Available online at www.ijournalse.org

**Emerging Science Journal**

(ISSN: 2610-9182)

Vol. 9, No. 5, October, 2025

# Modeling and Performance Optimization for Complex Workflow in IoT

Ting Lu [1], Huiling Li [1], Qingtian Zeng [2], Hua Duan [2], Cong Liu [3*]

[1] *School of Computer Science and Technology, Shandong University of Technology, Zibo 255000, China.*

[2] *College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China.*

[3] *NOVA Information Management School, Nova University of Lisbon, 1070-312 Lisbon, Portugal.*

**Abstract**

This study addresses the growing challenge of time scheduling in Internet of Things (IoT) workflows, where efficiency in time utilization and resource profitability is increasingly constrained by uncertainty. Real-world workflows are characterized by non-deterministic activity execution and resource preparation times, yet existing research often neglects these fundamental dynamics when modeling IoT-based processes. To bridge this gap, we propose a comprehensive modeling and performance optimization framework that explicitly incorporates uncertainty. Methodologically, the framework introduces two distinct types of places to represent activities and resources, with resource properties capturing reusability and preparation processes abstracted as specialized activities. For workflow activities, timing functions are defined to model minimum and maximum execution times, enabling the computation of earliest and latest start times and the identification of critical activities driving overall workflow duration. To mitigate resource conflicts during execution, three alternative resolution strategies are developed and systematically evaluated. Results demonstrate that the proposed approach effectively identifies optimal scheduling strategies under uncertainty, enhancing both temporal efficiency and resource utilization. A workflow case study illustrates the applicability of the framework, offering methodological and practical insights for designing resilient IoT workflow scheduling systems in complex, real-world environments.

**Keywords:**

Non-Deterministic Activity Execution;
Resource Preparation Time;
Workflow Optimization;
Key Activity Identification;
Resource Scheduling Strategy;
Internet of Things.

**Article History:**

| | | | |
|---|---|---|---|
| **Received:** | 21 | June | 2025 |
| **Revised:** | 23 | September | 2025 |
| **Accepted:** | 29 | September | 2025 |
| **Published:** | 01 | October | 2025 |

## 1- Introduction

The Internet of Things (IoT) is regarded as a future network trend, providing various intelligent services in everyday life. One of the promises of the Industrial Internet of Things (IIoT) in production is to achieve better control over workflow processes. Although existing workflow management systems perform well, their efficiency can still be enhanced. In recent years, workflow management systems have demonstrated their broad applicability and feasibility in managing business process execution in everyday activities. These applications include office automation, healthcare, banking, and others. A workflow is defined as a set of well-defined business activities. Over the past decades, the modeling and analysis of workflows have garnered significant attention and achieved considerable success. However, due to the complexity and flexibility of workflow systems, there is no uniform modeling and analysis framework applicable to all types of workflows. In real-world workflow settings, both activity execution time and resource preparation time are inherently non-deterministic during the modeling phase. Unfortunately, existing studies often overlook these two characteristics.

* **CONTACT**: cliu@novaims.unl.pt

**DOI:** http://dx.doi.org/10.28991/ESJ-2025-09-05-02

Petri nets [1] have been widely employed for modeling and analyzing discrete event systems, including workflows [2, 3]. Time and resources are critical dimensions in workflow systems. While most existing models emphasize structural aspects [4, 5], some have extended the analysis to temporal factors [6]. Beyond structure and timing, resource-related requirements have also been widely examined, since real-world workflow execution typically requires resource access [3]. However, resources themselves require preparation time, which is inherently non-deterministic. This paper focuses on complex workflows characterized by non-deterministic activity execution time and resource preparation time, referred to as non-deterministic workflows. They exhibit three essential features in IoT contexts: (1) uncertain activity execution time; (2) uncertain resource preparation time; and (3) classification of resources as reusable or non-reusable. Such workflows reflect real-world execution more accurately and are critical for modeling complex processes and improving business control in IoT environments.

The remainder of this paper is structured as follows. Section 2 reviews related work. Section 3 introduces the overview of the proposed approach and formalizes the workflow with non-deterministic activity execution times and resource preparation times. Section 4 presents the NWF-net-based modeling approach for non-deterministic workflows, together with its refinement and reduction rules. Section 5 proposes time performance metrics and the approach for identifying key workflow activities. Section 6 addresses resource conflict detection approaches and resolution strategies. Section 7 analyzes the optimized execution time of the global workflow. Finally, Section 8 concludes the paper.

## 2- Related Work

Modeling, analysis, and verification of workflow processes using Petri nets have been studied for several decades [6, 7]. A significant number of Petri net classes, such as E-net, CE-net, and CTWF-net, have been introduced to address various workflow processes. Liu et al. [8] present a hierarchical multi-instance process model that effectively supports the discovery of business processes with hierarchical multi-instance structures, yielding higher-quality models. Zeng et al. [9] present a systematic framework for modeling and verification of cross-department processes. The paper introduces RM_WF_Net, an enhanced WF-net integrating resource and message dimensions, whose soundness is validated through reachability. A variety of privacy-preserving extensions of Petri net process mining methods have been developed to guarantee correctness, efficiency, and data confidentiality in collaborative business processes, and their effectiveness has been demonstrated using transportation and healthcare case scenarios [10]. Building on this foundation, recent research has explored behavioral model discovery and process management in increasingly complex environments, including Internet of Things (IoT) applications. Automatic discovery of behavioral models from execution data has enabled more efficient software understanding and verification [11], while subsequent work proposed evaluation methods to assess the correctness and completeness of software component behavioral models [12]. Hierarchical discovery techniques have been introduced to identify subprocesses from lifecycle information [13], and formal modeling approaches have been applied to multi-instance processes, such as cloud-based resource management, which are highly relevant to IoT systems [14]. In addition, execution path-oriented strategies were developed to enforce data-aware business processes [15], and similarity measurement methods have been proposed to support process comparison and optimization in dynamic data-driven environments.

Collaborative and domain-specific workflows have also attracted attention. For instance, cross-department healthcare processes were discovered from event logs to improve transparency and efficiency in critical service systems [16]. Emergency response processes, which share similar requirements with IoT-enabled smart city applications, have been studied extensively using Petri nets [17, 18]. Contributions in this area include resource conflict detection and resolution in nondeterministic workflows, refinement-based hierarchical modeling and correctness verification for cross-organization collaboration, and correctness analysis frameworks to improve reliability. Complementing these works, task pre-emption strategies based on Petri nets have been proposed to optimize scheduling and resource allocation under dynamic and uncertain conditions [19]. Together, these studies highlight how Petri-net-based modeling and analysis provide robust foundations for handling data-aware, collaborative, and resource-sensitive workflows, which are essential characteristics of IoT-driven process management.

Dealing with time constraints is crucial when designing and managing workflow systems. As a result, effective time management is essential for workflow systems in large-scale enterprises [7]. To support time management, Eder et al. [20] defined a timed workflow graph, based on which time constraints are computed at runtime. Marjanovic & Orlowska [21] first assigned a time interval to each activity, from which temporal constraints are derived. Besides time constraints, resource constraints in workflow specifications have recently garnered increased attention, as activities typically require access to resources during execution. Before an activity execution, the required resources must be properly prepared. During execution, the required resources are exclusively allocated. Once the corresponding activities are completed, these resources are released and become available for access by other activities. If a resource constraint between two activities is not properly represented, the activities may compete for same resources, resulting in a resource conflict. Therefore, ensuring resource consistency during the design phase is crucial. Regarding the IoTs, Kotb et al. [22] proposed a workflow-net-based framework to facilitate collaboration among fog computing devices in an IoT service delivery system. It is demonstrated that the collaborative process increases the number of tasks achieved and ensures their timely completion. In addition, Kozma et al. [23] provided a concept for IIoT Systems based on workflow technology to improve the understanding and control of workflows, as well as to enhancing transparency.

