IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

SPECIAL SECTION ON INTELLIGENT INFORMATION SERVICES

Received February 18, 2020, accepted February 26, 2020, date of publication March 3, 2020, date of current version March 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2978101

A Time Petri Net With Relaxed Mixed Semantics for Schedulability Analysis of Flexible Manufacturing Systems

LI PAN[1]*, BO YANG[1]*, JUNQIANG JIANG[1], AND MENGCHU ZHOU[2], (Fellow, IEEE)
[1]Department of Information Science and Engineering, Hunan Institute of Science and Technology, Yueyang 414006, China
[2]Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA

Corresponding author: Mengchu Zhou (zhou@njit.edu)

*Li Pan and Bo Yang are contributed equally to this work.

This work was supported in part by the Hunan Provincial Natural Science Foundation of China under Grant 2017JJ2016, Grant 2018JJ2152, Grant 2018JJ2153, and Grant 2019JJ40105, in part by the Science and Technology Program of Hunan Province under Grant 2018TP2022, and in part by the Scientific Research Fund of Hunan Provincial Education Department under Grant 17A089 and Grant 18B356.

ABSTRACT Several semantics models are adopted by time Petri nets for different applications. Yet they have some limitations on schedulability analysis of flexible manufacturing systems. The scheduling scope of a strong semantics model is greatly limited because of the impact of strong timing requirements, perhaps keeping some optimal schedules out of the consideration. A weak semantics model cannot guarantee the scheduling timeliness as there lacks strong timing enforcement. A mixed semantics model cannot ensure that independent transitions with overlapping firing interval fire in an interleaving way, thus affecting the search for the optimal schedules. In this paper, we present a relaxed mixed semantics model for time Petri nets to address these problems by redefining the firability rules of transitions. In our model, the firability of a transition is determined by maximal concurrent sets containing the transition. This treatment not only extends the scheduling scope of TPN model greatly while avoiding the generation of invalid schedules, but also solves the problem of concurrent scheduling of independent transitions. A state class method is then proposed to support the verification and analysis of temporal properties. Finally, we apply the proposed model to schedulability analysis of a job shop scheduling problem, and compare the features of four semantics models.

INDEX TERMS Scheduling, semantics models, state class methods, schedulability analysis, time Petri nets.

I. INTRODUCTION

Petri nets as a mathematical tool, have been widely used to handle many problems in discrete event systems [1]–[4]. Flexible manufacturing systems are the typical discrete event dynamic systems. Time uncertainty exists in flexible manufacturing systems, and can be described by using time intervals [5]–[7]. Thus, the scheduling problems in the area of flexible manufacturing with time uncertainty can be modeled by time Petri nets (TPNs) [8]–[12].

In TPNs, a transition is associated with a time interval representing all its possible firing time relative to its enabling instant. Once a transition is enabled, its dynamic firing interval is initialized to its static interval. Its dynamic interval decrease synchronously with time. It can fire only when the lower bound of its dynamic interval reaches zero. Firing a transition takes no time.

When the upper bound of its dynamic interval decreases to zero, two different time semantics are usually adopted [8]–[10], [13]–[15]: strong (time) semantics and weak (time) one. The former forces a transition to fire when its upper bound reaches zero. The latter does not force transitions to fire within their time bounds. That is to say, it allows the upper bound of any transition to be below zero. In this case, this transition will no longer be possible to fire unless it becomes enabled again.

The key competitive strength of a manufacturing system lies in its flexibility, which represents the ability to respond effectively to changing circumstances. The efficient operation of a flexible manufacturing system can be achieved by exploiting routing flexibility and scheduling flexibility. Thus a Petri net-based scheduling model for a flexible

The associate editor coordinating the review of this manuscript and approving it for publication was Shouguang Wang.

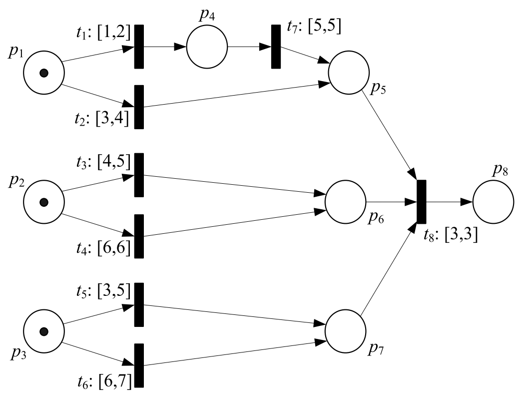


FIGURE 1. A TPN with choice structures.

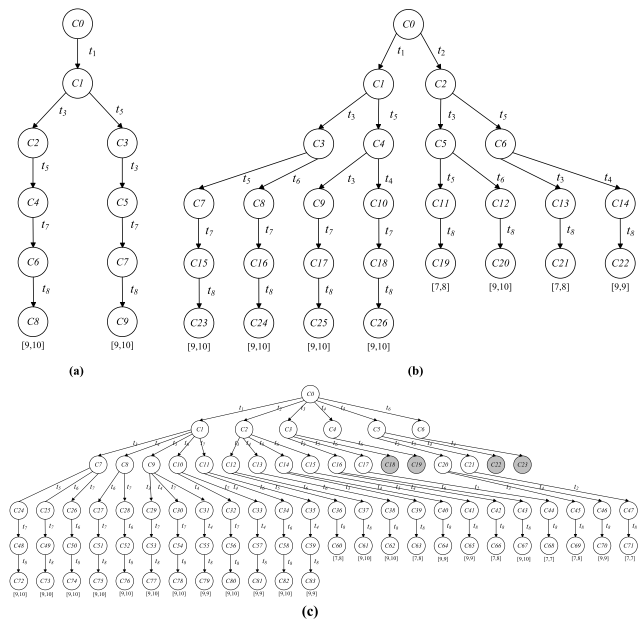


FIGURE 2. Reachability trees of the TPN in Fig. 1 for the existing semantics models. (a) Reachability tree for strong semantics model, (b) Reachability tree for mixed semantics model, (c) Reachability tree for weak semantics model.

manufacturing system usually has abundant choice structures that are used to model route selection and resource allocation.

The process of verifying whether a schedule of task execution meets the imposed timing constraints is referred to as schedulability analysis, which is critical in maintaining correctness of timed-dependent systems [16]–[19]. However, there are still some limitations on schedulability analysis of flexible manufacturing systems for the existing semantics models of time Petri nets.

We consider a simple TPN with optional tasks in Fig.1. In strong semantics, the firability of a transition is affected by time constraints of all enabled transitions. Thus t_2, t_4 and t_6 cannot be scheduled in any case. Only two schedules $t_1t_3t_5t_7t_8$ and $t_1t_5t_3t_7t_8$ are feasible by using such semantics (see Fig. 2a). The problem may cause some enabled transitions never to be scheduled, and perhaps keep some optimal schedules out of the consideration, such as $t_5t_3t_2t_8$ or $t_3t_5t_2t_8$.

In weak semantics, each enabled transition can fire in its time interval, and may miss its deadline. In Fig.1, t_1, t_2, t_3, t_4, t_5 and t_6 are firable at the initial state s_0. If t_4 or t_6 is fired at s_0, then it means that t_1 and t_2 have missed their own deadlines. That is to say, weak semantics cannot guarantee the timeliness of task execution. The problem may cause some invalid schedules produced in this model, such as t_3t_6, t_4t_6, t_5t_4 and t_6t_4 (see Fig. 2c).

In [19], Pan, et al. proposed a mixed semantics model to try to overcome limitations of the strong and weak semantics models. This model removes the impact of time constraints of conflicting transitions on the firability of a transition, i.e., the firability of a transition is determined by time constraints of its non-conflicting transitions. However, this model cannot guarantee the interleaving execution of independent transitions. For example, in Fig. 1, t_2 and t_5 are two independent enabled transitions with overlapping firing intervals. t_2 is firable at s_0 because it can fire before its all non-conflicting transitions (t_3, t_4, t_5 and t_6). But t_5 cannot be scheduled at s_0 due to the time constraint of its non-conflicting transition t_1. Thus, in the model, schedule $t_2t_5t_4t_8$ is feasible but $t_5t_2t_4t_8$ is not (see Fig. 2b).

In brief, strong semantics models may cause some enabled transitions never to be scheduled; weak semantics models may produce invalid schedules; and mixed semantics models cannot ensure that independent transitions with overlapping firing intervals fire in an interleaving way. To attack the above scheduling issues, a new semantics model is required for time Petri nets.

This paper presents a relaxed mixed semantics model to address these scheduling analysis issues. In the model, at least a progressive maximal concurrent transition set is preserved by redefining firabiltiy rules of transitions. The treatment not only extends the scheduling scope of the strong semantics models while avoiding the generation of invalid schedules in the weak semantics models, but also solves the problem of interleaving execution of independent transitions in the mixed semantics models.

Our model has the following properties: 1) any enabled transition can be fired in some reachable state; 2) there are no overdue enabled transition set; and 3) independent transitions with overlapping firing intervals in a maximal concurrent set can fire in an interleaving way.

The rest of the paper is organized as follows. In Section II, we define formal semantics of the existing semantics models. Section III presents a relaxed mixed semantics model, and proves that the model has particular properties such that the problems of the existing models can be overcome. Section IV presents a state class method for the schedulability analysis of timed systems. In Section V, we apply the proposed model to a job shop scheduling problem of flexible manufacturing systems, and compare the abilities of the four semantics models in schedulability analysis. Section VI concludes the paper.

II. THE EXISTING SEMANTICS MODELS

A. TIME PETRI NETS

Let R (R^+) be the set of (nonnegative) real numbers. An interval is a connected subset of R. Formally,

$$I = [a, b] \text{ is an interval if } I = \{x \in R | a \leq x \leq b\},$$

where $a \in R$, $b \in R \cup \{\infty\}$ and $a \leq b$. When $a = b$, we abbreviate $[a, a]$ to a. The lower and upper bounds of interval I are denoted by $\downarrow I$ and $\uparrow I$, respectively.

Let $\mathbb{I}(\mathbb{I}^+)$ denote the set of all (nonnegative) intervals. Let $I_1, I_2 \in \mathbb{I}$ and $a \in R$. Their operations are defined as:

1. $I_1 + I_2 = [\downarrow I_1 + \downarrow I_2, \uparrow I_1 + \uparrow I_2]$;
2. $I_1 - I_2 = [\downarrow I_1 - \uparrow I_2, \uparrow I_1 - \downarrow I_2]$;
3. $a^*I = [a^* \downarrow I_1, a^* \uparrow I_1]$, and $a + I = [a + \downarrow I_1, a + \uparrow I_1]$;
4. $I_1 \cap I_2 = [\max\{\downarrow I_1, \downarrow I_2\}, \min\{\uparrow I_1, \uparrow I_2\}]$, if $I_1 \cap I_2 \neq \emptyset$.

A TPN is a 6-tuple $TPN = (P, T, B, F, M_0, SI)$ where

1. $P = \{p_1, p_2, \ldots, p_m\}$ is a finite nonempty set of places;
2. $T = \{t_1, t_2, \ldots, t_n\}$ is a finite nonempty set of transitions;
3. $B: P \times T \to N$ is the backward incidence matrix;
4. $F: P \times T \to N$ is the forward incidence matrix;
5. $M_0: P \to N$ is the initial marking;
6. $SI: T \to \mathbb{I}^+$ is a mapping called static firing interval. $\forall t \in T$, $SI(t)$ represents t's static firing interval relative to the time at which t is enabled.

We denote by $B(t)$ the vector of input places of transition t, which corresponds to the vector of column t in the backward incidence matrix. Similarly, $F(t)$ represents the vector of output places of t.

A marking of a Petri net is an assignment of tokens to places, i.e., a mapping $M: P \to N$. A transition t is enabled at marking M, if

$$\forall p \in P : B(p, t) \leq M(p).$$

Let $En(M)$ be the set of transitions enabled at marking M. Let $New(M, t_f)$ denote the set of newly enabled transitions by firing t_f from M, which is defined by

$$New(M, t_f) = En(M - B(t_f) + F(t_f)) \backslash En(M - B(t_f)).$$

Note that a transition that is disabled at intermediate marking $M - B(t_f)$ but enabled at new marking $M - B(t_f) + F(t_f)$ is considered as a newly enabled one. In order to simplify the treatment of the problem, we do not consider multi-enabledness of transitions [20]–[22].

