

# An Object-Oriented Algorithm for Flexible Construction Scheduling with Automated Task Splitting

José Antonio González-Fajardo\*, Jesús Nicolás Zaragoza-Grifé, Sergio Omar Álvarez-Romero

Facultad de Ingeniería, Universidad Autónoma de Yucatán, Mérida, Yucatán, México

Email: \*antonio.gonzalez@correo.uady.mx

**How to cite this paper:** González-Fajardo, J.A., Zaragoza-Grifé, J.N. and Álvarez-Romero, S.O. (2026) An Object-Oriented Algorithm for Flexible Construction Scheduling with Automated Task Splitting. *Journal of Building Construction and Planning Research*, 14, 21-40.

<https://doi.org/10.4236/jbcpr.2026.141002>

**Received:** December 17, 2025

**Accepted:** December 28, 2025

**Published:** December 31, 2025

Copyright © 2026 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

The Precedence Diagram Method (PDM) is widely used for modeling large and complex construction projects characterized by non-linear execution logic, overlapping activities, and multiple precedence constraints. Despite involving standardized and recurrent operations (e.g., formwork, reinforcement, concreting), construction schedules often exhibit highly complex logic that challenges traditional Critical Path Method (CPM) analysis. In particular, PDM networks with multiple relationships complicate critical path identification and float determination, while current commercial scheduling tools impose restrictive logic and limited support for activity splitting. This study proposes a novel object-oriented algorithm and a supporting software tool that integrates automated multi-activity splitting, unrestricted handling of precedence types, and segment-level criticality identification within a unified scheduling framework. The proposed approach transforms complex PDM networks into equivalent Finish-to-Start-based structures solvable using conventional forward and backward pass algorithms, while strictly preserving the original scheduling logic. Results demonstrate that the proposed method improves analytical transparency, scheduling flexibility, and precision in critical path and float analysis, including the identification of partially critical activity segments. The proposed framework provides project managers with a practical and scalable solution to optimize productivity and reduce project duration in complex construction scheduling scenarios.

## Keywords

Construction Scheduling, Project Scheduling, Precedence Diagram Method, PDM with Multiple Relationships, Activity Splitting, Partial Criticality, Object-Oriented Algorithm, Construction Management Software

## 1. Introduction

Construction scheduling remains a critical component of project management, particularly for large and complex construction projects characterized by non-linear execution. Traditional methods such as the Critical Path Method (CPM) have dominated practice since the late 1950s, offering a systematic approach to determine project duration and critical activities [1] [2]. However, CPM's reliance on Finish-to-Start (FS) relationships limits its ability to model overlapping tasks and complex logic without cumbersome workarounds such as dummy activities or excessive task segmentation [3].

The evolution toward Activity-on-Node (AON) networks and the Precedence Diagram Method (PDM) introduced greater modeling flexibility by incorporating multiple relationship types—Finish-to-Start (FS), Start-to-Start (SS), Finish-to-Finish (FF), and Start-to-Finish (SF)—allowing partial overlaps and a more realistic representation of construction processes [4] [5]. It was first developed by Fondahl [6]. Nevertheless, recent studies have shown that the practical use of PDM in commercial scheduling software remains constrained by restrictive logic rules, limited support for activity splitting, and simplified criticality assumptions, which often prevent accurate representation of complex precedence structures [7]. This paper focuses on PDM with multiple relationships (PDM-MR), which optimizes productivity and scheduling flexibility.

Despite its advantages, PDM-MR introduces analytical challenges that hinder its practical application, particularly in identifying critical paths, calculating floats, and managing activity splitting. Recent extensions of the Precedence Diagram Method have attempted to address complex precedence relationships and analytical limitations [8].

### 1.1. PDM Drawbacks

Although PDM-MR enhances modeling capabilities, its complexity creates significant drawbacks. The inclusion of multiple relationship types complicates float determination and critical path analysis, often leading to ambiguous or inconsistent results [9] [10]. Existing algorithms struggle to detect partially critical activities, which occur when only segments of an activity influence project duration [11]. Furthermore, heuristic approaches for activity splitting, such as those proposed by Crandall in [12], have been shown to fail under certain conditions, resulting in solutions that fail to meet scheduler-imposed constraints [13].

As a result, PDM networks with multiple relationships (PDM-MR), while theoretically powerful, introduce significant analytical challenges in critical path identification and float determination. These challenges have been widely documented in literature, particularly when overlapping relationships and partial dependencies are present [1] [10].

### 1.2. Software for Scheduling

Recent studies have shown that the practical use of PDM in commercial schedul-

ing software remains constrained by restrictive logic rules, limited support for activity splitting, and simplified criticality assumptions [6]. Commercial scheduling tools such as Primavera P6 and Microsoft Project have incorporated PDM concepts to improve flexibility [14]. However, these platforms impose critical restrictions: Primavera does not allow activity splitting, while Microsoft Project permits only manual splitting. Both tools limit the number and type of relationships between activities, preventing full exploitation of PDM-MR's potential [15].

Additionally, current software fails to identify partially critical activities, undermining its usefulness for corrective actions and forensic analysis during project control [16]. These shortcomings often force practitioners to rely on manual adjustments, which are time-consuming and prone to error.

### 1.3. Efforts to Enhance PDM-MR

Researchers have proposed various strategies to address PDM-MR limitations. Valls *et al.* [13] introduced improved algorithms for activity splitting, while Lu and Lam [9] suggested transformation schemes to convert non-FS relationships into equivalent FS logic for easier analysis. Other approaches include Logic Diagramming Method [17], Relationship Diagramming Method [18], and Critical Path Segments (CPS) scheduling [19], which decomposes activities into finer time segments.

Recent studies have explored point-to-point relationships and continuous relations to model overlapping tasks more accurately [20]. Despite these advances, no widely adopted solution fully integrates automated multi-splitting, unrestricted relationship handling, and clear identification of partially critical activities.

### 1.4. Contribution of this Study

This research proposes an object-oriented algorithm and a supporting computational tool designed to overcome the limitations of current PDM-MR implementations. The solution enables: a) unrestricted use of FS, SS, FF, and SF relationships; b) automated single or multiple activity splitting to satisfy all scheduler-imposed constraints; and c) precise identification of critical paths and floats, including partially critical segments.

