# Minimum-time Decentralized Control of Choice-Free Continuous Petri Nets☆

Liewei Wang[a,*], Cristian Mahulea[a], Jorge Júlvez[a], Manuel Silva[a]

[a] *The authors are with the Aragón Institute of Engineering Research (I3A), University of Zaragoza, Maria de Luna 1, 50018*

**Abstract**

Aiming to reach a desired final state from a given initial one, this paper addresses the minimum-time decentralized control of Choice-Free continuous Petri nets. It is assumed that the original system is cut into disconnected subsystems by a set of places (buffers). Local control laws are first computed independently in subsystems, based on which the globally admissible ones are derived. In the process, two problems arise: 1) disconnected subsystems can exhibit different behaviours from the original ones, and 2) since the buffer places are essentially shared by more than one subsystems, there must be an agreement among the neighoring local controllers. The first problem can be overcome by complementing the disconnected subsystems with an abstraction of the parts that are missing. For this purpose, two reduction rules are proposed to substitute the missing parts by a set of places. For the second problem, a simple coordinate controller is introduced, and several algorithms are proposed to reach the agreement, without knowing the detailed structure of subsystems. Finally, by applying an ON-OFF control strategy in each subsystem, the final state is ensured to be reached in minimum-time.

*Keywords:* Discrete-event systems, continuous Petri nets, decentralized control, minimum-time control

☆This work has been partially supported by the European Community's Seventh Framework Programme under project DISC (Grant Agreement n. INFSO-ICT-224498), by CICYT - FEDER grants DPI2010-20413 and TIN2007-66523 and by Fundación Aragón I+D.

*Corresponding author. Tel: +34 976762472 Fax: +34 976761914

*Email addresses:* lwwang@unizar.es (Liewei Wang ), cmahulea@unizar.es (Cristian Mahulea), julvez@unizar.es (Jorge Júlvez), silva@unizar.es (Manuel Silva)

## 1. Introduction

### 1.1. Related work

Petri Nets (PN) is a well known paradigm used for modeling, analysis, and synthesis of *discrete event systems* (DES). Since it can easily represent sequences, conflicts, concurrency and synchronizations, it is widely applied in the industry, for the analysis of manufacturing, traffic, software systems, etc. Similarly to other modeling formalisms for DES, it also suffers from the *state explosion* problem. To overcome it, a classical relaxation technique called *fluidification* can be used.

*Continuous PN* (CPN) [1, 2] are fluid approximations of classical *discrete PN* obtained by removing the integrality constraints, which means that the firing count vector and consequently the marking are no longer restricted to be in the naturals but relaxed into the non-negative real numbers. An important advantage of this relaxation is that more efficient algorithms are available for their analysis [3, 4].

A simple and interesting way to introduce time to CPN is to assume that time is associated to transitions, obtaining timed CPN. Many works can be found in the literature about the control of different classes of timed CPN, e.g., [5, 6, 7]. For the kind of timed CPN under *infinite server semantics*, several control approaches have been considered. In [3], the optimal steady state control problem is studied. Model Predictive Control is used for optimal control problem in [8] assuming a discrete-time model. In [9], a Lyapunov-function-based dynamic control algorithm is studied, while in [10] a heuristics for minimum-time control is proposed. In this work, a minimum-time control problem of *timed CPN* under *infinite server semantics* is considered.

Decentralized control is extensively explored in recent decades for complex dynamic systems (e.g., [11, 12, 13, 14]), in which multiple controllers may be allocated to subsystems. In the context of decentralized control on PN, some approaches have been proposed. The centralized admissibility concept was extended to d-admissibility for the decentralized setting in [15]. Based on the d-admissibility concept, two suboptimal methods to design decentralized supervisors are proposed. Under certain assumptions, the methods in [16] focused on global state specifications given in terms of GMECs and on a control architecture without central coordinator and communication between local supervisors. In [17], a decentralized approach based on overlapping decompositions was proposed. By adding control places, the system is driven from an initial marking to a set of the desired markings. The method presented in

[18] considers continuous models composed of several subsystems that communicate through buffers (modelled by places). By executing the proposed algorithm iteratively in each subsystem, their respective target markings are reached and then maintained.

Different from the methods in [15, 16] which focus on enforcing system states to satisfy certain constraints (specifications), we address the problem of driving the system from an initial state to a specific final one, which is similar to the set-point control problem in a general continuous-state system. Considering the method in [17], systems are targeted to a set of desired states, but when a specific one is chosen, the control complexity may be increased (because more control places should be be added). On the other hands, its control structures are also strongly dependent on the desired markings. Our method is also different from the one of [18]. First, subsystems do not have to be strongly connected. Second, the globally admissible control laws are achieved inside a simple coordinator, therefore the iterative process executed in subsystems is not needed. Finally, the states of buffers are also specified in the control problem and reached in minimum-time.

### *1.2. The undertaken problem*

Imagining that there is large scale dynamic system, e.g., a complex transportation system connecting cities from different countries. Because of the distributed physical deployment or the high costs, it may be difficult to have a central controller which knows all the detailed structures and states of all subsystems, and the global control law can not be achieved directly. A more practical way is to have local controller allocated in each subsystem, which is the essence of decentralized control. The intersections among neighboring subystems (in our case, modeled by places) play a important role in facilitating the interaction and communication between neighboring subsystems.

It is assumed that the original system modelled by CPN is cut into disconnected subsystems by a set of places (buffers), and the addressed problem is how to compute the control law and drive the system from an initial state to desired final one, in a decentralized way: local controllers first compute control laws separately, then based on the local control laws, the globally admissible ones are derived without knowing the detailed structures of subsystems. There are two main problems arising in this process: 1) disconnected subsystems can exhibit different behaviours from the original ones, e.g., properties like liveness, boundedness in the original system may not be preserved. And 2) since the buffer places are essentially shared by more than one subsystems, there must be an agreement among the neighoring local controllers. The first problem can be overcome by complementing the subsystems with an abstraction of the parts that are missing. For this purpose, two reduction rules are proposed to substitute the missing parts by a set of places. For the second problem, a simple coordinate controller is introduced. Local controllers send limited information (the firing count vector and the T-semiflow) to the coordinator, and based on this information, algorithms are proposed to reach the agreement. After the globally admissible control laws are obtained, a simple ON-OFF controller is applied in each subsystem. Considering the system is Choice-Free, this ON-OFF strategy ensures the final state to be reached in minimum-time. The sketch of the system structure is shown in Fig. 1.