A state of a TPN is a pair $s = (M, f)$, where

1. M is a marking; and
2. f is a dynamic firing interval function. $\forall t \in En(M), f(t)$ represents t's firing interval in which each value is a possible firing time relative to the current state.

The initial state is defined as $s_0 = (M_0, f_0)$, where M_0 is the initial marking, and $f_0(t) = SI(t)$ for all $t \in En(M_0)$.

B. STRONG SEMANTICS MODELS

To describe different semantics models of time Petri nets in a uniform way, we introduce the concepts of time bound and efficient firing interval.

In a time Petri net with strong semantics (S-TPN), an enabled transition t is firable at state s if

$$\downarrow f(t) \leq \uparrow f(t_i) \text{ for any } t_i \in En(M).$$

Let $Fr(s)$ be the set of all firable transitions at s. The efficient firing time bound of firable transition t at state s is defined as

$$\Gamma(t) = \min\{\uparrow f(t_i) | t_i \in En(M)\}.$$

$\Gamma(t)$ represents t's maximal enabling time that is allowed from s to the next state if firing t in an S-TPN. The efficient firing interval of firable transition t at state s is defined as

$$\Theta(t) = [\downarrow f(t), \Gamma(t)].$$

The semantics of a TPN model can be characterized by a Labeled Transition System (LTS) [23], [24] that is defined below. An LTS is a quadruple $L = (S, s_0, \Sigma, \to)$ where

1. S is a finite set of states;
2. $s_0 \in S$ is the initial state;
3. Σ is a set of labels representing activities; and
4. $\to$ is the transition relation.

Given $TPN = (P, T, B, F, M_0, SI)$, the formal semantics of its S-TPN is defined as $L_s = (S_s, s_0, \Sigma, \to_s)$ such that

1. $S_s = N^P \times \mathbb{I}^T$;
2. $s_0 = (M_0, f_0)$;
3. $\Sigma \subseteq T \times R^+$;
4. $\to_s \subseteq S_s \times \Sigma \times S_s$ is the transition relation, $\forall d \in R^+, \forall t_f \in T, (M, f) \xrightarrow{t_f, d}_s (M', f')$ iff

$$\begin{cases} t_f \in Fr(s) & (1) \\ d \in \Theta(t_f) & (2) \\ M' = M - B(t_f) + F(t_f) & (3) \\ \forall t \in En(M'), f'(t) = \begin{cases} SI(t) & \text{if } t \in New(M, t_f) \\ f(t) - d & \text{otherwise} \end{cases} & (4) \end{cases}$$

From the transition relation, it is easy to see that (1) ensures that t_f is firable at state s; (2) determines the range of efficient firing time of t_f at s; (3) describes the marking transformation; (4) computes firing intervals of all transitions enabled at state s' after firing t_f.

For example, in Fig. 1, according to the formal semantics of S-TPNs, only t_1 is firable at s_0. Thus only schedules $t_1t_3t_5t_7t_8$ and $t_1t_5t_3t_7t_8$ are feasible in the model. The problem greatly narrows the scheduling scope of TPNs. Some desired schedules, like the optimal schedules $t_5t_3t_2t_8$ and $t_3t_5t_2t_8$, are not produced with such model.

C. WEAK SEMANTICS MODELS

In a time Petri net with weak semantics (W-TPN), enabled transition t is firable at state s if

$$\uparrow f(t) \geq 0.$$

The efficient firing time bound of firable transition t at state s is defined as

$$\Gamma(t) = \uparrow f(t).$$

The efficient firing interval of firable transition t at state s is defined as

$$\Theta(t) = [\max(0, \downarrow f(t)), \Gamma(t)].$$

Given $TPN = (P, T, B, F, M_0, SI)$, the formal semantics of its W-TPN is defined as $L_w = (S_w, s_0, \Sigma, \rightarrow_w)$ such that

1. $S_w = N^P \times \mathbb{I}^T$;
2. $s_0 = (M_0, f_0)$;
3. $\Sigma \subseteq T \times R^+$;
4. $\rightarrow_w \subseteq S_w \times \Sigma \times S_w$ is the transition relation, which is the same as that of an S-TPN.

In a W-TPN, the firability of a transition only depends on its own time constraint. As a result, firing a transition may cause some other non-conflicting enabled transitions to miss their deadlines. Mathematically, an enabled transition t is overdue at state s if $\uparrow f(t) < 0$. Otherwise t is progressive (i.e., $\uparrow f(t) \geq 0$). An overdue enabled transition is not firable, because its firing deadline has been missed.

For example, in Fig. 1, if we fire t_4, then both t_1 and t_2 become overdue for the missing of their deadlines. In this case, t_8 is not schedulable. As a result, the whole task cannot be finished. The problem results in some invalid schedules in the state space of such model.

D. MIXED SEMANTICS MODELS

Let $S \subseteq En(M)$. We define the minimal time upper bound of set S by $\underline{\uparrow} S = \min\{\uparrow f(t) | t \in S\}$. Let $En(M - B(t))$ be t's non-conflicting enabled transition set.

In a time Petri net with mixed semantics (M-TPNs), progressive enabled transition t is firable at state s if

$$\max(\downarrow f(t), 0) \leq \uparrow f(t_i) \text{ for any } t_i \in En(M - B(t)).$$

In other word, progressive enabled transition t is firable if it can fire before its all non-conflicting transitions. The efficient firing time bound of firable transition t at state s is defined as

$$\Gamma(t) = \underline{\uparrow}(En(M - B(t)) \cup \{t\}).$$

$\Gamma(t)$ indicates that the firing time of transition t at state s cannot overtake the upper bounds of firing intervals of its all non-conflicting transitions and itself. The efficient firing interval of firable transition t at state s is defined as

$$\Theta(t) = [\max(0, \downarrow f(t)), \Gamma(t)].$$

Given $TPN = (P, T, B, F, M_0, SI)$, the formal semantics of its M-TPN is defined as $L_m = (S_m, s_0, \Sigma, \rightarrow_m)$ such that

1. $S_m = N^P \times \mathbb{I}^T$;
2. $s_0 = (M_0, f_0)$;
3. $\Sigma \subseteq T \times R^+$;
4. $\rightarrow_m \subseteq S_m \times \Sigma \times S_m$ is the transition relation, which is the same as that of an S-TPN.

For example, in Fig. 1, t_2 is firable because it can fire before its all non-conflicting transitions. However, t_5 is not firable because it cannot fire before its non-conflicting transition t_1. This is irrational as t_2 and t_5 are two independent transitions with the overlapping firing intervals. Therefore, this model cannot ensure the interleaving execution of two independent transitions.

III. RELAXED MIXED SEMANTICS MODEL

A. FORMAL SEMANTICS

We present a relaxed mixed semantics model for time Petri nets (RM-TPN) to address the above problems. In this model, we redefine transition firability by further loosening firable conditions of M-TPNs to achieve the desired properties.

Two transitions t_i and t_j are concurrent (or independent) at marking M, denoted by $t_i || t_j$, if $B(t_i) + B(t_j) \leq M$. For transition set $U \subseteq T$, if $\forall t_i, t_j \in U$, $t_i || t_j$, then U is a concurrent set at M. We say that concurrent set U is maximal if it is not a subset of any other concurrent set. Formally, U is a maximal concurrent set if $\sum_{t \in U} B(t) \leq M$ and $En(M - \sum_{t \in U} B(t)) = \emptyset$.

Let $\mathbb{U}(M)$ be the set of all maximal concurrent sets at M and $\mathbb{U}(M, t) = \{U | U \in \mathbb{U}(M) \wedge t \in U\}$ be the set of maximal concurrent sets containing t. For $U \in \mathbb{U}(M)$, if $\forall t \in U$, t is progressive, we say that U is progressive. In other word, U is progressive if and only if $\underline{\uparrow} U \geq 0$.

In an RM-TPN, progressive enabled transition t is firable at state s, if

$$\exists U \in \mathbb{U}(M, t), \text{ such that } \max(0, \downarrow f(t)) \leq \underline{\uparrow} U.$$

In other words, t is firable if there is a maximal concurrent set U containing t, such that t may fire before any other transition in U. The efficient firing time bound of firable transition t at state s is defined as

$$\Gamma(t) = \max\{\underline{\uparrow} U | U \in \mathbb{U}(M, t)\}.$$

$\Gamma(t)$ indicates that the firing time of transition t at state s cannot overtake minimal time upper bounds of all maximal concurrent sets containing t at M. That is to say, each transition firing can keep at least a progressive maximal concurrent set at the new state. The efficient firing interval of firable transition t at state s is defined as

$$\Theta(t) = [\max(0, \downarrow f(t)), \Gamma(t)].$$

Given $TPN = (P, T, B, F, M_0, SI)$, the formal semantics of its RM-TPN is defined as $L_r = (S_r, s_0, \Sigma, \rightarrow_r)$ such that

1. $S_r = N^P \times \mathbb{I}^T$;
2. $s_0 = (M_0, f_0)$;
3. $\Sigma \subseteq T \times R^+$;
4. $\rightarrow_r \subseteq S_r \times \Sigma \times S_r$ is the transition relation, which is the same as that of S-TPN.

Determining the firability of a transition needs to enumerate all maximal concurrent sets containing the transition. Obviously, this enumeration takes exponential time. Thus, we must reduce its computational complexity.

Lemma 1: Given that $S \subseteq En(M)$, if $En(M - \sum_{t\in S} B(t)) = \emptyset$, then $\exists U \subseteq S$, such that $U \in \mathbb{U}(M)$.

Proof: If $En(M - \sum_{t\in S} B(t)) = \emptyset$ and $\sum_{t\in S} B(t) \leq M$, from the definition of a maximal concurrent set, it follows that $S \in \mathbb{U}(M)$. Otherwise, there is $p \in P$ such that $\sum_{t\in S} B(p, t) > M(p)$, i.e., there are conflicting transitions in S. We obtain $U \subseteq S$ by eliminating conflicting transitions from S until $\sum_{t\in U} B(t) \leq M$. Then U must be a maximal concurrent set. Because if $En(M - \sum_{t\in U} B(t)) \neq \emptyset$, then $\exists t_i \in S$ such that $t_i \in En(M - \sum_{t\in U} B(t))$. It follows that t_i is not in conflict with transitions in U and thus t_i should not be removed from S. This is a contradiction.

Lemma 1 shows that if $En(M - \sum_{t\in S} B(t))$ is empty, then S must include a maximal concurrent set.

Property 1: Let $S = \{t | t \in En(M - B(t_f)) \wedge \uparrow f(t) \geq \max(0, \downarrow f(t_f))\}$, where t_f is a progressive enabled transition at M. Then $t_f \in Fr(s)$ if and only if $En(M - \sum_{t\in S\cup\{t_f\}} B(t)) = \emptyset$.

Proof: ($\Leftarrow$) If $En(M - \sum_{t\in S\cup\{t_f\}} B(t)) = \emptyset$, by Lemma 1, then $\exists U \subseteq S \cup \{t_f\}$ such that $U \in \mathbb{U}(M, t_f)$. Since $\forall t \in S$, $\uparrow f(t) \geq \max(0, \downarrow f(t_f))$, it follows that $\underline{\uparrow} U \geq \max(0, \downarrow f(t_f))$. According to the definition of transition firability, we obtain that $t_j \in Fr(s)$.

($\Rightarrow$) If $En(M - \sum_{t\in S\cup\{t_f\}} B(t)) \neq \emptyset$, by the definition of S, then $\forall t' \in En(M - \sum_{t\in S\cup\{t_f\}} B(t))$, $\uparrow f(t') < \max(0, \downarrow f(t_f))$. Thus, $\forall U \in \mathbb{U}(M, t_f)$, there must be $t' \in En(M - \sum_{t\in S\cup\{t_f\}} B(t))$ such that $t' \in U$, and then $\underline{\uparrow} U \leq \uparrow f(t') < \max(0, \downarrow f(t_f))$. Thus $t_f \notin Fr(s)$.

Property 1 transforms the enumeration computation of transition firability into the emptiness determination of set $En(M - \sum_{t\in S\cup\{t_f\}} B(t))$, where S is a set of t_f' s non-conflicting transitions that can fire after t_f.

