

ENHANCING EFFICIENCY IN CONSTRUCTION SCHEDULING FOR SMES: A NOVEL MATRIX-BASED VBA-EXCEL MODEL

T. Thai-Phuong¹, L. Nguyen-Son^{2*}

Faculty of Civil Engineering, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam¹²

thaiphuongtruc@iuh.edu.vn¹, nguyensonlam@iuh.edu.vn^{2*}

Received: 17 September 2025, Revised: 31 March 2026, Accepted: 04 April 2026

*Corresponding Author

ABSTRACT

Small and medium-sized enterprises (SMEs) in construction face challenges managing complex project schedules cost-effectively. This study develops a user-friendly, matrix-based VBA-Excel model to enhance scheduling efficiency. The objective is to create an accessible tool that automates task dependency management without requiring advanced coding skills. The model employs the Precedence Diagramming Method (PDM), integrating all dependency types (FS, SS, FF, SF) with lag/lead adjustments through VBA automation in Excel. Implementation involved designing a novel matrix structure for dependency handling and a simple interface, validated via case studies with 11 and 9 tasks, benchmarked against industry standards. Results show accurate schedule calculations matching benchmarks, with seamless adjustments, though processing slows for large projects. The tool enables SMEs to optimize resources, reduce delays, and improve competitiveness at no cost. Limitations include Excel's computational limits and omission of real-world calendar constraints like holidays. Future enhancements could include cloud-based scalability and risk integration. This innovative model fosters inclusive project management, reducing infrastructure disparities in underserved regions.

Keywords : *Construction Scheduling, VBA-Excel, PDM, SMEs, Matrix-Based Model, Dependency Management, Cost-Effective Tool*

1. Introduction

Project scheduling is a critical component of construction management, directly influencing project delivery time, resource utilization, and overall cost efficiency. In an industry characterized by complex interdependencies, varying task durations, and tight deadlines, accurate scheduling plays a vital role in minimizing delays and optimizing project outcomes (Zhao et al., 2021). Globally, inefficiencies in project scheduling lead to significant financial losses, with construction delays estimated to cost the industry billions of dollars annually (K. Wang et al., 2025). These challenges are further exacerbated by external factors such as weather disruptions and accelerated project timelines, particularly in large-scale developments involving intricate task dependencies (Pranata et al., 2023). Small and medium-sized enterprises (SMEs) in the construction sector face additional and more severe difficulties (Al-Dhamad et al., 2025). Recent studies highlight that SMEs often encounter internal constraints, including limited financial resources, inadequate technical expertise, and difficulties in adopting advanced digital tools (Mafundu & Mafini, 2019). Supply chain risks and barriers to digital transformation further intensify these issues, especially in developing economies. Moreover, the slow adoption of modern project scheduling technologies among SMEs has been widely documented (Khan et al., 2024), with organizational readiness and knowledge integration acting as key moderating factors (Purushothaman et al., 2025). These empirical challenges underscore the urgent need for accessible and cost-effective scheduling solutions tailored to resource-constrained environments. Commercial scheduling software such as Microsoft Project and Primavera P6 provide robust features for modeling complex dependencies and constraints (S & George, 2021). However, their high licensing costs — frequently exceeding \$1,000 per user annually — combined with steep learning curves and limited flexibility for small-scale projects, make them largely inaccessible to most SMEs (Mafundu & Mafini, 2019). While manual scheduling methods remain economical, they are highly time-consuming and prone to errors when handling modern dependency types such as Start-to-Start (SS), Finish-to-Finish (FF), and Start-to-Finish (SF) relationships with lag and lead times. Although some low-cost digital solutions and business intelligence tools have

been proposed for cost management in SMEs, they typically lack comprehensive support for full Precedence Diagramming Method (PDM) functionalities. Similarly, comparative studies on spreadsheet tools show clear advantages in accessibility but reveal limitations in handling sophisticated scheduling requirements. This study identifies a critical research gap: existing Excel-based scheduling tools often support only Finish-to-Start (FS) relationships and lack full integration of all four dependency types (FS, SS, FF, SF) with lag/lead adjustments (Wan & He, 2004). More advanced matrix-based approaches, such as the Design Structure Matrix (DSM), tend to require specialized knowledge that is impractical for typical SME users. Furthermore, despite growing interest in dynamic scheduling factors and digital transformation in construction (Nicholas et al., 2024), few studies have delivered a practical, no-cost, and user-friendly tool that combines Excel's widespread availability with comprehensive PDM automation (Siriwardhana et al., 2025). To address these limitations, this study proposes a novel matrix-based VBA-Excel scheduling model. The novelty of this work lies in the development of an improved relationship matrix that utilizes regular expression parsing for flexible dependency input handling and diagonal storage optimization for enhanced computational efficiency, features that previous Excel-based solutions have not adequately addressed. By fully supporting all PDM dependency types with lag and lead times, the proposed model offers a practical and scalable solution specifically designed for SMEs in the construction industry.

The main objectives of this study are as follows: To develop an automated VBA-Excel model capable of performing complete PDM calculations using a matrix-based approach. To validate the accuracy of the model through comparison with established industry benchmarks, including *Schedula Anima* and Microsoft Project. To evaluate the scalability and practical applicability of the model for typical SME construction projects.

2. Literature Review

Project scheduling is fundamental to construction management, shaping timelines, resource allocation, and cost efficiency. Effective scheduling ensures proper task sequencing, dependency management, and optimal resource use to meet deadlines. The Precedence Diagramming Method (PDM) and Critical Path Method (CPM) are widely adopted, offering structured frameworks to identify critical paths and compute parameters like Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF), Free Float (FF), and Total Float (TF). These methods are applied in infrastructure projects, such as road construction and multipurpose buildings, to manage interlinked activities and optimize durations (Arditi & Bentotage, 1996). Project scheduling constitutes a core theoretical foundation in construction management, directly shaping project timelines, resource allocation, and cost performance. Within established project management frameworks such as the Project Management Body of Knowledge (PMBOK), the Precedence Diagramming Method (PDM) and Critical Path Method (CPM) serve as primary techniques for modeling task interdependencies and determining the critical path [4]. PDM represents activities as nodes connected by four types of logical relationships: Finish-to-Start (FS), Start-to-Start (SS), Finish-to-Finish (FF), and Start-to-Finish (SF). These relationships can be further adjusted using lag (positive delay) or lead (negative advance) times. Key scheduling parameters calculated through forward and backward passes include Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF), Free Float (FF), and Total Float (TF). These metrics enable project managers to assess scheduling flexibility and optimize resource utilization in complex construction environments. Commercial tools like Microsoft Project are industry standards due to their robust handling of complex dependencies. However, high licensing costs and steep learning curves make them inaccessible for small to medium-sized enterprises (SMEs). Commercial scheduling tools, particularly Microsoft Project and Primavera P6, are widely regarded as industry standards due to their sophisticated capability in handling multiple dependency types and resource constraints. However, their practical adoption by small and medium-sized enterprises (SMEs) remains severely limited. High annual licensing fees, often exceeding \$1,000 per user, combined with steep learning curves and the need for specialized training, render these tools largely unaffordable and inaccessible for most SMEs. Manual PDM methods, while cost-effective, are time-consuming and error-prone, especially for projects with numerous tasks and intricate dependencies like Start-to-Start (SS) or Finish-to-Finish (FF). Studies, such as those on the Surabaya Outer East Ring

Road, highlight manual methods' struggles with such complexities, limiting their effectiveness in modern construction. In contrast, manual PDM implementation, although cost-effective, proves highly time-consuming and error-prone when managing numerous tasks with intricate dependencies such as SS, FF, and SF relationships (Schkarin & Dobhan, 2022). Previous attempts to develop low-cost scheduling solutions have primarily relied on spreadsheet platforms. Several Excel-based tools with basic macros have been proposed; yet, most are restricted to Finish-to-Start (FS) relationships and offer limited or no support for lag/lead adjustments and advanced dependency types (SS, FF, SF) (Amer et al., 2021). Matrix-based approaches, such as the Design Structure Matrix (DSM), have been explored to improve dependency visualization and management. Nevertheless, DSM typically demands specialized technical knowledge, reducing its practicality for non-expert users in SMEs (Purushothaman et al., 2025). SMEs face significant challenges in adopting efficient scheduling due to financial and technical constraints, often relying on cumbersome manual approaches. Commercial tools lack adaptability for specific project needs, further hindering SMEs. To address this, research has explored matrix-based methods like the Design Structure Matrix (DSM) for improved dependency management. However, DSM requires specialized knowledge, reducing its accessibility for smaller firms. Excel-based solutions with basic macros have been proposed, but they often lack support for advanced dependencies (SS, FF, SF) and lag times, limiting their utility in diverse projects. Recent literature has increasingly focused on digital transformation and advanced scheduling techniques in construction. Studies have examined the application of Building Information Modelling (BIM) for scheduling and material management (Sapitri et al., 2025), dynamic scheduling factors (Ramahwati & Osly, 2024), stochastic project network techniques using GERT (Pregina & Ramesh Kannan, 2022), and Monte Carlo simulation for project acceleration probability (Alfandi Rizki Pratama et al., 2025). Optimization methods, including genetic algorithms (GA) for time-cost-quality-resource trade-offs (Elkliny et al., 2025), modified critical chain scheduling (Mohamed Meabed et al., 2025), and finance-based scheduling using MILP, GA, and PSO (Fares et al., 2025), have also demonstrated potential for improving project performance. Additionally, lean construction principles have been promoted to enhance scheduling efficiency and waste reduction (Bidhendi et al., 2025). Despite these advancements, a critical synthesis of the literature reveals persistent limitations. While BIM-integrated and optimization-driven approaches significantly improve scheduling accuracy and visualization, they often require substantial investment in software, training, and hardware — barriers that are particularly prohibitive for resource-constrained SMEs (Xu, 2025). Similarly, although dynamic and stochastic methods address uncertainty effectively, they rarely provide accessible, no-cost implementations suitable for everyday use by small firms. Low-cost digital solutions and business intelligence tools have been suggested for proactive cost management [33]; however, they generally lack comprehensive PDM functionality with full dependency support. Literature reveals critical gaps: scalable, no-cost tools for large projects are scarce, real-time data integration and sustainability considerations are rarely addressed, and user-friendly interfaces for non-expert users are lacking, a significant issue for SMEs. Recent studies emphasize optimizing resource-constrained scheduling for multi-project environments and incorporating time-cost trade-off models or advanced techniques like particle swarm optimization to enhance efficiency (Isah & Kim, 2021). SMEs in the construction industry continue to face multifaceted challenges. Internal constraints, such as limited capital and technical capacity (Le et al., 2025), coupled with broader issues of digital transformation readiness (Nicholas et al., 2024), knowledge management in AI adoption (Goh & Lee, 2023), and supply chain risks (Chen et al., 2025)

