PARAMETER
JURNAL MATEMATIKA, STATISTIKA DAN TERAPANNYA

e-ISSN. 2829-6303
Vol. 05 No. 1, April 2026
Pages 185-198

Check for updates

Feasibility Regions and Critical-Path Uniqueness in Inverse Project Scheduling Using Multilayer Acyclic Digraph Models

Robby[1*], Levina Michella[2], Yanuar Bhakti Wira Tama[3]

[1, 2] Center for Mathematics and Society, Faculty of Science, Parahyangan Catholic University, Bandung, 40141, Indonesia
[3] Department of Mathematics, Institut Teknologi Kalimantan, Balikpapan, Indonesia

E-mail Correspondence Author: *robby@unpar.ac.id*

Abstract

This paper examines an inverse project scheduling problem formulated using the Critical Path Method (CPM), in which one activity's duration is unknown. The objective is to derive analytical conditions that ensure a given project duration is feasible and that a particular path becomes the unique critical path. The project workflow is represented as a multilayered acyclic digraph, which facilitates symbolic analysis of all critical path candidates. A numerical example is implemented in Python on a six-layer network with two nodes in each intermediate layer and one unknown duration. From an initial set of 16 possible paths, only 4 remain after slack-based pruning, enabling symbolic characterization of the feasibility region. The findings contribute to a deeper understanding of structural conditions that guarantee critical path uniqueness in inverse project scheduling problems.

Keywords: *Critical Path Method (CPM), multilayered acyclic digraph, project scheduling problem.*

: https://doi.org/10.30598/parameterv5i1pp185-198

1. INTRODUCTION

Project management plays a fundamental role in enabling complex projects, like infrastructure, software, or even public services, to be delivered on time, within budget, and with the maximum utilization of resources. Scheduling plays a central role in integrating interrelated tasks, minimizing delays, and allocating resources efficiently. One of the core concepts of project management is the critical path, defined as the sequence of activities whose completion determines the minimum duration of the project. Any delay along this path directly affects the overall project completion time; therefore, identifying and monitoring the critical path is essential for managerial control and risk mitigation [1], [2].

The Critical Path Method (CPM) provides a deterministic framework for analyzing project schedules. By representing activities and precedence relationships as a directed acyclic graph, CPM computes earliest and latest start times, identifies slack, and highlights critical activities [1], [2], [3]. Closely related to CPM, the Program Evaluation and Review Technique (PERT) was developed to incorporate uncertainty in activity durations by modeling them as random variables, typically using beta distributions. PERT enables the estimation of expected project duration and variance, making it particularly suitable for projects characterized by high uncertainty [2], [4].

The theoretical foundations of project scheduling were established in the late 1950s and early 1960s with the development of CPM and PERT. Kelley and Walker [1] introduced CPM as a deterministic planning and control technique for industrial projects, emphasizing activity sequencing and time estimation. In parallel, Malcolm et al. [4] developed PERT to address uncertainty in research and development projects. Fulkerson [3] extended these ideas through network flow formulations that explicitly captured cost–time relationships, laying the groundwork for time–cost trade-off analysis. These methods were later consolidated and formalized in early textbooks by Moder and Phillips [2], which played a major role in disseminating CPM and PERT into engineering and management practice. Subsequent work by Haga and Marold [5] proposed simulation-based approaches to explore time–cost trade-offs, highlighting the increasing complexity of jointly optimizing project duration and budget.

Building on these foundations, modern research has extended CPM and PERT to address practical considerations such as uncertainty, simulation, and application-specific constraints. Levner et al. [6] applied CPM/PERT principles to high-throughput screening scheduling, demonstrating their relevance beyond traditional construction settings. Comparative studies by Santoso et al. [7] and Aulia and Cipta [8] evaluated CPM and PERT in construction project planning, while Kurniawan and Mulyono [9] combined crashing strategies with Monte Carlo simulation to mitigate project delays. Other studies have focused on time–cost trade-offs and project crashing in applied engineering contexts [10], [11].

From a theoretical perspective, significant attention has been devoted to robust and resource-constrained project scheduling. Koster et al. [12] introduced Γ-robust optimization frameworks for project scheduling problems, while Bold and Goerigk [13] and Balouka and Cohen [14] proposed improved formulations for robust multi-mode resource-constrained project scheduling. Further contributions by Liu et al. [15], Davari and Demeulemeester [16], Chen et al. [17], Blázquez González et al. [18], and Zhao et al. [19] incorporated uncertainty, resource allocation policies, and scheduling flexibility into CPM-related models, substantially broadening the scope of classical project scheduling theory.

Despite the extensive literature on project scheduling based on CPM and its extensions, the vast majority of existing work addresses the forward problem, namely, determining project duration, critical paths, and slack variables given fully specified activity durations. Numerous studies have enriched this forward analysis by incorporating uncertainty, resource constraints, multiple execution modes, or cost–time trade-offs [12]-[20]. In contrast, considerably less attention has been given to the inverse problem: identifying the conditions on activity durations under which a prescribed project completion time is feasible and a particular path emerges as the unique critical path.

This inverse perspective is particularly relevant in early-stage planning, design, or negotiation phases, where some activity durations are not yet fixed and must instead satisfy global feasibility and stability requirements. Although related inverse formulations have been studied in restricted settings, such as time–cost trade-off analysis, resource-limited scheduling, or solver-based reverse planning, these approaches primarily rely on numerical optimization, simulation, or heuristic procedures and do not provide explicit structural characterizations of the feasible duration space or analytical conditions for critical path uniqueness [3], [5], [10], [12], [14], [16], [20]. In contrast, existing CPM representations typically employ flat, unstructured networks with arbitrary node labeling, which limits symbolic reasoning and systematic enumeration of alternative paths. To address these gaps, this work introduces a structured multilayered acyclic digraph framework for inverse CPM analysis that explicitly encodes stages of project evolution and supports systematic path enumeration. Unlike existing inverse scheduling and time-cost trade-off approaches, which primarily rely on optimization, simulation, or heuristic search procedures, the proposed framework derives explicit symbolic feasibility and critical-path uniqueness conditions analytically through slack-based pruning and algebraic path characterization. The multilayer representation enables transparent structural analysis of admissible duration regions and controlling paths, providing an analytical treatment of inverse CPM problems that differs fundamentally from existing optimization-driven scheduling formulations.

