



ISSN: 2785-2997

Journal of Human, Earth, and Future

Vol. 7, No. 2, June, 2026



Optimizing Building's Energy Efficiency Analysis Using Multi-Objective Particle Swarm Optimization (MOPSO)

Tri Basuki Kurniawan ^{1*}, Deshinta Arrova Dewi ^{2, 3}, Mashal Alqudah ⁴,
Shahreen Kasim ⁵

¹ Magister of Information Technology, Postgraduate Program, Universitas Bina Darma, Palembang, Indonesia.

² Faculty of Data Science and Information Technology, INTI International University, Nilai 1800, Negeri Sembilan, Malaysia.

³ Faculty of Engineering Technology, Shinawatra University, Pathum Thani, Malaysia.

⁴ Department of Computer Information Science, Higher Colleges of Technology, Sharjah 25026, United Arab Emirates.

⁵ Faculty of Computer Sciences and Information Technology, Universiti Tun Hussain Onn Malaysia, Parit Raja 86400, Johor, Malaysia.

Received 25 May 2025; Revised 21 May 2026; Accepted 26 May 2026; Published 01 June 2026

Abstract

We successfully built a model and application that was constructed using the Revit API to help the building designers conduct the energy analysis by modifying the material of the elements, like walls, windows, roofs, and others. This work has addressed the necessity of computerized design work for energy buildings, which was too time-consuming and complex. With the built-in Revit API, the building designers can quickly calculate the energy usage in a building as early as possible from the design stage. However, there was a drawback in our previous work: the building designers selected parameters of materials using intuition and manual processing. In this study, we enhance the capability of previous work with intelligent automation to do selection, analysis, and optimization based on several factors, like the low value of Overall Thermal Transfer Value (OTTV) and Roof Thermal Transfer Value (RTTV) and the cost of materials. These three factors are defined as the multi-objective functions of this study. We identified eight dimensions generated from the list of materials used in a building, which led us to exploit particle swarm optimization (PSO) to find the optimum solutions for the energy analysis. Here, we use the built model to generate parameters and a list of materials from the library, and then we identify six items for OTTV and two items for RTTV as dimensions to run the multi-objective particle swarm optimization (MOPSO). The solutions are presented using the Pareto Front, allowing the building designer to focus on the set of efficient choices, including the tradeoffs within the solutions. We compared the performance of the enhanced approach with previous work, yielding major improvements of 52.43% for OTTV and 44.51% for RTTV, resulting in a 23.82% improvement.

Keywords: Multi-Objective Optimization; Energy Efficiency Building; Sustainable Buildings; BIM; Decision-Making; MOPSO; OTTV; RTTV; Material-Cost; Sustainable Buildings; Process Innovation.

1. Introduction

Optimizing the efficiency of the energy building always involves a multifaceted and complex process, such as designing, managing, and fine-tuning the system, aiming to reduce energy consumption while maintaining comfort and

* Corresponding author: tribasukikurniawan@binadarma.ac.id

 <https://doi.org/10.28991/HEF-2026-07-02-02>

➤ This is an open access article under the CC-BY license (<https://creativecommons.org/licenses/by/4.0/>).

© Authors retain all copyrights.

operational performance. Various disciplines are integrated in this view, such as architecture, engineering, data analytics, and smart technologies. The core objectives are to minimize the energy waste from heating, ventilation, air conditioning (HVAC), electrical systems, and lightning without compromising the system functionalities. The advanced building automation system (ABAS) plays an important role by continuously monitoring and adjusting energy use in a real-time manner, enabling adaptive responses to change the occupancy, energy demand, and weather conditions, which implies the system not only enhances the energy efficiency but also prolongs the equipment's lifespan and reduces the operational costs.

The difficulty in optimizing energy use arises when there are many variables to coordinate for optimal performance, such as indoor climate, daylighting, ventilation, and human interactivity. All these areas of focus must be well-connected for best results, and most of these utilize simulation and data analytics. The growing presence of Internet of Things (IoT) devices, such as sensors that offer highly granular data, allows building managers to truly utilize predictive maintenance and demand-side management strategies. Smart buildings and smart environments will become increasingly prevalent as buildings become more intelligent and responsive, but the emphasis will be on thorough energy management strategies that lead to sustainability and adherence to green building practices and standards. Ultimately, optimizing a building's energy use is not just a measuring and technology issue, but rather the creation of a smart environment that reacts dynamically to internal and external influences.

On the other hand, many revisions of the Multi-Objective Particle Swarm Optimization technique, MOPSO, have been made, all aimed at improving its effectiveness when solving sophisticated multi-objective optimization problems. As a metaheuristic algorithm, MOPSO has gained quite some popularity because of its ease of use, low cost, and high effectiveness. It is a revision of the old Particle Swarm Optimization PSO, which is based on the collective movement of birds and fish. MOPSO's main goal is to solve problems with several conflicting objectives by providing a rich set of optimal solutions, which are known as the Pareto-optimal set. The implementation given in this case is straightforward and does not require extensive resources, as well as efficient and compact. Thus, it can be used for multi-objective optimization functions.

MOPSO addresses complex problems involving multiple, often conflicting objectives by providing a comprehensive set of optimal solutions known as the Pareto-optimal set. Due to its efficiency and low computational resource requirements, MOPSO can be readily integrated into Building Information Modeling (BIM) environments, where multiple design parameters such as cost, energy efficiency, structural integrity, and space utilization must be considered simultaneously. The integration of MOPSO into BIM workflows enables more effective optimization of architectural and engineering designs, supports better-informed decision-making through data analysis, and generates solutions that satisfy a wide range of project requirements and stakeholder expectations.

Numerous studies have focused on achieving sustainable building design through both passive and active strategies [1, 2]. The advent of Building Information Modeling (BIM) technology has enabled sophisticated digital building modeling with precise geometric and informational representations required throughout various project stages, facilitating the development and refinement of the desired building structure.

The architecture, engineering, and construction (AEC) industries benefit significantly from BIM, as reported by numerous researchers. These benefits include a unified data environment, streamlined design workflows, accurate cost estimation, reduced time expenditures, and other advantages [3, 4]. In addition, BIM provides various analytical capabilities that support design decision-making (DM) processes [5, 6].

Several investigations have been conducted to demonstrate how adopting sustainable design principles early in the BIM process improves the efficiency and effectiveness of the project delivery process [7, 8]. It was also noted that further research is needed to explore various design options to improve the evaluation of sustainability [9, 10].

Although BIM data can be gathered via computational algorithms for numerous sustainability assessments [11, 12], it is evident that the DM process for sustainable building design, particularly in the very preliminary design phases, can be aided by BIM. Based on the literature review, it is clear that this particular sustainable design increment, Arianism, is a major contribution of BIM, as reported in [10, 13].

Previously Kumiawan et al. [14] built an application and a model that integrated with the Revit API so that the building designers could do energy analyses by modifying the construction components like windows, walls, and roofs. This allows building designers to conduct energy calculations during the design phase. On the other hand, the new DM design for sustainable buildings hinges on many heterogeneous analysis for different construction systems, like plumbing, power, and lighting systems. The model also examines building features such as landscaping, parking, and the generation of renewable energy. Many studies have underscored that the new design or pre-construction phases of a building provide the most sustainable development opportunities. In contrast, the studies suggest that the DM design for building sustainably has been practiced in architecture, engineering, and construction (AEC) during the later design phases.

Inside this scope, implementing Multi-Objective Particle Swarm Optimization (MOPSO) for assessing a building's energy efficiency provides a comprehensive and smart approach for evaluating the balance between thermal efficiency and construction cost. Through the minimization of OTTV, RTTV, and material cost simultaneously, MOPSO achieves the best possible design within project constraints and energy benchmarks, which enhances the economic viability of the project. This method generates numerous, diverse, and Pareto-optimal solutions, which improves the decision-making strategy in the preliminary design phases, thus offering a broader range of affordable and customized options to the project, considering specific objectives set by the designers' team. Moreover, more straightforward evaluations of energetic efficiency using real-time adaptive calculations could greatly advance the development of the building industry toward sustainable designs if MOPSO were incorporated into Building Information Modeling (BIM) environments.

To determine the optimal design energy-wise, an extensive sensitivity analysis was performed for the key parameters of the algorithm to confirm the flexibility and robustness of the MOPSO approach. The considered parameters include their cognitive and social factors, which were both set to 1.4. Continued by inertia eight in the range of 0.9-0.4, with random values being monitored as well, setting 0-1. The next parameters include 20 for the number of particles and 400 as a maximum iteration count. These parameters, as well as the control parameters of the MOPSO, were integrated within the sensitivity analysis. For each parameter of MOPSO efficiency, the algorithm was tested in a statistical framework to compute the sensitivity and determine the influence of each parameter. Such insights obtained from the sensitivity analysis are important in determining the best configurations of parameters to ensure the effectiveness of the algorithm in providing precise and energy-efficient building design solutions.

Considering the aforementioned data, the purpose of this research is to advance a technique for improving the analysis of energy-efficient buildings utilizing Multiple Objectives Particle Swarm Optimization (MOPSO). We enable an intelligent strategy to undertake selection and incorporate multiple objectives optimization into the approach since there are three components engaged in the system, namely the low value of Overall Thermal Transfer Value (OTTV), Roof Thermal Transfer Value (RTTV), and cost of materials. According to this theory, Multiple Objectives Particle Swarm Optimization (MOPSO) has proven to be extremely efficient and successful in handling complex multi-objective problems where traditional optimization technologies fall short.