A progressive enabled transition t_f is firable at state s, if

$$En(M - \sum_{t\in S\cup\{t_f\}} B(t)) = \emptyset$$

where $S = \{t | t \in En(M - B(t_f)) \wedge \uparrow f(t) \geq \max(\downarrow f(t_f), 0)\}$.

Consider the example in Fig. 1. At s_0, for t_5, we have that $f_0(t_5) = [3, 5]$, $S = \{t | t \in En(M_0 - B(t_5)) \wedge \uparrow f_0(t) \geq 3\} = \{t_2, t_3, t_4\}$ and $En(M_0 - B(t_5) - B(t_2) - B(t_3) - B(t_4)) = \emptyset$. Thus $t_5 \in Fr(s_0)$. For t_4, we have that $f_0(t_4) = [6, 6]$, $S = \{t | t \in En(M_0 - B(t_4)) \wedge \uparrow f_0(t) \geq 6\} = \{t_6\}$ and $En(M_0 - B(t_4) - B(t_6)) = \{t_1, t_2\} \neq \emptyset$. Thus $t_4 \notin Fr(s_0)$.

Next, we give an algorithm to compute efficient firing time bounds of firable transitions. Let t is a firable transition at s. As shown in Algorithm 1, if $En(M - B(t)) = \emptyset$, then $\{t\}$ is the only maximal concurrent set containing t at M. From the definition of efficient firing time bounds, it follows that $\Gamma(t) = \uparrow f(t)$. Otherwise, we construct S by selecting from $En(M - B(t))$ progressive enabled transitions whose upper firing time bounds are not less than $\downarrow f(t)$. Then we compare $\uparrow f(t)$ with $\underline{\uparrow} S$. If $\uparrow f(t) \leq \underline{\uparrow} S$, then $\forall U \in \mathbb{U}(M, t)$, thus we return $\Gamma(t) = \uparrow f(t)$. If $\uparrow f(t) > \underline{\uparrow} S$, then $\Gamma(t)$ is set to $\underline{\uparrow} S$, and S is updated by selecting progressive enabled transitions that can fire after $\underline{\uparrow} S$. Repeat this process until $En(M - \sum_{t_i\in S\cup\{t\}} B(t_i)) \neq \emptyset$.

Algorithm 1 Computation of Time Bound of Firable Transition t

Input: $TPN = (P, T, B, F, M_0, SI)$, $s = (M, f)$, $t \in Fr(s)$
Output: Firing time bound tb of transition t at state s

```
If En(M − B(t)) = ∅
    Return ↑f(t)
Let S = {t_i | t_i ∈ En(M − B(t)) ∧ ↑f(t_i)
    ≥ max(0, ↓f(t))}
While En(M − Σ_{t_i∈S∪{t}} B(t_i)) = ∅
    If ↑f(t) ≤ ↑S
        Return ↑f(t)
    Else
        tb = ↑S
        S = {t_i | t_i ∈ En(M − B(t)) ∧ ↑f(t_i) > tb}
Return tb
```

Let $n = |En(M - B(t))|$ be the number of t's non-conflicting enabled transitions. The number of iterations of Algorithm 1 is not more than n. In each iteration, the computation of set S needs time O (n). Hence, time complexity of the algorithm is $O(n^2)$.

We consider the example in Fig. 1. At s_0, for $t_2 \in Fr(s_0)$, we have that $f_0(t_2) = [3, 4]$, $S = \{t | t \in En(M_0 - B(t_2)) \wedge \uparrow f_0(t) \geq 3\} = \{t_3, t_4, t_5, t_6\}$ and $En(M_0 - B(t_2) - B(t_3) - B(t_4) - B(t_5) - B(t_6)) = \emptyset$. Since $\uparrow f_0(t_2) = 4 < \underline{\uparrow} S = 5$, $\Gamma(t_2) = 4$. For $t_5 \in Fr(s_0)$, we have that $f_0(t_5) = [3, 5]$, $S = \{t | t \in En(M_0 - B(t_5)) \wedge \uparrow f_0(t) \geq 3\} = \{t_2, t_3, t_4\}$ and $En(M_0 - B(t_5) - B(t_2) - B(t_3) - B(t_4)) = \emptyset$. Since $\uparrow f_0(t_5) = 5 > \underline{\uparrow} S = 4$, $tb = \underline{\uparrow} S = 4$. Then $S' = \{t | t \in En(M_0 - B(t_5)) \wedge \uparrow f_0(t) > 4\} = \{t_3, t_4\}$ and $En(M_0 - B(t_5) - B(t_3) - B(t_4)) \neq \emptyset$. Thus $\Gamma(t_5) = tb = 4$.

A firing sequence σ is a finite (or infinite) string consisting of symbols in transition set T. The *empty sequence* is the sequence with zero occurrences of symbols. A *run* ρ of a time Petri net is a finite or infinite sequence of the form $\rho = s_0 \xrightarrow{t_0, d_0} s_1 \xrightarrow{t_1, d_1} \ldots \xrightarrow{t_{n-1}, d_{n-1}} s_n \ldots$. We write $s \xrightarrow{*} s'$ if there is a run ρ such that s is the initial state of ρ and s' the final state of ρ. Let $Rs(s) = \{s' | s \xrightarrow{*} s'\}$ be the set of all states reachable from state s.

Next, we illustrate a run of the RM-TPN in Fig.1 with $\rho = s_0 \xrightarrow{t_3, 3}_r s_1 \xrightarrow{t_2, 1}_r s_2 \xrightarrow{t_5, 1}_r s_3 \xrightarrow{t_8, 1}_r s_4$. The computation process of the run is shown in Table 1. Note that schedule $t_3 t_2 t_5 t_8$ is feasible in the RM-TPN, but not in the S-TPN and M-TPN.

B. MODEL PROPERTIES

In the subsection, we prove that RM-TPNs have particular properties that can overcome the mentioned problems in the existing semantics models.

TABLE 1. Computation process of a run of the RM-TPN in Fig. 1.

State		Elapsed time	Fired transition
s_0	$M_0 = (1\ 1\ 1\ 0\ 0\ 0\ 0\ 0)$ $f_0(t_1) = [1,2]$ $f_0(t_2) = [3,5]$ $f_0(t_3) = [3,4]$ $f_0(t_4) = [6,6]$ $f_0(t_5) = [3,5]$ $f_0(t_6) = [6,7]$ $Fr(s_0) = \{t_1,t_2,t_3,t_5\}$ $\Gamma_0(t_1) = 2$ $\Gamma_0(t_2) = 4$ $\Gamma_0(t_3) = 4$ $\Gamma_0(t_5) = 4$	$d_0=3$	t_3
s_1	$M_1 = (1\ 0\ 1\ 0\ 0\ 1\ 0\ 0)$ $f_1(t_1) = [-2,-1]$ $f_1(t_2) = [0,2]$ $f_1(t_5) = [0,2]$ $f_1(t_6) = [3,4]$ $Fr(s_1) = \{t_2,t_5\}$ $\Gamma_1(t_2) = 2$ $\Gamma_1(t_5) = 2$	$d_1=1$	t_2
s_2	$M_2 = (0\ 0\ 1\ 0\ 1\ 1\ 0\ 0)$ $f_2(t_5) = [-1,1]$ $f_2(t_6) = [2,3]$ $Fr(s_2) = \{t_5,t_6\}$ $\Gamma_2(t_5) = 1$ $\Gamma_2(t_6) = 3$	$d_2=1$	t_5
s_3	$M_3 = (0\ 0\ 0\ 0\ 1\ 1\ 1\ 0)$, $f_3(t_8) = [1,1]$ $Fr(s_3) = \{t_8\}$ $\Gamma_3(t_8) = 1$	$d_3=1$	t_8
s_4	$M_4 = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 1)$ $Fr(s_4) = \varnothing$	/	/

Property 2: In an RM-TPN, if t is enabled at $s \in Rs(s_0)$, then $\exists s' \in Rs(s_0)$ such that $t \in Fr(s')$.

Proof: Assume that t begins enabled at $s \in Rs(s_0)$, then $f(t) = SI(t)$. We consider two cases:

(1) If $\downarrow f(t) \leq \max\{\underline{\uparrow} U | U \in \mathbb{U}(M,t)\}$, then $\exists U \in \mathbb{U}(M,t)$ such that $0 \leq\downarrow f(t)) \leq \underline{\uparrow} U$. From the definition of transition firability, it follows that $t \in Fr(s)$.

(2) If $\downarrow f(t) > \max\{\underline{\uparrow} U | U \in \mathbb{U}(M,t)\}$, then $\underline{\uparrow} f(t) \geq\downarrow f(t) > \max\{\underline{\uparrow} U | U \in \mathbb{U}(M,t)\}$. It follows that $\exists U \in \mathbb{U}(M,t)$ and $t_i \in U$ such that $\underline{\uparrow} U = \uparrow f(t_i)$, then $\exists U \in \mathbb{U}(M,t_i)$ such that $\downarrow f(t_i) \leq \uparrow f(t_i) \leq \underline{\uparrow} U$, and thus $t_i \in Fr(s)$. Let $d_i = \uparrow f(t_i)$. We execute $s \xrightarrow{t_i,d_i}_{\text{r}} s_1$. From the transition relation of RM-TPNs, it follows that $t \in En(M_1)$ and $f_1(t) = f(t) - d_i$. Similarly, at s_1, if $\downarrow f_1(t) > \max\{\underline{\uparrow} U | U \in \mathbb{U}(M_1,t)\}$, there must be $U \in \mathbb{U}(M_1,t)$ and $t_j \in U$, such that $\underline{\uparrow} U = \uparrow f(t_j)$. Then we run $s_1 \xrightarrow{t_j,d_j}_{\text{r}} s_2$ where $d_j = \uparrow f(t_j)$. As a result, $\downarrow f_2(t)$ is smaller than $\downarrow f_1(t)$. Repeating this way, there must be some state $s' \in Rs(s)$, such that $\downarrow f'(t) \leq \max\{\underline{\uparrow} U | U \in \mathbb{U}(M',t)\}$. According to the result of (1), we have that $t \in Fr(s')$.

Property 2 shows that a transition enabled at a state can be scheduled at some state reachable from the state, i.e., an enabled transition can surely be scheduled in an RM-TPN. But S-TPNs do not meet the property, like t_2 in Fig. 1.

Property 3: In an RM-TPN, $En(M) \neq \emptyset$ if and only if $Fr(s) \neq \emptyset$.

Proof: ($\Leftarrow$) If $En(M) = \emptyset$, by the firability definition, then there is no firable transition at s. Thus $Fr(s) = \emptyset$.

($\Rightarrow$) If $En(M) \neq \emptyset$, we use mathematical induction method to prove that $Fr(s) = \emptyset$. The proof is carried out by induction on state s_n.

For the basis case (state s_0), if $En(M_0) \neq \emptyset$, then $\forall t \in En(M_0)$, $f_0(t) = SI(t)$. We select transition t_m with the minimal upper time bound from $En(M_0)$, i.e., $\uparrow f_0(t_m) = \underline{\uparrow} En(M_0)$. If $En(M_0 - B(t_m)) = \emptyset$, then $\exists U = \{t_m\}$ such that $0 \leq\downarrow f_0(t_m) \leq \underline{\uparrow} U$. According to the firability definition, we have that $t_m \in Fr(s)$. If $En(M_0 - B(t_m)) \neq \emptyset$, we let $S = En(M_0 - B(t_m))$. Then $En(M - \sum_{t \in S\cup\{t_m\}} B(t)) = \emptyset$, by the rewrited firability definition, we obtain that $t_m \in Fr(s_0)$. Hence, the assert holds for $n = 0$.

Assume that the assertion holds for $n \leq k$. Consider $n = k+1$.

By induction hypothesis, $Fr(s_k) \neq \emptyset$. Let us suppose that $t_k \in Fr(s_k)$ and $s_k \xrightarrow{t_k}_{\text{r}} s_{k+1}$. Then $\exists U \in \mathbb{U}(M_k, t_k)$ such that $\max(0, \downarrow f_k(t_k)) \leq \underline{\uparrow} U$. Since t_k is independent of any other transition in U, the firing of t_k does not change independence relations between these transitions in U. Let $U' = U/\{t_k\}$. At s_{k+1}, we have that $\uparrow U' \geq 0$ by the formal semantics of RM-TPNs. Let U'' be a maximal concurrent set at M_{k+1} including U'. Since there is no multi-enabledness of transitions, the transitions in U''/U' are newly enabled at M_{k+1}. We let t_m be the transition with the minimal upper time bound in U''. It follows that $t_m \in Fr(s_{k+1})$ from the firability definition. Hence, the assertion holds for $n = k+1$.