By considering both resource and time constraints, Wang & Zeng [2] proposed a workflow model constrained by resource availability and non-deterministic duration. It paid much attention to the impacts of resource and non-determined duration on the execution of the workflow. However, they did not give sufficient attention to the preparation time and classification for each kind of resource. To address this problem, we address one kind of workflow with both non-deterministic activity execution time and resource preparation time, and resources are divided into reusable and non-reusable ones. Our work can be regarded as an extension of this work by exploring (1) resource preparation and classification; and (2) resource conflict detection and resolution and global time performance optimization.

## 3- An Approach Overview and Formal Definition of Non-Deterministic Workflow

Figure 1 outlines the sequential process of the proposed approach. The approach begins with modeling non-deterministic workflow activities, followed by constructing the IoT-oriented workflow model, termed NWF-net. The complex NWF-net is then refined and reduced to simplify analysis. Next, time performance metrics and an approach for identifying key workflow activities are proposed. Considering resource conflicts in IoT settings, detection approaches and three resolution strategies are introduced. Finally, these strategies are applied to achieve workflow time optimization.

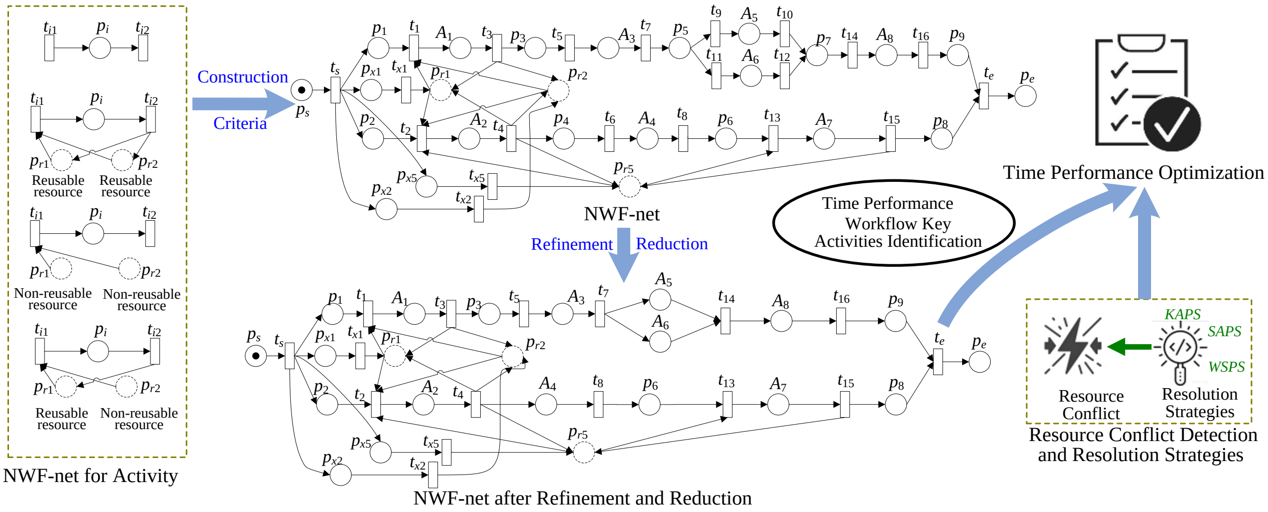



**Figure 1. An Approach Overview**

In this section, we formalize the specification of workflows characterized by non-deterministic activity execution and resource preparation times, followed by the discussion of a representative example. To ensure self-completeness, the following assumptions are introduced: (1) Resources require preparation time before use, and this process is abstracted as a preparation activity in the model; (2) Resources are classified as reusable or non-reusable; (3) The preparation of resources begins with workflow initiation, where reusable resources, once prepared, may be repeatedly utilized provided they remain available, while non-reusable resources must be re-prepared prior to each reuse; and (4) Activities demand resources during execution. For reusable resources, access is exclusive, and other activities must wait until they are released. For non-reusable resources, they are re-prepared immediately after they are occupied, becoming accessible again once preparation is completed.

An example of a non-deterministic workflow is shown in Figure 2. An illustrative non-deterministic workflow is depicted in Figure 2. Table 1 summarizes activity time constraints, connection relationships, and resource usage details, while Table 2 presents the non-deterministic preparation times and classifications of the corresponding resources.

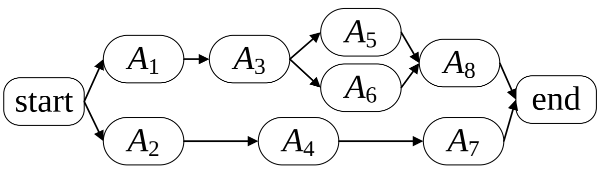



**Figure 2. A non-deterministic workflow example**

**Table 1.** Activity information

| Activity Name | Minimum Time | Maximum Time | Pre-Activity Set | Resource Set |
|---|---|---|---|---|
| $A_1$ | 10 | 12 | Null | $\{r_1, r_2\}$ |
| $A_2$ | 3 | 7 | Null | $\{r_1, r_2, r_5\}$ |
| $A_3$ | 15 | 20 | $\{A_1\}$ | $\{r_3, r_4\}$ |
| $A_4$ | 10 | 15 | $\{A_2\}$ | Null |
| $A_5$ | 5 | 15 | $\{A_3\}$ | Null |
| $A_6$ | 10 | 20 | $\{A_3\}$ | Null |
| $A_7$ | 10 | 18 | $\{A_4\}$ | $\{r_5\}$ |
| $A_8$ | 1 | 2 | $\{A_5, A_6\}$ | Null |

**Table 2.** Resource preparation time and classification

| Resource Name | Minimum Time | Maximum Time | Classification |
|---|---|---|---|
| $r_1$ | 4 | 7 | reusable |
| $r_2$ | 5 | 6 | reusable |
| $r_3$ | 16 | 18 | non-reusable |
| $r_4$ | 8 | 21 | non-reusable |
| $r_5$ | 7 | 10 | reusable |

Based on Tables 1 and 2, the following observation can be made: (1) The workflow involves eight activities, denoted as *Activity* $=\{A_i| 1\leq i\leq 8\}$. According to Table 1, activity $A_1$ is a pre-activity of $A_3$, which indicates that $A_3$ cannot start before $A_1$; and resources required are denoted by *Resource* $= \{r_j| 1\leq j\leq 5\}$. According to Table 1, $A_1$ and $A_2$ require resources $r_1$ and $r_2$, which indicates that they are shared. Assume that $A_1$ executes first, during which $r_1$ and $r_2$ are locked. $A_2$ must wait until $A_1$ finished and $r_1$ and $r_2$ are released. Since $r_1$ and r2 are reusable resources, they do not need to be re-prepared before $A_2$ can reuse them. Otherwise, $A_2$ does not need to wait for $A_1$ to release the resources. $A_2$ can use them after they have been properly re-prepared, and (2) there are two timing factors for each activity and resource. For each activity, the former represents the minimum execution time, and the latter represents the maximum one. For each resource, the one-timing factor represents the minimum preparation time of the resource, and the other represents the maximum one.

## 4- Modeling Workflow with NFW-Net

The modeling approaches for non-deterministic workflows using NWF-nets are given below.

### *4-1- NWF-net*

This section proposes an extension of Petri nets, termed NWF-net, to model workflows with non-deterministic activity execution and resource preparation times. For self-completeness, the basic concepts of Petri nets [1] are given.

**Definition 1.** A *Petri net* is a 4-tuple $\Sigma = (P, T, F, M_0)$, where (1) $P$ is a set of places, $T$ is a set of transitions, $P\cap T=\varnothing$, and $P\cup T\neq\varnothing$; (2) $F\subseteq(P\times T)\cup(T\times P)$ denotes the flow relation; and (3) $M_0$: $P\rightarrow Z$ is the initial marking.