1.1. Energy Efficiency Building

Due to the heavy use of natural resources over the last two decades, the predominant fossil resources in the world are on the verge of depletion. Consequently, concerns are increasing among governments, academics, policymakers, and scientists in both developed and developing countries about changes in climate conditions (i.e., global warming, ozone layer depletion, etc.); energy protection, and the adverse environmental effects [15]. The IEA raised environmental, energy, and economic prosperity concerns, which have been generally referred to as (3Es) in the current energy scenario [16]. However, the European Union (EU) is describing energy-shared targets until 2020 [17]: (a) the reduction of EU GHG emissions should be at least 20% below 1990 levels, (b) renewable energy contribution of at least 20% to the energy consumption of the EU, and (c) primary energy use would be reduced to 20% compared to the expected levels by energy efficiency measures [18].

Energy efficiency means using less energy to achieve the same function – that is, reducing wasted energy. Energy conservation offers many benefits: reducing greenhouse gas emissions, rising demand for energy imports, and reducing household and economic costs [19]. Constructors strive to improve building efficiency and then incorporate renewable energy technologies to build buildings that produce zero energy. Changes to existing buildings may also be made to reduce energy usage and costs. It can involve small steps, such as choosing LED bulbs and energy-efficient appliances, or more significant measures, such as upgrades to insulation and weathering [20].

One of the most important issues in energy-efficient buildings is how interior elements are used. The over-isolated and airtight envelopes found in highly efficient homes (and much more so in highly specialized net-zero energy homes) show that the internal environment's mode of use has a major impact on final performance [21]. A central collecting point, air conditioning, and window opening are just a few of the available controls that might considerably differ from family household to, doubling the energy usage of a single flat.

1.2. Building Information Modeling

Building Information Modeling (BIM), an emerging technology, is recognized to contribute to sustainable design, including energy efficiency. It provides an opportunity for architects and engineers to analyze, simulate, and visualize the building performance that was formerly in the design process [3, 22]. BIM has benefits for scheduling, planning, implementing, and facilities management in terms of greater flexibility and profitability. BIM helps owners, designers, contractors, and management teams to collaborate better, and visualize and execute construction work from a stakeholder perspective [23]. BIM technology is, therefore, attracting considerable attention from practitioners. In light of change, the use of information and communication technology (ICT) could help resolve some of the current challenges that have been encountered.

One of the most popular BIMs is Autodesk Revit or Revit API (Application Programming Interface), a creative technology modeling program designed for architects, landscape architects, architects, and engineers, mechanical, electrical, and plumbing engineers, consultants, and designers. The original software, developed by Charles River Software, was founded in 1997, purchased by Autodesk in 2002, and was renamed Revit Technology Company in 2000 [24]. By directly manual family creation [25] or commercial plugins [26] such as Scan-to-BIM and Leica CloudWorx, the Revit API has been widely used to support the reality-based parametric modeling process. The Revit API incorporates the parametric modeling and user features of the BIM. The interface programming methods give designers the ability to interactively design and manipulate Revit components using computational logic and algorithms. Revit can provide the UI interface, acts as a significant forum and database for display, and parametrically represents the person and constructs the relationship automatically. The software will reduce manual operation and carry out automation and batch processing, considering different functions [27]. Also, the current algorithms and libraries can be implemented and performed directly. They can, therefore, simultaneously automate the segmentation of elements and the parametric representation procedure in the BIM setting, employing specific functions.

1.3. Multi-Objective Optimization

Multi-objective optimization problems consist of several objectives that need to be addressed at the same time. For certain implementations, these problems emerged, where two or more objective functions, often overlapping and incommensurable, need to be minimized simultaneously. The multifaceted nature of these issues, leading to the concept of optimality in Pareto, means that the optimal solution should be defined [28].

In contrast to the single-objective optimization case, the multi-objective problems are characterized by trade-offs. Therefore, there are a multitude of Pareto optimal solutions that correspond to the different settings of the multi-objective problem being investigated. For example, different Pareto optimal solutions correspond to different structural configurations of equal fitness but different properties in the optimization of shape. The need to find the most significant number of such solutions, with a good variety of similar properties, is highly desirable [29].

Multi-objective optimization is the practice of changing multiple parameters, whether constrained or unrestricted, while at the same time trying to achieve different objectives. The optimization problem is defined as follows [30]:

$$\begin{aligned} \text{minimize:} & \quad f_1(x), f_2(x), \dots, f_k(x) \\ \text{subject to:} & \quad h(x) = 0, \quad g(x) \leq 0, \quad x \in R^n \end{aligned} \quad (1)$$

where, x is an array of the parameters being optimized, n is the number of optimization parameters, $f_k(x)$ is the objective function, p is the number of objective functions being optimized, and $h(x)$ and $g(x)$ are the equality and inequality constraints respectively, that may exist for the optimization problem.

1.4. Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a stochastic optimization technique developed in 1995 by Kennedy & Eberhart [31]. By simulating the bird flocking social behavior, this method finds an optimal solution. The PSO algorithm is comprised of a group of individuals called "particles." Every particle is a possible solution to a problem in n -dimensions.

The group can effectively find a solution using the group's shared data and the specifics themselves [32]. Particles change their state by "flying" around in an n -dimensional velocity-based search space until a relatively constant state has been encountered, or until computational limitations have been exceeded.

The velocity of each particle shall be calculated using Equation 2:

$$v_i^{k+1} = \omega v_i^k + c_1 r_1 (pbest_i^k - x_i^k) + c_2 r_2 (gbest^k - x_i^k) \quad (2)$$

where, v_i^k , v_i^{k+1} , and x_i^k , are the velocity vector, modified velocity vector, and position vector of particle i at generation k , respectively $pbest_i^k$ is the best position found by particle i and one another particle. $gbest^k$ is the best position located by the particle's neighborhood or the entire swarm. The parameter ω is the inertia weight, and c_1 and c_2 are the cognitive and social coefficients, respectively. The velocity equation includes several random parameters, r_1 and r_2 , to ensure the construction space is adequately protected and to prevent entanglement in local optima.

The modified position vector, x_i^{k+1} , is obtained using Equation 3:

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (3)$$

The updating process of each particle's velocity and position in two dimensions is depicted in Figure 1 [33].

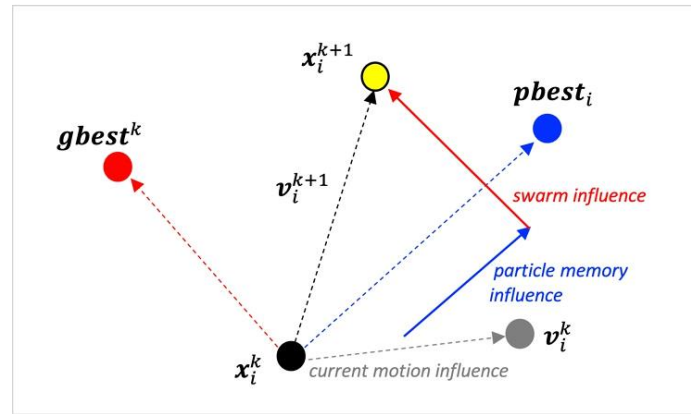


Figure 1. The velocity and position of the particle update mechanism in PSO [33]

An algorithm for finding the best PSO positioning vector using n-dimensional particles can be described as below [34]:

1. Initial positioning vector $S[n]$ and velocity vector $V[n]$ are generated by using random values, $x_1 = [x_1, x_2, \dots, x_n]$ and $v_1 = [v_1, v_2, \dots, v_n]$
2. The velocity vector v^{k+1} of a particle, i is calculated by using Equation 2.
3. The new positioning vector x^{k+1} of a particle, i is calculated by using Equation 3 and updated asynchronously.
4. If $F(x^{k+1})$ is better than the $F(pbest_i)$, the positioning vector $pbest$ is set to $pbest$. If $F(pbest_i)$ is better than $F(gbest)$, the positioning vector $F(pbest_i)$ is set to $gbest$.
5. When the predetermined number of generations is reached, or sufficiently good fitness is attained, the cycle will end. If not, move on to stage 2.

1.5. Multi-Objective Particle Swarm Optimization (MOPSO)

Several PSO studies on multi-objective problems have appeared so far, and new, advanced method variants have been created [35]. This section aims to provide a detailed review of the state-of-the-art multi-objective PSO variants. We can distinguish two fundamental approaches for designing PSO algorithms in multi-objective problems [35]. The first approach consists of algorithms that look separately at each objective function. In these methods, each particle is assessed only for one objective function at a time. The best positions' measurement is carried out in the same way as the single objective optimization case. The biggest challenge in such cases is the correct manipulation of the information coming from each objective function to guide the particles toward the optimal solutions of Pareto [36].

The second method consists of algorithms that assess all objective functions for each particle and, based on the Pareto optimality concept, generate non-dominated best positions (often referred to as leaders) that are used to guide the particles. In such methods, leadership determination is not straightforward because several non-dominated solutions in a particle's neighborhood can be found. Still, only one is typically chosen to participate in the speed update [35].