Property 3 shows that RM-TPNs can keep the consistency of the non-emptiness of the enabled and firable transition sets at any state. However, W-TPNs do not satisfy the property. For example, in Fig. 1, after firing t_4 and t_6, both t_1 and t_2 are overdue enabled, i.e., $En(M) = \{t_1, t_2\}$, but $Fr(s) = \emptyset$. Therefore, W-TPNs may produce some invalid schedules due to the lack of timing enforcement.

Property 4: In an RM-TPN, for two progressive enabled transitions t_i and t_j, if $\exists U \in \mathbb{U}(M)$ such that $t_i, t_j . \in U$ and $\max(0, \downarrow f(t_i), \downarrow f(t_j)) \leq \underline{\uparrow} U$, then t_i and t_j can fire in an interleaving way from s.

Proof: Since $t_i, t_j \in U$ and $\max(0, \downarrow f(t_i), \downarrow f(t_j)) \leq \underline{\uparrow} U$, we have that $t_i || t_j$ and $t_i, t_j \in Fr(s)$. Let $ED = [\max(0, \downarrow f(t_i), \downarrow f(t_j)), \underline{\uparrow} U]$. If we execute $s \xrightarrow{t_i,d}_{\text{r}} s'$ with $d \in ED$, then $\downarrow f'(t_j) =\downarrow f(t_j) - d \leq 0$ and $\uparrow f'(t_j) =\uparrow f(t_j) - d \geq 0$. At s', there must be $U' \in \mathbb{U}(M', t_j)$ such that $\downarrow f'(t_j) \leq 0 \leq \underline{\uparrow} U'$, thus we obtain that $t_j \in Fr(s')$. Similarly, we may fire t_j and t_i successively from s. Hence, t_i and t_j can fire independently from s.

Property 4 shows that two independent transitions with overlapping efficient firing intervals in a maximal concurrent set can fire independently in an RM-TPN. Their overlapping efficient firing interval is $[\max(0, \downarrow f(t_i), \downarrow f(t_j)), \underline{\uparrow} U]$. However, M-TPNs do not meet the property. For example, in Fig. 1, $t_2 t_5$ can be scheduled in the M-TPN but $t_5 t_2$ cannot. The cause of the issue is that the firabilities of two independent transitions are determined by different transition sets in an M-TPN. For example, t_2's firability is decided by transition set $\{t_2, t_3, t_4, t_5, t_6\}$ and t_5's by $\{t_5, t_1, t_2, t_3, t_4\}$.

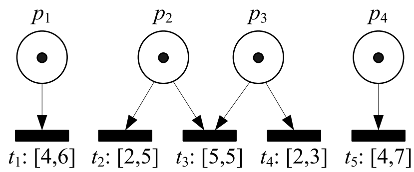


FIGURE 3. **Independent transitions and timed concurrency.**

Note that two independent firable transitions with overlapping firing intervals may not fire concurrently. For example, in Fig. 3, $t_2, t_5 \in Fr(s_0)$ and $t_2||t_5$. Their firing intervals are overlapping, i.e., $f_0(t_2) = [2, 5]$ and $f_0(t_5) = [4, 7]$. But their efficient firing intervals are not overlapping, i.e., $\Theta_0(t_2) = [2, 3]$ and $\Theta_0(t_5) = [4, 5]$. Thus they cannot fire in an interleaving way from s_0.

C. TIMED LANGUAGE ACCEPTANCE

In order to compare scheduling scopes of the four semantics models, we demonstrate their expressive powers in terms of timed language acceptance [23], [24].

A timed word w over T is a finite or infinite sequence $w = (t_0, \theta_0)(t_1, \theta_1)\ldots(t_n, \theta_n)\ldots$, such that for each $i \geq 0$, $t_i \in T$, $\theta_i \in R^+$ and $\theta_{i+1} \geq \theta_i$. Note that θ in timed word w is the absolute (global) firing time of transition t, while d in $s \xrightarrow{d} s'$ denotes the relative time elapsing from state s.

If a run ρ has the following form $\rho = s_0 \xrightarrow{t_0,d_0} s_1 \xrightarrow{t_1,d_1} \cdots s_n \xrightarrow{t_n,d_n} s_{n+1} \cdots$, then the timed word $w(\rho)$ can be computed by $w(\rho) = (t_0, d_0)(t_1, d_0 + d_1)\cdots(t_n, d_0 + d_1 + \ldots + d_n)\cdots$.

In Fig. 1, Timed word $(t_1, 1)(t_3, 4)(t_5, 5)(t_7, 6)(t_8, 7)$ can be accepted by all four semantics models. Timed word $(t_2, 3)(t_3, 4)(t_5, 5)(t_8, 6)$ is accepted by M-TPN, RM-TPN and W-TPN, but not by S-TPN. Timed word $(t_3, 4)(t_2, 4)(t_5, 5)(t_8, 8)$ is accepted by RM-TPN and W-TPN, but not by S-TPN and M-TPN. Timed word $(t_6, 7)$ is accepted by W-TPN but not by the others.

A timed language L accepted by a TPN model is a set of all timed words accepted by the model. Let L_A denote timed language L accepted by a model A. We say that a language L_A is a subclass of a language L_B, denoted by $L_A \subseteq L_B$, if any timed word accepted by A is also accepted by B.

Property 5: For a TPN, $L_{\text{S-TPN}} \subseteq L_{\text{M-TPN}} \subseteq L_{\text{RM-TPN}} \subseteq L_{\text{W-TPN}}$.

Proof: (1) $L_{\text{M-TPN}} \subseteq L_{\text{RM-TPN}}$.

For a TPN, its M-TPN model and RM-TPN model have the same initial state $s_0 = (M_0, f_0)$. Thus, we only need to prove that if $s \rightarrow_m s'$ then $s \rightarrow_r s'$.

Let $S = En(M - B(t))$. If $s \xrightarrow{t,d}_m s'$, then $\max(0, \downarrow f(t)) \leq d \leq \min\{\uparrow f(t), \underline{\uparrow} S\}$ by the formal semantics of M-TPN. From the rewrited firability definition of RM-TPN, it follows that $t \in Fr(s)$ for RM-TPN. Since $\Gamma^r(t) = \max\{\underline{\uparrow} U | U \in \mathbb{U}(M, t)\} \geq \min\{\uparrow f(t), \underline{\uparrow} S\} = \Gamma^m(t)$, we obtain that $\max(0, \downarrow f(t)) \leq d \leq \Gamma^r(t)$. From the semantics of RM-TPN, it follows that $s \xrightarrow{t,d}_r s'$.

Hence, if there is a run $\rho_1 = s_0 \xrightarrow{t_0,d_0}_m s_1 \xrightarrow{t_1,d_1}_m \cdots s_n$ in M-TPN, then there must be a run $\rho_2 = s_0 \xrightarrow{t_0,d_0}_r s_1 \xrightarrow{t_1,d_1}_r \cdots s_n$ in RM-TPN. According to the definition of timed word, we have that $w(\rho_1) = w(\rho_2)$. That is to say, if $w(\rho_1) \in L_{\text{M-TPN}}$, then $w(.\rho_2) \in L_{\text{RM-TPN}}$. Therefore, it follows that $L_{\text{M-TPN}} \subseteq L_{\text{RM-TPN}}$.

The proofs of $L_{\text{S-TPN}} \subseteq L_{\text{M-TPN}}$ and $L_{\text{RM-TPN}} \subseteq L_{\text{W-TPN}}$ are similar to that of (1) and thus omitted.

The property shows that the scheduling scope of an RM-TPN is between that of M-TPN and W-TPN. From the definitions of efficient firing time bound of a transition, we can find that the order of time bounds at a state for four semantics models is $\Gamma^s(t) \leq \Gamma^m(t) \leq \Gamma^r(t) \leq \Gamma^w(t)$. Thus the number of feasible firable transitions at a state for an RM-TPN is more than that for M-TPN and S-TPN.

On the other hand, from the definition of transition firability of RM-TPNs, we know that the firing of a transition can preserve at least a progressive maximal concurrent set. The property keeps the consistency in non-emptiness of the enabled and firable transition sets for RM-TPN, and ensures that the model does not produce overdue enabled transition set.

D. TURING EQUIVALENCE

It is known that time Petri nets with strong semantics that can perform zero testing is equivalent to a Turing machine [25]. In [13], timed Petri net models (with weak semantics) are proved to be not Turing powerful because they cannot simulate a counter with test on zero. Next, we show that M1-TPNs and RM-TPNs are expressive enough to model Two Counter Machines (2CM), which have been proved to be Turing equivalent.

A 2-counter machine (2CM), with nonnegative counters c_1 and c_2, is a sequence of labeled instructions

$$0 : Instr_0; 1 : Instr_1; \ldots; n : Instr_n$$

where $Instr_n$ is a Halt-command and $Instr_i$, $i = 0, 1, \ldots, n$-1, has one of the two following forms (assuming $0 \leq k, k_1, k_2 \leq n$ and $1 \leq j \leq 2$)

1. $c_j := c_j + 1$; goto k
2. if $c_j = 0$ then goto k_1 else ($c_j := c_j - 1$; goto k_2)

A configuration of 2CM is a tuple $C = (l, v_1, v_2)$ where

1. l is the value of the program counter (instruction pointer); and
2. v_1 and v_2 are the values of counters c_1 and c_2.

A computation of 2CM is a finite or infinite sequence of configurations.

Theorem 1: RM-TPNs can simulate 2-counter machines.

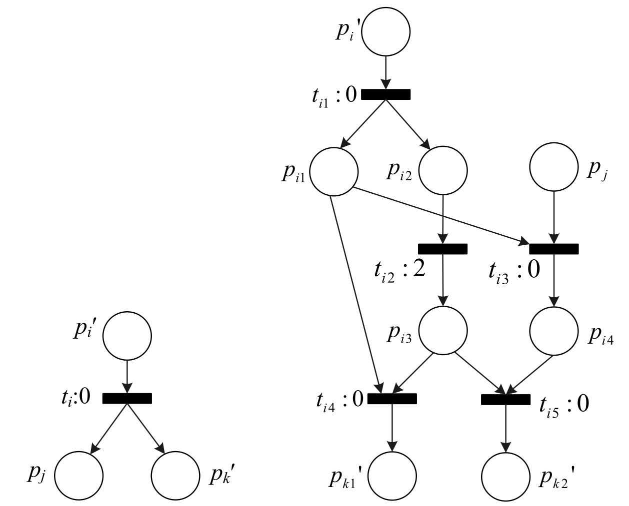
Proof: We construct an RM-TPN simulating a 2-counter machine (2CM) as follows.

(i) Adding places p_1, p_2 (representing two counters) and places $p'_0, p'_1, \ldots, p'_n$ (representing the instruction pointers of the program counter) into the place set P.

(ii) If $Instr_i$ is "$c_j := c_j+1$; goto k". Then adding transition t_i into the transition set T such that $B(p'_i, t_i) = F(p'_k, t_i) = F(p_j, t_i) = 1$, and $SI(t_i) = 0$.

(iii) If $Instr_i$ is "if $c_j = 0$ then goto k_1 else ($c_j := c_j - 1$; goto k_2)", then adding transitions $t_{i1}, t_{i2}, t_{i3}, t_{i4}, t_{i5}$ and places $p_{i1}, p_{i2}, p_{i3}, p_{i4}$ into M2-TPN such that

$$B(p'_i, t_{i1}) = F(p_{i1}, t_{i1}) = F(p_{i2}, t_{i1}) = 1,$$
$$B(p_{i2}, t_{i2}) = F(p_{i3}, t_{i2}) = 1,$$
$$B(p_{i1}, t_{i3}) = B(p_j, t_{i3}) = F(p_{i4}, t_{i3}) = 1,$$
$$B(p_{i1}, t_{i4}) = B(p_{i3}, t_{i4}) = F(p'_{k1}, t_{i4}) = 1,$$
$$B(p_{i3}, t_{i5}) = B(p_{i4}, t_{i5}) = F(p'_{k2}, t_{i5}) = 1,$$
$$SI(t_{i1}) = SI(t_{i3}) = SI(t_{i4}) = SI(t_{i5}) = 0 \text{ and } SI(t_{i2}) = 2.$$



(a) simulation of an increment (b) simulation of a zero-test

FIGURE 4. Representing a 2CM by an RM-TPN.