, further hinder the effective adoption of advanced scheduling tools. Post-COVID studies also highlight accelerated digitalisation needs, yet implementation remains slow among smaller contractors due to skill gaps and organizational barriers. To ensure a comprehensive and up-to-date review, literature was systematically sourced from Scopus and Web of Science databases using keywords including “construction scheduling SMEs”, “Excel VBA PDM”, “matrix-based scheduling”, “digital transformation construction SMEs”, and “low-cost project scheduling tools”. Only peer-reviewed journal articles and high-quality conference proceedings published primarily between 2020 and 2026 were considered. These gaps highlight the need for affordable, user-friendly tools that support all PDM relationships, lag times, and scalability. In summary, the

reviewed literature exposes a significant research gap: the absence of an affordable, user-friendly, and scalable scheduling tool that fully supports all four PDM dependency types with lag/lead adjustments while remaining accessible to non-expert SME users. This study addresses this gap by introducing a novel matrix-based VBA-Excel model that integrates regular expression parsing for flexible dependency handling and optimized diagonal matrix storage, providing a practical, cost-free alternative that overcomes the limitations of both commercial software and existing Excel-based solutions.

3. Research Methods

3.1. Overview of the Precedence Diagramming Method (PDM)

The Precedence Diagramming Method (PDM) is a widely used technique in construction project management for modeling complex task interdependencies. Tasks are represented as nodes in a network diagram, connected by directed arrows indicating four dependency types: Finish-to-Start (FS), Start-to-Start (SS), Finish-to-Finish (FF), and Start-to-Finish (SF). PDM represents activities as nodes connected by four types of logical relationships: Finish-to-Start (FS) — the successor activity cannot start until the predecessor finishes; Start-to-Start (SS) — the successor can start once the predecessor has begun; Finish-to-Finish (FF) — the successor cannot finish until the predecessor finishes; and Start-to-Finish (SF) — the successor cannot finish until the predecessor has started. These relationships can be further modified by lag (positive value for delay) or lead (negative value for acceleration), allowing realistic modeling of overlapping or accelerated tasks commonly encountered in construction projects. Key parameters—Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF), Free Float (FF), and Total Float (TF)—assess scheduling flexibility to prevent delays. In this study, PDM calculations are automated using Visual Basic for Applications (VBA) in Microsoft Excel, streamlining scheduling for construction projects. The process involves Forward Pass (for ES, EF) and Backward Pass (for LS, LF), followed by critical path identification. VBA enhances PDM's efficiency, supporting stochastic scheduling for task duration uncertainties (Zhang & Xu, 2016), simplifying complex project management for SMEs.

3.2. Computational Procedure

This study implements an automated scheduling process for the Precedence Diagramming Method (PDM), computing key parameters: Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF), Free Float (FF), and Total Float (TF). Using Visual Basic for Applications (VBA) in Microsoft Excel, the model provides a cost-effective, user-friendly solution for construction project planning, tailored for small to medium-sized enterprises (SMEs) in resource-constrained settings. Unlike previous Excel-based tools limited by complex dependencies, this model supports all dependency types (FS, SS, FF, SF) with lag/lead times, addressing a key gap in accessible scheduling solutions (Doukari et al., 2022). The VBA-Excel platform was deliberately chosen for its widespread availability, zero licensing cost, and ease of use among SMEs. This choice aligns with recent findings on the advantages of spreadsheet tools and ongoing digitalisation trends in the construction industry (Rosener, 2024). The framework includes two main components: the `CreateRelationshipMatrix` function, which builds a matrix to capture task dependencies, and the `ProjectScheduleMatrix` Sub, which performs PDM calculations and exports results to Excel, enhancing usability for non-experts. This scalable, matrix-based approach ensures accurate scheduling, with workflows detailed in Figures 1 and 2.

3.2.1. Building the Relationship Matrix

The initial step in the scheduling process involves constructing a dependency matrix to represent the interdependencies between tasks. This matrix is generated by the `CreateRelationshipMatrix` function, which processes input data such as task IDs, preceding tasks, dependency types (Finish-to-Start, Start-to-Start, Finish-to-Finish, Start-to-Finish), lag/lead values, and durations. The operational workflow of this function is visually depicted in Figure 1, which outlines the steps involved in constructing the task dependency structure.

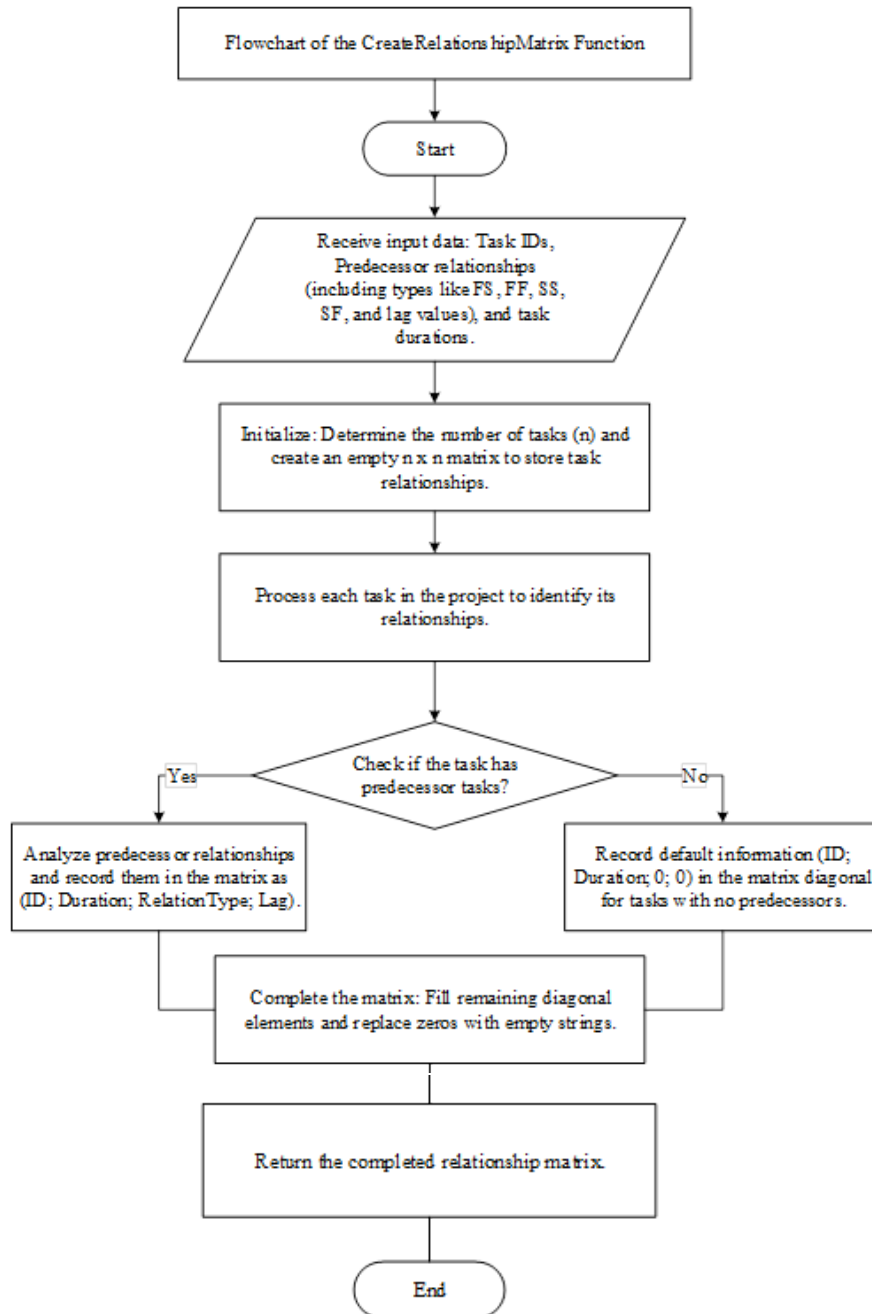


Fig. 1. Flowchart of the CreateRelationshipMatrix function, illustrating the process of constructing the relationship matrix for task dependencies. The flowchart depicts the steps of initializing the matrix, mapping task IDs, parsing predecessor strings, and recording dependency relationships.