In this work, the multi-objective particle swarm optimization (MOPSO) is modified and used to solve the multi-objective optimization problem in energy-efficient buildings. This is because MOPSO is not only simple to implement, with there being few parameters to adjust, but it has a great convergence speed and is one of the successful solutions to multi-objective optimization problems. Liu et al. [37] developed multi-objective models for uneven area dynamic and static facility layout problems; therefore, a customized PSO was proposed to solve them. Kaucic [38] proposed an MOPSO algorithm incorporating an objective spatial division strategy to optimize the UAFLP multi-objective. Wilding et al. [30] design the process of a nuclear power plant system. Zhang et al. [39] suggested a multi-swarm multi-objective particle swarm optimizer (MC-MOPSO) consisting of multiple slaves and master swarms to improve MOPSO performance. Mousavi et al. [40] presented a two-pronged distribution chain supply network by presenting a modified PSO to find an optimal location for manufacturers and retailers. Despite the exception of the above, many new variants of MOPSO have been explored in recent years and are commonly used in many areas, such as cost-based feature selection [41, 42], equity portfolio management [43], multi-workshop facility layout problems [16, 44], robot path planning [45], etc.

The outline of this paper is organized as follows. Section 2 discusses some fundamental concepts on energy-efficient construction, building information modeling, multi-objective optimization, particle swarm optimization, and multi-objective particle swarm optimization. In Chapter 3, the Pareto front and the development design process are covered as methodologies. Section 4, it is details how to execute energy-efficient building Optimization Using Multi-Objective Particle Swarm Optimization and Information-Building Modeling. We provide a thorough definition of the fitness functions utilized in the experiment and conduct a sensitivity analysis. This section displays and discusses the findings after first demonstrating the approach-based methodology that was applied to the investigation.

2. Methodology

2.1. Revit-API Plugin Design

The BIM's parametric modeling and device interfaces are implemented in the Revit API (Application Programming Interface). The programming interface methods give designers the ability to design and manipulate Revit components interactively using computational logic and algorithms. Revit can provide the UI interface, act as the essential display forum and database, and parametrically represent the person and automatically construct the relationship. The program would minimize manual operation and conduct automated processing, and batch processing would consider various functions [46]. Also, the existing algorithms and libraries can be implemented and directly executed. Therefore, they can simultaneously automate the segmentation of elements and the parametric representation procedure using similar functions in the BIM environment.

2.2. Pareto Front

A Pareto front is an optimal plot with the axes as the concept goals. In this plot, the line or plane formed by the optima shows the optimal trade-off between the opposing design goals. When looking at the Pareto front, it is crucial to understand that each one of the optima may be better than any other optima for one or more objectives, but not all of them. The Pareto front members are also referred to as "non-dominated," meaning that they are not entirely dominated by any other population member concerning all objective functions.

When comparing the performance of different multi-objective optimization methods, the metrics of success will probably depend on the problem of optimization. If the problem's solution is already somewhat or well known, the success metric is clearly to compare the method set's solution with the known Pareto front solution set [47]. There are potential weaknesses to be identified, as well. The solution set by the optimization method may be best compared to others, but it may also fail to find the true extremes of the Pareto front [48]. This metric is also subject to and dependent on the preferences of the DM. In other words, the method may not be able to find solutions to Pareto that strongly favor the specific objective that the DM is pursuing [49].

2.3. OptEEB-BIM-MOPSO Framework

The process aims to enhance the base model so that the best new model can be used for these objective functions in this experiment. First, the parameters needed to calculate the objective functions were extracted from the Revit Model and Material Library, as shown in Figure 2.

The second step is to map the price of each material in the Revit Library. Based on the list of parameters, the last step is the list of materials corresponding to their cost as input for the optimization process. The results are an excellent solution that can be a frontline solution for Pareto. This framework is called OptEEB-BIM-MOPSO, which stands for Optimization of Energy Efficiency Building based on Building Information Modeling and Multi-Objective Particle Swarm Optimization.

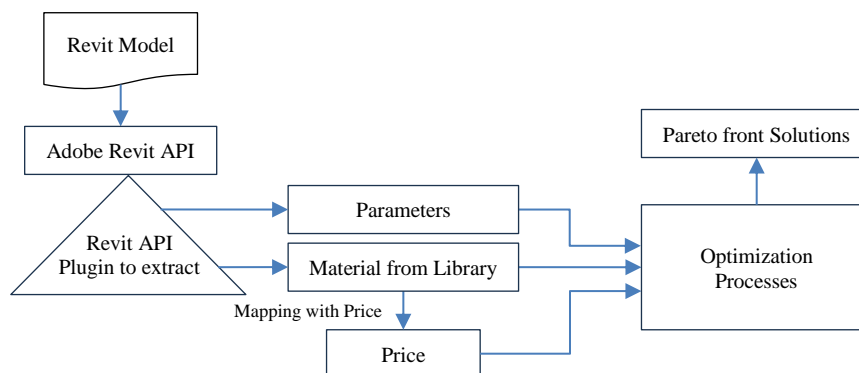


Figure 2. The framework for the optimization of energy efficiency in buildings based on MOPSO

The proposed multi-objective optimization process can be described in detail, as shown in Figures 3 and 4:

Step 1: Initialize parameters $c1, c2, r1, r2$ and ω for the particles.
 Step 2: Initialize random positions for all dimension each particle and their associated velocities.
 Step 3: Evaluate the fitness function for each particle.
 Step 4: Check criterion termination based on number of iterations.
 Step 5: Update velocities and position based using eq. (2) and (3).
 Step 6: Update local best ($pbest$), which compares the current value of fitness function with the previous best value of the particles.
 Step 7: Update global best ($gbest$), which determines the current global minimum fitness value among the current positions of the particles.
 Step 8: The process will continue to step 3, till end condition when maximum number of iterations is reached.

Figure 3. The optimization pseudocode

Pseudocode shows that the optimization process started with the initialization parameters $c1, c2, r1, r2$, and inertia weight, ω . Then follow the initialization of the random position for all dimensions of each particle and its velocity. In Step 3, fitness functions are evaluated based on OTTV, RTTV formulas, and the cost of each material's Revit model. Next, the stop criterion is checked; if it meets the criterion or reaches the end of the number of iterations, the algorithm stops. Step 5 will update the new position and velocity used in equations 2 and 3, which we discussed earlier. In steps 6 and 7, $pbest$ and $gbest$ are updated, respectively. The last step is to continue Step 3 until the maximum number of iterations is reached.

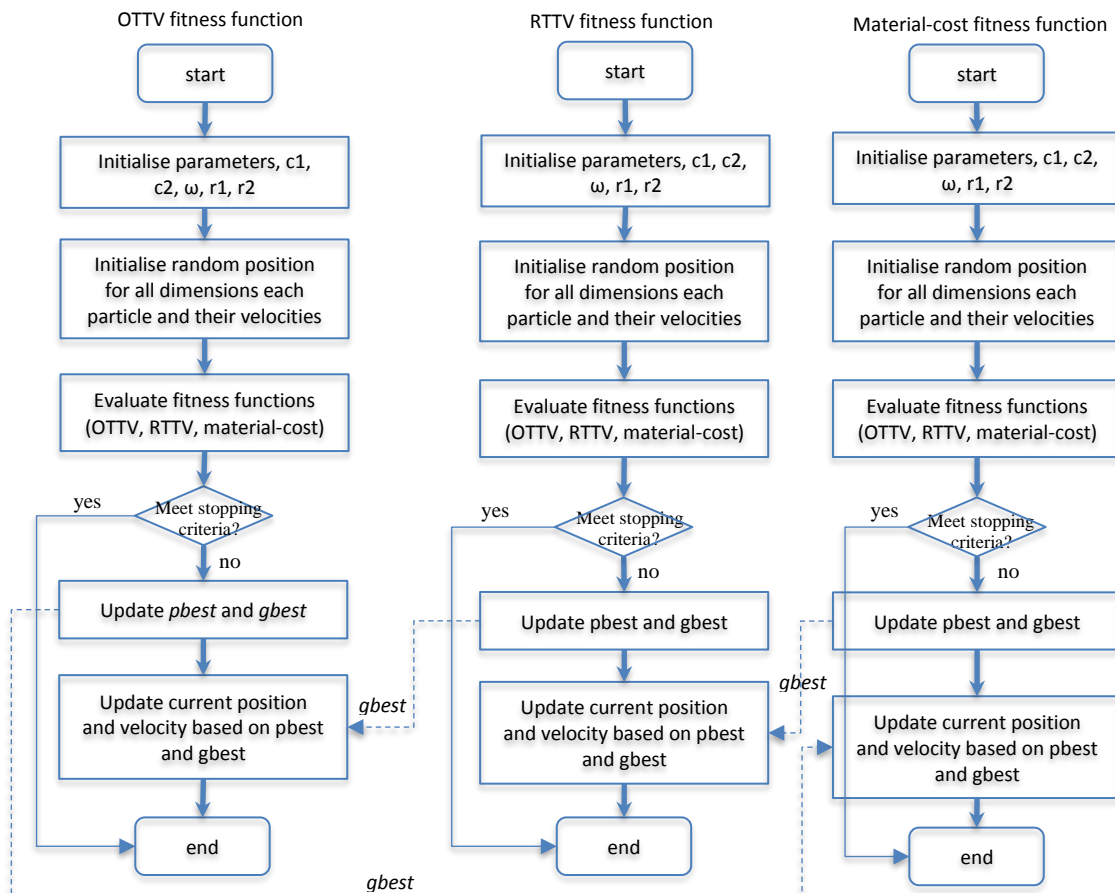


Figure 4. The proposed multi-objective optimization flowchart

Likewise, with pseudocode, the flowchart shows that three (3) objective functions are processed in different particle swarms for each objective function. Each swarm will process the same procedure as the pseudocode, but it has a different one if the $gbest$ of each swarm is used to update the other swarms' position and velocity.