This construction is schematized in Fig. 4.

For any input counter values v_1 and v_2, we put v_1 and v_2 tokens in places p_1 and p_2 respectively, one token in p'_1, and 0 tokens elsewhere in RM-TPN.

Now, we prove that a run of RM-TPN can simulate a computation of 2CM. The proof is carried out by induction on h.

For the basis case ($h = 0$), we let $M_0(p_1) = v_1^0$, $M_0(p_2) = v_2^0$, $M_0(p'_1) = 1$, and $M_0(p) = 0$ for any other place p. Hence M_0 can simulate the initial configuration (l_0, v_1^0, v_2^0) with $l_0 = 0$.

Assume that the assertion holds for $h \leq r$. Consider $h = r + 1$.

By induction hypothesis, we know that the run $M_0 \to M_1 \ldots, \to M_r$ simulates the computation (l_0, v_1^0, v_2^0), $(l_1, v_1^1, v_2^1), \ldots, (l_r, v_1^r, v_2^r)$ such that $M_r(p_1) = v_1^r$, $M_r(p_2) = v_2^r$, $M_r(p'_{l_r}) = 1$ and $M_r(p) = 0$ for any other place p. We consider two cases:

Case 1: If $Instr_{l_r}$ is "$c_j := c_j + 1$; goto k". Then t_{l_r} can fire from M_r as shown in Fig. 4a. Firing t_{l_r} leads to marking M_{r+1} such that $M_{r+1}(p_j) = v_j^r + 1$, $M_{r+1}(p_{j\oplus 1}) = v_{j\oplus 1}^r$, $M_{r+1}(p'_k) = 1$ and $M_{r+1}(p) = 0$ for any other place p, where $j \in \{1, 2\}$ and $j \oplus 1 = (j \bmod 2) + 1$. Hence the $(r + 1)$th configuration is $(l_{r+1}, v_1^{r+1}, v_2^{r+1})$ with $l_{r+1} = k$, $v_j^{r+1} = v_j^r + 1$ and $v_{j\oplus 1}^{r+1} = v_{j\oplus 1}^r$.

Case 2: If $Instr_{l_r}$ is "if $c_j = 0$ then goto k_1 else ($c_j := c_j - 1$; goto k_2)". Then this case is divided into two subcases:

(i) If $c_j = 0$, then the transition sequence $t_{l_r1}t_{l_r2}t_{l_r4}$ can be fired from M_r (Fig. 4b). Firing the sequence takes 2 time units and reaches M_{r+1} with $M_{r+1}(p_j) = v_j^r = 0$, $M_{r+1}(p_{j\oplus 1}) = v_{j\oplus 1}^r$, $M_{r+1}(p'_{k_1}) = 1$ and $M_{r+1}(p) = 0$ for any other place p. Hence the $(r + 1)$th configuration is $(l_{r+1}, v_1^{r+1}, v_2^{r+1})$ where $l_{r+1} = k_1$, $v_j^{r+1} = v_j^r = 0$ and $v_{j\oplus 1}^{r+1} = v_{j\oplus 1}^r$.

(ii) If $c_j > 0$, then the transition sequence that can be fired from M_r is $t_{l_r1}\ t_{l_r3}\ t_{l_r2}\ t_{l_r5}$ (Fig. 4b). Firing the sequence takes 2 time units and reaches marking M_{r+1} with $M_{r+1}(p_j) = v_j^r - 1$, $M_{r+1}(p_{j\oplus 1}) = v_{j\oplus 1}^r$, $M_{r+1}(p'_{k_2}) = 1$ and $M_{r+1}(p) = 0$ for any other place p. Hence the $(r + 1)$th configuration is (l_{r+1}, v_1^r, v_2^r) where $l_{r+1} = k_2$, $v_j^{r+1} = v_j^r - 1$ and $v_{j\oplus 1}^{r+1} = v_{j\oplus 1}^r$.

Therefore, a RM-TPN can simulate a 2-counter machine.

Theorem 1 shows that RM-TPN has the power of a Turing machine. This is because the model has timing enforcement on minimal time upper bounds of maximal concurrent sets.

IV. STATE CLASS METHOD

State class methods are a fundamental and most widely applied technique for schedulability analysis and timing verification of TPNs [8]–[10], [26]–[29]. In this subsection, we propose a state class method for M2-TPNs.

A state class of a TPN is a pair $C = (M, D)$, where

1. M is a marking;
2. D: $\dot{T} \times \dot{T} \to \mathbb{I}$ is a firing domain. For $t_i, t_j \in \dot{T}$, $D(t_i, t_j)$ represents the firing interval of t_j relative to t_i, where $\dot{T} = T \cup \{t_0, t_*\}$, t_* is the generative transition of C, and t_0 is a special reference transition, called zero transition, which global firing time is 0 (i.e. $SI(t_0) = 0$).

If firing transition t directly leads to state class C, we say that t is a generative transition of C. t_0 can be viewed as the generative transition of the initial state class C_0.

It is easy to find that $\uparrow D(t_i, t_j) = - \downarrow D(t_j, t_i)$. The initial state class $C_0 = (M_0, D_0)$ is defined as follows:

1. M_0 is the initial marking; and
2. $\forall t_i, t_j \in En(M_0) \cup \{t_0\}$, if $t_i \neq t_j$, $D_0(t_i, t_j) = SI(t_j) - SI(t_i)$; otherwise $D_0(t_i, t_j) = 0$.

Firability: Transition $t_f \in En(M)$ is *firable* at state class $C = (M, D)$, if

1. $\uparrow D(t_*, t_f) \geq 0$; and
2. $\exists U \in \mathbb{U}(M, t_f)$, such that $\forall t_i \in U, \uparrow D(t_f, t_i) \geq 0$.

For RM-TPNs, a progressive enabled transition is firable if it can fire before the other transitions in a maximal concurrent transition set containing itself.

According to Property 1, firable condition 2 of RM-TPNs can be rewritten as:

$$En(M - \sum\nolimits_{t \in S \cup \{t_f\}} B(t)) = \emptyset,$$

where $S = \{t | t \in En(M - B(t_f)) \wedge \uparrow D(t_f, t) \geq 0\}$.

Let $Fr(C)$ be the set of all firable transitions at state class C.

Firing rules: In RM-TPNs, state class $C_{k+1} = (M_{k+1}, D_{k+1})$ reached from $C_k = (M_k, D_k)$ by firing $t_f \in Fr(C_k)$ is computed as follows:

1. Marking M_{k+1}: $M_{k+1} = M_k - B(t_f) + F(t_f)$
2. Firing domain D_{k+1}:

Let $t_* = t_f, \forall t_i, t_j \in En(M_{k+1}) \cup \{t_0\}$,

(i) $D_{k+1}(t_i, t_i) = 0$

(ii) $D_{k+1}(t_*, t_i) = \begin{cases} SI(t_i) \text{ if } t_i \in New(M_k, t_f) \\ [\overline{\downarrow} D_k(t_i, t_f), \uparrow D_k(t_f, t_i)] \text{ } otherwise \end{cases}$

where $\overline{\downarrow} D_k(t_i, t_f) = \max\{\downarrow D_k(t_f, t_i), \downarrow D_k(t, t_i) | \ t \in En(M_k - B(t_f)) \wedge \uparrow D_k(t_f, t) \geq 0\}$.

(iii) $D_{k+1}(t_i, t_j)$

$= \begin{cases} D_{k+1}(t_*, t_j) - D_{k+1}(t_*, t_i) \text{ if } t_i \text{ or } t_j \in New(M_k, t_f) \\ D_k(t_i, t_j) \cap (D_{k+1}(t_i, t_*) + D_{k+1}(t_*, t_j)) \text{ } otherwise \end{cases}$

Firing rule 2(ii) computes the firing interval of t_i relative to firing t_f, i.e., the firing interval of t_i relative to C_{k+1}, where $\overline{\downarrow} D_k(t_i, t_f)$ indicates the maximal lower bound among firing intervals of all persistently progressive enabled transitions relative to t_f. Rule 2(iii) computes the firing interval of t_j relative to t_i at C_{k+1} according to the results of Rule 2(ii).

A correctness proof of the firing rules is showed in Appendix.

Next, we take schedule $t_3t_5t_2t_8$ in Fig. 1 as an example to illustrate the proposed state class method.

The initial state class is $C_0 = (M_0, D_0)$ where

$$M_0 = (1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0),$$

$$D_0 = \begin{matrix} t_0 \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{matrix} \begin{pmatrix} 0 & [1,2] & [3,4] & [4,5] & [6,6] & [3,5] & [6,7] \\ & 0 & [1,3] & [2,4] & [4,5] & [1,4] & [4,6] \\ & & 0 & [0,2] & [2,3] & [-1,2] & [2,4] \\ & & & 0 & [1,2] & [-2,1] & [1,3] \\ & & & & 0 & [-3,-1] & [0,1] \\ & & & & & 0 & [1,4] \\ & & & & & & 0 \end{pmatrix}.$$

At C_0, t_2 is firable because $En(M - \sum_{t \in S \cup \{t_2\}} B(t)) = \emptyset$ with $S = \{t_3, t_4, t_5, t_6\}$. t_4 is not because $En(M - \sum_{t \in S \cup \{t_4\}} B(t)) \neq \emptyset$ where $S = \{t_6\}$. In the same way, we obtain that $Fr(C_0) = \{t_1, t_2, t_3, t_5\}$.

Firing t_3 reaches $C_1 = (M_1, D_1)$ where

$$M_1 = (10100100),$$

$$D_1 = \begin{matrix} t_0 \\ t_*(t_3) \\ t_2 \\ t_5 \\ t_6 \end{matrix} \begin{pmatrix} 0 & [4,4] & [4,4] & [4,5] & [6,7] \\ & 0 & [0,0] & [0,1] & [2,3] \\ & & 0 & [0,1] & [2,3] \\ & & & 0 & [1,3] \\ & & & & 0 \end{pmatrix}.$$

By the transition firability, $Fr(C_1) = \{t_2, t_5\}$. At C_1, firing t_5 leads to $C_2 = (M_2, D_2)$ such that

$$M_2 = (10000110),$$

$$D_2 = \begin{matrix} t_0 \\ t_*(t_5) \\ t_2 \end{matrix} \begin{pmatrix} 0 & [4,4] & [4,4] \\ & 0 & [0,0] \\ & & 0 \end{pmatrix}.$$

By the transition firability, $Fr(C_2) = \{t_2\}$. Firing t_2 leads to $C_3 = (M_3, D_3)$ where

$$M_3 = (00001110),$$

$$D_3 = \begin{matrix} t_0 \\ t_*(t_2) \\ t_8 \end{matrix} \begin{pmatrix} 0 & [4,4] & [7,7] \\ & 0 & [3,3] \\ & & 0 \end{pmatrix}.$$

$Fr(C_3) = \{t_8\}$. Firing t_8 leads to $C_4 = (M_4, D_4)$ where

$$M_4 = (00000001),$$

$$D_4 = \begin{matrix} t_0 \\ t_*(t_8) \end{matrix} \begin{pmatrix} 0 & [7,7] \\ & 0 \end{pmatrix}.$$

$En(M_4) = \emptyset$ and $Fr(C_4) = \emptyset$. Thus, the firing interval of schedule $t_3t_5t_2t_8$ is [7,7]. We generate a reachability class tree of RM-TPN in Fig. 5.

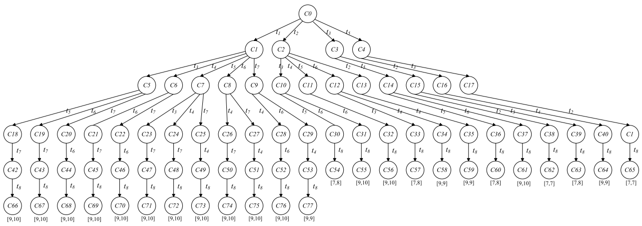


FIGURE 5. Reachability tree of the TPN in Fig. 1 for RM-TPN.