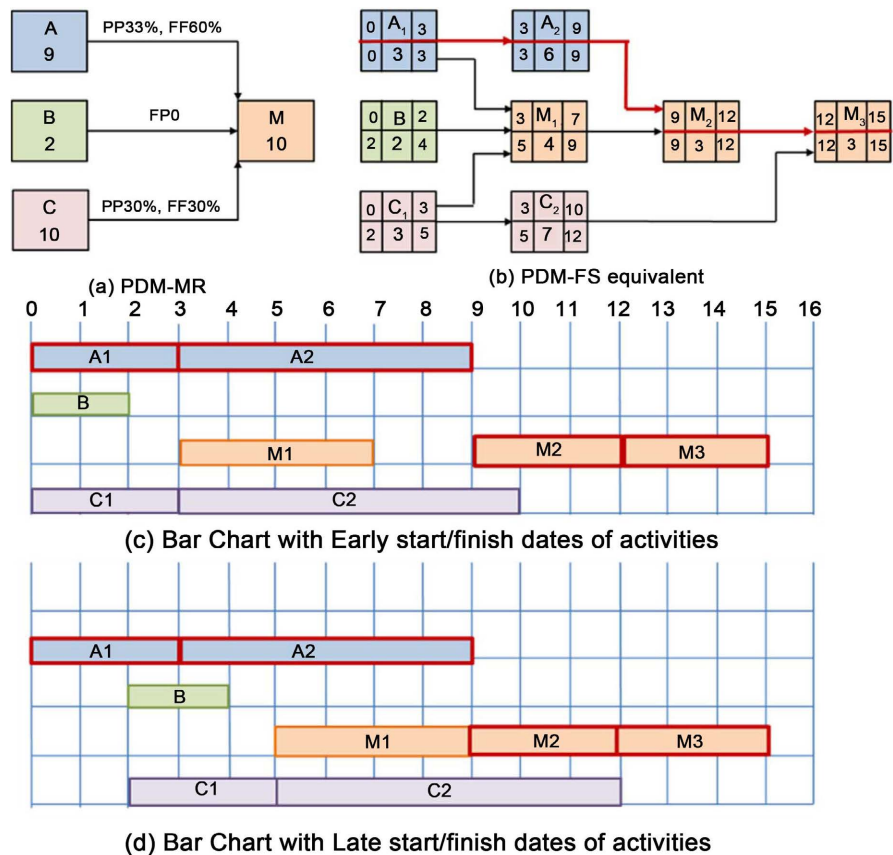
The approach transforms complex PDM-MR networks into equivalent PDM-FS structures, allowing the use of conventional algorithms for forward and backward pass computations while preserving original logic. This transformation ensures accuracy, transparency, and ease of analysis, providing schedulers with immediate feedback and improved flexibility. By integrating these capabilities into a practical software tool, the study offers a robust foundation for enhanced planning, resource allocation, and BIM-based visualization, ultimately contributing to more reliable and efficient construction scheduling. The algorithm deals with time constraints, but it does not consider resource constraints. However, the method shows clearly, as a result, the “activity segments” that are very useful for realistic scheduling of project resources. The solution provides the simplicity of program-

ming PDM-MR networks, preserving complete flexibility; at the same time, it offers precision in the calculations and ease in the analysis of the critical path and floats.

## 2. Method

### 2.1. Basic Concepts for Transformation

The proposed approach builds on transforming complex PDM-MR networks into equivalent PDM-FS networks, enabling the use of conventional algorithms for forward and backward pass computations. Transforming a PDM-MR network into an equivalent PDM-FS network requires applying transformation schemes iteratively until all precedence constraints are satisfied [8]. Each transformation may require dividing an activity into multiple sub-activities to reflect the scheduler's intended logic. For example, **Figure 1** illustrates how activities A, C, and M are split into sub-activities (A1, A2; C1, C2; M1, M2, M3) to comply with overlapping relationships, while activity B remains intact.



**Figure 1.** Comparison of original PDM-MR network (with multiple relationships) and its transformed PDM-FS equivalent used for computational analysis; includes early/late bars showing segments and interruptions.

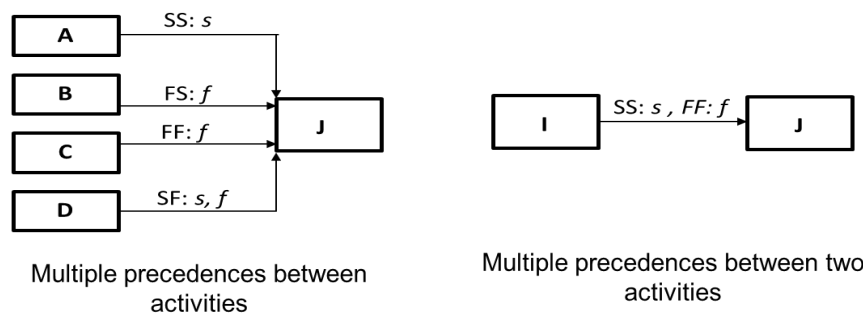
The transformation ensures that early and late start/finish dates remain consistent across both representations, provided activity splitting is allowed. This pro-

cess introduces three key concepts: a) interruptions (non-performance periods within an activity); b) sub-activities (portions of an original activity created during transformation); and c) segments (continuous portions of an activity after computation, which may consist of one or more sub-activities). **Figure 1(c)** and **Figure 1(d)** illustrate how bar charts based on early and late dates reveal interruptions and the composition of the critical path. When splitting is permitted, project duration typically decreases because critical activities can start earlier. However, if uninterrupted execution is required (e.g., concrete casting), the scheduler can override splitting during logic definition.

In this study, the concept of partially critical activity is formalized at the segment level. An activity segment is defined as partially critical when its total float is equal to zero, while at least one other segment belonging to the same original activity exhibits positive float. Under this condition, only a portion of the activity directly constrains the project duration, a situation that cannot be correctly identified when activities are treated as indivisible units.

## 2.2. Flexibility and Restrictions

To maximize flexibility, the transformation algorithm should not impose limits on the number or type of precedence relationships (FS, SS, FF, SF) between activities (**Figure 2**). This capability allows modeling of complex scenarios, such as staggered SS/FF dependencies (ZZij), which are common in construction processes like excavation and formwork operations. This relationship between two activities (ZZij) was defined by Moder *et al.* [5] and included in the interface to enter the PDM-MR information; this represents the Lu and Lam [9] SS/FF staggered case. It means that, between two activities, namely “A” and “B”, both a dependence SSAB type and an FFAB type could coexist, which can be invaluable to speed up the project and increase productivity.



**Figure 2.** Illustration of unrestricted precedence types (FS, SS, FF, SF) and multiple relationships allowed between activities, enabling complex overlapping logic.

Transformation schemes summarized in **Figure 3** define how each relationship type (SS, FS, FF, SF) is converted into FS logic for computational purposes. These schemes are applied iteratively to satisfy all constraints. Restrictions include: FS lag may be zero or positive; FF lag must be positive and less than the successor’s

duration; SS lag must be positive and less than the predecessor’s duration; and SF lag is computed as  $SF' + SF''$ , where each component is positive and less than the respective activity duration.

The algorithm automatically determines the number and size of sub-activities, calculates early/late dates, and identifies critical paths without violating scheduler-imposed logic. This ensures that all original constraints are preserved while simplifying analysis.

Staggered relationships (ZZ), defined as the coexistence of Start-to-Start (SS) and Finish-to-Finish (FF) constraints between the same pair of activities, are handled by the algorithm as independent logical restrictions. During transformation, each constraint generates its corresponding CPMRecord, and the combined effect is enforced through automated activity splitting. This approach preserves the intended overlapping logic while maintaining compatibility with conventional forward and backward pass computations.

