ISSN 1392-3730/eISSN 1822-3605

JOURNAL of CIVIL ENGINEERING and MANAGEMENT

2025
Volume 31
Issue 8
Pages 881–892
https://doi.org/10.3846/jcem.2025.24957

OPTIMISING SCHEDULED MAINTENANCE ON OPERATIONAL BUILDINGS: A MICROSERVICE-BASED BIM FRAMEWORK

Edoardo DE SANTIS ✉, Francesco Livio ROSSINI

DICEA Department, Sapienza University of Rome, Rome, Italy

Article History:
- received 19 November 2024
- accepted 22 August 2025

Abstract. Operation and Maintenance (O&M) aims to preserve the quality of the building throughout its life, keeping maintenance costs within acceptable limits. Maintenance involves different tasks, from replacing air conditioning filters to restoring structural elements. Each task has an optimal frequency, which can be flexible within a specific time range, a cost, and a duration. These maintenance activities may disrupt building operations by repeatedly interrupting ongoing activities. This research seeks to reduce these disruptions by grouping tasks within reasonably close time frames to schedule preventive maintenance plans while respecting their frequency. We propose an optimisation model, solvable using a general-purpose solver, which identifies the best time range for grouping O&M tasks. By penalising deviations from the optimal period, the model ensures that tasks are performed at the most cost-effective time. Integrated within a microservice-based architecture, the optimisation engine seamlessly links an input database and a BIM model, orchestrated using Dynamo for Revit. A case study illustrates the effectiveness of this system, consolidating multiple tasks into optimised work clusters and significantly reducing operational disruptions. The originality of this work lies in its innovative combination of optimisation techniques and BIM tools, providing a practical and scalable solution for efficient O&M management.

Keywords: maintenance, architecture, project management, O&M, BIM, microservice, scheduling, optimisation, Integer Linear Programming.

✉Corresponding author. E-mail: *edoardo.desantis@uniroma1.it*

1. Introducion

Operation and Maintenance (O&M) in buildings is critical to preserving functionality, safety, and value throughout their lifecycle. The O&M phase is usually the longest in a building's lifecycle (Li et al., 2024), involves complex interactions among various stakeholders, facilities, professionals, and management activities, as well as diverse tasks such as scheduling, space planning, repairs, and emergency management (Gnekpe et al., 2024). It encompasses the daily functions, duties, and labour required to ensure that a facility asset retains its original functionality and remains available for its primary use. Effective O&M ensures that building systems operate optimally while minimising costs and disruptions, which is particularly significant as approximately 70–80% of facility asset costs are incurred during this phase (Yussuf & Asfour, 2024; Zhang et al., 2023). Furthermore, the majority of the current building stock was built before the third millennium, meaning they were designed with outdated approaches for maintenance, energy consumption, ICT infrastructure integration, and operational aspects (Gao & Pishdad-Bozorgi, 2019). These factors account a significant impact on the total lifecycle costs for building owners (Abuhussain et al., 2024). This highlights the importance of maintenance strategies in prolonging the useful life of structures, optimising costs, and enhancing user satisfaction.

Three main maintenance approaches are commonly adopted: preventive, corrective, and condition-based (Horner et al., 1997). Preventive Maintenance (PM) is configured to perform maintenance activities at predetermined intervals or based on predetermined criteria, regardless of the equipment's current operational state. The primary objective of PM is to reduce the likelihood of equipment failure and extend its lifespan through systematic inspections, replacements, and repairs. Corrective Maintenance (CM), on the other hand, refers to the actions taken to restore equipment to operational condition after a failure has occurred. This strategy is typically reactive and involves repairs or replacements only after a fault has been identified, leading to unplanned downtime and potentially increased operational costs. Condition-Based Maintenance (CBM) incorporates real-time monitoring of equipment conditions and operational parameters to dictate the timing of maintenance interventions. This approach allows for maintenance to be performed only when indicators suggest that a system's performance is degrading, thereby optimizing maintenance frequency and costs.

Maintenance scheduling presents significant challenges, particularly in complex facilities such as office buildings, universities, or hospitals. Unplanned downtime and inefficient planning often lead to operational disruptions and cost overruns. Conventional approaches frequently fail to address the interdependencies between tasks, resulting in fragmented schedules and compromised efficiency (Konig et al., 2012; Ni et al., 2003). O&M tasks are often interpreted as a series of independent projects that must be organised in conjunction with the ongoing operational activities within the building (Wettewa et al., 2024). This issue is particularly relevant for infrastructural facilities that must remain functional even in scenarios of risk and disaster (Hosseinzadeh et al., 2023). Recent shifts, driven by events like pandemics, natural disasters, and advancements in digitalisation, have reframed maintenance as a continuous process of innovation rather than repetitive interventions to ensure functionality (Scaife, 2024).

Building Information Modelling (BIM) has emerged as a transformative tool for improving maintenance management by providing a parametric and detailed representation of building components, facilitating reliable information exchange from design to O&M phases (Peng et al., 2017). On the other hand, the heterogeneity of information, complexity in storage, and specialised functions of users result in increasingly non-intuitive data (Torres-Sainz et al., 2024). Additionally, BIM allows the integration and synchronisation of extensive data, supporting stakeholder collaboration. It also enables visualisation and management of non-repetitive tasks in construction and maintenance (Dallasega et al., 2019). However, current BIM-based systems rarely integrate O&M scheduling into a unified framework, missing the ability to monitor spatial and contextual information effectively.

BIM-based facility management systems enhance scheduling and task planning compared to traditional manual approaches (Golabchi et al., 2016). Several studies report that using BIM enables real-time access to facility data, automates fault detection, and streamlines work order management (Davtalab, 2017). For example, a BIM system with augmented reality integration reduced task completion times significantly, while real-time data-driven BIM implementations and lean-agile approaches shortened operation and maintenance cycles (Khan et al., 2023). Improved accuracy of geometric information and more effective information handover also support efficient resource allocation and proactive maintenance scheduling (Kelly et al., 2013). These findings suggest that BIM's integrated digital environment affords scheduling and planning efficiencies that traditional approaches often lack (Golabchi et al., 2016).

A comprehensive system architecture that bridges the BIM environment with task scheduling processes can address these gaps. Leveraging microservices and orchestrators to manage data flow and automation, O&M activities can be transformed from isolated projects into interconnected processes that align with the ongoing operational needs of the building (Dubey et al., 2024).

Advances in optimisation techniques, such as Integer Linear Programming (ILP) (Schrijver, 1998), have been leveraged to address some of these issues. These techniques employ exact algorithms to find the optimal solution, or, when computational complexity is excessive, heuristic algorithms to identify feasible solutions. An advantage of utilising ILP is the wide availability of powerful solver packages, ranging from open-source tools such as SCIP and CBC to commercial solvers like CPLEX and Gurobi which offer academic licences alongside enterprise-grade solutions (Ashouri et al., 2013). Also, some studies have explored grouping maintenance activities using hybrid clustering approaches that combine hierarchical and k-means algorithms (Ahmed et al., 2022).

The review of existing literature reveals several key gaps that this study aims to address. Firstly, there is a lack of integration between BIM and scheduling optimisation models; current approaches rarely establish a connection between advanced scheduling algorithms and the spatial or contextual data embedded within BIM environments. Secondly, the use of microservices in O&M planning remains underexplored. Despite their potential to enable modular, scalable, and resilient system design, microservice-based architectures have yet to be widely adopted to support multi-user, multi-project environments in the domain of building maintenance and operations.