The CreateRelationshipMatrix function begins by initializing a dependency matrix of size $n \times n$, where n is the number of tasks, determined by the maximum task ID. It employs a dictionary (IDDict) to effectively map task IDs to their corresponding row indices, minimizing computational overhead compared to conventional iterative searches. For each task, the function assesses the presence of preceding tasks. If none exist, the task's default details (ID, Duration, 0, 0) are recorded on the matrix diagonal, adhering to standard PDM initialization practices. When preceding tasks are present, the function extracts dependency details from each predecessor string using regular expressions, identifying the preceding task ID, relationship type, and lag/lead value. This method excels in processing formatted input, ensuring accurate extraction of dependency details across varied project scenarios. The dependency is then recorded in the matrix as a tuple (ID, Duration, Relationship Type, Lag), facilitating precise scheduling calculations in subsequent steps. A key innovation in this implementation is the use of regular expressions, which

significantly enhances the flexibility and accuracy of dependency processing compared to traditional VBA approaches (Aslan & Türkakın, 2022). The detailed logic for constructing the task dependency structure, including the extraction of dependency details, is outlined in Algorithm 1: CreateRelationshipMatrix Algorithm, which delineates the steps of matrix initialization, task ID mapping, dependency processing, and structured recording of relationships. This matrix-based design offers superior efficiency and clarity over list-based predecessor methods, particularly when handling complex and multiple dependencies.

Algorithm: CreateRelationshipMatrix Algorithm

Input: ID, Predecessors, Duration, Optional ReturnColumn
Output: Relationship Matrix or Specific Column

begin
 set n as maximum value in ID
 initialize $Matrix$ as $n \times n$ array
 initialize $IDDict$ for ID-to-row mapping
for i from 1 to n **do**
 if $Predecessors[i]$ is empty **then**
 set $Matrix[i, i]$ to $(ID[i], Duration[i], 0, 0)$
 else
 for each $Pred$ in $Predecessors[i]$ split by ", " **do**
 parse $Pred$ with pattern $^(+)([A-Z]{2})?(-)?(+)?\$$ into
 $PredID$, $Type$ (default "FS"), Lag (default 0)
 if $PredID$ exists in $IDDict$ **then**
 set j to $IDDict[PredID]$
 set $Matrix[j, i]$ to $(ID[i], Duration[i], Type, Lag)$
 end
 end
end
end
for i from 1 to n **do**
 if $Matrix[i, i]$ is empty **then**
 set $Matrix[i, i]$ to $(ID[i], Duration[i], " - ", 0)$
 end
end
if $ReturnColumn$ is 1 to 4 **then**
 extract $ResultMatrix$ from $Matrix$'s $ReturnColumn$ -th component
 return $ResultMatrix$
else
 return $Matrix$
end

Algorithm 1. CreateRelationshipMatrix Algorithm, illustrating the process of constructing the relationship matrix for task dependencies, including parsing of predecessor strings to extract dependency information (Predecessor ID, Relationship Type, Lag/Lead Value).

This method effectively manages all dependency types, addressing the shortcomings of manual PDM implementations that often falter with intricate dependencies. Upon processing all tasks, the CreateRelationshipMatrix function finalizes the dependency matrix by filling the remaining diagonal elements and replacing zero values with empty strings to enhance storage efficiency and clarity, ensuring the matrix remains user-friendly for project managers, particularly those in SMEs. This structured task dependency framework supports subsequent PDM calculations by providing a clear foundation, consistent with established matrix-based scheduling methodologies, and aligns with the study's goal of delivering a scalable and accessible scheduling solution for construction projects. An example of this matrix for the sample project in Case Study 2 is provided in Table A1 in Appendix A, which illustrates the dependency relationships for the 9 tasks, including tuples like (2; 15; FS; -10) for Task 2's relationship with Task 1.

3.2.2. Forward Pass - Calculating ES and EF

The initial scheduling phase, known as the Forward Pass, calculates the Early Start (ES) and Early Finish (EF) for each task, progressing through the task network from the starting point to the end. This phase is executed within the ProjectScheduleMatrix Sub, ensuring automated and efficient computation of task timing metrics, particularly beneficial for small to medium-sized enterprises (J. Wang et al., 2017). The detailed workflow of this phase is illustrated in Figure 2,

which provides a comprehensive overview of the steps involved in calculating task timing metrics and exporting results.

The forward calculation phase begins with the first task, defined as the one with no preceding tasks, and proceeds progressively from left to right through the dependency network. The calculation of ES and EF varies based on the interdependency type, reflecting the diverse timing relationships between tasks. For instance, in a Finish-to-Start (FS) relationship, a task cannot begin until its preceding task is complete, while a Start-to-Start (SS) relationship allows tasks to commence concurrently. The process is structured as follows:

Step 1: Initialization

For the first task (with no predecessors), set $ES = 0$

Compute (EF) as:

$$EF = ES + Duration \tag{1}$$

where Duration is the task duration. This initialization ensures a defined starting point for the scheduling process.

Step 2: Compute ES and EF for Remaining Tasks

The Forward Pass iterates through each task (j), considering its predecessor tasks (i). The relationship matrix generated by *CreateRelationshipMatrix* is used to determine the dependency type and lag/lead value. The ES of task (j) is calculated as follows:

FS (Finish-to-Start):

$$ES_j = EF_i + Lag \tag{2}$$

SS (Start-to-Start):

$$ES_j = ES_i + Lag \tag{3}$$

FF (Finish-to-Finish):

$$EF_j = EF_i + Lag, \text{ then } ES_j = EF_j - Duration_j \tag{4}$$

SF (Start-to-Finish):

$$EF_j = ES_i + Lag, \text{ then } ES_j = EF_j - Duration_j \tag{5}$$

If task (j) has multiple predecessors, take the maximum $ES_j = 0$

$$ES_j = \max(ES_j \text{ from all predecessors}) \tag{6}$$

Once ES_j is determined, compute EF_j :

$$EF_j = ES_j + Duration_j \tag{7}$$

Step 3: Determine Project Duration

For example, in Case Study 2, Task 2 with a 10-day lead (1FS-10) correctly yields an Early Start of day -5, demonstrating the model's accurate handling of lead times:

$$\text{Project Duration} = \max(EF_j) \text{ for all tasks } j \tag{8}$$

The logic for this phase, which calculates ES and EF based on interdependency types as described in Equations (2)–(7), forms a critical component of Algorithm 2: ProjectScheduleMatrix Algorithm, which outlines the complete process of determining task timing metrics and exporting results. This automated approach within the ProjectScheduleMatrix Sub enhances efficiency, enabling seamless handling of diverse interdependencies for SMEs, and ensures accurate timing calculations, aligning with the study's goal of providing an accessible and scalable scheduling solution for construction projects.

□ **Algorithm:** ProjectScheduleMatrix Algorithm

Input: ID, Predecessors, Duration
Output: Scheduling Matrices (ES, EF, LS, LF, FF, TF) and Diagonal Results

```

begin
// Initialize matrices
set n as maximum task ID
initialize ES, EF, LS, LF, FF, TF as  $n \times n$  arrays
initialize diagonal arrays ES_Diag, EF_Diag, LS_Diag, LF_Diag,
FF_Diag, TF_Diag
// Build relationship matrices
(ID, Relationship, Duration, Lag) ←
CreateRelationshipMatrix(ID, Predecessors, Duration)
// Forward Pass: Compute ES and EF
for j from 1 to n do
  for i from 1 to j do
    if Relationship[i, j] exists then
      compute ES[i, j], EF[i, j] based on dependency type (FS, SS,
      FF, SF) and lag
      update ES_Diag[j], EF_Diag[j]
    end
  end
  update ProjectDuration ← max(ProjectDuration, EF[j, j])
end
// Backward Pass: Compute LS, LF, FF, and TF
for i from n down to 1 do
  for j from i down to 1 do
    if Relationship[i, j] exists then
      compute LS[i, j], LF[i, j] based on dependency type and lag
      compute FF[i, i] based on successors' early dates
      compute TF[i, i] ← LF[i, i] - EF[i, i]
      update LS_Diag[i], LF_Diag[i], FF_Diag[i], TF_Diag[i]
    end
  end
end
// Export results to Excel
export ID, Relationship, Duration, Lag, ES, EF, LS, LF, FF, TF matrices
export diagonal results ES_Diag, EF_Diag, LS_Diag, LF_Diag, FF_Diag,
TF_Diag
end

```

Algorithm 2. ProjectScheduleMatrix Algorithm, outlining the steps for calculating scheduling parameters (ES, EF, LS, LF, FF, TF) and exporting the results.

3.2.3. Backward Pass - Calculating LS and LF

The reverse scheduling phase, known as the Backward Pass, determines the Late Start (LS) and Late Finish (LF) for each task, progressing in reverse order through the task network from the final task to the starting point. This phase is seamlessly integrated into the ProjectScheduleMatrix Sub, building on the results of the Forward Pass to ensure a cohesive scheduling process for construction projects, particularly for small to medium-sized enterprises (SMEs) (Shalaby & Ezeldin, 2022). The operational workflow of this phase is depicted in Figure 2, which provides a comprehensive overview of the steps involved in determining task timing metrics and exporting results.

The Backward Pass starts from the last task and moves backward through the dependency network, calculating the latest possible times without delaying the overall project completion. For instance, in a Finish-to-Start (FS) relationship, the latest finish time of a task directly influences

the start of its following task, while a Finish-to-Finish (FF) relationship ties the completion times of both tasks. The process is structured as follows

Step 1: Initialization

For the last task, which has no following tasks, set:

$$LF = \text{Project Duration} \tag{9}$$

Then, calculate LS as:

$$LS = LF - \text{Project Duration} \tag{10}$$

This initialization ensures alignment with the project’s completion time, establishing a reference point for the reverse scheduling process.