For any $x\in P\cup T$, ${}^{\bullet}x = \{y| y\in P\cup T\wedge(y, x)\in F\}$ represents its pre-set, and $x^{\bullet} = \{y| y\in P\cup T \wedge(x, y) \in F\}$ represents its post-set. $M_0$ denotes the initial marking and $R(M_0)$ denotes the set of reachable markings of $\Sigma$. For any $p\in P$ is marked by $M$ if $M(p)>0$. A transition $t\in T$ is enabled under $M$, if for any $p\in{}^{\bullet}t$, we have $M(p)>0$, denoted as $M[t>$. The firing of $t$ results in a new marking $M'$, denoted as $M[t>M'$, such that $M'(p)=M(p)-1$ if $\forall p\in{}^{\bullet}t \setminus t^{\bullet}$, $M'(p)=M(p)+1$ if $\forall p\in t^{\bullet} \setminus {}^{\bullet}t$, and otherwise $M'(p)=M(p)$. Let $R$ be the set of non-negative real number.

**Definition 2.** $\Sigma_{NWF} = (P, T, F, M_0, l\ \alpha, \beta)$ is an *NWF-net* such that:

(1) $(P, T, F, M_0)$ is a Petri net;

(2) $P=P_A\cup P_R$, $P_A\cap P_R=\varnothing$ where $P_A$ is the activity place set, and $P_R$ is the resource place set respectively;

(3) $\forall p\in P_A$, $P_{Ri}\subseteq P_R$, execution of activity $p$ requires resources $P_{Ri}$ if $P_{Ri}\subseteq{}^{\bullet}({}^{\bullet}p)$;

(4) $l$: $P_R\rightarrow\{reusable, non\text{-}reusable\}$, $\forall p_r\in P_R$, $l(p_r)=reusable$ represents that resource $p_r$ is reusable, and $l(p_r) = non\text{-}reusable$ denotes that $p_r$ is non-reusable;

(5) $\alpha$: $P \rightarrow R$. $\forall p\in P$, $\alpha(p)\geq 0$ denotes the minimum time to execute an activity $p$ or to prepare resource $p$;

(6) $\beta$: $P \rightarrow R$. $\forall p\in P$, $\beta(p)\geq 0$ denotes the maximum time to execute activity $p$ or to prepare resource $p$ satisfying $\alpha(p)\leq\beta(p)$; and (7) $\forall p\in P$, $M_0(p)=1$ if ${}^{\bullet}p=\varnothing$, and $M_0(p)=0$ otherwise.

For each activity place, two timing factors are defined: the minimum and maximum execution time. Similarly, for each resource place, two timing factors are defined: the minimum and maximum preparation time.

### *4-2- NWF-net for Single Activity*

The modeling of a single activity within the NWF-net is introduced first. A single activity is represented by one place and two associated transitions, as illustrated in Figure 3. Specifically, $p_i$ denotes the activity place, while $t_{i1}$ and $t_{i2}$ correspond to its start and completion transitions, respectively. An activity can be denoted as $[t_{i1}, p_i, t_{i2}]$. When resource requirements are involved, the activity is modelled using the two transitions together with an additional set of resource places. Each resource is represented by a dedicated place, highlighted by a dashed outline. The initiation of an activity then requires the corresponding resources as input.

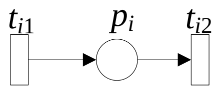



**Figure 3. NWF-net for a single activity without resources**

When activities require multiple types of resources, the corresponding NWF-net models are constructed, as illustrated in Figure 4. Specifically, when a single activity requires two reusable resources, the corresponding modeling process is illustrated in Figure 4-a, where $p_{r1}$ and $p_{r2}$ become available for reuse once activity $p_i$ finishes. When both resources are non-reusable, the modeling process is depicted in Figure 4-b, where $p_{r1}$ and $p_{r2}$ must be re-prepared and cannot be reused once activity $p_i$ finishes. When the activity involves one reusable and one non-reusable resource, the modeling process is presented in Figure 4-c. In this case, $p_{r1}$ is a reusable resource and can be reused after activity $p_i$ finishes, whereas $p_{r2}$ is a non-reusable resource and cannot be reused after activity $p_i$ finishes.

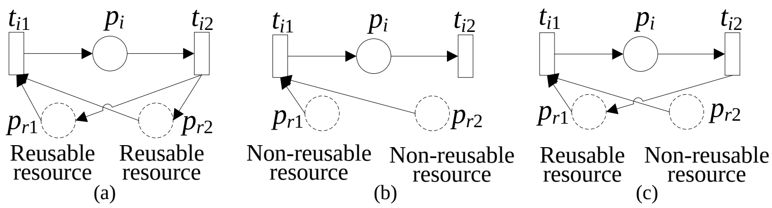



**Figure 4. NWF-net for multiple types of resources**

### *4-3- NWF-net for Global Workflow*

In this section, four construction criteria of the NWF-net are defined, followed by a detailed introduction to the construction of NWF-nets for non-deterministic workflows.

#### *4-3-1- Criteria One: NWF-net for Two Sequence Activities*

If activity $p_i$ is one of the per-activities of activity $p_j$, a place $p_{ij}$ is inserted between transitions $t$ $t_{i2}$ and $t_{j1}$ to establish the connection, as shown in Figure 5. This auxiliary place, termed a virtual place, serves only as a structural element without semantic meaning in the workflow.

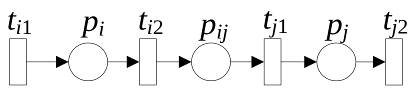



**Figure 5. NWF-net for two sequence activities**

#### *4-3-2- Criteria Two: NWF-net Models for Activities Sharing Heterogeneous Resources*

If a set of reusable resources $\{p_{r1}, p_{r2}, \ldots, p_{rk}\}$ is shared by activities $p_i$ and $p_j$, then $\{t_{i2}, t_{j2}\}\subseteq {}^\bullet p_{rj}$ and $\{t_{i1}, t_{j1}\}\subseteq p_{rj}{}^\bullet$ are incorporated into the NWF-net, as illustrated in Figure 6(a). If a set of non-reusable resources $\{p_{r1}, p_{r2}, \ldots, p_{rk}\}$ is shared by activities $p_i$ and $p_j$, then only $\{t_{i1}, t_{j1}\}\subseteq p_{rj}{}^\bullet$ is incorporated into the NWF-net, as illustrated in Figure 6(b). If the resource set composes both non-reusable and reusable resources, the NWF-net is shown in Figure 6(c), where $p_{r1}$ is a reusable resource and $p_{r2}$ is a non-reusable resource.

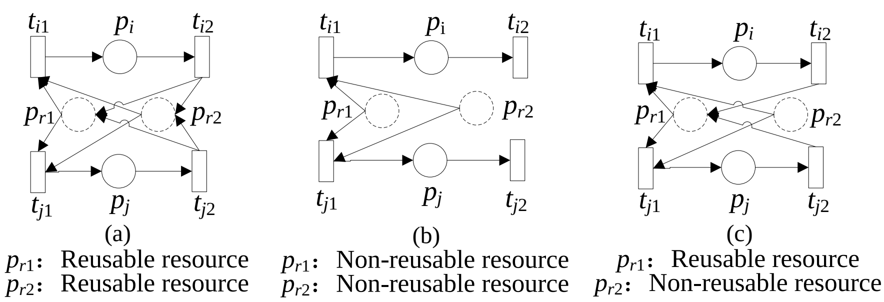
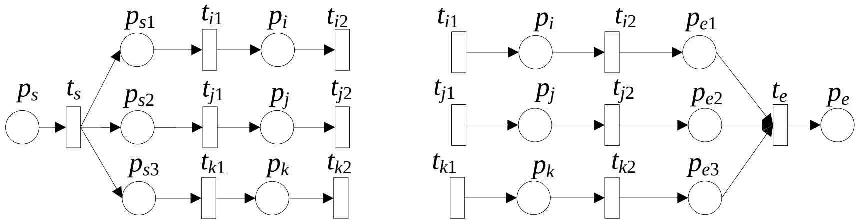



**Figure 6. NWF-net models for activities sharing heterogeneous resources**

### 4-3-3- Criteria Three: NWF-net for Activities without Pre-activities (Post-activities)

A set of virtual activities is added to connect activities without pre-activities. Their start transitions are merged into a single transition $t_s$. A source place $p_s$ is then added, satisfying ${}^\bullet t_s=\{p_s\}$, $p_s{}^\bullet=\{t_s\}$, and ${}^\bullet p_s=\varnothing$. The NWF-net for three activities without pre-activities is shown in Figure 7-a. Similarly, a set of virtual activities is added to connect activities without post-activities. Their termination transitions are merged into one transition $t_e$. A sink place $p_e$ is then added, satisfying ${}^\bullet p_e=\{t_e\}$, $t_e{}^\bullet=\{t_s\}$, and $p_e{}^\bullet=\varnothing$. The NWF-net for three activities without post-activities is shown in Figure 7-b.