In light of these gaps, the objectives of this research are as follows:

- To develop a mathematical optimisation model that clusters maintenance tasks to minimise operational disruption and ensure compliance with time-related constraints.
- To integrate this model with BIM, enabling spatial visualisation of the schedule within the building model.
- To leverage microservice architecture for scalable, modular, and collaborative O&M scheduling.

This study introduces a novel system that combines BIM, optimisation techniques, and a microservice architecture to support the efficient scheduling of preventive maintenance in operational buildings. The system enables the grouping of tasks into optimised clusters, thus reducing the number of active workdays and minimising spatial disruptions. The microservice design promotes flexibility, allowing future integration with external modules, such as IoT data feeds or FM platforms, while the BIM integration ensures spatial and contextual awareness. A case study illustrates the feasibility and impact of the proposed methodology, reducing active maintenance days by over 58% compared to conventional scheduling. The outcome is a scalable, adaptable, and practical solution to a persistent challenge in facility operations.

This paper is structured to explore the proposed methodology and its application comprehensively. Section 2 focuses on the primary methodology, beginning with the Model Definition (Section 2.1), which explains the problem parameters, variables, objective function, and constraints. This is followed by Model Solution (Section 2.2), which addresses solver selection, implementation, visualisation, and results from an example. Section 3 discusses the System Architecture, emphasising the role of the microservices structure, the BIM integration, and the use of Dynamo for orchestrating system components. A Case Study (Section 4) applies the methodology to a realistic scenario, examining the database setup, optimisation process, and analysis of the results. Section 5 discusses results, limitations, and future work. Finally, Section 6 concludes the paper.

2. Optimisation Engine

This section outlines the methodology developed to optimise the scheduling of O&M tasks in an operational building through an ILP formulation and its solution. The approach tackles challenges such as minimising costs, enhancing scheduling efficiency, and reducing operational disruptions. The outcome of this section is the definition of an Optimisation Engine that, in Section 3, will be integrated with a BIM workflow and a microservice-based system architecture.

While the overall structure of the model draws on common ILP principles used in scheduling problems, the formulation introduces novel elements specific to the context of O&M in operational buildings. In particular, the inclusion of extended scheduling intervals with cost penalties for early or late starts, constraints on simultaneous task execution, and the mapping of tasks to spatial elements within a BIM-integrated environment represent distinct contributions not previously formalised in this way.

2.1. Model definition

A set of decision variables is introduced to support the formulation of the problem. Given the nature of the problem, an ILP approach is deemed appropriate. This paragraph details the construction of the formulation, consisting of a linear objective function and a set of linear constraints.

The initial step involves the definition of the model that will be solved in the following sections. The objective is to minimise the working days in an operational building thereby limiting the duration and extent of off-limits areas during maintenance operations. This approach aims to avoid or reduce maintenance activities that could interfere with daily operations and, in some cases, pose a risk to occupants due to the use of auxiliary work equipment. To this extent, a set of variables is defined to guide the solution. ILP formulations are well-suited to represent the defined problem due to their capability to handle such constraints. This paragraph outlines the construction of the formulation with a set of linear constraints and one objective function.

In this study, the ILP formulation focuses on temporal clustering of maintenance activities, abstracting resource and spatial interdependencies into an aggregate concurrency limit Z. This simplification was chosen to clearly isolate and evaluate the clustering effect and to ensure a compact, reproducible case study. The model is designed to be modular: additional constraints for shared resources, personnel allocation, and spatial conflicts can be integrated within the same mathematical framework (see Section 2.1.5) but were not activated in the current work.

2.1.1. Problem parameters

The scheduling problem involves a set of parameters that define the characteristics of the building and its maintenance activities. To generalise the problem, a set is defined that includes:

- T: Number of time units in the project schedule.
- N: Number of maintenance activities to be scheduled.
- Z: Maximum number of concurrent activities in the same time unit.
- F: List of time units during which activities cannot be carried out.
- d_i: Duration of activity i.
- l_i, r_i: Time unit of the beginning and end of the standard time interval in which maintenance activity i should start to reduce extra costs.
- g_{l_i}, g_{r_i}: Number of time units to add, respectively before l_i and after r_i, providing flexibility for early and late starts of activity i, defining an extended time interval in which activity i can start, if necessary, with extra costs.
- P: Maximum number of activities that can start within the margin given by the extended time interval.
- c_i: Extra cost per time unit for any delay or anticipation from the standard time interval for activity i.
- C: Maximum extra cost allowed for the whole scheduling solution.

To accommodate varying project requirements and resource constraints, the values of the problem parameters can be adjusted. This parameter relaxation allows for tailored scheduling solutions, optimising the trade-off between strict adherence to constraints and a more flexible schedule that can better adapt to real-world construction scenarios.

For example, the flexibility to adjust Z, the maximum number of concurrent activities allowed within a time unit allows the model to handle varying resource availability. Similarly, the list F, which designates time units during which no activities can be carried out, accounts for operational restrictions or other project-specific constraints. Furthermore, the standard time interval can be extended by adding g_{l_i} and g_{r_i}, providing flexibility for early or late starts, creating an extended time interval (Malucelli & Nicoloso, 2007). By adjusting these parameters, the strictness or flexibility of the schedule can be controlled, accommodating potential delays or early starts while dealing with additional costs. The model further allows for the inclusion of constraints on these costs through parameters such as c_i (extra cost per time unit of delay or anticipation for activity i) and C (the maximum extra cost allowed for the project).

2.1.2. Variables

The decision variables required for the ILP formulation are defined as follows:

- x_{ij}: Binary variable representing whether each activity i starts on time unit j (1 if activity i starts on day j, 0 otherwise).
- y_j: Binary variable representing whether there are any activities scheduled in time unit j (1 if time unit j has at least one activity carried out, 0 otherwise).
- w_{ij}: Auxiliary binary variable used for modeling purposes, representing whether activity i is active on time unit j (1 if activity i is active on day j, 0 otherwise).
- k_i: Auxiliary variable representing the number of time units an activity i starts outside its standard interval.

2.1.3. Variables

The objective function of the problem is to minimise the total number of time units with scheduled activities to reduce operational disruption and to reduce risks to the occupants. It is formulated as follows:

$$\min z = \sum_{j=1}^{T} y_j . \tag{1}$$

The optimisation model aims to find an optimal schedule that efficiently allocates activities to time units, minimising the number of days construction workers occupy the building. By minimising the value of z, the model improves the utilisation of available time slots.