Step 2: Compute LS and LF for Remaining Tasks

The backward calculation phase iterates through each task (*i*), evaluating its following tasks (*j*). The dependency matrix generated by the CreateRelationshipMatrix function is used to identify the interdependency type and lag/lead value. The LF for task (*i*) is determined as follows: Finish-to-Start (FS):

$$LF_i = LS_j - \text{Lag} \tag{11}$$

Start-to-Start (SS):

$$LS_i = LS_j - \text{Lag}, \text{ then } LF_i = LS_i + \text{Duration}_i \tag{12}$$

Finish-to-Finish (FF):

$$LF_i = LF_j - \text{Lag}, \text{ then } LS_i = LF_i - \text{Duration}_i \tag{13}$$

SF (Start-to-Finish):

$$LS_i = LF_j - \text{Lag}, \text{ then } LF_i = LS_j + \text{Duration}_i \tag{14}$$

If task (*i*) has multiple successors, take the minimum

$$LF_i = \min(LF_i \text{ from all successors}) \tag{15}$$

Once LF_i is determined, compute LS_i :

$$LS_i = LF_i - \text{Duration}_i \tag{16}$$

For example, in Case Study 2, the Backward Pass correctly calculates the Late Finish and Late Start values that align perfectly with Microsoft Project results, even with complex dependencies such as SF with lead times. This phase ensures precise determination of the latest allowable start and finish times (late dates), which are critical for assessing scheduling flexibility through float calculations. These steps are seamlessly integrated into Algorithm 2: ProjectScheduleMatrix Algorithm, ensuring a cohesive process for computing all task timing metrics. This automated implementation within the ProjectScheduleMatrix Sub streamlines the process, enabling efficient management of scheduling constraints for SMEs, and ensures precise timing calculations, aligning with the study’s objective of delivering a scalable and accessible scheduling solution for construction projects.

3.2.4. Calculating Free Float (FF) and Total Float (TF)

Free Float (FF)

The Free Float of task (*i*) is the amount of time the task can be delayed without affecting the early start (ES) of any successor task. The FF is calculated based on the dependency type with each successor task (*j*):

FS (Finish-to-Start):

$$FF_i = ES_j - \text{Lag} - ES_i \tag{17}$$

SS (Start-to-Start):

$$FF_i = ES_j - \text{Lag} - ES_i \quad (18)$$

FF (Finish-to-Finish):

$$FF_i = EF_j - \text{Lag} - EF_i \quad (19)$$

SF (Start-to-Finish):

$$FF_i = EF_j - \text{Lag} - ES_i \quad (20)$$

If task (i) has multiple successors, take the minimum

$$LF_i = \min(FF_i \text{ from all successors}) \quad (21)$$

If $FF_i < 0$, set $FF_i = 0$. It is set to 0 to ensure practical applicability.

Total Float (FF):

The Total Float of task (i) is the amount of time the task can be delayed without delaying the overall project completion:

$$TF_i = LF_i - EF_i \quad (22)$$

Alternatively:

$$TF_i = LS_i - ES_i \quad (23)$$

Both formulas yield the same result, providing a measure of scheduling flexibility. These calculations are also part of Algorithm 2, ensuring all parameters are computed systematically (Hu et al., 2021). These float calculations are essential for identifying scheduling flexibility and supporting project managers in making informed decisions on resource allocation and delay mitigation.

3.2.5. Integration in the ProjectScheduleMatrix Sub

The ProjectScheduleMatrix Sub integrates the preceding steps—Forward Pass, Backward Pass, and float calculations—to determine all task timing metrics, including Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF), Free Float (FF), and Total Float (TF), before outputting the results to an Excel worksheet. This module ensures a streamlined scheduling process by combining these phases into a cohesive workflow, as outlined in Algorithm 2: ProjectScheduleMatrix Algorithm, which provides a comprehensive framework for automating the entire computation and output process (Xie et al., 2021).

This subroutine serves as the core engine of the model, sequentially executing the CreateRelationshipMatrix function followed by the forward pass, backward pass, float calculations, and critical path identification. It optimizes memory usage by storing only the diagonal elements of the scheduling matrices (ES_Diag, EF_Diag, LS_Diag, LF_Diag, FF_Diag, and TF_Diag), significantly reducing processing demands and making the tool suitable for resource-constrained environments typical of SMEs. The output process, detailed in Algorithm 3: Exporting Scheduling Results, generates matrices such as those presented in Tables A1–A7 in Appendix A, ensuring that the results are transparent and easily interpretable for project managers.

A significant advantage of this integrated implementation is its seamless connection with Microsoft Excel, allowing non-expert users to input project data in a familiar spreadsheet format and receive comprehensive scheduling outputs without the need for specialized software.

Algorithm: ExportResults Algorithm

Input: wsResult, n, ES_Diag, EF_Diag, LS_Diag, LF_Diag, FF_Diag, TF_Diag
Output: Diagonal Scheduling Parameters Exported to Excel Worksheet
begin
// Export diagonal results for scheduling parameters
set range [1, n + 2] to [n, n + 2] of wsResult to transposed ES_Diag
set range [1, n + 3] to [n, n + 3] of wsResult to transposed EF_Diag
set range [1, n + 4] to [n, n + 4] of wsResult to transposed LS_Diag
set range [1, n + 5] to [n, n + 5] of wsResult to transposed LF_Diag
set range [1, n + 6] to [n, n + 6] of wsResult to transposed FF_Diag
set range [1, n + 7] to [n, n + 7] of wsResult to transposed TF_Diag
end

Algorithm 3. Exporting Scheduling Results in the ProjectScheduleMatrix Algorithm

3.2.6. Novelty and Creativity

The proposed scheduling framework introduces several contributions to construction project management by addressing key limitations in existing tools and methods. By combining advanced computational techniques with the accessibility of Excel, this approach enhances scheduling efficiency and reduces reliance on costly commercial software (Datta et al., 2023).

The key novel aspects of this framework include:

- **All-Encompassing Dependency Management:** Unlike prior Excel-based tools that often focus solely on Finish-to-Start (FS) relationships, this methodology manages all dependency types (FS, SS, FF, SF) along with lag/lead values, filling a critical gap in accessible scheduling solutions.
- **Advanced Parsing with Regular Expressions:** The adoption of regular expressions in the CreateRelationshipMatrix function enhances the flexibility and accuracy of processing complex task interdependencies, offering a significant improvement over conventional VBA-based approaches.
- **Practical and Optimized Output:** The output process generates both detailed relationship and timing matrices as well as simplified diagonal results (e.g., Tables A1–A7 in Appendix A), ensuring that task timing metrics are both comprehensive and easily interpretable for non-expert users.
- **Memory-Efficient and Expandable Design:** By storing only diagonal elements and leveraging Excel's capabilities, the model can accommodate a large number of tasks, limited primarily by hardware resources, making it a practical and scalable solution for SMEs operating in resource-limited environments.

This matrix-based VBA-Excel model provides a reliable, cost-free alternative that overcomes many limitations of both manual methods and commercial software, offering SMEs an accessible tool for effective construction project scheduling.

3.3. Determining the Critical Path

The critical path is defined as the longest sequence of dependent activities with zero total float (TF = 0), meaning any delay in these tasks will directly extend the overall project completion time. The critical path, defined as the chain of activities with zero scheduling flexibility (TF=0), ensures that any postponement in these tasks directly extends the project timeline. To determine the path of critical tasks, the process begins with the initial task and traces the chain of activities with no scheduling flexibility until the final task is reached. This identification process is seamlessly integrated into the ProjectScheduleMatrix Sub (see Algorithm 2). It automatically traces activities with TF = 0 after completing the forward and backward passes, using the computed timing metrics to construct the critical path. This step builds on the task timing metrics to provide a cohesive scheduling solution (Abuhasel, 2023). The scheduling outcomes, such as those shown in Table 4, which summarizes the metrics for the sample project, and Table A7 in Appendix A, highlight the critical tasks with zero TF alongside their timing metrics (ES, EF, LS,

LF, FF, and TF). This automated process within the ProjectScheduleMatrix Sub simplifies a traditionally complex and error-prone manual task, making critical path identification accessible to non-expert users. It enables project managers, especially in small to medium-sized enterprises (SMEs), to prioritize critical tasks effectively, thereby supporting timely project completion and better resource allocation decisions.

4. Results and Discussions

4.1. Case Study 1: Comparison with Schedula Anima

This section evaluates the accuracy of the proposed VBA-Excel scheduling model by comparing its determined task timing metrics with those of Schedula Anima, a dynamic visualization tool developed by Maravas and Pantouvakis (Maravas & Pantouvakis, 2025). While Schedula Anima enhances schedule perception through animated Gantt charts and resource usage graphs, the proposed model focuses on streamlining Precedence Diagramming Method (PDM) calculations within Microsoft Excel, with particular emphasis on accessibility and cost-effectiveness for small to medium-sized enterprises (SMEs). This case study utilizes a dataset from the sample construction project (detailed in Appendix A) to assess the key metrics—Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF), Free Float (FF), and Total Float (TF)—computed by both tools.

4.1.1. Project Data and Setup

The dataset for this evaluation is a construction project activity network adapted from a prior study, as detailed in Table 1. This dataset includes 11 activities (A to K), each with descriptions, predecessors, successors, durations, and resource requirements, resulting in a total duration of 280 days along the critical path; all dependencies between tasks are set to Finish-to-Start (FS) with no lag or lead times, reflecting standard PDM assumptions and aligning with the visualization tool’s baseline settings, thus ensuring an objective evaluation to validate the proposed model’s accuracy for SME users.

Table 1 - Project Activity Network Data (Source: Maravas and Pantouvakis, 2025)

ID	Activity	Description	Predecessors	Successors	Duration (Days)
1	A	Utility relocation		B, C, D	30
2	B	Tunnel Portals	A	E, F	30
3	C	Culverts	A	F, H	60
4	D	Earthworks	A	H	130
5	E	Primary Support	B	G	90
6	F	Retaining walls	B, C	J, K	40
7	G	Tunnel lining	E	I	30
8	H	Pavement	C, D	J	60
9	I	Electromechanical	G	K	40
10	J	Asphalt	F, H	K	30
11	K	Signage	F, I, J		30

The proposed model handles this data using the ProjectScheduleMatrix Sub (as outlined in Algorithm 2), which streamlines standard PDM calculations—including the Forward Pass, Backward Pass, and float calculations—as described in Section 3.2. In contrast, Schedula Anima employs a time sweep simulation method that models schedule progression between early and late dates. This comparison highlights the proposed model’s ability to deliver accurate results comparable to an established visualization tool, while maintaining simplicity and accessibility for SME users.