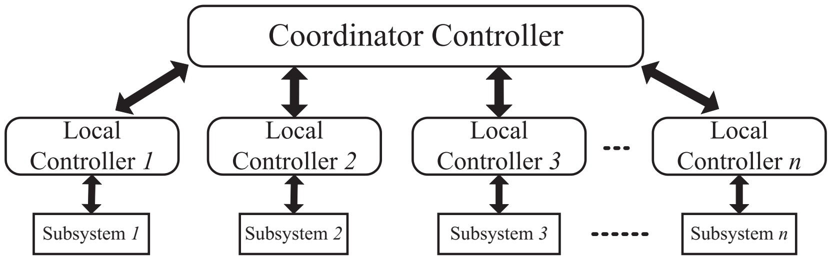



Figure 1: System Structures

This paper is organized as follows: Section 2 briefly recalls some basic concepts of CPN. Section 3 introduces the decomposition method for CFPN and proposes two reduction rules in order to obtain complemented subsystems. Section 4 proposes the approach for decentralized control of CFPN system. In section 5, we illustrate the proposed methods by using a manufacturing system as the case study. The conclusions and some final remarks are in section 6.

## 2. Basic Concepts and Notations

### *2.1. Continuous Petri Nets*

The reader is assumed to be familiar with basic concepts of CPN (see [1, 2] for a gentle introduction).

**Definition 2.1.** *A CPN system is a pair* $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ *where* $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$ *is a net structure where:*

- *$P$ and $T$ are the sets of places and transitions respectively.*
- $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{Q}_{\geq 0}^{|P|\times|T|}$ *are the pre and post incidence matrices.*
- $\boldsymbol{m}_0 \in \mathbb{R}_{\geq 0}^{|P|}$ *is the initial marking (state).*

For $v \in P \cup T$, the sets of its input and output nodes are denoted as ${}^\bullet v$ and $v^\bullet$, respectively. Let $p_i$, $i = 1, \ldots, |P|$ and $t_j, j = 1, \ldots, |T|$ denote the places and transitions. Each place can contain a non-negative real number of tokens, its marking. The distribution of tokens in places is denoted by $\boldsymbol{m}$. The enabling degree of a transition $t_j \in T$ is given by:

$$enab(t_j, \boldsymbol{m}) = \min_{p_i \in {}^\bullet t_j} \left\{ \frac{\boldsymbol{m}(p_i)}{\boldsymbol{Pre}(p_i, t_j)} \right\}$$

which represents the maximum amount in which $t_j$ can fire. Transition $t_j$ is called *k-enabled* under marking $\boldsymbol{m}$, if $enab(t, \boldsymbol{m}) = k$, being enabled if $k > 0$. An enabled transition $t_j$ can fire in any real amount $\alpha$, with $0 < \alpha \leq enab(t_j, \boldsymbol{m})$ leading to a new state $\boldsymbol{m'} = \boldsymbol{m} + \alpha \cdot \boldsymbol{C}(\cdot, t_j)$ where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the *token flow matrix* and $\boldsymbol{C}(\cdot, j)$ is its $j^{th}$ column.

Non negative left and right natural annullers of the token flow matrix $\boldsymbol{C}$ are called P-*semiflows* (denoted by $\boldsymbol{y}$) and T-*semiflows* (denoted by $\boldsymbol{x}$), respectively. If $\exists \boldsymbol{y} > 0$, $\boldsymbol{y} \cdot \boldsymbol{C} = 0$, then the net is said to be *conservative*. If $\exists \boldsymbol{x} > 0$, $\boldsymbol{C} \cdot \boldsymbol{x} = 0$ it is said to be *consistent*. The support of a vector $\boldsymbol{v}$, denoted by $||\boldsymbol{v}||$, is the set of index of nonzero components. A semiflow $\boldsymbol{v}$ is said to be minimal when its support is not a proper superset of any other, and the greatest common divisor of its components is one.

A PN system is bounded when every place is bounded, i.e., its token content is less than some bounds at every reachable marking. It is live when every transition is live, i.e., it can ultimately occur from every reachable marking.

If $\boldsymbol{m}$ is reachable from $\boldsymbol{m}_0$ through a finite sequence $\sigma$, the state (or fundamental) equation is satisfied: $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, where $\boldsymbol{\sigma} \in \mathbb{R}_{\geq 0}^{|T|}$ is the *firing count vector*, i.e., $\boldsymbol{\sigma}(t_j)$ is the cumulative amount of firings of $t_j$ in the sequence $\sigma$. A vector $\boldsymbol{\sigma}$ is said to be a fireable firing count vector, if there exists a corresponding sequence $\sigma$ which can be fired. A firing count vector $\boldsymbol{\sigma}$ is said to be *minimal* one driving the system to $\boldsymbol{m}$ if for any T-*semiflow* $\boldsymbol{x}$, $||\boldsymbol{x}|| \not\subseteq ||\boldsymbol{\sigma}||$.

If for all $p \in P$, $|p^\bullet| \leq 1$ then $\mathcal{N}$ is called *Choice-Free PN* (CFPN). A CFPN is structurally persistent in the sense that independently of the initial marking, the net has no conflict, i.e., it is *conflict-free* [19]. A net is said to be a marked graph (MG) when the weight of every arc is equal to 1, and each place has exactly one input and exactly one output arc. Weighted T-systems (WTS) are the weighted generalization of MGs. The following property holds for conflict-free PN, it is also true for CFPN.

**Property 2.2.** *In a CFPN system, if transition $t_j$ is k-enabled, its enabling degree will be at least k until $t_j$ is fired.*

In timed CPN (TCPN) the state equation has an explicit dependence on time: $\boldsymbol{m}(\tau) = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}(\tau)$ which through time differentiation becomes $\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \dot{\boldsymbol{\sigma}}(\tau)$. The derivative of the firing count $\boldsymbol{f}(\tau) = \dot{\boldsymbol{\sigma}}(\tau)$ is called the *firing flow*. Depending on how the flow is defined, many firing semantics appear, being the most used ones *infinite* (or variable speed) and *finite* (or constant speed) server semantics [1, 2]. In this paper we assume the system is under infinite server semantics, because for a broad class of PN it offers a better approximation of the throughput in steady state of discrete systems [20]. For each transition $t_j \in T$, let $\lambda_j \in \mathbb{R}_{>0}$ be its firing rate. Under infinite server semantics, the flow of a transition $t_j$ at time $\tau$ is the product of its firing rate, $\lambda_j$, and its enabling degree at $\boldsymbol{m}(\tau)$:

$$f(t_j, \tau) = \lambda_j \cdot enab(t_j, \boldsymbol{m}(\tau)) = \lambda_j \cdot \min_{p_i \in {}^\bullet t_j} \left\{ \frac{\boldsymbol{m}(p_i, \tau)}{\boldsymbol{Pre}(p_i, t_j)} \right\} \tag{1}$$

*2.2. Gains and Weighted Markings*

The gain of a weighted path was introduced in [21] for WTS, it represents the mean firing ratio between the last transition and the first one in the path. It can be naturally extended to CFPN systems:

**Definition 2.3.** *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a CFPN system, and $\pi = \{t_0, p_1, t_1, p_2, ..., p_n, t_n\}$ be a path in $\mathcal{N}$ from transition $t_0$ to $t_n$. The gain of $\pi$ is:*

$$G(\pi) = \prod_{i=1}^{n} \frac{\boldsymbol{Post}(p_i, t_{i-1})}{\boldsymbol{Pre}(p_i, t_i)}$$

The weighted marking $M(\pi, \boldsymbol{m})$ of a path $\pi$ under marking $\boldsymbol{m}$ in a CFPN system is the natural extension of the sum of tokens of paths in marked graphs.