Type of Relationship	PDM-MR network	PDM-FS network	Restriction
Finish-to-Start (FS: d)			$d \geq 0$
Finish-to-Finish (FF: f)			$0 < f < b$
Start-to-Start (SS: s)			$0 < s < a$
Start-to-Finish (SF: s, f)			$0 < s < a$ $0 < f < b$
Start-to-Start and Finish-to-Finish (ZZ: s, f)			$0 < s < a$ $0 < f < b$
	<i>a and b : duration of activities</i>	<i>s and f : lead/lag factors</i>	

**Figure 3.** Rules for converting PDM-MR relationships into equivalent FS-based logic for network computation, with constraints on lag values (FS, SS, FF, SF).

### 2.3. Transformation of Information Structures

To implement the proposed solution, the algorithm relies on an object-oriented approach using the Unified Modeling Language (UML). This strategy provides a robust framework for modeling complex information structures and their transformation from PDM-MR to PDM-FS representations.

The domain model (**Figure 4**) defines persistent classes—Project, Activity, and Precedence—which store the scheduler’s original PDM-MR network. Non-persistent classes represent intermediate structures derived during transformation and computation. The Project class orchestrates the entire process through its public method *Calculate* (), which invokes six private methods responsible for initialization, transformation, linking, and computation. The figure shows the acronyms “PDM” and “CPM” repeatedly; the information linked to the PDM acronym is related to the PDM-MR information structure, whereas the information linked to the CPM acronym is related to the PDM-FS information structure.

The transformation process consists of the following steps: 1) Initialization and Ordering: Activities are assigned an order attribute based on precedence logic using a depth-first search algorithm. 2) Creation of PDM-FS Activities: Each PDM-MR activity is decomposed into sub-activities as required to satisfy all constraints. Delays for non-zero FS relationships are modeled as separate activities. 3) Linking Activities: Sub-activities are connected using FS = 0 relationships to form a solvable PDM-FS network. 4) Computation: Conventional forward and backward pass algorithms calculate early/late dates and project duration. 5) Update and Output: Results are mapped back to the original PDM-MR schema, identifying segments, interruptions, floats, and critical paths. **Figure 5** shows the algorithm of the public method *Calculate* ().

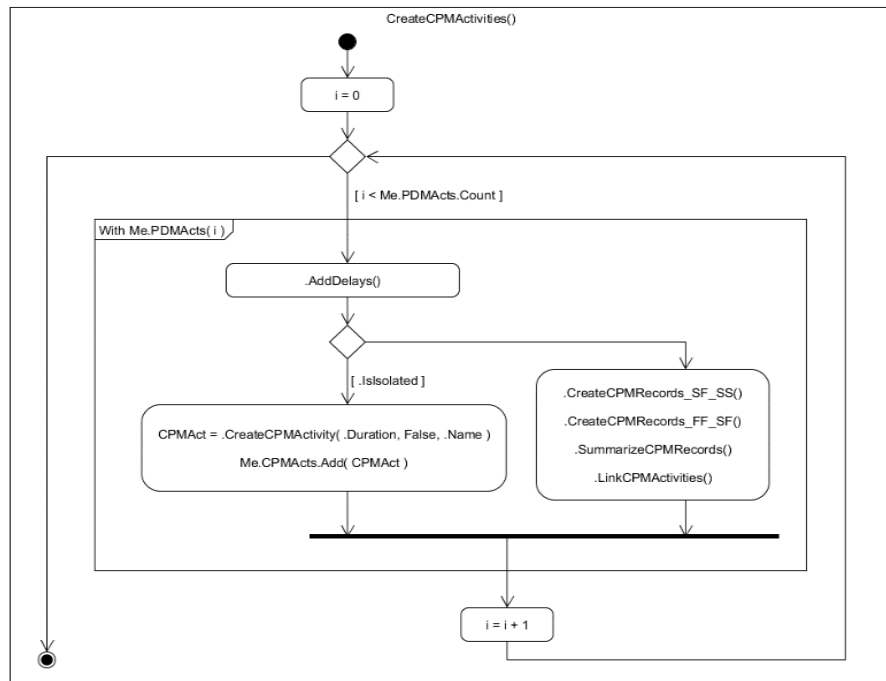
The *InitializePDMActivities* and *VisitPDMActivities* methods determine the computing sequence for the PDM-MR activities in three steps: 1) attributes order and visited are initialized for each PDM-MR activity, as well as the *CPMRecords* and *CPMActs* collections; 2) a value is assigned to the order attribute for each PDM-MR activity, according to their precedencies in a recursive fashion using a deep search algorithm; and 3) PDM-MR activities are sorted in ascending order by the order attribute. The current implementation assumes that the PDM-MR network is non-cyclic. Prior to transformation, activities are ordered using a depth-first search procedure. If cyclic logic were present, the transformation would be undefined; therefore, loop detection and user notification are identified as a necessary pre-validation step and a topic for future development.

**Figure 5** shows the algorithm of the public method *Calculate* ().

The *CreateCPMActivities* method shown in **Figure 6** is mainly responsible for obtaining all PDM-FS activities corresponding to each PDM-MR activity, meeting strictly the logic established by the scheduler. This method’s algorithm consists of a main cycle running through all PDM-MR activities of the project. Within this cycle, the *AddDelays* method is invoked first. This method is responsible for creating a required delay for each precedence of the non-zero FS type, which is equivalent to a PDM-FS activity with a duration equal to the value of the attribute lag1 from the constraint FS for the current PDM activity analyzed.

If the activity is isolated (stands alone), with no predecessors or successors, the next step within the *CreateCPMActivities* cycle creates a PDM-FS activity with a duration equal to the current PDM-MR activity. Otherwise, the *CreateCPMRecords\_SF\_SS*, *CreateCPMRecords\_FF\_SF*, *SummarizeCPMRecords* and





**Figure 6.** Creation of PDM-FS activities from PDM-MR activities, including delays for non-zero FS and derivation of sub-activities.

*LinkCPMActivities* methods are invoked for the current PDM-MR activity. The *CPMRecords* values are eventually transformed into either full or partial future PDM-FS activities to satisfy the precedence relationships.

The *CreateCPMRecords\_SF\_SS* method runs through all precedencies of the PDM-MR activity to create a *CPMRecord* value for each constraint of the SF and SS type that links the current PDM-MR activity with its successor. The *CreateCPMRecords\_FF\_SF* runs through all precedencies of the PDM-MR activity to create a *CPMRecord* value for each constraint of the FF and SF type, linking the predecessors to the current PDM-MR activity.

The *SummarizeCPMRecords* method takes all *CPMRecords* values to create all PDM-FS activities required for the current PDM-MR activity. The method sorts the *CPMRecords* collection incrementally, from the lowest to the highest duration. Then, determines the duration for each PDM-FS activity that is created. The sum of the durations of all PDM-FS activities from each PDM-MR activity is equal to the duration of the latter.