4.1.2. Comparison of Scheduling Parameters

The task timing metrics determined by the proposed model are evaluated against those reported by Schedula Anima (abbreviated as ‘SA’ in Table 2), presenting the Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF), Free Float (FF), and Total Float (TF) for each activity side by side, along with the critical path and project duration identified by both tools.

Table 2 - Comparison of scheduling parameters between VBA-Excel model and Scheduling Anima

ID	Activity	Predecessors	ES (VBA/ SA)	EF (VBA/ SA)	LS (VBA/ SA)	LF (VBA/ SA)	FF (VBA/S A)	TL (VBA/S A)
1	A - Utility relocation		0/0	30/30	0/0	30/30	0/0	0/0
2	B - Tunnel Portals	1	30/30	60/60	60/60	90/90	0/0	30/30
3	C - Culverts	1	30/30	90/90	100/100	160/160	0/0	70/70
4	D - Earthworks	1	30/30	160/160	30/30	160/160	0/0	0/0
5	E - Primary Support	2	60/60	150/150	90/90	180/180	0/0	30/30
6	F - Retaining walls	2, 3	90/90	130/130	180/180	220/220	90/90	90/90
7	G - Tunnel lining	5	150/150	180/180	180/180	210/210	0/0	30/30
8	H - Pavement	3, 4	160/160	220/220	160/160	220/220	0/0	0/0
9	I - Electromechanical	7	180/180	220/220	210/210	250/250	30/30	30/30
10	J - Asphalt	6, 8	220/220	250/250	220/220	250/250	0/0	0/0
11	K - Signage	6, 9, 10	250/250	280/280	250/250	280/280	0/0	0/0

As shown in Table 2, the proposed VBA-Excel model achieved 100% consistency with Scheduling Anima across all scheduling parameters. Both tools identified the same critical path: A → D → H → J → K, with a total project duration of 280 days. Tasks on this critical path exhibit zero total float (TF=0), indicating that any delay in these activities will directly extend the project completion time. This perfect match validates the correctness of the model’s Forward Pass, Backward Pass, and float calculation algorithms, even when limited to simple Finish-to-Start dependencies. The results confirm the reliability of the proposed model for SME users, offering a cost-effective and transparent alternative to more complex visualization tools, while maintaining ease of use for non-expert users.

4.2. Case Study 2: Comparison with Microsoft Project

This evaluation assesses the proposed VBA-Excel scheduling model by comparing its results against those obtained from Microsoft Project, a widely used commercial scheduling tool in the construction industry (Elazouni & Abido, 2014). Unlike Case Study 1, which was limited to simple Finish-to-Start (FS) relationships without lag or lead times, this case study incorporates all four dependency types (FS, SS, FF, SF) along with various lag and lead adjustments. This allows a more rigorous test of the model’s versatility and its ability to handle complexities commonly found in real-world construction projects. Microsoft Project serves as the primary benchmark due to its robust handling of such dependencies, providing a thorough assessment of the proposed model’s accuracy and practical applicability (Shah & Chandragade, 2023).

4.2.1. Project Data, Setup, and Practical Implications

The dataset for this evaluation is a hypothetical construction project comprising 9 tasks (Task 1 to Task 9), spanning 32 days from December 27, 2024, to January 28, 2025, as detailed in Table 3. Designed using Microsoft Project, the dataset intentionally incorporates complex dependency types—Finish-to-Start (FS), Start-to-Start (SS), Finish-to-Finish (FF), and Start-to-Finish (SF)—along with lag/lead times (e.g., “1FS-10” for Task 2, indicating a 10-day lead; “1SS+2, 2SF+3” for Task 3, combining a 2-day lag and a 3-day lag) to rigorously benchmark the proposed VBA-Excel model against a leading commercial scheduling tool. The dataset includes task names, durations, predecessors, early/late dates, free float (FF), and total float (TF), testing the model’s ability to handle intricate dependencies typical in real-world construction scenarios (Srinath & Varghese, 2023).

Table 3 - Raw Scheduling Data from MS Project for Project Tasks

Task Name	Duration (days)	Predecessors	Early Start (date)	Early Finish (date)	Late Start (date)	Late Finish (date)	Free Slack (days)	Total Slack (days)
Project	32		12/27/2024	1/28/2025	12/31/2024	1/28/2025	0	0
Task 1	5		1/1/2025	1/6/2025	1/1/2025	1/6/2025	0	4
Task 2	15	1FS-10	12/27/2024	1/11/2025	12/31/2024	1/15/2025	0	0
Task 3	4	1SS+2, 2SF+3	1/3/2025	1/7/2025	1/3/2025	1/7/2025	0	0

Task 4	10	3FS+2,2FF+4	1/9/2025	1/19/2025	1/9/2025	1/19/2025	6	6
Task 5	3	2FF+5	1/13/2025	1/16/2025	1/19/2025	1/22/2025	7	7
Task 6	2	1FS+3	1/9/2025	1/11/2025	1/16/2025	1/18/2025	0	0
Task 7	4	6,4FF+3	1/18/2025	1/22/2025	1/18/2025	1/22/2025	3	3
Task 8	3	7,4SF+8	1/22/2025	1/25/2025	1/25/2025	1/28/2025	0	0
Task 9	6	5,8SF-5,7	1/22/2025	1/28/2025	1/22/2025	1/28/2025	0	0

Description: Table 3 presents the raw scheduling data generated from Microsoft Project, including columns for: (1) Task Name; (2) Duration (days); (3) Predecessors with dependency types (FS, SS, FF, SF) and lag/lead times (positive “+” for lag, negative “-” for lead); (4) Early Start and Early Finish (MM/DD/YYYY); (5) Late Start and Late Finish (MM/DD/YYYY); (6) Free Slack (days, equivalent to Free Float, FF); and (7) Total Slack (days, equivalent to Total Float, TF).

The proposed model processes this data using the ProjectScheduleMatrix Sub (Algorithm 2), which automates Precedence Diagramming Method (PDM) calculations, including the Forward Pass, Backward Pass, and float calculations (Section 3.2). Input data is entered into an Excel worksheet (“Sheet1”), and results are exported to “Sheet2” via the export process in Algorithm 3. Microsoft Project serves as the benchmark, determining the same task timing metrics using its internal algorithms.

This case study specifically tests the model’s capability to handle complex dependencies such as “1SS+2” (Task 3), “2FF+4” (Task 4), and “8SF-5” (Task 9), which are frequently encountered in actual construction projects. The ability to manage these relationships addresses a major limitation of many prior Excel-based tools and offers SMEs a cost-effective, user-friendly alternative to commercial software. The model also provides transparent matrix outputs (Tables A1–A7 in Appendix A), enabling non-expert users to verify and adjust schedules easily, thereby enhancing decision-making efficiency in resource-constrained environments (Yang et al., 2022).

4.2.2. Comparison of Scheduling Parameters

This section compares the scheduling parameters generated by the proposed VBA-Excel model with those from Microsoft Project, validating the model’s accuracy in handling complex dependency types (Finish-to-Start (FS), Start-to-Start (SS), Finish-to-Finish (FF), Start-to-Finish (SF)) and lag/lead times. Table 4 presents the Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF), Free Float (FF), and Total Float (TF) for each of the 9 tasks, with values from the VBA-Excel model (abbreviated as “VBA”) and Microsoft Project (abbreviated as “MS”) shown side by side. To facilitate direct comparison, dates from Microsoft Project (in MM/DD/YYYY format) are converted to a day-based format relative to January 1, 2025 (day 0), where December 27, 2024, corresponds to day -5 and January 28, 2025, to day 27.

Table 4 - Comparison of scheduling parameters between VBA-Excel model and Microsoft Project.

ID	Task Name	Duration (days)	Predecessors	ES (VBA/MS)	EF (VBA/MS)	LS (VBA/MS)	LF (VBA/MS)	FF (VBA/MS)	TF (VBA/MS)
1	Task 1	5		0/0	5/5	0/0	5/5	0/0	0/0
2	Task 2	15	1FS-10	-5/-5	10/10	-1/-1	14/14	0/0	4/4
3	Task 3	4	1SS+2, 2SF+3	2/2	6/6	2/2	6/6	0/0	0/0
4	Task 4	10	3FS+2, 2FF+4	8/8	18/18	8/8	18/18	0/0	0/0
5	Task 5	3	2FF+5	12/12	15/15	18/18	21/21	6/6	6/6
6	Task 6	2	1FS+3	8/8	10/10	15/15	17/17	7/7	7/7
7	Task 7	4	6, 4FF+3	17/17	21/21	17/17	21/21	0/0	0/0
8	Task 8	3	7, 4SF+8	21/21	24/24	24/24	27/27	3/3	3/3
9	Task 9	6	5, 8SF-5, 7	21/21	27/27	21/21	27/27	0/0	0/0

As shown in Table 4, the proposed VBA-Excel model achieved complete consistency with Microsoft Project across all task timing metrics. The results demonstrate 100% agreement in Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF), Free Float (FF), and Total Float (TF).

For example, Task 2’s Early Start (ES = -5, corresponding to December 27, 2024) reflects the model’s ability to accurately handle a 10-day lead time, allowing the task to start five days before the official project commencement date of January 1, 2025 (day 0). Similarly, Task 4 (ES = 8, EF = 18) and Task 9 (ES = 21, EF = 27) align perfectly with Microsoft Project’s outputs. These examples validate the correctness of the Forward Pass (Equations 2–7 in Section 3.2.2) and Backward Pass (Equations 10–15 in Section 3.2.3) calculations executed in Algorithm 2.