RTTV swarm movement will be used in the OTTV swarm, and the $gbest$ material-cost swarm movement will be used in the RTTV swarm. In the end, the $gbest$ OTTV swarm movement will be used in a material-cost swarm. Based on this scenario, in addition to its objective function, each swarm needs to learn and consider other objective functions. The solution of one swarm will, therefore, also have proper criteria or characteristics for different objectives. Multi-objective functions will be addressed in this mechanism.

3. Implementation of OptEEB-BIM-MOPSO

3.1. Fitness Function

In the Malaysian Standard, the OTTV code has been considered since 2001, particularly for commercial conditioning buildings [50]. The main objective of the Malaysian Energy Consumption Codes (also developed as the Malaysian Standard, MS) is to promote quality, standardization, and accreditation to improve the national economy.

There are two main Malaysian energy efficiency building requirements, MS2860 for energy efficiency and sustainable energy use in residential buildings [51] and MS1525 for energy efficiency and renewable energy use in non-residential buildings [52].

In that Standard, the design parameter specifies the solar thermal load transmitted through the building envelope, Total Thermal Transfer Value (OTTV), excluding the thermal roofing (Roof Thermal Transfer Value, RTTV). The MS1525 in clause 5.2 shows that the value is less than 50W/m². On the other hand, the roof's thermal (Roof Thermal Transfer Value, RTTV), based on the same Standard, shows that the value should be less than 25W/m². The OTTV and RTTV calculations aim at achieving the design of the building envelope to reduce heat gain through the building envelope and hence reduce the cooling load of the air-conditioning system.

Equations 4 and 5 show the OTTV and RTTV formulas:

$$OTTV = 15\alpha(1 - WWR) \cdot U_w + 6(WWR)U_f + 194 (OF)(WWR)(SC) \tag{4}$$

$$RTTV = \frac{A_r U_r T D_{eq} + A_s U_s \Delta T + A_s (SC)(SF)}{A_0} \tag{5}$$

The basic definition of the parameters in Equations 1 and 2 is summarized in Tables 1 and 2, respectively.

Table 1. The basic definition of OTTV formulas

Notation	Description
α	is the opaque wall solar absorptive
WWR	is the exterior wall to windows the gross ratio
U_w	is opaque wall thermal transmittance (W/m ² K)
U_f	is the fenestration device thermal transmittance (window element) (W/m ² K)
OF	is the factor of solar orientation
SC	is the system fenestration coefficient for shading

Table 2. The basic definition of RTTV formulas

Notation	Description
A_r	is the opaque roof layer (m ²)
U_r	is the opaque roof layer thermal transmittance (W/m ² K)
$T D_{eq}$	is the equivalent temperature difference (k)
A_s	is the roof skylight layer (m ²)
U_s	is the thermal transmittance of the skylight layer (W/m ²)
ΔT	is the temperature difference between exterior and interior architecture (5K)
SC	is the skylight shading coefficient
SF	is the solar factor (W/m ²)
A_0	is the gross roof layer (m ²); $A_0=A_r+A_s$

Based on the problem definition above, this research will involve three objective functions, OTTV, RTTV, and material costs which are included in the OTTV and RTTV calculations, such as wall material, windows, window glass and roofing material.

That objective function can be formulated, as shown in Equation 6:

$$\begin{aligned} & \text{minimise } f(eeb) = \sum w_i f_i \\ & \text{subjected to } OTTV \leq 50 \text{ W/m}^2 \text{ and } RTTV \leq 25 \text{ W/m}^2 \end{aligned} \tag{6}$$

Subject to OTTV less than 50 W/m² and RTTV less than 25 W/m² constraints, where $f(eeb)$ is the minimizing function for energy efficiency building (eeb), f_i is the objective function for each $i \in \{OTTV, RTTV, material-cost\}$, and ω_i is the weight for each f_i . Since in this study, we use the Pareto front to get the multi-objective solutions, ω_i for each f_i is set to 1.

The next step is to define the PSO modeling particle. Since the PSO consists of a few particles in the swarm, each particle should be described as a fitness function component. Based on Equation 6, three (3) fitness functions are used in this study, namely OTTV, RTTV, and material-cost. The OTTV, RTTV, and material-cost components are shown in Table 3 and displayed in Figure 5.

Table 3. Component of PSO Particle dimensions

#	Objective Function	Component Item
1	OTTV	The material of wall's paint (for solar absorption value)
2		The material of wall (for heat transfer coefficient value)
3		The material of window (for heat transfer coefficient value)
4		Height of window
5		Width of window
6		The material glass of window (for shading coefficient of sub-system 1 value)
7	RTTV	The material of Opaque Roof (for layer thermal transmittance value, roof thickness, the weight of roof and roof's density value)
8		The material of Skylight Roof (for layer thermal transmittance and skylight_shading coefficient value)

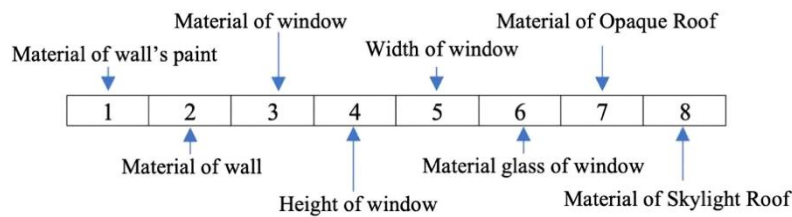


Figure 5. Component of PSO Particle dimensions illustration

For each dimension, there is an integer value representing the material number extracted from the Revit Material Library for each element of the wall, window, glass window, opaque roof, and skylight roof. Additional data is the price for each element material used to calculate the material-cost fitness function.

3.2. Control PSO Parameters

The original version of PSO has specific parameters, called the control parameters, which impact the algorithm's overall search capabilities. These control parameters, i.e., the acceleration coefficients, the speed clamping limit, the swarm size, and the maximum number of iterations [40, 53]:

Acceleration coefficient: When modified, the terms c_1 and c_2 could lead to particles flying to undesirable regions; hence, the limit of the acceleration coefficients has been set as 2 since the PSO method was built [32, 54].

This parameter may, however, be counterproductive to the overall search for an optimal solution, as these parameters are crucial to monitoring the algorithm's performance and effectiveness, as shown in Equations 7 and 8:

$$c_1^k = c_{1,min} + (c_{1,max} - c_{1,min}) \left(\frac{K_{max} - k}{K_{max}} \right) \tag{7}$$

$$c_2^k = c_{2,max} - (c_{2,max} - c_{2,min}) \left(\frac{K_{max} - k}{K_{max}} \right) \tag{8}$$

where, c_1^k and c_2^k are the cognitive and social parameters are respectively at iteration, $c_{1,max}$ and $c_{2,max}$ are the maximum cognitive and social parameters, respectively. $c_{1,min}$ and $c_{2,min}$ are the minimum cognitive and social parameters, respectively, k is the current iteration, and K_{max} is the maximum number of iterations.

Velocity Clamping limit: To clamp the maximum increase in the velocity size, the velocities of the particles are bound by $(-V_{max}, V_{max})$:

$$V_{max} = \delta \times (X_{max} - X_{min}) \tag{9}$$

where, δ is a constant clamping set between (0.1,~1.0), and X_{max} and X_{min} are the maximum and minimum value of the design variables [55].

Swarm Size: Swarm scale, N , plays a significant role in algorithm efficiency. This property refers to the number of particles it explicitly affects the swarm's diversity in the hyperspace.

Maximum number of iterations: Maximum number of iterations, K_{max} , refers to the number of iterations the PSO algorithm goes through until it is finished.

Inertia weight strategies: Introducing the weight of inertia helped promote better particle search by balancing global search and local search [56]. The inertia weight, ω , as shown in Equation 2, can be calculated as follows:

$$\omega^k = \omega_{max} - \left(\frac{\omega_{max} - \omega_{min}}{K_{max}} \right) k \tag{10}$$

Based on the PSO control parameter, Table 4 shows the parameters used in this experiment.

Table 4. The value of PSO control parameters

Parameters	Value
Cognitive factor, c_1	1.4
Social factor, c_2	1.4
Inertia weight, ω	0.9 ~ 0.4
Random values: r_1 and r_2	[0, 1]
No. of particles	20
Max iterations, K_{max}	400

4. Implementation of OptEEB-BIM-MOPSO

The Revit API plugins were generated based on their basic prototype, and the simulation -related parameters were extracted from the BIM model [57]. The development of the Revit API plugin is exploited in this study, based on C# programming. Microsoft's Visual Studio 2019 tool was selected as an Integrated Development Environment (IDE).

4.1. Experiment Setup

The experiment was developed using C# programming based on the Revit API plugin, and the simulations were run on an Intel Core i7 with 16 GB of RAM. The model used in this experiment is shown in Figure 6.