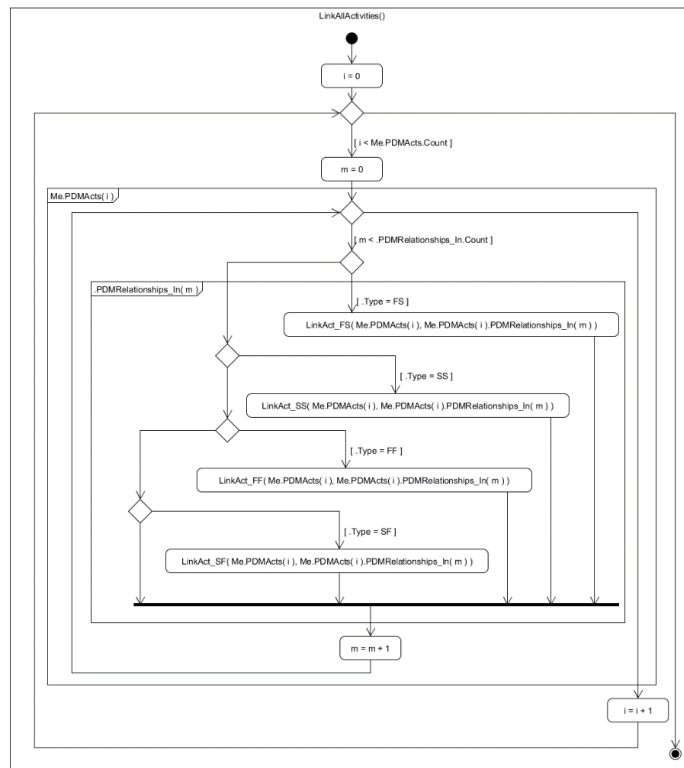
The *LinkCPMActivities* method creates relationships of the FS = 0 type between all PDM-FS activities from the current PDM-MR activity. Additionally, it calculates the accumulated duration (*AcumDur*) and the previous accumulated duration (*PreviousAcumDur*) for each PDM-FS activity.

The *LinkAllActivities* method implemented by the project class, shown in **Figure 7**, uses: 1) all PDM-FS activities previously created, 2) the *AcumDur* and *PreviousAcumDur* values, and 3) the PDM-MR constraints of the FS, SS, FF, and SF types originally defined by the scheduler, all these to build the PDM-FS network

equivalent to the initial PDM-MR network. This method consists of a main cycle that runs through all PDM-MR activities. For each iteration within the cycle, there is a nested cycle that runs through all precedencies of the current PDM-MR activity. Depending upon the type of current precedence, one or more of the *LinkAct\_FS*, *LinkAct\_SS*, *LinkAct\_FF*, and *LinkAct\_SF* methods are invoked.

The *CalculateCPMForwardBackward()* method calculates the corresponding indexes for the early/late start/finish dates for each PDM-FS activity within the project. Then, calculates the total duration of the project.

Finally, the *UpdateAllDates()* method calculates the early/late calendar dates based on the indexes generated from invoking the previous method.



**Figure 7.** Algorithm for linking all activities: construction of FS = 0 relations among sub-activities and enforcement of original FS/SS/FF/SF constraints.

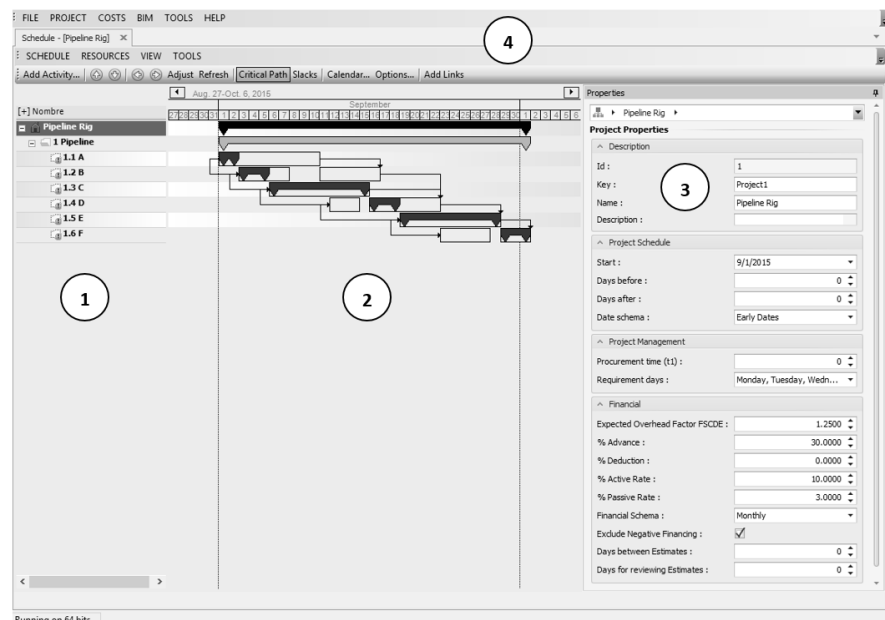
### 2.4. Computational Complexity Considerations

From a computational standpoint, the proposed algorithm relies exclusively on polynomial-time procedures. Activity ordering is performed using a depth-first search algorithm, which scales with the number of activities and precedence relationships. The transformation process, including the generation of *CPMRecords* and PDM-FS sub-activities, grows proportionally with the number of PDM-MR activities and their associated constraints. Forward and backward pass computations follow standard CPM complexity. As a result, the overall computational effort increases polynomially with network size, supporting the applicability of the proposed approach to construction projects of practical size and complexity.

### 3. Proof of Concept

To validate the proposed algorithm, a prototype software tool named *ProFin* was developed using Visual Basic within Microsoft Visual Studio 2013. The tool implements the transformation and computation process transparently, allowing users to input project data in PDM-MR format and obtain results in the form of Gantt charts and critical path analysis without manual intervention.

The graphical user interface (Figure 8) is divided into four main sections: 1) Work Breakdown Structure (WBS); 2) Gantt Chart; 3) Properties Window for selected entities; and 4) Menu and Toolbars. *ProFin* offers features for adding activities individually or in batches, defining relationships interactively via the Gantt chart or through a properties window, configuring calendars, exporting CSV files for integration with BIM tools such as Autodesk Navisworks®, and dynamically displaying dependencies, slacks, and critical paths. *ProFin* has several features to facilitate the creation and subsequent management of a project schedule.