**Definition 2.4.** *Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a CFPN system, and $\pi = \{t_0,\ p_1,\ t_1,\ p_2,\ ...,\ p_n,\ t_n\}$ be a path in $\mathcal{N}$ from transition $t_0$ to $t_n$. The weighted marking of $\pi$ under marking $\boldsymbol{m}$ is:*

$$M(\pi, \boldsymbol{m}) = \sum_{i=1}^{n} \left( \frac{\boldsymbol{m}(p_i)}{\boldsymbol{Post}(p_i, t_{i-1})} \prod_{j=1}^{i-1} \frac{\boldsymbol{Pre}(p_j, t_j)}{\boldsymbol{Post}(p_j, t_{j-1})} \right)$$

Let $t_{in}$ and $t_{out}$ ($t_0$ and $t_n$ in the former definition) be the first and last transitions of $\pi$, $M(\pi, \boldsymbol{m})$ can be interpreted as the number of firings $t_{in}$ is required to be fired to reach $\boldsymbol{m}$, in the case that $\pi$ is initially empty. It can be deduced that, starting from $\boldsymbol{m}$, if all the intermediate transitions between $t_{in}$ and $t_{out}$ are fired with the maximal amounts, the enabling degree of $t_{out}$ becomes $G(\pi) \cdot M(\pi, \boldsymbol{m})$.

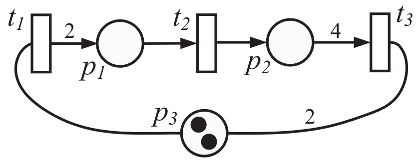



Figure 2: A simple CFPN system with $\boldsymbol{m}_0 = [0\ 0\ 2]^T$

**Example 2.5.** *Let us consider the CFPN system in Fig. 2. The path between $t_1$ and $t_3$ is $\pi = \{t_1,\ p_1,\ t_2,\ p_2,\ t_3\}$, according to the definition of gains, $G(\pi) = \frac{Post(p_1,t_1) \cdot Post(p_2,t_2)}{Pre(p_1,t_2) \cdot Pre(p_2,t_3)} = \frac{2 \cdot 1}{1 \cdot 4} = 1/2$. It means that if $t_1$ fires once, $t_3$ can fire $1/2$ times (in the case that $p_1$ and $p_2$ are empty initially).*

*In the initial state, path $\pi$ is empty, i.e., $\boldsymbol{m}_0(p_1) = 0$, $\boldsymbol{m}_0(p_2) = 0$. In order to reach a marking $\boldsymbol{m}$, such that $\boldsymbol{m}(p_1) = 1$, $\boldsymbol{m}(p_2) = 1$, so $\boldsymbol{\sigma}$ =[1 1 0]$^T$, $t_1$ needs to fire once, therefore, the weighted marking of $\pi$ under $\boldsymbol{m}$ is $M(\pi, \boldsymbol{m}) = 1$.*

*Suppose that from $\boldsymbol{m}$ the intermediate transition $t_2$ is fired in the maximal amount that is equal to 1, the enabling degree of $t_3$ becomes $1/2$, obviously it is equal to $G(\pi) \cdot M(\pi, \boldsymbol{m})$.*

### *2.3. System Under Control*

Now the net system is considered to be subject to external control actions, and it is assumed that the only admissible control law consists in *slowing down* the firing flow of transitions [2], i.e., transitions, modeling machines for example, cannot work faster than their nominal speeds. Under this assumption, the controlled flow of a TCPN system is denoted as: $\boldsymbol{w}(\tau) = \boldsymbol{f}(\tau) - \boldsymbol{u}(\tau)$, with $0 \leq \boldsymbol{u}(\tau) \leq \boldsymbol{f}(\tau)$. The overall behaviour of the system is ruled by: $\dot{\boldsymbol{m}} = \boldsymbol{C} \cdot (\boldsymbol{f}(\tau) - \boldsymbol{u}(\tau))$. In this paper, it is assumed that every transition is *controllable* ($t_j$ is uncontrollable if the only control action that can be applied is $u(t_j) = 0$).

## 3. Structural Decomposition of CFPN systems

In this section, a structural decomposition approach for CFPN is introduced, obtaining subsystems that have behaviours consistent with in the original system.

Given a large scale system, naturally it may be divided into several parts, for example, due to its physical deployments. Here we suppose that the original system is cut into subsystems through a given set of places (buffers). Local control laws will be separately computed in subsystems, but because these subsystems become disconnected with the other parts, their behaviours may be different from the ones in the original system. To overcome this problem, a set of reduction rules is proposed to obtain the abstraction of the missing parts, by which the disconnected subsystem is *complemented*. It is then proved that the behaviours (firing sequences and consequently, the reachable markings) of the original system are preserved.

### *3.1. Cutting*

Here the structural cutting method developed in [22] for MGs is extended to CFPN. In order to simplify the notation, we assume that the system is cut into two parts.

**Definition 3.1.** *Let $\mathcal{S} = \langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a strongly connected CFPN system, where $\mathcal{N} = \langle P \cup B, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$. $B$ is said to be a* cut *if there exist two subnets $\mathcal{N}_i = \langle P_i, T_i, \boldsymbol{Pre}_i, \boldsymbol{Post}_i \rangle$, $i = 1, 2$, such that:*

*(1)* $T_1 \cup T_2 = T$, $T_1 \cap T_2 = \emptyset$

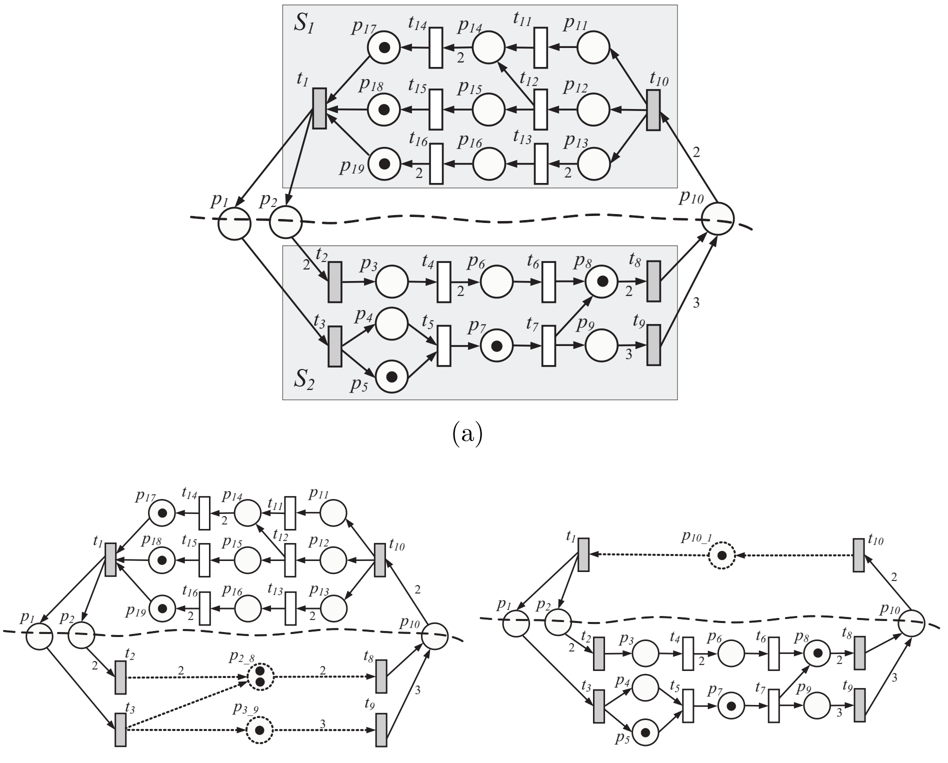
*(2)* $P_1 \cup P_2 = P$, $P_1 \cap P_2 = \emptyset$