2. METHOD

2.1. Problem Formulation and Assumptions

In the CPM, a project is modeled using a digraph, where each node represents a distinct event or milestone in the project timeline, and each directed edge corresponds to an activity whose execution links two such events. An activity is thus represented by an edge from node $\boldsymbol{i}$ to node $\boldsymbol{j}$, indicating that the activity begins upon the occurrence of event $\boldsymbol{i}$ and must be completed before event $\boldsymbol{j}$ can occur. Within this framework, precedence relations are encoded directly in the graph: if there exists a path from activity $\boldsymbol{a}$ to activity $\boldsymbol{b}$ (i.e., if the terminal node of $\boldsymbol{a}$ is the initial node of $\boldsymbol{b}$, either directly or via intermediate events), then activity $\boldsymbol{a}$ is said to be a *predecessor* of activity $\boldsymbol{b}$, and activity $\boldsymbol{b}$ is a *successor* of activity $\boldsymbol{a}$.

Let $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ denote the directed graph representing the project schedule, where $\boldsymbol{V}$ is the set of vertices corresponding to project events, and $\boldsymbol{E}$ is the set of directed edges, each representing a distinct project activity. Associated with each edge $\boldsymbol{e}$ in $\boldsymbol{E}$ is a non-negative weight $\boldsymbol{d}(\boldsymbol{e})$, which denotes the duration required to complete the corresponding activity. The digraph $\boldsymbol{G}$ is acyclic and structured such that all activity sequences lead to a unique terminal event, typically represented as the final node in the

graph, signifying the completion of the project. To ensure structural clarity and unambiguous identification of activities, CPM conventionally disallows multiple activities from sharing the same pair of initial and terminal nodes. When such a structure is needed to encode parallel precedence constraints, a dummy activity, an edge with zero duration, is introduced.

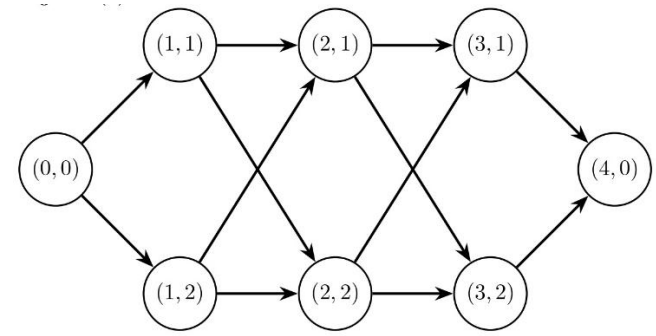
Given that a classical CPM digraph is inherently acyclic and directed toward a unique terminal node, it is natural to reinterpret its structure through the framework of a multilayered network. In this paper, we impose a structural restriction by modeling the project digraph as a cascade-type multilayer directed acyclic digraph. Any finite directed acyclic graph (DAG) can, in principle, be embedded into such a multilayer structure by assigning each node to a layer in a way that respects the directionality of the edges. This can be accomplished by computing a valid topological ordering and assigning each node a layer index based on its depth within the graph. Under this transformation, each edge in the DAG becomes an interlayer connection, linking nodes in successive layers. However, due to the generality of DAGs, the resulting multilayer representation may have irregular characteristics, such as varying layer sizes, intralayer connections, or incomplete interlayer connectivity, that complicate structural analysis. To address this, we restrict our focus to a well-defined subclass of multilayer digraphs with uniform structure and strict interlayer connectivity.

To facilitate structured analysis, this paper focuses on a simplified subclass of multilayer digraphs. We consider a network with exactly $\boldsymbol{n+1}$ layers, indexed as $\boldsymbol{L_0, L_1, \dots, L_n}$ where each $\boldsymbol{L_i}$ denotes the set of nodes occupying layer $\boldsymbol{i}$. Layers $\boldsymbol{0}$ and $\boldsymbol{n}$ each contain a single node, representing the start and end events of the project, respectively. Each intermediate layer $\boldsymbol{L_1, L_2, \dots, L_{n-1}}$ contains exactly $\boldsymbol{m}$ nodes. For clarity and analytical simplicity, we impose the following structural constraints:

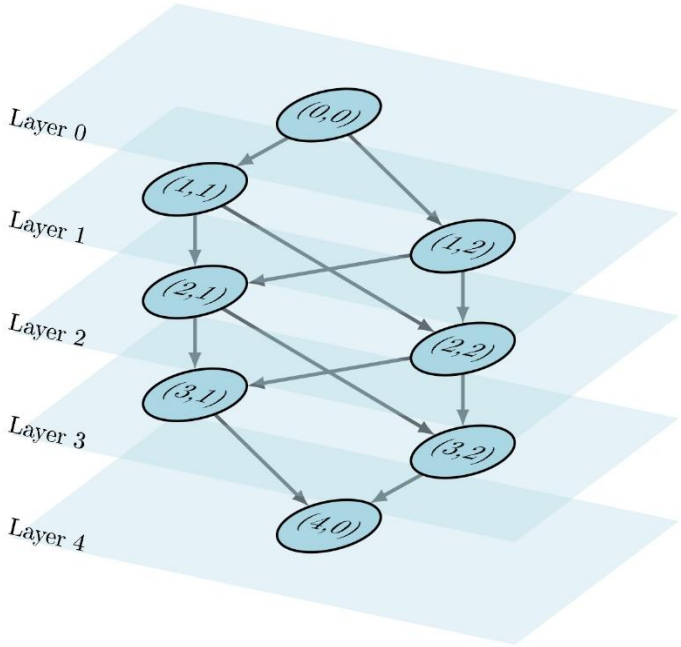
1. No intralayer edges: Activities (i.e., directed edges) may only occur between nodes in different layers, never within the same layer.
2. Adjacent-layer connectivity: Every activity connects a node in layer $\boldsymbol{L_i}$ to a node in layer $\boldsymbol{L_{i+1}}$, for $\boldsymbol{i \in \{0, 1, \dots, n\}}$.
3. Full bipartite interlayer connections: For every $\boldsymbol{i \in \{0, 1, \dots, n-1\}}$, and for every node $\boldsymbol{u \in L_i}$ and $\boldsymbol{v \in L_{i+1}}$, the edge $\boldsymbol{(u, v)}$ exists in the network.

To formalize the multilayer structure, we index each node in the network as a pair $\boldsymbol{(i, j)}$, where $\boldsymbol{i \in \{0, 1, \dots, n\}}$ denotes the layer index, and $\boldsymbol{j}$ is the position of the node within layer $\boldsymbol{i}$. Thus, the set of nodes in layer $\boldsymbol{i}$ is written as $\boldsymbol{L_i = \{(i, j): 1 \le j \le m\}}$, while $\boldsymbol{L_0 = \{(0, 0)\}}$ and $\boldsymbol{L_n = \{(n, 0)\}}$ contain the unique start and end nodes of the project, respectively. Each activity is represented by a directed edge from node $\boldsymbol{(p, q)}$ to node $\boldsymbol{(r, s)}$, denoted $\boldsymbol{\big((p, q), (r, s)\big)}$, and is assigned a non-negative duration $\boldsymbol{d_{(p,q)\to(r,s)} \in \mathbb{R}_{\ge 0}}$. Under the structural assumptions of this paper, all activities are strictly interlayer: they connect only adjacent layers, meaning $\boldsymbol{r = p + 1}$. Furthermore, we assume full interlayer connectivity, i.e., for each layer index $\boldsymbol{i \in \{0, 1, \dots, n\}}$, and for every node $\boldsymbol{(i, j) \in L_i}$ and $\boldsymbol{(i+1, k) \in L_{i+1}}$, the activity $\boldsymbol{\big((i, j), (i+1, k)\big)}$ exists in the network. An illustration indicating comparison between DAG and multilayer structure can be seen in **Figure 1**.

The structural assumptions imposed in this study, including uniform intermediate layer size and full bipartite interlayer connectivity, are introduced primarily to facilitate symbolic analysis and complete path enumeration in the inverse CPM setting. Although real-world project networks are often sparse and structurally irregular, any finite directed acyclic graph can still be embedded into a multilayer representation through topological layering.



(a)



(b)

Figure 1. An Example of Project Activities Representation with (a) A Directed Acyclic Graph and (b) Cascade-type Multilayer Acyclic Digraph.

2.2. CPM Algorithm on Multilayered Acyclic Digraph

The Critical Path Method (CPM) is a fundamental technique in project scheduling that aims to determine the earliest and latest time at which each event or activity in a project can occur without delaying the overall completion time.

Given the durations of all activities and the precedence constraints encoded in a directed acyclic graph, CPM computes three key quantities:

- The **earliest time** an event can occur, obtained through a forward pass.
- The **latest time** an event can occur without delaying the project, obtained through a backward pass.
- The **slack time** of each activity or event, used to identify the critical path, a sequence of activities with zero slack that dictates the project's minimum duration.

This method produces three key outputs that are fundamental to project scheduling. First, it identifies the critical path, which is the sequence of activities that directly determines the minimum time required to complete the project; any delay in these activities will delay the entire project. Second, it computes the project duration, defined as the earliest possible completion time of the final event in the schedule. Third, it provides the slack times for all non-critical activities,

indicating the amount of time each activity can be delayed without affecting the overall project completion.

In our multilayered framework, each event is represented by a node of the form $(\boldsymbol{i},\boldsymbol{j})$, where $\boldsymbol{i}$ indicates the layer and $\boldsymbol{j}$ identifies the node within that layer. Each activity corresponds to a directed edge $\big((\boldsymbol{p},\boldsymbol{q}),(\boldsymbol{r},\boldsymbol{s})\big)$ and is associated with a non-negative duration $\boldsymbol{d}_{(\boldsymbol{p},\boldsymbol{q})\to(\boldsymbol{r},\boldsymbol{s})}$. The precedence relation requires that activity $\big((\boldsymbol{p},\boldsymbol{q}),(\boldsymbol{r},\boldsymbol{s})\big)$ can start only after event $(\boldsymbol{p},\boldsymbol{q})$ has occurred, and that event $(\boldsymbol{r},\boldsymbol{s})$ occurs after the activity is completed.

To model and compute the timing of events efficiently, we employ the framework of max-plus algebra, which is particularly well-suited for systems governed by synchronization and delay constraints. In max-plus algebra, the conventional addition and multiplication operations are replaced by

- maximum: $\boldsymbol{a}\oplus\boldsymbol{b}\coloneqq\mathbf{max}\{\boldsymbol{a},\boldsymbol{b}\}$ and
- addition: $\boldsymbol{a}\otimes\boldsymbol{b}\coloneqq\boldsymbol{a}+\boldsymbol{b}$.