Figure 6. The Revit model which used in this experiment

It is possible to insert all types of walls, windows, and roofs into the project as an initial population using the Revit Material Library. From the wall and window of the Revit library, all required information, such as size, type, area, R-value, U-value, α , OF, SC, WWR, and even unit cost and total cost, can be extracted to run the optimization process.

4.2. Results and Discussion

After selecting the types of walls, windows, and roofs from the library, the optimization plugin was run to generate the solutions formed by the Pareto front optima set (blue-colored dots). A 3D graph with OTTV, RTTV, and material cost was plotted with matplotlib from the Python Library, as shown in Figure 6.

The user can then choose a solution from the Pareto front optima set as an optimization result. Afterward, a report was generated to summarize the optimization result solution for each orientation and the overall reduction of OTTV, RTTV, and material costs, as shown in Table 5. It shows that the optimization process was able to reduce the OTTV value from 42.72 W/m² to 20.32 W/m² (which was 52.43% of the reduction). The optimization also reduces the RTTV from 8.02 W/m² to 4.45 W/m² (which was 44.51% of the reduction) and also reduces the material cost to 23.82%, which was from RM250,156.05 to RM190,560.45.

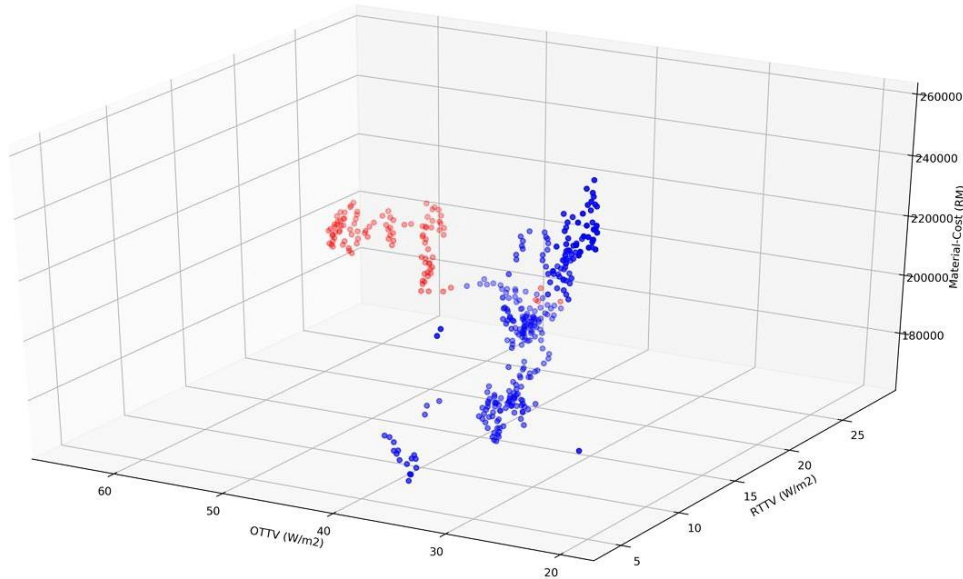


Figure 6. 3D plotting of graph results

Table 5. Comparison original model with the optimization result model

Elevation	Based Model			Optimization Result Model		
	OTTV	RTTV	Material-Cost	OTTV	RTTV	Material-Cost
North	31.18		47,529.95	14.65		36,206.49
East	61.03		75,046.51	28.68		57,168.14
South	44.09		70,043.70	20.72		53,356.93
West	36.39		57,535.89	17.11		43,828.90
Total	42.72	8.02	250,156.05	20.32	4.45	190,560.45
%				52.43	44.51	23.82

Next, to analyze the control parameters' performance, *gbest* value for each iteration is plotted as a convergence curve to show each swarm's performance in the optimization process, as shown in Figures 7 and 8 for OTTV-RTTV and Material-cost swarm, respectively.

The results show the number of objectives (*gbest*) values achieved using the MOPSO algorithm between the first iteration and the 400th iteration for each swarm. Figure 7 shows that only in 210-th of the iteration, RTTV already has the convergence *gbest* value, but for OTTV, until near the end of the maximum iteration, the swarm still has a smaller *gbest* value. The figure also illustrates clearly that RTTV's best-plummeting values range from 16 to 4 at 210th of the iteration to the end of the maximum iteration.

In the material-cost swarm shown in Figure 8, the convergence value of *gbest* is obtained at the 275th iteration at 190k. Similarly, with the RTTV swarm, the Material-cost swarm also shows the same behavior for the *gbest* values at 275-th of the iteration. It decreases moderately from 170k to 167k until the end of the maximum iteration.

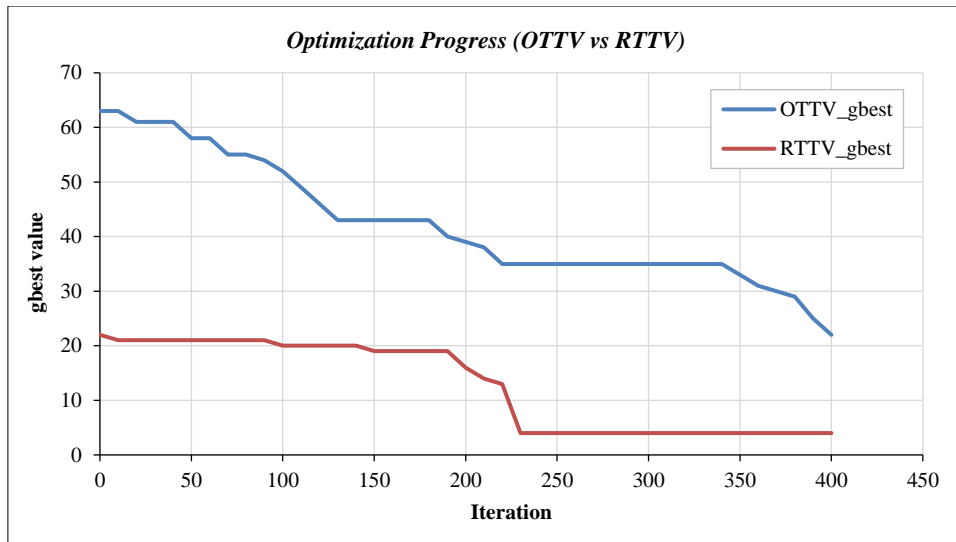


Figure 7. The convergence curve of TTV and RTTV swarms

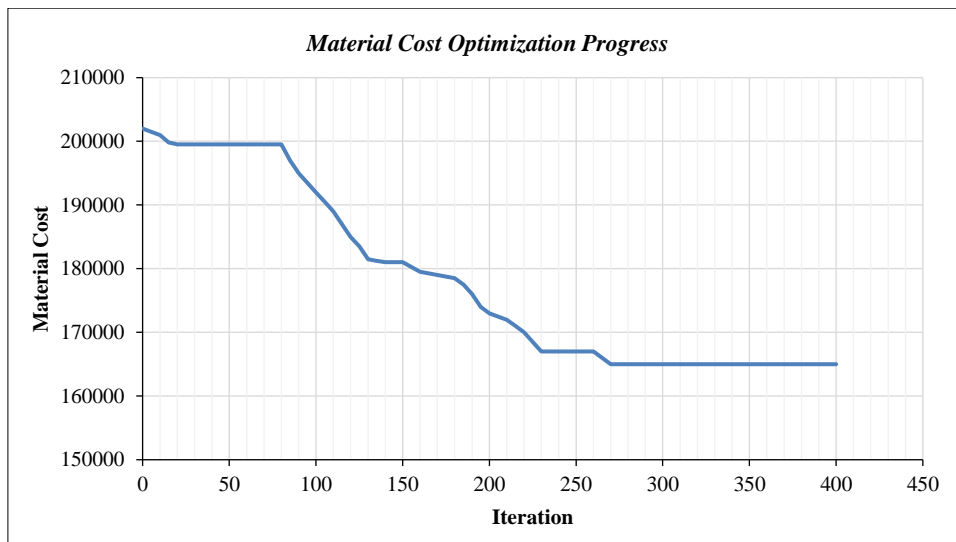


Figure 8. The convergence curve of the Material-cost swarm

4.3. Sensitivity Analysis Results

Sensitivity analysis included four MOPSO control parameters. The investigation determined the sensitivity of the MOPSO efficiency to variations in each parameter using a statistical method. For each parameter set, the speed-reducer problem is solved by MOPSO 100 times. The best global solution is stored at the end of each run and ranked after 100 runs to determine the best and worst solutions, corresponding to their best parameters. The mean value and the standard deviation are calculated by Equations 11 and 12, respectively:

$$f(x_m) = \frac{1}{100} \sum_{i=1}^{100} f(x_G^{(i)}) \tag{11}$$

$$SD = \sqrt{\frac{\sum_{i=1}^{100} [f(x_G^{(i)}) - f(x_m)]^2}{100 - 1}} \tag{12}$$

where, $f(x_m)$ is the mean value obtained across 100 runs, $f(x_G^{(i)})$ is the best global solution obtained at run i , and SD is the standard deviation.

4.3.1. Sensitivity Analysis Results–Maximum Number of Iterations (K_{max})

Using ten different numbers of K_{max} , from 100 to 1000, and with other parameters using vanilla's parameters, as shown in Table 4, each set of parameters is run, and the result is shown in Table 6. For each swarm, the best and worst $gbest$ values are obtained, and mean and standard deviation values are calculated.