*(3)* $P_1 \cup B = {}^\bullet T_1 \cup T_1{}^\bullet$, $P_2 \cup B = {}^\bullet T_2 \cup T_2{}^\bullet$

*(4)* $T_1 = {}^\bullet P_1 \cup P_1{}^\bullet$, $T_2 = {}^\bullet P_2 \cup P_2{}^\bullet$

where $U = {}^\bullet B \cup B^\bullet$ is said to be interface, which is partitioned into $U_1$, $U_2$, such that $U_1 \cup U_2 = U$, $U_i = T_i \cap U$.

**Example 3.2.** *Fig. 3(a) shows a CFPN system. The set of places* $B = \{p_1, p_2, p_{10}\}$ *is a cut decomposing the original system into two subsystems,* $\mathcal{S}_1$ *and* $\mathcal{S}_2$*, where the interface transitions are* $U_1 = \{t_1, t_{10}\}$ *and* $U_2 = \{t_2, t_3, t_8, t_9\}$.




(a)

(b) $\mathcal{CS}_1$ (c) $\mathcal{CS}_2$

Figure 3: (a) A live and bounded CFPN system and a cut $B = \{p_1, p_2, p_{10}\}$; (b) complemented subsystem $\mathcal{CS}_1$; (c) complemented subsystem $\mathcal{CS}_2$

### *3.2. Complemented Subsystems*

Due to the cut, different behaviours can be introduced, because subsystems become disconnected with the remaining parts. For instance, the net system in Fig.3(a) is live and bounded. After cutting by $B = \{p_1, p_2, p_{10}\}$, both obtained subsystems $\mathcal{S}_1$ and $\mathcal{S}_2$ become unbounded. A solution to this problem is to build an abstraction of the missing parts and use it to complement the disconnected subsystem.

Two rules are proposed to reduce paths between (interface) transitions to a set of places. Let us still consider the system in Fig.3(a). By applying the proposed rules, the path between interface place $t_1$ and $t_{10}$ can be reduced to a single place $p_{10\_1}$, obtaining the abstraction of $\mathcal{S}_1$. Using this abstraction to complement $\mathcal{S}_2$, the complemented subsystem $\mathcal{CS}_2$ is obtained, shown in Fig.3(c). Similarly, the abstraction of $\mathcal{S}_2$ can be constructed, and the complemented subsystem $\mathcal{CS}_1$ is shown in Fig.3(b). Notice that, the cutting places and interface transitions are shared in both complemented subsystems.

In the sequel, net systems are assumed to be live and bounded (in the case of CFPN, it is equal to strongly connected and consistent).

**Reduction Rule 1.** *Let $t_j$ be a transition in a continuous CFPN system $\mathcal{S} = \langle \mathcal{N}, \boldsymbol{m}_0 \rangle$, with $|^\bullet t_j| = n$, $|t_j{}^\bullet| = k$. Let us denote its inputs by $P_{in} = {}^\bullet t_j$, and its outputs by $P_{out} = t_j{}^\bullet$. Let $p_x \in P_{in}$, $p_y \in P_{out}$. Transition $t_j$ with its input and output places can be reduced to $n \cdot k$ places, obtaining the reduced system $\mathcal{S}' = \langle \mathcal{N}', \boldsymbol{m}'_0 \rangle$, by using the following process:*

*(1) Replace each elementary path $\{p_x, t_j, p_y\}$ with a place $p_{x\_y}$.*

*(2) Add arcs such that ${}^\bullet p_{x\_y} = {}^\bullet p_x \cup {}^\bullet p_y$, $p_{x\_y}{}^\bullet = p_y{}^\bullet$.*

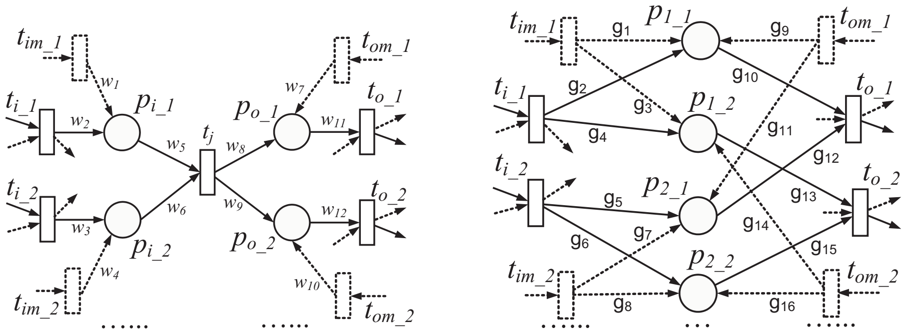
*(3) Add weights such that $G(\pi(t_{in}, t_{out})) = G(\pi'(t_{in}, t_{out}))$, where $t_{in} \in {}^\bullet P_{in} \cup {}^\bullet P_{out}$, $t_{out} \in P_{out}{}^\bullet$, $\pi(t_{in}, t_{out})$ and $\pi'(t_{in}, t_{out})$ are the paths from $t_{in}$ to $t_{out}$, in $\mathcal{S}$ and $\mathcal{S}'$ respectively.*

*(4) Put the initial marking $\boldsymbol{m}'_0(p_{x\_y}) = \boldsymbol{Post}(p_{x\_y}, t_{in}) \cdot M(\pi, \boldsymbol{m}_0)$, where $\pi = \{t_{in}, p_x, t_j, p_y, t_{out}\}$.*

**Remark 3.3.** *In step (3), the the weight on the arcs of the reduced net is not unique, but the gains of paths should be maintained. For instance, in the CPN in Fig. 2, by keeping the gain of path $\{t_1, p_1, t_2\}$, we can put weight $\boldsymbol{Post}(p_1, t_1) = 4$ and $\boldsymbol{Pre}(p_1, t_2) = 2$ (in this case, the marking of $p_1$ is still zero). Obviously, the overall behaviours of the system are not changed (notice that this conclusion only holds for continuous systems).*

In the sequel, it is assumed that for any place $p$ obtained by applying *rule* 1, the weights on the arcs connecting with $p$, are constrained to natures, and have the greatest common divisor equal to one. In this way, the obtained system is uniquely (structurally) determined.