V. SCHEDULING PROBLEM WITH DUE WINDOWS

Scheduling problems with due windows have their origins in Just-in-Time (JIT) philosophy. Usually, JIT scheduling models assume an existence of job due-dates and penalize both early and tardy jobs. However, in manufacturing industry it is often expected that the jobs are finished in a certain time interval (due window) rather than at a single point in time (due date) [5]–[7], [30], [31].

Consider a scheduling problem with distinct due windows, under the assumptions that a machine can process at most one job at a time and all jobs are ready for processing at time $t = 0$. For convenience, we list the notations used throughout the paper as follows:

J_i: the ith job;

m_k: the kth machine;

O_{ij}: the processing operation of job i on machine j;

Π : the set of all schedules;

$\pi \in \Pi$: a schedule;

d_j: the due window of job j, where $\downarrow d_j$ is the earliest due date and $\uparrow d_j$ is the latest due date;

$c(\pi)$: the interval completion time of schedule π;

$c_j(\pi)$: the interval completion time of job j in schedule π;

$E_j(\pi) = [\max\{0, \downarrow d_j - \uparrow c_j(\pi)\}, \max\{0, \downarrow d_j - \downarrow c_j(\pi)\}]$: the interval earliness of job j in schedule π;

$T_j(\pi) = [\max\{0, \downarrow c_j(\pi) - \uparrow d_j\}, \max\{0, \uparrow c_j(\pi) - \uparrow d_j\}]$: the interval tardiness of job j in schedule π;

α_j: the unit earliness penalty for job j;

β_j: the unit tardiness penalty for job j;

$g_j(\pi) = \alpha_j^* E_j(\pi) + \beta_j^* T_j(\pi)$: the penalty interval of job j in schedule π.

TABLE 2. A scheduling problem with 4 jobs and 4 machines.

JOB	PROCESSING ROUTES AND TIME INTERVALS 1	2	3	4
1	m3 [54,54]	m1 [34,40]	m4 [61,63]	m2 [2,6]
2	m4 [9,13]	m1 [15,15]	m2 [89,89]	m3 [70,73]
3	m1 [38,41]	m2 [19,20]	m3 [28,33]	m4 [87,87]
4	m1 [95,99]	m3 [34,37]	m2 [7,9]	m4 [29,33]

$G(\pi) = \sum_i^n g_i(\pi)$: the total penalty interval of schedule π.

We consider a scheduling problem with four jobs and three machines in [32], which is described in Table 2. Each job has four operations. The data on each operation is composed of two parts: the first is the machine that processes the operation and the second is interval processing time. Let O_{ij} represent the operation of Job J_i on machine m_j, where $i = 1, 2, 3, 4$ and $j = 1, 2, 3, 4$. Processing time interval of operation O_{ij} is $[o_{ij}, o_{ij} + r_{ij}]$, where o_{ij} is the processing time of the corresponding operation in [32] and random integer $r_{ij} \in [0, 6]$.

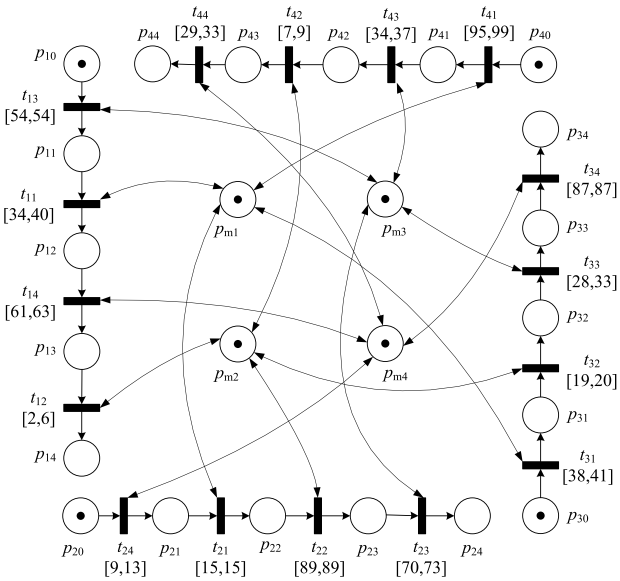


FIGURE 6. The TPN model of the scheduling problem.

According to the description of the problem, we build the TPN model with the initial marking represented in Fig. 6. Transition t_{ij} represents operation O_{ij}. Place p_{mk} stands for machine m_k $(k = 1, 2, 3, 4)$. Place p_{i0} represents the initial status of job J_i and place p_{ij} the completion status of the jth operation process.

Next, we evaluate makespan and total penalty by using the state class methods of the proposed models, which are important performance indexes for scheduling problems with due windows.

Let $\pi^\#$ be the optimal schedule with the minimal makespan. As shown in Table 3, S-TPN produces 2 feasible schedules, where $\pi^\# = O_{24}O_{21}O_{13}O_{31}O_{32}O_{11}O_{33}O_{14}O_{22}$ $O_{41}O_{43}O_{42}O_{34}O_{44}$ O_{23} and $c(\pi^\#) = [300, 318]$.

TABLE 3. Comparison of schedulability analysis for four semantics models.

	S-TPN	M-TPN	RM-TPN	W-TPN
Number of all schedules	2	51	1512	80876
Optimal schedule $\pi^\#$	$O_{24}O_{21}O_{13}O_{31}$ $O_{32}O_{11}O_{33}O_{14}$ $O_{22}O_{12}O_{41}O_{43}$ $O_{42}O_{34}O_{44}O_{23}$	$O_{24}O_{31}O_{21}O_{13}$ $O_{32}O_{11}O_{33}O_{22}$ $O_{14}O_{12}O_{41}O_{23}$ $O_{34}O_{43}O_{42}O_{44}$	$O_{24}O_{31}O_{21}O_{13}$ $O_{32}O_{33}O_{22}O_{41}$ $O_{34}O_{11}O_{43}O_{42}$ $O_{14}O_{12}O_{23}O_{44}$	$O_{24}O_{13}O_{41}$ $O_{43}O_{42}O_{44}$
completi on time $c(\pi^\#)$	[300,318]	[286,300]	[272,286]	[165,178]
total penalty $G(\pi^\#)$	[80.5,103.7]	[64.6,80.3]	[0,0]	/
Run time	0.014 seconds	0.425 seconds	46.853 seconds	221.387 seconds

M-TPN generates 51 feasible schedules, where $\pi^\# = O_{24}O_{31}O_{21}O_{13}O_{32}O_{11}O_{33}O_{22}O_{14}O_{12}O_{41}$ $O_{23}O_{34}O_{43}O_{42}O_{44}$ and $c(\pi^\#) = [286, 300]$. RM-TPN produces 1508 feasible schedules, in which $\pi^\# = O_{24}O_{31}O_{21}O_{13}O_{32}O_{33}$ O_{22} $O_{41}O_{34}O_{11}O_{43}O_{42}O_{14}O_{12}O_{23}O_{44}$ and $c(\pi^\#) = [272, 286]$.

W-TPN generates 80876 schedules, where the schedule with minimal firing interval is $O_{24}O_{13}O_{41}O_{43}O_{42}O_{44}$ and its firing interval is [165,178]. Obviously, this is an invalid schedule because jobs J_1, J_2 and J_3 are not completed. If we remove the invalid schedules from W-TPN, we can obtain the same optimal solution as RM-TPN.

Assume that the due dates of four jobs are $d_1 = [240, 260]$, $d_2 = [250, 270]$, $d_3 = [170, 190]$ and $d_4 = [270, 290]$, and that the earliness penalty weight $\alpha_j = 0.3$ and the tardiness penalty weight $\beta_j = 0.7$.

For RM-TPN, $c_1(\pi^\#) = [245, 259]$, $c_2(\pi^\#) = [252, 263]$, $c_3(\pi^\#) = [172, 179]$ and $c_4(\pi^\#) = [272, 286]$. $E_1(\pi^\#) = [\max\{0, 240 - 259\}, \max\{0, 240 - 245\}] = [0, 0]$ and $T_1(\pi^\#) = [\max\{0, 245 - 260\}, \max\{0, 259 - 260\}] = [0, 0]$, and thus $g_1(\pi^\#) = 0.3^*E_1(\pi^\#) + 0.7^*T_1(\pi^\#) = [0, 0]$. Similarly, $g_2(\pi^\#) = [0, 0]$, $g_3(\pi^\#) = [0, 0]$ and $g_4(\pi^\#) = [0, 0]$. Therefore, the total penalty interval of schedule $\pi^\#$ is $G(\pi^\#) = \sum_i^n g_i(\pi^\#) = [0, 0]$.

In the same way, we can obtain total penalties of optimal schedules in S-TPN and M-TPN, as shown in Table 3. For W-TPN, the listed schedule $O_{24}O_{13}O_{41}O_{43}O_{42}O_{44}$ does not finish jobs J_1, J_2 and J_3, thus we cannot compute its total penalty. If these invalid schedules are removed from W-TPN, the optimal schedule is the same as that of RM-TPN.

Therefore, we obtain the optimal schedule of the scheduling problem with the minimal makespan and total penalty, which is $\pi^\# = O_{24}O_{31}O_{21}O_{13}O_{32}O_{33}O_{22}O_{41}O_{34}O_{11}O_{43}$ $O_{42}O_{14}O_{12}O_{23}O_{44}$. The Gantt chart of the schedule is shown in Fig. 7.

As shown in Table 3, the computation time of feasible schedules is in positive correlation with the size of the state

TABLE 4. Summary of four semantics models.

	S-TPN	M-TPN	RM-TPN	W-TPN
Firability of transition t	depends on all enabled transitions	depends on itself and its non-conflicting transitions	depends on maximal concurrent sets containing it	itself
Efficient time upper bound of transition t	$\uparrow En(M)$	$\min\{\uparrow f(t), \uparrow En(M-B(t))\}$	$\max\{\uparrow U \mid U \in \mathbb{U}(M,t)\}$	$\uparrow f(t)$
Overdue enabled transitions	None	Maybe	Maybe	Maybe
Incomplete schedules	None	None	None	Maybe
Scheduling scope	$L_{\text{S-TPN}} \subseteq L_{\text{M-TPN}} \subseteq L_{\text{RM-TPN}} \subseteq L_{\text{W-TPN}}$			
Turing equivalence	Yes	Yes	Yes	No

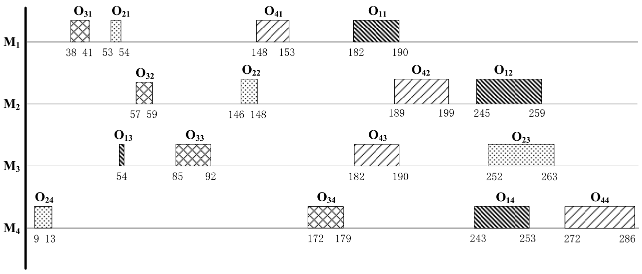


FIGURE 7. The optimal schedule of the problem in Fig. 6.

spaces produced by these semantics models. The size of the state space of RM-TPN is between that of M-TPN and W-TPN. However, the state space of W-TPN includes a lot of invalid schedules. But removing these invalid schedules needs plenty of time. Therefore, from the state space of RM-TPN, we can usually obtain better scheduling solutions than the other semantics models.

The comparison of four semantics models is summarized in Table 4. All the experiments are performed on Dell Latitude E6430 laptop with a 2.60GHz Intel i7-3720QM processor, 8GB Memory and 64-bit Windows 7 operating system.

VI. CONCLUSION

We have presented a relaxed mixed semantics model for schedulability analysis and timing verification of time Petri nets. Our model can preserve particular properties, which are not satisfied in the existing models, by relaxing the firable conditions of transitions and keeping strong timing on minimal time upper bounds of maximal concurrent sets.

In this model, the firability of a transition depends on maximal concurrent sets containing the transition. The treatment not only extends the scheduling scope of TPN models, but also avoids the generation of overdue enabled transition sets. Invalid schedules that cause overdue enabled transition sets are avoided in our model. Therefore, by using the model, we may obtain better feasible solutions than the existing models.