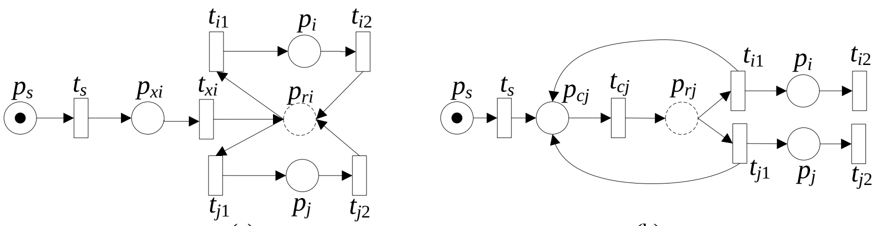

(a) NWF-net for activities without pre-activities

(b) NWF-net for activities without post-activities

**Figure 7.** **NWF-net for activities without pre-activities (post-activities)**

### 4-3-4- Criteria Four: NWF-net of Reusable Resource (non-Reusable Resource)

Since resources require time to be properly prepared, this preparation begins as soon as the workflow starts. For each reusable resource $p_{ri}\in P_R$, a preparation place $p_{xi}$ and a transition $t_{xi}$ are added, such that: ${}^\bullet p_{xi}=\{t_s\}$, $p_{xi}{}^\bullet=\{t_{xi}\}$, ${}^\bullet t_{xi}=\{p_{xi}\}$, $t_{xi}{}^\bullet=\{p_{ri}\}$, $\alpha(p_{xi})=\alpha(p_{ri})$, and $\beta(p_{xi})=\beta(p_{ri})$. In this way, the preparation time of $p_{ri}$ is abstracted as its corresponding preparation activity $p_{xi}$. Because $p_{ri}$ is reusable, once released it can be utilized by other activities without undergoing re-preparation. The resource model for reusable resources is illustrated in Figure 8-a.




(a) NWF-net for reusable resource

(b) NWF-net for non-reusable resource

**Figure 8.** **NWF-net of reusable resource (non-reusable resource)**

For the non-reusable resource $p_{rj}\in P_R$, if it is shared by $k$ activities, a preparation place $p_{cj}$ and a transition $t_{cj}$ are added, such that: ${}^\bullet p_{cj}=\{t_s\}$, and $p_{cj}{}^\bullet=\{t_{cj}\}$. In this way, the preparation time of $p_{rj}$ is abstracted into its corresponding preparation activity $p_{cj}$, which will be repeated $k$ times. Since $p_{rj}$ is non-reusable, other activities have to re-prepare *it* before reuse. The resource model for non-reusable resources is illustrated in Figure 8-b.

Based on the above modeling criteria, a non-deterministic workflow can be systematically transformed into its corresponding NWF-net. For the case presented in Section 3, the constructed NWF-net is shown in Figure 9. However, the initial model may contain redundant places and transitions, which complicate further analysis. To address this issue, the refinement operation proposed by Wang & Zeng [2] is applied, ensuring that all activities and their connection relations remain unchanged. The redundant places, regarded as virtual activities, carry no semantic meaning and are introduced solely for structural control. Taking the non-deterministic workflow as an example, its NWF-net after refinement is shown in Figure 10.

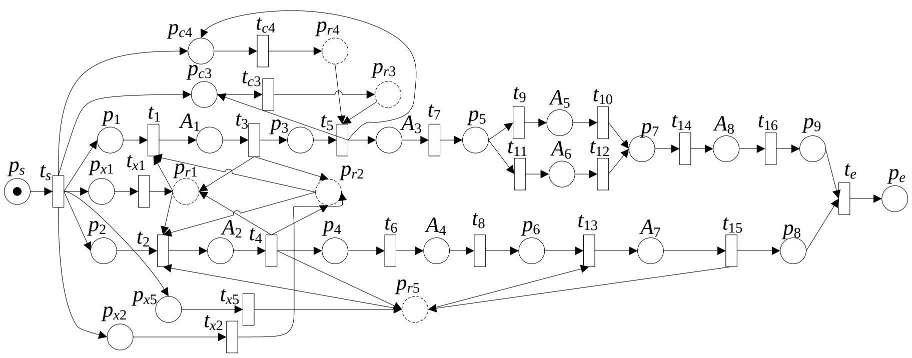



**Figure 9.** **NWF-net for non-deterministic workflow in Section 3**

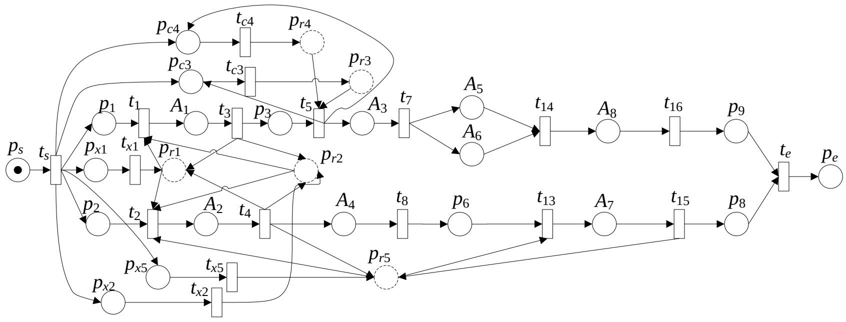
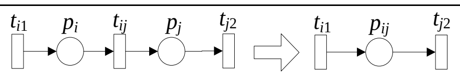
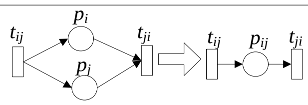
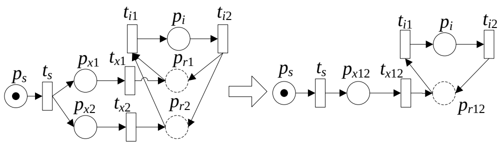
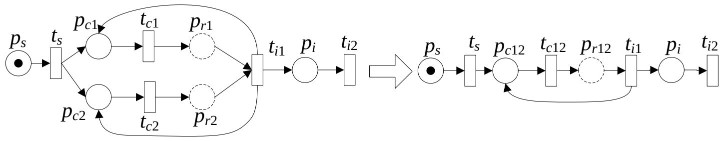
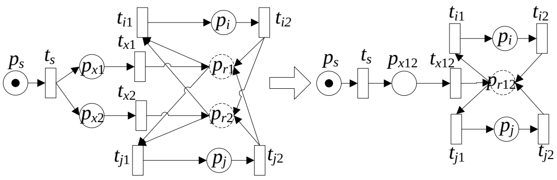



**Figure 10.** NWF-net of Figure 9 after refinement

### *4-4- Invariant Reduction of NWF-net*

When a workflow is highly complex, the corresponding NWF-net after refinement may also become excessively large. Following Wang & Zeng [2], a set of reduction rules is defined for the NWF-net, ensuring that the structure, activity execution times, and resource properties are preserved. In this way, the reduction process simplifies subsequent analysis of the model. For brevity, only the definitions of the reduction rules are presented in Table 3, without delving into the proof