**Example 3.4.** *In Fig. 4(a) is a CFPN system $\mathcal{S}$, it is shown how to reduce $t_j$ by applying rule 1. $t_j$ has two inputs $P_{in} = \{p_{i\_1},\ p_{i\_2}\}$ and two outputs $P_{out} = \{p_{o\_1},\ p_{o\_2}\}$, therefore $n = k = 2$. Transitions $t_{i\_1}$ and $t_{i\_2}$ are the inputs of $p_{i\_1}$ and $p_{i\_2}$ which may have more inputs denoted by $t_{im\_1}$ and $t_{im\_2}$. Transitions $t_{o\_1}$ and $t_{o\_2}$ are the outputs of $p_{o\_1}$ and $p_{o\_2}$ which may also have more inputs denoted by $t_{om\_1}$ and $t_{om\_2}$.*




(a) Original system $\mathcal{S} = \langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ (b) Reduced system $\mathcal{S}' = \langle \mathcal{N}', \boldsymbol{m}_0' \rangle$

Figure 4: Reduction Rule 1

*Fig. 4(b) shows the reduced system $\mathcal{S}'$, where $p_{1\_1}$, $p_{1\_2}$, $p_{2\_1}$ and $p_{2\_2}$ are the new places. In particular, $p_{1\_1}$ is the reduction of path $\{p_{i\_1}, t_j, p_{o\_1}\}$, $p_{1\_2}$ is the reduction of path $\{p_{i\_1}, t_j, p_{o\_2}\}$, etc. Observe that the gain of the path from $t_{i\_1}$ to $t_{o\_1}$, i.e., $\pi = \{t_{i\_1}, p_{i\_1}, t_j, p_{o\_1}, t_{o\_1}\}$ is $G(\pi) = \frac{w_2 \cdot w_8}{w_5 \cdot w_{11}}$. The weights $g_2, g_{10}$ on the path of the reduced net between the same transitions, i.e., $\pi' = \{t_{i\_1}, p_{1\_1}, t_{o\_1}\}$, should satisfy $\frac{g_2}{g_{10}} = G(\pi)$. Considering $p_{1\_1}$ in $\mathcal{S}'$, step (4) implies that $\boldsymbol{m}_0'(p_{1\_1}) = g_2 \cdot M(\pi, \boldsymbol{m}_0)$.*

Let $\mathcal{S} = \langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ and $\mathcal{S}' = \langle \mathcal{N}', \boldsymbol{m}_0' \rangle$ be the original and reduced CFPN systems, $\sigma$ be a firing sequence in $\mathcal{S}$. Sequence $\varsigma$ is said to be the projection of $\sigma$ from $\mathcal{S}$ to $\mathcal{S}'$ when $\varsigma$ is obtained from $\sigma$ by removing the elements corresponding to transitions $t_j$, $t_j \notin T \cap T'$.

**Proposition 3.5.** *Let $\mathcal{S}$ be a continuous CFPN system, and $\mathcal{S}'$ be its reduced system obtained by applying* rule *1, removing a transition $t_j$. Assume $\sigma$ is a firing sequence of $\mathcal{S}$, and $\varsigma$ is its projection to $\mathcal{S}'$. Then $\sigma$ is fireable in $\mathcal{S}$ if and only if $\varsigma$ is fireable in $\mathcal{S}'$.*

*Proof.* In order to prove the result, we will first consider a given firing sequence $\sigma$, and prove that $\sigma$ is fireable in $\mathcal{S}$ iff $\varsigma$ is fireable in $\mathcal{S}'$. Then it is shown that the proof can be easily extended for any firing sequence.

Considering *rule* 1 applied in Fig. 4 to reduce transition $t_j$ and its input, output places. Let us assume, without loss of generality, the firing sequence in $\mathcal{S}$, $\sigma = t_{i\_1}(\alpha_1)t_{i\_2}(\alpha_2)\, t_j(\beta)t_{im\_1}(\alpha_3)t_{o\_1}(\alpha_4)$, and its projection to the reduced system $\mathcal{S}'$, $\varsigma = t_{i\_1}(\alpha_1)\; t_{i\_2}(\alpha_2)\; t_{im\_1}(\alpha_3)t_{o\_1}(\alpha_4)$.

In $\mathcal{S}$, let $\pi_1 = \{t_{i\_1},\ p_{i\_1},\ t_j,\ p_{o\_1},\ t_{o\_1}\}$, $\pi_2 = \{t_{i\_2},\ p_{i\_2},\ t_j,\ p_{o\_1},\ t_{o\_1}\}$, and $\pi_3 = \{t_{im\_1},\ p_{i\_1},\ t_j,\ p_{o\_1},\ t_{o\_1}\}$. In $\mathcal{S}'$, let $\pi_1'$, $\pi_2'$ and $\pi_3'$ be the paths corresponding to the same transitions as $\pi_1$, $\pi_2$ and $\pi_3$ respectively, i.e., $\pi_1' = \{t_{i\_1},\ p_{1\_1},\ t_{o\_1}\}$, $\pi_2' = \{t_{i\_2},\ p_{2\_1},\ t_{o\_1}\}$, and $\pi_3 = \{t_{im\_1},\ p_{1\_1},\ t_{o\_1}\}$.

Let us first consider a subsequence of $\sigma$, $\sigma_1 = t_{i\_1}(\alpha_1)t_{i\_2}(\alpha_2)t_j(\beta)t_{im\_1}(\alpha_3)$, and its corresponding projection to $\mathcal{S}'$, $\varsigma_1 = t_{i\_1}(\alpha_1)t_{i\_2}(\alpha_2)t_{im\_1}(\alpha_3)$. Obviously, $\sigma_1$ is fireable in $\mathcal{S}$ iff $\varsigma_1$ is fireable in $\mathcal{S}'$ because transitions $t_{i\_1}$, $t_{i\_2}$ and $t_{im\_1}$ have the same input places and corresponding markings in $\mathcal{S}$ and $\mathcal{S}'$.

