
7	A	63.50	101.60	165.10	29.299514	140.987204	0.827940
8	B	50.80	25.40	76.20	157.289097	32.471731	0.171119
9	B	50.80	50.80	101.60	136.476069	40.067763	0.226956
10	C	50.80	25.40	76.20	24.089570	147.308795	0.859453

Table 19 Alternatives' Top 10 Rank

Finally, after clicking the close and Run buttons, the user is presented with the a result window, as shown in Figure 16 below. The inner class function is constructed to call all necessary information into the result interface in order to present the results. The final result window is divided into three sections. The top section displays the product and thickness combination's best recommendation. Then, on the left, there is a scatter plot of the pareto set of each material's LCA and LCC, with the top ten ranks highlighted. The defuzzified weighted and normalised matrix for the top 10 rank is shown on the right. Appendix C contains the complete Python code, while Appendix B contains an enlarged view of the result interface.



Best alternative material and thickness is C (5 0 . 8 + 5 0 . 8) 1 0 1 . 6

	work_du	Durability	Water_vapour	Fir_Resis	LCC	Initial_C	LCA
Weight	0.82	0.82	0.23	0.3	0.82	0.23	0.82

	M_ID	th1	th2	THKsum	work_du	Durability	Water_vapour	Fir_Resis	LCC	Initial_C	LCA	Po_di	Ne_di	Score	Rank
12	C	50.8	50.8	101.6	0.38	0.35	0.12	0.15	16.2	3.86	77.06	10.21	165.61	0.94	1.0
11	C	50.8	39.37	90.17	0.38	0.35	0.12	0.15	15.65	3.51	86.22	13.26	157.69	0.92	2.0
14	C	39.37	39.37	78.74	0.38	0.35	0.12	0.15	15.2	3.19	97.67	22.51	147.85	0.87	3.0
10	C	50.8	25.4	76.2	0.38	0.35	0.12	0.15	11.44	2.07	100.6	24.09	147.31	0.86	4.0
5	A	76.2	76.2	152.4	0.31	0.28	0.09	0.12	11.81	2.64	103.27	26.81	144.77	0.84	5.0
4	A	76.2	63.5	139.7	0.31	0.28	0.09	0.12	11.6	2.48	104.27	27.74	144.04	0.84	6.0
6	A	63.5	63.5	127.0	0.31	0.28	0.09	0.12	11.43	2.33	106.1	29.5	142.6	0.83	7.0
7	A	63.5	101.6	165.1	0.31	0.28	0.09	0.12	19.45	5.17	102.96	29.3	140.99	0.83	8.0
3	A	76.2	38.1	114.3	0.31	0.28	0.09	0.12	9.8	1.77	108.94	32.06	141.17	0.81	9.0
2	A	76.2	25.4	101.6	0.31	0.28	0.09	0.12	8.46	1.31	113.0	36.0	138.61	0.79	10.0

Figure 16 *Final Outcome Interface*

Chapter 5 **Verification and Validation**

Materials are analysed for their average life cycle, environmental impact, and cost as part of the study's verification. Then, two distinct sensitivity analyses are performed. The first compares different life cycle years, while the second compares different TOPSIS weight inputs to observe rank changes. Lastly, the method is verified by comparing 'each material's Pareto sets +TOPSIS method' with 'TOPSIS without the Pareto set' and 'only one Pareto set of all materials +TOPSIS'.

5.1 **Material Life cycle impact**

Table 20 shows the average GHG emissions of all pareto sets of each material by life cycle stage over a 70-year period. The percentage of B6, the operation stage, is dominant, as shown in table, averaging 94 percent.

GHG	A1 ~ A3 stage (kg CO2 eq./m^2)	A4 stage (kg CO2 eq./m^2)	B6 stage (kg CO2 eq./m^2)	Total Quantity (kg CO2 eq./m^2)
A	78.81	4.46	392.96	476.24
B	17.35	4.49	491.79	513.63
C	11.97	1.17	395.97	409.11
D	3.74	0.00	550.28	554.03
Sum	111.88	10.13	1,831.01	1,953.01
Percentage	6%	1%	94%	

Table 20 Pareto Set Average of 70 years LCA by stages

Table 20 is depicted in Figure 17 below for further understanding. In Figure 17 material A-XPS has the highest proportion of embodied energy, while material D-EPS has the lowest. Material D-EPS, on the other hand, has the highest overall LCA and material C-Polyiso has the lowest overall LCA. Thus, it appears that the level of embodied energy does not directly relate to overall LCA.

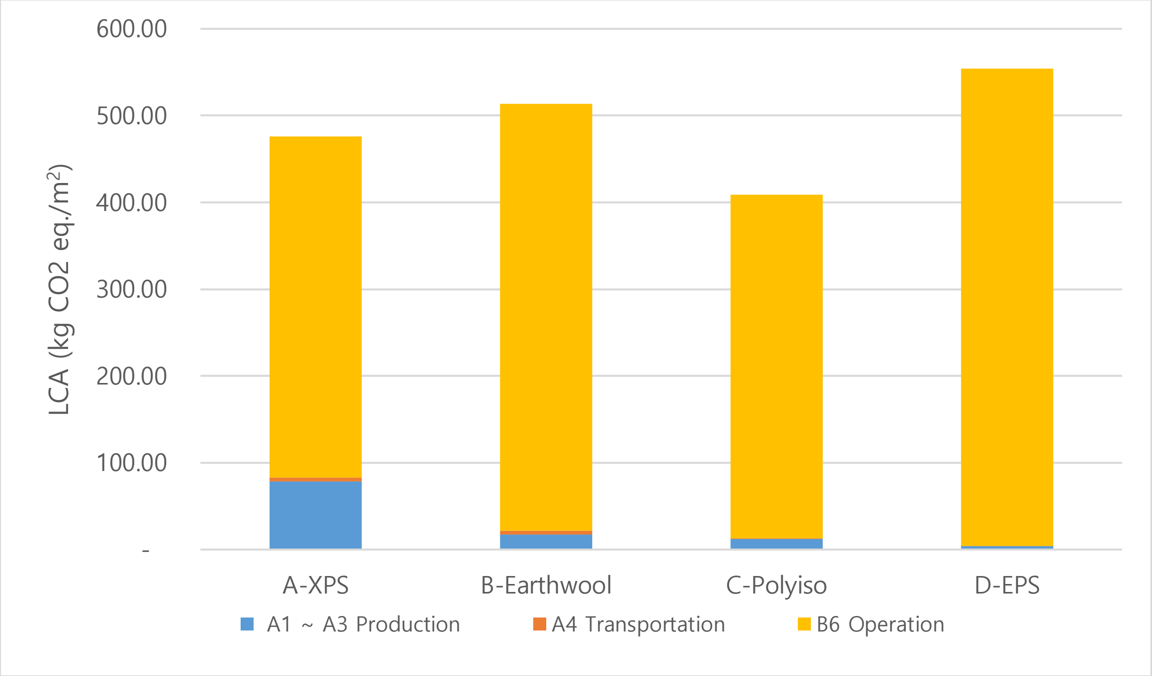


Figure 17 Pareto Set Average of 70 years LCA

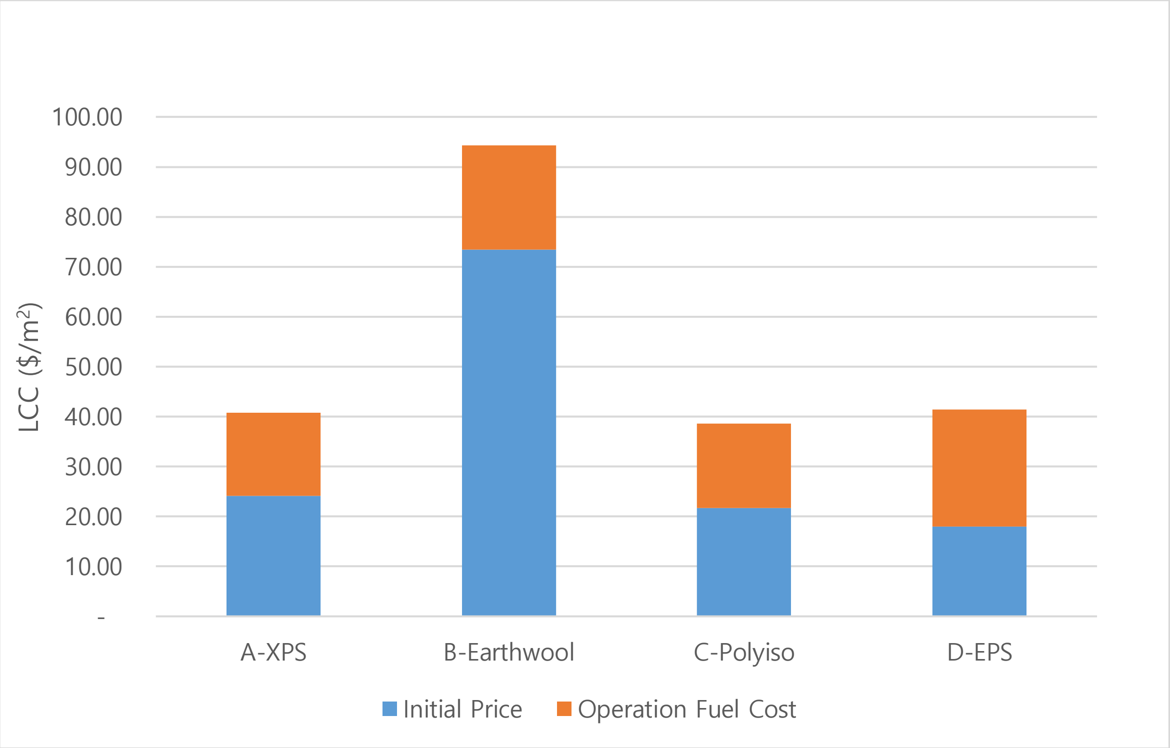
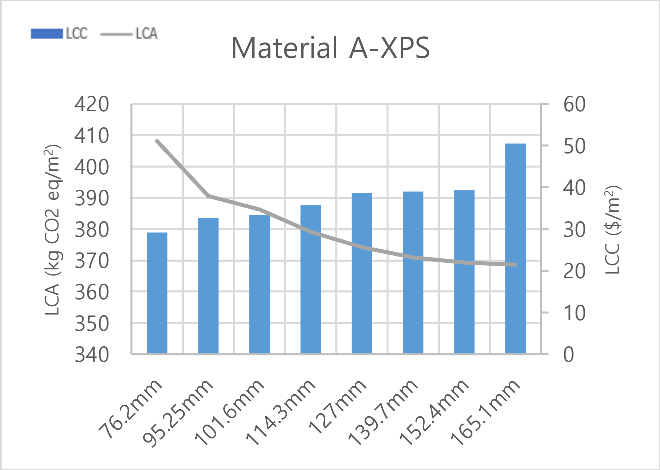


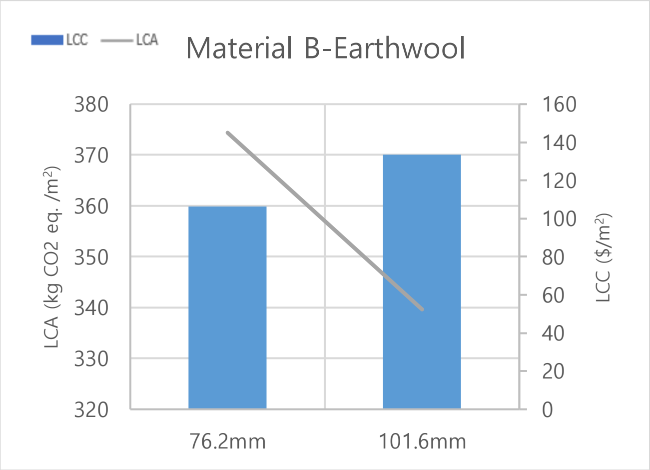
Figure 18 Pareto Set Average of 70 years LCC

As an experiment, the number of years needed for operational emissions to equal embodied emissions for each material is found using the Excel solver. To have the same amount of operational emissions for all options of material A-XPS, an average of 14.834 years is required. The B-Earthwool options take an average of 3.1 years. The C-Polyiso options take an average of 2.3 years. Finally, the D-EPS option requires 0.47 years on average. Therefore, from an environmental standpoint, material A-XPS should be used for longer-term projects, while material D-EPS could be used for shorter-term projects.

Looking at the average life cycle cost of all alternatives over 70 years, the initial price accounts for 64% of the cost and the operational fuel cost accounts for 36%. The initial cost is much higher than the operating cost. In contrast to the life cycle environmental impact, the majority of spending occurs at the initial stage rather than the operational stage. Material C-Polyiso has the lowest rating for both the LCA and the LCC, followed by Material A-XPS. Based on the above two analyses, material C-Polyiso appears to be the most sustainable material.

The correlation of LCC and LCA for the alternatives on the pareto set is examined in Figure 19. According to the analysis, LCC and LCA are in conflict with each other because a high LCC implies a low LCA.





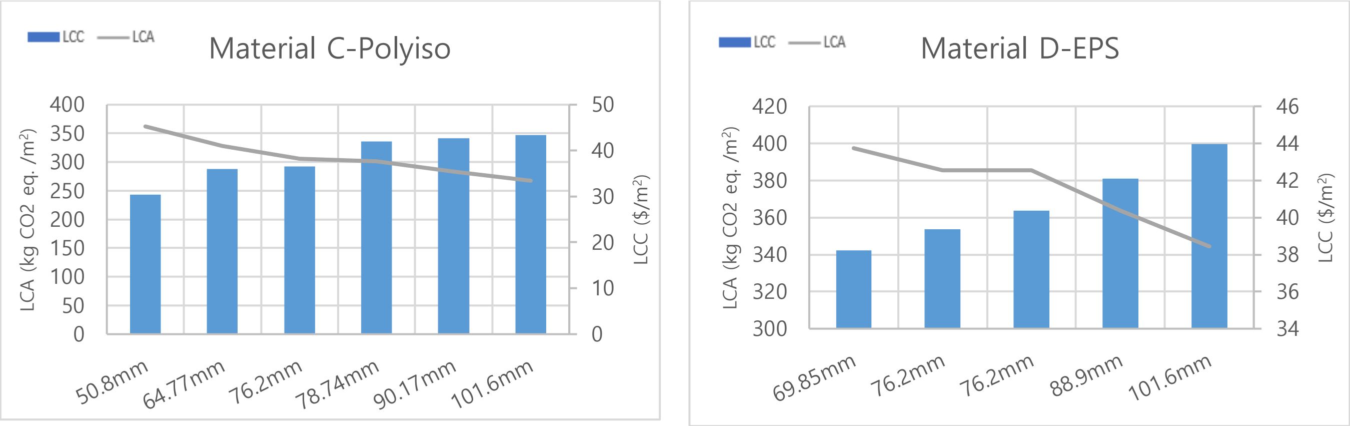


Figure 19 Thickness Correlation between LCC and LCA

Nonetheless, if the pareto set is discarded and all generated options of material A-XPS are compared, there will be cases with the same overall thicknesses but different thickness combinations. For example, the LCA of 50.8mm (50.8+0) and 50.8mm (25.4+25.4) will be the same as they have identical overall thickness. LCCs, on the other hand, will not be the same due to the commercial price difference for different combination of thicknesses. As a result, the combination of layers can also an important consideration.

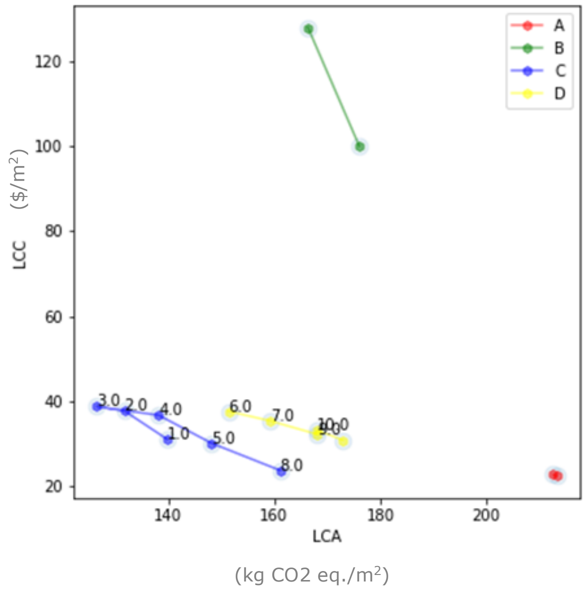
5.2 **Sensitivity analysis -Life cycle year**

The first sensitivity analysis is performed by entering different life cycle years into the program while leaving the other inputs unchanged. Life cycles of 30, 40, 50, 60 and 70 years are tested and compared.

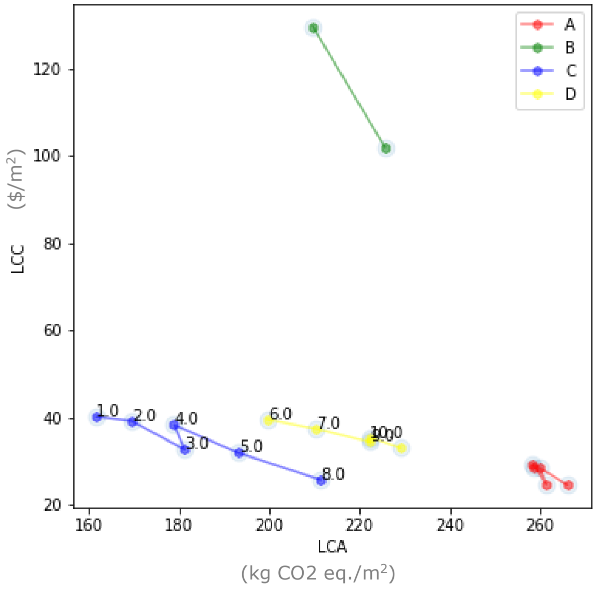
Years	**30 years**	**40 years**	**50 years**	**60 years**	**70 years**
Best Alternative	Material C 76.19mm (50.8+25.4)	Material C 101.6mm (50.8+50.8)	Material C 101.6mm (50.8+50.8)	Material C 101.6mm (50.8+50.8)	Material C 101.6mm (50.8+50.8)

Table 21 Best Alternative for Various Years

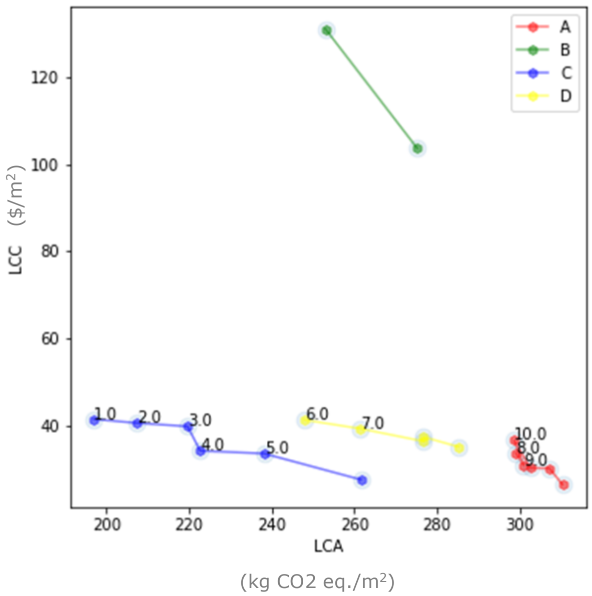
Table 21 shows that material C-Polyiso with thickness 101.6mm (50.8+50.8) option ranks at the top, except for a 30-year life cycle. Although TOPSIS had an impact on the outcome, the program recommends a thicker option for the longer terms and a thinner option for the shortest term. Figure 20 shows that when the life cycle year changes, there are slight changes in the rank of alternatives.



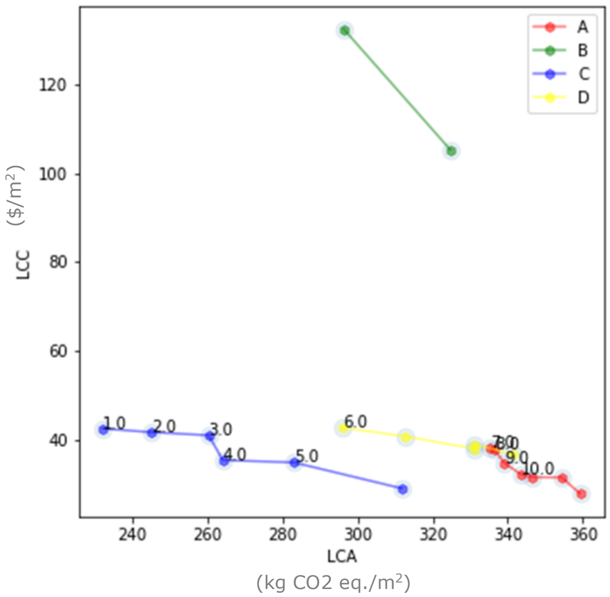
30 years



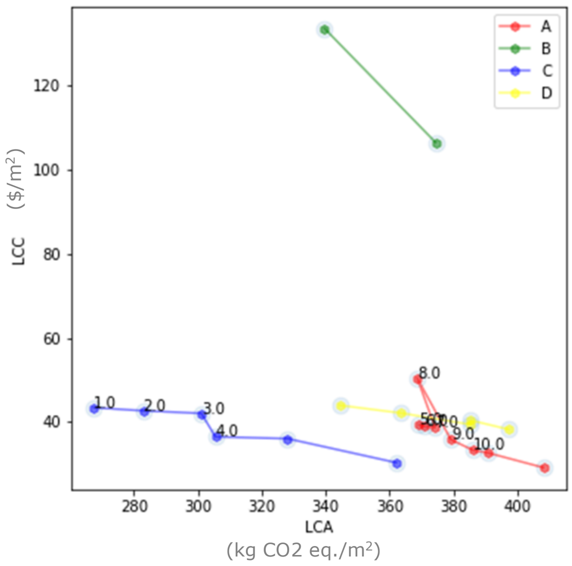
40 years



50 years



60years



70 years

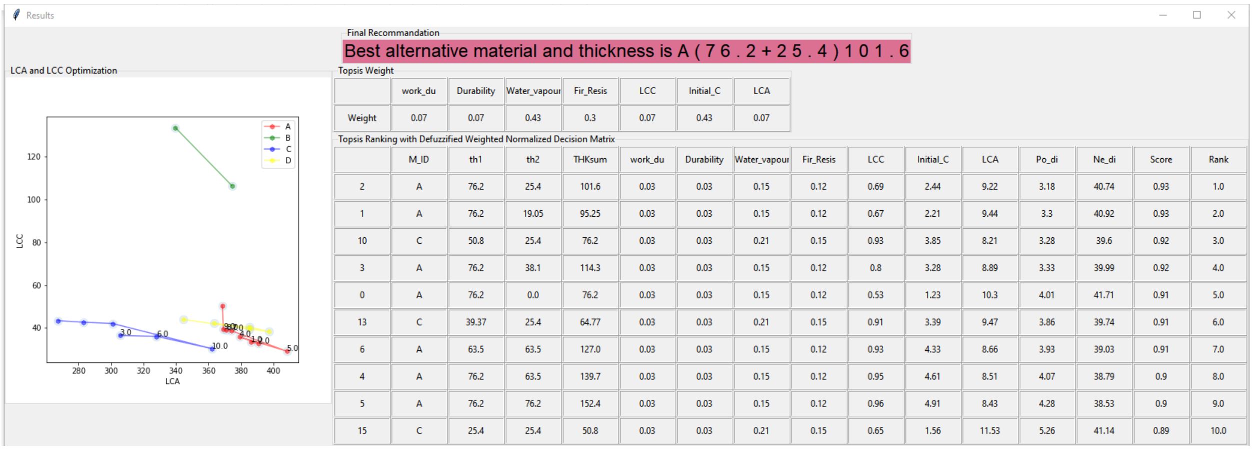
Figure 20 Alternatives in LCC and LCA Graph for Various Years

5.3 **Sensitivity analysis -TOPSIS input**

Another sensitivity analysis is performed by entering opposite TOPSIS linguistic values for weighting instead of the input from Chapter 4. For example, ‘very high’ is replaced with ‘very low’, ‘high’ with ‘low’, while ‘medium’ remains the same and rating inputs are kept the same as in Chapter 4. Table 22 shows the TOPSIS input for this sensitivity trial. Aside from the TOPSIS linguistic value, no other information has changed, including the LCA and LCC of each alternative. The result, the top 10 rank is illustrated in Figure 21 Result Interface. This trial's best recommendation is material A-XPS with a thickness of (76.2+25.4) 101.6mm. Material C-Polyiso, on the other hand, is ranked third, despite being the best recommendation in the previous trial, due to the low weight given to LCA and LCC criteria.

		DM1	DM2	DM3
C1	**Work Duration**	L	VL	VL
C2	**Durability**	VL	L	VL
C3	**Water vapor diffusion**	M	H	M
C4	**Fire Resistance**	M	M	M
C5	**LCC**	VL	VL	L
C6	**Initial Cost**	M	H	M
C7	**LCA**	L	VL	VL

Table 22 TOPSIS Input



Final Recommandation

Best alternative material and thickness is A (7 6 . 2 + 2 5 . 4) 1 0 1 . 6

Topsis Weight

	work_du	Durability	Water_vapour	Fir_Resis	LCC	Initial_C	LCA
Weight	0.07	0.07	0.43	0.3	0.07	0.43	0.07

Topsis Ranking with Defuzzified Weighted Normalized Decision Matrix

	M_ID	th1	th2	THKsum	work_du	Durability	Water_vapour	Fir_Resis	LCC	Initial_C	LCA	Po_di	Ne_di	Score	Rank
2	A	76.2	25.4	101.6	0.03	0.03	0.15	0.12	0.69	2.44	9.22	3.18	40.74	0.93	1.0
1	A	76.2	19.05	95.25	0.03	0.03	0.15	0.12	0.67	2.21	9.44	3.3	40.92	0.93	2.0
10	C	50.8	25.4	76.2	0.03	0.03	0.21	0.15	0.93	3.85	8.21	3.28	39.6	0.92	3.0
3	A	76.2	38.1	114.3	0.03	0.03	0.15	0.12	0.8	3.28	8.89	3.33	39.99	0.92	4.0
0	A	76.2	0.0	76.2	0.03	0.03	0.15	0.12	0.53	1.23	10.3	4.01	41.71	0.91	5.0
13	C	39.37	25.4	64.77	0.03	0.03	0.21	0.15	0.91	3.39	9.47	3.86	39.74	0.91	6.0
6	A	63.5	63.5	127.0	0.03	0.03	0.15	0.12	0.93	4.33	8.66	3.93	39.03	0.91	7.0
4	A	76.2	63.5	139.7	0.03	0.03	0.15	0.12	0.95	4.61	8.51	4.07	38.79	0.9	8.0
5	A	76.2	76.2	152.4	0.03	0.03	0.15	0.12	0.96	4.91	8.43	4.28	38.53	0.9	9.0
15	C	25.4	25.4	50.8	0.03	0.03	0.21	0.15	0.65	1.56	11.53	5.26	41.14	0.89	10.0

Figure 21 Result Interface

5.4 **Method verification -without Pareto**

Table 23 displays the top ranking of the same materials and inputs as Chapter 4 using the same methodology, but without sorting the pareto set. TOPSIS is the only ranking rule in this case. When the pareto search technique is not used, the ranking changes. Because of drastic changes in the cost criteria's positive ideal solutions, the majority of material A-XPS's options rose to the top of the rankings. The non-optimized alternatives have lower minimums for each cost criterion (LCC, Initial Cost, and LCA) for normalised values than optimised alternatives.