The figure shows the best result for OTTV is 19.18 when using K_{max} is set to 1000, but at K_{max} is set to 600 and 500, it also gets 19.87 and 19.88. The best value for RTTV is 4.01, if K_{max} is set to 700 and 1000. The RTTV values of the best display nearly identical values are 4. In conclusion, the K_{max} 's parameter is not sufficiently sensitive to the RTTV swarm.

Table 6. Sensitivity analysis result – K_{max} with other parameters used as in Table 4

Max _x	OTTV				RTTV				Material-cost			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD	Best	Worst	Mean	STD
100	20.44	66.78	43.61	12.77	4.44	30.89	17.67	6.7	189,209.92	278,922.46	234,066.19	23,424.35
200	20.32	65.78	43.05	12.15	4.22	29.19	16.71	5.66	177,922.81	268,792.92	223,357.87	24,242.87
300	20.18	66.78	43.48	12.95	4.89	20.09	16.99	5.11	189,289.21	278,792.12	234,040.67	23,276.11
400	20.32	64.43	42.38	10.63	4.45	28.40	16.43	5.68	166,543.18	256,986.16	211,764.67	22,959.27
500	19.88	61.71	40.80	9.58	4.11	29.89	17	6.23	166,929.92	267,298.92	217,114.42	30,959.6
600	19.87	62.71	41.29	10.29	4.21	28.81	16.51	5.39	178,782.92	289,892.88	234,337.9	38,554.61
700	20.16	61.87	41.02	9.49	4.01	29.36	16.69	5.93	178,292.18	278,789.22	228,540.7	31,050.14
800	20.07	61.88	40.98	9.56	4.22	28.91	16.57	5.46	162,882.82	256,382.92	2096,32.87	26,102.55
900	21.91	67.81	44.86	12.46	4.78	29.01	16.9	5.13	158,928.29	278,393.26	2186,60.78	44,462.49
1000	19.18	68.91	44.05	15.16	4.01	28.99	16.5	5.66	158,928.29	267,983.19	2134,55.74	37,101.46

4.3.2. Sensitivity Analysis Results–Swarm size (N)

The swarm size number, N, is set at [5, 10, 20, 50, 75, 100, 125, 150, 175, and 200], and other parameters are set in Table 4. Each set of parameters runs 100 times (K_{max} is set to 100). The best and worst *gbest* values are obtained, and the mean and standard deviation values are determined for each swarm, as shown in Table 7.

Table 7. Sensitivity analysis result – N with other parameters used as in Table 4

N	OTTV				RTTV				Material-cost			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD	Best	Worst	Mean	STD
5	21.09	63.28	42.19	9.83	4.91	28.99	16.95	5.03	178,292.18	256,382.32	217,337.25	15,206.07
10	22.82	62.89	42.86	8.33	4.55	29.36	16.96	5.54	166,929.92	256,382.32	211,656.12	23,240.40
20	21.82	67.92	44.87	10.63	4.22	28.4	16.51	5.06	166,929.18	267,983.19	223,137.69	23,409.12
50	20.82	62.89	41.86	9.75	4.22	29.36	16.95	5.44	162,882.82	289,892.88	226,387.85	49,797.67
75	21.92	61.71	41.87	8.45	4.65	30.89	17.77	6.55	178,292.18	268,792.92	223,542.55	23,981.69
100	21.66	63.28	42.47	9.37	3.78	27.92	16.71	5.29	166,543.18	256,986.16	211,764.67	22,959.27
125	21.09	61.82	41.78	8.35	4.91	28.4	16.59	5.03	178,292.18	256,382.32	217,337.25	15,206.07
150	19.00	67.92	45.27	12.03	4.55	28.91	16.57	5.46	166,929.18	267,983.19	223,137.69	23,409.12
175	19.61	63.89	43.39	8.99	4.22	29.01	16.9	5.39	166,929.92	256,382.22	222,859.57	39,084.47
200	19.00	64.43	42.38	10.63	4.22	28.40	16.43	5.68	166,929.92	267,298.92	217,114.42	30,959.60

The results show that the best OTTV value is 19.61 at 175 particle numbers, and at 150 and 200 particles, the other 19 values are also extracted. On the other hand, the best RTTV value is 3.78 at 100 particles, and these results indicate that when the number of particles is decreased or increased, the best value for RTTV is increased. For this problem, we can assume that the number of particles is significant in influencing the best outcome for OTTV and RTTV swarms. For the OTTV swarm, compared with the RTTV formulation, the OTTV formulation is more complicated. To explore all the possibilities available in the OTTV search space, it needs more particles than RTTV.

4.3.3. Sensitivity Analysis Results–Inertia Weight (ω)

The fixed inertia weight values are used, and [0.1 – 1.2] are set, and other parameters are set based on Table 4. Each set of parameters runs 100 times (K_{max} is set to 100). The results are also compared with the dynamic weight inertia result, as shown in Table 4, [0.9 ~ 0.1]. The best and worst *gbest* values are obtained, and the mean and standard deviation values are calculated, as shown in Table 8 for each swarm.

Table 8. Sensitivity analysis result – ω with other parameters used as in Table 4

ω	OTTV				RTTV				Material-cost			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD	Best	Worst	Mean	STD
0.10	21.09	63.28	42.19	9.83	4.91	28.99	16.95	5.03	162,882.82	289,892.88	226,387.85	49,797.67
0.25	22.82	62.89	42.86	8.33	4.55	29.36	16.96	5.54	178,292.18	268,792.92	223,542.55	23,981.69
0.50	21.82	67.92	44.87	10.63	4.22	28.4	16.51	5.06	166,543.18	256,986.16	211,764.67	22,959.27
0.75	20.82	62.89	41.86	9.75	4.22	29.36	16.95	5.44	178,292.18	256,382.32	217,337.25	15,206.07
1.00	21.92	61.71	41.87	8.45	4.65	30.89	17.77	6.55	166,929.92	256,382.32	211,656.12	23,240.40
1.20	21.66	63.28	42.47	9.37	4.21	27.92	16.71	5.29	166,929.18	267,983.19	223,137.69	23,409.12
1.25	21.73	61.82	41.78	8.35	4.32	28.4	16.59	5.03	166,929.92	256,382.22	222,859.57	39,084.47
1.50	22.62	67.92	45.27	12.03	4.22	28.91	16.57	5.46	166,929.92	267,298.92	217,114.42	30,959.60
2.00	22.89	63.89	43.39	8.99	4.28	29.01	16.9	5.39	176,292.18	278,393.26	228,342.72	30,770.15
0.9-0.4	20.32	64.43	42.38	10.63	4.45	28.40	16.43	5.68	166,543.18	256,986.16	211,764.67	22,959.27

The results show that the best OTTV value is 20.82 at an inertia weight of 0.75, although the experiment using a dynamic inertia weight still produced a higher value than the average. The best RTTV value obtained by the swarm was 4.21 when the inertia weight was set to 1.2. Other favorable RTTV values were 4.22 at inertia weights of 0.75 and 1.5. These results indicate that a fixed inertia weight produced better RTTV performance than the dynamic inertia weight approach. It can be concluded that dynamic inertia weight is more suitable for complex objective functions, such as those used in OTTV formulation. In this scenario, the algorithm can utilize dynamic inertia weight to balance exploration and exploitation mechanisms.

4.3.4. Sensitivity Analysis Results–Acceleration Coefficient (c_1 and c_2)

The coefficient of acceleration, c_1 and c_2 , is set to [0.1, 0.25, 0.5, 1.0, 1.25, 1.50, 1.75, 2.00, 2.50, and 4.00], and other parameters are set based on Table 4. Each set of parameters runs 100 times (K_{max} is set to 100). The best and worst g_{best} values are obtained, and the mean and standard deviation values are determined, as shown in Table 9 for each swarm.

The acceleration coefficients, c_1 and c_2 , were set to values of 0.1, 0.25, 0.5, 1.0, 1.25, 1.50, 1.75, 2.00, 2.50, and 4.00, while the remaining parameters were configured according to Table 4. Each parameter set was executed 100 times (with K_{max} set to 100). The best and worst g_{best} values were recorded, and the corresponding mean and standard deviation values were calculated, as presented in Table 9 for each swarm.