In the max–plus semiring, the symbol $-\infty$ serves as the additive identity for the maximum operation, meaning that for any real number $\boldsymbol{a}$, taking the maximum with $-\infty$ returns $\boldsymbol{a}$ itself. It also satisfies the property that adding $-\infty$ to any element yields $-\infty$. Conversely, in the min–plus semiring, the operation $\boldsymbol{a}\ominus\boldsymbol{b}$ denotes $\mathbf{min}\{\boldsymbol{a},\boldsymbol{b}\}$, while $\boldsymbol{a}\oslash\boldsymbol{b}$ denotes the arithmetic subtraction $\boldsymbol{a}-\boldsymbol{b}$. The symbol $+\infty$ serves as the additive identity for the minimum operation, meaning that for any real number $\boldsymbol{a}$, taking the minimum with $+\infty$ returns $\boldsymbol{a}$. It likewise satisfies the property that adding $+\infty$ to any element yields $+\infty$. These two special elements extend the real numbers to $\mathbb{R}\cup\{-\infty,+\infty\}$ and are essential for representing unreachable or unconstrained values in max–plus and min–plus computations.

In the CPM framework, each activity $\big((\boldsymbol{p},\boldsymbol{q}),(\boldsymbol{r},\boldsymbol{s})\big)$ is associated with four fundamental scheduling quantities. The earliest start time (ES) is the earliest time at which the activity can begin, determined by the occurrence time of its starting event $(\boldsymbol{p},\boldsymbol{q})$. The earliest finish time (EF) is obtained by adding the activity's duration $\boldsymbol{d}_{(\boldsymbol{p},\boldsymbol{q})\to(\boldsymbol{r},\boldsymbol{s})}$ to its ES value. The latest finish time (LF) is the latest time at which the activity can be completed without delaying the overall project, while the latest start time (LS) is obtained by subtracting the duration from LF. Within this algebraic framework, the forward CPM recursion corresponds to repeated max-plus aggregation of predecessor completion times, whereas the backward recursion corresponds to min-plus propagation of admissible successor times. Consequently, the classical CPM forward-backward procedure can be expressed entirely in terms of max-plus and min-plus algebraic operations. The detailed version of this procedure is presented in **Algorithm 1**.

Algorithm 1. CPM in a Multilayered Acyclic Digraph using Max-Plus/Min-Plus Algebra

Input:	Number of layers n (indexed from 0) and number of nodes each layer m. Layers $L_0, L_1, \dots, L_n$ with $L_0 = \{(0,0)\}$ and $L_n = \{(n,0)\}$. Activity set $A = \{((p,q),(r,s)) : (p,q) \in L_i, (r,s) \in L_{i+1}\}$. Durations $d_{(p,q)\to(r,s)} \geq 0$ for each activity.	
Process:	1.	**Forward pass initialization:** Set $E_{(0,0)} \leftarrow 0$ and $E_{(i,j)} \leftarrow -\infty$ otherwise.
	2.	**Forward pass:** For all $i \in \{1,2,\dots,n\}$: $E_{(i,j)} \leftarrow \bigoplus_{(p,q)\in L_{i-1}} \left(E_{(p,q)} \otimes d_{(p,q)\to(i,j)}\right)$.
	3.	**ES and EF**: For each $((p,q),(r,s)) \in A$: $ES_{(p,q)\to(r,s)} \leftarrow E_{(p,q)}$ and $EF_{(p,q)\to(r,s)} \leftarrow E_{(p,q)} \otimes d_{(p,q)\to(r,s)}$.
	4.	**Project duration:** $T_{proj} = E_{(n,0)}$.
	5.	**Backward pass initialization:** Set $L_{(n,0)} \leftarrow T_{\text{proj}}$ and $L_{(i,j)} \leftarrow +\infty$ otherwise.
	6.	**Backward pass:** For all $i \in \{n-1, n-2, \dots, 0\}$: $L_{(i,j)} \leftarrow \bigominus_{(r,s)\in L_{i+1}} \left(L_{(r,s)} \oslash d_{(i,j)\to(r,s)}\right)$.
	7.	**LS and LF**: For each $((p,q),(r,s)) \in A$: $LF_{(p,q)\to(r,s)} \leftarrow L_{(r,s)}$ and $LS_{(p,q)\to(r,s)} \leftarrow L_{(r,s)} \oslash d_{(p,q)\to(r,s)}$.
	8.	**Slack**: $S_{(p,q)\to(r,s)} \leftarrow LS_{(p,q)\to(r,s)} \oslash ES_{(p,q)\to(r,s)}$.
	9.	**Critical paths:** Define $E_c = \{((p,q),(r,s)) \in A : S_{(p,q)\to(r,s)} = 0\}$. A critical path is any maximal sequence of activities in E_c starting at $(0,0)$ and ending at $(n,0)$.
Output:	Earliest start time $ES_{(p,q)\to(r,s)}$, earliest finish time $EF_{(p,q)\to(r,s)}$, latest start time $LS_{(p,q)\to(r,s)}$, latest finish time $LF_{(p,q)\to(r,s)}$, slack time $S_{(p,q)\to(r,s)}$, project duration T_{proj}, and critical path(s).	

3. INVERSE CPM ON MULTILAYERED ACYCLIC DIGRAPH

Having established the max–plus and min–plus formulation of the classical CPM in the previous chapter, we now turn to the **inverse** scheduling problem. In practical project management, it is common to face situations where a project deadline is fixed, yet one or more activity durations remain unknown at the planning stage. These uncertainties may arise from incomplete information, resource allocation negotiations, or dependency on external deliverables. If the network structure and the remaining activity durations are known, it is possible, by tracing through CPM relations, to determine the set of feasible values for the unknown duration that ensure the project is completed on or before the specified deadline. This forms the basis of the **inverse CPM** problem on a multilayered acyclic digraph.