2.1.4. Constraints

A set of constraints is established to ensure the problem is modelled accurately. The following constraints have been defined for this purpose:

- Each maintenance activity must be scheduled to start exactly once inside the extended interval.

$$\sum_{j=l_i-g_{l_i}}^{r_i+g_{r_i}} x_{i,j} = 1 \quad \text{for } i = 1,\ldots,N. \tag{2}$$

- At the same time, we must restrict the other values to zero, meaning that each activity cannot be scheduled outside the extended interval.

$$\sum_{j=1}^{l_i-g_{l_i}-1} x_{i,j} + \sum_{j=r_i+g_{r_i}+1}^{T} x_{i,j} = 0 \quad \text{for } i = 1,\ldots,N. \tag{3}$$

- The duration of each activity must be satisfied. Specifically, the sum of the values of the auxiliary variable $w_{i,j}$ must be, for each activity i, equal to its duration d_i.

$$\sum_{j=1}^{T} w_{i,j} = d_i \quad \text{for } i = 1,\ldots,N. \tag{4}$$

- This constraint introduces a link between $x_{i,j}$ and $w_{i,j}$ and helps to model the duration of each activity. It ensures that the activity is considered in progress at a particular moment if it started in the same time unit or in one of the following d_i time units.

$$w_{i,j} \geq \sum_{h=\max\{1,\, j-d_i+1\}}^{j} x_{i,h} \quad \text{for } i = 1,\ldots,N,\ j = 1,\ldots,T. \tag{5}$$

- The number of all the activities starting within the extensions of the standard interval must not exceed the value of P.

$$\sum_{i=1}^{N} \left(\sum_{j=l_i-g_{l_i}}^{l_i-1} x_{i,j} + \sum_{j=r_i+1}^{r_i+g_{r_i}} x_{i,j} \right) \leq P. \tag{6}$$

- This constraint links the variables in the objective function to those in the problem. It ensures that y_j takes value 1 when at least one activity is scheduled during time unit j. Specifically, the variable is forced to be 1 if the number of activities carried out during that time unit (represented on the left-hand side of the inequality) is greater than or equal to 1. Conversely, if such number is 0, the objective function, in order to minimise its value, would assign value 0 to y_j.

$$\sum_{i=1}^{N} w_{i,j} \leq N y_j \quad \text{for } j = 1,\ldots,T. \tag{7}$$

- The number of activities going on in each time unit must not exceed Z, the maximum number of concurrent activities allowed in each time unit.

$$\sum_{i=1}^{N} w_{i,j} \leq Z \quad \text{for } j = 1,\ldots,T. \tag{8}$$

- The sum of the extra cost due to working in the extensions of the standard interval must not exceed the value of C.

$$k_i \geq l_i - \left(\sum_{j=l_i-g_{l_i}}^{r_i+g_{r_i}} j\, x_{i,j} \right) \quad \text{for } i = 1,\ldots,N; \tag{9}$$

$$k_i \geq \left(\sum_{j=l_i-g_{l_i}}^{r_i+g_{r_i}} j\, x_{i,j} \right) - r_i \quad \text{for } i = 1,\ldots,N; \tag{10}$$

$$\sum_{i=1}^{N} c_i k_i \leq C. \tag{11}$$

- Each activity must not be performed during time intervals included in F.

$$w_{i,j} = 0 \quad \text{for } i = 1,\ldots,N,\ j \in F. \tag{12}$$

- These variables are constrained to assume binary values.

$$x_{i,j} \in \{0,1\} \text{ for } i = 1,\ldots,N,\ j = 1,\ldots,T; \quad (13)$$

$$w_{i,j} \in \{0,1\} \text{ for } i = 1,\ldots,N,\ j = 1,\ldots,T; \quad (14)$$

$$y_j \in \{0,1\} \text{ for } j = 1,\ldots,T. \quad (15)$$

- The last variable can assume every value greater than or equal to zero.

$$k_i \geq 0 \text{ for } i = 1,\ldots,N. \quad (16)$$

2.1.5. Potential model extensions

The formulation above captures the temporal clustering of maintenance activities with a global concurrency limit Z and the defined cost and flexibility parameters. This scope was deliberately selected to isolate and evaluate the benefit of time-based clustering within a BIM-enabled workflow, and to maintain a compact, transparent optimisation model for the case study.

For operational deployment, the same ILP framework can be extended with additional constraints to represent common real-world interdependencies. Examples include:

- **Cumulative resource capacities:**
  Let K be the number of resource types, such as technicians or equipment units. Each activity i requires an amount $r_{i,k}$ of resource type k, while the total available capacity of that resource at time unit j is given by $R_{j,k}$. To ensure feasibility, the sum of all resource demands must not exceed availability, which is enforced through the constraint $\sum_{i=1}^{N} r_{i,k} w_{i,j} \leq R_{j,k}$ for each time unit $j = 1,\ldots,T$, and resource type $k = 1,\ldots,K$.
- **Spatial or operational conflicts:**
  A binary conflict matrix D is introduced to indicate whether two activities i and i' cannot occur simultaneously due to adjacency, access restrictions, or operational interference. Element $d_{i,i'} = 1$ if activities i and i' cannot be accessed simultaneously, and $d_{i,i'} = 0$ otherwise. For each conflicting pair (i, i'), the constraint $w_{i,j} + w_{i',j} \leq 1$ for all $j = 1,\ldots,T$ guarantees that these activities are not scheduled at the same time. In practice, D is treated as input data derived from BIM (e.g., room adjacency or blocked circulation paths) and operational rules (e.g., noise or safety constraints) and can be updated without altering the core formulation.
- **Area-level capacity limits:**
  Let A be the number of areas in the building, and S_a the set of tasks assigned to area a. The maximum number of tasks allowed to be active in area a during time unit j is given by $Z_{a,j}$. This is modelled by the constraint $\sum_{i \in S_a} w_{i,j} \leq Z_{a,j}$, for each time unit $j = 1,\ldots,T$, and area $a = 1,\ldots,A$ allowing differentiated limits per zone, respecting each local capacity.

All these are standard linear constraints that are compatible with the decision variables and solver approach presented above and can be implemented as modular additions in the same optimisation engine. In this study, they were not activated, as the objective was to demonstrate the temporal clustering principle and BIM integration pipeline on a simplified, clearly defined scenario.

2.1.6. Visual summary of model components

The relationships between the problem parameters, decision variables, and constraints are summarised visually to provide an overview of the core elements of the ILP formulation.

Figure 1 illustrates the interaction between each component of the model. In this example, the problem involves ten time units (T = 10) and two activities (N = 2). For activity 1, the standard time interval is spanning four time units from $l_1 = 4$ to $r_1 = 7$, and the extended time interval spans eight time units, considering three time units for early starts ($g_{l_1} = 3$) and one time unit for late starts ($g_{r_1} = 1$) of the maintenance activity. The orange bar represents the duration of the activity, which is four time units ($d_1 = 4$), starting from the fifth time unit ($s_1 = 5$).

It is evident that the smallest value of $l_i - g_{l_i}$ must be greater than the first time unit of the scheduling problem, and the total number of time units T must be greater than the greatest value of $r_i + g_{r_i} + d_i$ to allow the scheduling problem to work correctly.

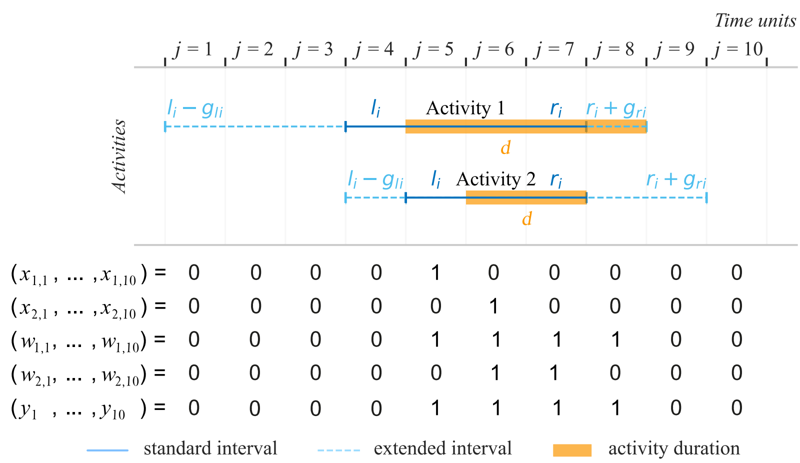

Figure 1. Visual summary of most of the model components

The values of the variables are also represented. For example, $x_{i,j}$ for activity 1 and time unit 5 is $x_{1,5}=1$ since the activity starts at that time unit. On the other hand, all other $x_{1,j}$ values are zero since they are not the starting point of the activity. Similarly, $w_{1,5}=w_{1,6}=w_{1,7}=w_{1,8}=1$, indicating that the activity is ongoing during those time units, while it equals zero for all the other time units. Lastly, it is shown that $y_5=y_6=y_7=y_8=1$, as at least one activity is carried out during those time units.