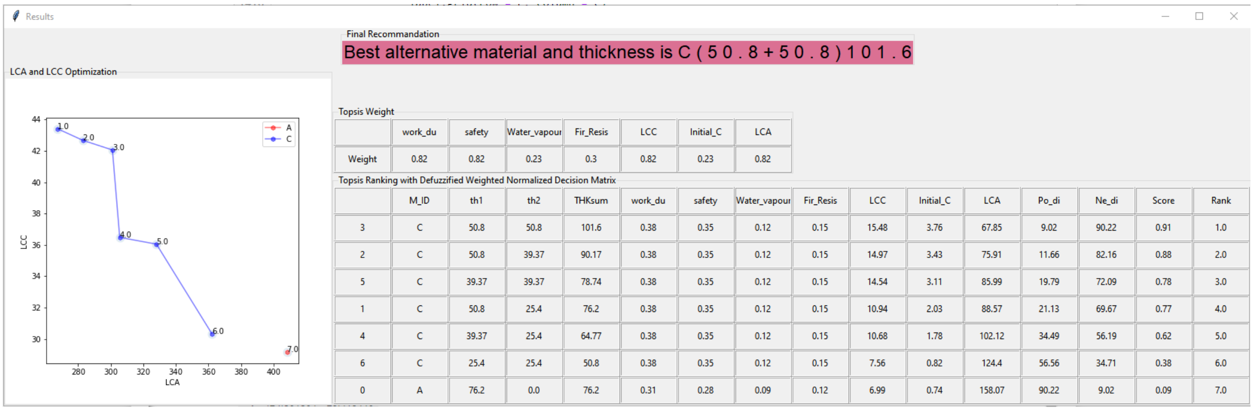
Alternatives		D*+D-	Ci	Final Ranking
A	**139.7**	193.401911	0.985652723	1
A	**152.4**	193.8661536	0.985476388	2
A	**127**	192.870949	0.984112902	3
A	**114.3**	192.4232964	0.981413901	4
A	**127**	193.6552382	0.976376269	5

A	**127**	193.7399611	0.975582177	6
A	**120.65**	193.300356	0.975193242	7
A	**114.3**	192.9978695	0.972972486	8
A	**101.6**	192.6070615	0.972208935	9
A	**139.7**	194.9289213	0.971941459	10
A	**101.6**	192.4786865	0.971354878	11
A	**101.6**	192.4224988	0.968834969	12
A	**165.1**	196.3019079	0.965486151	13
A	**95.25**	192.77689	0.965438022	14
A	**177.8**	196.2844651	0.964469596	15
A	**101.6**	193.0513738	0.958715631	16
A	**88.9**	192.8923904	0.957251347	17
A	**152.4**	197.3334657	0.955560973	18

Table 23 Ranks Without Pareto Search

5.5 **Method verification - One Pareto**

Instead of establishing a pareto set for each type of material, this time only one pareto set is used for all material types while employing the same materials and inputs as Chapter 4. As a result, just seven options remain in the pareto set, as seen in Figure 22. Among the seven, only one is material A-XPS and the rest are material C-Polyiso. Because no input information for calculation has changed since Chapter 4, the LCC and LCA values have not changed. With the same TOPSIS linguistic value input, the top ranked alternative from the system remains the same as in Chapter 4 – material C-Polyiso (50.8+50.8) 101.6mm. The rest of the rank order is the same as Chapter 4, except for the options that were deleted due to one pareto.



Final Recommandation

Best alternative material and thickness is C (5 0 . 8 + 5 0 . 8) 1 0 1 . 6

Topsis Weight

	work_du	safety	Water_vapour	Fir_Resis	LCC	Initial_C	LCA
Weight	0.82	0.82	0.23	0.3	0.82	0.23	0.82

Topsis Ranking with Defuzzified Weighted Normalized Decision Matrix

	M_ID	th1	th2	THKsum	work_du	safety	Water_vapour	Fir_Resis	LCC	Initial_C	LCA	Po_di	Ne_di	Score	Rank
3	C	50.8	50.8	101.6	0.38	0.35	0.12	0.15	15.48	3.76	67.85	9.02	90.22	0.91	1.0
2	C	50.8	39.37	90.17	0.38	0.35	0.12	0.15	14.97	3.43	75.91	11.66	82.16	0.88	2.0
5	C	39.37	39.37	78.74	0.38	0.35	0.12	0.15	14.54	3.11	85.99	19.79	72.09	0.78	3.0
1	C	50.8	25.4	76.2	0.38	0.35	0.12	0.15	10.94	2.03	88.57	21.13	69.67	0.77	4.0
4	C	39.37	25.4	64.77	0.38	0.35	0.12	0.15	10.68	1.78	102.12	34.49	56.19	0.62	5.0
6	C	25.4	25.4	50.8	0.38	0.35	0.12	0.15	7.56	0.82	124.4	56.56	34.71	0.38	6.0
0	A	76.2	0.0	76.2	0.31	0.28	0.09	0.12	6.99	0.74	158.07	90.22	9.02	0.09	7.0

Figure 22 Result Interface- One Pareto

5.6 **Experts’ validation**

In order to obtain an opinion and validation from experts, two meetings were conducted. The meetings were primarily focused on the validation of the approach and the selection of criteria. Expert 1 is an experienced program manager in the Architecture area. Expert 2 works in the public sector for building sustainability. Both have prior knowledge with embodied energy and life cycle assessment. They agreed that, in addition to renewable energy for operational energy, embodied energy is becoming increasingly important in material selection.

Furthermore, they advised on criteria selection, especially for the importance of water vapor diffusion, material durability, and fire resistance. The first meeting, with Expert 1, took place while the methodology and software were still in the early stages of development. The next meeting with Expert 2 took place after the methodology and software had been developed. Therefore, the first meeting was mostly focused on the direction of the study and the criteria of TOPSIS. In the second meeting, the reason behind the double layer limit for the alternatives’ thickness was explained, and the expert agreed that the constructability of more than a double-layer could be challenging. The expert also pointed out the missing consideration of thermal bridges. The thermal bridge was not taken into account because the study is based on continuous sheathing insulation, and thermal bridging impact assessments are case-specific, it is difficult to generalize. Furthermore, this decision support system ranks the alternatives by comparing options rather than measuring the actual quantity. Finally, it was discussed whether the software and methodology could be applied to real case scenarios. The experts agreed that this method suggested a new dimension on sustainable material selection.

5.7 **Summary**

From the above verifications, material C-Polyiso appeared to be the best material from the material analysis when only looking at the material-to-material comparison for sustainability. Also, the same overall thickness with different thickness combinations resulted in a change in

LCC owing to commercial cost variances, which showed that thickness combination can also be important. Following that, the ‘life cycle year sensitivity analysis' revealed that the life cycle year is also a component to consider when making decisions. Material C-Polyiso 76.19mm (50.8+25.4) was the best material and thickness before 30 years, while Material C-Polyiso 101.6mm (50.8+50.8) became the best material and thickness for 40 years and up, according to the analysis. Also, a sensitivity analysis was done with the opposite set of TOPSIS inputs, which revealed different ranking orders within the same pareto set, as expected. Hence, it can be said that the TOPSIS gives decision-makers flexibility within the pareto set.

Finally, the meaning of the pareto set in this method was examined by omitting the pareto search strategy and by creating a single pareto set for all materials rather than creating individual pareto sets for each material. When no pareto set was applied, the program recommended material A-XPS instead of material C-Polyiso due to changes in positive ideal solutions in LCC, Initial cost, and LCA criteria. Without the pareto set, the system is strongly reliant on the opinions of decision makers. When only one pareto set is created for all materials, however, the majority of the remaining options in the pareto set came from material C-Polyiso. As a result, TOPSIS freedom is constrained in this case.

Chapter 6 **Final Discussion**

This chapter summarizes all the work conducted in this research. Section 6.1 presents the final discussion and summary. Section 6.2 outlines the contribution of this thesis. Finally, Section 6.3 discusses the current study's limitations and provides some recommendations for further research.

6.1 **Final discussion and summary**

The research started from the question of 'what is sustainability in a building?' In considering this question, focus was placed on one of the bulkiest materials in a building, the building envelope. Especially in building envelopes, insulation is the critical element for operational energy. Hence the subject material of study was chosen. Sustainability in a building is commonly understood as a low energy requirement for building operation, but embodied energy is reported high in low energy buildings. Although embodied energy is widely studied as part of LCA, it has not been a primary concern when selecting materials. To incorporate the embodied energy aspect into the decision-making process, the decision support system is suggested in this study.

Recently, more experts are including LCC in their decision-making process. On the other hand, LCA has not been much utilised in the decision-making process. There has been some effort to include LCA as a decision-making tool by integrating and optimizing LCA and LCC. However, due to the difficulty of assessing LCA, it has not been widely used in decision-making. The study utilized EPD documents to address this difficulty. Some manufacturers voluntarily publish EPDs for their products in accordance with ISO requirements, which include quantity of product's embodied energy from raw materials, manufacturing, and transportation.

By optimising the LCA and LCC of each candidate material and then adopting the TOPSIS technique, this thesis developed decision-making supporting tools. The reason for optimising each material rather than all materials at once is to offer freedom for material ratings in TOPSIS. If optimizing all materials at once, only one or two materials would remain in the pareto front. Those materials that were removed from consideration for TOPSIS evaluations could achieve high ratings in the TOPSIS.

Python was used to create a software application for the framework. The application required three types of data: project information, material information for LCA and LCC optimization, and experts' language preferences for TOPSIS. After running the application, it displayed the final recommendation along with ranks. In order to confirm the software result, a manual calculation was performed on the excel sheet as attached in Appendix A.

The average LCA and LCC of each material's pareto set were compared as part of the verification and validation procedure, and the most sustainable material was ascertained from among the materials, namely material C-Polyiso.

In the calculation, the study assumed that only up to two layers of the same product could be used. When the same overall thicknesses with different set of thickness combinations were compared, there were LCC differences between two of the same overall thickness combinations due to the commercial pricing variations of each thickness. Therefore, the study was designed to let the user know the thickness combination as well. Finally, sensitivity analysis was performed on the life cycle year and TOPSIS input. The life cycle year setting impacted thickness recommendations within the same material, but changes in the TOPSIS input resulted in changes to the material.

Additionally, the method was tested first by skipping optimization and only applying TOPSIS techniques. Due to non-optimized alternatives, the result gave different material recommendations. When the pareto set was employed as alternatives, the system found more sustainable materials in terms of LCA and LCC. Secondly, one pareto set for all materials was tested and this resulted in the elimination of most alternatives during the pareto search. Such a case limited the freedom of TOPSIS.

To sum up, a framework and software prototype has been suggested in this thesis. This should help find the most sustainable insulation material within commercially available options. Also, the framework is tailored to the subjectivity of a given project by incorporating the TOPSIS technique while keeping objectivity by limiting the alternatives to pareto set for minimum LCA and LCC.

6.2 **Contribution**

- Incorporate Environmental Product Declaration (EPD) documents with material selection.

Even though some manufacturers publish product EPD, these have not been utilized in the material selections decision-making process. By including EPD to LCA calculation, embodied energy is included for material selection.

- Utilizing Pareto front and TOPSIS technique in selection

Other material selection support systems were reviewed in chapter 2, and some of them are based on optimization, environmental impact comparison, or MCDM methodologies. The thesis suggests merging the pareto search technique of multi-objective optimization and TOPSIS techniques to find the best material and thicknesses, adding more objectivity to the TOPSIS.

- Commercial applications

The uniqueness of this thesis is that the framework requires specific product EPD and recommends the commercially available material and thickness.

6.3 **Limitation and further studies**

One of the difficulties in doing the research was obtaining the EPDs for the North American market because not all manufacturers publish EPDs. Even if an EPD is made public, the documents are not always in the same format. Therefore, the subject material studied was naturally limited to those for which EPDs were accessible. Furthermore, some EPDs do not contain all the necessary information, especially for the C1-C4 stage (disposal) stage. Therefore, the C1-C4 stage (disposal) stage was excluded from the study. Although ISO initiated an integrated EPD database to unify all EPDs, it is still under construction.

Another limitation of this study is the thermal bridge. It is hard to generalize each thermal bridge case in LCC and LCA calculations. The study was limited to continuous sheathing insulation to mitigate this problem, but thermal bridges can occur for continuous sheathing insulation. Nevertheless, the decision support system is to find the best material by comparing alternatives; it is not for quantifying the exact LCC and LCA.

To improve holistic estimates in LCA, adding the A5 stage (during construction) and the C1-C4 stage (disposal) stage can be attempted in further studies. While emissions during the construction stages may be minor, the overall impact on the environment is substantial. Also, the environmental impact of disposal is a critical topic for sustainability. Therefore, as a following study it is recommended to include disposals in the decision-making process. Also, the same technique can be applied for selecting other building material than insulation.

Finally, the software developed in this thesis is a prototype that can be improved on in a variety of ways to make it more user-friendly. Currently, it is set up to accept numerical values for all basic data of the project. However, heating degree days could be replaced by selecting a region. The fuel’s emission rates, and energy content could be replaced with a simple selection of fuel types.

References

Al-Hussein, M., Manrique, J., & Mah, D. (2009). North Ridge CO2 analysis report: com parison between modular and on site construction. University of Alberta, Canada, 1–20. https://www.modular.org/IMAGES/foundation/NorthRidgeCO2Report.pdf

ASHRAE. (2013). Energy standard for buildings except low-rise residential buildings. In ASHRAE Standard.

BILD Alberta. (n.d.). BILD Alberta – One Voice, Best Source. Retrieved October 25, 2020, from https://bildalberta.ca/

BSI. (2014). BS EN 15804:2012+A1:2013. Sustainability of construction works — Environmental product declarations — Core rules for the product category of construction products. International Standard, February.

Building Science Corporation. (2007). Guide to Insulating Sheathing - Wall insulation. January, 1–30. https://buildingscience.com/sites/default/files/migrate/pdf/GM_Guide_Insulating_Sheathing.pdf

Burke, R. D., & Kristen, P. (2008). System Engineering Analysis Approach to Building Material Selection for Sustainable Buildings. Journal of Comparative Neurology, 506(5), 759–774. https://doi.org/10.1002/cne.21575

Calado, E. A., Leite, M., & Silva, A. (2018). Selecting composite materials considering cost and environmental impact in the early phases of aircraft structure design. Journal of Cleaner Production, 186, 113–122. https://doi.org/10.1016/j.jclepro.2018.02.048

Castro-Lacouture, D., Sefair, J. A., Flórez, L., & Medaglia, A. L. (2009). Optimization model for the selection of materials using a LEED-based green building rating system in Colombia.

Building and Environment, 44(6), 1162–1170. https://doi.org/10.1016/j.buildenv.2008.08.009

Chen, P. H., & Nguyen, T. C. (2019). A BIM-WMS integrated decision support tool for supply chain management in construction. Automation in Construction, 98 (August 2018), 289–301. https://doi.org/10.1016/j.autcon.2018.11.019

Chen, Z., Martínez, L., Chang, J., Wang, X., & Xionge, S. (2019). Engineering Applications of Artificial Intelligence Sustainable building material selection : A QFD- and ELECTRE III-embedded hybrid MCGDM approach with consensus building. Engineering Applications of Artificial Intelligence, 85 (December 2018), 783–807. https://doi.org/10.1016/j.engappai.2019.08.006

Chiandussi, G., Codegone, M., Ferrero, S., & Varesio, F. E. (2012). Comparison of multi-objective optimization methodologies for engineering applications. In Computers and Mathematics with Applications (Vol. 63, Issue 5). Elsevier Ltd. https://doi.org/10.1016/j.camwa.2011.11.057

ColoradoENERGY.org (n.d.) R-Value Table. Retrieved June 20, 2021, from https://www.coloradoenergy.org/procorner/stuff/r-values.htm

Custódio, A. L., Madeira, J. F. A., Vaz, A. I. F., & Vicente, L. N. (2011). Direct multisearch for multiobjective optimization. SIAM Journal on Optimization, 21 (3), 1109–1140. https://doi.org/10.1137/10079731X

edmonton.weatherstats.ca. (n.d.). Heating Degree Days (18°C) - Monthly data for Edmonton. Retrieved November 10, 2020, from https://edmonton.weatherstats.ca/charts/hdd-yearly.html

Elisabeth Girgis. (n.d.). National Energy Code for Buildings (NECB) Overview. https://www.alberta.ca/assets/documents/ma-national-energy-code-for-buildings-overview.pdf

Environment and Climate Change Canada - Canada.ca. (n.d.). Audit at a glance: Joint audit and evaluation of the management of the Pan-Canadian Framework, Retrieved July 28, 2021, from https://www.canada.ca/en/environment-climate-change/corporate/transparency/priorities-management/internal-audits/audit-evaluation-management-pan-canadian-framework.html

EPD International AB. (2019). What is an EPD? - The International EPD® System. https://www.environdec.com/What-is-an-EPD/

Fang, Y., Lu, X., & Zhang, Y. (2019). Carbon Emissions Analysis for Construction Process. Iccrem 2019 648. ICCREM, 648–654.

Feist, W. (n.d.). Passivhaus Institut. Passiv.De. Retrieved October 24, 2020, from https://passivehouse.com/

Florez, L., & Castro-Lacouture, D. (2013). Optimization model for sustainable materials selection using objective and subjective factors. Materials and Design, 46, 310–321. https://doi.org/10.1016/j.matdes.2012.10.013

Fu, Y., Dong, N., Ge, Q., Xiong, F., & Gong, C. (2020). Driving-paths of green buildings industry (GBI) from stakeholders' green behavior based on the network analysis. Journal of Cleaner Production, 273. https://doi.org/10.1016/j.jclepro.2020.122883

Gbadebo, M., & Ajibike, W. A. (2019). Analysis of some factors driving ecological sustainability in construction firms. Journal of Cleaner Production, 208, 1537–1545. https://doi.org/10.1016/j.jclepro.2018.10.229

Gholizadeh, P., Behzad, E., & Memarian, B. (2018). Construction Research Congress 2018 725. Construction Research Congress 2018 Downloaded, 2010(2012), 725–735.

Giorgi, S., Lavagna, M., & Campioli, A. (2019). LCA and LCC as decision-making tools for a sustainable circular building process. IOP Conference Series: Earth and Environmental Science, 296(1). https://doi.org/10.1088/1755-1315/296/1/012027

GreenSpec. (2007). Product category rules (pcr). https://www.greenspec.co.uk/product-category-rules-pcr/

Hafner, A., & Storck, M. (2019). Life Cycle Analysis of Vertical Building Extensions – Environmental Impacts of Different Material Selection Life Cycle Analysis of Vertical Building Extensions – Environmental Impacts of Different Material Selection.2019 IOP Conf. Ser.: Earth Environ. Sci. 290 012046. https://doi.org/10.1088/1755-1315/290/1/012046

Hammond, G. P., & Jones, C. I. (2008). Embodied energy and carbon in construction materials. Proceedings of Institution of Civil Engineers: Energy, 161(2), 87–98. https://doi.org/10.1680/ener.2008.161.2.87

Higham, A. P., & Fortune, C. (2016). Sustainability and investment appraisal for housing regeneration projects. https://doi.org/10.1108/SS-09-2015-0044

ISO 14044:2006 + A1:2018 Environmental management - Life cycle assessment - Requirements and guidelines, International Organization for Standardization (ISO) (2018).

ISO. (2020). ISO/CD 22057 Enabling use of Environmental Product Declarations (EPD) at construction works level using building information modelling (BIM). https://www.iso.org/standard/72463.html

Jaini, N. I., & Utyuzhnikov, S. v. (2017). A fuzzy trade-off ranking method for multi-criteria decision-making. Axioms, 7(1), 1–21. https://doi.org/10.3390/axioms7010001

Kallioğlu, M. A., Ercan, U., Avcı, A. S., Fidan, C., & Karakaya, H. (2020). Empirical modeling between degree days and optimum insulation thickness for external wall. Energy Sources, Part A: Recovery, Utilization and Environmental Effects, 42(11), 1314–1334. https://doi.org/10.1080/15567036.2019.1651797

Kawecki, L. R. (2010). Environmental Performance of Modular Fabrication: Calculating the Carbon Footprint of Energy Used in the Construction of a Modular Home. In ARIZONA STATE UNIVERSITY UMI Number: 3432331.

Kim, S., Choi, M., Mha, H., & Joung, J. (2013). Environmental impact assessment and eco-friendly decision-making in civil structures. Journal of Environmental Management, 126, 105–112. https://doi.org/10.1016/j.jenvman.2013.03.045

Leite, M., Silva, A., Henriques, E., & A. Madeira, J. F. (2015). Materials selection for a set of multiple parts considering manufacturing costs and weight reduction with structural isoperformance using direct multisearch optimization. Structural and Multidisciplinary Optimization, 52(4), 635–644. https://doi.org/10.1007/s00158-015-1247-7

Li, H. X., Patel, D., Al-Hussein, M., Yu, H., & Gül, M. (2018). Stakeholder studies and the social networks of NetZero energy homes (NZEHs). Sustainable Cities and Society, 38(October 2017), 9–17. https://doi.org/10.1016/j.scs.2017.12.014

Li, H., Wang, W., Fan, L., Li, Q., & Chen, X. (2020). A novel hybrid MCDM model for machine tool selection using fuzzy DEMATEL, entropy weighting and later defuzzification VIKOR. Applied Soft Computing Journal, 91, 106207. https://doi.org/10.1016/j.asoc.2020.106207

Li, S., Lu, Y., Kua, H. W., & Chang, R. (2020). The economics of green buildings: A life cycle cost analysis of non-residential buildings in tropic climates. Journal of Cleaner Production, 252, 119771. https://doi.org/10.1016/j.jclepro.2019.119771

Mahmoudkelaye, S., Azari, K. T., & Pourvaziri, M. (2018). Case Studies in Construction Materials Sustainable material selection for building enclosure through ANP method Impact assessment. 9.

Masanet, E., & Chang, Y. (2014). Who cares about life cycle assessment?: A survey of 900 prospective life cycle assessment practitioners masanet and chang who cares about LCA? Journal of Industrial Ecology, 18(6), 787–791. https://doi.org/10.1111/jiec.12189

Miah, J. H., Koh, S. C. L., & Stone, D. (2017). A hybridised framework combining integrated methods for environmental Life Cycle Assessment and Life Cycle Costing. Journal of Cleaner Production, 168, 846–866. https://doi.org/10.1016/j.jclepro.2017.08.187

NAIMA Canada. (2018). Comparing Insulation Types for Professionals. https://www.naimacanada.ca/for-professionals/building-insulation/comparing-insulation-types/

Natural Resources Canada. (n.d.). Canada's GHG Emissions by Sector, End Use and Subsector – Including Electricity-Related Emissions Retrieved July 29, 2021, from https://oee.nrcan.gc.ca/corporate/statistics/neud/dpa/showTable.cfm?type=HB&sector=aaa&juris=ca&rn=3&page=0

Nofal, A., & Hammad, A. (2020). APPLICATION OF FUZZY TOPSIS FOR SELECTING MOST SUSTAINABLE BUILDING WALL MATERIAL. Mcdm, 1–6.

Odland, T. (2021). paretoset · PyPI (1.2.0). https://pypi.org/project/paretoset/

Ontario Energy Board. (2020). Natural gas rates | Ontario Energy Board. https://www.oeb.ca/rates-and-your-bill/natural-gas-rates

Opher, T., Duhamel, M., Posen, I. D., Panesar, D. K., Brugmann, R., Roy, A., Zizzo, R., Sequeira, L., Anvari, A., & MacLean, H. L. (2021). Life cycle GHG assessment of a building restoration: Case study of a heritage industrial building in Toronto, Canada. Journal of Cleaner Production, 279, 123819. https://doi.org/10.1016/j.jclepro.2020.123819

Ozturk, H. M., Dombayci, O. A., & Caliskan, H. (2019). Life-Cycle Cost , Cooling Degree Day , and Carbon Dioxide Emission Assessments of Insulation of Refrigerated Warehouses Industry in Turkey. 145(Cdd), 1–8. https://doi.org/10.1061/(ASCE)EE.1943-7870.0001575

Perera, O., Morton, B., & Perfrement, T. (2009). Life Cycle Costing in Sustainable Public Procurement - A Question of Value. 1–19.

pyodbc · PyPI. (n.d.). Retrieved August 7, 2021, from https://pypi.org/project/pyodbc/

Ramesh, T., Prakash, R., & Shukla, K. K. (2010). Life cycle energy analysis of buildings: An overview. Energy and Buildings, 42(10), 1592–1600. https://doi.org/10.1016/j.enbuild.2010.05.007

Rao, R. V. (2007). Decision making in the manufacturing environment: using graph theory and fuzzy multiple attribute decision making methods. In Springer Science & Business Media.

Raouf, A. M., & Al-Ghamdi, S. G. (2020). Effect of R-values changes in the baseline codes: Embodied energy and environmental life cycle impacts of building envelopes. Energy Reports, 6, 554–560. https://doi.org/10.1016/j.egyr.2019.09.025

Rigid Board Installation Builder’s Guide (pp. 1–42). (2018). Amvic System Inc.

Ruzgys, A., Volvačiovas, R., Ignatavičius, Č., & Turskis, Z. (2014). Integrated evaluation of external wall insulation in residential buildings using SWARA-TODIM MCDM method. Journal of Civil Engineering and Management, 20(1), 103–110. https://doi.org/10.3846/13923730.2013.843585

Santoyo-Castelazo, E., & Azapagic, A. (2014). Sustainability assessment of energy systems: Integrating environmental, economic and social aspects. Journal of Cleaner Production, 80(2014), 119–138. https://doi.org/10.1016/j.jclepro.2014.05.061

Schwartz, Y., Eleftheriadis, S., Raslan, R., & Mumovic, D. (2016). Semantically Enriched BIM Life Cycle Assessment to Enhance Buildings' Environmental Performance. CIBSE Technical Symposium, Edinburgh, UK, April, 14 pages.

Shadram, F., Sandberg, M., Schade, J., & Olofsson, T. (2014). BIM-based environmental assessment in the building design process. International Conference on Construction Applications of Virtual Reality in Construction and Islamic Architecture, Fava.