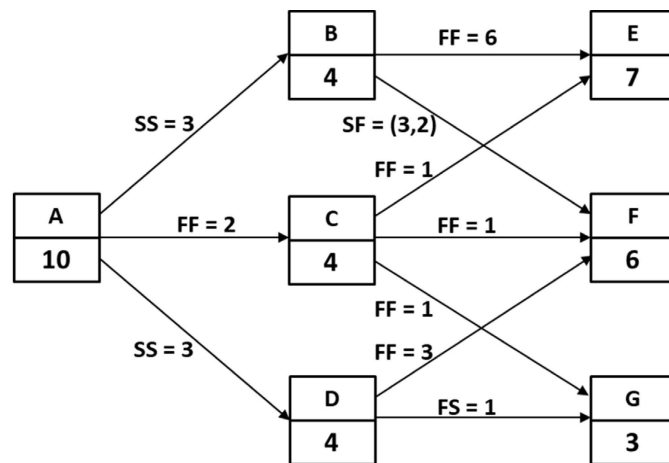


**Figure 8.** *ProFin* graphical user interface showing WBS, Gantt chart, properties panel, and toolbars.

The algorithm was tested on three published examples and one real-world project. In Case 1 [13], *ProFin* successfully handled multiple splits and clearly identified critical paths and floats. In Case 2 [5], *ProFin* matched manual solutions while providing immediate visualization of critical paths and segments. In Case 3 [11] [19], *ProFin* accurately identified partially critical segments and interruptions where commercial tools failed. Finally, in Case 4, a university campus project with 234 activities demonstrated scalability and complex logic handling, including SF relationships, producing a 124-day schedule linked to a 3D BIM model for validation and optimization.

### 3.1. Case 1—Valls *et al.* [10]

This case extends the example presented by Valls *et al.* [13], who demonstrated that the heuristic algorithm commonly used to solve PDM networks [5] may produce inconsistent results when activity splitting is allowed. They reported the complications that can arise under these conditions and proposed a modification to the original algorithm that improved numerical outcomes. However, their approach did not address the problem of clearly identifying the critical path and activity slacks. **Figure 9** shows the network of the example and the information that the scheduler would have to enter into the software.



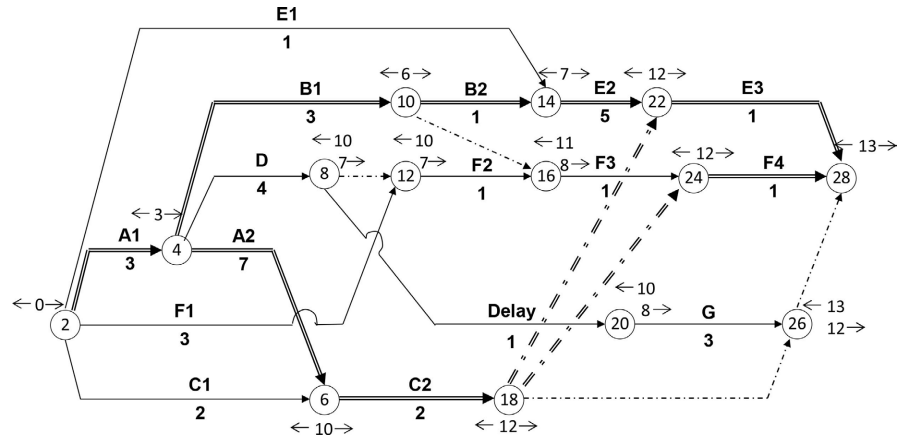
Activity PDM	Duration	Precedence – Lag Factor
1 – A	10	None
2 – B	4	A-SS3
3 – C	4	A-FF2
4 – D	4	A-FF3
5 – E	7	B-FF6; C-FF1
6 – F	6	B-SF3,2; C-FF1; D-FF3
7 – G	3	C-FF0; D-FS1

**Figure 9.** Input data for Valls *et al.* example: activities, durations, and precedence/lag values.

This example was manually resolved to compare the results with those that would be found from the computational tool. **Figure 10** shows the equivalent A-O-A network resulting from applying the transformation schemes described successively. Then, the A-O-A network was calculated by using the forward-pass and backward-pass procedures to find the early/late start/finish dates for each project activity. This figure shows 22 PDM-FS activities derived from 7 PDM-MR activities, including the 6 dummy activities that the A-O-A method requires to correctly express the logic. It also clearly shows the 3 critical paths derived in the process, which are difficult to determine only with the PDM heuristic algorithms.

**Table 1** shows the PDM-FS activities and segments resulting from the example. The collection of PDM-FS activities that are derived from a PDM-MS activity is

named from the original PDM identifier. The durations of PDM-FS activities are also displayed. Non-split PDM-FS sub-activities, derived from a PDM activity, make a segment. **Table 1** shows the 11 segments created in the network for the solution resulting from the forward pass calculation using early dates.

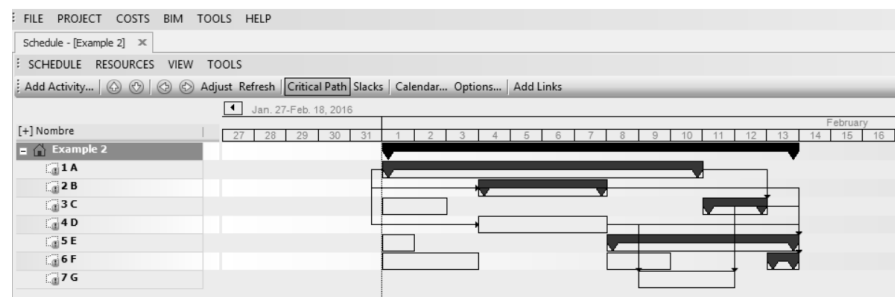


**Figure 10.** Resulting CPM (AON/AOA) network for Valls *et al.* example after transformation, including critical paths.

**Table 1.** PDM-FS activities and segments resulting from Valls *et al.* example.

PDM-MR Activities	PDM-FS Activities	Duration	Segments	PDM-MR Activities	PDM-FS Activities	Duration	Segments
	Delay	1		E	E1	1	E1
A	A1	3	A1	E	E2	5	E2
	A2	7			E3	1	E3
B	B1	3	B1		F	F1	3
	B2	1		F2		1	F2
C	C1	2	C1	F3		1	F3
	C2	2		F4		1	F4
D	D	4	D	G	G	3	G

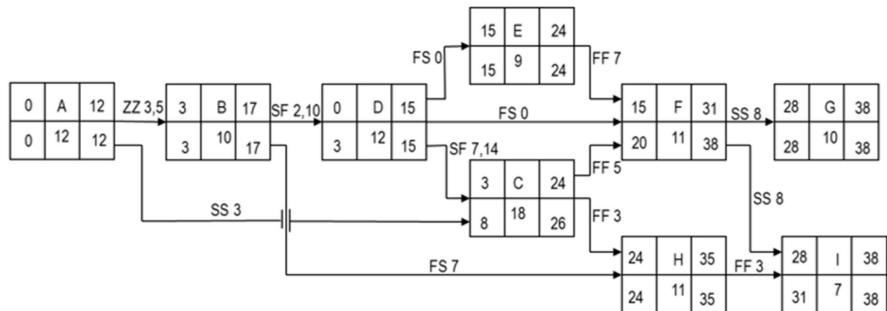
*ProFin* successfully handled multiple splits and clearly identified critical paths and floats. **Figure 11** shows the resulting Gantt chart, where activities C, E, and F are split automatically, and partially critical segments are highlighted.