In $\mathcal{S}$, if $t_j$ is fired with the maximal amount in $\sigma_1$, $t_{o\_1}$ will get the maximal enabling degree. Therefore by firing of $\sigma_1$, the enabling degree of $t_{o\_1}$ can be maximally increased by:

$$\phi = \min\{\alpha_1 \cdot G(\pi_1) + \alpha_3 \cdot G(\pi_3),\ \alpha_2 \cdot G(\pi_2)\}$$

Considering the initial marking $\boldsymbol{m}_0$, the maximal enabling degree of $t_{o\_1}$ by firing of $\sigma_1$ is:

$$\min\{G(\pi_1) \cdot M(\pi_1, \boldsymbol{m}_0),\ G(\pi_2) \cdot M(\pi_2, \boldsymbol{m}_0)\} + \phi$$

In $\mathcal{S}'$, the enabling degree of $t_{o\_1}$ under the initial marking is equal to:

$$\min\left\{\frac{\boldsymbol{m}_0'(p_{1\_1})}{g_{10}}, \frac{\boldsymbol{m}_0'(p_{2\_1})}{g_{12}}\right\}$$

According to according the reduction step (4), it is equal to

$$\begin{aligned}&\min\left\{\frac{g_2 \cdot M(\pi_1, \boldsymbol{m}_0)}{g_{10}}, \frac{g_5 \cdot M(\pi_2, \boldsymbol{m}_0)}{g_{12}}\right\}\\ &= \min\{G(\pi_1) \cdot M(\pi_1, \boldsymbol{m}_0),\ G(\pi_2) \cdot M(\pi_2, \boldsymbol{m}_0)\}\end{aligned}$$

By the firing of $\varsigma_1$, it is increased by the same amount $\phi$ as in $\mathcal{S}$, because $G(\pi_i) = G(\pi_i')$, $i = 1, 2, 3$.

Therefore, if $\sigma$ is fireable in $\mathcal{S}$, $\varsigma$ is for sure fireable in $\mathcal{S}'$. On the other side, if $\varsigma$ is fireable in $\mathcal{S}'$, $\sigma$ is fireable in $\mathcal{S}$ when the intermediate transition $t_j$ is fired in the maximal amount.

Similar proof can be achieved for any firing sequence following the procedure: 1) any sequence that consists of the transitions whose input places are the same in $\mathcal{S}$ and $\mathcal{S}'$ (like $t_{i\_1}, t_{i\_2}$ in Fig.4), is fireable in $\mathcal{S}$ iff its projection in $\mathcal{S}'$ is fireable . 2) any other transitions (like $t_{o\_1}, t_{o\_2}$ in Fig.4) can get the same enabling degrees in $\mathcal{S}$ and $\mathcal{S}'$, when sequences in 1) fire. □

**Remark 3.6.** *Reduction rule 1 is a generalization of the methods discussed in [23] for continuous CFPN systems. For instance, in [23], only ordinary nets are considered; on the other side, a transition that has multiple inputs or outputs while its output places have multiple inputs, might not be reducible.*

*It can be observed that, each time* rule *1 is applied to a subnet formed by paths between $T_{in} \in T$ and $T_{out} \in T$, one transition $t \notin T_{in} \cup T_{out}$ is removed. Therefore the repetitive application of* rule *1 results in a set of places between $T_{in}$ and $T_{out}$ but no transitions.*

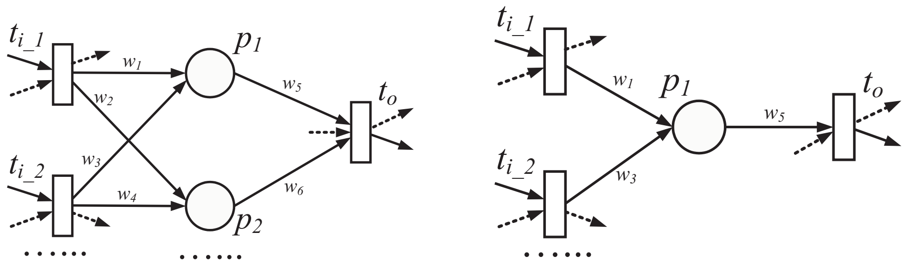
**Reduction Rule 2.** *Let $p_1, p_2$ be two places in a continuous CFPN system, such that ${}^\bullet p_1 = {}^\bullet p_2 = T_{in} \subseteq T, p_1{}^\bullet = p_2{}^\bullet = t_{out}$. If for any $t_{in} \in T_{in}$, paths $\pi_a = \{t_{in}, p_1, t_{out}\}$ and $\pi_b = \{t_{in}, p_2, t_{out}\}$ have the same gain, i.e., $G(\pi_a) = G(\pi_b)$. Then, if $\frac{\boldsymbol{m}_0(p_1)}{\boldsymbol{Pre}(p_1,t_{out})} \leq \frac{\boldsymbol{m}_0(p_2)}{\boldsymbol{Pre}(p_2,t_{out})}$, $p_2$ can be removed, otherwise, $p_1$ can be removed.*

In order to applying *rule* 2, $G(\pi_a) = G(\pi_b)$ has to be satisfied. Notice that if $G(\pi_a) \neq G(\pi_b)$, it implies not live or not bounded system.

**Example 3.7.** *Fig. 5(a) shows a CFPN system in which $T_{in} = \{t_{i\_1}, t_{i\_2}\}$. In order to apply* rule *2, the weights of arcs should satisfy $\frac{w_1}{w_5} = \frac{w_2}{w_6}$, and $\frac{w_3}{w_5} = \frac{w_4}{w_6}$. Suppose $\frac{\boldsymbol{m}_0(p_1)}{w_5} \leq \frac{\boldsymbol{m}_0(p_2)}{w_6}$, then by removing $p_2$, the reduced system is shown in Fig. 5(b).*

**Proposition 3.8.** *Let $\mathcal{S}$ be a continuous CFPN system, and $\mathcal{S}'$ be the reduced system obtained by applying* rule *2, sequence $\sigma$ is fireable in $\mathcal{S}$ if and only if $\sigma$ is fireable in $\mathcal{S}'$.*

*Proof.* It is easy to verify that the places being removed by applying *rule* 2 belong to a particular type of implicit places, i.e., those places that never uniquely restrict the firing of its output transitions (see [24]). Therefore, they can be removed without affecting the behaviour of the rest of the system. □




(a) Original subsystem $\mathcal{S}$ (b) Reduced subsystem $\mathcal{S}'$

Figure 5: Reduction Rule 2

**Example 3.9.** *Let us apply the reduction rules on subsystem $\mathcal{S}_2$ in Fig. 3(a). The net system in Fig. 6(a) is obtained by applying* rule *2 to remove places $p_5$. By applying* rule *1 to the path between $t_2$ and $t_6$, $p_{2\_6}$ is obtained (6(b)). Similarly, applying* rule *1 to the path between $t_3$ and $t_7$ in 6(b), removes $t_5$ and obtains $p_{3\_7}$ (6(c)). Applying* rule *1 to the path between $t_2$ and $t_8$ in 6(c), removes $t_6$ and obtains $p_{2\_8}$ (6(d)). Applying* rule *1 to the path between $t_3$ and $t_9$ in 6(d), removes $t_7$ and obtains $p_{3\_9}$ (6(e)). Finally, only two places are left with markings $\boldsymbol{m}_0'(p_{2\_8}) = 2$, $\boldsymbol{m}_0'(p_{3\_9}) = 1$. The reduced subsystem in Fig. 6(e) is the abstraction of $\mathcal{S}_2$.*

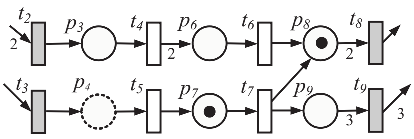
Assume that, using *rule* 1 and 2, we reduce the paths between two sets of transitions $T_{in}$ and $T_{out}$. Now we will discuss the uniqueness of the fully reduced system.