**Table 3.** Reduction rules for NWF-net

| Rule | Reduction rules |
|---|---|
| Rule 1 |  |
| Rule 2 |  |
| Rule 3 |  |
| Rule 4 |  |
| Rule 5 |  |

Rules 1-2 simplify sequential and parallel structures without resource constraints. Rule 3 allows for the reduction of reusable resources and their corresponding preparation activities. Rule 4 facilitates the reduction of non-reusable resources and their associated preparation activities. Rule 5 is designed to reduce shared reusable resources and their corresponding preparation activities. The reduced NWF-net significantly decreases the number of places while preserving both temporal and resource constraints. Taking the NWF-net in Figure 10 as an example, the reduced model is shown in Figure 11, where: (1) $p_{r12}$ is obtained by merging $p_{r1}$ and $p_{r2}$ via Rule 5; (2) $A_{568}$ is obtained by merging $A_5$, $A_6$, and $A_8$ through Rules 1-2; (3) $A_{47}$ is derived from $A_4$ and $A_7$ using Rule 1; and (4) $p_{r34}$ is obtained by merging $p_{r34}$ with Rule 4.

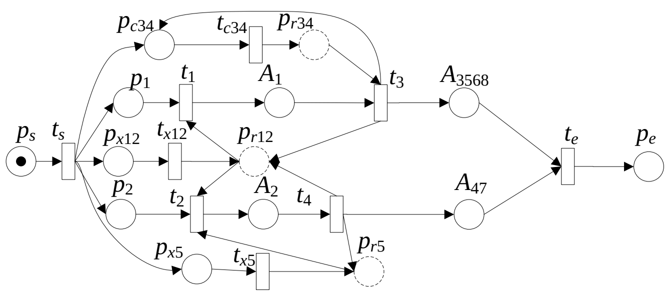



**Figure 11.** NWF-net in Figure 10 after reduction

## 5- Time Performance Evaluation and Workflow Key Activities

This section details how to evaluate the time performance of non-deterministic activities and identify the key workflow activities based on the constructed NWF-net.

### 5-1- Time Performance Evaluation

The reduced NWF-net is used to compute the earliest and latest start times of each business activity when resource factors are not taken into account. Specifically, the earliest time to start activity $p$ if activities are completed in the minimum time, can be obtained as follows:

$$T_{e1}(p) = \begin{cases} 0 & p = p_s \\ max\{T_{e1}(p') + \alpha(p') | p' \in {}^\bullet({}^\bullet p)\} & otherwise \end{cases} \quad (1)$$

Without considering resources, the earliest time to start activity $p$ if activities are completed in the maximum time, can be obtained as follows:

$$T_{e2}(p) = \begin{cases} 0 & p = p_s \\ max\{T_{e2}(p') + \beta(p') | p' \in {}^\bullet({}^\bullet p)\} & otherwise \end{cases} \quad (2)$$

Let $T_{E1}$=$T_{e1}(p_e)$ and $T_{E2}$=$T_{e2}(p_e)$, where $p_e$ denotes the sink place of the workflow. In the absence of resource constraints, $T_{E1}$ and $T_{E2}$ correspond to the minimum and maximum completion times of the workflow. To ensure completion within $T_{E1}$, the latest time to start activity $p$, denoted as $T_{l1}(p)$, is computed as follows:

$$T_{l1}(p) = \begin{cases} T_{E1}(p) & p = p_e \\ min\{T_{l1}(p') - \alpha(p') | p' \in (p^\bullet)^\bullet\} & otherwise \end{cases} \quad (3)$$

To ensure the workflow completes within $T_{E2}$, the latest start time to start activity $p$, denoted $T_{l2}(p)$, is calculated as follows:

$$T_{l2}(p) = \begin{cases} T_{E2}(p) & p = p_e \\ min\{T_{l2}(p') - \beta(p') | p' \in (p^\bullet)^\bullet\} & otherwise \end{cases} \quad (4)$$

### 5-2- Workflows Key Activities Identification

**Definition 3.** In an NWF-net $\Sigma_{NWF}$ = ($P$, $T$, $F$, $M_0$, $l$ $\alpha$, $\beta$), a direct path from $p_s$ to $p_e$ on which for any place $p \in P_A$ satisfying $T_{e1}(p)$= $T_{l1}(p)$ or $T_{e2}(p)$= $T_{l2}(p)$, is defined as the key activity path that influences the time to finish the workflow in $T_{E1}$ or $T_{E2}$. Activities on the key path are called key activities.

If the execution time of a key activity is prolonged, the completion of the entire workflow will be delayed. Therefore, the execution of key activities must be preserved to ensure workflow efficiency. Following Wang & Zeng [2], we establish the existence property of a key path for any given NWF-net.

**Theorem 1.** Let $\Sigma_{NWF}$ = ($P$, $T$, $F$, $M_0$, $l$ $\alpha$, $\beta$) be an NWF-net, such that: (1) there exists a directed path from $p_s$ to $p_e$ such that $T_{e1}(p)$= $T_{l1}(p)$ holds for all activities; and (2) there exists directed path from $p_s$ to $p_e$ such that $T_{e2}(p)$= $T_{l2}(p)$ holds for all activities.

**Proof.** Given $T_{e1}(p)$ = $T_{l1}(p)$ = $T_{E1}$, where $p_e$ is the sink place of the workflow. From Equations 1 and 3, there exists $p' \in {}^\bullet({}^\bullet p_e)$ such that $T_{e1}(p')$ = $T_{l1}(p')$. Similarly, there exists such that $p'' \in {}^\bullet({}^\bullet p')$ satisfying $T_{e1}(p'')$=$T_{l1}(p'')$. By iteratively applying this reasoning, we obtain $T_{e1}(p_s)$ = $T_{l1}(p_s)$. Therefore, a directed path from $p_s$ to $p_e$ exists such that $T_{e1}(p)$ = $T_{l1}(p)$ holds for all activities.

The proof for the case $T_{e2}(p)$ = $T_{l2}(p)$ is analogous and is omitted here due to space limitations.

Taking the NWF-net in Figure 11 as an example, $T_{e1}(p)$, $T_{e2}(p)$, $T_{l1}(p)$, and $T_{l2}(p)$ are computed and shown in Table 4. According to Table 4, the key activity path $p_s \rightarrow p_{c34} \rightarrow p_{r34} \rightarrow A_{3568} \rightarrow p_e$ is identified, as illustrated in Figure 12. The key activities are critical because they directly determine the global workflow's execution time. For example, $A_{3568}$ is a key activity; the delay of $A_{3568}$ will prolong the workflow completion. By contrast, $A_{47}$ is not a key activity, and its delay does not necessarily affect the overall finishing time.

**Table 4. $T_{e1}(p)$, $T_{e2}(p)$, $T_{l1}(p)$, and $T_{l2}(p)$ of Figure 11**

| Activity | $p_s$ | $p_{x12}$ | $p_{c34}$ | $p_{x5}$ | $p_1$ | $p_2$ | $p_{r12}$ | $p_{r34}$ | $p_{r5}$ | $A_1$ | $A_2$ | $A_{47}$ | $A_{3568}$ | $p_e$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $T_{e1}(p)$ | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 16 | 7 | 5 | 7 | 10 | 16 | 42 |
| $T_{e2}(p)$ | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 21 | 10 | 7 | 10 | 17 | 21 | 63 |
| $T_{l1}(p)$ | 0 | 1 | 0 | 12 | 6 | 19 | 6 | 16 | 19 | 6 | 19 | 22 | 16 | 42 |
| $T_{l2}(p)$ | 0 | 2 | 0 | 14 | 9 | 24 | 9 | 21 | 24 | 9 | 24 | 31 | 21 | 63 |

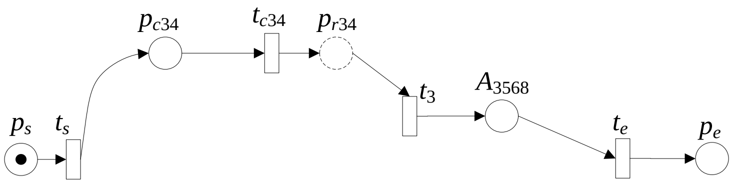



**Figure 12.** Key Activity Path of Figure 11

# 6- Resource Conflict Detection and Resolution Strategies

The optimal start and completion times of the workflow are determined. Nevertheless, resource conflicts are unavoidable during execution due to inherent resource constraints. To ensure efficient completion with improved time performance and utilization, verification approaches and resolution strategies for resource conflicts are introduced.