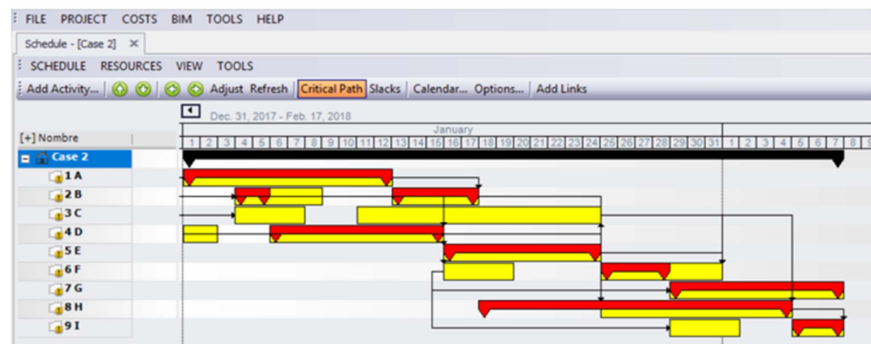
**Figure 11.** *ProFin* output for Valls *et al.* example, showing automated multi-splitting and critical path segments.

### 3.2. Case 2—Moder *et al.* [5]

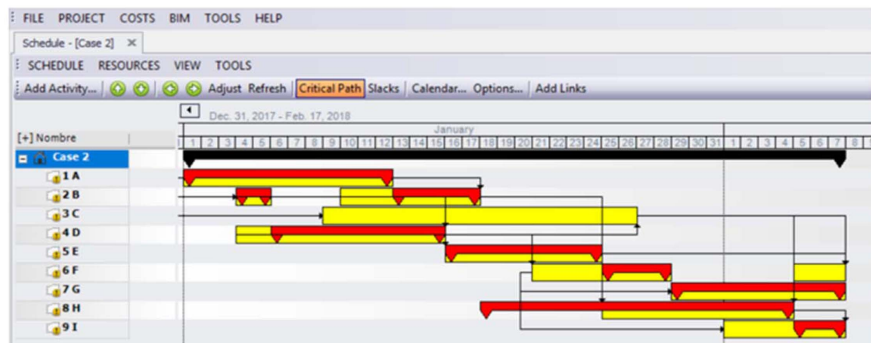
Figure 12 illustrates a textbook example originally presented by Moder *et al.* [5]. A small network with complex precedence relationships was solved manually and compared with the results generated by *ProFin*. Both solutions matched; however, *ProFin* provided immediate visualization of critical paths and segments, significantly reducing analysis complexity.



(a) Textbook example with complex relationships of the 4 types.



(b) Bar chart with early start/finish PDM-MR activities.

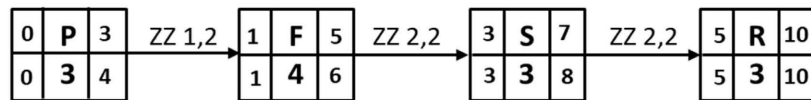


(c) Bar chart with late start/finish of PDM-MR activities.

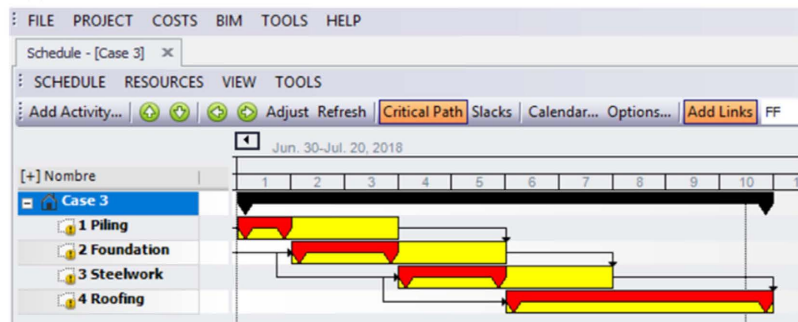
Figure 12. Case 2 composite: original network, early-date Gantt, and late-date Gantt produced by *ProFin*.

### 3.3. Case 3—Lowsley & Linnett; Hegazy & Menesi [11] [12]

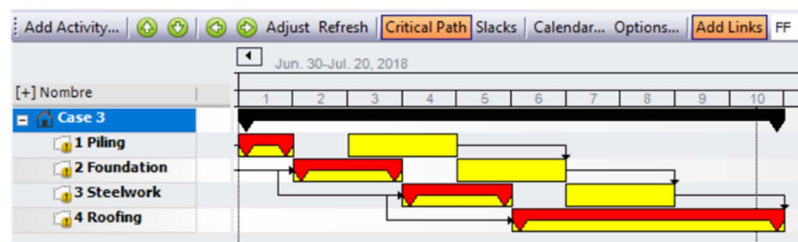
Figure 13 shows a simple case study discussed in Lowsly and Linnett [11] and Hegazy and Menesi [12]. This case demonstrates partially critical activities linked by SS and FF relationships. While Primavera P6 incorrectly marked all activities as critical, *ProFin* accurately identified critical segments and interruptions.