We focus first on the simplest setting of the inverse problem. The information available to us is the expected project duration T_{exp}. All activity durations $d_{(p,q)\to(r,s)}$ are known except for one, denoted by x, which represents the duration of a single activity $((p^*,q^*),(r^*,s^*))$ in the network. The objective is to determine the feasible range of values for x such that the resulting schedule is consistent with the given T_{exp} and critical path structure.

The framework for the inverse CPM algorithm follows the same max–plus/min–plus principles as in the classical case, but with adjustments to account for the unknown duration. The key idea is to execute the CPM procedure **as far as possible** using the available information, while leaving expressions involving the unknown duration x unevaluated. This partial computation provides two immediate benefits. First, it yields earliest and latest times for all events that are independent of x, allowing direct calculation of slack times where possible. Second, any activity found to have **positive**

slack under the known durations can be excluded from consideration, as it cannot belong to any feasible critical path for the given T_{exp}. This pruning step reduces the search space for possible critical paths and focuses attention on the subnetwork where x may influence the project duration.

To systematically identify the influence of the unknown duration x, we initialize its value as $+\infty$ in the forward pass computation. In the max–plus formulation, this ensures that any ES or EF time depending on x will also evaluate to $+\infty$, thereby marking all nodes and activities downstream of $\big((p^*, q^*), (r^*, s^*)\big)$ as dependent on x. This approach cleanly separates the network into two regions: the **upstream region**, where all times can be computed exactly using known durations, and the **downstream region**, where times are expressed in terms of x. By tracking this propagation, we can later substitute x back as a symbolic variable and derive explicit equations linking it to the target project duration T_{exp} and the given critical path.

The **feasibility condition** requires that the unknown activity duration x be such that the resulting project duration does not exceed the expected project duration T_{exp}. This inequality defines an interval or a set of admissible values for x. The **uniqueness condition** holds when a single controlling path, containing the unknown activity, determines the project duration for all feasible values of x. If multiple paths can achieve the maximum duration for some feasible x, then the solution is not unique, and additional information would be required to select a single value. The complete procedure for implementing this approach is summarized in **Algorithm 2.**

Algorithm 2. Inverse CPM on a Multilayered Acyclic Digraph with One Unknown Duration

<table>
<tr><td>Input:</td><td colspan="2">Number of layers n (indexed from 0) and number of nodes each layer m.
Layers $L_0, L_1, \dots, L_n$ with $L_0 = \{(0,0)\}$ and $L_n = \{(n,0)\}$.
Activity set $A = \{((p,q),(r,s)) : (p,q) \in L_i, (r,s) \in L_{i+1}\}$.
Durations $d_{(p,q)\to(r,s)} \geq 0$ for each activity except one unknown x for activity $\big((p^*,q^*),(r^*,s^*)\big)$.
Expected project duration T_{exp}.</td></tr>
<tr><td rowspan="8">Process:</td><td>1.</td><td>Set $d_{(p^*,q^*)\to(r^*,s^*)} \leftarrow +\infty$.</td></tr>
<tr><td>2.</td><td>Do forward pass, obtaining ES and EF.</td></tr>
<tr><td>3.</td><td>Do backward pass, obtaining LS and LF by setting $L_{(n,0)} \leftarrow T_{\text{exp}}$.</td></tr>
<tr><td>4.</td><td>Slack: For each $\big((p,q),(r,s)\big) \in A$:
$S_{(p,q)\to(r,s)} \leftarrow LS_{(p,q)\to(r,s)} \oslash ES_{(p,q)\to(r,s)}$.</td></tr>
<tr><td>5.</td><td>If $S_{(p,q)\to(r,s)} > 0$, exclude the activity from possible controlling paths for T_{exp}.</td></tr>
<tr><td>6.</td><td>Set $d_{(p^*,q^*)\to(r^*,s^*)} \leftarrow x$.
Enumerate all directed paths from $(0,0)$ to $(n,0)$.</td></tr>
<tr><td>7.</td><td>Let $P = \{P_1, P_2, \dots, P_\ell\}$ be the set of candidate critical paths in step 6. Compute the project duration
$T_{\text{proj}}(x) = \bigoplus_{1\leq k\leq \ell} \bigotimes_{((p,q),(r,s))\in P_k} d_{(p,q)\to(r,s)}$.</td></tr>
<tr><td>8.</td><td>Feasibility condition: The solution of system of linear inequality
$T_{\text{proj}}(x) \leq T_{\text{exp}}$
yields the feasible condition of x (if it exists).

Uniqueness condition: For a given feasible x, the critical path is unique if and only if exactly one path attains the project duration $T_{\text{proj}}(x)$.</td></tr>
<tr><td>Output:</td><td colspan="2">Feasibility and uniqueness condition of x.</td></tr>
</table>

4. NUMERICAL EXAMPLE

In this section, we present a numerical example to illustrate the application of the proposed algorithms for inverse CPM on a multilayered acyclic digraph. The network under consideration consists of $n+1=6$ layers, indexed from L_0 to L_5, where $L_0 = \{(0,0)\}$ represents the project start event and $L_5 = \{(5,0)\}$ represents the project completion event. Each intermediate layer L_i for $1 \le i \le 4$ contains $m=2$ nodes, indexed as $(i,1)$ and $(i,2)$. All activities are directed from one layer to its immediate successor layer, and each activity $\big((p,q),(r,s)\big)$ is assigned a non-negative duration $d_{(p,q)\to(r,s)}$. For this example, all durations are known except for a single activity $\big((2,2),(3,2)\big)$, whose duration is denoted by x and will be determined through the inverse CPM procedure. The expected project duration is set to $T_{\text{exp}} = 20$. The full set of activities and their durations is given in **Table 1**, with the unknown duration indicated explicitly.