2.2. Model solution

While the previous section presented the proposed ILP formulation of the problem, this section focuses on its solution.

A mathematical optimisation solver was employed. Several options are available, such as CPLEX, SCIP, or Gurobi Optimizer. Gurobi Optimizer was selected due to its strong support for multiple programming languages, particularly its seamless integration with Python (Gurobi Optimization, LLC, 2024). This compatibility aligns well with the requirements of the project, as Python facilitates smooth integration with the microservice architecture that will be discussed in Section 3.

The process began with the creation of a Gurobi environment and an empty optimisation model, which acts as a container for variables, constraints, and the objective function that define the optimisation problem. In this case, binary variables *x*, *y*, and *w*, and variable *k* are created with the *Model.addVar*() function. Then, all the constraints and the objective function are defined through the *Model.addConstrs*() and *Model.setObjective*() functions, respectively. Once fully defined, the model is ready to be solved and optimised by Gurobi, which outputs the value of each defined variable, the objective function value, and a set of technical information.

Based on the solver's output, a JSON file was generated to summarise the essential information in a format accessible to both users and other system architecture components. In particular, the only relevant information from the solution is the values $x_{i,j}=1$ which indicates that activity *i* should start at time *j*, and the variable y_j = 1, which indicates that time unit *j* has at least one activity going on. All other variables are functional to the problem's solution but are not directly used for interpreting the results. The JSON file also includes all input data, such as activity parameters and problem variables, for future reference.

This file can additionally be used to generate visualisation diagrams for user interpretation. These graphical representations help project managers and stakeholders better understand the impact of the optimisation and make informed decisions. They provide a clear, intuitive way to assess the results, offering an immediate understanding of the activity timelines, as illustrated in Figures 2 and 3. The X-axis represents time, while the Y-axis displays activities positioned on horizontal tracks. For the optimised solution, the activity's start time and duration are highlighted on the timeline.

The input data for this example is as follows: *T* = 30, *N* = 6, *P* = 3, *Z* = 3. For simplicity, and to focus on the process demonstration, in this illustration parameters such as cost and the list of days to avoid have been omitted. The values of the input parameters are shown in Figure 2; for instance, $r_1=23,\ l_1=27,\ g_{l_1}=g_{r_1}=2$.

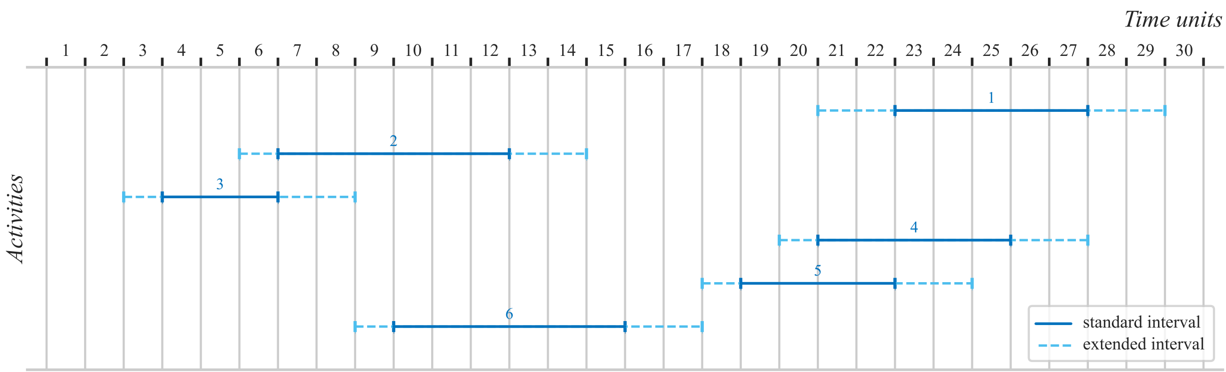

Figure 2. Visual diagram of example input data

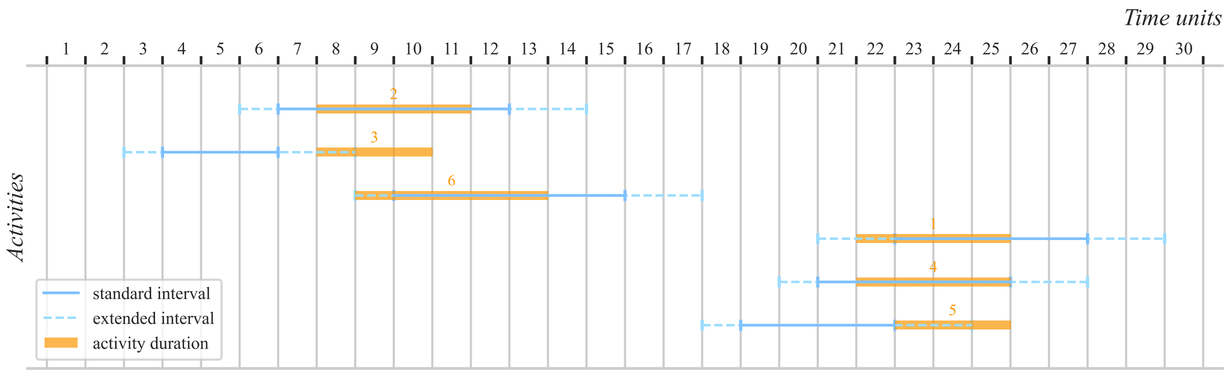

Figure 3. Visual diagram of example output data

Figure 3 presents the output diagram. In this example, the variables $x_{i,j} = 1$, which indicates the starting times, are $x_{2,8}$, $x_{3,8}$, $x_{6,9}$, $x_{1,22}$, $x_{4,22}$, $x_{5,23}$. The variables $y_j = 1$, representing time units with at least one maintenance task executed, are: y_8, y_9, y_{10}, y_{11}, y_{12}, y_{13}, y_{22}, y_{23}, y_{24}, y_{25}. In other words, the optimisation results show that the best solution is to group activities 2, 3, and 6 with working time units from 8 to 13 and activities 1, 4, and 5 with working time units from 22 to 25. This configuration minimises the total active time units to 10.

A comprehensive real-world case study will be explored in Section 4.

3. System architecture

The previously discussed Optimisation engine (Section 2) can be integrated within a broader system architecture that bridges the BIM environment with task scheduling processes. Figure 4 outlines the four main components of the system: a database, the Optimisation Engine, a BIM model, and an orchestrator that coordinates these elements. Data is organised in an accessible format and processed through the Optimisation Engine. The resulting optimised schedule is then visualised and integrated into the database and the BIM environment, facilitating decision-making.

The system adopts a microservice architecture (Nadareishvili et al., 2016), enabling independent development, deployment, and scaling of the core components. This approach enhances flexibility, allowing each service to use the most suitable technologies for its functionality. It also supports seamless integration of new modules or updates, such as connectivity to facility management software or the incorporation of new additional functionalities.

Following this logic, the Optimisation Engine was encapsulated within a container, allowing it to be executed on a server via POST HTTP requests. This approach ensures that multiple users can access and use the engine simultaneously, enabling scalability and promoting collaborative use across different teams or projects.