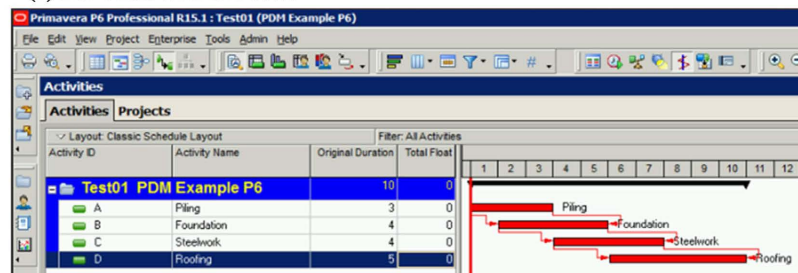
(a) PDM-MR network



(b) Bar Chart with early dates



(c) Bar Chart with late dates



(d) Bar Chart using Primavera P6

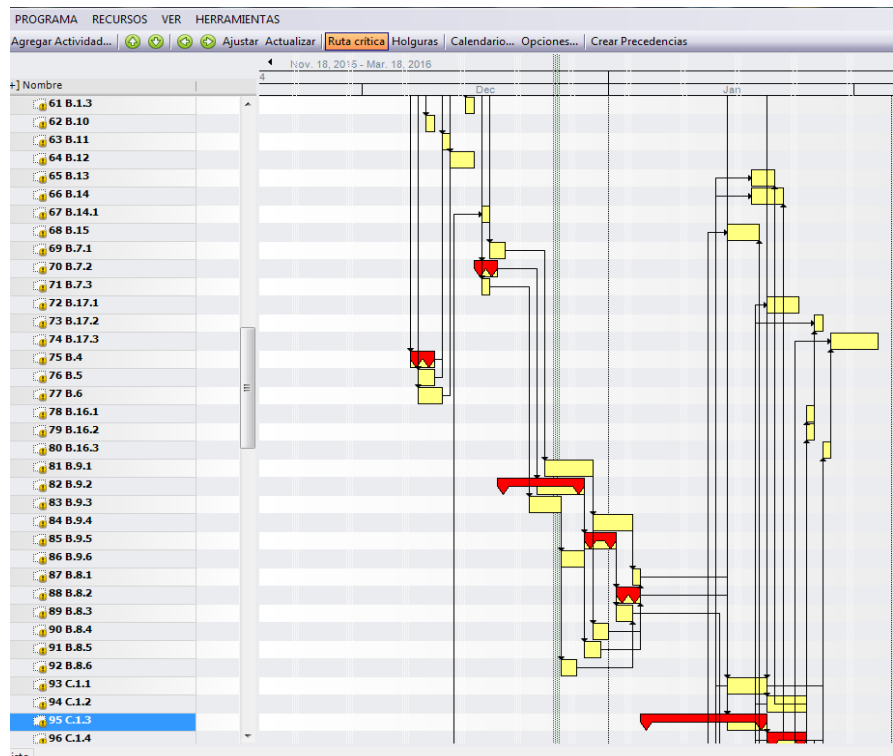
**Figure 13.** Case 3 demonstration of partially critical activities under SS and FF relationships; comparison with commercial tool outputs.

### 3.4. Case 4—Real Project

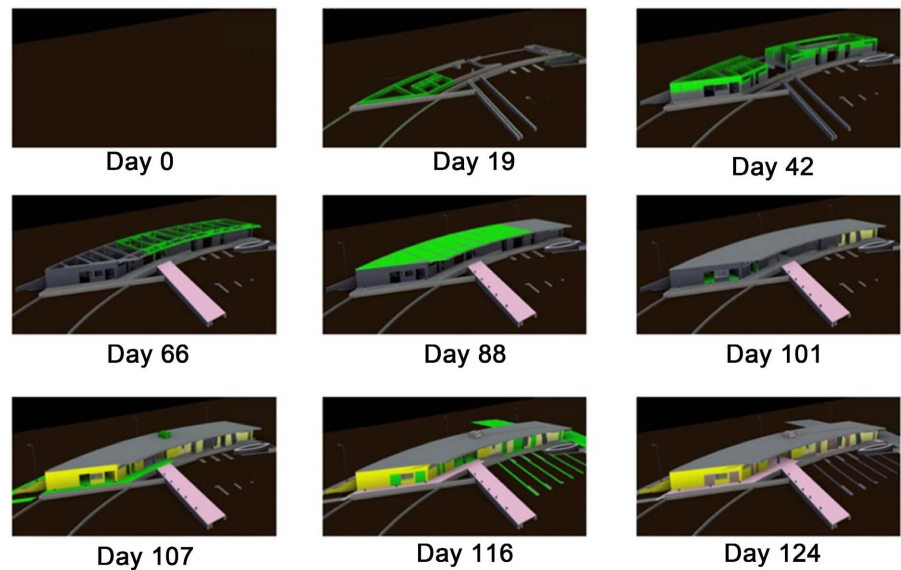
The real-world case study consisted of a university campus project comprising 234 PDM-MR activities and a dense network of mixed FS, SS, FF, and SF relationships. The transformation process resulted in approximately 470 PDM-FS sub-activities and 350 segments, while preserving all original constraints. The resulting schedule had a total duration of 124 days and enabled clear identification of fully and partially critical segments, which could not be distinguished using commercial scheduling software. This case demonstrates that the proposed algorithm remains computationally stable and logically consistent when applied to networks of practical size and complexity.

**Figure 14** presents a partial bar chart of the resulting schedule. The schedule was also linked to a 3D BIM model for validation and optimization purposes. **Figure 15** illustrates selected views of the simulated construction progress in Auto-

desk Navisworks®, using schedule data imported from *ProFin*.



**Figure 14.** Partial bar chart of the resulting schedule (Case 4).



**Figure 15.** Selected views of the simulated construction progress in Autodesk Navisworks.

#### 4. Discussion

The Precedence Diagram Method (PDM) remains the dominant scheduling technique due to its simplicity and modeling flexibility compared to alternative approaches [20]. Incorporating multiple precedence types (FS, SS, FF, SF) in PDM-

MR networks enhances productivity by enabling overlapping activities and providing schedulers with greater flexibility. However, these benefits come at the cost of increased complexity in critical path analysis, float determination, and activity splitting [4] [13] [19].

The transformation strategy proposed in this study addresses these challenges by converting PDM-MR networks into equivalent PDM-FS structures, which are easier to analyze using conventional algorithms. This approach eliminates ambiguities in float interpretation and critical path identification, overcoming limitations of heuristic algorithms that often fail to satisfy all scheduler-imposed constraints [13]. By leveraging object-oriented modeling and UML-based design, the algorithm ensures logical consistency and transparency throughout the transformation process.

Results from the case studies confirm that the proposed method provides immediate and accurate identification of critical paths, even when they consist of partially critical segments. This capability resolves a major drawback of existing commercial software, which assumes activities as indivisible blocks and therefore misrepresents criticality [11] [19]. Furthermore, the automated multi-splitting feature optimizes project duration without manual intervention, reducing scheduling effort and improving reliability.

By decomposing into time-continuous segments, the proposed method provides a natural interface with resource-based scheduling techniques, which are valuable for advanced scheduling analyses such as resource allocation, progress monitoring, and forensic delay analysis [16]. Each segment can be treated as an atomic unit for resource allocation, leveling, or simulation purposes, enabling more realistic modeling of crew continuity and productivity variations. Each segment can be treated as an independent activity for simulation purposes, enabling more accurate Monte Carlo risk assessments and resource leveling strategies. This added detail enhances short-interval planning and supports integration with BIM-based visualization tools, facilitating 4D simulation and improving decision-making during project execution [21] [22]. Although resource constraints are not addressed in the present study, the resulting segmentation establishes a clear foundation for future integration with resource-constrained scheduling and optimization methods.

The algorithm's ability to handle unlimited precedence relationships and automate splitting without exceptions represents a significant improvement over current commercial tools, which impose restrictive logic and produce inconsistent results [15] [23]. By providing immediate feedback during data entry, the proposed solution supports iterative scheduling and enables schedulers to refine logic dynamically, ultimately leading to more realistic and optimized project plans.

While the transformation process is transparent to the user, its implications are substantial: it not only simplifies complex network analysis but also establishes a solid foundation for future enhancements, including resource-constrained scheduling and deeper integration with BIM-based environments for decision support

during construction planning and control.

## 5. Conclusions

Effective scheduling is essential for achieving reliable delivery of complex construction projects. This study presented an object-oriented algorithm and a supporting computational tool that address fundamental limitations of traditional CPM and PDM-based scheduling approaches when multiple precedence relationships and activity splitting are required.

By enabling unrestricted use of FS, SS, FF, and SF relationships and automating single and multiple activity splitting, the proposed method eliminates the need for heuristic adjustments and manual interventions commonly required in commercial software. The transformation of PDM-MR networks into equivalent Finish-to-Start-based structures allows the use of conventional forward and backward pass algorithms while strictly preserving the original logical constraints imposed by the scheduler.