Both tools identify the same critical path: Task 1 → Task 3 → Task 4 → Task 7 → Task 9, with a total duration of 27 days (from day 0 to day 27), or 32 days including the pre-commencement period (from day -5 to day 27, corresponding to December 27, 2024, to January 28, 2025). Tasks on this path have zero total float (TF = 0), consistent with the PDM definition of critical tasks.

The detailed scheduling matrices generated by the ProjectScheduleMatrix Sub — including the Dependency Relationship Matrix (Table A1), Early Start Matrix (Table A2), Early Finish Matrix (Table A3), Late Start Matrix (Table A4), Late Finish Matrix (Table A5), Free Float Matrix (Table A6), and Total Float Matrix (Table A7) in Appendix A — provide a comprehensive and transparent view of all task relationships and timing metrics. This transparency enables project managers, particularly in small to medium-sized enterprises (SMEs), to verify calculations and make adjustments efficiently.

The model’s capability to handle intricate dependency types (e.g., “1SS+2” for Task 3, “2FF+4” for Task 4, and “8SF-5” for Task 9) together with lag/lead times directly addresses a major limitation of many prior Excel-based tools, which typically support only Finish-to-Start relationships. By achieving the same level of precision as Microsoft Project, the proposed VBA-Excel model demonstrates strong potential as a cost-effective, user-friendly alternative for managing sophisticated activity networks without requiring expensive licenses or advanced technical expertise.

4.2.3. Validation of Algorithm Accuracy

The results in Table 4 reveal complete consistency between the proposed model and Microsoft Project across all task timing metrics (ES, EF, LS, LF, FF, TF), validating the accuracy of the proposed model’s implementation of the Precedence Diagramming Method (PDM). This matching of early/late dates and float values confirms the correctness of the Forward Pass (Equations 2–7 in Section 3.2.2) and Backward Pass (Equations 10–15 in Section 3.2.3) as executed in Algorithm 2. For example, Task 2’s Early Start (ES = -5) in both tools corresponds to December 27, 2024 — five days before the official project start date of January 1, 2025 (day 0) — demonstrating the model’s accurate handling of lead times. Similarly, Task 4 (ES = 8, EF = 18) and Task 9 (ES = 21, EF = 27) align perfectly with Microsoft Project outputs. Both the proposed model and the commercial tool identify the same critical path: Task 1 → Task 3 → Task 4 → Task 7 → Task 9, with an overall duration of 32 days (from day -5 to day 27). Tasks on this path have zero total float (TF = 0). This high level of consistency underscores the reliability of the model in managing diverse dependency types with lag and lead times, even in challenging scenario.

Table 5 - Computation Time Comparison Across Different Project Sizes

Number of Tasks	MS Project (CPM) - Average	MS Project (CPM) - Std Dev	Adjusted Std Dev" (Std Dev ×15)	VBA Sub - Average	VBA Sub - Std Dev	Adjusted Std Dev" (Std Dev × 15)	VBA Function - Average	VBA Function - Std Dev	Adjusted Std Dev" (Std Dev × 15)
100	0.0340	0.0002	0.0036	0.3631	0.0017	0.0257	0.7185	0.0126	0.1897
500	0.1453	0.0018	0.0277	2.8457	0.0300	0.4505	5.5514	0.0251	0.3778
1000	0.2791	0.0012	0.0190	7.9931	0.0104	0.1570	15.9169	0.0523	0.7847
2000	0.5531	0.0067	0.1018	26.4210	0.1996	2.9949	52.8790	0.1993	2.9906
3000	0.8189	0.0024	0.0361	68.3563	0.7351	11.0276	136.8939	0.8348	12.5225
4000	1.0995	0.0063	0.0957	147.6794	1.3333	20.0009	246.1785	2.7093	40.6395
5000	1.3916	0.0273	0.4109	264.4575	1.2750	19.1262	414.0009	3.6739	55.1088

7000	1.9520	0.0060	0.0909	517.0639	1.6375	24.5638	672.1069	5.7173	85.7602
10000	2.8299	0.0130	0.1950	899.9395	2.4756	37.1354	931.5435	5.9961	89.9423

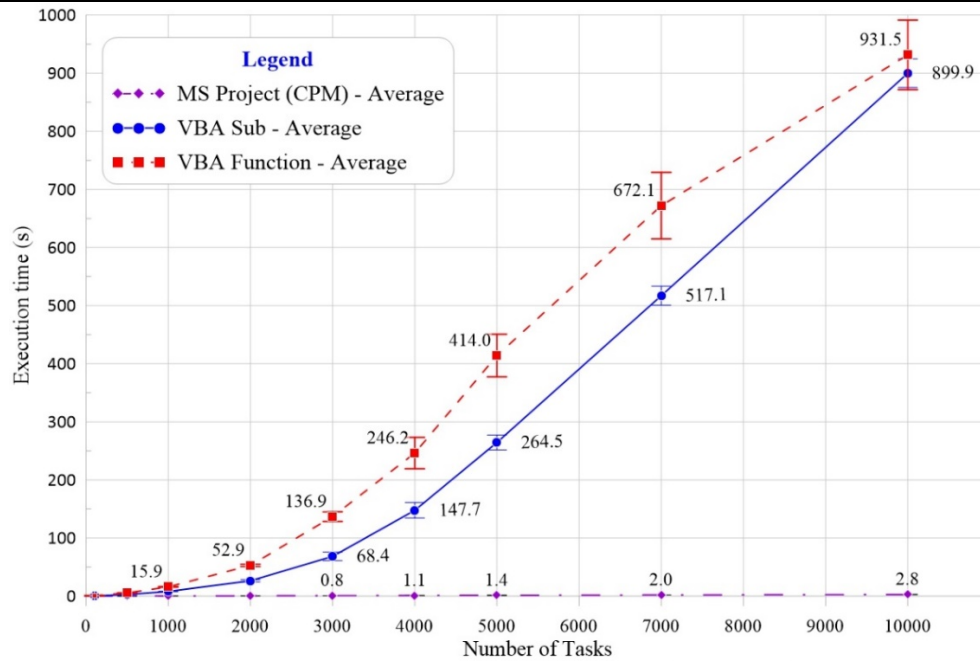


Fig.2. Computation Time Trends for Microsoft Project, VBA Sub, and VBA Function Across Different Project Sizes. The graph plots the average computation times from Table 5, showing Microsoft Project’s superior performance and the scalability challenges of the VBA implementations for large projects.

The commercial tool consistently outperforms both VBA implementations, with execution times increasing more gradually (e.g., 2.830 seconds for 10,000 tasks versus approximately 900 seconds for the VBA Sub). Despite this slower performance, the VBA model remains feasible for typical SME projects, which usually involve fewer than 5,000 tasks. However, Excel’s computational constraints and the assumption of a continuous working calendar (without holidays or non-working days) represent practical limitations that should be addressed in future enhancements.

5. Conclusion

This study introduced a VBA-Excel scheduling model designed to provide an affordable and accessible solution for construction project management, particularly for SMEs, with its accuracy validated through comparisons with *Schedula Anima* (Table 2) and Microsoft Project (Table 4), demonstrating its reliability across diverse scenarios involving intricate dependency types and lag/lead times (Bettemir & Bulak, 2022). The results from Section 4, supported by detailed outputs in Tables 1–5 and matrices in Appendix A (Tables A1–A7), highlight the tool’s efficiency and applicability, while its ability to manage complex dependencies fills a critical gap in existing Excel-based solutions, empowering non-expert project managers (Corral et al., 2023). This study developed a matrix-based VBA-Excel scheduling model that fully supports all four Precedence Diagramming Method (PDM) dependency types (FS, SS, FF, SF) with lag and lead adjustments. Through two case studies, the model demonstrated complete consistency with *Schedula Anima* for simple FS relationships and achieved 100% accuracy when benchmarked against Microsoft Project across complex dependencies and lag/lead scenarios. The tool generates transparent matrix outputs, enabling easy verification and adjustment of schedules. These features make it particularly suitable for small and medium-sized enterprises (SMEs) seeking an accessible scheduling solution.

The proposed model offers significant advantages over manual methods and commercial tools, leveraging the platform’s ease of access to deliver a cost-free, user-friendly alternative, with Algorithms 1–3 ensuring effective dependency management and streamlined result export, significantly enhancing usability for SME users. The tool’s scalability supports multi-project

environments, addressing the needs of SMEs managing concurrent construction tasks (Kannimuthu et al., 2020). Furthermore, its potential for integration with advanced optimization techniques, such as particle swarm optimization, enhances its applicability for time-cost trade-offs. However, efficiency trends in Table 5 and Figure 3 reveal scalability challenges for activity networks exceeding 5,000 tasks. The main contribution of this research lies in the development of an improved relationship matrix combined with regular expression parsing for flexible dependency handling and diagonal storage optimization for computational efficiency. Theoretically, the study advances the literature on low-cost digital tools in construction project management by providing a practical framework that bridges the gap between commercial software capabilities and the accessibility needs of resource-constrained users. Practically, the proposed model offers SMEs a cost-free alternative that can reduce scheduling time and errors while maintaining high accuracy, thereby supporting more efficient project planning and resource allocation without the burden of expensive licensing fees. Despite its strengths, the model has several limitations. Its computational performance degrades for very large projects due to Excel’s inherent constraints, and the current version does not incorporate real-world calendar features (such as holidays and non-working days), risk assessment, or resource leveling. Future research could focus on optimizing computational efficiency, incorporating calendar functionalities to account for non-working days, and integrating risk assessment methods to further enhance the tool’s practical utility (Lotfi et al., 2022). Future research should focus on enhancing computational efficiency, integrating cloud-based scalability (Hansen et al., 2023), incorporating AI-driven knowledge management (Tomczak & Jaśkowski, 2024), supply chain risk assessment (Tembo & Abdullahi, 2023), resource leveling, BIM integration (Babu et al., 2026), lean construction principles (Singh, 2025), and alignment with Construction 4.0 technologies to further broaden the tool’s applicability in complex construction environments. This study developed a VBA-Excel scheduling model to improve construction project management for SMEs, with the proposed tool—implemented through Algorithms 1–3—accommodating diverse dependency types (FS, SS, FF, SF) with lag/lead times, matching the accuracy of industry-standard tools like Microsoft Project (Table 4), and scaling effectively up to 5,000 tasks (Table 5, Figure 3), thus providing a viable alternative to costly commercial software (Yu et al., 2023). The tool offers detailed outputs (Tables A1–A7) for enhanced transparency and decision-making support, making it a feasible solution for small to medium-sized enterprises with considerable potential for future enhancements, such as integrating risk assessment and optimizing computational efficiency (Sibiya et al., 2023). Additionally, incorporating earned value management could improve project tracking and cost control, further aligning the tool with industry needs.