Stanley, S. (2020). Ireland reaches green building milestone with 100 LEED-certified projects. USGBC. https://www.usgbc.org/articles/ireland-reaches-green-building-milestone-100-leed-certified-projects

Statista. (n.d.). Canada - Inflation rate 1986-2026. Retrieved July 23, 2021, from https://www.statista.com/statistics/271247/inflation-rate-in-canada/

Stiel, F., Michel, T., & Teuteberg, F. (2016). Enhancing manufacturing and transportation decision support systems with LCA add-ins. Journal of Cleaner Production, 110, 85–98. https://doi.org/10.1016/j.jclepro.2015.07.140

Strömberg, L. (2017). Conceptual Framework for Calculation of Climate Performance with Pre-verified LCA-Tools. Journal of Civil Engineering and Architecture, 11(1), 29–37. https://doi.org/10.17265/1934-7359/2017.01.004

Strömberg, L. (2017). Current difficulties with creation of standardized digital climate calculations for infrastructural projects. Pavement Life-Cycle Assessment - Proceedings of the Pavement Life-Cycle Assessment Symposium, 2017, February, 23–29. https://doi.org/10.1201/9781315159324-4

Strömberg, L. (n.d.). Web based EPD-tool for evaluation of contractors ’ design.

Takano, A., Kumar, S., Kuittinen, M., Alanne, K., Hughes, M., & Winter, S. (2015). The effect of material selection on life cycle energy balance : A case study on a hypothetical building model in Finland. Building and Environment, 89, 192–202. https://doi.org/10.1016/j.buildenv.2015.03.001

Thormark, C. (2002). A low energy building in a life cycle - Its embodied energy, energy need for operation and recycling potential. Building and Environment, 37(4), 429–435. https://doi.org/10.1016/S0360-1323(01)00033-6

tkinter — Python interface to Tcl/Tk — Python 3.9.6 documentation. (n.d.). Retrieved August 7, 2021, from https://docs.python.org/3/library/tkinter.html

UL. (2013). Product Category Rules Concrete. UL. https://www.ul.com/offerings/product-category-rules-pcrs

Vahdani, B., Mousavi, S. M., & Tavakkoli-Moghaddam, R. (2011). Group decision making based on novel fuzzy modified TOPSIS method. Applied Mathematical Modelling, 35(9), 4257–4269. https://doi.org/10.1016/j.apm.2011.02.040

Wao, J. O. W. (2017). Aei 2017 1091. AEI 2017, 1091–1102.

Willms, D. (2007). What is your carbon footprint Chem 13 News, 17(November), 16–17. http://en.wikipedia.org/wiki/Fuel_efficiency_in_transportation%0Ahttps://uwaterloo.ca/chem13news/sites/ca.chem13news/files/uploads/files/pages16_17_351.pdf

World Green Building Council. (2019). New report: the building and construction sector can reach net zero carbon emissions by 2050. https://www.worldgbc.org/news-media/WorldGBC-embodied-carbon-report-published

Appendix A: Demonstration in table

1. Generated Alternatives

ID	Material	Material	1st Layer	2nd Layer	Thickness
1	xps	A	19.05		19.05
2	xps	A	25.4		25.4
3	xps	A	38.1		38.1
4	xps	A	50.8		50.8
5	xps	A	63.5		63.5
6	xps	A	76.2		76.2
7	xps	A	101.6		101.6
8	xps	A	19.05	19.05	38.1
9	xps	A	19.05	25.4	44.45
10	xps	A	19.05	38.1	57.15
11	xps	A	19.05	50.8	69.85
12	xps	A	19.05	63.5	82.55
13	xps	A	19.05	76.2	95.25
14	xps	A	19.05	101.6	120.65
15	xps	A	25.4	25.4	50.8
16	xps	A	25.4	38.1	63.5
17	xps	A	25.4	50.8	76.2
18	xps	A	25.4	63.5	88.9
19	xps	A	25.4	76.2	101.6
20	xps	A	25.4	101.6	127
21	xps	A	38.1	38.1	76.2
22	xps	A	38.1	50.8	88.9
23	xps	A	38.1	63.5	101.6
24	xps	A	38.1	76.2	114.3
25	xps	A	38.1	101.6	139.7
26	xps	A	50.8	50.8	101.6
27	xps	A	50.8	63.5	114.3
28	xps	A	50.8	76.2	127
29	xps	A	50.8	101.6	152.4
30	xps	A	63.5	63.5	127
31	xps	A	63.5	76.2	139.7
32	xps	A	63.5	101.6	165.1
33	xps	A	76.2	76.2	152.4
34	xps	A	76.2	101.6	177.8
35	xps	A	101.6	101.6	203.2
36	Fiberglass	B	25.4		25.4
37	Fiberglass	B	50.8		50.8
38	Fiberglass	B	25.4	25.4	50.8
39	Fiberglass	B	25.4	50.8	76.2
40	Fiberglass	B	50.8	50.8	101.6

41	Polyiso	C	25.4		25.4
42	Polyiso	C	39.37		39.37
43	Polyiso	C	50.8		50.8
44	Polyiso	C	25.4	25.4	50.8
45	Polyiso	C	25.4	39.37	64.77
46	Polyiso	C	25.4	50.8	76.2
47	Polyiso	C	39.37	39.37	78.74
48	Polyiso	C	39.37	50.8	90.17
49	Polyiso	C	50.8	50.8	101.6
50	EPS	D	19.05		19.05
51	EPS	D	25.4		25.4
52	EPS	D	38.1		38.1
53	EPS	D	50.8		50.8
54	EPS	D	19.05	19.05	38.1
55	EPS	D	25.4	19.05	44.45
56	EPS	D	38.1	19.05	57.15
57	EPS	D	50.8	19.05	69.85
58	EPS	D	25.4	25.4	50.8
59	EPS	D	38.1	25.4	63.5
60	EPS	D	50.8	25.4	76.2
61	EPS	D	38.1	38.1	76.2
62	EPS	D	50.8	38.1	88.9
63	EPS	D	50.8	50.8	101.6

2. Alternatives over minimum thickness

ID	Material	Material	1st Layer	2nd Layer	Thickness
Material A -XPS					
5	xps	A	63.5	0	63.5
6	xps	A	76.2	0	76.2
7	xps	A	101.6	0	101.6
10	xps	A	19.05	38.1	57.15
11	xps	A	19.05	50.8	69.85
12	xps	A	19.05	63.5	82.55
13	xps	A	19.05	76.2	95.25
14	xps	A	19.05	101.6	120.65
16	xps	A	25.4	38.1	63.5
17	xps	A	25.4	50.8	76.2
18	xps	A	25.4	63.5	88.9
19	xps	A	25.4	76.2	101.6
20	xps	A	25.4	101.6	127
21	xps	A	38.1	38.1	76.2
22	xps	A	38.1	50.8	88.9
23	xps	A	38.1	63.5	101.6
24	xps	A	38.1	76.2	114.3

25	xps	A	38.1	101.6	139.7
26	xps	A	50.8	50.8	101.6
27	xps	A	50.8	63.5	114.3
28	xps	A	50.8	76.2	127
29	xps	A	50.8	101.6	152.4
30	xps	A	63.5	63.5	127
31	xps	A	63.5	76.2	139.7
32	xps	A	63.5	101.6	165.1
33	xps	A	76.2	76.2	152.4
34	xps	A	76.2	101.6	177.8
35	xps	A	101.6	101.6	203.2
Material B-Earthwool					
39	Fiberglass	B	25.4	50.8	76.2
40	Fiberglass	B	50.8	50.8	101.6
Material C-Polyiso					
42	Polyiso	C	39.37	0	39.37
43	Polyiso	C	50.8	0	50.8
44	Polyiso	C	25.4	25.4	50.8
45	Polyiso	C	25.4	39.37	64.77
46	Polyiso	C	25.4	50.8	76.2
47	Polyiso	C	39.37	39.37	78.74
48	Polyiso	C	39.37	50.8	90.17
49	Polyiso	C	50.8	50.8	101.6
Material D-EPS					
57	EPS	D	50.8	19.05	69.85
60	EPS	D	50.8	25.4	76.2
61	EPS	D	38.1	38.1	76.2
62	EPS	D	50.8	38.1	88.9
63	EPS	D	50.8	50.8	101.6

3. Pareto Set

ID	**Material**	**Material**	**Thickness**	**LCA**	**LCC**
Material -XPS					
6	xps	A	76.2	408.16	29.15
13	xps	A	95.25	390.60	32.71
19	xps	A	101.6	386.20	33.27
24	xps	A	114.3	379.20	35.80
30	xps	A	127	374.24	38.66
31	xps	A	139.7	370.99	38.95
32	xps	A	165.1	368.64	50.44
33	xps	A	152.4	369.19	39.30
Material -Earthwool					

39	Fiberglass	B	76.2	374.38	106.47
40	Fiberglass	B	101.6	339.64	133.53
Material -Polyiso					
44	Polyiso	C	50.8	362.09	30.33
45	Polyiso	C	64.77	328.06	36.04
46	Polyiso	C	76.2	305.52	36.48
47	Polyiso	C	78.74	301.04	42.05
48	Polyiso	C	90.17	282.84	42.67
49	Polyiso	C	101.6	267.41	43.41
Material -EPS					
57	EPS	D	69.85	397.39	38.22
60	EPS	D	76.2	385.39	39.38
61	EPS	D	76.2	385.39	40.38
62	EPS	D	88.9	363.61	42.11
63	EPS	D	101.6	344.36	43.95

4. Ratings of Alternatives under subjective criteria

Criteria	Subjective Alternatives	DM1	DM2	DM3
C1	A	Fair (F)	Fair (F)	Good (G)
	B	Fair (F)	Good (G)	Fair (F)
	C	Good (G)	Fair (F)	Very good (VG)
	D	Good (G)	Fair (F)	Very good (VG)
C2	A	Good (G)	Poor (P)	Good (G)
	B	Very good (VG)	Good (G)	Very good (VG)
	C	Good (G)	Fair (F)	Very good (VG)
	D	Very good (VG)	Good (G)	Very good (VG)
C3	A	Fair (F)	Fair (F)	Good (G)
	B	Good (G)	Fair (F)	Very good (VG)
	C	Good (G)	Good (G)	Good (G)
	D	Good (G)	Good (G)	Good (G)
C4	A	Fair (F)	Fair (F)	Good (G)
	B	Fair (F)	Good (G)	Fair (F)
	C	Good (G)	Fair (F)	Very good (VG)
	D	Good (G)	Fair (F)	Very good (VG)

5. Ratings of criteria under objective criteria

ID	**Material**	**Material**	**Thickness**	**LCA**	**LCC**
Material A					
6	xps	A	76.2	408.16	29.15
13	xps	A	95.25	390.60	32.71
19	xps	A	101.6	386.20	33.27
24	xps	A	114.3	379.20	35.80
30	xps	A	127	374.24	38.66
31	xps	A	139.7	370.99	38.95
32	xps	A	165.1	368.64	50.44
33	xps	A	152.4	369.19	39.30
Material B					
39	Fiberglass	B	76.2	374.38	106.47
40	Fiberglass	B	101.6	339.64	133.53
Material C					
44	Polyiso	C	50.8	362.09	30.33
45	Polyiso	C	64.77	328.06	36.04
46	Polyiso	C	76.2	305.52	36.48
47	Polyiso	C	78.74	301.04	42.05
48	Polyiso	C	90.17	282.84	42.67
49	Polyiso	C	101.6	267.41	43.41
Material D					
57	EPS	D	69.85	397.39	38.22
60	EPS	D	76.2	385.39	39.38
61	EPS	D	76.2	385.39	40.38
62	EPS	D	88.9	363.61	42.11
63	EPS	D	101.6	344.36	43.95

6. Aggregated fuzzy ratings of the alternatives under subjective criteria by three DMs.

Criteria	Sub. Alt	DM1			DM2			DM3			Aggregated ratings		
C1	A	(3	5	7)	(3	5	7)	(7	9	10)	(4.33	6.33	8.00)
	B	(3	5	7)	(7	9	10)	(3	5	7)	(4.33	6.33	8.00)
	C	(7	9	10)	(3	5	7)	(9	10	10)	(6.33	8.00	9.00)
	D	(7	9	10)	(3	5	7)	(9	10	10)	(6.33	8.00	9.00)
C2	A	(7	9	10)	(0	1	3)	(7	9	10)	(4.67	6.33	7.67)
	B	(9	10	10)	(7	9	10)	(9	10	10)	(8.33	9.67	10.00)
	C	(7	9	10)	(3	5	7)	(9	10	10)	(6.33	8.00	9.00)
	D	(9	10	10)	(7	9	10)	(9	10	10)	(8.33	9.67	10.00)
C3	A	(3	5	7)	(3	5	7)	(7	9	10)	(4.33	6.33	8.00)
	B	(7	9	10)	(3	5	7)	(9	10	10)	(6.33	8.00	9.00)
	C	(7	9	10)	(7	9	10)	(7	9	10)	(7.00	9.00	10.00)
	D	(7	9	10)	(7	9	10)	(7	9	10)	(7.00	9.00	10.00)
C4	A	(3	5	7)	(3	5	7)	(7	9	10)	(4.33	6.33	8.00)
	B	(3	5	7)	(7	9	10)	(3	5	7)	(4.33	6.33	8.00)
	C	(7	9	10)	(3	5	7)	(9	10	10)	(6.33	8.00	9.00)
	D	(7	9	10)	(3	5	7)	(9	10	10)	(6.33	8.00	9.00)

7. Aggregation of the relative importance of each selected criteria by three DMs.

Criteria	DM1			DM2			DM3			Aggregated ratings			Defuzzified fuzzy weights
C1	(0.6	0.7	0.8)	(0.7	0.9	1)	(0.7	0.9	1)	(0.67	0.83	0.93)	0.82
C2	(0.7	0.9	1)	(0.6	0.7	0.8)	(0.7	0.9	1)	(0.67	0.83	0.93)	0.82
C3	(0.1	0.3	0.5)	(0	0.1	0.2)	(0.1	0.3	0.5)	(0.07	0.23	0.40)	0.23
C4	(0.1	0.3	0.5)	(0.1	0.3	0.5)	(0.1	0.3	0.5)	(0.10	0.30	0.50)	0.30
C5	(0.7	0.9	1)	(0.7	0.9	1)	(0.6	0.7	0.8)	(0.67	0.83	0.93)	0.82
C6	(0.1	0.3	0.5)	(0	0.1	0.2)	(0.1	0.3	0.5)	(0.07	0.23	0.40)	0.23
C7	(0.6	0.7	0.8)	(0.7	0.9	1)	(0.7	0.9	1)	(0.67	0.83	0.93)	0.82

8. Normalized decision matrix

Alternatives		C1			C2			C3			C4			C5	C6	C7
A	76.2	(0.25	0.37	0.47)	(0.25	0.34	0.42)	(0.23	0.34	0.43)	(0.25	0.37	0.47)	7.96	2.83	154.54
A	95.25	(0.25	0.37	0.47)	(0.25	0.34	0.42)	(0.23	0.34	0.43)	(0.25	0.37	0.47)	10.02	5.10	141.54
A	101.6	(0.25	0.37	0.47)	(0.25	0.34	0.42)	(0.23	0.34	0.43)	(0.25	0.37	0.47)	10.36	5.63	138.37
A	114.3	(0.25	0.37	0.47)	(0.25	0.34	0.42)	(0.23	0.34	0.43)	(0.25	0.37	0.47)	12.00	7.58	133.39
A	127	(0.25	0.37	0.47)	(0.25	0.34	0.42)	(0.23	0.34	0.43)	(0.25	0.37	0.47)	14.00	9.99	129.92
A	139.7	(0.25	0.37	0.47)	(0.25	0.34	0.42)	(0.23	0.34	0.43)	(0.25	0.37	0.47)	14.20	10.65	127.68
A	165.1	(0.25	0.37	0.47)	(0.25	0.34	0.42)	(0.23	0.34	0.43)	(0.25	0.37	0.47)	23.82	22.15	126.07
A	152.4	(0.25	0.37	0.47)	(0.25	0.34	0.42)	(0.23	0.34	0.43)	(0.25	0.37	0.47)	14.46	11.33	126.45
B	76.2	(0.25	0.37	0.47)	(0.45	0.52	0.54)	(0.34	0.43	0.48)	(0.25	0.37	0.47)	66.37	55.50	277.27
B	101.6	(0.25	0.37	0.47)	(0.45	0.52	0.54)	(0.34	0.43	0.48)	(0.25	0.37	0.47)	104.4	96.04	228.21
C	50.8	(0.37	0.47	0.53)	(0.34	0.43	0.49)	(0.38	0.48	0.54)	(0.37	0.47	0.53)	9.69	3.61	173.02
C	64.77	(0.37	0.47	0.53)	(0.34	0.43	0.49)	(0.38	0.48	0.54)	(0.37	0.47	0.53)	13.68	7.83	142.03
C	76.2	(0.37	0.47	0.53)	(0.34	0.43	0.49)	(0.38	0.48	0.54)	(0.37	0.47	0.53)	14.01	8.89	123.18
C	78.74	(0.37	0.47	0.53)	(0.34	0.43	0.49)	(0.38	0.48	0.54)	(0.37	0.47	0.53)	18.62	13.67	119.59
C	90.17	(0.37	0.47	0.53)	(0.34	0.43	0.49)	(0.38	0.48	0.54)	(0.37	0.47	0.53)	19.17	15.06	105.57
C	101.6	(0.37	0.47	0.53)	(0.34	0.43	0.49)	(0.38	0.48	0.54)	(0.37	0.47	0.53)	19.83	16.52	94.36

D	69.85	(0.37	0.47	0.53)	(0.45	0.52	0.54)	(0.38	0.48	0.54)	(0.37	0.47	0.53)	15.99	8.20	187.98
D	76.2	(0.37	0.47	0.53)	(0.45	0.52	0.54)	(0.38	0.48	0.54)	(0.37	0.47	0.53)	16.97	9.54	176.80
D	76.2	(0.37	0.47	0.53)	(0.45	0.52	0.54)	(0.38	0.48	0.54)	(0.37	0.47	0.53)	17.85	10.38	176.80
D	88.9	(0.37	0.47	0.53)	(0.45	0.52	0.54)	(0.38	0.48	0.54)	(0.37	0.47	0.53)	19.41	12.82	157.38
D	101.6	(0.37	0.47	0.53)	(0.45	0.52	0.54)	(0.38	0.48	0.54)	(0.37	0.47	0.53)	21.14	15.51	141.16

9. Weighted normalized decision matrix

Alt.		C1			C2			C3			C4			C5			C6			C7		
A	76.2	0.17	0.31	0.44	0.17	0.29	0.39	0.02	0.08	0.17	0.03	0.11	0.23	5.30	6.63	7.43	0.19	0.66	1.13	103.0	128.7	144.2
A	95.25	0.17	0.31	0.44	0.17	0.29	0.39	0.02	0.08	0.17	0.03	0.11	0.23	6.68	8.35	9.35	0.34	1.19	2.04	94.36	117.9	132.1
A	101.6	0.17	0.31	0.44	0.17	0.29	0.39	0.02	0.08	0.17	0.03	0.11	0.23	6.91	8.63	9.67	0.38	1.31	2.25	92.24	115.3	129.1
A	114.3	0.17	0.31	0.44	0.17	0.29	0.39	0.02	0.08	0.17	0.03	0.11	0.23	8.00	10.0	11.2	0.51	1.77	3.03	88.93	111.1	124.5
A	127	0.17	0.31	0.44	0.17	0.29	0.39	0.02	0.08	0.17	0.03	0.11	0.23	9.33	11.6	13.0	0.67	2.33	4.00	86.62	108.2	121.2
A	139.7	0.17	0.31	0.44	0.17	0.29	0.39	0.02	0.08	0.17	0.03	0.11	0.23	9.47	11.8	13.2	0.71	2.48	4.26	85.12	106.4	119.1
A	165.1	0.17	0.31	0.44	0.17	0.29	0.39	0.02	0.08	0.17	0.03	0.11	0.23	15.8	19.8	22.2	1.48	5.17	8.86	84.05	105.0	117.6
A	152.4	0.17	0.31	0.44	0.17	0.29	0.39	0.02	0.08	0.17	0.03	0.11	0.23	9.64	12.0	13.4	0.76	2.64	4.53	84.30	105.3	118.0
B	76.2	0.17	0.31	0.44	0.30	0.44	0.51	0.02	0.10	0.19	0.03	0.11	0.23	44.2	55.3	61.9	3.70	12.9	22.2	184.8	231.0	258.7
B	101.6	0.17	0.31	0.44	0.30	0.44	0.51	0.02	0.10	0.19	0.03	0.11	0.23	69.6	87.0	97.4	6.40	22.4	38.4	152.1	190.1	213.0
C	50.8	0.25	0.39	0.49	0.23	0.36	0.46	0.03	0.11	0.22	0.04	0.14	0.26	6.46	8.07	9.04	0.24	0.84	1.44	115.3	144.1	161.4

C	64.77	0.25	0.39	0.49	0.23	0.36	0.46	0.03	0.11	0.22	0.04	0.14	0.26	9.12	11.4	12.7	0.52	1.83	3.13	94.69	118.3	132.5
C	76.2	0.25	0.39	0.49	0.23	0.36	0.46	0.03	0.11	0.22	0.04	0.14	0.26	9.34	11.6	13.0	0.59	2.07	3.56	82.12	102.6	114.9
C	78.74	0.25	0.39	0.49	0.23	0.36	0.46	0.03	0.11	0.22	0.04	0.14	0.26	12.4	15.5	17.3	0.91	3.19	5.47	79.73	99.66	111.6
C	90.17	0.25	0.39	0.49	0.23	0.36	0.46	0.03	0.11	0.22	0.04	0.14	0.26	12.7	15.9	17.8	1.00	3.51	6.02	70.38	87.98	98.53
C	101.6	0.25	0.39	0.49	0.23	0.36	0.46	0.03	0.11	0.22	0.04	0.14	0.26	13.2	16.5	18.5	1.10	3.86	6.61	62.91	78.64	88.07
D	69.85	0.25	0.39	0.49	0.30	0.44	0.51	0.03	0.11	0.22	0.04	0.14	0.26	10.6	13.3	14.9	0.55	1.91	3.28	125.3	156.6	175.4
D	76.2	0.25	0.39	0.49	0.30	0.44	0.51	0.03	0.11	0.22	0.04	0.14	0.26	11.3	14.1	15.8	0.64	2.23	3.81	117.8	147.3	165.0
D	76.2	0.25	0.39	0.49	0.30	0.44	0.51	0.03	0.11	0.22	0.04	0.14	0.26	11.9	14.8	16.6	0.69	2.42	4.15	117.8	147.3	165.0
D	88.9	0.25	0.39	0.49	0.30	0.44	0.51	0.03	0.11	0.22	0.04	0.14	0.26	12.9	16.1	18.1	0.85	2.99	5.13	104.9	131.1	146.8
D	101.6	0.25	0.39	0.49	0.30	0.44	0.51	0.03	0.11	0.22	0.04	0.14	0.26	14.1	17.6	19.7	1.03	3.62	6.21	94.10	117.6	131.7

10.Defuzzified weighted normalized decision matrix

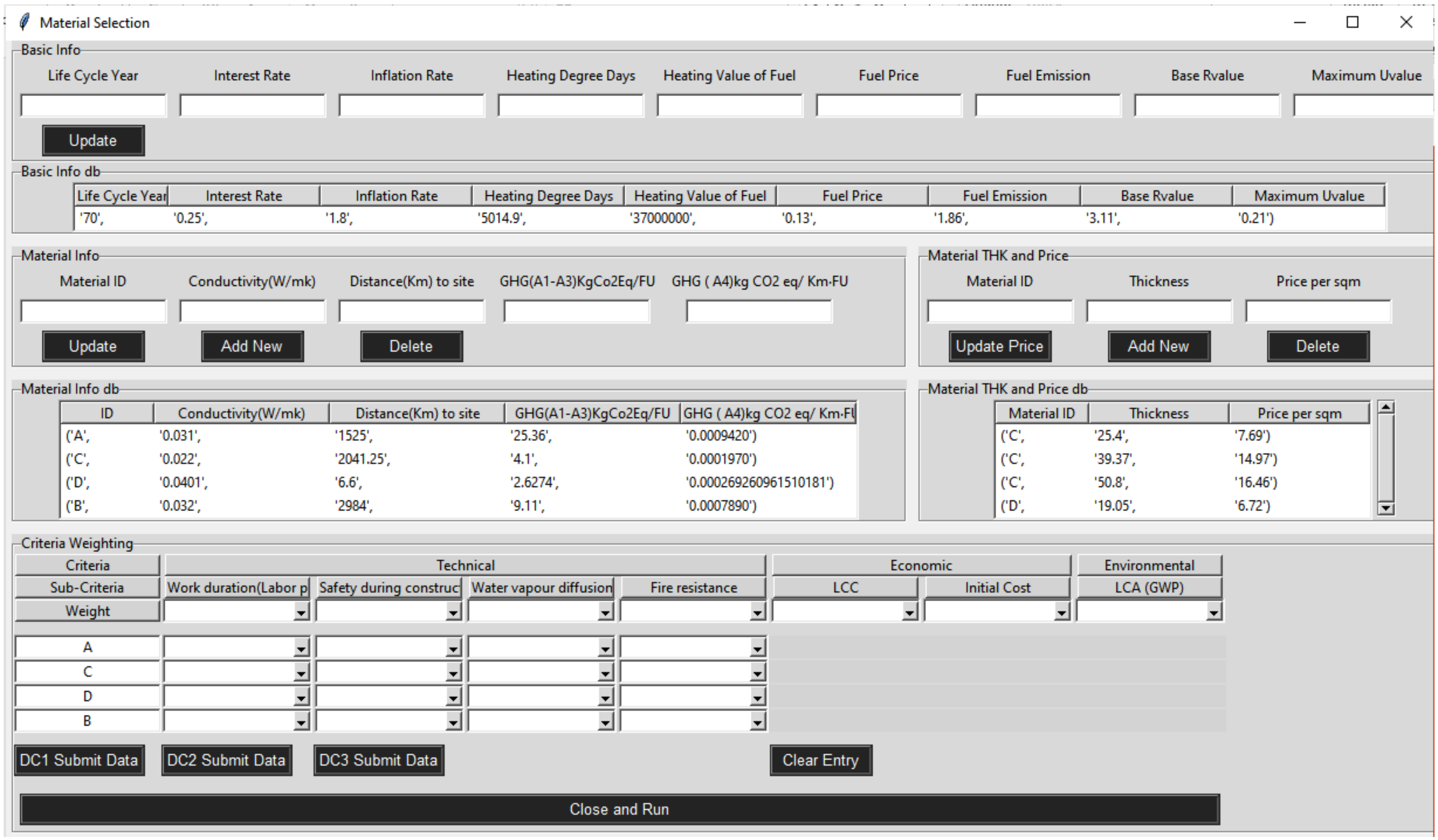
Alternatives		C1	C2	C3	C4	C5	C6	C7
A	76.2	0.306986	0.282403	0.086739	0.120869	6.498338	0.660868	126.210
A	95.25	0.306986	0.282403	0.086739	0.120869	8.179110	1.191027	115.587
A	101.6	0.306986	0.282403	0.086739	0.120869	8.461441	1.313107	112.999
A	114.3	0.306986	0.282403	0.086739	0.120869	9.800383	1.767983	108.937
A	127	0.306986	0.282403	0.086739	0.120869	11.430112	2.331010	106.105
A	139.7	0.306986	0.282403	0.086739	0.120869	11.599375	2.484786	104.272
A	165.1	0.306986	0.282403	0.086739	0.120869	19.449942	5.168542	102.956
A	152.4	0.306986	0.282403	0.086739	0.120869	11.806953	2.643474	103.265
B	76.2	0.306986	0.420440	0.104386	0.120869	54.203843	12.950289	226.439
B	101.6	0.306986	0.420440	0.104386	0.120869	85.267015	22.409323	186.372
C	50.8	0.381041	0.352024	0.116649	0.145826	7.910241	0.841577	141.303