Benefiting from the preservation of strong timing, a RM-TPN is proved to be Turing equivalent. The comparison of four semantics models w.r.t. timed language acceptance reflects the difference in scheduling scopes of these models. We further develop a state class method for the schedulability analysis of the model.

The analysis tool for RM-TPNs has been developed based on a Matlab platform. The tool is available at http://sice.hnist.cn/info/1066/1491.htm. Future work intends to improve our analysis tool and apply it to the modeling and schedulability analysis of complex systems [33]–[35].

APPENDIX

In order to prove the correctness of the firing rules, we introduce a concept of the Difference Bound Matrix (DBM) [36], [37]. DBM is a practical matrix representation of time constraints ($x - y \leq c$). Let $Var = \{x_0, x_*, x_1, \ldots, x_s\}$ be a finite set of firing time variables of transitions with its elements in $R^+ \cup \{\infty\}$, where $x_0 \equiv 0$, and x_* is firing time variable of transition t_*.

For any $t_i, t_j \in En(M) \cup \{t_0, t_*\}$, firing interval $D(t_i, t_j)$ can be represented by two time constraints $x_j - x_i \leq \uparrow D(t_i, t_j)$ and $x_i - x_j \leq - \downarrow D(t_i, t_j)$. Let $B(D)$ be the DBM representing a firing domain D. Its element b_{ij} can be computed in three steps:

1. $\forall t_i, t_j \in En(M) \cup \{t_0, t_*\}$, let $b_{ij} = \uparrow D(t_i, t_j)$ and $b_{ji} = - \downarrow D(t_i, t_j)$;
2. If $t_i, t_j \notin En(M) \cup \{t_0, t_*\}$, let $b_{ij} = \infty$; and
3. Add an implicit constraint $x_i - x_i \leq 0$, and let $b_{ii} = 0$.

The closure of a time constraint set computes its tightest time representation. The closure provides a canonical representation, as two closed firing domains describe the same solution set precisely when they are identical [37]. Thus, a DBM is normal if its time constraints have been tightened by the Floyd's shortest path algorithm [26], [37].

The proof of the following theorem is inspired from the normalization proof in [10], [26].

Theorem 2: In RM-TPNs, if $C_n = (M_n, D_n)$ is a reachable state class from the initial state class $C_0 = (M_0, D_0)$, then $B(D_n)$ is normal.

Proof: The proof is carried out by induction on n.

The base case ($n = 0$). Let $t_i, t_j, t_k \in En(M_0) \cup \{t_0\}$. According to the definition of the initial state class C_0, we have that $D_0(t_i, t_j) = SI(t_j) - SI(t_i)$, i.e., $b^0_{ij} = \uparrow SI(t_j) - \downarrow SI(t_i)$ in $B(D_0)$. Then, we obtain that $b^0_{ik} + b^0_{kj} = \uparrow SI(t_k) - \downarrow SI(t_i) + \uparrow SI(t_j) - \downarrow SI(t_k) = b^0_{ij} + \uparrow SI(t_k) - \downarrow SI(t_k) \geq b^0_{ij}$. Thus b^0_{ij} is not replaced by $b^0_{ik} + b^0_{kj}$ during the execution of the Floyd's shortest path algorithm, and $B(D_0)$ is identical to its shortest path closure. Therefore, $B(D_0)$ is normal.

Assume that the assertion holds for $n \leq h$. Consider $n = h + 1$.

Suppose that $t_f \in F(C_h)$. By the inductive assumption, $B(D_h)$ is normal.

For any $t_i \in En(M_{h+1}) \cup \{t_0\}$, if $t_i \in New(M_h, t_f)$, then its firing interval should equal $x_f + SI(t_r)$, where x_f is t_f's firing time. Thus $b^{h+1}_{*i} = \max\{x_i - x_f\} = \uparrow SI(t_i)$ and $b^{h+1}_{i*} = \max\{x_f - x_i\} = -\downarrow SI(t_i)$, and therefore, $D_{h+1}(t_*, t_i) = [-b^{h+1}_{i*}, b^{h+1}_{*i}] = SI$.

If $t_i \in En(M_h - B(t_f)) \cup \{t_0\}$, then the earliest firing time of each transition t_i that remains enabled should be postponed to x_f, but its latest firing time remains the same. It follows that $b^{h+1}_{*i} = \max\{x_i - x_f\} = b^h_{fi} = \uparrow D_h(t_f, t_i)$. According to the definition of transition firability, the firing time of t_f cannot overtake the minimum of the latest firing time of its non-conflicting transitions. Thus $b^{h+1}_{i*} = \min\{\max\{x_f - x_i\}, \max\{x_l - x_i | l \neq i, f\}\} = \min\{b^h_{if}, b^h_{il}\} = \min\{\uparrow D_h(t_i, t_f), \uparrow D_h(t_i, t_l)\} = \min\{- \downarrow D_h(t_f, t_i), - \downarrow D_h(t_l, t_i)\} = -\max\{\downarrow D_h(t_f, t_i), \downarrow D_h(t_l, t_i)\}$. Therefore, $D_{h+1}(t_*, t_i) = [-b^{h+1}_{i*}, b^{h+1}_{*i}] = [\max\{\downarrow D_h(t_f, t_i), \downarrow D_h(t_l, t_i)\}, \uparrow D_h(t_f, t_i)]$.

For any $t_i, t_j \in En(M_{h+1}) \cup \{t_0\}$, if t_i or $t_j \in New(M_h, t_f)$, then $b^{h+1}_{ij} = \max\{x_j - x_i\} = \max\{x_j - x_f\} + \max\{x_f - x_i\} = b^{h+1}_{*j} + b^{h+1}_{i*}$ and $b^{h+1}_{ji} = b^{h+1}_{*i} + b^{h+1}_{j*}$. Therefore $D_{k+1}(t_i, t_j) = [-b^{h+1}_{ji}, b^{h+1}_{ij}] = [-b^{h+1}_{*i} - b^{h+1}_{j*}, b^{h+1}_{*j} + b^{h+1}_{i*}] = [\downarrow D_{k+1}(t_*, t_j) - \uparrow D_{k+1}(t_*, t_i), \uparrow D_{k+1}(t_*, t_j) - \downarrow D_{k+1}(t_*, t_i)] = D_{k+1}(t_*, t_j) - D_{k+1}(t_*, t_i)$.

If $t_i, t_j \in En(M_h - B(t_f)) \cup \{t_0\}$, then $b^{h+1}_{ij} = \min\{b^h_{ij}, b^{h+1}_{i*} + b^{h+1}_{*j}\} = \min\{\uparrow D_h(t_i, t_j), \uparrow D_{h+1}(t_i, t_*) + \uparrow D_{h+1}(t_*, t_j)\}$ and $-b^{h+1}_{ji} = -\min\{\uparrow D_h(t_j, t_i), \uparrow D_{h+1}(t_j, t_*) + \uparrow D_{h+1}(t_*, t_i)\} = \max\{\downarrow D_h(t_i, t_j), \downarrow D_{h+1}(t_i, t_*) + \downarrow D_{h+1}(t_*, t_j)\}$. Thus, $D_{h+1}(t_i, t_j) = [-b^{h+1}_{ji}, b^{h+1}_{ij}] = D_h(t_i, t_j) \cap (D_{h+1}(t_i, t_*) + D_{h+1}(t_*, t_j))$.

For any $t_i, t_j, t_k \in En(M_{h+1}) \cup \{t_0\}$, if t_i or $t_j \in New(M_h, t_f)$ (assume that $t_j \in New(M_h, t_f)$), we face two cases:

1) If $t_k \in New(M_h, t_f)$, then $b^{h+1}_{ij} = b^{h+1}_{i*} + b^{h+1}_{*j}$, $b^{h+1}_{ik} = b^{h+1}_{i*} + b^{h+1}_{*k}$ and $b^{h+1}_{kj} = b^{h+1}_{*j} + b^{h+1}_{k*}$. It follows that $b^{h+1}_{ik} + b^{h+1}_{kj} = b^{h+1}_{i*} + b^{h+1}_{*k} + b^{h+1}_{*j} + b^{h+1}_{k*} = b^h_{fk} + \min\{b^h_{kf}, b^h_{kl}\} + b^{h+1}_{i*} + b^{h+1}_{*j} = \min\{b^h_{fk} + b^h_{kf}, b^h_{fk} + b^h_{kl}\} + b^{h+1}_{i*} + b^{h+1}_{*j}$. By the inductive assumption and the transition firability, we obtain that $\min\{b^h_{fk} + b^h_{kf}, b^h_{fk} + b^h_{kl}\} \geq 0$. Thus, $b^{h+1}_{ik} + b^{h+1}_{kj} \geq b^{h+1}_{i*} + b^{h+1}_{*j} = b^{h+1}_{ij}$.

2) If $t_k \in En(M_h - B(t_{h+1})) \cup \{t_0\}$, then $b^{h+1}_{ij} = b^{h+1}_{i*} + b^{h+1}_{*j}$, $b^{h+1}_{ik} = \min\{b^h_{ik}, b^{h+1}_{*k} + b^{h+1}_{i*}\}$ and $b^{h+1}_{kj} = b^{h+1}_{*j} + b^{h+1}_{k*}$. It follows that $b^{h+1}_{ik} + b^{h+1}_{kj} = \min\{b^h_{ik}, b^{h+1}_{*k} + b^{h+1}_{i*}\} + b^{h+1}_{*j} + b^{h+1}_{k*} = \min\{b^h_{ik}, b^h_{fk} + b^{h+1}_{i*}\} + \min\{b^h_{kf}, b^h_{kl}\} + b^{h+1}_{*j} = \min\{b^h_{ik} + b^h_{kf}, b^h_{ik} + b^h_{kl}, b^h_{fk} + b^h_{kf} + b^{h+1}_{i*}, b^h_{fk} + b^h_{kl} + b^{h+1}_{i*}\} + b^{h+1}_{*j}$. By the inductive assumption and transition firability, $b^{h+1}_{ik} + b^{h+1}_{kj} \geq \min\{b^h_{if}, b^h_{il}, b^{h+1}_{i*}, b^{h+1}_{i*}\} + b^{h+1}_{*j} = b^{h+1}_{i*} + b^{h+1}_{*j} = b^{h+1}_{ij}$.

If $t_i, t_j, \in En(M_h\text{-}B(t_{h+1})) \cup \{t_0\}$, there are two cases:

1) If $t_k \in New(M_h, t_f)$, then $b^{h+1}_{ij} = \min\{b^h_{ij}, b^{h+1}_{i*} + b^{h+1}_{*j}\}$, $b^{h+1}_{ik} = b^{h+1}_{i*} + b^{h+1}_{*k}$ and $b^{h+1}_{kj} = b^{h+1}_{*j} + b^{h+1}_{k*}$. It follows that $b^{h+1}_{ik} + b^{h+1}_{kj} = b^{h+1}_{*k} + b^{h+1}_{i*} + b^{h+1}_{*j} + b^{h+1}_{k*} = \min\{b^h_{fk} + b^h_{kf}, b^h_{fk} + b^h_{kl}\} + b^{h+1}_{i*} + b^{h+1}_{*j}$. By the inductive assumption and transition firability, we obtain that $\min\{b^h_{fk} + b^h_{kf}, b^h_{fk} + b^h_{kl}\} \geq 0$. Thus, $b^{h+1}_{ik} + b^{h+1}_{kj} \geq b^{h+1}_{i*} + b^{h+1}_{*j} \geq b^{h+1}_{ij}$.