Acknowledgement

Appendix A

This appendix presents the detailed scheduling matrices generated by the ProjectScheduleMatrix Sub (as described in Algorithm 2 and exported via Algorithm 3) for the sample construction project in Case Study 2 (Section 4.2), a project with 9 tasks involving diverse dependency types, including the Dependency Relationship Matrix (Table A1), Early Start (ES) Matrix (Table A2), Early Finish (EF) Matrix (Table A3), Late Start (LS) Matrix (Table A4), Late Finish (LF) Matrix (Table A5), Free Float (FF) Matrix (Table A6), and Total Float (TF) Matrix (Table A7). These tables offer a holistic view of task timing metrics and relationships, ensuring transparency and reproducibility, which enables project managers, particularly in SMEs, to verify and analyze scheduling outcomes effectively.

Table A1 - Dependency Relationship Matrix for the Sample Construction Project in Case Study 2

	1	2	3	4	5	6	7	8	9
1	(1; 5; 0; 0)	(2; 15; FS; -10)	(3; 4; SS; +2)			(6; 2; FS; +3)			
2		(2; 15; -; 0)	(3; 4; SF; +3)	(4; 10; FF; +4)	(5; 3; FF; +5)				
3			(3; 4; -; 0)	(4; 10; FS; +2)					

4				(4; 10; -; 0)			(7; 4; FF; +3)	(8; 3; SF; +8)	
5					(5; 3; -; 0)				(9; 6; FS; 0)
6						(6; 2; -; 0)	(7; 4; FS; 0)		
7							(7; 4; -; 0)	(8; 3; FS; 0)	(9; 6; FS; 0)
8								(8; 3; -; 0)	(9; 6; SF; -5)
9									(9; 6; -; 0)

Table A2 - Early Start (ES) Matrix for the Sample Construction Project in Case Study 2

	1	2	3	4	5	6	7	8	9
1	0	-5	2			8			
2		-5	-6	4	12				
3			2	8					
4				8			17	13	
5					12				15
6						8	10		
7							17	21	21
8								21	10
9									21

Table A3 - Early Finish (EF) Matrix for the Sample Construction Project in Case Study 2

	1	2	3	4	5	6	7	8	9
1	5	10	6			10			
2		10	-2	14	15				
3			6	18					
4				18			21	16	
5					15				21
6						10	14		
7							21	24	27
8								24	16
9									27

Table A4 - Late Start (LS) Matrix for the Sample Construction Project in Case Study 2

	1	2	3	4	5	6	7	8	9
1	0	4	0			7			
2		-1	3	-1	1				
3			2	2					
4				8			8	19	
5					18				18
6						15	15		
7							17	20	17
8								24	32
9									21

Table A5 - Late Finish (LF) Matrix for the Sample Construction Project in Case Study 2

	1	2	3	4	5	6	7	8	9
1	5	9	5			12			
2		14	18	14	16				
3			6	6					
4				18			18	29	
5					21				21
6						17	17		
7							21	24	21
8								27	35
9									27

Table A6 - Free Float (FF) Matrix for the Sample Construction Project in Case Study 2

	1	2	3	4	5	6	7	8	9
1	0								
2		4							
3			0						
4				0					
5					6				
6						7			
7							0		
8								3	
9									0

Table A7 - Total Float (TF) Matrix for the Sample Construction Project in Case Study 2

	1	2	3	4	5	6	7	8	9
1	0								
2		0							
3			0						
4				0					
5					6				
6						7			
7							0		
8								3	
9									0

References

Abuhasel, K. (2023). Sustainable Green City Development Project Analysis using the Critical Path Method (CPM) and the Crashing Project Method on Time and Cost Optimization. *Engineering, Technology and Applied Science Research*, 13(3). <https://doi.org/10.48084/etasr.5980>

Al-Dhamad, S. H. R., Kurwi, S. I. B., Varouqa, I. F., Hayder, G., Obaid, A. H., & Al-Zwainy, F. M. S. (2025). Optimizing Project Management: Planning, Scheduling, and Cost Estimation Using Microsoft Project. *Journal of Applied Science, Engineering, Technology, and Education*, 7(2). <https://doi.org/10.35877/454RI.asci4165>

Alfandi Rizki Pratama, R., Djakfar, L., & Negara, K. P. (2025). Monte Carlo Mathematical Model Simulation: An Evaluation of The Probability of Construction Project Acceleration.

- Journal of Multidisciplinary Applied Natural Science*, 5(2).
<https://doi.org/10.47352/JMANS.2774-3047.269>
- Amer, F., Koh, H. Y., & Golparvar-Fard, M. (2021). Automated Methods and Systems for Construction Planning and Scheduling: Critical Review of Three Decades of Research. *Journal of Construction Engineering and Management*, 147(7).
[https://doi.org/10.1061/\(asce\)co.1943-7862.0002093](https://doi.org/10.1061/(asce)co.1943-7862.0002093)
- Arditi, D., & Bentotage, S. N. (1996). System for scheduling highway construction projects. *Microcomputers in Civil Engineering*, 11(2).
<https://doi.org/https://doi.org/10.1111/j.1467-8667.1996.tb00316.x>
- Aslan, S., & Türkakın, O. H. (2022). A construction project scheduling methodology considering COVID-19 pandemic measures. *Journal of Safety Research*, 80.
<https://doi.org/10.1016/j.jsr.2021.11.007>
- Babu, V. K. V., Saini, P., & Charu, K. M. (2026). Integrating Building Information Modelling (BIM) in Construction Projects: A Study to Increase the Efficiency of Construction Projects. *Lecture Notes in Networks and Systems*, 1511 LNNS.
https://doi.org/10.1007/978-981-96-8796-1_3
- Bettemir, Ö. H., & Bulak, Ö. F. (2022). Scheduling, Management and Optimization of Construction Process. *Teknik Dergi/Technical Journal of Turkish Chamber of Civil Engineers*, 33(6). <https://doi.org/10.18400/tekderg.981601>
- Bidhendi, A., Nahri, Z., MotamedManesh, M., & Poshdar, M. (2025). Leveraging BIM and Prefabrication for Enhanced Architectural Excellence in Sustainable Construction. *Lecture Notes in Civil Engineering*, 591 LNCE. https://doi.org/10.1007/978-981-96-4051-5_106
- Chen, S., Eysers, D. R., Gosling, J., & Huang, Y. (2025). Supply chain risks for SMEs in construction projects: a structured literature review and research agenda. In *International Journal of Logistics Management* (Vol. 36, Number 3). <https://doi.org/10.1108/IJLM-12-2023-0548>
- Corral, F., Forcael, E., & Linfati, R. (2023). Workforce scheduling efficiency assessment in construction projects through a multi-objective optimization model in the COVID-19 context. *Heliyon*, 9(6). <https://doi.org/10.1016/j.heliyon.2023.e16745>
- Datta, S. D., Sobuz, M. H. R., Mim, N. J., & Nath, A. D. (2023). Investigation on the effectiveness of using building information modeling (BIM) tools in project management: a case study. *Revista de La Construcción*, 22(2). <https://doi.org/10.7764/RDLC.22.2.306>
- Doukari, O., Seck, B., & Greenwood, D. (2022). The Creation of Construction Schedules in 4D BIM: A Comparison of Conventional and Automated Approaches. *Buildings*, 12(8).
<https://doi.org/10.3390/buildings12081145>
- Elazouni, A., & Abido, M. A. (2014). Enhanced Trade-Off of Construction Projects: Finance-Resource-Profit. *Journal of Construction Engineering and Management*, 140(9).
[https://doi.org/10.1061/\(asce\)co.1943-7862.0000880](https://doi.org/10.1061/(asce)co.1943-7862.0000880)
- Elkliny, A., Sanad, H., & Etman, E. (2025). GA-driven MCDM model for time-cost-quality-resource trade-off in construction projects. *Engineering, Construction and Architectural Management*. <https://doi.org/10.1108/ECAM-06-2025-1046>
- Fares, A., Elazouni, A., Al-Shihabi, S., & Al-Alawi, M. (2025). Finance-based scheduling for projects in owners' portfolios: MILP versus improved GA and PSO. *Engineering, Construction and Architectural Management*. <https://doi.org/10.1108/ECAM-07-2024-0960>
- Goh, C. S., & Lee, S. S. H. (2023). an Investigation of the Implications of Covid-19 for Digitalisation in the Malaysian Construction Industry. *Journal of Engineering Science and Technology*, 18(5). <https://doi.org/10.25046/aj0505119>
- Hansen, S., Fassa, F., & Wijaya, S. (2023). Factors Influencing Scheduling Activities of Construction Projects. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, 15(1). [https://doi.org/10.1061/\(asce\)la.1943-4170.0000594](https://doi.org/10.1061/(asce)la.1943-4170.0000594)
- Hu, C., Wang, J., & Mei, Y. (2021). Uncertain Time-Resource-Cost Trade-Off Models for Construction Project Schedule. *KSCE Journal of Civil Engineering*, 25(8).
<https://doi.org/10.1007/s12205-021-1798-7>