Table 1. Activity Durations in the Multilayer Network

From	To	Duration	From	To	Duration	From	To	Duration
(0,0)	(1,1)	1	(2,1)	(3,1)	5	(3,2)	(4,1)	4
(0,0)	(1,2)	2	(2,1)	(3,2)	1	(3,2)	(4,2)	3
(1,1)	(2,1)	2	(2,2)	(3,1)	2	(4,1)	(5,0)	3
(1,1)	(2,2)	3	(2,2)	(3,2)	x	(4,2)	(5,0)	5
(1,2)	(2,1)	1	(3,1)	(4,1)	1			
(1,2)	(2,2)	1	(3,1)	(4,2)	2			

Using the parameters and activity durations in **Table 1**, we perform the forward–backward computations of **Algorithm 2** while assigning $x = +\infty$ for the unknown duration. This approach allows the CPM procedure to propagate earliest and latest times as far as possible using only the known values, while marking as "unknown" any quantities that depend on x. The resulting partial schedule information is presented in **Table 2**.

Table 2. Partial CPM Results

From	To	Duration	ES	EF	LS	LF	Slack
(0,0)	(1,1)	1	0	1	unknown	unknown	unknown
(0,0)	(1,2)	2	0	2	unknown	unknown	unknown
(1,1)	(2,1)	2	1	3	6	8	5
(1,1)	(2,2)	3	1	4	unknown	unknown	unknown
(1,2)	(2,1)	1	2	3	7	8	5
(1,2)	(2,2)	1	2	3	unknown	unknown	unknown
(2,1)	(3,1)	5	3	8	8	13	5
(2,1)	(3,2)	1	3	4	11	12	8
(2,2)	(3,1)	2	4	6	11	13	7
(2,2)	(3,2)	unknown	4	unknown	unknown	12	unknown
(3,1)	(4,1)	1	8	9	16	17	8
(3,1)	(4,2)	2	8	10	13	15	5
(3,2)	(4,1)	4	unknown	unknown	13	17	unknown
(3,2)	(4,2)	3	unknown	unknown	12	15	unknown
(4,1)	(5,0)	3	unknown	unknown	17	20	unknown
(4,2)	(5,0)	5	unknown	unknown	15	20	unknown

From the partial CPM results in **Table 2**, any activity with a known positive slack can be excluded from further consideration, as such activities cannot belong to any

controlling path for the project duration. Removing these activities yields a reduced, or **pruned**, multilayered acyclic digraph that contains only the unknown activity and those with zero slack or slack values dependent on x. This pruned network is illustrated in **Figure 2**.

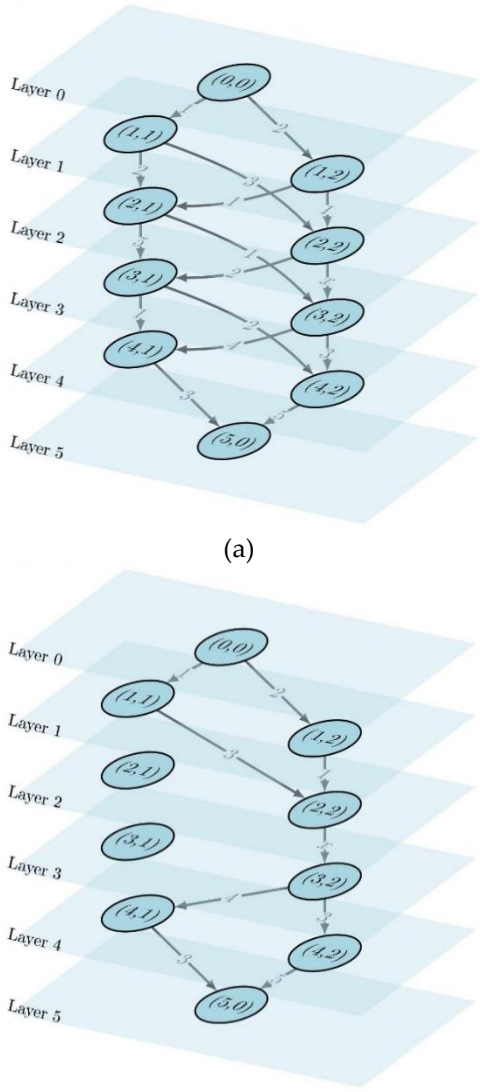


Figure 2. Multilayered Acyclic Digraph Representation for Numerical Example: (a) Before Pruning and (b) After Pruning.

By eliminating activities with known positive slack, the number of possible critical paths in the network is significantly reduced. In the original multilayered acyclic digraph, there were 16 potential critical paths from the start event to the project completion event. After pruning, only 4 candidate paths remain. This reduction greatly improves computational efficiency, as subsequent feasibility and uniqueness checks in the inverse CPM need only be performed on this smaller set of paths.

The four remaining candidate paths in the pruned network are listed in **Table 3** along with their total durations expressed algebraically in terms of the unknown activity duration x (corresponding to $\big((2,2),(3,2)\big)$). All other durations are known constants, and the total duration of each path is obtained by summing the durations of its constituent activities.