C	64.77	0.381041	0.352024	0.116649	0.145826	11.169702	1.826842	115.991
C	76.2	0.381041	0.352024	0.116649	0.145826	11.439155	2.074987	100.599
C	78.74	0.381041	0.352024	0.116649	0.145826	15.204662	3.189221	97.667
C	90.17	0.381041	0.352024	0.116649	0.145826	15.654038	3.514551	86.217
C	101.6	0.381041	0.352024	0.116649	0.145826	16.197451	3.855678	77.064
D	69.85	0.381041	0.420440	0.116649	0.145826	13.058241	1.914218	153.514
D	76.2	0.381041	0.420440	0.116649	0.145826	13.858669	2.225098	144.384
D	76.2	0.381041	0.420440	0.116649	0.145826	14.578106	2.422850	144.384
D	88.9	0.381041	0.420440	0.116649	0.145826	15.853419	2.991339	128.523
D	101.6	0.381041	0.420440	0.116649	0.145826	17.266891	3.619685	115.278
	Positive ideal solutions (R*)	0.381041	0.420440	0.116649	0.145826	6.498338	0.660868	77.064033
	Negative ideal solutions (R-)	0.306986	0.282403	0.086739	0.120869	85.267015	22.409323	226.439428

11. Euclidean Distance of each alternative from positive ideal solution and negative ideal solution D*+D-, relative closeness of each alternative to the ideal solution Ci and Final Ranking

Alt.		D*+D-	Ci	Final
A	76.2	178.47	0.72	14
A	95.25	175.24	0.78	11
A	101.6	174.61	0.79	10
A	114.3	173.23	0.81	9
A	127	172.11	0.83	7
A	139.7	171.79	0.84	6
A	165.1	170.29	0.83	8
A	152.4	171.58	0.84	5
B	76.2	189.76	0.17	21
B	101.6	176.54	0.23	20
C	50.8	181.29	0.65	16
C	64.77	173.81	0.77	12
C	76.2	171.40	0.86	4
C	78.74	170.36	0.87	3
C	90.17	170.94	0.92	2
C	101.6	175.83	0.94	1
D	69.85	181.39	0.58	19
D	76.2	178.37	0.62	17
D	76.2	177.96	0.62	18
D	88.9	173.94	0.70	15
D	101.6	171.47	0.77	13

Appendix B: Python Interface -Input



Basic Info db

Life Cycle Year	Interest Rate	Inflation Rate	Heating Degree Days	Heating Value of Fuel	Fuel Price	Fuel Emission	Base Rvalue	Maximum Uvalue
'70',	'0.25',	'1.8',	'5014.9',	'37000000',	'0.13',	'1.86',	'3.11',	'0.21')

Material Info db

ID	Conductivity(W/mk)	Distance(Km) to site	GHG(A1-A3)KgCo2Eq/FU	GHG (A4)kg CO2 eq/ Km-Fl
('A',	'0.031',	'1525',	'25.36',	'0.0009420')
('C',	'0.022',	'2041.25',	'4.1',	'0.0001970')
('D',	'0.0401',	'6.6',	'2.6274',	'0.000269260961510181')
('B',	'0.032',	'2984',	'9.11',	'0.0007890')

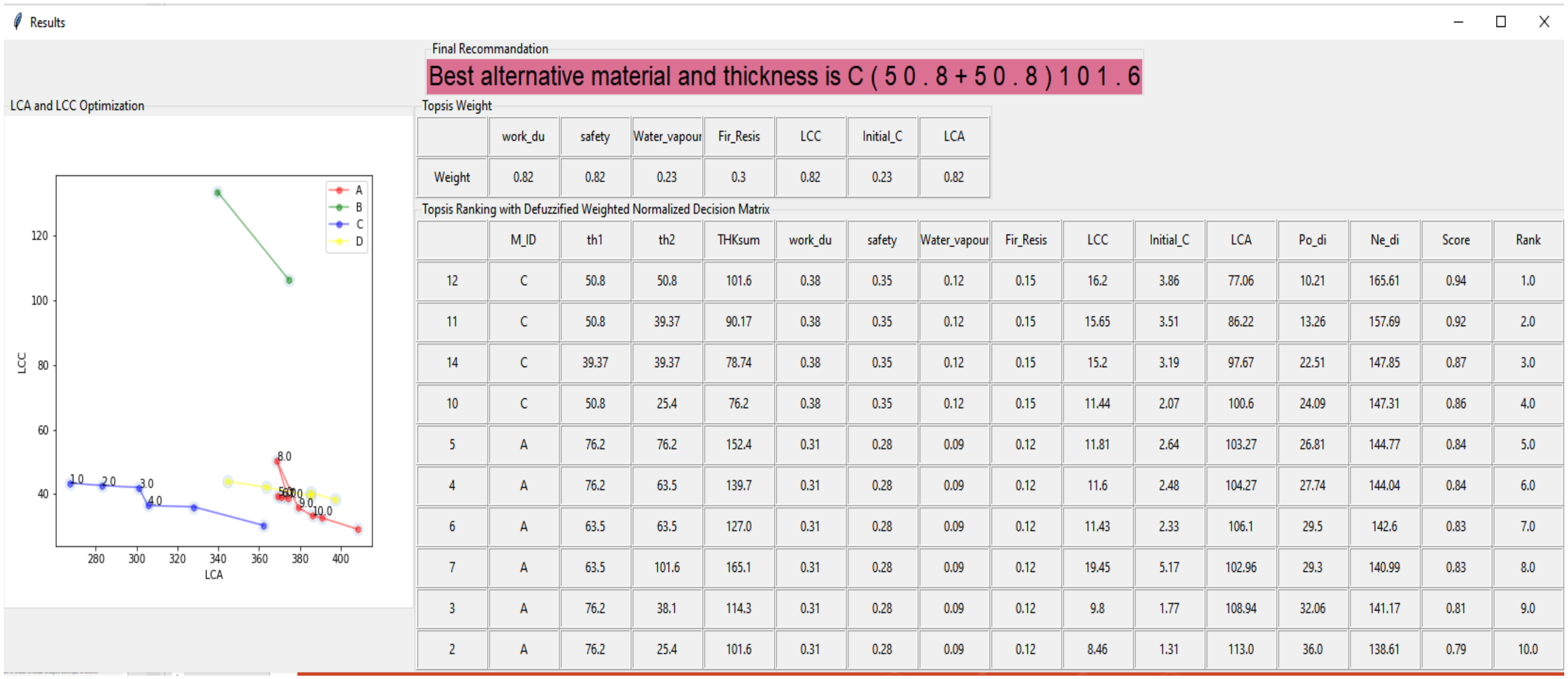
Material THK and Price db

Material ID	Thickness	Price per sqm
('C',	'25.4',	'7.69')
('C',	'39.37',	'14.97')
('C',	'50.8',	'16.46')
('D',	'19.05',	'6.72')

Criteria Weighting

Criteria	Technical				Economic		Environmental
Sub-Criteria	Work duration(Labor p	Safety during construc	Water vapour diffusion	Fire resistance	LCC	Initial Cost	LCA (GWP)
Weight							
A							
C							
D							
B							

-Result



Final Recommandation

Best alternative material and thickness is C (5 0 . 8 + 5 0 . 8) 1 0 1 . 6

Topsis Weight

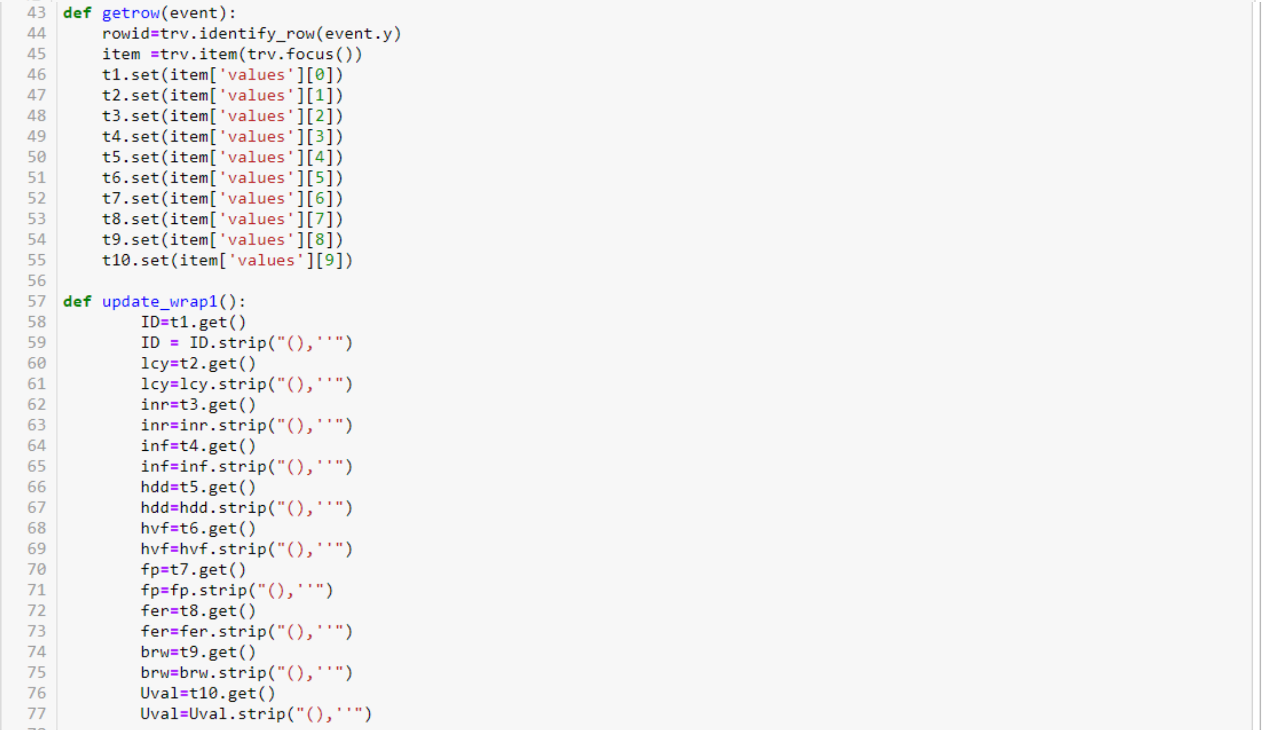
	work_du	safety	Water_vapour	Fir_Resis	LCC	Initial_C	LCA
Weight	0.82	0.82	0.23	0.3	0.82	0.23	0.82

Topsis Ranking with Defuzzified Weighted Normalized Decision Matrix

	M_ID	th1	th2	THKsum	work_du	safety	Water_vapour	Fir_Resis	LCC	Initial_C	LCA	Po_di	Ne_di	Score	Rank
12	C	50.8	50.8	101.6	0.38	0.35	0.12	0.15	16.2	3.86	77.06	10.21	165.61	0.94	1.0
11	C	50.8	39.37	90.17	0.38	0.35	0.12	0.15	15.65	3.51	86.22	13.26	157.69	0.92	2.0
14	C	39.37	39.37	78.74	0.38	0.35	0.12	0.15	15.2	3.19	97.67	22.51	147.85	0.87	3.0
10	C	50.8	25.4	76.2	0.38	0.35	0.12	0.15	11.44	2.07	100.6	24.09	147.31	0.86	4.0
5	A	76.2	76.2	152.4	0.31	0.28	0.09	0.12	11.81	2.64	103.27	26.81	144.77	0.84	5.0
4	A	76.2	63.5	139.7	0.31	0.28	0.09	0.12	11.6	2.48	104.27	27.74	144.04	0.84	6.0
6	A	63.5	63.5	127.0	0.31	0.28	0.09	0.12	11.43	2.33	106.1	29.5	142.6	0.83	7.0
7	A	63.5	101.6	165.1	0.31	0.28	0.09	0.12	19.45	5.17	102.96	29.3	140.99	0.83	8.0
3	A	76.2	38.1	114.3	0.31	0.28	0.09	0.12	9.8	1.77	108.94	32.06	141.17	0.81	9.0
2	A	76.2	25.4	101.6	0.31	0.28	0.09	0.12	8.46	1.31	113.0	36.0	138.61	0.79	10.0

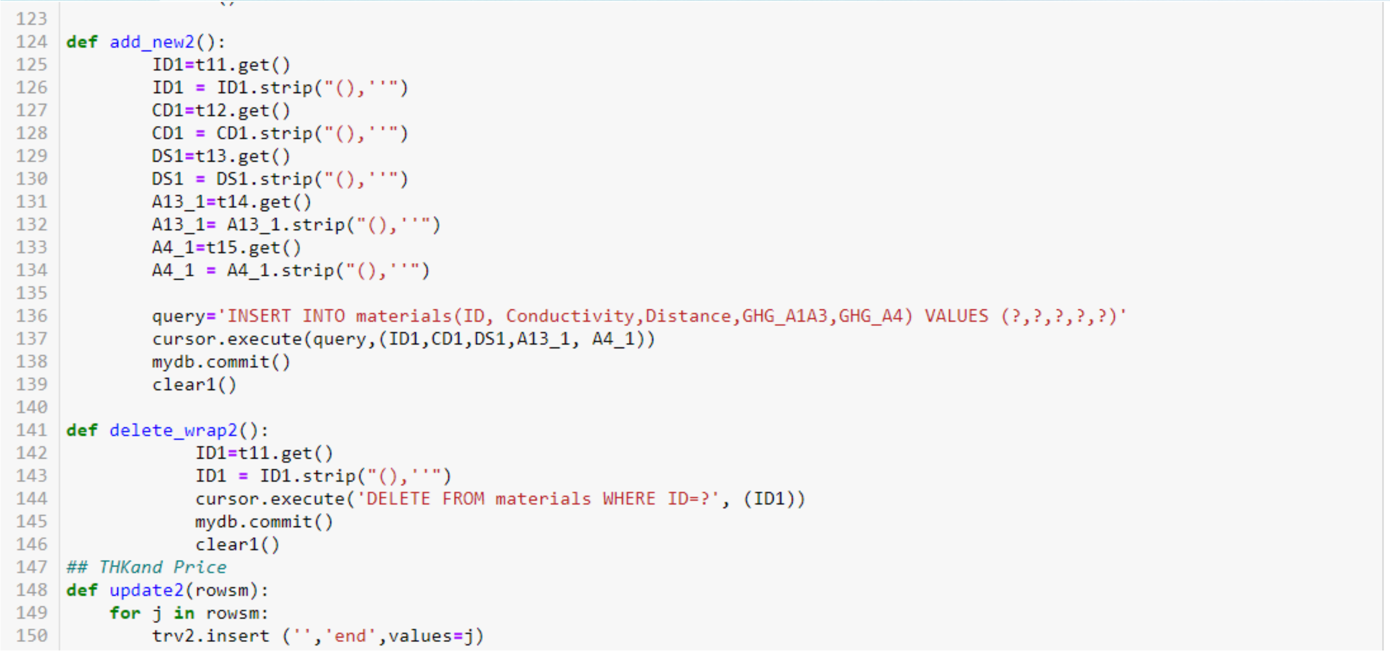
Appendix C: Python Code

```python
from tkinter import*
import tkinter as tk
from tkinter import ttk
from tkinter.ttk import *
from tkinter import messagebox
from tkinter import font  as tkfont
import pyodbc
import pandas as pd
from pandas import DataFrame
from pandastable import Table, TableModel
from numpy import *
import numpy as np
from paretoset import paretoset
import matplotlib.pyplot as plt
import matplotlib
import timeit
from matplotlib.backends.backend_tkagg import ( FigureCanvasTkAgg, NavigationToolbar2Tk)
import matplotlib
matplotlib.use('TkAgg')
from matplotlib.figure import Figure
from topsis import topsis
import topsispy as tp
from tkinter import messagebox
import csv


LARGE_FONT=("Verdana",12)
NORM_FONT=("Verdana",10)
SMALL_FONT=("Verdana",8)

##Basics
def update(rws):
    for i in rws:
        trv.insert ('','end',values=i)
def clear():
    for i in trv.get_children():
        trv.delete(i)
    cursor.execute("SELECT * FROM Basics;")
    rws=cursor.fetchall()
    #rws=rws.strip("(),'")
    update(rws)

```



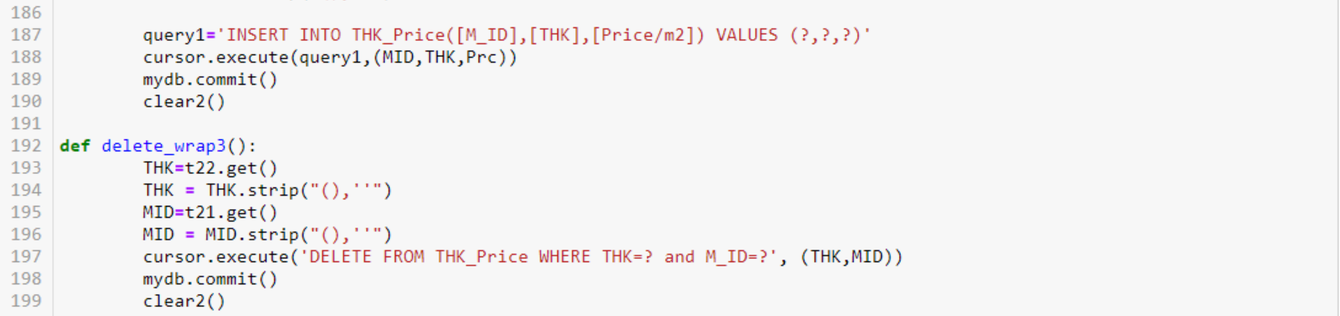
```python
def getrow(event):
    rowid=trv.identify_row(event.y)
    item =trv.item(trv.focus())
    t1.set(item['values'][0])
    t2.set(item['values'][1])
    t3.set(item['values'][2])
    t4.set(item['values'][3])
    t5.set(item['values'][4])
    t6.set(item['values'][5])
    t7.set(item['values'][6])
    t8.set(item['values'][7])
    t9.set(item['values'][8])
    t10.set(item['values'][9])

def update_wrap1():
        ID=t1.get()
        ID = ID.strip("(),'")
        lcy=t2.get()
        lcy=lcy.strip("(),'")
        inr=t3.get()
        inr=inr.strip("(),'")
        inf=t4.get()
        inf=inf.strip("(),'")
        hdd=t5.get()
        hdd=hdd.strip("(),'")
        hvf=t6.get()
        hvf=hvf.strip("(),'")
        fp=t7.get()
        fp=fp.strip("(),'")
        fer=t8.get()
        fer=fer.strip("(),'")
        brw=t9.get()
        brw=brw.strip("(),'")
        Uval=t10.get()
        Uval=Uval.strip("(),'")
```

```python

        query='UPDATE Basics SET LCY=?, InR=?, Inf=?, HDD=?, HVF=?,FP=?,FER=?,BRW=?,Uval=? WHERE ID=?'
        cursor.execute(query,( lcy, inr, inf, hdd, hvf, fp, fer, brw,Uval,ID))
        mydb.commit()
        clear()


## material

def update1(rows):
    for j in rows:
        trv1.insert ('','end',values=j)

def clear1():
    for i in trv1.get_children():
        trv1.delete(i)
    cursor.execute("SELECT * FROM materials;")
    rows=cursor.fetchall()
    update1(rows)

def getrow1(event):
    rowid=trv1.identify_row(event.y)
    item1 =trv1.item(trv1.focus())
    t11.set(item1['values'][0])
    t12.set(item1['values'][1])
    t13.set(item1['values'][2])
    t14.set(item1['values'][3])
    t15.set(item1['values'][4])

def update_wrap2():
        ID1=t11.get()
        ID1 = ID1.strip("(),''")
        CD1=t12.get()
        CD1 = CD1.strip("(),''")
        DS1=t13.get()
        DS1 = DS1.strip("(),''")
        A13_1=t14.get()
        A13_1=A13_1.strip("(),''")
        A4_1=t15.get()
        A4_1 = A4_1.strip("(),''")

        query='UPDATE materials SET Conductivity=?,Distance=?,GHG_A1A3=?,GHG_A4=? WHERE ID=?'
        cursor.execute(query,(CD1,DS1,A13_1, A4_1,ID1))
        mydb.commit()
        clear1()
```



```python

def add_new2():
        ID1=t11.get()
        ID1 = ID1.strip("(),''")
        CD1=t12.get()
        CD1 = CD1.strip("(),''")
        DS1=t13.get()
        DS1 = DS1.strip("(),''")
        A13_1=t14.get()
        A13_1= A13_1.strip("(),''")
        A4_1=t15.get()
        A4_1 = A4_1.strip("(),''")

        query='INSERT INTO materials(ID, Conductivity,Distance,GHG_A1A3,GHG_A4) VALUES (?,?,?,?,?)'
        cursor.execute(query,(ID1,CD1,DS1,A13_1, A4_1))
        mydb.commit()
        clear1()

def delete_wrap2():
            ID1=t11.get()
            ID1 = ID1.strip("(),''")
            cursor.execute('DELETE FROM materials WHERE ID=?', (ID1))
            mydb.commit()
            clear1()
## THKand Price
def update2(rowsm):
    for j in rowsm:
        trv2.insert ('','end',values=j)
```

```

def clear2():
    for i in trv2.get_children():
        trv2.delete(i)
    cursor.execute("SELECT * FROM THK_Price;")
    rowsm=cursor.fetchall()
    update2(rowsm)

def getrow2(event):
    rowid=trv2.identify_row(event.y)
    item2 =trv2.item(trv2.focus())
    t21.set(item2['values'][0])
    t22.set(item2['values'][1])
    t23.set(item2['values'][2])

def update_wrap3():
        MID=t21.get()
        MID = MID.strip("(),''")
        THK=t22.get()
        THK = THK.strip("(),''")
        Prc=t23.get()
        Prc = Prc.strip("(),''")

        query1='UPDATE THK_Price SET [Price/m2]=? WHERE [M_ID]=? AND [THK]=?'
        cursor.execute(query1,(Prc,MID,THK))
        mydb.commit()
        clear2()

def add_new3():
        MID=t21.get()
        MID = MID.strip("(),''")
        THK=t22.get()
        THK = THK.strip("(),''")
        Prc=t23.get()
        Prc = Prc.strip("(),''")
```

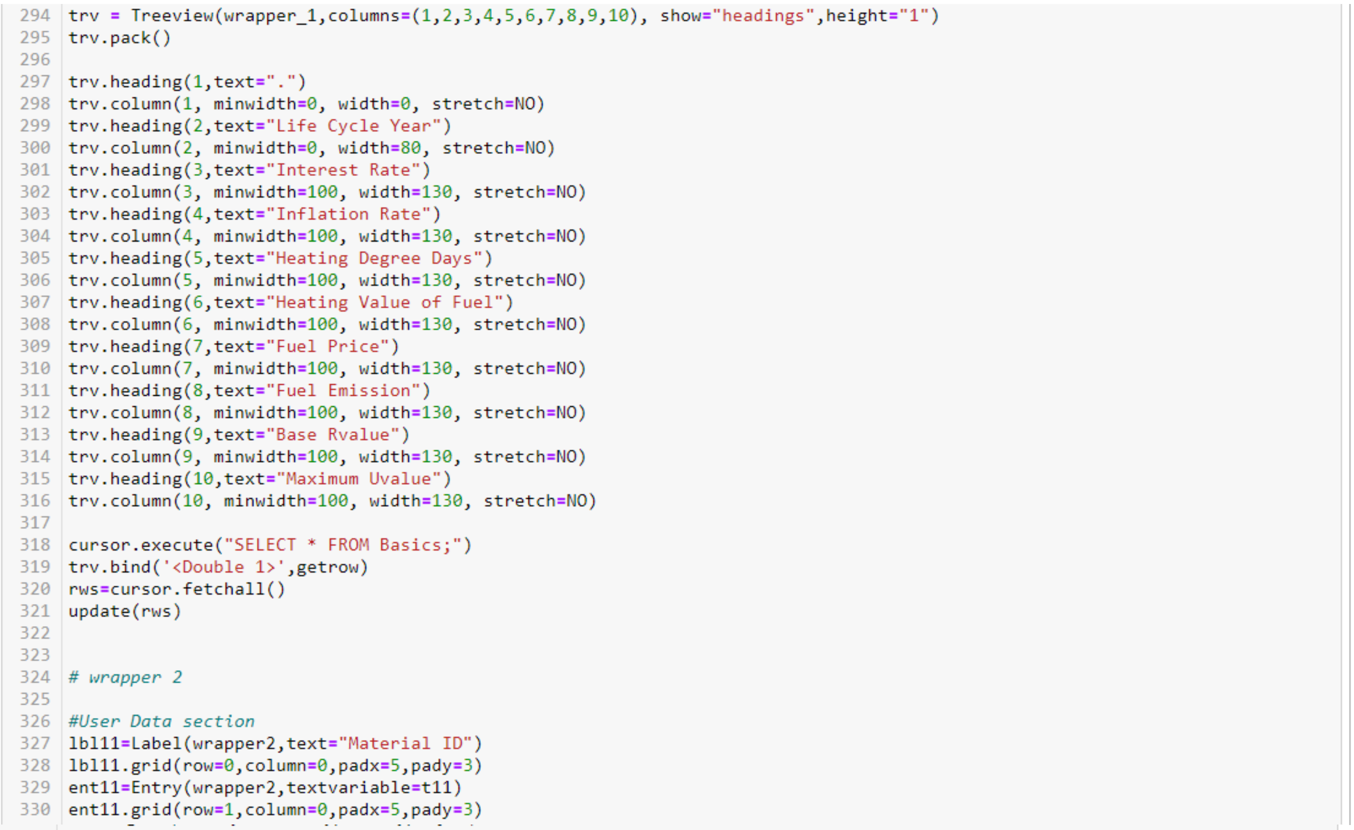


```

        query1='INSERT INTO THK_Price([M_ID],[THK],[Price/m2]) VALUES (?,?,?)'
        cursor.execute(query1,(MID,THK,Prc))
        mydb.commit()
        clear2()

def delete_wrap3():
        THK=t22.get()
        THK = THK.strip("(),''")
        MID=t21.get()
        MID = MID.strip("(),''")
        cursor.execute('DELETE FROM THK_Price WHERE THK=? and M_ID=?', (THK,MID))
        mydb.commit()
        clear2()
```

```


mydb = pyodbc.connect(r'Driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=C:\Users\sungyikim\Desktop\Writings\testdb.ac
cursor=mydb.cursor()

root=Tk()

t1=StringVar()
t2=StringVar()
t3=StringVar()
t4=StringVar()
t5=StringVar()
t6=StringVar()
t7=StringVar()
t8=StringVar()
t9=StringVar()
t10=StringVar()
t11=StringVar()
t12=StringVar()
t13=StringVar()
t14=StringVar()
t15=StringVar()
t21=StringVar()
```

```
t22=StringVar()
t23=StringVar()

wrapper1 = LabelFrame(root,text="Basic Info")
wrapper_1 = LabelFrame(root,text="Basic Info db")
wrapper2 = LabelFrame(root, text="Material Info")
wrapper_2 = LabelFrame(root,text="Material Info db")
wrapper2_1 = LabelFrame(root, text="Material THK and Price")
wrapper_2_1 = LabelFrame(root,text="Material THK and Price db")
wrapper3 = LabelFrame(root, text="Criteria Weighting")

wrapper1.grid(row=0, columnspan=2,sticky='news',padx=5)
wrapper_1.grid(row=1, columnspan=2,sticky='news',padx=5)
wrapper2.grid(row=2, column=0,sticky='news',pady=10,padx=5)
wrapper_2.grid(row=3, column=0,sticky='news',padx=5)
wrapper2_1.grid(row=2, column=1,sticky='news',pady=10,padx=5)
wrapper_2_1.grid(row=3, column=1,sticky='news',padx=5)
wrapper3.grid(row=4, columnspan=2,sticky='news',pady=10,padx=5)


#wrapper1
lbl1=Label(wrapper1,text="Life Cycle Year")
lbl1.grid(row=0,column=0,padx=5,pady=3)
ent1=Entry(wrapper1,textvariable=t2)
ent1.grid(row=1,column=0,padx=5,pady=3)

lbl2=Label(wrapper1,text="Interest Rate")
lbl2.grid(row=0,column=1,padx=5,pady=3)
ent2=Entry(wrapper1,textvariable=t3)
ent2.grid(row=1,column=1,padx=5,pady=3)

lbl3=Label(wrapper1,text="Inflation Rate")
lbl3.grid(row=0,column=2,padx=5,pady=3)
ent3=Entry(wrapper1,textvariable=t4)
ent3.grid(row=1,column=2,padx=5,pady=3)
```

```

lbl5=Label(wrapper1,text="Heating Degree Days")
lbl5.grid(row=0,column=3,padx=5,pady=3)
ent5=Entry(wrapper1,textvariable=t5)
ent5.grid(row=1,column=3,padx=5,pady=3)

lbl6=Label(wrapper1,text="Heating Value of Fuel")
lbl6.grid(row=0,column=4,padx=5,pady=3)
ent6=Entry(wrapper1,textvariable=t6)
ent6.grid(row=1,column=4,padx=5,pady=3)

lbl7=Label(wrapper1,text="Fuel Price")
lbl7.grid(row=0,column=5,padx=5,pady=3)
ent7=Entry(wrapper1,textvariable=t7)
ent7.grid(row=1,column=5,padx=5,pady=3)

lbl8=Label(wrapper1,text="Fuel Emission")
lbl8.grid(row=0,column=6,padx=5,pady=3)
ent8=Entry(wrapper1,textvariable=t8)
ent8.grid(row=1,column=6,padx=5,pady=3)

lbl9=Label(wrapper1,text="Base Rvalue")
lbl9.grid(row=0,column=7,padx=5,pady=3)
ent9=Entry(wrapper1,textvariable=t9)
ent9.grid(row=1,column=7,padx=5,pady=3)

lbl10=Label(wrapper1,text="Maximum Uvalue")
lbl10.grid(row=0,column=8,padx=5,pady=3)
ent10=Entry(wrapper1,textvariable=t10)
ent10.grid(row=1,column=8,padx=5,pady=3)


up_btn=Button(wrapper1,text="Update",command=update_wrap1)
up_btn.grid(row=4,column=0,padx=5,pady=3)