**Property 3.10.** *Any arbitrary and interleaved application of rule 1 and 2 until none of them can be applied produces the same reduced system.*

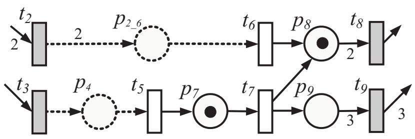
*Proof.* It is first proved that the order of adjacent rules that are applied can be interchanged, obtaining the same reduced system. Otherwise stated, let $A$ and $B$ be the instances of two rules, by applying $AB$, the same system is obtained as by applying $BA$. Then we will show that any sequence of rules, leading to the fully reduced system, can be reordered. After that, the uniqueness of the reduced system can be easily proved.

1) if $A$ and $B$ are both instances of *rule* 1 (or *rule 2*), it is trivial.

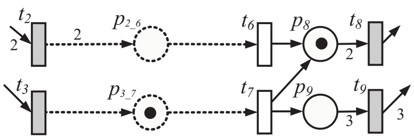
2) if $A$ and $B$ are instances of different rules. Without loss of generality, suppose $A$ is an instance of *rule* 1, removing a transition $t_j$ and B is an instance of *rule* 2, removing an implicit place $p_x$. Obviously, if $t_j \notin {}^\bullet p_x \cup p_x{}^\bullet$,



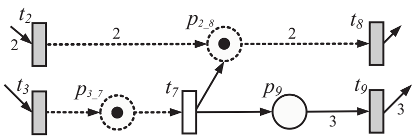
(a) applying *rule* 2 to $\mathcal{S}_2$



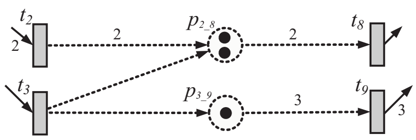
(b) applying *rule* 1 to (a)



(c) applying *rule* 1 to (b)



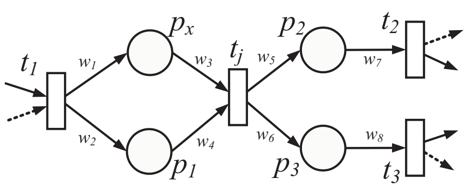
(d) applying *rule* 1 to (c)



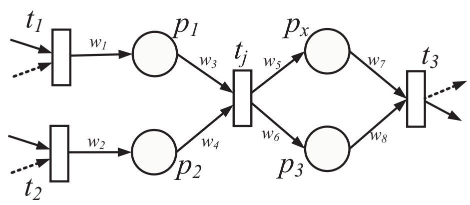
(e) applying *rule* 1 to (d)

Figure 6: Reduction process of $\mathcal{S}_2$ in Fig. 3(a)

$A$ and $B$ are independent, so the system obtained after applying $AB$ is equivalent to the one obtained after applying $BA$. Therefore, we only need to consider the two cases shown in Fig.7, where $t_j$ can be removed by using *rule* 1, at the same time, its input or/and output places can be reduced by using *rule* 2. Its extension to more general structures is quite straightforward.



(a)

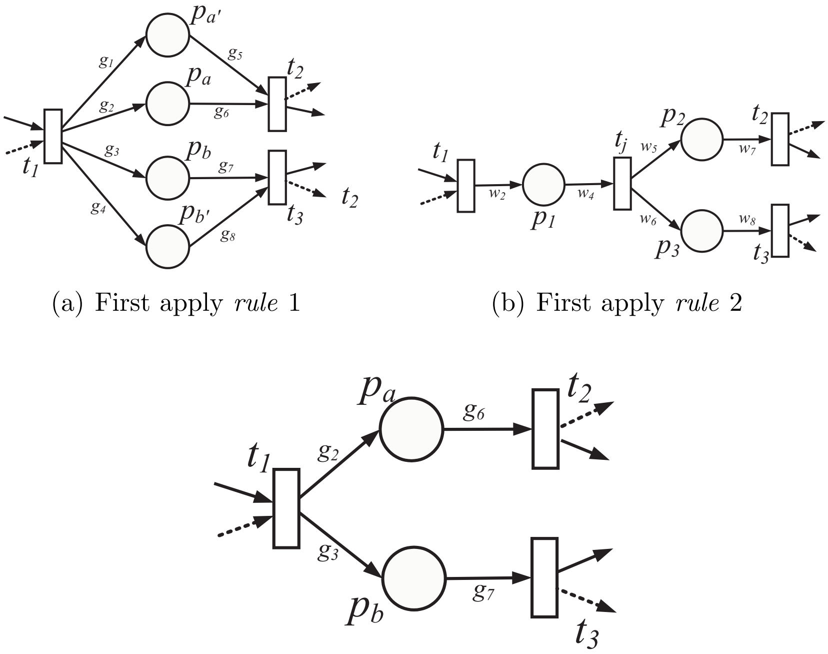


(b)

Figure 7: The two cases with $t_j \in {}^{\bullet}p_i \cup p_i{}^{\bullet}$

It will be shown that for case (a), by applying $AB$ and $BA$, the same system is obtained. The analysis to case (b) is similar.

Since $p_x$ can be removed by using *rule 2*, then $w_1/w_3 = w_2/w_4$ and in




(a) First apply *rule* 1

(b) First apply *rule* 2

(c) Finally reduced

Figure 8: Reducing by applying rules in different order

the initial state $\boldsymbol{m}_0(p_x)/w_3 \geq \boldsymbol{m}_0(p_x)/w_4$. Let path $\pi_1 = \{t_1, p_x, t_j, p_2, t_2\}$ and $\pi_2 = \{t_1, p_1, t_j, p_2, t_2\}$, then we have the weighted marking $M(\pi_1, \boldsymbol{m}_0) \geq M(\pi_2, \boldsymbol{m}_0)$.

If first *rule* 1 has been applied to remove $t_j$, the system in Fig.8(a) is obtained. Let us first consider the obtained place $p_a$ and $p'_a$. Without loss of generality, we should have: $\frac{g_1}{g_5} = \frac{w_1 \cdot w_5}{w_3 \cdot w_7} = \frac{g_2}{g_6} = \frac{w_2 \cdot w_5}{w_4 \cdot w_7}$, moreover, with the initial marking $\boldsymbol{m}'_0(p'_a) = g_1 \cdot M(\pi_1, \boldsymbol{m}_0)$ and $\boldsymbol{m}'_0(p_a) = g_2 \cdot M(\pi_2, \boldsymbol{m}_0)$, therefore, $\frac{\boldsymbol{m}'_0(p'_a)}{g_5} \geq \frac{\boldsymbol{m}'_0(p_a)}{g_6}$, $p'_a$ is implicit place. Then, it can be removed by applying *rule* 2. Similarly, for $p_b$ and $p'_b$, let $\frac{g_3}{g_7} = \frac{w_2 \cdot w_6}{w_4 \cdot w_8}$, $\frac{g_4}{g_8} = \frac{w_1 \cdot w_6}{w_3 \cdot w_8}$, $p'_b$ is also implicit and can be removed. The obtained system is shown in Fig.8(c).