2) If $t_k \in En(M_h\text{-}B(t_{h+1})) \cup \{t_0\}$, then $b^{h+1}_{ij} = \min\{b^h_{ij}, b^{h+1}_{i*} + b^{h+1}_{*j}\}$, $b^{h+1}_{ik} = \min\{b^h_{ik}, b^{h+1}_{i*} + b^{h+1}_{*k}\}$ and $b^{h+1}_{kj} = \min\{b^h_{kj}, b^{h+1}_{k*} + b^{h+1}_{*j}\}$. It follows that $b^{h+1}_{ik} + b^{h+1}_{kj} = \min\{b^h_{ik} + b^h_{kj}, b^h_{ik} + b^{h+1}_{k*} + b^{h+1}_{*j}, b^{h+1}_{i*} + b^{h+1}_{*k} + b^h_{kj}, b^{h+1}_{i*} + b^{h+1}_{*k} + b^{h+1}_{k*} + b^{h+1}_{*j}\}$. By the inductive assumption and transition firability, we obtain that $b^{h+1}_{ik} + b^{h+1}_{kj} \geq b^h_{ij}, b^h_{ik} + b^{h+1}_{k*} + b^{h+1}_{*j} = b^h_{ik} + \min\{b^h_{kf}, b^h_{kl}\} + b^{h+1}_{*j} \geq \min\{b^h_{if}, b^h_{il}\} + b^{h+1}_{*j} = b^{h+1}_{i*}\, b^{h+1}_{*j}, b^{h+1}_{i*} + b^{h+1}_{*k} + b^h_{kj} = b^{h+1}_{i*} + b^h_{fk} + b^h_{kj} \geq b^{h+1}_{i*} + b^h_{fj} = b^{h+1}_{i*}\, b^{h+1}_{*j}$ and $b^{h+1}_{i*} + b^{h+1}_{*k} + b^{h+1}_{k*} + b^{h+1}_{*j} = b^{h+1}_{i*} + b^{h+1}_{*j} + \min\{b^h_{kf}, b^h_{kl}\} + b^h_{fk} \geq b^{h+1}_{i*}\, b^{h+1}_{*j}$. Thus, $b^{h+1}_{ik} + b^{h+1}_{kj} \geq b^{h+1}_{ij}$.

Therefore, $B(D_{h+1})$ is normal.

REFERENCES

[1] X. Guo, S. Wang, D. You, Z. Li, and X. Jiang, "A siphon-based deadlock prevention strategy for S3PR," *IEEE Access*, vol. 7, pp. 86863–86873, 2019.

[2] W. Duo, X. Jiang, O. Karoui, X. Guo, D. You, S. Wang, and Y. Ruan, "A deadlock prevention policy for a class of multithreaded software," *IEEE Access*, vol. 8, pp. 16676–16688, 2020.

[3] S. Wang, D. You, and M. Zhou, "A necessary and sufficient condition for a resource subset to generate a strict minimal siphon in s 4PR," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4173–4179, Aug. 2017.

[4] F. Yang, N. Wu, Y. Qiao, and R. Su, "Polynomial approach to optimal one-wafer cyclic scheduling of treelike hybrid multi-cluster tools via Petri nets," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 270–280, Jan. 2018.

[5] A. Janiak, W. A. Janiak, T. Krysiak, and T. Kwiatkowski, "A survey on scheduling problems with due windows," *Eur. J. Oper. Res.*, vol. 242, no. 2, pp. 347–357, Apr. 2015.

[6] T. Jamrus, C.-F. Chien, M. Gen, and K. Sethanan, "Hybrid particle swarm optimization combined with genetic operators for flexible job-shop scheduling under uncertain processing time for semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 1, pp. 32–41, Feb. 2018.

[7] Q.-K. Pan, R. Ruiz, and P. Alfaro-Fernández, "Iterated search methods for earliness and tardiness minimization in hybrid flowshops with due windows," *Comput. Oper. Res.*, vol. 80, pp. 50–60, Apr. 2017.

[8] B. Berthomieu and M. Diaz, "Modeling and verification of time dependent systems using time Petri nets," *IEEE Trans. Softw. Eng.*, vol. 17, no. 3, pp. 259–273, Mar. 1991.

[9] J. Wang, Y. Deng, and G. Xu, "Reachability analysis of real-time systems using time Petri nets," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 30, no. 5, pp. 725–736, Oct. 2000.

[10] L. Pan, Z. J. Ding, and M. C. Zhou, "A configurable state class method for temporal analysis of time Petri nets," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 44, no. 4, pp. 482–493, Apr. 2014.

[11] F. Basile, M. P. Cabasino, and C. Seatzu, "Diagnosability analysis of labeled time Petri net systems," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1384–1396, Mar. 2017.

[12] L. Li, F. Basile, and Z. Li, "An approach to improve permissiveness of supervisors for GMECs in time Petri net systems," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 237–251, Jan. 2020.

[13] T. Bolognesi and P. Cremonese, "The Weakness of Some Timed Models for Concurrent Systems," NR-Classic-Cars, Rudersberg, Germany, Tech. Rep. CNUCE C89-29, 1989.

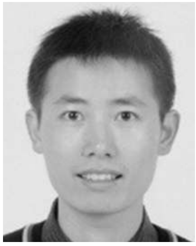
[14] P. A. Reynier and A. Sangnier, "Weak time Petri nets strike back!" in *Proc. Int. Conf. Concurrency Theory*, vol. 5710. Berlin, Germany: Springer, 2009, pp. 557–571.

[15] J. J. P. Tsai, S. Jennhwa Yang, and Y.-H. Chang, "Timing constraint Petri nets and their application to schedulability analysis of real-time system specifications," *IEEE Trans. Softw. Eng.*, vol. 21, no. 1, pp. 32–49, Jan. 1995.

[16] D. Xu, X. He, and Y. Deng, "Compositional schedulability analysis of real-time systems using time Petri nets," *IEEE Trans. Softw. Eng.*, vol. 28, no. 10, pp. 984–996, Oct. 2002.
[17] H. Lee and J.-Y. Choi, "Constraint solving approach to schedulability analysis in real-time systems," *IEEE Access*, vol. 6, pp. 58418–58426, 2018.
[18] Y. Qiao, N. Wu, and M. Zhou, "Schedulability and scheduling analysis of dual-arm cluster tools with wafer revisiting and residency time constraints based on a novel schedule," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 3, pp. 472–484, Mar. 2015.
[19] L. Pan and W. Zhao, "A mixed semantics model for schedulability analysis of time Petri nets," in *Proc. Int. Conf. Manage. E-Commerce E-Government*, 2009, pp. 503–508.
[20] H. Boucheneb, D. Lime, and O. H. Roux, "On multi-enabledness in time Petri nets," in *Proc. Int. Conf. Appl. Theory Petri Nets Concurrency*, vol. 7927. Berlin, Germany: Springer, 2013, pp. 130–149.
[21] Z. Ma, Z. Li, and A. Giua, "Marking estimation in a class of time labeled Petri nets," *IEEE Trans. Autom. Control*, vol. 65, no. 2, pp. 493–506, Feb. 2020, doi: 10.1109/TAC.2019.2907413.
[22] Z. He, Z. Li, and A. Giua, "Optimization of deterministic timed weighted marked graphs," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1084–1095, Apr. 2017.
[23] B. Berard, F. Cassez, S. Haddad, D. Lime, and O. H. Roux, "Comparison of the expressiveness of timed automata and time Petri nets," in *Proc. Int. Conf. Formal Model. Anal. Timed Syst.*, vol. 3829. Berlin, Germany: Springer, 2005, pp. 211–225.
[24] B. Bérard, F. Cassez, S. Haddad, D. Lime, and O. H. Roux, "The expressive power of time Petri nets," *Theor. Comput. Sci.*, vol. 474, pp. 1–20, Feb. 2013.
[25] N. D. Jones, L. H. Landweber, and Y. Edmund Lien, "Complexity of some problems in Petri nets," *Theor. Comput. Sci.*, vol. 4, no. 3, pp. 277–299, Jun. 1977.
[26] E. Vicario, "Static analysis and dynamic steering of time-dependent systems," *IEEE Trans. Softw. Eng.*, vol. 27, no. 8, pp. 728–748, Aug. 2001.
[27] H. Boucheneb and K. Barkaoui, "Stubborn sets for time Petri nets," *ACM Trans. Embedded Comput. Syst.*, vol. 14, no. 1, pp. 1–25, Jan. 2015.
[28] K. Klai, "Timed aggregate graph: A finite graph preserving event-and state-based quantitative properties of time Petri nets," in *Transactions on Petri Nets and Other Models of Concurrency*, vol. 9410. Berlin, Germany: Springer, 2015, pp. 34–54.
[29] H. Boucheneb, D. Lime, B. Parquier, O. H. Roux, and C. Seidner, "Optimal reachability in cost time Petri nets," in *Formal Modeling and Analysis of Timed Systems*, vol. 10419. Berlin, Germany: Springer, 2017, pp. 58–73.
[30] A. Janiak, W. Janiak, and R. Januszkiewicz, "Algorithms for parallel processor scheduling with distinct due windows and unit-time jobs," *Bull. Polish Acad. Sci., Tech. Sci.*, vol. 57, no. 3, pp. 209–215, Jan. 2009.
[31] M. A. Weber, R. Leisten, and M. A. Weber, "Minimizing Total Time Deviation from Due Windows for Jobs with Multiple Delivery Due Windows Using Lot-streaming in Permutation Flow Shops," *Int. J. Oper. Quant. Manage.*, vol. 23, no. 3, pp. 161–185, 2017.
[32] E. Taillard, "Benchmarks for basic scheduling problems," *Eur. J. Oper. Res.*, vol. 64, no. 2, pp. 278–285, 1993.
[33] C. Pan, M. Zhou, Y. Qiao, and N. Wu, "Scheduling cluster tools in semiconductor manufacturing: Recent advances and challenges," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 586–601, Apr. 2018.
[34] H. Alhussian, N. Zakaria, A. Patel, A. Jaradat, S. J. Abdulkadir, A. Y. Ahmed, H. T. Bahbouh, S. O. Fageeri, A. A. Elsheikh, and J. Watada, "Investigating the schedulability of periodic real-time tasks in virtualized cloud environment," *IEEE Access*, vol. 7, pp. 29533–29542, 2019.
[35] K. Gao, Z. Cao, L. Zhang, Z. Chen, Y. Han, and Q. Pan, "A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 4, pp. 904–916, Jul. 2019.
[36] J. Bengtsson and W. Yi, "Timed automata: Semantics, algorithms and tools," in *Lectures on Concurrency and Petri Nets*, vol. 3098. Berlin, Germany: Springer, pp. 87–124, 2004.
[37] D. Dill, "Timing assumptions and verification of finite-state concurrent systems," *Proc. Workshop Comput. Aided Verification Methods Finite State Syst.*, vol. 407. Berlin, Germany: Springer, pp. 197–212, 1989.

LI PAN received the M.S. degree in computer software and theory from Sun Yat-sen University, Guangzhou, China, in 2004, and the Ph.D. degree in computer applied technology from Tongji University, Shanghai, China, in 2009. He is currently a Professor with the Department of Information Science and Engineering, Hunan Institute of Science and Technology, China. He has published more than 30 articles in refereed journals and conference proceedings. His main research interests include Petri nets and computational intelligence.

BO YANG received the B.Sc. degree in mechanical engineering from Zhengzhou University, China, in 1996, the M.Sc. degree in computer application technology from Xiangtan University, China, in 2004, and the Ph.D. degree in mechanical and electronic engineering from Central South University, China, in 2010. Since 2012, he has been an Associate Professor with the College of Information Science and Technology, Hunan Institute of Science and Technology. His main research interests include MR brain image analysis, statistical pattern recognition, and machine learning.



JUNQIANG JIANG received the Ph.D. degree in software engineering from Hunan University, Changsha, China, in 2017. He is currently a Master Supervisor with the School of Information Science and Engineering, Hunan Institute of Science and Technology, China. His main research interests include cloud computing, parallel computing and workflow scheduling, and machine learning. He is a member of China Computer Federation.

MENGCHU ZHOU (Fellow, IEEE) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and system engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

In 1990, he joined the New Jersey Institute of Technology, Newark, NJ, USA, where he is currently a Distinguished Professor of electrical and computer engineering. He has authored or coauthored over 640 publications, including 12 books, over 330 journal articles (majority in the IEEE TRANSACTIONS), and 28 book chapters. His current research interests include Petri nets, manufacturing, transportation, the Internet of Things, big data, web service, sensor networks, and energy systems.

Dr. Zhou is a Fellow of the International Federation of Automatic Control and American Association for the Advancement of Science. In 1999, he served as the President of the Chinese Association for Science and Technology, USA, where he is currently a Life Member. He received the Humboldt Research Award for U.S. Senior Scientists, the Franklin V. Taylor Memorial Award, and the Norbert Wiener Award from the IEEE Systems, Man, and Cybernetics Society. He is the Founding Editor of the IEEE Press Book Series on Systems Science and Engineering.