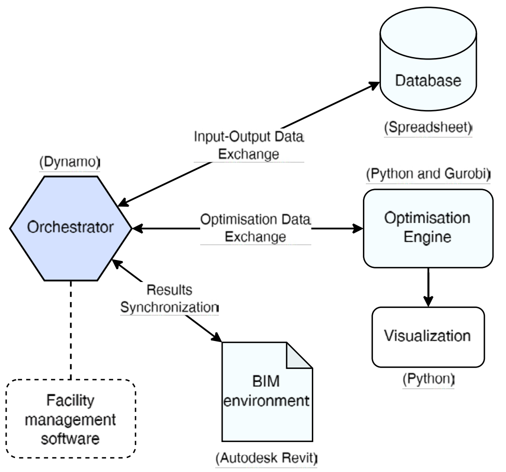

Figure 4. Summary diagram of the system architecture

3.1. Database

One system component is a database to store data before and after processing by other modules. To enhance accessibility for a broader user base, a spreadsheet-based database is utilised to manage input and output data. This solution is beneficial for users who are more comfortable working in familiar environments like Microsoft Excel or Google Sheets, and it simplifies tasks such as updating, reviewing, and sharing. The spreadsheet format provides a flexible interface for manual adjustments while still allowing the storing of structured data, such as task schedules, resources, and other inputs necessary for optimisation.

Moreover, the spreadsheet database integrates effectively with the microservice architecture, enabling rapid data exchange between the optimisation model, visualisation tool, and BIM systems. However, the data in the spreadsheet must comply with the specified model; otherwise, the other components may not function correctly, potentially causing issues in the optimisation process and data integration.

The database is structured to meet the input requirements of the Optimisation Engine, including task durations, associated costs, and temporal constraints. Each task entry is linked to a specific room in the BIM model representing the building. This data is later used to integrate the optimisation results into the model, visualising the maintenance schedule.

It may be worth considering replacing the spreadsheet-based database with a more modern and efficient solution in a microservice-based structure. However, this might come at the expense of user accessibility. Alternatively, a module could be added as an interface, allowing users to retain simplified usage while the system leverages a more advanced back-end database.

3.2. BIM environment

Another component is the BIM model of the building to be maintained. Autodesk Revit was selected due to the widespread diffusion in both the academia and the AEC sector. Its flexibility and functionalities make it ideal for modelling, visualising, and managing building information. Nevertheless, from a microservice-based perspective, the system architecture could potentially support multiple models, even across different software platforms.

Within the BIM environment, the building is modelled in detail, with custom parameters added to each element to store information related to maintenance tasks. Since each room and building component may require multiple maintenance activities over time, only the details of the first scheduled activity for each room was stored, as determined by the results by the optimisation output. This approach ensures the BIM model reflects the most immediate maintenance needs while preventing unnecessary clutter from storing all tasks simultaneously.

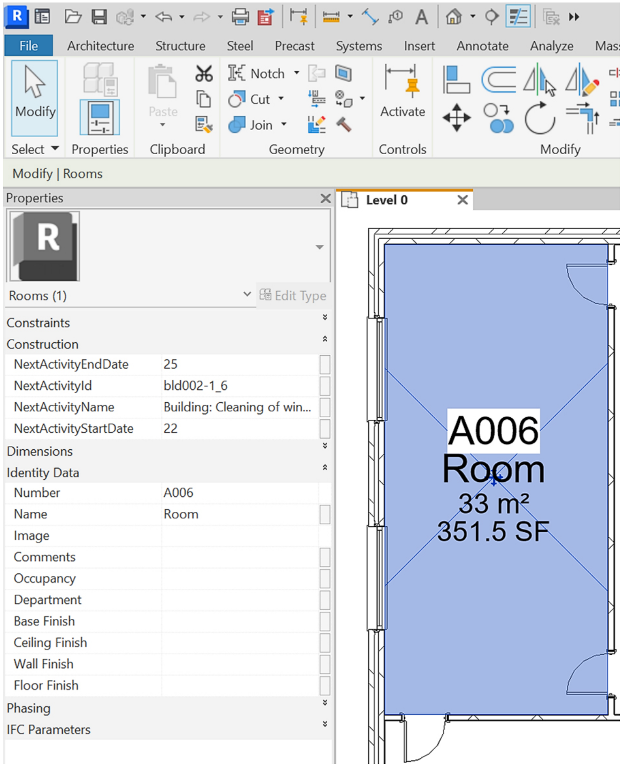

Figure 5. Room modelled in a BIM environment and its four associated parameters

To achieve this, four parameters are defined within the model: *NextActivityId*, *NextActivityName*, *NextActivityStartDate*, and *NextActivityEndDate*. The results from the Optimisation Engine are mapped back into these parameters of the BIM elements, ensuring that the model reflects both the spatial and temporal aspects of the project.

Beyond visualisation, the BIM platform can also be employed to provide structured data to facilitate spatially-aware optimisation. For example, room adjacency data, routes of access, or types of area can be accessed directly from the BIM model and translated into conflict matrices or zone capacity parameters to be used by the Optimisation Engine. While these spatial constraints were not utilised in this study, the framework has been set up so that this kind of data is able to be generated automatically in the BIM environment and propagated by the orchestrator to the Optimisation Engine. This ensures that the framework can evolve from schedule visualisation towards spatially informed scheduling without requiring changes to the underlying design.

3.3. Orchestrator component

The critical component of the system is the orchestrator, which manages and coordinates individual microservices within a distributed architecture. For this implementation, Dynamo for Revit was selected due to its visual programming environment, which offers flexible and intuitive control over data workflows and benefits from its native integration with the chosen BIM software.

The orchestrator serves as a bridge between the BIM environment, the Optimisation Engine, the database, and any other additional component. When interacting with the input database, maintenance data is translated into a format suitable for analysis by the Optimisation Engine and requests are sent to the server hosting the engine, as illustrated in Figure 6. Additionally, the same Dynamo script triggers the visualisation modules, which generate graphical representations of the optimisation results, providing users with clear insights into the proposed schedules.

A dedicated script is also employed to synchronise the optimised schedule back into the BIM model. This script updates the model by embedding the optimisation results, specifically highlighting each room or area's first scheduled maintenance activity. The mapping between data and building elements is performed by referencing