- Isah, M. A., & Kim, B. S. (2021). Integrating schedule risk analysis with multi-skilled resource scheduling to improve resource-constrained project scheduling problems. *Applied Sciences (Switzerland)*, 11(2). <https://doi.org/10.3390/app11020650>
- Kannimuthu, M., Raphael, B., Ekambaram, P., & Kuppuswamy, A. (2020). Comparing optimization modeling approaches for the multi-mode resource-constrained multi-project scheduling problem. *Engineering, Construction and Architectural Management*, 27(4). <https://doi.org/10.1108/ECAM-03-2019-0156>
- Khan, A. N., Mehmood, K., & Soomro, M. A. (2024). Knowledge Management-Based Artificial Intelligence (AI) Adoption in Construction SMEs: The Moderating Role of Knowledge Integration. *IEEE Transactions on Engineering Management*, 71. <https://doi.org/10.1109/TEM.2024.3403981>
- Le, L. Q., Teo, P., & Gajanayake, A. (2025). Explicit CSR Communication Among Trade and Subcontractor SMEs in the Construction Industry. *Corporate Social Responsibility and Environmental Management*, 32(3). <https://doi.org/10.1002/csr.3160>
- Lotfi, R., Yadegari, Z., Hosseini, S. H., Khameneh, A. H., Tirkolaee, E. B., & Weber, G. W. (2022). A ROBUST TIME-COST-QUALITY-ENERGY-ENVIRONMENT TRADE-OFF WITH RESOURCE-CONSTRAINED IN PROJECT MANAGEMENT: A CASE STUDY FOR A BRIDGE CONSTRUCTION PROJECT. *Journal of Industrial and Management Optimization*, 18(1). <https://doi.org/10.3934/jimo.2020158>
- Mafundu, R. H., & Mafini, C. (2019). Internal constraints to business performance in black-owned small to medium enterprises in the construction industry. *Southern African Journal of Entrepreneurship and Small Business Management*, 11(1). <https://doi.org/10.4102/SAJESBM.V11I1.165>
- Maravas, A., & Pantouvakis, J.-P. (2025). Schedules Anima: Dynamic Visualization of Gantt Charts and Resource Usage Graphs in Project Scheduling. *Buildings*, 15(3), 393. <https://doi.org/10.3390/buildings15030393>
- Mohamed Meabed, E. S., Mahfouz, S. Y., & Alhady, A. (2025). Modified critical chain scheduling for construction projects. *HBRC Journal*, 21(1). <https://doi.org/10.1080/16874048.2025.2459038>
- Nicholas, M., Monko, R., Malekela, K., & Zulu, S. (2024). Assessment of building information modeling adoption in building material scheduling in Tanzania's construction industry. *Sustainable Engineering and Innovation*, 6(2). <https://doi.org/10.37868/sei.v6i2.id307>
- Pranata, C., Setyandito, O., Kosalim, V., Purnamasari, C., & Syahputra, M. Y. (2023). Analysis of Cost Planning and Time Control of Breakwater Embankment Ground Improvement Method at Port in Subang Regency West Java. *AIP Conference Proceedings*, 2846(1). <https://doi.org/10.1063/5.0155433>
- Pregina, K., & Ramesh Kannan, M. (2022). Stochastic Project Network Scheduling Technique for Construction Projects Using GERT. *Lecture Notes in Civil Engineering*, 191. https://doi.org/10.1007/978-981-16-5839-6_33
- Purushothaman, M. B., Elenzano, D., GhaffarianHoseini, A., & Ghaffarianhoseini, A. (2025). Dynamic scheduling: influential and influenced factors and their interrelationships in the New Zealand construction industry. *Smart and Sustainable Built Environment*. <https://doi.org/10.1108/SASBE-11-2024-0488>
- Ramahwati, A., & Osly, P. J. (2024). Estimasi Biaya Dan Penjadwalan Pembangunan Prototipe Ruang Modular Operating Theatre (Cost Estimation And Scheduling Modular Operating Theatre Prototype Construction). *Jurnal Artesis*, 4(2).
- Rosener, W. J. (2024). Comparative Analysis of Google Sheets and Microsoft Excel. *Journal of Cases on Information Technology*, 26(1). <https://doi.org/10.4018/jcit.351672>
- S, S., & George, G. (2021). A COMPARISON OF PROJECT PLANNING AND SCHEDULING SOFTWARE'S PRIMAVERA AND MICROSOFT PROJECT USED IN THE CONSTRUCTION INDUSTRY. *International Journal of Engineering Technologies and Management Research*, 8(9). <https://doi.org/10.29121/ijetmr.v8.i9.2021.1039>
- Sapitri, Aziz Azizi, F. A., Yulianti, A., Retno, D. P., Elizar, & Susanti, B. (2025). Application of BIM Technology in Construction Industry for Modeling, Quantity Takeoff, and

- Scheduling Simulation. *2025 7th International Conference on Smart Applications, Communications and Networking, SmartNets 2025*.
<https://doi.org/10.1109/SmartNets65254.2025.11106790>
- Schkarin, T., & Dobhan, A. (2022). Prerequisites for Applying Artificial Intelligence for Scheduling in Small- and Medium-sized Enterprises. *International Conference on Enterprise Information Systems, ICEIS - Proceedings*, 1.
<https://doi.org/10.5220/0011064000003179>
- Shah, P., & Chandragade, A. A. (2023). Application of project management tool in construction for Planning, Scheduling and Optimization. *Materials Today: Proceedings*, 77.
<https://doi.org/10.1016/j.matpr.2022.11.446>
- Shalaby, A., & Ezeldin, A. S. (2022). A model for work packages optimization in results-based-finance projects. *Engineering, Construction and Architectural Management*, 29(7).
<https://doi.org/10.1108/ECAM-10-2019-0556>
- Sibiya, A., Sibiya, B., Jordaan, J., & Mahosi, B. N. (2023). Enhancing Project Management for SMEs: A Hybrid Approach. *Journal of Management and Entrepreneurship Research*, 4(2). <https://doi.org/10.34001/jmer.2023.12.04.2-42>
- Singh, J. (2025). Lean Construction Principles and Practices. In *Handbook of Construction Project Management*. https://doi.org/10.1007/978-981-96-7631-6_30
- Siriwardhana, S. D., Moehler, R. C., & Fang, Y. (2025). Construction 4.0 in Australia: Evaluating technological implementation, organisational adaptation and skills development. *Journal of Innovation and Knowledge*, 10(6). <https://doi.org/10.1016/j.jik.2025.100822>
- Srinath, P., & Varghese, K. (2023). Identifying and Analysing the Root Causes of Quality Non-Conformance in Construction Project Baseline Schedules. *Journal of The Institution of Engineers (India): Series A*, 104(2). <https://doi.org/10.1007/s40030-023-00713-0>
- Tembo, M., & Abdullahi, C. M. (2023). Improving the Efficiency and Effectiveness of Construction Project Planning and Scheduling Using Lean Principles. *International Journal of Construction Engineering and Management*, 2023(3).
<https://doi.org/10.5923/j.ijcem.20231203.01>
- Tomczak, M., & Jaśkowski, P. (2024). Customized particle swarm optimization for harmonizing multi-section construction projects. *Automation in Construction*, 162.
<https://doi.org/10.1016/j.autcon.2024.105359>
- Wan, Z., & He, J. (2004). Construction project scheduling problem with uncertain resource constraints. *Chinese Journal of Engineering*, (70371032). <https://doi.org/10.1007/s12205-023-2164-8>
- Wang, J., Zeng, S., & Ju, J. (2017). Construction Schedule Management Using Resource-Constrained Project Scheduling Model. *Tongji Daxue Xuebao/Journal of Tongji University*, 45(10). <https://doi.org/10.11908/j.issn.0253-374x.2017.10.020>
- Wang, K., Xu, X., Mao, P., Deng, X., & Cao, D. (2025). Digital transformation and organizational readiness: evidence from Chinese construction SMEs with a dynamic managerial capabilities lens. *Engineering, Construction and Architectural Management*.
<https://doi.org/10.1108/ECAM-01-2025-0069>
- Xie, L., Chen, Y., & Chang, R. (2021). Scheduling optimization of prefabricated construction projects by genetic algorithm. *Applied Sciences (Switzerland)*, 11(12).
<https://doi.org/10.3390/app11125531>
- Xu, Z. (2025). Integrating BIM Technology in High-Rise Building Construction: Applications in Deep Foundation Optimization, Slipform Process Control, and Safety Management. *Applied and Computational Engineering*, 167(1). <https://doi.org/10.54254/2755-2721/2025.g125441>
- Yang, Y., Pan, M., & Pan, W. (2022). Integrated Offsite Logistics Scheduling Approach for High-Rise Modular Building Projects. *Journal of Construction Engineering and Management*, 148(6). [https://doi.org/10.1061/\(asce\)co.1943-7862.0002280](https://doi.org/10.1061/(asce)co.1943-7862.0002280)
- Yu, Z., Wang, C., Zhao, Y., Hu, Z., & Tang, Y. (2023). Linear Project-Scheduling Optimization Considering a Reverse Construction Scenario. *Applied Sciences (Switzerland)*, 13(16).
<https://doi.org/10.3390/app13169407>

- Zhang, Z., & Xu, J. (2016). Bi-level multiple mode resource-constrained project scheduling problems under hybrid uncertainty. *Journal of Industrial and Management Optimization*, 12(2). <https://doi.org/10.3934/jimo.2016.12.565>
- Zhao, J., Pikas, E., Seppänen, O., & Peltokorpi, A. (2021). Using Real-Time Indoor Resource Positioning to Track the Progress of Tasks in Construction Sites. *Frontiers in Built Environment*, 7. <https://doi.org/10.3389/fbuil.2021.661166>