```



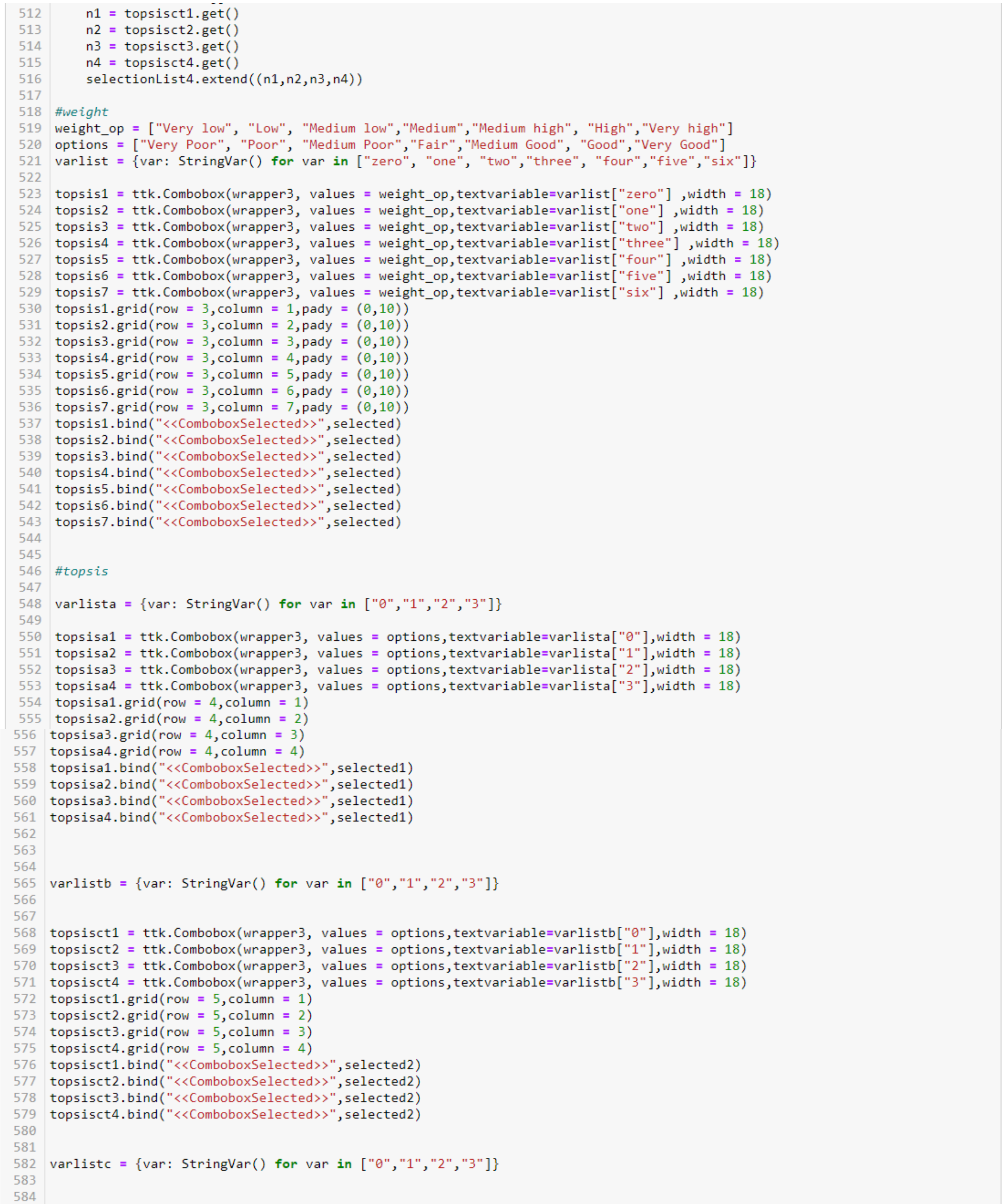
```
trv = Treeview(wrapper_1,columns=(1,2,3,4,5,6,7,8,9,10), show="headings",height="1")
trv.pack()

trv.heading(1,text=".")
trv.column(1, minwidth=0, width=0, stretch=NO)
trv.heading(2,text="Life Cycle Year")
trv.column(2, minwidth=0, width=80, stretch=NO)
trv.heading(3,text="Interest Rate")
trv.column(3, minwidth=100, width=130, stretch=NO)
trv.heading(4,text="Inflation Rate")
trv.column(4, minwidth=100, width=130, stretch=NO)
trv.heading(5,text="Heating Degree Days")
trv.column(5, minwidth=100, width=130, stretch=NO)
trv.heading(6,text="Heating Value of Fuel")
trv.column(6, minwidth=100, width=130, stretch=NO)
trv.heading(7,text="Fuel Price")
trv.column(7, minwidth=100, width=130, stretch=NO)
trv.heading(8,text="Fuel Emission")
trv.column(8, minwidth=100, width=130, stretch=NO)
trv.heading(9,text="Base Rvalue")
trv.column(9, minwidth=100, width=130, stretch=NO)
trv.heading(10,text="Maximum Uvalue")
trv.column(10, minwidth=100, width=130, stretch=NO)

cursor.execute("SELECT * FROM Basics;")
trv.bind('<Double 1>',getrow)
rws=cursor.fetchall()
update(rws)


# wrapper 2

#User Data section
lbl11=Label(wrapper2,text="Material ID")
lbl11.grid(row=0,column=0,padx=5,pady=3)
ent11=Entry(wrapper2,textvariable=t11)
ent11.grid(row=1,column=0,padx=5,pady=3)
```

```

lbl12=Label(wrapper2,text="Conductivity(W/mk)")
lbl12.grid(row=0,column=1,padx=5,pady=3)
ent12=Entry(wrapper2,textvariable=t12)
ent12.grid(row=1,column=1,padx=5,pady=3)

lbl13=Label(wrapper2,text="Distance(Km) to site")
lbl13.grid(row=0,column=2,padx=5,pady=3)
ent13=Entry(wrapper2,textvariable=t13)
ent13.grid(row=1,column=2,padx=5,pady=3)

lbl15=Label(wrapper2,text="GHG(A1-A3)KgCo2Eq/FU")
lbl15.grid(row=0,column=3,padx=5,pady=3)
ent15=Entry(wrapper2,textvariable=t14)
ent15.grid(row=1,column=3,padx=5,pady=3)

lbl16=Label(wrapper2,text="GHG ( A4)kg CO2 eq/ Km·FU")
lbl16.grid(row=0,column=4,padx=5,pady=3)
ent16=Entry(wrapper2,textvariable=t15)
ent16.grid(row=1,column=4,padx=5,pady=3)

up_btn1=Button(wrapper2,text="Update",command=update_wrap2)
add_btn1=Button(wrapper2,text="Add New",command=add_new2)
delete_btn1=Button(wrapper2,text="Delete",command=delete_wrap2)

up_btn1.grid(row=4,column=0,padx=5,pady=3)
add_btn1.grid(row=4,column=1,padx=5,pady=3)
delete_btn1.grid(row=4,column=2,padx=5,pady=3)

trv1 = Treeview(wrapper_2,columns=(1,2,3,4,5), show="headings",height="4")
trv1.pack()

trv1.heading(1,text="ID")
trv1.column(1, width=80, stretch=Y)
trv1.heading(2,text="Conductivity(W/mk)")
trv1.column(2, width=150, stretch=Y)
```

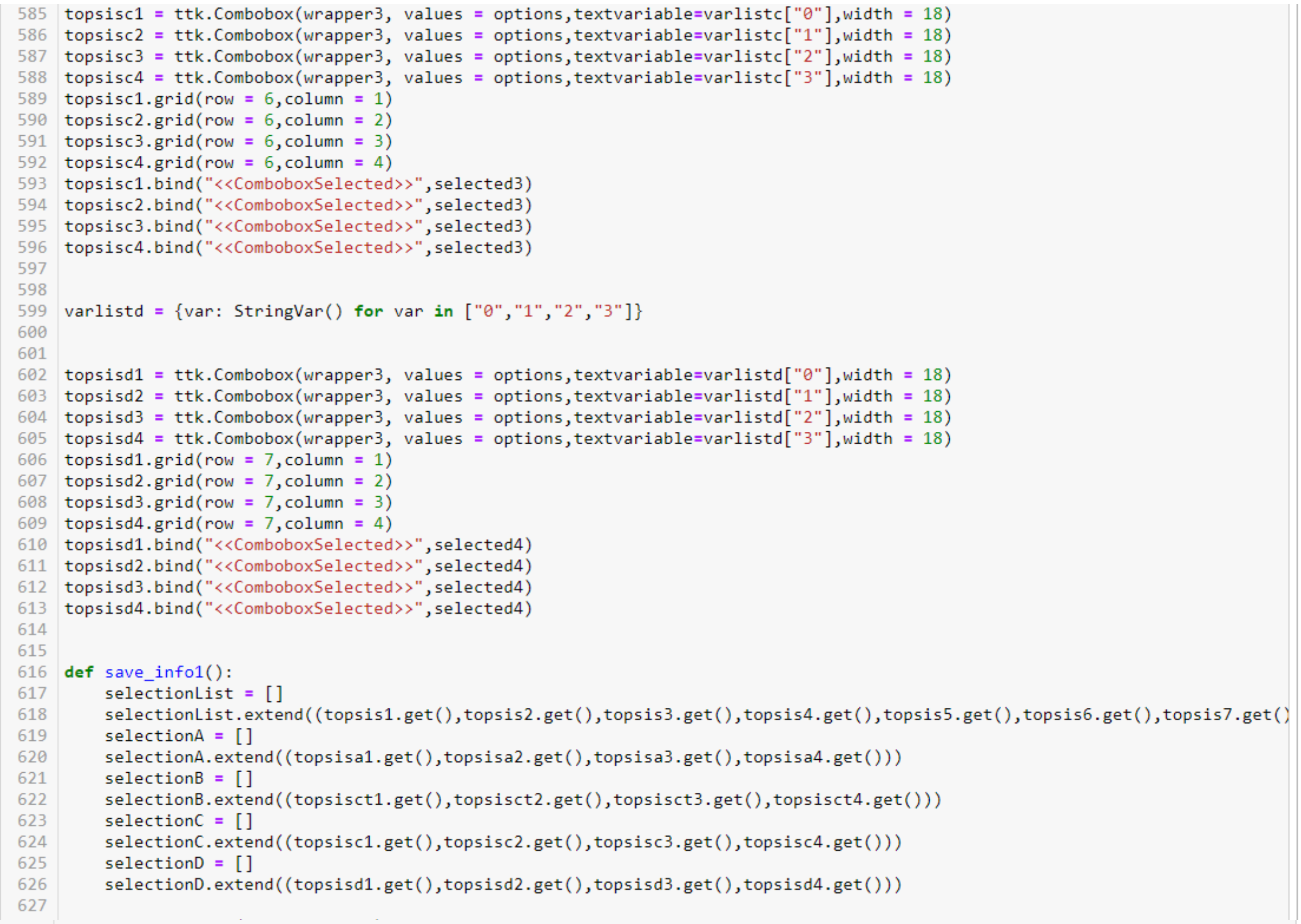
```
trv1.heading(3,text="Distance(Km) to site")
trv1.column(3, width=150, stretch=Y)
trv1.heading(4,text="GHG(A1-A3)KgCo2Eq/FU")
trv1.column(4, width=150, stretch=NO)
trv1.heading(5,text="GHG ( A4)kg CO2 eq/ Km·FU")
trv1.column(5, width=150, stretch=Y)

trv1.bind('<Double 1>',getrow1)

cursor.execute("SELECT * FROM materials;")
rows=cursor.fetchall()
update1(rows)

# wrapper 2_1

#User Data section
lbl21=Label(wrapper2_1,text="Material ID")
lbl21.grid(row=0,column=0,padx=5,pady=3)
ent21=Entry(wrapper2_1,textvariable=t21)
ent21.grid(row=1,column=0,padx=5,pady=3)

lbl22=Label(wrapper2_1,text="Thickness")
lbl22.grid(row=0,column=1,padx=5,pady=3)
ent22=Entry(wrapper2_1,textvariable=t22)
ent22.grid(row=1,column=1,padx=5,pady=3)

lbl23=Label(wrapper2_1,text="Price per sqm")
lbl23.grid(row=0,column=2,padx=5,pady=3)
ent23=Entry(wrapper2_1,textvariable=t23)
ent23.grid(row=1,column=2,padx=5,pady=3)

up_btn3=Button(wrapper2_1,text="Update Price",command=update_wrap3)
add_btn3=Button(wrapper2_1,text="Add New",command=add_new3)
delete_btn3=Button(wrapper2_1,text="Delete",command=delete_wrap3)

up_btn3.grid(row=4,column=0,padx=5,pady=3)
add_btn3.grid(row=4,column=1,padx=5,pady=3)
delete_btn3.grid(row=4,column=2,padx=5,pady=3)

trv2 = Treeview(wrapper_2_1,columns=(1,2,3), show="headings",height="4")
trv2.pack()

vsb = ttk.Scrollbar(wrapper_2_1, orient="vertical", command=trv2.yview)
vsb.place(x=390, y=0,height=100)

```

```
trv2.heading(1,text="Material ID")
trv2.column(1, width=80, stretch=Y)
trv2.heading(2,text="Thickness")
trv2.column(2, width=120, stretch=Y)
trv2.heading(3,text="Price per sqm")
trv2.column(3, width=120, stretch=Y)

trv2.bind('<Double 1>',getrow2)

cursor.execute("SELECT * FROM THK_Price;")
rowsm=cursor.fetchall()
update2(rowsm)

#wrapper=3

l1=Label(wrapper3,text="Criteria",width = 20,relief="raised",background='lightgrey',anchor=CENTER)
l1.grid(row=1,column=0)
l1=Label(wrapper3,text="Technical",width =85,relief="raised",background='lightgrey',anchor=CENTER)
l1.grid(columnspan=4,row=1,column=1)
l2=Label(wrapper3,text="Economic",width =42,relief="raised",background='lightgrey',anchor=CENTER)
l2.grid(columnspan=2,row=1,column=5)
l3=Label(wrapper3,text="Environmental",width =20,relief="raised",background='lightgrey',anchor=CENTER)
l3.grid(row=1,column=7)
l4=Label(wrapper3,text="Sub-Criteria",width = 20,relief="raised",background='lightgrey',anchor=CENTER)
l4.grid(row=2,column=0)
C4=Label(wrapper3,text="Weight",width = 20,relief="raised",background='lightgrey',anchor=CENTER)
C4.grid(row=3,column=0,pady = (0,10))
```

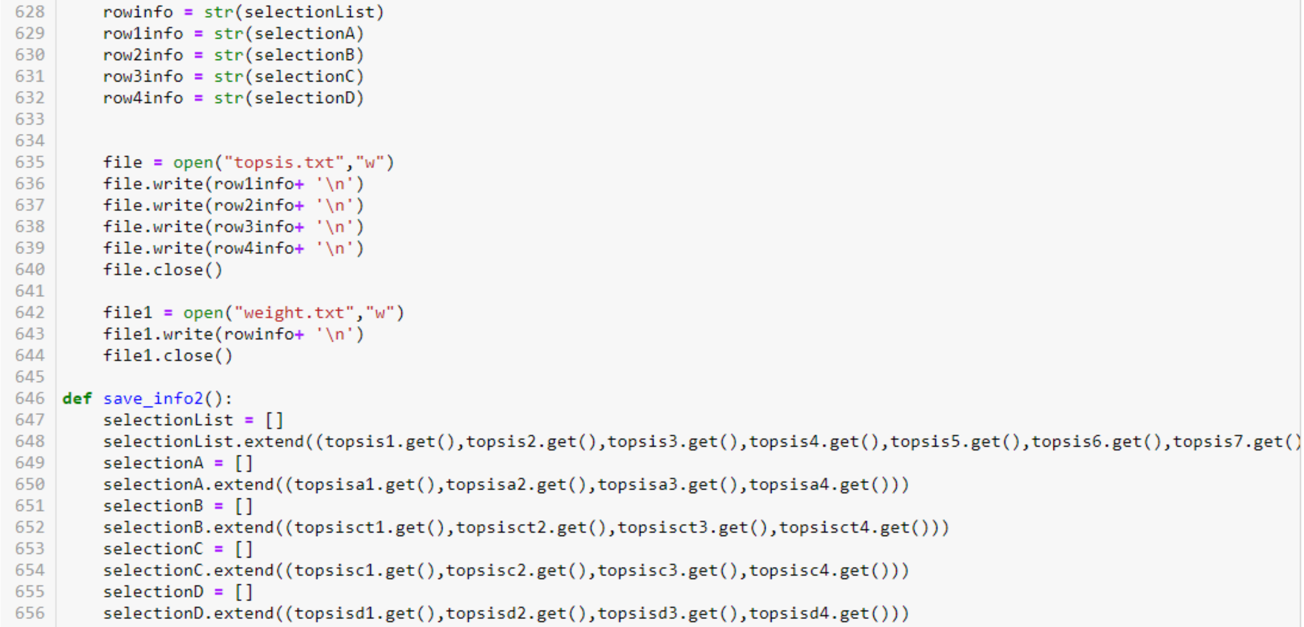
```
l5=Label(wrapper3,text="Work duration(Labor productivity)",width = 20,relief="raised",background='lightgrey',anchor=CENTER)
l5.grid(row=2,column=1)
l6=Label(wrapper3,text="Safety during construction",width = 20,relief="raised",background='lightgrey',anchor=CENTER)
l6.grid(row=2,column=2)
l7=Label(wrapper3,text="Water vapour diffusion",width = 20,relief="raised",background='lightgrey',anchor=CENTER)
l7.grid(row=2,column=3)
l8=Label(wrapper3,text="Fire resistance",width = 20,relief="raised",background='lightgrey',anchor=CENTER)
l8.grid(row=2,column=4)
l9=Label(wrapper3,text="LCC",width = 20,relief="raised",background='lightgrey',anchor=CENTER)
l9.grid(row=2,column=5)
l20=Label(wrapper3,text="Initial Cost",width = 20,relief="raised",background='lightgrey',anchor=CENTER)
l20.grid(row=2,column=6)
l21=Label(wrapper3,text="LCA (GWP)",width = 20,relief="raised",background='lightgrey',anchor=CENTER)
l21.grid(row=2,column=7)


total_rows = len(trv1.get_children())
total_columns =7
op_columns =4


#material lists
r_set=cursor.execute("SELECT ID FROM materials")
i=0
for materials in r_set:
    for j in range(len(materials)):
        e = Entry(wrapper3, width=20,background='lightgrey',justify='center')
        e.grid(row=i+4, column=0)
        e.insert(END, materials[j])
    i=i+1
e = Label(wrapper3,text=materials[j],borderwidth=2,relief='ridge',anchor=CENTER)

#topsis-blank
for i in range(total_rows):
    for j in range(total_columns):
        k = Label(wrapper3,text="", width=21,background='lightgrey')
        k.grid(row = i+4,column = j+1)

def selected(event):
    selectionList = []
    my1 = topsis1.get()
    my2 = topsis2.get()
    my3 = topsis3.get()
    my4 = topsis4.get()
    my5 = topsis5.get()
    my6 = topsis6.get()
    my7 = topsis7.get()
    selectionList.extend((my1,my2,my3,my4,my5,my6,my7))

def selected1(event):
    selectionList1= []
    n1 = topsisct1.get()
    n2 = topsisct2.get()
    n3 = topsisct3.get()
    n4 = topsisct4.get()
    selectionList1.extend((n1,n2,n3,n4))

def selected2(event):
    selectionList2= []
    n1 = topsisct1.get()
    n2 = topsisct2.get()
    n3 = topsisct3.get()
    n4 = topsisct4.get()
    selectionList2.extend((n1,n2,n3,n4))
def selected3(event):
    selectionList3= []
    n1 = topsisct1.get()
    n2 = topsisct2.get()
    n3 = topsisct3.get()
    n4 = topsisct4.get()
    selectionList3.extend((n1,n2,n3,n4))
def selected4(event):
    selectionList4= []
```



```
    n1 = topsisct1.get()
    n2 = topsisct2.get()
    n3 = topsisct3.get()
    n4 = topsisct4.get()
    selectionList4.extend((n1,n2,n3,n4))

#weight
weight_op = ["Very low", "Low", "Medium low","Medium","Medium high", "High","Very high"]
options = ["Very Poor", "Poor", "Medium Poor","Fair","Medium Good", "Good","Very Good"]
varlist = {var: StringVar() for var in ["zero", "one", "two","three", "four","five","six"]}

topsis1 = ttk.Combobox(wrapper3, values = weight_op,textvariable=varlist["zero"] ,width = 18)
topsis2 = ttk.Combobox(wrapper3, values = weight_op,textvariable=varlist["one"] ,width = 18)
topsis3 = ttk.Combobox(wrapper3, values = weight_op,textvariable=varlist["two"] ,width = 18)
topsis4 = ttk.Combobox(wrapper3, values = weight_op,textvariable=varlist["three"] ,width = 18)
topsis5 = ttk.Combobox(wrapper3, values = weight_op,textvariable=varlist["four"] ,width = 18)
topsis6 = ttk.Combobox(wrapper3, values = weight_op,textvariable=varlist["five"] ,width = 18)
topsis7 = ttk.Combobox(wrapper3, values = weight_op,textvariable=varlist["six"] ,width = 18)
topsis1.grid(row = 3,column = 1,pady = (0,10))
topsis2.grid(row = 3,column = 2,pady = (0,10))
topsis3.grid(row = 3,column = 3,pady = (0,10))
topsis4.grid(row = 3,column = 4,pady = (0,10))
topsis5.grid(row = 3,column = 5,pady = (0,10))
topsis6.grid(row = 3,column = 6,pady = (0,10))
topsis7.grid(row = 3,column = 7,pady = (0,10))
topsis1.bind("<<ComboboxSelected>>",selected)
topsis2.bind("<<ComboboxSelected>>",selected)
topsis3.bind("<<ComboboxSelected>>",selected)
topsis4.bind("<<ComboboxSelected>>",selected)
topsis5.bind("<<ComboboxSelected>>",selected)
topsis6.bind("<<ComboboxSelected>>",selected)
topsis7.bind("<<ComboboxSelected>>",selected)


#topsis

varlista = {var: StringVar() for var in ["0","1","2","3"]}

topsisa1 = ttk.Combobox(wrapper3, values = options,textvariable=varlista["0"],width = 18)
topsisa2 = ttk.Combobox(wrapper3, values = options,textvariable=varlista["1"],width = 18)
topsisa3 = ttk.Combobox(wrapper3, values = options,textvariable=varlista["2"],width = 18)
topsisa4 = ttk.Combobox(wrapper3, values = options,textvariable=varlista["3"],width = 18)
topsisa1.grid(row = 4,column = 1)
topsisa2.grid(row = 4,column = 2)
topsisa3.grid(row = 4,column = 3)
topsisa4.grid(row = 4,column = 4)
topsisa1.bind("<<ComboboxSelected>>",selected1)
topsisa2.bind("<<ComboboxSelected>>",selected1)
topsisa3.bind("<<ComboboxSelected>>",selected1)
topsisa4.bind("<<ComboboxSelected>>",selected1)



varlistb = {var: StringVar() for var in ["0","1","2","3"]}


topsisct1 = ttk.Combobox(wrapper3, values = options,textvariable=varlistb["0"],width = 18)
topsisct2 = ttk.Combobox(wrapper3, values = options,textvariable=varlistb["1"],width = 18)
topsisct3 = ttk.Combobox(wrapper3, values = options,textvariable=varlistb["2"],width = 18)
topsisct4 = ttk.Combobox(wrapper3, values = options,textvariable=varlistb["3"],width = 18)
topsisct1.grid(row = 5,column = 1)
topsisct2.grid(row = 5,column = 2)
topsisct3.grid(row = 5,column = 3)
topsisct4.grid(row = 5,column = 4)
topsisct1.bind("<<ComboboxSelected>>",selected2)
topsisct2.bind("<<ComboboxSelected>>",selected2)
topsisct3.bind("<<ComboboxSelected>>",selected2)
topsisct4.bind("<<ComboboxSelected>>",selected2)


varlistc = {var: StringVar() for var in ["0","1","2","3"]}