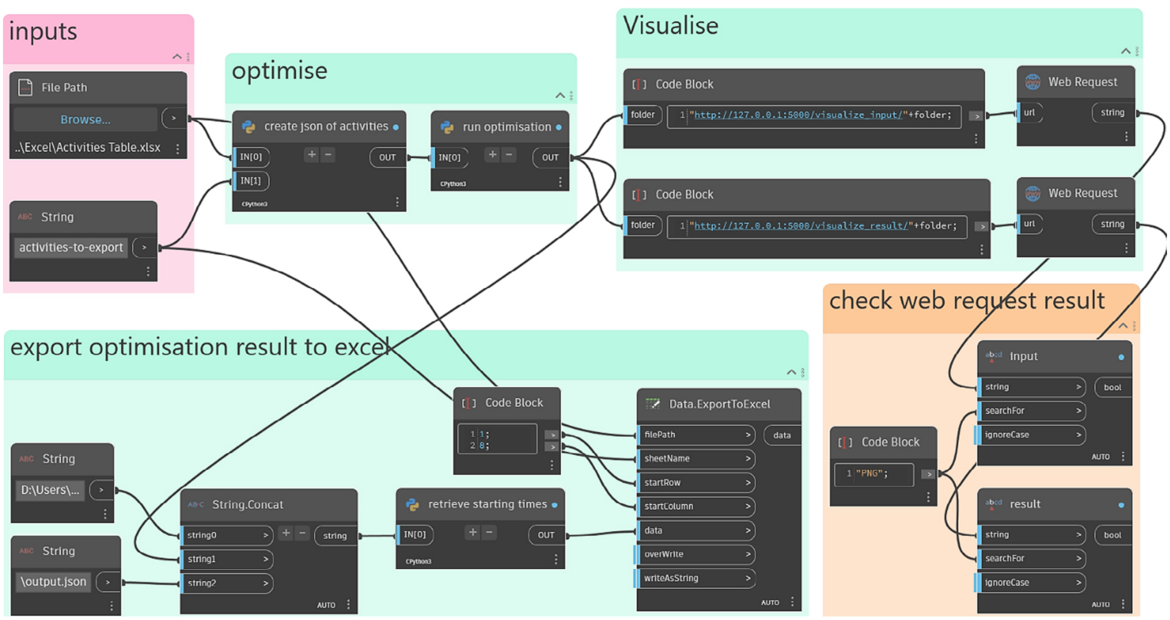

Figure 6. Dynamo script to read input data, process the optimisation, visualise results, and export results

the elements through their IDs, ensuring that the schedule is accurately integrated within the spatial context of the building.

In addition to exchanging task and schedule information, the orchestrator can also serve as the channel through which spatial attributes from BIM are translated into optimisation-ready inputs. For instance, Dynamo scripts can extract adjacency relationships, room categories, or circulation constraints from the BIM model. This creates a bidirectional flow where BIM not only visualises the scheduling results but also actively informs the optimisation process with spatial awareness.

4. Case study

To evaluate the functionality of the proposed methodology, a case study was conducted by defining a set of maintenance activities commonly scheduled in operational buildings, such as academic institutions or office buildings. The data employed in this case study are realistic in nature, although generated for research and experimental purposes.

An existing university building from the authors' institution was selected for modelling. The building comprises both intermittently and continuously used spaces, including receptions, hallways, study rooms, administration offices, and libraries. As universities are typically high-occupancy environments with constant movement of staff and students, minimising disruption caused by maintenance activities is essential.

A set of 19 distinct maintenance activities, ranging from cleaning air conditioning filters to replacing light bulbs, was defined. Each activity was assigned an Id, a description, and parameter values l, r, g_l, g_r, d, and c as presented in Table 1.

The optimisation model was configured with a total number of time units T = 365, assuming yearly planning of the maintenance activities. Certain time units were excluded from scheduling due to known high-priority events (e.g., conferences) represented by F = [30, 31, 32, 33, 34, 35, 105, 106, 107, 108, 212, 213, 214, 300, 301, 302, 303, 304, 305]. Additionally, the following constraints were defined: a maximum of Z = 5 concurrent activities, a limit of P = 20 on the maximum number of activities starting early or late, and a maximum total cost C = 1000 for early or late starts.

Although only 19 unique maintenance activities were defined, many of them recur multiple times over the course of the year based on standard maintenance frequencies or manufacturer recommendations. Using this information, the number of occurrences for each activity was determined and it is represented by the parameter p of Table 1. Altogether, these recurring activities result in a total of N = 56 activity instances across the year. For simplicity, Table 1 only displays l and r parameters for the first occurrence of each activity. However, the parameters of subsequent occurrences can be calculated using the activity's frequency p as $l_k = l_i + 365 / p_i (k-1)$ and $r_k = r_i + 365 / p_i (k-1)$, respectively. In these expressions l_k and r_k are the adjusted left and right margins for the standard time intervals of the k^{th} occurrence of activity i, and p_i is the number of times it recurs annually. This ensures maintenance is evenly distributed over the planning horizon.

Table 1. Values of the parameters for each unique maintenance activity (l and r only of the first occurrence)

Id	Maintenance activity	l	r	g_l	g_r	d	c	p
hvc001	HVAC: Cleaning of air conditioning filters	31	61	15	15	7	60	2
hvc002	HVAC: Replacement of air conditioning filters	99	129	20	20	3	15	1
mep001	MEP: Inspection for water leaks in plumbing systems	120	150	20	20	3	23	1
mep002	MEP: Inspection of electrical grounding systems	55	85	15	15	1	41	2
mep003	MEP: Sanitisation of air ducts and ventilation grilles	137	167	15	15	10	57	2
mep004	MEP: Thermal shock treatment of the sanitary water system	277	307	15	15	1	36	1
mep005	MEP: Lift systems inspections and maintenance	130	160	20	20	2	17	1
mep006	MEP: Replacement of light bulbs and fixtures as required	280	310	30	15	5	39	1
bld001	Building: Painting and decoration of shared areas	18	48	17	17	5	38	1
bld002	Building: Cleaning of windows and window frames	15	30	7	7	5	20	6
bld003	Building: Inspection of furniture for wear and tear	84	114	30	30	1	55	2
bld004	Building: Rodent control and prevention measures	45	75	15	15	2	25	2
bld005	Building: Indoor plant care and maintenance	10	17	3	3	1	30	12
ict001	ICT: maintenance of data network and systems	25	27	15	15	5	35	2
ict002	ICT: management and distribution of consumables	18	28	3	2	1	52	12
fir001	Fire Safety: testing of evacuation speaker system	25	55	15	15	1	50	2
fir002	Fire Safety: inspection and servicing of fire extinguishers	55	85	20	5	2	33	2
fir003	Fire Safety: inspection and maintenance of emergency doors	90	120	20	5	1	47	2
fir004	Fire Safety: inspection and maintenance of emergency lights	110	140	20	5	2	46	2

Note: The data in this table is realistic but was created for research and experimental purposes.

After preparing the input data, the Optimisation Engine was executed using the dedicated Dynamo script. From the output the start times for all 56 scheduled activities were extracted and summarised in Table 2. Each line is an activity, identified by the Id, and in column *s* the start times for every occurrence of each unique maintenance activity are listed. For instance, activity 'hvc001' has a number p = 2 of occurrences in the year that start at time unit 17 and time unit 199.

Additionally, the results are visualised in an output diagram, ordered by starting time. Figure 7 presents a cropped view of this diagram, focusing on time units from 10 to 60 for nine of the 56 activities.

From the diagram, the first two activity clusters can be observed, optimised to minimise operational disruption in the building and consequently to reduce risks for the occupants. For example, between time units 17 and 23, the first occurrences of activities hvc001, bld005, bld001, ict002, bld002, and ict001 are grouped into a single seven-day cluster, with only two activities starting earlier than their optimal frequency. If executed independently, these six activities would have occupied the facility for a cumulative total of 21 days. Through optimisation, this duration is reduced by 66.7%.

Table 2. Optimised start times for every occurrence of each unique maintenance activity

Id	*s*
hvc001	17, 199
hvc002	129
mep001	132
mep002	75, 230
mep003	128, 321
mep004	290
mep005	131
mep006	258
bld001	18
bld002	19, 73, 131, 201, 258, 321
bld003	109, 290
bld004	75, 258
bld005	17, 46, 73, 109, 132, 169, 199, 230, 261, 290, 321, 350
ict001	19, 201
ict002	18, 46, 76, 109, 137, 169, 205, 230, 259, 290, 330, 350
fir001	46, 205
fir002	75, 261
fir003	109, 290
fir004	133, 321

A full analysis of the diagram shows that the 56 activities are grouped into 12 clusters, reducing the total number of working days to 62. In contrast, a non-optimised scenario, where each activity's start time coincided with the left value of its standard interval (s_i = l_i), would result in 151 working days. Overall, the optimisation reduces the total number of active maintenance days by 58.9%.