A key contribution of the proposed approach is the precise identification of fully and partially critical activity segments, overcoming the limitations of existing tools that treat activities as indivisible units. This capability provides clearer insight into schedule sensitivity and enables more informed decision-making during planning and project control.

The applicability and robustness of the algorithm were demonstrated through multiple published benchmark cases and a real-world construction project involving a large and complex network. Results confirm that the proposed method remains computationally stable and logically consistent when applied to networks of practical size, while significantly improving analytical transparency.

By representing activities as time-continuous segments, the method establishes a solid foundation for advanced scheduling applications, including resource allocation, short-interval control, forensic delay analysis, and BIM-based 4D simulation. Although resource constraints are not explicitly addressed in the present study, the resulting segmentation provides a natural interface for future integration with resource-constrained scheduling and optimization techniques.

Future research will focus on extending the algorithm to incorporate resource constraints, handling non-interruptible activities, and strengthening integration with active BIM environments to support dynamic and data-driven construction planning and control.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Hajdu, M. (2019) Critical Path Analysis in Precedence Diagramming Method Schedules. *Automation in Construction*, **103**, 192-202.
- [2] Andersson, N. and Christensen, K. (2007) Practical Implications of Location-Based Scheduling. *Proceedings of the Conference Construction Management and Economics*.

- Past, Present and Future*, Reading, 16-18 July 2007, 1367-1376.
- [3] McDaniel, N. and Bahnmaier, W. (2001) Scheduling Guide for Program Managers. Department of Defense.
  - [4] Wiest, J.D. (1981) Precedence Diagramming Method: Some Unusual Characteristics and Their Implications for Project Managers. *Journal of Operations Management*, **1**, 121-130. [https://doi.org/10.1016/0272-6963\(81\)90015-2](https://doi.org/10.1016/0272-6963(81)90015-2)
  - [5] Moder, J.J., Phillips, C.R. and Davis, E.W. (1983) Project Management with CPM, PERT and Precedence Diagramming. McGraw-Hill.
  - [6] Kim, J. and Chi, S. (2021) Limitations of Commercial CPM Scheduling Tools in Representing Complex Construction Logic. *Journal of Construction Engineering and Management*, **147**, Article ID: 04021044.
  - [7] Hajdu, M. and Isaac, S. (2020) Extensions of the Precedence Diagramming Method: A Review and Analysis. *Engineering, Construction and Architectural Management*, **27**, 1525-1544.
  - [8] Lu, M. and Lam, H. (2009) Transform Schemes Applied on Non-Finish-to-Start Logical Relationships in Project Network Diagrams. *Journal of Construction Engineering and Management*, **135**, 863-873. [https://doi.org/10.1061/\(asce\)co.1943-7862.0000062](https://doi.org/10.1061/(asce)co.1943-7862.0000062)
  - [9] Valls, V. and Lino, P. (2001) Criticality Analysis in Activity-on-Node Networks with Minimal Time Lags. *Annals of Operations Research*, **102**, 17-37. <https://doi.org/10.1023/a:1010941729204>
  - [10] Valls, V., Martí, R. and Lino, P. (1996) A Heuristic Algorithm for Project Scheduling with Splitting Allowed. *Journal of Heuristics*, **2**, 87-104. <https://doi.org/10.1007/bf00226294>
  - [11] Lowsley, S. and Linnett, P. (2006) About Criticality in Project Networks. *International Journal of Project Management*, **24**, 37-49.
  - [12] Hegazy, T. and Menesi, W. (2010) Critical Path Segments Scheduling Technique. *Journal of Construction Engineering and Management*, **136**, 1078-1085. [https://doi.org/10.1061/\(asce\)co.1943-7862.0000212](https://doi.org/10.1061/(asce)co.1943-7862.0000212)
  - [13] Hegazy, T. and Menesi, W. (2012) Enhancing the Critical Path Segments Scheduling Technique for Project Control. *Canadian Journal of Civil Engineering*, **39**, 968-977. <https://doi.org/10.1139/l2012-024>
  - [14] Francis, A. and Miresco, E. (2006) Discussion of “Critical Path Method with Multiple Calendars” by Kyunghwan Kim and Jesús M. De La Garza. *Journal of Construction Engineering and Management*, **132**, 542-543. [https://doi.org/10.1061/\(asce\)0733-9364\(2006\)132:5\(542\)](https://doi.org/10.1061/(asce)0733-9364(2006)132:5(542))
  - [15] Kim, K. and de la Garza, J.M. (2005) Critical Path Method with Multiple Calendars. *Journal of Construction Engineering and Management*, **131**, 330-342. [https://doi.org/10.1061/\(asce\)0733-9364\(2005\)131:3\(330\)](https://doi.org/10.1061/(asce)0733-9364(2005)131:3(330))
  - [16] Ponce de Leon, G. (2008) Project Planning Using Logic Diagramming Method. AACE International Transactions, PS.S05.1-PS.S05.6.
  - [17] Plotnick, F.L. (2006) Relationship Diagramming Method (RDM). AACE International Transactions, PS.08.1-PS.08.10.
  - [18] Hajdu, M., Skibniewski, M.J., Vanhoucke, M., Horvath, A. and Brilakis, I. (2016) How Many Types of Critical Activities Exist? A Conjecture in Need of Proof. *Procedia Engineering*, **164**, 3-11. <https://doi.org/10.1016/j.proeng.2016.11.585>
  - [19] Moon, H., Kim, H., Kamat, V.R. and Kang, L. (2015) Bim-Based Construction Scheduling Method Using Optimization Theory for Reducing Activity Overlaps. *Journal of Computing in Civil Engineering*, **29**, Article ID: 04014048.

- [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000342](https://doi.org/10.1061/(asce)cp.1943-5487.0000342)
- [20] Choi, B., Lee, H., Park, M., Cho, Y.K. and Kim, H. (2014) Framework for Work-Space Planning Using Four-Dimensional BIM in Construction Projects. *Journal of Construction Engineering and Management*, **140**, Article ID: 04014041.  
[https://doi.org/10.1061/\(asce\)co.1943-7862.0000885](https://doi.org/10.1061/(asce)co.1943-7862.0000885)
- [21] Crandall, K. (1973) Project Planning with Precedence Lead/Lag Factors. *Project Management Quarterly*, **4**, 18-27.
- [22] Fondahl, J.W. (1961) A Non-Computer Approach to the Critical Path Method for the Construction Industry. Technical Report No. 9, Stanford University.
- [23] Galloway, P.D. (2006) Survey of the Construction Industry Relative to the Use of CPM Scheduling for Construction Projects. *Journal of Construction Engineering and Management*, **132**, 697-711.  
[https://doi.org/10.1061/\(asce\)0733-9364\(2006\)132:7\(697\)](https://doi.org/10.1061/(asce)0733-9364(2006)132:7(697))