```



```
topsisc1 = ttk.Combobox(wrapper3, values = options,textvariable=varlistc["0"],width = 18)
topsisc2 = ttk.Combobox(wrapper3, values = options,textvariable=varlistc["1"],width = 18)
topsisc3 = ttk.Combobox(wrapper3, values = options,textvariable=varlistc["2"],width = 18)
topsisc4 = ttk.Combobox(wrapper3, values = options,textvariable=varlistc["3"],width = 18)
topsisc1.grid(row = 6,column = 1)
topsisc2.grid(row = 6,column = 2)
topsisc3.grid(row = 6,column = 3)
topsisc4.grid(row = 6,column = 4)
topsisc1.bind("<<ComboboxSelected>>",selected3)
topsisc2.bind("<<ComboboxSelected>>",selected3)
topsisc3.bind("<<ComboboxSelected>>",selected3)
topsisc4.bind("<<ComboboxSelected>>",selected3)


varlistd = {var: StringVar() for var in ["0","1","2","3"]}


topsisd1 = ttk.Combobox(wrapper3, values = options,textvariable=varlistd["0"],width = 18)
topsisd2 = ttk.Combobox(wrapper3, values = options,textvariable=varlistd["1"],width = 18)
topsisd3 = ttk.Combobox(wrapper3, values = options,textvariable=varlistd["2"],width = 18)
topsisd4 = ttk.Combobox(wrapper3, values = options,textvariable=varlistd["3"],width = 18)
topsisd1.grid(row = 7,column = 1)
topsisd2.grid(row = 7,column = 2)
topsisd3.grid(row = 7,column = 3)
topsisd4.grid(row = 7,column = 4)
topsisd1.bind("<<ComboboxSelected>>",selected4)
topsisd2.bind("<<ComboboxSelected>>",selected4)
topsisd3.bind("<<ComboboxSelected>>",selected4)
topsisd4.bind("<<ComboboxSelected>>",selected4)


def save_info1():
    selectionList = []
    selectionList.extend((topsis1.get(),topsis2.get(),topsis3.get(),topsis4.get(),topsis5.get(),topsis6.get(),topsis7.get()
    selectionA = []
    selectionA.extend((topsisa1.get(),topsisa2.get(),topsisa3.get(),topsisa4.get()))
    selectionB = []
    selectionB.extend((topsisct1.get(),topsisct2.get(),topsisct3.get(),topsisct4.get()))
    selectionC = []
    selectionC.extend((topsisc1.get(),topsisc2.get(),topsisc3.get(),topsisc4.get()))
    selectionD = []
    selectionD.extend((topsisd1.get(),topsisd2.get(),topsisd3.get(),topsisd4.get()))

```

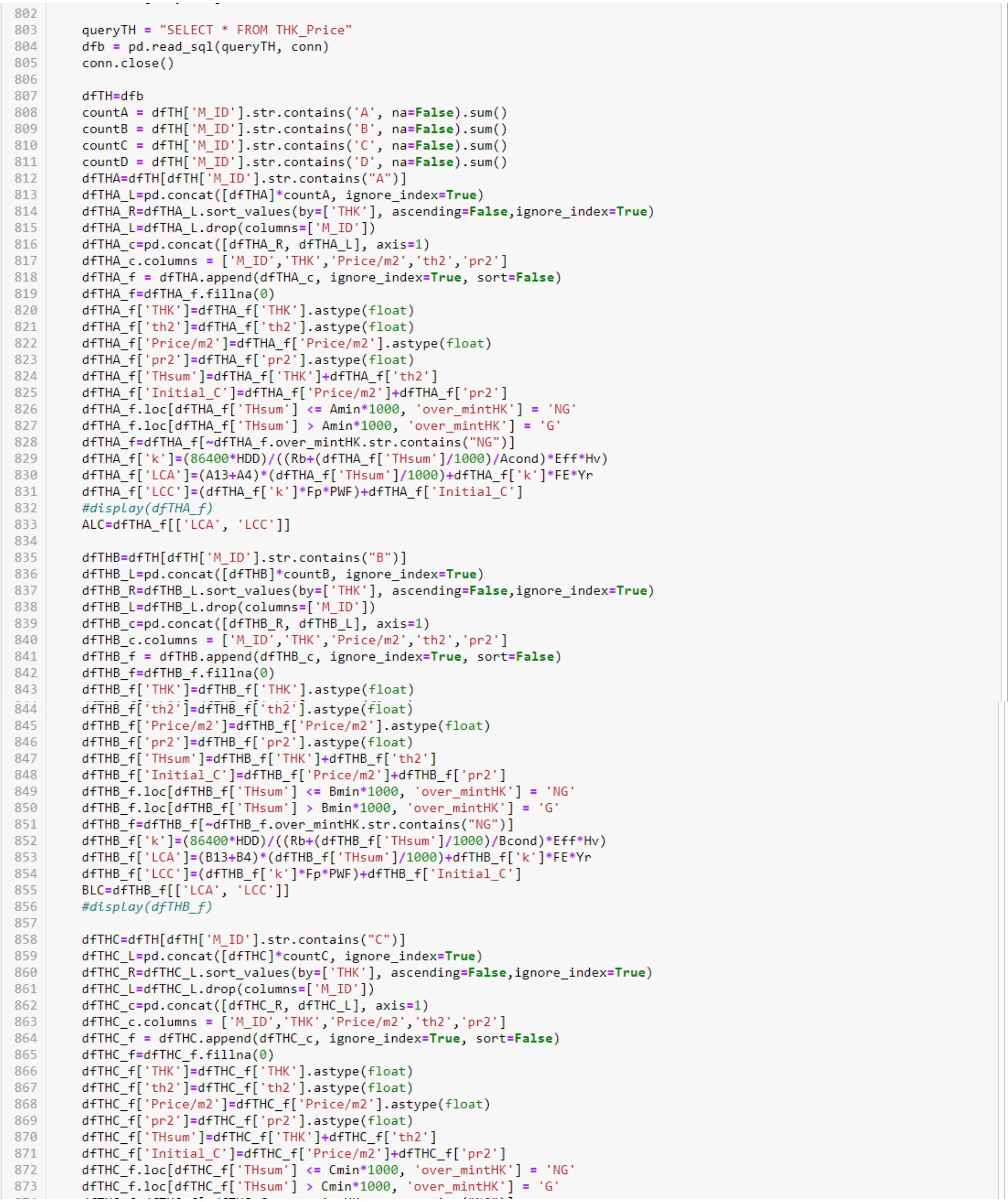


```
    rowinfo = str(selectionList)
    row1info = str(selectionA)
    row2info = str(selectionB)
    row3info = str(selectionC)
    row4info = str(selectionD)


    file = open("topsis.txt","w")
    file.write(row1info+ '\n')
    file.write(row2info+ '\n')
    file.write(row3info+ '\n')
    file.write(row4info+ '\n')
    file.close()

    file1 = open("weight.txt","w")
    file1.write(rowinfo+ '\n')
    file1.close()

def save_info2():
    selectionList = []
    selectionList.extend((topsis1.get(),topsis2.get(),topsis3.get(),topsis4.get(),topsis5.get(),topsis6.get(),topsis7.get()
    selectionA = []
    selectionA.extend((topsisa1.get(),topsisa2.get(),topsisa3.get(),topsisa4.get()))
    selectionB = []
    selectionB.extend((topsisct1.get(),topsisct2.get(),topsisct3.get(),topsisct4.get()))
    selectionC = []
    selectionC.extend((topsisc1.get(),topsisc2.get(),topsisc3.get(),topsisc4.get()))
    selectionD = []
    selectionD.extend((topsisd1.get(),topsisd2.get(),topsisd3.get(),topsisd4.get()))
```

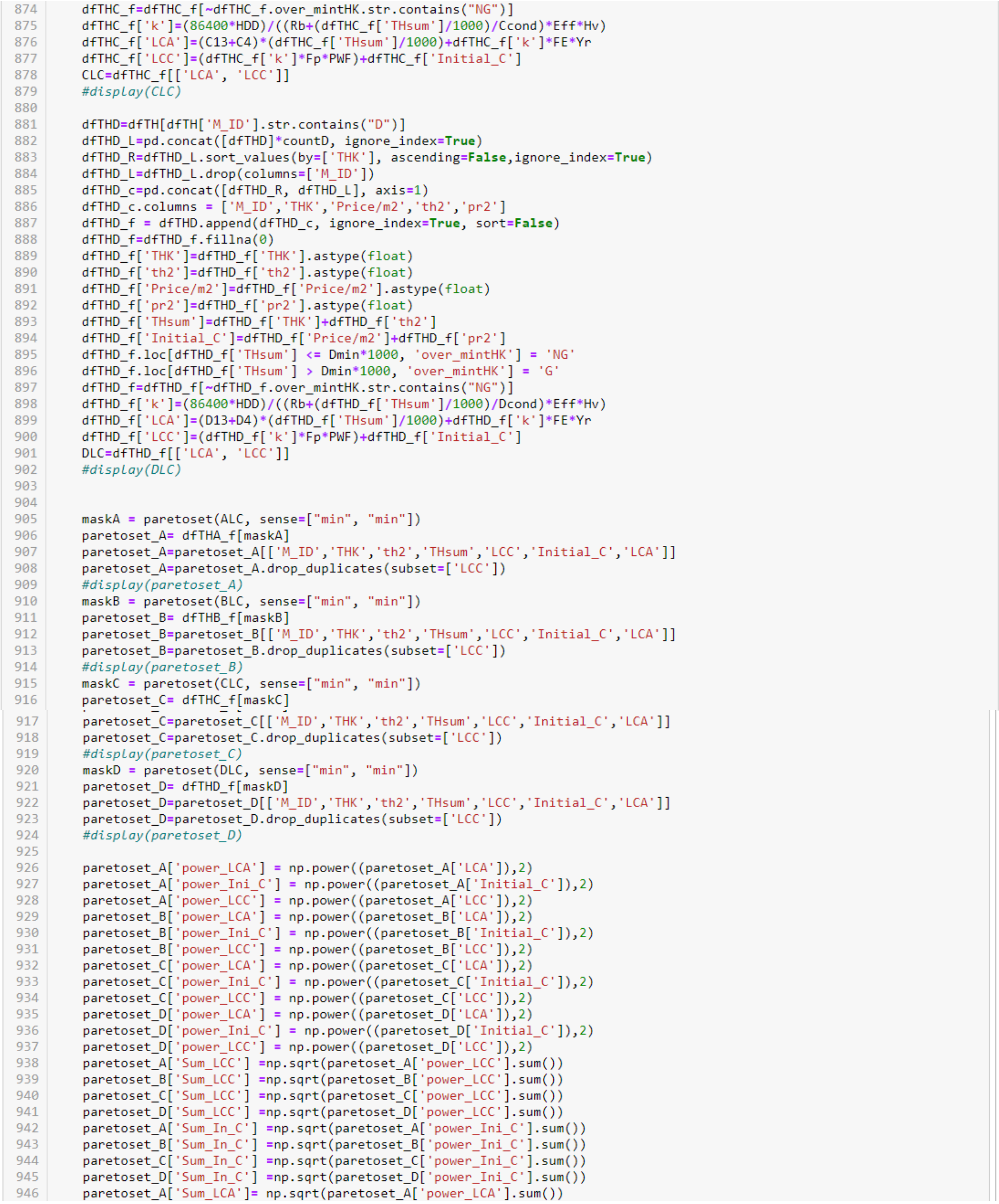
```
    rowinfo = str(selectionList)
    row1info = str(selectionA)
    row2info = str(selectionB)
    row3info = str(selectionC)
    row4info = str(selectionD)

    file = open("topsis.txt","+a")
    file.write(row1info+ '\n')
    file.write(row2info+ '\n')
    file.write(row3info+ '\n')
    file.write(row4info+ '\n')
    file.close()

    file1 = open("weight.txt","+a")
    file1.write(rowinfo+ '\n')
    file1.close()

def save_info3():
    selectionList = []
    selectionList.extend((topsis1.get(),topsis2.get(),topsis3.get(),topsis4.get(),topsis5.get(),topsis6.get(),topsis7.get()
    selectionA = []
    selectionA.extend((topsisa1.get(),topsisa2.get(),topsisa3.get(),topsisa4.get()))
    selectionB = []
    selectionB.extend((topsisct1.get(),topsisct2.get(),topsisct3.get(),topsisct4.get()))
    selectionC = []
    selectionC.extend((topsisc1.get(),topsisc2.get(),topsisc3.get(),topsisc4.get()))
    selectionD = []
    selectionD.extend((topsisd1.get(),topsisd2.get(),topsisd3.get(),topsisd4.get()))

    rowinfo = str(selectionList)
    row1info = str(selectionA)
    row2info = str(selectionB)
    row3info = str(selectionC)
    row4info = str(selectionD)


    file = open("topsis.txt","+a")
    file.write(row1info+ '\n')
    file.write(row2info+ '\n')
    file.write(row3info+ '\n')
    file.write(row4info+ '\n')
    file.close()
```

```

    file1 = open("weight.txt","+a")
    file1.write(rowinfo+ '\n')
    file1.close()


def delete_info():
    selectionList = []
    selectionList.extend((topsis1.set(""),topsis2.set(""),topsis3.set(""),topsis4.set(""),topsis5.set(""),topsis6.set(""),t
    selectionA = []
    selectionA.extend((topsisa1.set(""),topsisa2.set(""),topsisa3.set(""),topsisa4.set("")))
    selectionB = []
    selectionB.extend((topsisct1.set(""),topsisct2.set(""),topsisct3.set(""),topsisct4.set("")))
    selectionC = []
    selectionC.extend((topsisc1.set(""),topsisc2.set(""),topsisc3.set(""),topsisc4.set("")))
    selectionD = []
    selectionD.extend((topsisd1.set(""),topsisd2.set(""),topsisd3.set(""),topsisd4.set("")))


style = ttk.Style()
sbutton = Button(wrapper3,text="DC1 Submit Data",command=save_info1)
sbutton.grid(column=0,row=9,pady=10,sticky=W)

sbutton1 = Button(wrapper3,text="DC2 Submit Data",command=save_info2)
sbutton1.grid(column=1,row=9,sticky=W)

sbutton2 = Button(wrapper3,text="DC3 Submit Data",command=save_info3)
sbutton2.grid(column=2,row=9,sticky=W)

```

```
sbutton4 = Button(wrapper3,text="Clear Entry",command=delete_info)
sbutton4.grid(column=5,row=9,sticky=W)

sbutton5 = Button(wrapper3,text="Close and Run",command=lambda :[task(), root.destroy()],width = 80)
sbutton5.grid(columnspan=11,row=10,column=0,padx=5,pady=5,sticky=W+E)

style.theme_use('alt')
style.configure('TButton', font=('American typewriter', 10), background='#232323', foreground='white')
style.map('TButton', background=[('active', '#ff0000')])



def task():
    conn = pyodbc.connect(r'Driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=C:\Users\sungyikim\Desktop\Writings\testc
    cursor=conn.cursor()
    cursor.execute('select * from Basics')

    for row in cursor.fetchall():
        print (row)
    Yr=float(row[1])
    Int=float(row[2])
    Inf=float(row[3])
    Eff=0.8 #System Efficiency
    HDD=float(row[4])
    Hv=float(row[5])
    Fp=float(row[6])
    FE=float(row[7])
    Rb=float(row[8])
    Uv=float(row[9])


    if Int > Inf:
        r=(Int-Inf)/((100+Inf))
    else:
        r=(Inf-Int)/((100+Int))

    print(r)

    PWF = (((1+r)**Yr)-1)/(r*(1+r)**Yr)
    print(PWF)

    conn = pyodbc.connect(r'Driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=C:\Users\sungyikim\Desktop\Writings\testc
    query = "SELECT * FROM materials"
    dfa = pd.read_sql(query, conn)

    df=dfa
    df.rename(columns = {'GHG_A1A3' : 'GHG ( A1 ~ A3)', 'GHG_A4' : 'GHG ( A4) per km'}, inplace = True)
    df
    df['Conductivity'] = df['Conductivity'].astype(float)
    df['Distance'] = df['Distance'].astype(float)
    df['GHG ( A1 ~ A3)'] = df['GHG ( A1 ~ A3)'].astype(float)
    df['GHG ( A4) per km'] = df['GHG ( A4) per km'].astype(float)
    df['MinThick']= df["Conductivity"]*((1/Uv)-Rb)
    df['a13']= df["GHG ( A1 ~ A3)"]/df["Conductivity"]
    df['a4']= (df["GHG ( A4) per km"]*df["Distance"])/df["Conductivity"]
    df.set_index('ID', inplace=True)
    #display(df)
    Amin=df.at['A','MinThick']
    Bmin=df.at['B','MinThick']
    Cmin=df.at['C','MinThick']
    Dmin=df.at['D','MinThick']
    Acond=df.at['A','Conductivity']
    Bcond=df.at['B','Conductivity']
    Ccond=df.at['C','Conductivity']
    Dcond=df.at['D','Conductivity']
    A13=df.at['A','a13']
    B13=df.at['B','a13']
    C13=df.at['C','a13']
    D13=df.at['D','a13']
    A4=df.at['A','a4']
    B4=df.at['B','a4']
    C4=df.at['C','a4']
    D4=df.at['D','a4']
```



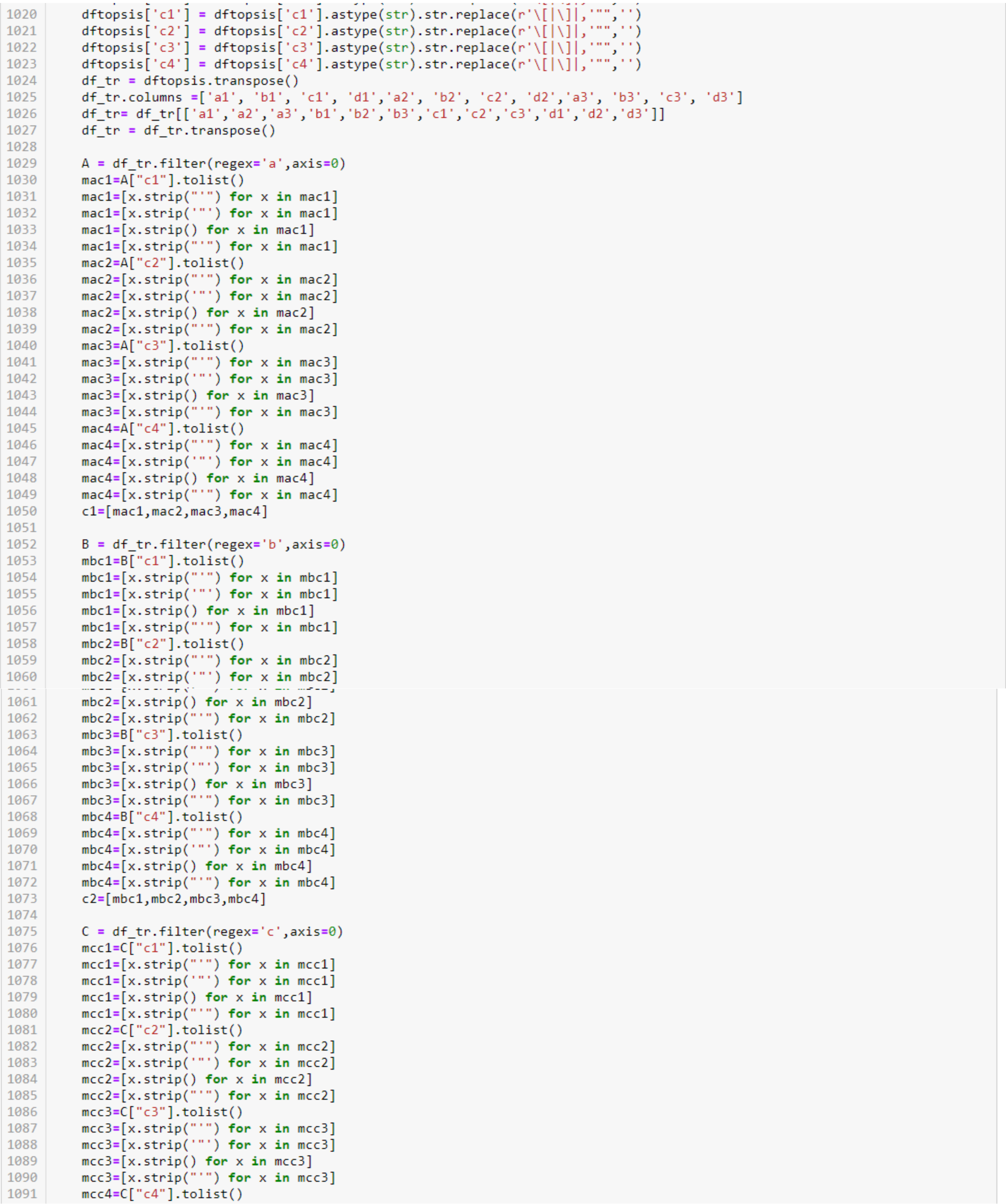
```
    queryTH = "SELECT * FROM THK_Price"
    dfb = pd.read_sql(queryTH, conn)
    conn.close()

    dfTH=dfb
    countA = dfTH['M_ID'].str.contains('A', na=False).sum()
    countB = dfTH['M_ID'].str.contains('B', na=False).sum()
    countC = dfTH['M_ID'].str.contains('C', na=False).sum()
    countD = dfTH['M_ID'].str.contains('D', na=False).sum()
    dfTHA=dfTH[dfTH['M_ID'].str.contains("A")]
    dfTHA_L=pd.concat([dfTHA]*countA, ignore_index=True)
    dfTHA_R=dfTHA_L.sort_values(by=['THK'], ascending=False,ignore_index=True)
    dfTHA_L=dfTHA_L.drop(columns=['M_ID'])
    dfTHA_c=pd.concat([dfTHA_R, dfTHA_L], axis=1)
    dfTHA_c.columns = ['M_ID','THK','Price/m2','th2','pr2']
    dfTHA_f = dfTHA.append(dfTHA_c, ignore_index=True, sort=False)
    dfTHA_f=dfTHA_f.fillna(0)
    dfTHA_f['THK']=dfTHA_f['THK'].astype(float)
    dfTHA_f['th2']=dfTHA_f['th2'].astype(float)
    dfTHA_f['Price/m2']=dfTHA_f['Price/m2'].astype(float)
    dfTHA_f['pr2']=dfTHA_f['pr2'].astype(float)
    dfTHA_f['THsum']=dfTHA_f['THK']+dfTHA_f['th2']
    dfTHA_f['Initial_C']=dfTHA_f['Price/m2']+dfTHA_f['pr2']
    dfTHA_f.loc[dfTHA_f['THsum'] <= Amin*1000, 'over_mintHK'] = 'NG'
    dfTHA_f.loc[dfTHA_f['THsum'] > Amin*1000, 'over_mintHK'] = 'G'
    dfTHA_f=dfTHA_f[~dfTHA_f.over_mintHK.str.contains("NG")]
    dfTHA_f['k']=(86400*HDD)/((Rb+(dfTHA_f['THsum']/1000)/Acond)*Eff*Hv)
    dfTHA_f['LCA']=(A13+A4)*(dfTHA_f['THsum']/1000)+dfTHA_f['k']*FE*Yr
    dfTHA_f['LCC']=(dfTHA_f['k']*Fp*PWF)+dfTHA_f['Initial_C']
    #display(dfTHA_f)
    ALC=dfTHA_f[['LCA', 'LCC']]

    dfTHB=dfTH[dfTH['M_ID'].str.contains("B")]
    dfTHB_L=pd.concat([dfTHB]*countB, ignore_index=True)
    dfTHB_R=dfTHB_L.sort_values(by=['THK'], ascending=False,ignore_index=True)
    dfTHB_L=dfTHB_L.drop(columns=['M_ID'])
    dfTHB_c=pd.concat([dfTHB_R, dfTHB_L], axis=1)
    dfTHB_c.columns = ['M_ID','THK','Price/m2','th2','pr2']
    dfTHB_f = dfTHB.append(dfTHB_c, ignore_index=True, sort=False)
    dfTHB_f=dfTHB_f.fillna(0)
    dfTHB_f['THK']=dfTHB_f['THK'].astype(float)
    dfTHB_f['th2']=dfTHB_f['th2'].astype(float)
    dfTHB_f['Price/m2']=dfTHB_f['Price/m2'].astype(float)
    dfTHB_f['pr2']=dfTHB_f['pr2'].astype(float)
    dfTHB_f['THsum']=dfTHB_f['THK']+dfTHB_f['th2']
    dfTHB_f['Initial_C']=dfTHB_f['Price/m2']+dfTHB_f['pr2']
    dfTHB_f.loc[dfTHB_f['THsum'] <= Bmin*1000, 'over_mintHK'] = 'NG'
    dfTHB_f.loc[dfTHB_f['THsum'] > Bmin*1000, 'over_mintHK'] = 'G'
    dfTHB_f=dfTHB_f[~dfTHB_f.over_mintHK.str.contains("NG")]
    dfTHB_f['k']=(86400*HDD)/((Rb+(dfTHB_f['THsum']/1000)/Bcond)*Eff*Hv)
    dfTHB_f['LCA']=(B13+B4)*(dfTHB_f['THsum']/1000)+dfTHB_f['k']*FE*Yr
    dfTHB_f['LCC']=(dfTHB_f['k']*Fp*PWF)+dfTHB_f['Initial_C']
    BLC=dfTHB_f[['LCA', 'LCC']]
    #display(dfTHB_f)

    dfTHC=dfTH[dfTH['M_ID'].str.contains("C")]
    dfTHC_L=pd.concat([dfTHC]*countC, ignore_index=True)
    dfTHC_R=dfTHC_L.sort_values(by=['THK'], ascending=False,ignore_index=True)
    dfTHC_L=dfTHC_L.drop(columns=['M_ID'])
    dfTHC_c=pd.concat([dfTHC_R, dfTHC_L], axis=1)
    dfTHC_c.columns = ['M_ID','THK','Price/m2','th2','pr2']
    dfTHC_f = dfTHC.append(dfTHC_c, ignore_index=True, sort=False)
    dfTHC_f=dfTHC_f.fillna(0)
    dfTHC_f['THK']=dfTHC_f['THK'].astype(float)
    dfTHC_f['th2']=dfTHC_f['th2'].astype(float)
    dfTHC_f['Price/m2']=dfTHC_f['Price/m2'].astype(float)
    dfTHC_f['pr2']=dfTHC_f['pr2'].astype(float)
    dfTHC_f['THsum']=dfTHC_f['THK']+dfTHC_f['th2']
    dfTHC_f['Initial_C']=dfTHC_f['Price/m2']+dfTHC_f['pr2']
    dfTHC_f.loc[dfTHC_f['THsum'] <= Cmin*1000, 'over_mintHK'] = 'NG'
    dfTHC_f.loc[dfTHC_f['THsum'] > Cmin*1000, 'over_mintHK'] = 'G'
```