## 6-1- Resource Conflicts

We assume that each resource requires a certain preparation time, which may affect or delay the execution of activities. Resources are further categorized into two types: reusable and non-reusable. A reusable resource needs to be prepared only once and can be repeatedly utilized throughout the workflow once released. In contrast, a non-reusable resource must be re-prepared before every subsequent use. For example, if a reusable resource set $P_{r1}$ is shared by activities $p_1$ and $p_2$, with $p_1$ executing first, then $p_2$ must wait until $p_1$ completes and the resources are released. Conversely, if a non-reusable resource set $P_{r1}$ is shared by activities $p_1$ and $p_2$, with $p_1$ executing first, then $p_2$ has to wait until the non-reusable resources are re-prepare. In general, if the resource set $P_{r1}$ is shared by activities $p_1$ and $p_2$, it is denoted as $p_1 \otimes p_2 = P_{r1}$. As shown in Figure 6, resource sets $P_{r1}$ and $P_{r2}$ are both shared by $p_i$ and $p_j$, so the set of shared resources is $p_i \otimes p_i = \{P_{r1}, P_{r2}\}$.

**Definition 4.** Let $\Sigma_{NWF} = (P, T, F, M_0, l\ \alpha, \beta)$ be an NWF-net. For any $p_i, p_j \in P_A$ ($p_i \neq p_j$), $p_i$ and $p_j$ are in *resource conflict* over $P_{r1}$, denoted as $p_i © p_j$, if (1) $p_1 \otimes p_2 = P_{r1}$; and (2) $[T_{start}(p_i), T_{end}(p_i)]$ and $[T_{start}(p_j), T_{end}(p_j)]$ are overlapping if $P_{r1}$ is a reusable resource set; and $[T_{start}(p_i), \max\{T_{end}(p_i), T_{start}(p_i) + \beta(P_{r1})\}]$ and $[T_{start}(p_j), \max\{T_{end}(p_j), T_{start}(p_j) + \beta(P_{r1})\}]$ are overlapping if $P_{r1}$is a non-reusable resource set. $T_{start}(p)$ and $T_{end}(p)$ is the real start and complete time of activity $p$.

Resource conflicts between two activities can be identified using two criteria: (1) the activities must share at least one resource; and (2) if the resource is reusable, their execution intervals overlap; if it is non-reusable, the overlap occurs between the execution time of one activity and the re-preparation period of the resource.

## 6-2- Resolution Strategies for Potential Resource Conflicts

To address potential resource conflicts, three resolution strategies are introduced following Van Der Aalst [4]: *key-activity priority strategy* (*KAPS*), *waiting-short priority strategy* (*WSPS*), and *start-early priority strategy* (*SAPS*). Let $W(p_i, p_j)$ represent the waiting time of $p_j$ for the shared resources occupied by $p_i$. If the two activities do not share any resources, then $W(p_i, p_j) = W(p_j, p_i) = 0$.

***Key-Activity Priority Strategy* (*KAPS*):** Given $p_i © p_j$. If $p_i$ is a key activity but $p_j$ is not. The priority for the shared resources of $p_i$ is higher than that of $p_j$, i.e. $W(p_i, p_j) = 0$ and $W(p_j, p_i) \geq 0$.

*KAPS* applies only to conflicts between a key and a non-key activity; when both are of the same type, two other strategies-*SAPS* and *WSPS*-are employed. *SAPS* ensures that an activity with an earlier start time has a higher priority regarding shared resources, while *WSPS* guarantees that the waiting time for activities in conflicts is minimized.

***Start-early Priority Strategy* (*SAPS*):** Given $p_i © p_j$. If the start time of $p_i$ is earlier than that of $p_j$. The priority of $p_i$ is higher than that of $p_j$, i.e. $W(p_i, p_j) = 0$ and $W(p_j, p_i) \geq 0$.

***Waiting-short Priority Strategy* (*WSPS*):** Given $p_i © p_j$, when both are non-key activities, priority is assigned by comparing waiting times: if $W(p_i, p_j) \geq W(p_j, p_i)$, then $W(p_i, p_j) = 0$; otherwise, $W(p_j, p_i) = 0$.

By applying the three conflict resolution strategies described above, resource contention among activities can be effectively mitigated. The subsequent section presents detailed procedures for employing these strategies to resolve conflicts and optimize time performance.

# 7- Time Performance Optimization

Since each strategy may yield different workflow execution performances, the global workflow's optimized execution time is analyzed by comparing all resolution strategies.

## 7-1- Earliest Time to Start Activity

An activity can start only after all its pre-activities are completed and the necessary resources are prepared-that is, reusable resources are released and non-reusable ones have finished their preparation. In workflow execution, even if every activity runs at its minimum duration, $T_{e1}(p)$ may not represent the true earliest start time of $p$, since preceding activities might be delayed by resource conflict. The actual earliest start time of $p$, denoted $E_1(p)$, is given in Equation 5, where $W_1(p, p'')$ denotes the waiting time of $p$ for resources.

$$E_1(p) = \begin{cases} 0 & p = p_s \\ max\{E_1(p') + \alpha(p') + W_1(p, p'') \mid p' \in {}^\bullet({}^\bullet p), p \otimes p'' = P_{r1}\} & otherwise \end{cases} \quad (5)$$

$E_1(p_e)$ denotes the minimum completion time of the global workflow. As discussed in Section 6.2, different resolution strategies for resource conflicts may lead to varying execution durations. To ensure the workflow finishes as early as possible, the procedure for selecting the optimal resolution strategy is as follows:

- **Step 1**: Detecting resource conflicts in a workflow using resource consistency approaches.
- **Step 2**: Compute the earliest start times of all activities and obtain $E_1(p_e)$, which reflects the minimal workflow completion time. As $W_1(p, p'')$ for conflicting activities cannot be determined at this phase, the exact value of $E_1(p_e)$ is unavailable. Hence, symbolic expressions are used to represent both $W_1(p, p'')$ and $E_1(p_e)$. For instance, if $p_1$ and $p_2$ conflict, $wait_{12}$ denotes the waiting time of $p_1$ for $p_2$, and $wait_{21}$ that of $p_2$ for $p_1$.
- **Step 3**: Apply the three conflict resolution strategies to optimize the actual value of $E_1(p_e)$ under each strategy.

**(3.1) *KAPS***: When $p$ is a key activity and $p''$ is not, resources are assigned to $p$ with priority, meaning $p''$ must wait. In this case, $W_1(p, p'') = 0$, and the corresponding expression of $W_1(p, p'')$ is discussed as follows.

If $P_{r1}$ is reusable resource, then:

$$W_1(p'', p) = \begin{cases} 0 & T_{sp1}(p) + \alpha(p) \le T_{sp1}(p'') \\ T_{sp1}(p) + \alpha(p) - T_{sp1}(p'')\} & otherwise \end{cases} \quad (6)$$

If $P_{r1}$ is non-reusable resource, then:

$$W_1(p'', p) = \begin{cases} 0 & T_{sp1}(p) + \alpha(P_{r1}) \le T_{sp1}(p'') \\ T_{sp1}(p) + \alpha(P_{r1}) - T_{sp1}(p'')\} & otherwise \end{cases} \quad (7)$$

where $\alpha(p)$ shows the execution time of activity $p$ and $\alpha(P_{r1})$ shows the preparation time of resource $P_{r1}$.