Table 9. Sensitivity analysis result - c_1 and c_2 with other parameters used as in Table 4

c_1	c_2	$c_1 + c_2$	OTTV				RTTV				Material-cost			
			Best	Worst	Mean	STD	Best	Worst	Mean	STD	Best	Worst	Mean	STD
0.10	0.1	0.20	22.82	62.89	42.86	8.33	4.22	27.91	16.07	4.75	178,882.82	278,789.20	228,836.02	30,632.49
0.25	0.25	0.50	20.82	67.92	44.37	13.3	4.55	30.78	17.67	6.55	178,292.44	256,382.32	217,337.38	15,025.88
0.50	0.50	1.00	22.62	67.92	45.27	10.63	4.45	29.74	17.10	5.68	166,543.18	267,687.33	217,115.26	22,959.27
1.00	1.00	2.00	21.92	63.28	42.60	9.25	4.22	29.36	16.79	5.78	178,292.37	256,382.32	217,337.35	22,959.27
1.25	1.25	2.50	22.89	61.71	42.30	7.45	4.65	31.82	18.24	7.21	178,292.18	256,382.32	217,337.25	15,006.87
1.50	1.50	3.00	20.32	67.92	44.12	13.66	4.62	30.32	17.47	6.17	166,929.92	267,298.16	222,859.04	39,084.47
1.75	1.75	3.50	21.73	61.82	41.87	8.35	4.65	29.92	17.29	5.87	166,929.92	267,298.22	216,921.57	31,215.93
2.00	2.00	4.00	22.62	67.92	45.27	12.03	4.22	28.91	16.57	5.46	166,543.33	256,382.92	222,859.13	39,084.47
2.50	2.50	5.00	21.92	61.82	41.87	8.21	4.28	29.01	16.65	5.49	178,292.43	267,298.16	216,921.30	31,215.93
4.00	4.00	8.00	20.32	64.43	42.38	10.63	4.28	28.40	16.34	5.68	166,543.18	256,986.16	211,764.67	22,959.27

As shown in Table 9, the best OTTV value of 20.32 was obtained when c_1 and c_2 were set to 1.5 and 4.0, respectively. Another favorable OTTV result was 20.82 when c_1 and c_2 were set to 0.25. For RTTV, the best value of 4.22 was achieved when c_1 and c_2 were set to 0.10, 1.00, and 2.00. Another good RTTV result was 4.28 when c_1 and c_2 were set to 2.50 and 4.00. Based on these results, it can be concluded that higher values of c_1 and c_2 provide the algorithm with greater opportunities to explore the search space for each objective function.

5. Conclusion

Using integrated computer modeling and optimization methodologies, this study effectively demonstrates the efficacy and feasibility of Multi-Objective Particle Swarm Optimization (MOPSO) in improving the energy efficiency of building envelope design. Significant improvements were achieved across all building orientations (North, East, South, and West) by applying MOPSO to optimize key thermal performance indicators, namely Overall Thermal Transfer Value (OTTV), Roof Thermal Transfer Value (RTTV), and material cost. Compared with the base model, the results indicate substantial reductions of 52.43% in OTTV, 44.51% in RTTV, and 23.82% in material cost, demonstrating the model's ability to balance energy performance with economic feasibility.

The convergence study further demonstrates that the algorithm stabilizes after 400 iterations, highlighting both its computational efficiency and its suitability for practical design applications. A range of 400–600 iterations provides the optimal balance between computational time and solution accuracy. Furthermore, sensitivity analysis reveals that an inertia weight (ω) between 1.0 and 1.25 achieves the best trade-off between convergence speed and solution stability. Across multiple test scenarios, the conventional PSO acceleration coefficients ($c_1 = c_2 = 2.0$) were confirmed to perform reliably.

The application of MOPSO in this context supports both sustainability goals and economic constraints in modern architectural design by reducing thermal loads, enhancing passive performance, and facilitating cost-effective material selection. Owing to its robustness and adaptability, the proposed approach demonstrates strong potential as a decision-support tool for the design of energy-efficient buildings, particularly in tropical climates where thermal performance is a critical consideration.

6. Declarations

6.1. Author Contributions

Conceptualization, T.B.K. and D.A.D.; methodology, T.B.K.; software, T.B.K.; validation, D.A.D., M.A., and S.K.; formal analysis, T.B.K. and D.A.D.; investigation, T.B.K.; resources, T.B.K. and D.A.D.; data curation, T.B.K.; writing—original draft preparation, T.B.K.; writing—review and editing, D.A.D., M.A., and S.K.; visualization, T.B.K.; supervision, D.A.D. and S.K.; project administration, D.A.D.; funding acquisition, T.B.K. and D.A.D. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available in the article.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that there are no conflicts of interest concerning the publication of this manuscript. Furthermore, all ethical considerations, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7. References

- [1] Al-Ashmori, Y. Y., Othman, I., Rahmawati, Y., Amran, Y. H. M., Sabah, S. H. A., Rafindadi, A. D. u., & Mikić, M. (2020). BIM benefits and its influence on the BIM implementation in Malaysia. *Ain Shams Engineering Journal*, 11(4), 1013–1019. doi:10.1016/j.asej.2020.02.002.
- [2] Nielsen, A. N., Jensen, R. L., Larsen, T. S., & Nissen, S. B. (2016). Early stage decision support for sustainable building renovation - A review. *Building and Environment*, 103, 165–181. doi:10.1016/j.buildenv.2016.04.009.
- [3] Azhar, S., Nadeem, A., Mok, J. Y. N., & Leung, B. H. Y. (2008). Building Information Modeling (BIM): A New Paradigm for Visual Interactive Modeling and Simulation for Construction Projects. *First International Conference on Construction in Developing Countries (ICCIDC-I)*, 435–446.

- [4] Gu, N., & London, K. (2010). Understanding and facilitating BIM adoption in the AEC industry. *Automation in Construction*, 19(8), 988–999. doi:10.1016/j.autcon.2010.09.002.
- [5] Nowak, P., Książek, M., Draps, M., & Zawistowski, J. (2016). Decision Making with Use of Building Information Modeling. *Procedia Engineering*, 153, 519–526. doi:10.1016/j.proeng.2016.08.177.
- [6] Lu, Q., Won, J., & Cheng, J. C. P. (2016). A financial decision making framework for construction projects based on 5D Building Information Modeling (BIM). *International Journal of Project Management*, 34(1), 3–21. doi:10.1016/j.ijproman.2015.09.004.
- [7] Jalaei, F., & Jrade, A. (2015). Integrating building information modeling (BIM) and LEED system at the conceptual design stage of sustainable buildings. *Sustainable Cities and Society*, 18, 95–107. doi:10.1016/j.scs.2015.06.007.
- [8] Lim, Y. W., Shahsavari, F., Sediadi, E., & Mohamad Nor Azli, N. F. (2016). Building Information Modelling for Building Energy Efficiency Evaluation Integration with Green Building Index (GBI) in Malaysia. In Malaysia. doi:10.5176/2301-394x_ace16.56.
- [9] Bank, L. C., McCarthy, M., Thompson, B. P., & Menassa, C. C. (2010). Integrating BIM with System Dynamics as a Decision-Making Framework for Sustainable Building Design and Operation. In *Proceedings of the First International Conference on Sustainable Urbanization (ICSU 2010)*, 15–23. doi:10.13140/RG.2.2.29163.23848.
- [10] Gossard, D., Lartigue, B., & Thellier, F. (2013). Multi-objective optimization of a building envelope for thermal performance using genetic algorithms and artificial neural network. *Energy and Buildings*, 67, 253–260. doi:10.1016/j.enbuild.2013.08.026.
- [11] Bui, N., Merschbrock, C., & Munkvold, B. E. (2016). A Review of Building Information Modelling for Construction in Developing Countries. *Procedia Engineering*, 164, 487–494. doi:10.1016/j.proeng.2016.11.649.
- [12] Chen, L., & Pan, W. (2016). BIM-aided variable fuzzy multi-criteria decision making of low-carbon building measures selection. *Sustainable Cities and Society*, 27, 222–232. doi:10.1016/j.scs.2016.04.008.
- [13] Daouas, N. (2011). A study on optimum insulation thickness in walls and energy savings in Tunisian buildings based on analytical calculation of cooling and heating transmission loads. *Applied Energy*, 88(1), 156–164. doi:10.1016/j.apenergy.2010.07.030.
- [14] Kurniawan, T. B., Dewi, D. A., Usman, F., & Fadly, F. (2023). Towards Energy Analysis and Efficiency for Sustainable Buildings. *Emerging Science Journal*, 7(6), 2226–2238. doi:10.28991/ESJ-2023-07-06-022.
- [15] Shaikh, P. H., Nor, N. B. M., Nallagownden, P., Elamvazuthi, I., & Ibrahim, T. (2014). A review on optimized control systems for building energy and comfort management of smart sustainable buildings. *Renewable and Sustainable Energy Reviews*, 34, 409–429. doi:10.1016/j.rser.2014.03.027.
- [16] Guan, C., Zhang, Z., Liu, S., & Gong, J. (2019). Multi-objective particle swarm optimization for multi-workshop facility layout problem. *Journal of Manufacturing Systems*, 53, 32–48. doi:10.1016/j.jmsy.2019.09.004.
- [17] European Union. (2020). Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions. European Union, Brussels, Belgium
- [18] Dixon, R. K., McGowan, E., Onysko, G., & Scheer, R. M. (2010). US energy conservation and efficiency policies: Challenges and opportunities. *Energy Policy*, 38(11), 6398–6408. doi:10.1016/j.enpol.2010.01.038.
- [19] Ross Morrow, W., Gallagher, K. S., Collantes, G., & Lee, H. (2010). Analysis of policies to reduce oil consumption and greenhouse-gas emissions from the US transportation sector. *Energy Policy*, 38(3), 1305–1320. doi:10.1016/j.enpol.2009.11.006.
- [20] Shi, X., Tian, Z., Chen, W., Si, B., & Jin, X. (2016). A review on building energy efficient design optimization ROM the perspective of architects. *Renewable and Sustainable Energy Reviews*, 65, 872–884. doi:10.1016/j.rser.2016.07.050.
- [21] Naspì, F., Arnesano, M., Zampetti, L., Stazi, F., Revel, G. M., & D’Orazio, M. (2018). Experimental study on occupants’ interaction with windows and lights in Mediterranean offices during the non-heating season. *Building and Environment*, 127, 221–238. doi:10.1016/j.buildenv.2017.11.009.
- [22] Bakr, E. H., Elbeltagi, E., & Tantawy, M. (2025). BIM Utilization to Eliminate Claims, Risks, and Improve Productivity in Construction Projects. *Civil Engineering Journal*, 11(12), 5100–5131. doi:10.28991/CEJ-2025-011-12-011.
- [23] Azhar, S., Khalfan, M., & Maqsood, T. (2012). Building information modeling (BIM): Now and beyond. *Australasian Journal of Construction Economics and Building*, 12(4), 15–28. doi:10.5130/ajceb.v12i4.3032.
- [24] Wikipedia. (2026). Autodesk Revit. Available online: https://en.wikipedia.org/wiki/Autodesk_Revit (accessed on May 2026).
- [25] Garagnani, S., & Manferdini, A. M. (2013). Parametric accuracy: Building information modeling process applied to the cultural heritage preservation. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 40(5W1), 87–92. doi:10.5194/isprsarchives-xl-5-w1-87-2013.