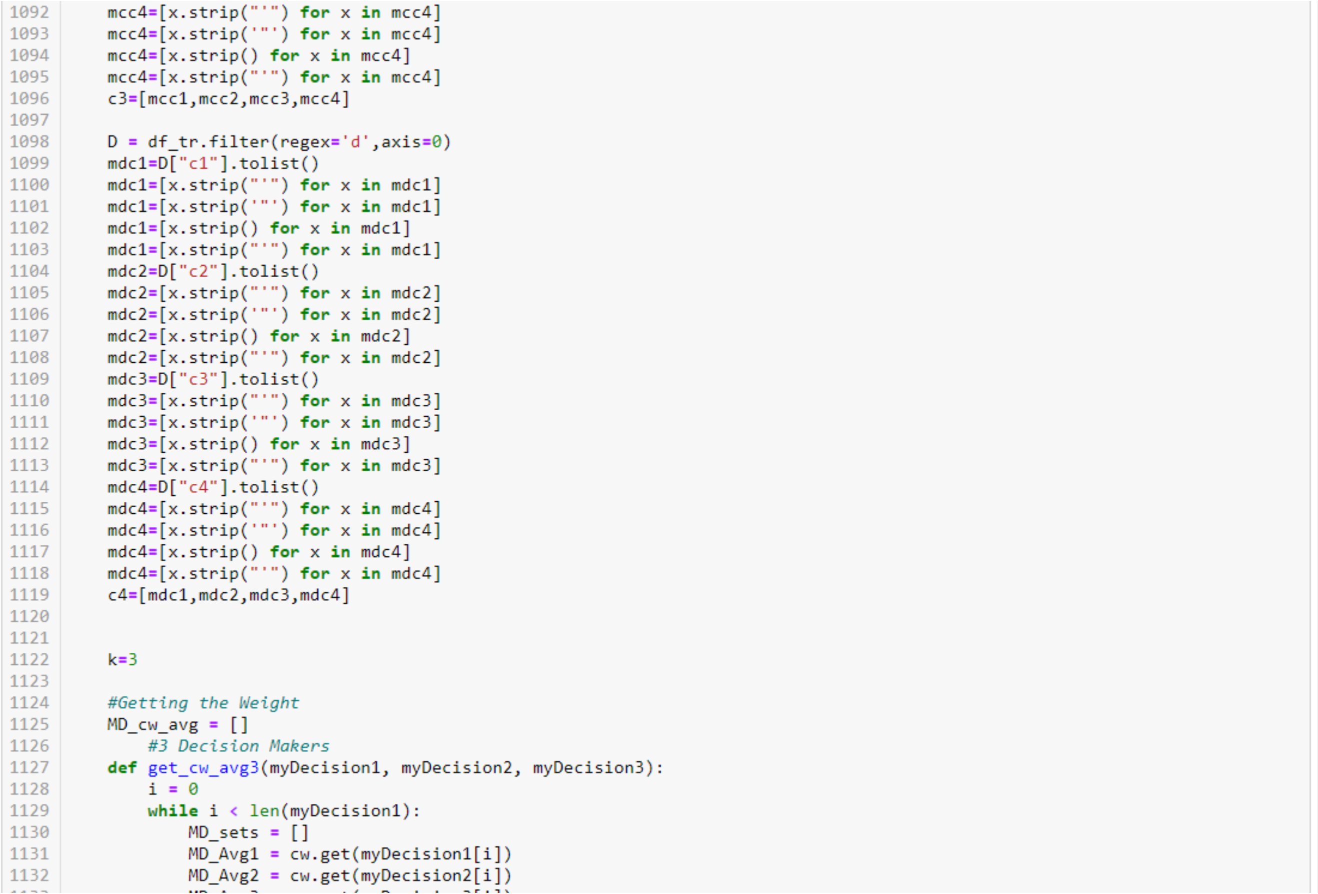
```
    dfTHC_f=dfTHC_f[~dfTHC_f.over_mintHK.str.contains("NG")]
    dfTHC_f['k']=(86400*HDD)/((Rb+(dfTHC_f['THsum']/1000)/Ccond)*Eff*Hv)
    dfTHC_f['LCA']=(C13+C4)*(dfTHC_f['THsum']/1000)+dfTHC_f['k']*FE*Yr
    dfTHC_f['LCC']=(dfTHC_f['k']*Fp*PWF)+dfTHC_f['Initial_C']
    CLC=dfTHC_f[['LCA', 'LCC']]
    #display(CLC)

    dfTHD=dfTH[dfTH['M_ID'].str.contains("D")]
    dfTHD_L=pd.concat([dfTHD]*countD, ignore_index=True)
    dfTHD_R=dfTHD_L.sort_values(by=['THK'], ascending=False,ignore_index=True)
    dfTHD_L=dfTHD_L.drop(columns=['M_ID'])
    dfTHD_c=pd.concat([dfTHD_R, dfTHD_L], axis=1)
    dfTHD_c.columns = ['M_ID','THK','Price/m2','th2','pr2']
    dfTHD_f = dfTHD.append(dfTHD_c, ignore_index=True, sort=False)
    dfTHD_f=dfTHD_f.fillna(0)
    dfTHD_f['THK']=dfTHD_f['THK'].astype(float)
    dfTHD_f['th2']=dfTHD_f['th2'].astype(float)
    dfTHD_f['Price/m2']=dfTHD_f['Price/m2'].astype(float)
    dfTHD_f['pr2']=dfTHD_f['pr2'].astype(float)
    dfTHD_f['THsum']=dfTHD_f['THK']+dfTHD_f['th2']
    dfTHD_f['Initial_C']=dfTHD_f['Price/m2']+dfTHD_f['pr2']
    dfTHD_f.loc[dfTHD_f['THsum'] <= Dmin*1000, 'over_mintHK'] = 'NG'
    dfTHD_f.loc[dfTHD_f['THsum'] > Dmin*1000, 'over_mintHK'] = 'G'
    dfTHD_f=dfTHD_f[~dfTHD_f.over_mintHK.str.contains("NG")]
    dfTHD_f['k']=(86400*HDD)/((Rb+(dfTHD_f['THsum']/1000)/Dcond)*Eff*Hv)
    dfTHD_f['LCA']=(D13+D4)*(dfTHD_f['THsum']/1000)+dfTHD_f['k']*FE*Yr
    dfTHD_f['LCC']=(dfTHD_f['k']*Fp*PWF)+dfTHD_f['Initial_C']
    DLC=dfTHD_f[['LCA', 'LCC']]
    #display(DLC)


    maskA = paretoset(ALC, sense=["min", "min"])
    paretoset_A= dfTHA_f[maskA]
    paretoset_A=paretoset_A[['M_ID','THK','th2','THsum','LCC','Initial_C','LCA']]
    paretoset_A=paretoset_A.drop_duplicates(subset=['LCC'])
    #display(paretoset_A)
    maskB = paretoset(BLC, sense=["min", "min"])
    paretoset_B= dfTHB_f[maskB]
    paretoset_B=paretoset_B[['M_ID','THK','th2','THsum','LCC','Initial_C','LCA']]
    paretoset_B=paretoset_B.drop_duplicates(subset=['LCC'])
    #display(paretoset_B)
    maskC = paretoset(CLC, sense=["min", "min"])
    paretoset_C= dfTHC_f[maskC]
    paretoset_C=paretoset_C[['M_ID','THK','th2','THsum','LCC','Initial_C','LCA']]
    paretoset_C=paretoset_C.drop_duplicates(subset=['LCC'])
    #display(paretoset_C)
    maskD = paretoset(DLC, sense=["min", "min"])
    paretoset_D= dfTHD_f[maskD]
    paretoset_D=paretoset_D[['M_ID','THK','th2','THsum','LCC','Initial_C','LCA']]
    paretoset_D=paretoset_D.drop_duplicates(subset=['LCC'])
    #display(paretoset_D)

    paretoset_A['power_LCA'] = np.power((paretoset_A['LCA']),2)
    paretoset_A['power_Ini_C'] = np.power((paretoset_A['Initial_C']),2)
    paretoset_A['power_LCC'] = np.power((paretoset_A['LCC']),2)
    paretoset_B['power_LCA'] = np.power((paretoset_B['LCA']),2)
    paretoset_B['power_Ini_C'] = np.power((paretoset_B['Initial_C']),2)
    paretoset_B['power_LCC'] = np.power((paretoset_B['LCC']),2)
    paretoset_C['power_LCA'] = np.power((paretoset_C['LCA']),2)
    paretoset_C['power_Ini_C'] = np.power((paretoset_C['Initial_C']),2)
    paretoset_C['power_LCC'] = np.power((paretoset_C['LCC']),2)
    paretoset_D['power_LCA'] = np.power((paretoset_D['LCA']),2)
    paretoset_D['power_Ini_C'] = np.power((paretoset_D['Initial_C']),2)
    paretoset_D['power_LCC'] = np.power((paretoset_D['LCC']),2)
    paretoset_A['Sum_LCC'] =np.sqrt(paretoset_A['power_LCC'].sum())
    paretoset_B['Sum_LCC'] =np.sqrt(paretoset_B['power_LCC'].sum())
    paretoset_C['Sum_LCC'] =np.sqrt(paretoset_C['power_LCC'].sum())
    paretoset_D['Sum_LCC'] =np.sqrt(paretoset_D['power_LCC'].sum())
    paretoset_A['Sum_In_C'] =np.sqrt(paretoset_A['power_Ini_C'].sum())
    paretoset_B['Sum_In_C'] =np.sqrt(paretoset_B['power_Ini_C'].sum())
    paretoset_C['Sum_In_C'] =np.sqrt(paretoset_C['power_Ini_C'].sum())
    paretoset_D['Sum_In_C'] =np.sqrt(paretoset_D['power_Ini_C'].sum())
    paretoset_A['Sum_LCA']= np.sqrt(paretoset_A['power_LCA'].sum())
```

```
    paretoset_B['Sum_LCA']= np.sqrt(paretoset_B['power_LCA'].sum())
    paretoset_C['Sum_LCA']= np.sqrt(paretoset_C['power_LCA'].sum())
    paretoset_D['Sum_LCA']= np.sqrt(paretoset_D['power_LCA'].sum())
    paretoset_A['qLCC'] =paretoset_A['power_LCC']/paretoset_A['Sum_LCC']
    paretoset_B['qLCC'] =paretoset_B['power_LCC']/paretoset_B['Sum_LCC']
    paretoset_C['qLCC'] =paretoset_C['power_LCC']/paretoset_C['Sum_LCC']
    paretoset_D['qLCC'] =paretoset_D['power_LCC']/paretoset_D['Sum_LCC']
    paretoset_A['qIn_C'] =paretoset_A['power_Ini_C']/paretoset_A['Sum_In_C']
    paretoset_B['qIn_C'] =paretoset_B['power_Ini_C']/paretoset_B['Sum_In_C']
    paretoset_C['qIn_C'] =paretoset_C['power_Ini_C']/paretoset_C['Sum_In_C']
    paretoset_D['qIn_C'] =paretoset_D['power_Ini_C']/paretoset_D['Sum_In_C']
    paretoset_A['qLCA'] = paretoset_A['power_LCA']/ paretoset_A['Sum_LCA']
    paretoset_B['qLCA'] = paretoset_B['power_LCA']/ paretoset_B['Sum_LCA']
    paretoset_C['qLCA'] = paretoset_C['power_LCA']/ paretoset_C['Sum_LCA']
    paretoset_D['qLCA'] = paretoset_D['power_LCA']/ paretoset_D['Sum_LCA']

    obj=pd.concat([paretoset_A,paretoset_B,paretoset_C,paretoset_D])
    orlc = obj[['M_ID','THK','th2','THsum','LCC','LCA']]
    orlc = orlc.rename(columns={'THK': "th1", "THsum": "THKsum"})
    display(orlc)
    obj = obj[['M_ID','THK','th2','THsum','qLCC','qIn_C','qLCA']]
    #display(obj)

    # Steps 1 and 2
    # Define a dictionary with the linguistic variables for the
    # criteria weights
    cw = {'Very low':[0, 0, 0.2], 'Low':[0, 0.1, 0.2],'Medium low':[0, 0.2, 0.3], 'Medium':[0.1, 0.3, 0.5],\
          'Medium high':[0.4, 0.5, 0.6], 'High':[0.6, 0.7, 0.8],'Very high':[0.7, 0.9, 1]}

    # Define a dictionary with the linguistic variables for the
    # ratings
    r = {'Very Poor':[0, 0, 1], 'Poor':[0, 1, 3], 'Medium Poor':[1, 3, 5],'Fair':[3, 5, 7],\
         'Medium Good':[5, 7, 9], 'Good':[7, 9, 10], 'Very Good':[9, 10, 10]}

    # The matrix with the criteria weights
    # The ratings of the six candidate sites by the decision
    # makers under all criteria
    weigt_list = pd.read_csv('weight.txt',header=None)
    dfweigt = pd.DataFrame(weigt_list)
    dfweigt.columns =['cr1', 'cr2', 'cr3', 'cr4','cr5','cr6','cr7']
    dfweigt['cr1'] = dfweigt['cr1'].astype(str).str.replace(r'\[|\]|,'"",'')
    dfweigt['cr2'] = dfweigt['cr2'].astype(str).str.replace(r'\[|\]|,'"",'')
```

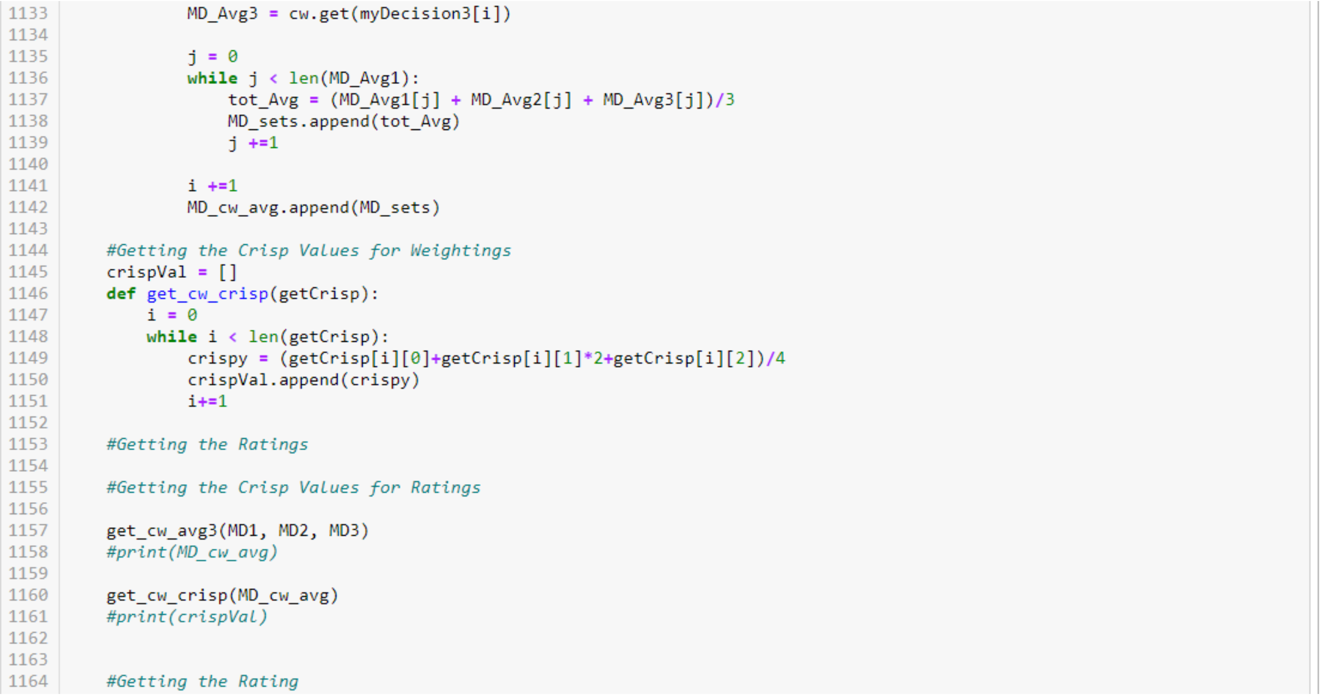
```
    dfweigt['cr3'] = dfweigt['cr3'].astype(str).str.replace(r'\[|\]|,'"",'')
    dfweigt['cr4'] = dfweigt['cr4'].astype(str).str.replace(r'\[|\]|,'"",'')
    dfweigt['cr5'] = dfweigt['cr5'].astype(str).str.replace(r'\[|\]|,'"",'')
    dfweigt['cr6'] = dfweigt['cr6'].astype(str).str.replace(r'\[|\]|,'"",'')
    dfweigt['cr7'] = dfweigt['cr7'].astype(str).str.replace(r'\[|\]|,'"",'')

    MD1=dfweigt.iloc[0].values.tolist()
    MD1=[elem.strip("'") for elem in MD1]
    MD1=[elem.strip('"') for elem in MD1]
    MD1=[x.strip() for x in MD1]
    MD1=[x.strip("'") for x in MD1]

    MD2=dfweigt.iloc[1].values.tolist()
    MD2=[elem.strip("'") for elem in MD2]
    MD2=[elem.strip('"') for elem in MD2]
    MD2=[x.strip() for x in MD2]
    MD2=[x.strip("'") for x in MD2]

    MD3=dfweigt.iloc[2].values.tolist()
    MD3=[elem.strip("'") for elem in MD3]
    MD3=[elem.strip('"') for elem in MD3]
    MD3=[x.strip() for x in MD3]
    MD3=[x.strip("'") for x in MD3]

    print(MD1,MD2,MD3)

    data_list = pd.read_csv('topsis.txt', sep=",",header=None)
    dftopsis = pd.DataFrame(data_list)
    dftopsis.columns =['c1', 'c2', 'c3', 'c4']

    dftopsis['c1'] = dftopsis['c1'].astype(str).str.replace(r'\[|\]|,'"",'')
```



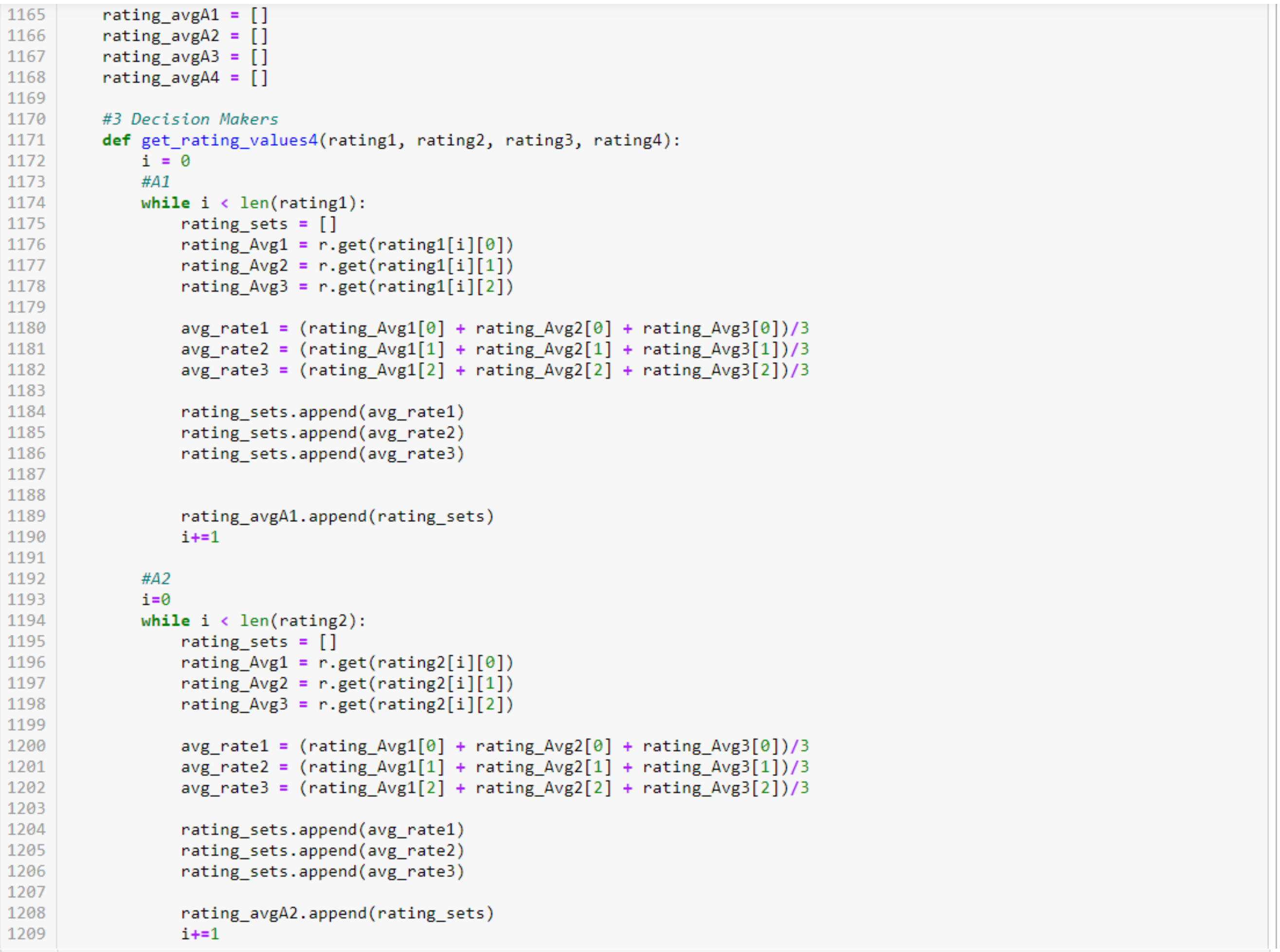
```
    dftopsis['c1'] = dftopsis['c1'].astype(str).str.replace(r'\[|\]|,'"",'')
    dftopsis['c2'] = dftopsis['c2'].astype(str).str.replace(r'\[|\]|,'"",'')
    dftopsis['c3'] = dftopsis['c3'].astype(str).str.replace(r'\[|\]|,'"",'')
    dftopsis['c4'] = dftopsis['c4'].astype(str).str.replace(r'\[|\]|,'"",'')
    df_tr = dftopsis.transpose()
    df_tr.columns =['a1', 'b1', 'c1', 'd1','a2', 'b2', 'c2', 'd2','a3', 'b3', 'c3', 'd3']
    df_tr= df_tr[['a1','a2','a3','b1','b2','b3','c1','c2','c3','d1','d2','d3']]
    df_tr = df_tr.transpose()

    A = df_tr.filter(regex='a',axis=0)
    mac1=A["c1"].tolist()
    mac1=[x.strip("'") for x in mac1]
    mac1=[x.strip('"') for x in mac1]
    mac1=[x.strip() for x in mac1]
    mac1=[x.strip("'") for x in mac1]
    mac2=A["c2"].tolist()
    mac2=[x.strip("'") for x in mac2]
    mac2=[x.strip('"') for x in mac2]
    mac2=[x.strip() for x in mac2]
    mac2=[x.strip("'") for x in mac2]
    mac3=A["c3"].tolist()
    mac3=[x.strip("'") for x in mac3]
    mac3=[x.strip('"') for x in mac3]
    mac3=[x.strip() for x in mac3]
    mac3=[x.strip("'") for x in mac3]
    mac4=A["c4"].tolist()
    mac4=[x.strip("'") for x in mac4]
    mac4=[x.strip('"') for x in mac4]
    mac4=[x.strip() for x in mac4]
    mac4=[x.strip("'") for x in mac4]
    c1=[mac1,mac2,mac3,mac4]

    B = df_tr.filter(regex='b',axis=0)
    mbc1=B["c1"].tolist()
    mbc1=[x.strip("'") for x in mbc1]
    mbc1=[x.strip('"') for x in mbc1]
    mbc1=[x.strip() for x in mbc1]
    mbc1=[x.strip("'") for x in mbc1]
    mbc2=B["c2"].tolist()
    mbc2=[x.strip("'") for x in mbc2]
    mbc2=[x.strip('"') for x in mbc2]
    mbc2=[x.strip() for x in mbc2]
    mbc2=[x.strip("'") for x in mbc2]
    mbc3=B["c3"].tolist()
    mbc3=[x.strip("'") for x in mbc3]
    mbc3=[x.strip('"') for x in mbc3]
    mbc3=[x.strip() for x in mbc3]
    mbc3=[x.strip("'") for x in mbc3]
    mbc4=B["c4"].tolist()
    mbc4=[x.strip("'") for x in mbc4]
    mbc4=[x.strip('"') for x in mbc4]
    mbc4=[x.strip() for x in mbc4]
    mbc4=[x.strip("'") for x in mbc4]
    c2=[mbc1,mbc2,mbc3,mbc4]

    C = df_tr.filter(regex='c',axis=0)
    mcc1=C["c1"].tolist()
    mcc1=[x.strip("'") for x in mcc1]
    mcc1=[x.strip('"') for x in mcc1]
    mcc1=[x.strip() for x in mcc1]
    mcc1=[x.strip("'") for x in mcc1]
    mcc2=C["c2"].tolist()
    mcc2=[x.strip("'") for x in mcc2]
    mcc2=[x.strip('"') for x in mcc2]
    mcc2=[x.strip() for x in mcc2]
    mcc2=[x.strip("'") for x in mcc2]
    mcc3=C["c3"].tolist()
    mcc3=[x.strip("'") for x in mcc3]
    mcc3=[x.strip('"') for x in mcc3]
    mcc3=[x.strip() for x in mcc3]
    mcc3=[x.strip("'") for x in mcc3]
    mcc4=C["c4"].tolist()
```