**(3.2) *WSPS* and *SAPS***: When $p$ and $p''$ are of the same type (either key or non-key activities), their initial priorities are identical. To accelerate workflow completion, waiting times among activities should be minimized. Under this condition, two strategies are applied: *SAPS* and *WSPS*.

The *WSPS* is described as follows:

If $W_1(p'', p) \le W_1(p, p'')=0$, then $W_1(p, p'')=0$; otherwise, $W_1(p'', p) = 0$.

(a) If $W_1(p, p'') = 0$, then $W_1(p'', p)$ is calculated the same as that in (3.1).

(b) If $W_1(p'', p) = 0$, then $W_1(p, p'')$ is calculated the same as that in (a).

The *SAPS* is described as follows:

If $T_{start}(p) \le T_{start}(p'')$, then $W_1(p, p'')=0$; otherwise, $W_1(p'', p) = 0$.

(a) If $W_1(p, p'') = 0$, then $W_1(p'', p)$ is calculated the same as that in (3.1).

(b) If $W_1(p'', p) = 0$, then $W_1(p, p'')$ is calculated the same as that in (a).

- **Step 4**: Choose the shortest execution time and the corresponding resource resolution strategies for different conflicts in the workflow.

This approach not only eliminates resource conflict but also improves the overall time performance of the global workflow. Applying these strategies, the earliest time for each activity is computed. To illustrate the process in detail, the NWF-net in Figure 11 is used as an example.

**4.1**: According to Section 6.2, activities $A_1$ and $A_2$ are identified as conflicting.

**4.2**: Let $wait_{12}$ denote the waiting time of $A_1$ for $A_2$, and $wait_{21}$ denote the waiting time of $A_2$ for $A_1$. In this approach, the shortest completion time of the workflow is expressed as: $E_1(p_e) = max\{wait_{12}+41, wait_{21}+30, 42\}$.

**4.3**: Since $A_1$ and $A_2$ are not key activities, both *SAPS* and *WSPS* are applied to minimize $E_1(p_e)$.

*WSPS*: $wait_{12}=5$ and $wait_{21}=0$. Thus, $E_1(p_e)=max\{wait_{12}+41, wait_{21}+30, 42\}=46$.

*SAPS*: $wait_{12}=0$ and $wait_{21}=8$. Thus, $E_1(p_e)=max\{wait_{12}+41, wait_{21}+30, 42\}=42$.

**4.4**: To minimize the execution time, the *SAPS* is selected to resolve resource conflicts between $A_1$ and $A_2$.

The minimized execution time in Section 7 coincides with the ideal value, demonstrating that the proposed conflict resolution strategies ensure effective optimal execution of the workflow. In certain special cases, the globally optimal execution can also be achieved. Similarly, if every activity is completed at its maximum duration, the earliest start time of activity $p$, denoted as $E_2(p)$, is expressed as:

$$E_2(p) = \begin{cases} 0 & p = p_s \\ max\{ E_2(p') + \beta(p') + W_2(p, p'') | p' \in {}^\bullet({}^\bullet p), p \otimes p'' = P_{r1}\} & otherwise \end{cases} \tag{8}$$

where $W_2(p, p'')$ is the waiting time of activity $p$ for resources occupied by $p''$. When $p \otimes p'' = P_{r1}$, the computation of $W_2(p, p'')$ follows the same approach as $W_1(p, p'')$. $E_1(p)$ and $E_2(p)$ are the earliest time to start $p$ if all activities before $p$ are completed in $\alpha(p)$ and $\beta(p)$. Based on the constructed NWF-net, $E_1$ and $E_2$ can be obtained from Equations 5 and 8 along each directed path from $p_s$ to $p_e$.

### 7-2- Latest Time to Start Activity

Let $TE_1 = E_1(p_e)$ and $TE_2 = E_2(p_e)$, where $p_e$ is the sink place of the workflow. $E_1(p)$ and $E_2(p)$ are the time to complete the workflow if each activity is completed in its minimum and maximum execution time. In the non-deterministic workflow, the latest time to start activity is to ensure its execution can be completed in time $TE_1$ or $TE_2$, where $TE_1 = E_1(p_e)$ and $TE_2 = E_2(p_e)$. To ensure the global non-deterministic workflow to be completed in $TE_1$, the latest time to start $p$, denoted as $L_1(p)$, is given in Equation 9.

$$L_1(p) = \begin{cases} E_1(p) & p = p_e \\ min\{ L_1(p') - \alpha(p) | p' \in (p^\bullet)^\bullet \vee (p \otimes p'' = P_{r1} \wedge W_1(p', p) > 0\} & otherwise \end{cases} \tag{9}$$

To ensure the global non-deterministic workflow to be completed in $TE_2$, the latest time to start $p$, denoted as $L_2(p)$, is given in Equation 10.

$$L_2(p) = \begin{cases} E_2(p) & p = p_e \\ min\{ L_2(p') - \beta(p) | p' \in (p^\bullet)^\bullet \vee (p \otimes p'' = P_{r1} \wedge W_2(p', p) > 0\} & otherwise \end{cases} \tag{10}$$

By calculating $L_1(p)$ and $L_2(p)$ based on Equations 9-10, the feasible latest start times of all activities can be determined. If an activity cannot be started at that time, it will influence the initialization of follow-up activities, and the whole workflow cannot be finished in time $TE_1$ or $TE_2$. Considering resource conflict resolutions, the optimized $T_{e1}(p)$, $T_{e2}(p)$, $T_{l1}(p)$, and $T_{l2}(p)$ for the NWF-net in Figure 11 are computed and shown in Table 5. The comparison with Table 4 shows that the actual execution times align with the ideal values, confirming that the proposed strategies ensure both high temporal efficiency and effective resource utilization.

**Table 5.** $T_{e1}(p)$, $T_{e2}(p)$, $T_{l1}(p)$ and $T_{l2}(p)$ of Figure 11

| Activity | $p_s$ | $p_{x12}$ | $p_{c34}$ | $p_{x5}$ | $p_1$ | $p_2$ | $p_{r12}$ | $p_{r34}$ | $p_{r5}$ | $A_1$ | $A_2$ | $A_{47}$ | $A_{3568}$ | $p_e$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $W_1(p, p'')$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 0 |
| $E_1(p)$ | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 16 | 7 | 5 | 15 | 11 | 16 | 42 |
| $W_2(p, p'')$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9 | 0 | 0 | 0 |
| $E_2(p)$ | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 21 | 10 | 7 | 19 | 26 | 21 | 63 |
| $L_1(p)$ | 0 | 1 | 0 | 12 | 6 | 19 | 6 | 16 | 19 | 6 | 19 | 22 | 16 | 42 |
| $L_2(p)$ | 0 | 2 | 0 | 14 | 9 | 24 | 9 | 21 | 24 | | | | | |

## 8- Conclusion

This paper investigates the modeling, performance evaluation, and optimization of non-deterministic workflows within Internet of Things (IoT) environments, emphasizing both methodological rigor and practical applicability. By abstracting workflow activities and resources into distinct representations, the study enables the computation of earliest and latest start times under uncertainty and the identification of critical activities that directly shape global workflow duration. This temporal analysis provides a systematic foundation for anticipating performance bottlenecks even in the absence of strict resource constraints. Building on this foundation, the study further addresses the practical challenge of resource conflicts inherent to real-world workflow execution. Three resolution strategies were designed, each balancing trade-offs between scheduling flexibility and execution efficiency. Comparative analysis revealed that selecting the most appropriate resolution strategy for each conflict scenario significantly enhances both time performance and resource utilization. Such adaptive optimization moves beyond static scheduling models and aligns more closely with the dynamic, heterogeneous nature of IoT systems.