The distribution of the number of activities per cluster is as follows: [6, 3, 6, 4, 8, 2, 6, 3, 6, 5, 5, 2], with corresponding cluster durations of [7, 1, 5, 1, 10, 1, 7, 1, 5, 1, 10, 1] time units. This results in an average of 4.7 activities per cluster and a mean cluster duration of 4.8 days, indicating efficient grouping without overloading the schedule. Among the 56 activities, 17 required early or late starts, slightly deviating from their optimal time intervals. These deviations were accepted within the model's predefined constraints.

The resulting schedule shows that maintenance efforts are concentrated in well-defined time blocks, improving planning and coordination across teams. These outcomes demonstrate the practical advantages of the proposed optimisation approach in real-world O&M scenarios.

It should be noted that the results are influenced by the parameters defined initially. Adjusting these parameters may lead to a different solution. Furthermore, although the solution provided is exact, multiple equivalent solutions may exist and are not explicitly presented by the Optimisation Engine.

5. Discussion

The current modelling approach, while effective in optimising maintenance scheduling, presents several areas for improvement. One key limitation is the assumption of static occupancy patterns across the building. In practice, room usage varies dynamically, particularly in high-traffic areas such as corridors or communal spaces. To ad-

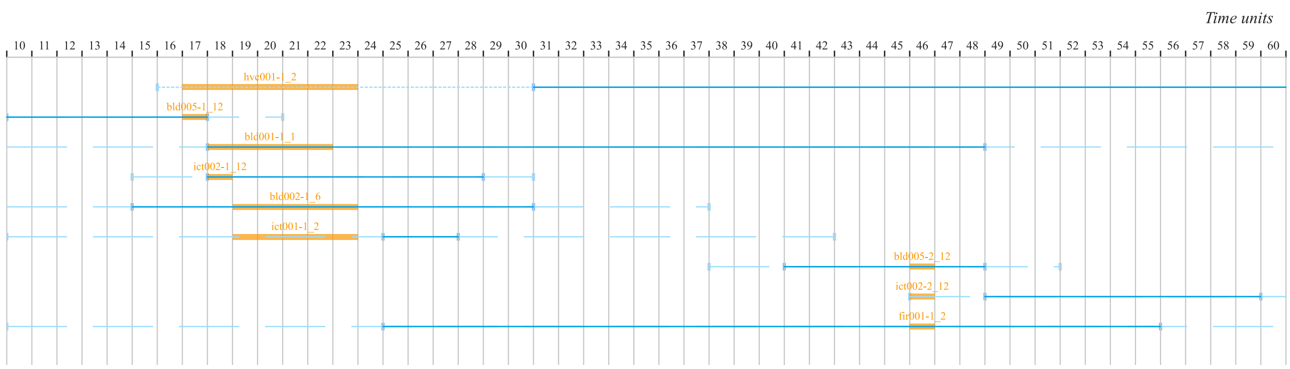

Figure 7. Results of the optimisation visually represented in this cropped view of the diagram, showing time units 10–60

dress this, future iterations of the system could integrate real-time occupancy data sourced from IoT sensors, allowing the model to dynamically adjust scheduling constraints and avoid disruptions during peak usage hours. Additionally, the current constraint on the number of concurrent activities is applied globally, without distinguishing between different room functions. Introducing function-specific limits, for example, stricter restrictions for hallways or receptions during business hours, would enhance the realism and practicality of the schedule.

Another important limitation concerns the treatment of maintenance tasks as independent. While this simplification was deliberate in order to isolate and test the temporal clustering principle, in operational practice tasks often share resources (e.g., staff or equipment) and may generate conflicts if scheduled in spatial proximity. To address this, the same ILP formulation can be readily extended with additional constraints for resource capacities, spatial conflicts, and area-specific activity limits (see Section 2.1.5).

From a technical perspective, the integration with BIM remains limited in scope. Currently, only a small number of parameters are embedded in the BIM model, and spatial constraints, such as adjacency, shared infrastructure, or restricted access zones, are not yet considered in the optimisation. This limits BIM's role to a post-optimisation visualisation tool rather than an active driver of scheduling. Accordingly, future developments should move towards embedding spatial and functional constraints directly from BIM into the optimisation model, leveraging adjacency matrices, access rules, or circulation paths derived from the building geometry. These improvements would significantly enhance the system's adaptability and applicability across diverse operational settings.

The reliance on Autodesk Dynamo also poses interoperability issues, as this restricts usability for teams using alternative BIM platforms. Moreover, the mapping between optimisation data and building elements relies on Revit element IDs, which are susceptible to change during model updates, potentially breaking the data link. A more robust solution would involve persistent, unique identifiers or the use of standardised metadata tags to ensure consistent referencing across software environments.

Finally, the assumptions of uniform cost penalties and static occupancy patterns have not yet been validated against empirical building data. This study relied on realistic but synthetic data to demonstrate the feasibility of the approach. While this is appropriate for a framework, future work should include benchmarking against actual maintenance records and occupancy datasets to test sensitivity to assumption choices and confirm practical relevance.

6. Conclusions

From the state-of-the-art analysis, a need emerges for a novel approach to reducing inefficiencies in the O&M phase, identified as the most expensive stage of the building lifecycle. In response to this, the presented study introduces a methodology for optimising O&M schedules by integrating a mathematical optimisation model within a microservice-based system architecture, supported by a BIM environment.

The proposed approach consolidates maintenance tasks into optimised clusters, effectively reducing their frequency, and thus associated disruption or risks to the occupants. This is achieved while ensuring compliance with task-specific temporal constraints and maintaining cost efficiency. The integration of the optimisation engine with a spreadsheet-based database, BIM tools, and an orchestrator allows for a streamlined workflow from data input through to visualisation and execution within the building model. The case study confirms the system's capability to manage realistic maintenance scenarios, demonstrating both practical feasibility and potential for application in real-world settings.

This system directly affects operations and facility managers. By minimizing maintenance frequency and area coverage, the system lessens building downtime and labour coordination complexity. Spatial integration in the BIM model simplifies planning, especially for multi-use or high-traffic areas. Also, the modular design accommodates different stakeholders working with the system using tools to which they are accustomed (e.g., spreadsheets), supporting easy adoption at no great cost of learning. These functionalities allow managers to make effective, fact-based decisions and improve service continuity and user satisfaction.

It should be noted that the model, as applied here, deliberately emphasises temporal clustering to demonstrate the feasibility of the optimisation–BIM pipeline on a simplified case. The omission of interdependencies such as shared resources or adjacency conflicts does not represent a structural limitation, but rather a scope decision. As shown in Section 2.1.5, the ILP framework can be expanded to include cumulative resource capacities, spatial or operational conflicts, and differentiated area-level constraints, without altering the optimisation core.

In summary, this work establishes a framework foundation for BIM-enabled, optimisation-driven O&M planning. While simplifications were made for clarity, the extensibility of the ILP formulation, the modular microservice design, and the planned empirical validation together ensure that the framework remains scalable, adaptable, and practically relevant.

Acknowledgements

The authors would like to acknowledge Sara Nicoloso for her valuable assistance in the formulation of the ILP model used in this work.

Funding

Author Edoardo De Santis acknowledges the financial support from Project ECS 0000024 Rome Technopole – CUP B83C22002820006, NRP Mission 4 Component 2 Investment 1.5, funded by the European Union – NextGenerationEU; and from Progetti per Avvio alla Ricerca – Tipo 1 – year 2023 – n. AR123188B4D4EA67, funded by Sapienza University of Rome.