Table 3. Candidate Critical Paths After Pruning and Their Symbolic Total Durations

Path Candidate	Total Duration
$((0,0),(1,1)) \rightarrow ((1,1),(2,2)) \rightarrow ((2,2),(3,2)) \rightarrow ((3,2),(4,1)) \rightarrow ((4,1),(5,0))$	$x+11$
$((0,0),(1,1)) \rightarrow ((1,1),(2,2)) \rightarrow ((2,2),(3,2)) \rightarrow ((3,2),(4,2)) \rightarrow ((4,2),(5,0))$	$x+12$
$((0,0),(1,2)) \rightarrow ((1,2),(2,2)) \rightarrow ((2,2),(3,2)) \rightarrow ((3,2),(4,1)) \rightarrow ((4,1),(5,0))$	$x+10$
$((0,0),(1,2)) \rightarrow ((1,2),(2,2)) \rightarrow ((2,2),(3,2)) \rightarrow ((3,2),(4,2)) \rightarrow ((4,2),(5,0))$	$x+11$

From this, the project duration can be expressed as

$$T_{\text{proj}}(x) = (x+11) \oplus (x+12) \oplus (x+10) \oplus (x+11).$$

The feasibility condition requires that $T_{\text{proj}}(x) \leq T_{\text{exp}} = 20$, which yields the system of linear inequalities

$$\begin{cases} x+10 \leq 20, \\ x+11 \leq 20, \\ x+12 \leq 20. \end{cases}$$

Hence, the feasibility condition can be reduced to $0 \leq x \leq 8$. Moreover, for the uniqueness condition, since $x+12$ is strictly greater than the other three expressions for all x, it always determines the maximum as long as x satisfies the feasibility condition. Therefore, for every $0 \leq x \leq 8$, the critical path is unique and corresponds to the path

$$((0,0),(1,1)) \rightarrow ((1,1),(2,2)) \rightarrow ((2,2),(3,2)) \rightarrow ((3,2),(4,2)) \rightarrow ((4,2),(5,0)).$$

The numerical example illustrates how the proposed inverse CPM framework complements and extends existing project scheduling studies. In classical CPM and PERT analyses, numerical examples are typically used to compute critical paths and slack values under fully specified activity durations [1], [2], [4]. Similarly, studies on crashing and time–cost trade-offs employ numerical cases to evaluate how modifying selected activity durations affects overall project duration and cost, often relying on repeated forward computations or optimization procedures [5], [10], [20]. In contrast, the present example demonstrates a fundamentally different use of numerical analysis: rather than adjusting durations iteratively, the multilayered digraph structure enables systematic pruning of infeasible paths and symbolic derivation of feasibility and uniqueness conditions for an unknown duration. This aligns conceptually with robust and resource-constrained scheduling studies that seek stability of project completion under uncertainty [12], [14], [16], but differs in that the results here provide explicit analytical conditions rather than scenario-based or solver-dependent solutions. The reduction of candidate critical paths from 16 to 4 in this example highlights how structural pruning can significantly simplify inverse reasoning, offering a transparent alternative to simulation-based or optimization-driven approaches reported in related literature [9], [15], [19]. As such, the numerical results do not merely validate the algorithm, but demonstrate how inverse CPM analysis on multilayered acyclic digraphs provides insights that are not accessible through existing forward or optimization-based project scheduling methods.

5. CONCLUSION

This study presented a max–plus and min–plus algebraic framework for solving both the classical and inverse Critical Path Method (CPM) on multilayered acyclic digraphs. The classical CPM formulation was adapted to the multilayer structure, enabling systematic computation of earliest and latest event times, slack values, and critical paths. The inverse CPM procedure was then developed for the case of a single unknown activity duration, introducing a pruning strategy based on positive slack elimination to reduce the search space for possible critical paths. The proposed approach offers a clear and computationally efficient methodology for addressing inverse project scheduling problems with incomplete duration information. Its combination of algebraic modeling and structural pruning makes it particularly suitable for large-scale networks where exhaustive path enumeration would otherwise be computationally prohibitive. The framework developed in this study focused on the inverse CPM problem with a single unknown activity duration in a multilayered acyclic digraph. Several natural extensions merit further investigation. One direction is to generalize the method to handle multiple unknown durations, which would require solving systems of inequalities in multiple variables and could introduce coupling effects between different unknowns. Another avenue is to explore the application of the inverse CPM framework to other classes of project networks, such as partially layered acyclic graphs, graphs with non-uniform connectivity, or networks incorporating dummy activities for more complex precedence relationships. Additionally, extending the approach to accommodate stochastic durations or fuzzy time estimates could make it more robust in environments with high uncertainty. From a practical perspective, the proposed inverse CPM framework may be particularly useful during early-stage project planning, negotiation-based scheduling, or infrastructure design processes where some activity durations are not yet fixed, but overall completion targets are predetermined. The ability to derive explicit feasibility intervals and unique controlling paths analytically may support decision-making under incomplete scheduling information without requiring repeated optimization or simulation procedures. These directions would broaden the applicability of the method and enhance its practical relevance in diverse project scheduling contexts.

Funding Information

This research was made possible through the research funding provided by LPPM Institut Teknologi Kalimantan in 2025. We sincerely thank them for their support.

Author Contributions Statement

In accordance with the Contributor Roles Taxonomy (CRediT), the specific contributions of each author to this study are summarized as the following. All authors have made substantial intellectual contributions to the work and meet the established criteria for authorship. Robby contributed to the conceptualization of the research, development of the methodology, formal analysis, investigation, and preparation of the original draft, as well as visualization of results. Levina Michella contributed primarily to software development, validation of the proposed framework, investigation, visualization, and review and editing of the manuscript. Yanuar Bhakti W. T. contributed to conceptualization, oversight of the research process, supervision, project administration, and acquisition of funding.

Conflict of Interest Statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. The authors

also declare no non-financial competing interests, such as political, personal, religious, ideological, academic, or intellectual conflicts.

Data Availability

Data availability is not applicable to this article as no external datasets were generated or analyzed during the current study. All results are derived from symbolic and structural analysis conducted within the paper.