- [26] Klein, L., Li, N., & Becerik-Gerber, B. (2012). Imaged-based verification of as-built documentation of operational buildings. *Automation in Construction*, 21(1), 161–171. doi:10.1016/j.autcon.2011.05.023.
- [27] Sacks, R., Koskela, L., Dave, B. A., & Owen, R. (2010). Interaction of Lean and Building Information Modeling in Construction. *Journal of Construction Engineering and Management*, 136(9), 968–980. doi:10.1061/(asce)co.1943-7862.0000203.
- [28] Li, M. (2015). *Evolutionary Many-Objective Optimisation: Pushing the Boundaries*. Brunel University Research Archive (BURA), Uxbridge, England. Available online: <https://bura.brunel.ac.uk/handle/2438/11778> (accessed on May 2026).
- [29] Hu, W., Yen, G. G., & Zhang, X. (2014). Multiobjective particle swarm optimization based on Pareto entropy. *Ruan Jian Xue Bao/Journal of Software*, 25(5), 1025–1050. doi:10.13328/j.cnki.jos.004496.
- [30] Wilding, P. R., Murray, N. R., & Memmott, M. J. (2020). The use of multi-objective optimization to improve the design process of nuclear power plant systems. *Annals of Nuclear Energy*, 137, 107079. doi:10.1016/j.anucene.2019.107079.
- [31] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimisation. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, 1942–1948. doi:10.1109/ICNN.1995.488968.
- [32] Ibrahim, Z., Khalid, N. K., Ibrahim, I., Sheng, L. K., Buyamin, S., Md. Yusof, Z., & Muhammad, M. S. (2011). Function minimization in DNA sequence design based on binary particle swarm optimization. *Jurnal Teknologi (Sciences and Engineering)*, 54(1), 331–342. doi:10.11113/jt.v54.819.
- [33] Hassan, R., Cohanin, B., De Weck, O., & Venter, G. (2005). A comparison of particle swarm optimization and the genetic algorithm. *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2, 1138–1150. doi:10.2514/6.2005-1897.
- [34] Khalid, N. K., Kurniawan, T. B., Ibrahim, Z., Yusof, Z. M., Khalid, M., & Engelbrecht, A. P. (2008). A model to optimize DNA sequences based on particle swarm optimization. *Proceedings - 2nd Asia International Conference on Modelling and Simulation, AMS 2008*, 534–539. doi:10.1109/AMS.2008.25.
- [35] Coello Coello, C. A., & Reyes-Sierra, M. (2006). Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*, 2(3), 287–308. doi:10.5019/j.ijcir.2006.68.
- [36] Parsopoulos, K. E., & Vrahatis, M. N. (2008). Multi-objective particles swarm optimisation approaches. *Multi-Objective Optimisation in Computational Intelligence: Theory and Practice*, 20–42. doi:10.4018/978-1-59904-498-9.ch002.
- [37] Liu, J., Zhang, H., He, K., & Jiang, S. (2018). Multi-objective particle swarm optimization algorithm based on objective space division for the unequal-area facility layout problem. *Expert Systems with Applications*, 102, 179–192. doi:10.1016/j.eswa.2018.02.035.
- [38] Kaucic, M. (2019). Equity portfolio management with cardinality constraints and risk parity control using multi-objective particle swarm optimization. *Computers and Operations Research*, 109, 300–316. doi:10.1016/j.cor.2019.05.014.
- [39] Zhang, Y., Gong, D. W., & Ding, Z. H. (2011). Handling multi-objective optimization problems with a multi-swarm cooperative particle swarm optimizer. *Expert Systems with Applications*, 38(11), 13933–13941. doi:10.1016/j.eswa.2011.04.200.
- [40] Mousavi, S. M., Bahreininejad, A., Musa, S. N., & Yusof, F. (2017). A modified particle swarm optimization for solving the integrated location and inventory control problems in a two-echelon supply chain network. *Journal of Intelligent Manufacturing*, 28(1), 191–206. doi:10.1007/s10845-014-0970-z.
- [41] Yong, Z., Dun-Wei, G., & Wan-Qiu, Z. (2016). Feature selection of unreliable data using an improved multi-objective PSO algorithm. *Neurocomputing*, 171, 1281–1290. doi:10.1016/j.neucom.2015.07.057.
- [42] Zhang, Y., Gong, D. W., & Cheng, J. (2017). Multi-objective particle swarm optimization approach for cost-based feature selection in classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 14(1), 64–75. doi:10.1109/TCBB.2015.2476796.
- [43] Li, M., & Yao, X. (2019). Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Computing Surveys (CSUR)*, 52(2), 1–38. doi:10.1145/3300148.
- [44] Tu, P. T., Oanh, D. L. K., & Trang, D. D. (2025). Machine Learning and Parameter Optimization for Banking Stability Prediction and Determinants Identification in ASEAN. *Emerging Science Journal*, 9(3), 1189–1208. doi:10.28991/ESJ-2025-09-03-04.
- [45] Zhang, Y., Gong, D. W., & Zhang, J. H. (2013). Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neurocomputing*, 103, 172–185. doi:10.1016/j.neucom.2012.09.019.
- [46] Isiet, M., & Gadala, M. (2020). Sensitivity analysis of control parameters in particle swarm optimization. *Journal of Computational Science*, 41, 101086. doi:10.1016/j.jocs.2020.101086.

- [47] Alemi-Ardakani, M., Milani, A. S., Yannacopoulos, S., & Shokouhi, G. (2016). On the effect of subjective, objective and combinative weighting in multiple criteria decision making: A case study on impact optimization of composites. *Expert Systems with Applications*, 46, 426–438. doi:10.1016/j.eswa.2015.11.003.
- [48] London, K., Singh, V., Taylor, C., & Gu, N. (2008). Building Information Modelling project decision support framework QUT Digital Repository. Proceedings of the Twenty-Fourth Annual Conference Association of Research. Twenty-Fourth Annual Conference Association of Researchers in Construction Management (ARCOM), (May 2014), 665 –673.
- [49] Li, M., Chen, T., & Yao, X. (2022). How to Evaluate Solutions in Pareto-Based Search-Based Software Engineering: A Critical Review and Methodological Guidance. *IEEE Transactions on Software Engineering*, 48(5), 1771–1799. doi:10.1109/TSE.2020.3036108.
- [50] Djamila, H., Rajin, M., & Rizalman, A. N. (2018). Energy efficiency through building envelope in Malaysia and Singapore. *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences*, 46(1), 96–105.
- [51] MS2860:2017. (2017). Energy efficiency and use of renewable energy for residential buildings. Department of Standards Malaysian, Cyberjaya, Malaysia.
- [52] MS1525:2014. (2014). Energy efficiency and use of renewable energy for non - residential buildings - Code of Practice (Second Revision). Department of Standards Malaysian, Department of Standards Malaysian, Cyberjaya, Malaysia.
- [53] Abdelaziz, A., Elhoseny, M., & Santos, V. (2025). Advancing Network Security: Integrating Salp Swarm Optimization with LSTM for Intrusion Detection. *HighTech and Innovation Journal*, 6(4), 1185–1219. doi:10.28991/HIJ-2025-06-04-05.
- [54] Muhammad, M. S., Selvan, K. V., Masra, S. M. W., Ibrahim, Z., & Abidin, A. F. Z. (2011). An improved binary particle swarm optimization algorithm for DNA encoding enhancement. 2011 IEEE Symposium on Swarm Intelligence, 1-8. doi:10.1109/SIS.2011.5952579.
- [55] Banks, A., Vincent, J., & Anyakoha, C. (2007). A review of particle swarm optimization. Part I: Background and development. *Natural Computing*, 6(4), 467–484. doi:10.1007/s11047-007-9049-5.
- [56] Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. *Evolutionary Computation Proceedings*, 890, 69-73.
- [57] He, P., Ali, A. B. M., Hussein, Z. A., Singh, N. S. S., Bains, P. S., Saydaxmetova, S., Baghoolizadeh, M., Salahshour, S., & Alizadeh, A. (2025). Optimizing the thermostat setting points of residential and insulated buildings in the direction of economic efficiency and thermal comfort through advanced multi-purpose techniques. *Energy and Buildings*, 332. doi:10.1016/j.enbuild.2025.115428.