The results carry important implications for theory and practice. From a methodological perspective, the framework advances workflow modeling by explicitly incorporating non-deterministic execution and preparation times, a dimension often neglected in existing studies. From a managerial standpoint, the findings provide actionable guidance for achieving operational agility in complex IoT-driven business processes, where uncertainty and resource scarcity are the norm rather than exceptions. Ultimately, this work underscores the necessity of integrating uncertainty-aware scheduling mechanisms into workflow design, offering both resilience and efficiency. Future research may extend these insights by exploring hybrid optimization methods or adaptive learning-based approaches to further enhance decision-making in dynamic workflow environments.

## 9- Declarations

### *9-1- Author Contributions*

Conceptualization, T.L. and H.L.; methodology, Q.Z.; software, H.L.; validation, H.D. and H.L.; formal analysis, T.L.; investigation, C.L.; data curation, C.L.; writing—original draft preparation, T.L.; writing—review and editing, T.L. and H.L.; visualization, C.L.; supervision, Q.Z.; project administration, H.D.; funding acquisition, C.L. All authors have read and agreed to the published version of the manuscript.

### *9-2- Data Availability Statement*

The data presented in this study are available on request from the corresponding author.

### *9-3- Funding*

This work was partly supported by the National Natural Science Foundation of China (No. 62472264 and 52574256), Natural Science Foundation of Shandong Province under Grants ZR2025QA13 and ZR2025ZD17, Education Ministry Humanities and Social Science Research Youth Fund Project of China under grant 24YJCZH194, the Taishan Scholar Program of Shandong Province, China under Grant TSTP20250506, and National funds through FCT (Fundação para a Ciência e a Tecnologia), under the project - UID/04152/2025 - Centro de Investigação em Gestão de Informação (MagIC)/NOVA IMS and UID/PRR/04152/2025.

### *9-4- Institutional Review Board Statement*

Not applicable.

### *9-5- Informed Consent Statement*

Not applicable.

### *9-6- Conflicts of Interest*

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

## 10- References

[1] Reisig, W. (2011). Petri Nets. Modeling in Systems Biology. Computational Biology, Volume 16. Springer, London, United Kingdom. doi:10.1007/978-1-84996-474-6_3.

[2] Wang, H., & Zeng, Q. (2008). Modeling and analysis for workflow constrained by resources and nondetermined time: An approach based on Petri nets. IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, 38(4), 802–817. doi:10.1109/TSMCA.2008.923056.

[3] Li, J. Q., Fan, Y. S., & Zhou, M. C. (2004). Performance modeling and analysis of workflow. IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, 34(2), 229–242. doi:10.1109/TSMCA.2003.819490.

[4] Van Der Aalst, W. M. P. (1998). The application of Petri nets to workflow management. Journal of Circuits, Systems and Computers, 8(1), 21–66. doi:10.1142/S0218126698000043.

[5] Li, J. Q., Fan, Y. S., & Zhou, M. C. (2003). Timing constraint workflow nets for workflow analysis. IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, 33(2), 179–193. doi:10.1109/TSMCA.2003.811771.

[6] Li, J., Fan, Y., & Zhou, M. (2003). Approximate performance analysis of Workflow model. SMC’03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483), 1175–1180 vol.2. doi:10.1109/icsmc.2003.1244570.

[7] De Maria, E., Montanari, A., & Zantoni, M. (2006). An automaton-based approach to the verification of timed workflow schemas. Thirteenth International Symposium on Temporal Representation and Reasoning (TIME’06), 87–94. doi:10.1109/time.2006.6.

[8] Liu, C., Wang, Y., Wen, L., Cheng, J., Cheng, L., & Zeng, Q. (2024). Discovering Hierarchical Multi-Instance Business Processes from Event Logs. IEEE Transactions on Services Computing, 17(1), 142–155. doi:10.1109/TSC.2023.3335360.

[9] Zeng, Q., Lu, F., Liu, C., Duan, H., & Zhou, C. (2015). Modeling and Verification for Cross-Department Collaborative Business Processes Using Extended Petri Nets. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 45(2), 349–362. doi:10.1109/tsmc.2014.2334276.

[10] Mo, Q., Wang, J., Xie, Z., Liu, C., & Dai, F. (2024). Enforcing Correctness of Collaborative Business Processes Using Plans. IEEE Transactions on Software Engineering, 50(9), 2313–2336. doi:10.1109/tse.2024.3431585.

[11] Liu, C. (2018). Automatic Discovery of Behavioral Models from Software Execution Data. IEEE Transactions on Automation Science and Engineering, 15(4), 1897–1908. doi:10.1109/TASE.2018.2844725.

[12] Liu, C. (2021). Discovery and Quality Evaluation of Software Component Behavioral Models. IEEE Transactions on Automation Science and Engineering, 18(4), 1538–1549. doi:10.1109/TASE.2020.3008897.

[13] Liu, C. (2020). Hierarchical Business Process Discovery: Identifying Sub-processes Using Lifecycle Information. 2020 IEEE International Conference on Web Services (ICWS), 423–427. doi:10.1109/icws49710.2020.00062.

[14] Liu, C. (2022). Formal Modeling and Discovery of Multi-instance Business Processes: A Cloud Resource Management Case Study. IEEE/CAA Journal of Automatica Sinica, 9(12), 2151–2160. doi:10.1109/JAS.2022.106109.

[15] Mo, Q., Wang, J., Jiang, Y., Xie, Z., Wang, W., Liu, C., & Dai, F. (2024). Enforcing Data-Aware Business Processes Using Execution Path-Oriented Strategies. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 54(11), 6708–6722. doi:10.1109/TSMC.2024.3427838.

[16] Liu, C., Li, H., Zhang, S., Cheng, L., & Zeng, Q. (2023). Cross-Department Collaborative Healthcare Process Model Discovery from Event Logs. IEEE Transactions on Automation Science and Engineering, 20(3), 2115–2125. doi:10.1109/TASE.2022.3194312.

[17] Duan, H., Liu, C., Zeng, Q., & Zhou, M. (2020). Refinement-Based Hierarchical Modeling and Correctness Verification of Cross-Organization Collaborative Emergency Response Processes. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 50(8), 2845–2859. doi:10.1109/TSMC.2018.2838053.

[18] Mo, Q., Wang, J., Jiang, C., Xie, Z., Liu, C., & Dai, F. (2024). Correctness Analysis of Cross-Organization Emergency Response Processes Based on Petri Nets. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 54(2), 800–812. doi:10.1109/TSMC.2023.3321060.

[19] Yu, W., Jia, M., Liu, C., & Ma, Z. (2020). Task Preemption Based on Petri Nets. IEEE Access, 8(1), 11512–11519. doi:10.1109/ACCESS.2020.2964382.

[20] Eder, J., Panagos, E., & Rabinovich, M. (1999). Time Constraints in Workflow Systems. Advanced Information Systems Engineering. CAiSE 1999. Lecture Notes in Computer Science, Volume 1626. Springer, Berlin, Germany. doi:10.1007/3-540-48738-7_22.

[21] Marjanovic, O., & Orlowska, M. E. (1999). On Modeling and Verification of Temporal Constraints in Production Workflows. Knowledge and Information Systems, 1(2), 157–192. doi:10.1007/bf03325097.

[22] Kotb, Y., Al Ridhawi, I., Aloqaily, M., Baker, T., Jararweh, Y., & Tawfik, H. (2019). Cloud-Based Multi-Agent Cooperation for IoT Devices Using Workflow-Nets. Journal of Grid Computing, 17(4), 625–650. doi:10.1007/s10723-019-09485-z.

[23] Kozma, D., Varga, P., & Larrinaga, F. (2021). Dynamic Multilevel Workflow Management Concept for Industrial IoT Systems. IEEE Transactions on Automation Science and Engineering, 18(3), 1354–1366. doi:10.1109/tase.2020.3004313.