REFERENCES

[1] E. Kelley Jr. and M. R. Walker, "Critical path planning and scheduling," in *Proceedings of the Eastern Joint Computer Conference*, Boston, MA, USA, 1959, pp. 160–173.

[2] J. J. Moder and C. R. Phillips, *Project Management with CPM and PERT*, 2nd ed. New York, NY: USA: Reinhold Publishing, 1964.

[3] D. R. Fulkerson, "A network flow computation for project cost curves," *Manage Sci*, vol. 7, no. 2, pp. 167–179, 1961.

[4] D. G. Malcolm, J. H. Roseboom, C. E. Clark, and W. Fazar, "Application of a Technique for Research and Development Program Evaluation," *Oper Res*, vol. 7, no. 5, pp. 646–669, 1959.

[5] W. A. Haga and K. A. Marold, "A Simulation Approach to the PERT/CPM Time-Cost Trade-Off Problem," *Project Management Journal*, vol. 35, no. 2, pp. 31–37, Jun. 2004, doi: 10.1177/875697280403500205.

[6] E. Levner, V. Kats, P. Yan, and A. Che, "Fast Algorithm for High-Throughput Screening Scheduling Based on the PERT/CPM Project Management Technique," *Algorithms*, vol. 17, no. 3, p. 127, Mar. 2024, doi: 10.3390/a17030127.

[7] F. W. Santoso, F. S. Handayani, and Setiono, "A Comparative Analysis of the CPM and PERT Methods in Project Time Management for a High-Rise Building Construction Project in Yogyakarta," *Sustainable Civil Building Management and Engineering Journal*, vol. 2, no. 2, p. 13, May 2025, doi: 10.47134/scbmej.v2i2.3927.

[8] S. Aulia and H. Cipta, "Network Planning Analysis Using CPM and PERT Methods on Optimization of Time and Cost," *Sinkron*, vol. 8, no. 1, pp. 171–177, Jan. 2023, doi: 10.33395/sinkron.v8i1.11961.

[9] B. N. Kurniawan and N. B. Mulyono, "Minimizing Delay in Construction Project at PT Freeport Indonesia using Crashing CPM-PERT Approach and Monte Carlo Simulation," *European Journal of Business and Management Research*, vol. 9, no. 5, pp. 61–69, Sep. 2024, doi: 10.24018/ejbmr.2024.9.5.2448.

[10] R. T. Prasetyawidandi, R. N. Rachmadita, D. A. Utari, and L. E. Puspandari, "Analisis Optimasi Waktu dan Biaya Crashing Project Dengan Metode Critical Path dan Time Cost Trade Off Pada Proyek Pembangunan Galangan Kapal (Studi Kasus PT. Batam Expresindo Shipyard)," *Jurnal Teknologi Maritim* , vol. 5, no. 2, 2022.

[11] Y. F. Ryandre and G. Sarya, "EVALUASI WAKTU DAN BIAYA MENGGUNAKAN METODE CRITICAL PATH METHOD DAN CRASHING PADA PROYEK PEMBANGUNAN GEDUNG ARSIP TRENGGALEK," *Jurnal Taguchi : Jurnal Ilmiah Keilmuan Teknik dan Manajemen Industri* , vol. 3, no. 2, pp. 1463–1473, 2023.

[12] A. M. C. A. Koster, J. Segschneider, and N. Ventsch, "Γ-robust optimization of project scheduling problems," *Comput Oper Res*, vol. 161, p. 106453, Jan. 2024, doi: 10.1016/j.cor.2023.106453.

[13] M. Bold and M. Goerigk, "A faster exact method for solving the robust multi-mode resource-constrained project scheduling problem," *Operations Research Letters*, vol. 50, no. 5, pp. 581–587, Sep. 2022, doi: 10.1016/j.orl.2022.08.003.

[14] N. Balouka and I. Cohen, "A robust optimization approach for the multi-mode resource-constrained project scheduling problem," *Eur J Oper Res*, vol. 291, no. 2, pp. 457–470, Jun. 2021, doi: 10.1016/j.ejor.2019.09.052.

[15] W. Liu, L. Ge, C. Qu, and S. Yang, "Bi-objective Optimization for Resource-constrained Robust Construction Project Scheduling," *KSCE Journal of Civil Engineering*, vol. 28, no. 1, pp. 15–28, Jan. 2024, doi: 10.1007/s12205-023-0633-8.

[16] M. Davari and E. Demeulemeester, "Important classes of reactions for the proactive and reactive resource-constrained project scheduling problem," *Ann Oper Res*, vol. 274, no. 1–2, pp. 187–210, Mar. 2019, doi: 10.1007/s10479-018-2899-7.

[17] Z. Chen, E. Demeulemeester, S. Bai, and Y. Guo, "Efficient priority rules for the stochastic resource-constrained project scheduling problem," *Eur J Oper Res*, vol. 270, no. 3, pp. 957–967, Nov. 2018, doi: 10.1016/j.ejor.2018.04.025.

[18] J. Blázquez González, J. A. Barro, and E. Rubio Romero, "Construction project scheduling considering resource leveling and slack sensitivity," *Applied Sciences*, vol. 11, no. 16, 2021.

[19] Y. Zhao, X. Hu, J. Wang, and N. Cui, "A robust multi-project scheduling problem under a resource dedication-transfer policy," *Ann Oper Res*, vol. 337, no. 1, pp. 425–457, Jun. 2024, doi: 10.1007/s10479-024-05854-4.

[20] A. Andiyan, R. M. Putra, G. D. Rembulan, and H. Tannady, "Construction Project Evaluation Using CPM-Crashing, CPM-PERT and CCPM for Minimize Project Delays," *J Phys Conf Ser*, vol. 1933, no. 1, p. 012096, Jun. 2021, doi: 10.1088/1742-6596/1933/1/012096.