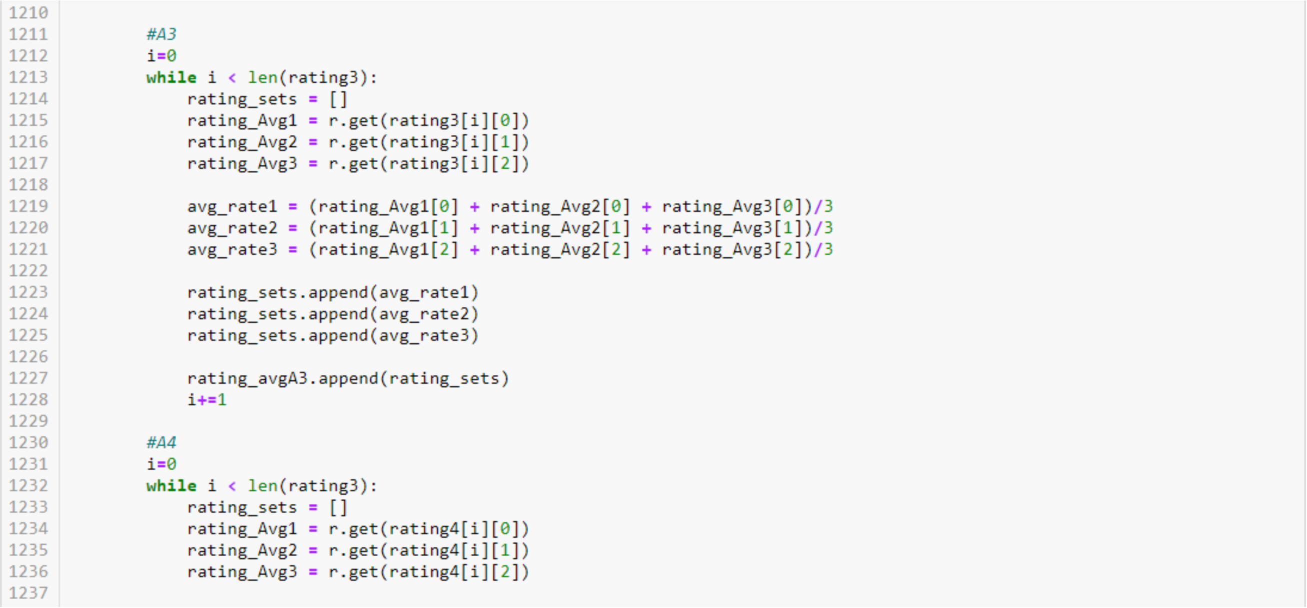
```
    mcc4=[x.strip("'") for x in mcc4]
    mcc4=[x.strip('"') for x in mcc4]
    mcc4=[x.strip() for x in mcc4]
    mcc4=[x.strip("'") for x in mcc4]
    c3=[mcc1,mcc2,mcc3,mcc4]

    D = df_tr.filter(regex='d',axis=0)
    mdc1=D["c1"].tolist()
    mdc1=[x.strip("'") for x in mdc1]
    mdc1=[x.strip('"') for x in mdc1]
    mdc1=[x.strip() for x in mdc1]
    mdc1=[x.strip("'") for x in mdc1]
    mdc2=D["c2"].tolist()
    mdc2=[x.strip("'") for x in mdc2]
    mdc2=[x.strip('"') for x in mdc2]
    mdc2=[x.strip() for x in mdc2]
    mdc2=[x.strip("'") for x in mdc2]
    mdc3=D["c3"].tolist()
    mdc3=[x.strip("'") for x in mdc3]
    mdc3=[x.strip('"') for x in mdc3]
    mdc3=[x.strip() for x in mdc3]
    mdc3=[x.strip("'") for x in mdc3]
    mdc4=D["c4"].tolist()
    mdc4=[x.strip("'") for x in mdc4]
    mdc4=[x.strip('"') for x in mdc4]
    mdc4=[x.strip() for x in mdc4]
    mdc4=[x.strip("'") for x in mdc4]
    c4=[mdc1,mdc2,mdc3,mdc4]


    k=3

    #Getting the Weight
    MD_cw_avg = []
        #3 Decision Makers
    def get_cw_avg3(myDecision1, myDecision2, myDecision3):
        i = 0
        while i < len(myDecision1):
            MD_sets = []
            MD_Avg1 = cw.get(myDecision1[i])
            MD_Avg2 = cw.get(myDecision2[i])
```



```
            MD_Avg3 = cw.get(myDecision3[i])

            j = 0
            while j < len(MD_Avg1):
                tot_Avg = (MD_Avg1[j] + MD_Avg2[j] + MD_Avg3[j])/3
                MD_sets.append(tot_Avg)
                j +=1

            i +=1
            MD_cw_avg.append(MD_sets)

    #Getting the Crisp Values for Weightings
    crispVal = []
    def get_cw_crisp(getCrisp):
        i = 0
        while i < len(getCrisp):
            crispy = (getCrisp[i][0]+getCrisp[i][1]*2+getCrisp[i][2])/4
            crispVal.append(crispy)
            i+=1

    #Getting the Ratings

    #Getting the Crisp Values for Ratings

    get_cw_avg3(MD1, MD2, MD3)
    #print(MD_cw_avg)

    get_cw_crisp(MD_cw_avg)
    #print(crispVal)


    #Getting the Rating
```

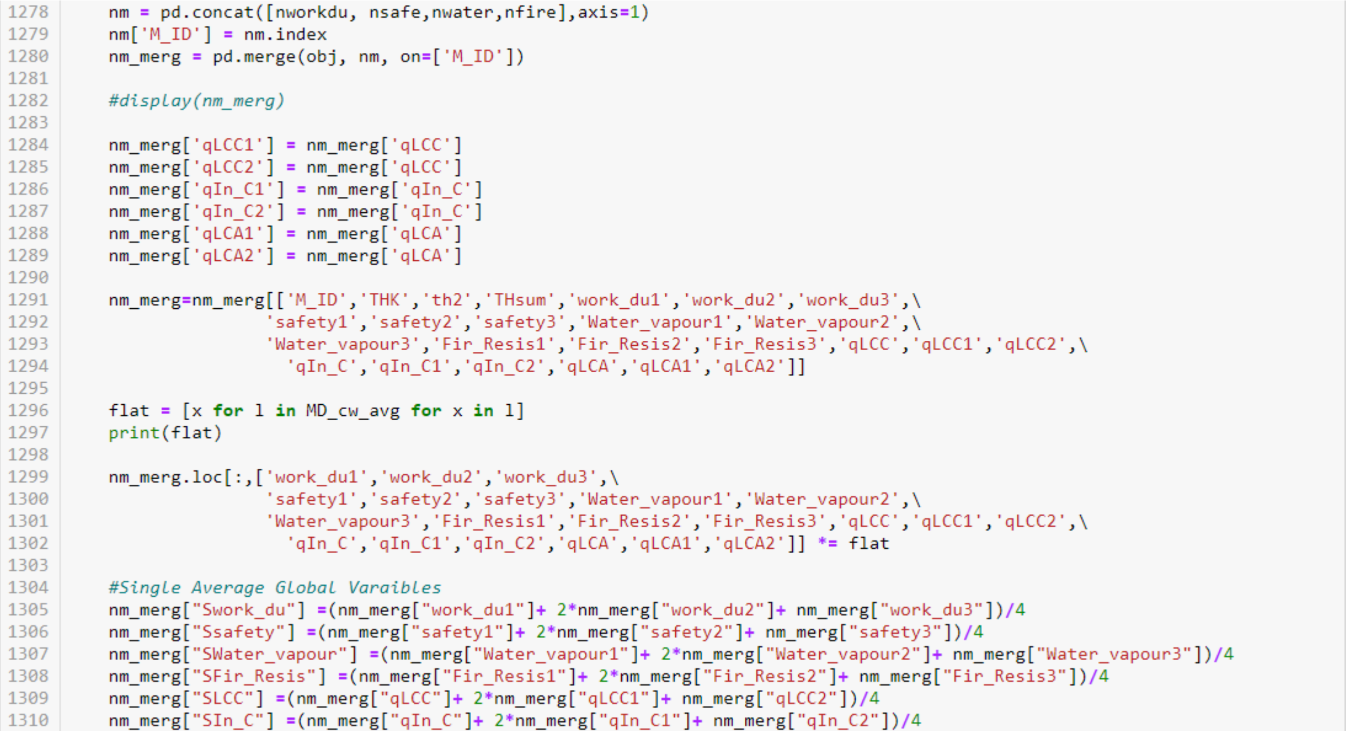


```
    rating_avgA1 = []
    rating_avgA2 = []
    rating_avgA3 = []
    rating_avgA4 = []

    #3 Decision Makers
    def get_rating_values4(rating1, rating2, rating3, rating4):
        i = 0
        #A1
        while i < len(rating1):
            rating_sets = []
            rating_Avg1 = r.get(rating1[i][0])
            rating_Avg2 = r.get(rating1[i][1])
            rating_Avg3 = r.get(rating1[i][2])

            avg_rate1 = (rating_Avg1[0] + rating_Avg2[0] + rating_Avg3[0])/3
            avg_rate2 = (rating_Avg1[1] + rating_Avg2[1] + rating_Avg3[1])/3
            avg_rate3 = (rating_Avg1[2] + rating_Avg2[2] + rating_Avg3[2])/3

            rating_sets.append(avg_rate1)
            rating_sets.append(avg_rate2)
            rating_sets.append(avg_rate3)


            rating_avgA1.append(rating_sets)
            i+=1

        #A2
        i=0
        while i < len(rating2):
            rating_sets = []
            rating_Avg1 = r.get(rating2[i][0])
            rating_Avg2 = r.get(rating2[i][1])
            rating_Avg3 = r.get(rating2[i][2])

            avg_rate1 = (rating_Avg1[0] + rating_Avg2[0] + rating_Avg3[0])/3
            avg_rate2 = (rating_Avg1[1] + rating_Avg2[1] + rating_Avg3[1])/3
            avg_rate3 = (rating_Avg1[2] + rating_Avg2[2] + rating_Avg3[2])/3

            rating_sets.append(avg_rate1)
            rating_sets.append(avg_rate2)
            rating_sets.append(avg_rate3)

            rating_avgA2.append(rating_sets)
            i+=1
```



```

        #A3
        i=0
        while i < len(rating3):
            rating_sets = []
            rating_Avg1 = r.get(rating3[i][0])
            rating_Avg2 = r.get(rating3[i][1])
            rating_Avg3 = r.get(rating3[i][2])

            avg_rate1 = (rating_Avg1[0] + rating_Avg2[0] + rating_Avg3[0])/3
            avg_rate2 = (rating_Avg1[1] + rating_Avg2[1] + rating_Avg3[1])/3
            avg_rate3 = (rating_Avg1[2] + rating_Avg2[2] + rating_Avg3[2])/3

            rating_sets.append(avg_rate1)
            rating_sets.append(avg_rate2)
            rating_sets.append(avg_rate3)

            rating_avgA3.append(rating_sets)
            i+=1

        #A4
        i=0
        while i < len(rating3):
            rating_sets = []
            rating_Avg1 = r.get(rating4[i][0])
            rating_Avg2 = r.get(rating4[i][1])
            rating_Avg3 = r.get(rating4[i][2])

```

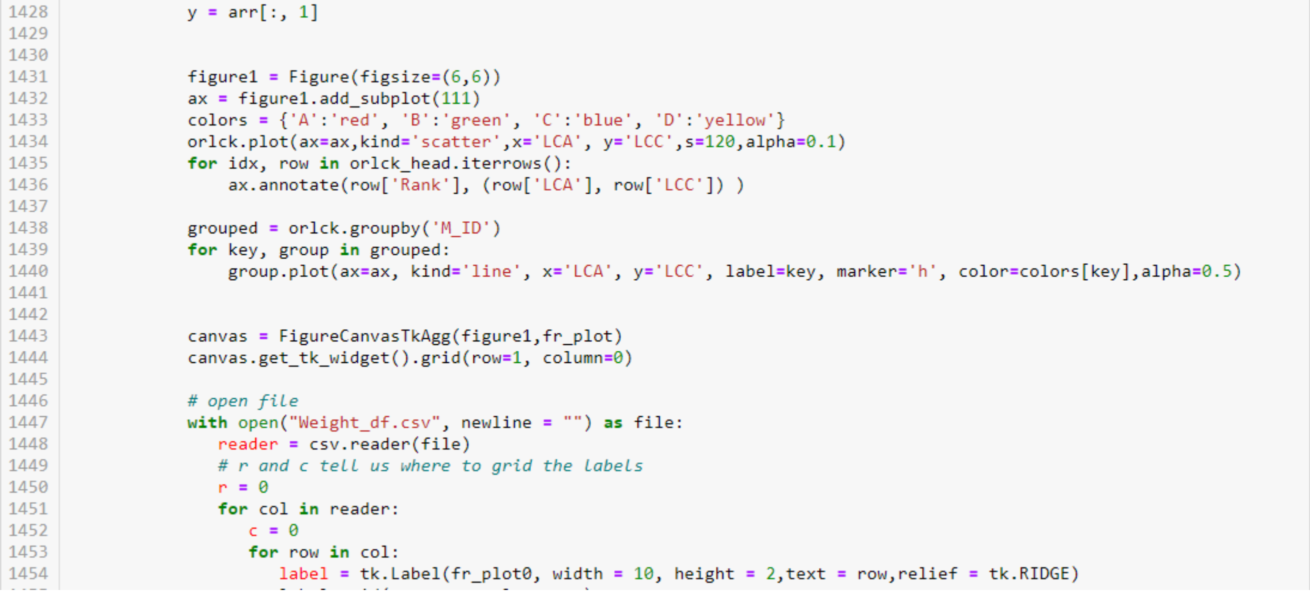
```
            avg_rate1 = (rating_Avg1[0] + rating_Avg2[0] + rating_Avg3[0])/3
            avg_rate2 = (rating_Avg1[1] + rating_Avg2[1] + rating_Avg3[1])/3
            avg_rate3 = (rating_Avg1[2] + rating_Avg2[2] + rating_Avg3[2])/3

            rating_sets.append(avg_rate1)
            rating_sets.append(avg_rate2)
            rating_sets.append(avg_rate3)

            rating_avgA4.append(rating_sets)
            i+=1

    #Normalize Varaibles
    get_rating_values4(c1, c2, c3, c4)
    average_ratings=pd.DataFrame({'A':rating_avgA1,'B':rating_avgA2,'C':rating_avgA3,'D':rating_avgA4})
    average_ratings=average_ratings.transpose()
    average_ratings.columns =['work_du', 'safety', 'Water_vapour', 'Fir_Resis']
    average_ratings['powork_du']=average_ratings['work_du'].apply(max)
    average_ratings['posafety']=average_ratings['safety'].apply(max)
    average_ratings['poWater_vapour']=average_ratings['Water_vapour'].apply(max)
    average_ratings['poFir_Resis']=average_ratings['Fir_Resis'].apply(max)
    average_ratings['powork_du']=  average_ratings['powork_du']**2
    average_ratings['posafety']=average_ratings['posafety']**2
    average_ratings['poWater_vapour']=average_ratings['poWater_vapour']**2
    average_ratings['poFir_Resis']=average_ratings['poFir_Resis']**2
    average_ratings['S_powork_du'] =np.sqrt(average_ratings['powork_du'].sum())
    average_ratings['S_posafety'] =np.sqrt(average_ratings['posafety'].sum())
    average_ratings['S_poWater_vapour'] =np.sqrt(average_ratings['poWater_vapour'].sum())
    average_ratings['S_poFir_Resisr'] =np.sqrt(average_ratings['poFir_Resis'].sum())
    kr= average_ratings.iloc[:, -4:]
    listdf = average_ratings.iloc[:, 0:4]
    listdf[['work_du1','work_du2','work_du3']] = pd.DataFrame(listdf .work_du.tolist(), index= listdf.index)
    listdf[['safety1','safety2','safety3']] = pd.DataFrame(listdf .safety.tolist(), index= listdf.index)
    listdf[['Water_vapour1','Water_vapour2','Water_vapour3']] = pd.DataFrame(listdf .Water_vapour.tolist(), index= listdf.i
    listdf[['Fir_Resis1','Fir_Resis2','Fir_Resis3']] = pd.DataFrame(listdf .Fir_Resis.tolist(), index= listdf.index)
    listdf=listdf.iloc[: , -12:]
    normal = pd.concat([listdf, kr],axis=1)
    nworkdu=normal.iloc[:,:3].div(normal.S_powork_du, axis=0)
    nsafe=normal.iloc[:,3:6].div(normal.S_posafety, axis=0)
    nwater=normal.iloc[:,6:9].div(normal.S_poWater_vapour, axis=0)
    nfire=normal.iloc[:,9:12].div(normal.S_poFir_Resisr, axis=0)
```



```
    nm = pd.concat([nworkdu, nsafe,nwater,nfire],axis=1)
    nm['M_ID'] = nm.index
    nm_merg = pd.merge(obj, nm, on=['M_ID'])

    #display(nm_merg)

    nm_merg['qLCC1'] = nm_merg['qLCC']
    nm_merg['qLCC2'] = nm_merg['qLCC']
    nm_merg['qIn_C1'] = nm_merg['qIn_C']
    nm_merg['qIn_C2'] = nm_merg['qIn_C']
    nm_merg['qLCA1'] = nm_merg['qLCA']
    nm_merg['qLCA2'] = nm_merg['qLCA']

    nm_merg=nm_merg[['M_ID','THK','th2','THsum','work_du1','work_du2','work_du3',\
                   'safety1','safety2','safety3','Water_vapour1','Water_vapour2',\
                   'Water_vapour3','Fir_Resis1','Fir_Resis2','Fir_Resis3','qLCC','qLCC1','qLCC2',\
                     'qIn_C','qIn_C1','qIn_C2','qLCA','qLCA1','qLCA2']]

    flat = [x for l in MD_cw_avg for x in l]
    print(flat)

    nm_merg.loc[:,['work_du1','work_du2','work_du3',\
                   'safety1','safety2','safety3','Water_vapour1','Water_vapour2',\
                   'Water_vapour3','Fir_Resis1','Fir_Resis2','Fir_Resis3','qLCC','qLCC1','qLCC2',\
                     'qIn_C','qIn_C1','qIn_C2','qLCA','qLCA1','qLCA2']] *= flat

    #Single Average Global Varaibles
    nm_merg["Swork_du"] =(nm_merg["work_du1"]+ 2*nm_merg["work_du2"]+ nm_merg["work_du3"])/4
    nm_merg["Ssafety"] =(nm_merg["safety1"]+ 2*nm_merg["safety2"]+ nm_merg["safety3"])/4
    nm_merg["SWater_vapour"] =(nm_merg["Water_vapour1"]+ 2*nm_merg["Water_vapour2"]+ nm_merg["Water_vapour3"])/4
    nm_merg["SFir_Resis"] =(nm_merg["Fir_Resis1"]+ 2*nm_merg["Fir_Resis2"]+ nm_merg["Fir_Resis3"])/4
    nm_merg["SLCC"] =(nm_merg["qLCC"]+ 2*nm_merg["qLCC1"]+ nm_merg["qLCC2"])/4
    nm_merg["SIn_C"] =(nm_merg["qIn_C"]+ 2*nm_merg["qIn_C1"]+ nm_merg["qIn_C2"])/4
```

```
    nm_merg["SLCA"] =(nm_merg["qLCA"]+ 2*nm_merg["qLCA1"]+ nm_merg["qLCA2"])/4

    singledfz=nm_merg.drop(['work_du1','work_du2','work_du3',\
                   'safety1','safety2','safety3','Water_vapour1','Water_vapour2',\
                   'Water_vapour3','Fir_Resis1','Fir_Resis2','Fir_Resis3','qLCC','qLCC1','qLCC2',\
                     'qIn_C','qIn_C1','qIn_C2','qLCA','qLCA1','qLCA2'],axis=1)

    #display(singledfz)

    #positive solutions:
    sign = [1, 1,1,1,-1,-1,-1]

    po_work=singledfz["Swork_du"].max()
    singledfz["D+work"]=(singledfz["Swork_du"]-po_work)**2
    po_safe=singledfz["Ssafety"].max()
    singledfz["D+safe"]=(singledfz["Ssafety"]-po_safe)**2
    po_Water=singledfz["SWater_vapour"].max()
    singledfz["D+water"]=(singledfz["SWater_vapour"]-po_Water)**2
    po_Fi=singledfz["SFir_Resis"].max()
    singledfz["D+Fire"]=(singledfz["SFir_Resis"]-po_Fi)**2
    po_LCC=singledfz["SLCC"].min()
    singledfz["D+LCC"]=(singledfz["SLCC"]-po_LCC)**2
    po_In_C=singledfz["SIn_C"].min()
    singledfz["D+Inco"]=(singledfz["SIn_C"]-po_In_C)**2
    po_LCA=singledfz["SLCA"].min()
    singledfz["D+LCA"]=(singledfz["SLCA"]-po_LCA)**2

    #Negative solutions
    ne_work=singledfz["Swork_du"].min()
    singledfz["D-work"]=(singledfz["Swork_du"]-ne_work)**2
    ne_safe=singledfz["Ssafety"].min()
    singledfz["D-safe"]=(singledfz["Ssafety"]-ne_safe)**2
    ne_Water=singledfz["SWater_vapour"].min()
    singledfz["D-water"]=(singledfz["SWater_vapour"]-ne_Water)**2
    ne_Fi=singledfz["SFir_Resis"].min()
    singledfz["D-Fire"]=(singledfz["SFir_Resis"]-ne_Fi)**2
    ne_LCC=singledfz["SLCC"].max()
    singledfz["D-LCC"]=(singledfz["SLCC"]-ne_LCC)**2
    ne_In_C=singledfz["SIn_C"].max()
    singledfz["D-Inco"]=(singledfz["SIn_C"]-ne_In_C)**2
    ne_LCA=singledfz["SLCA"].max()
    singledfz["D-LCA"]=(singledfz["SLCA"]-ne_LCA)**2

    Distan=singledfz.drop(["Swork_du","Ssafety","SWater_vapour","SFir_Resis","SLCC","SIn_C","SLCA"],axis=1)
    rslt1=singledfz[['M_ID','THK','th2','THsum',"Swork_du","Ssafety","SWater_vapour","SFir_Resis","SLCC","SIn_C","SLCA"]]
    rslt1=rslt1.set_axis(['M_ID','THK','th2','THsum','work_du','safety','Water_vapour','Fir_Resis','LCC','Initial_C','LCA']

    Po_col = ['D+work', 'D+safe', 'D+water', 'D+Fire', 'D+LCC','D+Inco', 'D+LCA' ]
    Ne_col = ['D-work', 'D-safe', 'D-water', 'D-Fire', 'D-LCC', 'D-Inco', 'D-LCA']

    Distan['Po_di']= np.sqrt(Distan[Po_col].sum(axis=1))
    Distan['Ne_di']= np.sqrt(Distan[Ne_col].sum(axis=1))
    Distan['Score']=Distan['Ne_di']/(Distan['Po_di']+Distan['Ne_di'])
    Distan
    Distan = Distan[['M_ID','THK','th2','THsum','Po_di', 'Ne_di','Score']]
    display(Distan)


    w_df= DataFrame (crispVal)
    w_df=w_df.transpose()
    w_df.rename(columns={0:'work_du',1:'safety',2:'Water_vapour',3:'Fir_Resis',4:'LCC',5:'Initial_C',6:'LCA'},index={0:'Wei
    w_dfsave=w_df.round(2)
    w_dfsave.to_csv(r'Weight_df.csv', header=True)
    #display(w_df)

    ek = w_df.append(rslt1)
    ek = ek.reindex(columns=['M_ID','THK','th2','THsum','work_du','safety','Water_vapour','Fir_Resis','LCC','Initial_C','LC
    ek = pd.merge(ek, Distan, on=['M_ID','THK','th2','THsum'])
    e = ek.rename(columns={'THK': "th1", "THsum": "THKsum"})
    #display(e)

    e["Rank"] = e["Score"].rank(ascending=False)
```

```
    e.sort_values("Score", inplace = True,ascending=False)
    e_head=e.head(10)
    e_head=e_head.round(2)
    e_head = e_head.rename_axis(None)
    e_head.to_csv(r'rank_df.csv', header=True)
    erank = e[["M_ID","th1",'th2',"THKsum","Rank"]]
    display(erank)

    Best_A=e.iloc[0,0:4]
    Best_A=Best_A.values.tolist()
    Best_A=str(Best_A[0]) + "(" + str(Best_A[1])+"+"+str(Best_A[2])+")"+str(Best_A[3])

    X= ' '.join([str(elem) for elem in Best_A])
    print(X)

    orlck = pd.merge(erank, orlc, how="outer", on=['M_ID',"th1",'th2',"THKsum"])
    orlck.sort_values("Rank", inplace = True,ascending=True)
    orlck_head=orlck.head(10)
    display(orlck)

    class DisplayResults():

        def __init__(self):

            self.root = Tk()
            self.root.title("Results")
            self.root.minsize(width=1000, height=300)
            self.root.rowconfigure(0, weight=1)
            self.root.columnconfigure(1, weight=1)

            self.createWidgets()

        def createWidgets(self):

            fr_plot = LabelFrame(self.root,text="LCA and LCC Optimization")
            fr_plot.grid(rowspan=2,row=2, column=0, sticky=S)
            fr_plot0 = LabelFrame(self.root,text="Topsis Weight")
            fr_plot0.grid(row=2, column=1, sticky=SW)
            fr_plot1 = LabelFrame(self.root,text="Topsis Ranking with Defuzzified Weighted Normalized Decision Matrix")
            fr_plot1.grid(rowspan=3,row=3, column=1, sticky=N)
            fr_plot2 = LabelFrame(self.root,text="Final Recommandation")
            fr_plot2.grid(columnspan=3,row=0, column=0, sticky=N)

            arr = orlck.to_numpy()
            x = arr[:, 0]
```



```
            y = arr[:, 1]


            figure1 = Figure(figsize=(6,6))
            ax = figure1.add_subplot(111)
            colors = {'A':'red', 'B':'green', 'C':'blue', 'D':'yellow'}
            orlck.plot(ax=ax,kind='scatter',x='LCA', y='LCC',s=120,alpha=0.1)
            for idx, row in orlck_head.iterrows():
                ax.annotate(row['Rank'], (row['LCA'], row['LCC']) )

            grouped = orlck.groupby('M_ID')
            for key, group in grouped:
                group.plot(ax=ax, kind='line', x='LCA', y='LCC', label=key, marker='h', color=colors[key],alpha=0.5)


            canvas = FigureCanvasTkAgg(figure1,fr_plot)
            canvas.get_tk_widget().grid(row=1, column=0)

            # open file
            with open("Weight_df.csv", newline = "") as file:
               reader = csv.reader(file)
               # r and c tell us where to grid the labels
               r = 0
               for col in reader:
                  c = 0
                  for row in col:
                     label = tk.Label(fr_plot0, width = 10, height = 2,text = row,relief = tk.RIDGE)
```

```
                     label.grid(row = r, column = c)
                     c += 1
                  r += 1

            # open file
            with open("rank_df.csv", newline = "") as file:
               reader = csv.reader(file)
               # r and c tell us where to grid the labels
               r = 0
               for col in reader:
                  c = 0
                  for row in col:
                     label = tk.Label(fr_plot1, width = 10, height = 2,text = row,relief = tk.RIDGE)
                     label.grid(row = r, column = c)
                     c += 1
                  r += 1

            f_rst = Label(fr_plot2,text="", background='pale violet red', font=("Helvetica", 18))
            f_rst.grid(row=0,column=10)
            f_rst.configure(text="Best alternative material and thickness is "+X)

    gui = DisplayResults()

root.title("Material Selection")
root.geometry("1220x700")
root.mainloop()

```