Author contributions

De Santis E: Conceptualization, Methodology, Software, Writing – Original Draft, Writing – Review & Editing, Visualization, Project administration, Funding acquisition. Rossini F L: Conceptualization, Writing – Original Draft, Writing – Review & Editing.

Disclosure statement

The authors confirm that all content, figures, and tables in the submitted manuscript work are original works created by the authors. There are no competing financial, professional, or personal interests from other parties.

References

Abuhussain, M. A., Waqar, A., Khan, A. M., Othman, I., Alotaibi, B. S., Althoey, F., & Abuhussain, M. (2024). Integrating Building Information Modeling (BIM) for optimal lifecycle management of complex structures. *Structures*, *60*, Article 105831. https://doi.org/10.1016/j.istruc.2023.105831

Ahmed, R., Nasiri, F., & Zayed, T. (2022). Two-stage predictive maintenance planning for hospital buildings: A multiple-objective optimization-based clustering approach. *Journal of Performance of Constructed Facilities*, *36*(1), Article 04021105. https://doi.org/10.1061/(ASCE)CF.1943-5509.0001691

Ashouri, A., Fux, S. S., Benz, M. J., & Guzzella, L. (2013). Optimal design and operation of building services using mixed-integer linear programming techniques. *Energy*, *59*, 365–376. https://doi.org/10.1016/j.energy.2013.06.053

Dallasega, P., Revolti, A., Follini, C., Schimanski, C. P., & Matt, D. T. (2019). BIM-Based construction progress measurement of non-repetitive HVAC installation works. In *Proceedings of the 27th Annual Conference of the International Group for Lean Construction* (*IGLC*) (pp. 819–830). IGLC.net. https://doi.org/10.24928/2019/0152

Davtalab, O. (2017). Benefits of real-time data driven BIM for FM departments in operations control and maintenance. In *Computing in Civil Engineering 2017* (pp. 202–210). https://doi.org/10.1061/9780784480823.025

Dubey, R., Gunasekaran, A., & Papadopoulos, T. (2024). Benchmarking operations and supply chain management practices using Generative AI: Towards a theoretical framework. *Transportation Research Part E*: *Logistics and Transportation Review*, *189*, Article 103689. https://doi.org/10.1016/j.tre.2024.103689

Gao, X., & Pishdad-Bozorgi, P. (2019). BIM-enabled facilities operation and maintenance: A review. *Advanced Engineering Informatics*, *39*, 227–247. https://doi.org/10.1016/j.aei.2019.01.005

Gnekpe, C., Tchuente, D., Nyawa, S., & Dey, P. K. (2024). Energy performance of building refurbishments: Predictive and prescriptive AI-based machine learning approaches. *Journal of Business Research*, *183*, Article 114821. https://doi.org/10.1016/j.jbusres.2024.114821

Golabchi, A., Akula, M., & Kamat, V. (2016). Automated building information modeling for fault detection and diagnostics in commercial HVAC systems. *Facilities*, *34*(3/4), 233–246. https://doi.org/10.1108/F-06-2014-0050

Gurobi Optimization, LLC. (2024). *Gurobi optimizer reference manual*. https://www.gurobi.com

Horner, R. M. W., El-Haram, M. A., & Munns, A. K. (1997). Building maintenance strategy: A new management approach. *Journal of Quality in Maintenance Engineering*, *3*(4), 273–280. https://doi.org/10.1108/13552519710176881

Hosseinzadeh, A., Frank Chen, F., Shahin, M., & Bouzary, H. (2023). A predictive maintenance approach in manufacturing systems via AI-based early failure detection. *Manufacturing Letters*, *35*, 1179–1186. https://doi.org/10.1016/j.mfglet.2023.08.125

Kelly, G., Serginson, M., Lockley, S., Dawood, N., & Kassem, M. (2013). BIM for facility management: A review and a case study investigating the value and challenges. In *Proceedings of the 13th International Conference on Construction Applications of Virtual Reality* (pp. 191–199), London, UK.

Khan, S., Panuwatwanich, K., & Usanavasin, S. (2023). Integrating building information modeling with augmented reality: Application and empirical assessment in building facility management. *Engineering*, *Construction and Architectural Management*, *31*(7), 2809–2828. https://doi.org/10.1108/ECAM-12-2021-1146

Konig, M., Koch, C., Habenicht, I., & Spieckermann, S. (2012). Intelligent BIM-based construction scheduling using discrete event simulation. In *Proceedings of the 2012 Winter Simulation Conference* (*WSC*), Berlin, Germany. https://doi.org/10.1109/WSC.2012.6465232

Li, A., Xiao, F., Xiao, Z., Yan, R., Li, A., Lv, Y., & Su, B. (2024). Active learning concerning sampling cost for enhancing AI-enabled building energy system modeling. *Advances in Applied Energy*, *16*, Article 100189. https://doi.org/10.1016/j.adapen.2024.100189

Malucelli, F., & Nicoloso, S. (2007). Shiftable intervals. *Annals of Operations Research*, *150*(1), 137–157. https://doi.org/10.1007/s10479-006-0161-1

Nadareishvili, I., Mitra, R., McLarty, M., & Amundsen, M. (2016). *Microservice architecture*: *Aligning principles*, *practices*, *and culture*. O'Reilly Media, Inc.

Ni, M., Luh, P. B., & Moser, B. (2003). An optimization-based approach for distributed project scheduling. In *2003 IEEE International Conference on Robotics and Automation* (*Cat. No.03CH37422*) (Vol. 2, pp. 1756–1761), Taipei, Taiwan. IEEE. https://doi.org/10.1109/ROBOT.2003.1241848

Peng, Y., Lin, J.-R., Zhang, J.-P., & Hu, Z.-Z. (2017). A hybrid data mining approach on BIM-based building operation and maintenance. *Building and Environment*, *126*, 483–495. https://doi.org/10.1016/j.buildenv.2017.09.030

Scaife, A. D. (2024). Improve predictive maintenance through the application of artificial intelligence: A systematic review. *Results in Engineering*, *21*, Article 101645. https://doi.org/10.1016/j.rineng.2023.101645

Schrijver, A. (1998). *Theory of linear and integer programming*. John Wiley & Sons.

Torres-Sainz, R., Lorente-Leyva, L. L., Arbella-Feliciano, Y., Trinchet-Varela, C. A., Pérez-Vallejo, L. M., & Pérez-Rodríguez, R. (2024). Data-enabled Bayesian inference for strategic maintenance decisions in industrial operations. *Data in Brief*, *57*, Article 111058. https://doi.org/10.1016/j.dib.2024.111058

Wettewa, S., Hou, L., & Zhang, G. (2024). Graph Neural Networks for building and civil infrastructure operation and maintenance enhancement. *Advanced Engineering Informatics*, *62*, Article 102868. https://doi.org/10.1016/j.aei.2024.102868

Yussuf, R. O., & Asfour, O. S. (2024). Applications of artificial intelligence for energy efficiency throughout the building lifecycle: An overview. *Energy and Buildings*, *305*, Article 113903. https://doi.org/10.1016/j.enbuild.2024.113903

Zhang, A., Yang, J., & Wang, F. (2023). Application and enabling digital twin technologies in the operation and maintenance stage of the AEC industry: A literature review. *Journal of Building Engineering*, *80*, Article 107859. https://doi.org/10.1016/j.jobe.2023.107859