If first *rule* 2 has been applied to remove $p_x$, the system in Fig.8(b) is obtained. Then by applying *rule* 1, $t_j$ is removed, it is clear that the same reduced system in 8(c) is achieved.

Now we know that the order of applying reduction rules is not important. Let $\Gamma_1$ and $\Gamma_2$ be two sequences of rules leading to two fully reduced systems

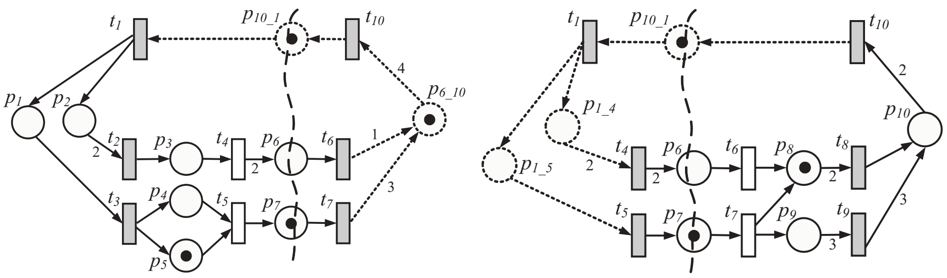
$\mathcal{S}_1$ and $\mathcal{S}_2$. It is clear that, the same number of *rule* 1 is applied in $\Gamma_1$ and $\Gamma_2$ (because applying *rule* 1 once, one transition between $T_{in}$ and $T_{out}$ is removed). From 1) and 2), we can transform the sequence $\Gamma_1$ to $\Gamma_1'$ by interchanging the order of adjacent rules, until all the instances of *rule* 1 are moved ahead of instances of *rule* 2. Assume that by applying all the instances of *rule* 1, the obtained system is $\mathcal{S}_1'$. On the other side, we can also transform the sequence $\Gamma_2$ to $\Gamma_2'$ by doing the same interchanging and assume that by applying all the instances of *rule* 1, the obtained system is $\mathcal{S}_2'$. Obviously, $\mathcal{S}_1'$ and $\mathcal{S}_2'$ are equivalent, and there are only places (but no transitions) left between $T_{in}$ and $T_{out}$. After that, the instances of *rule* 2 are applied to reduce implicit places in $\mathcal{S}_1'$ and $\mathcal{S}_2'$. If they are fully reduced, for sure the finally obtained systems are the same, i.e., $\mathcal{S}_1$ and $\mathcal{S}_2$ are equivalent. Therefore, the fully reduced system is unique. □

**Remark 3.11.** *In order to obtain the fully reduced system, we need to explore the paths between transitions. Concerning the computational complexity, it is suggested that before considering to apply rule 1, we should first apply rule 2 (or other possible methods) as much as possible, to remove the redundant implicit places, e.g., in Ex.3.9, rule 2 is first applied to remove a implicit place* $p_5$.

**Proposition 3.12.** *Let* $\mathcal{S}$ *be a continuous CFPN system, and* $\mathcal{S}_i$, $i = 1, 2$ *be its subsystems obtained by cutting with places* $B \in P$. $\mathcal{CS}_i$ *is the complemented subsystems obtained from* $\mathcal{S}$ *by substituting* $\mathcal{S}_j$, $j = 1, 2$, $j \neq i$ *with its abstraction. The firing sequences and reachable markings of* $S$ *are preserved in the complemented subsystems.*

*Proof.* Since the abstractions of subsystems are obtained by using the proposed reductions rules, it is a direct consequence of Proposition. 3.5 and 3.8. □

Sometimes for a complex system, it may be divided into more than two parts through several given sets of places. Therefore, this cut and complement process should be executed iteratively. For instance, the CFPN system in Fig.3(c) can be cut one more time with $B_2 = \{p_{10\_1}, p_6, p_7\}$. Using the same reduction process, two second level subsystems $\mathcal{CS}_{21}$ and $\mathcal{CS}_{22}$ are obtained, they are given in Fig.9.




(a) $\mathcal{CS}_{21}$ (b) $\mathcal{CS}_{22}$

Figure 9: Second level decomposition of $\mathcal{CS}_2$ with $B_2 = \{p_{10\_1}, p_6, p_7\}$

## 4. Decentralized Control of CFPN Systems

### 4.1. Computing the Control Laws

Using the decomposition method proposed in Section 3, we obtain the complemented subsystems whose behaviours are consistent with in the original system. The local control laws (minimal firing count vectors) driving subsystems to their corresponding final states can be computed separately. But consider the fact that buffers are essentially shared by neighboring subsystems, local control laws should be compatible with each other, more specifically, the interface transitions between two neighboring subsystems should be fired with the amount in both of them (this is not true in general, see Ex.4.1 for a example where local control laws are not compatible). For this purpose, a coordinator is introduced. Local controllers will send limited information (the local control law and the minimal T-*semiflow*) to the coordinator. Algorithms are proposed to compute the globally admissible control laws base on this information, without knowing the detailed structures of subsystems.

**Example 4.1.** *Let us consider the CFPN in Ex. 3.2 and the two obtained complemented subsystems in Fig.3(b) and Fig.3(c). The initial and final marking* $\boldsymbol{m}_0$, $\boldsymbol{m}_f$ *of the original system, and its corresponding minimal firing count vector* $\boldsymbol{\sigma}_{min}$ *is shown in Tab. 1. The minimal firing count vectors* $\boldsymbol{\sigma}^i_{min}$ *of* $\mathcal{CS}_i$ *for reaching the corresponding final marking* $\boldsymbol{m}^i_f$ *from* $\boldsymbol{m}^i_0$ *are computed separately, they are also given in Tab. 1. It can be observed that* $\boldsymbol{\sigma}^1_{min}$ *and* $\boldsymbol{\sigma}^2_{min}$ *are not compatible, because their interface transitions do not have the same firing counts, for instance,* $\boldsymbol{\sigma}^1_{min}(t_1) \neq \boldsymbol{\sigma}^2_{min}(t_1)$.

Let $\mathcal{S} = \langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be the original system, with $\boldsymbol{m}_f > 0$ the desired final

Table 1: Markings and firing count vectors

| $P$ | $\boldsymbol{m}_0$ ($\boldsymbol{m}_f$) | $\boldsymbol{m}_0^1$ ($\boldsymbol{m}_f^1$) | $\boldsymbol{m}_0^2$ ($\boldsymbol{m}_f^2$) | $T$ | $\boldsymbol{\sigma}_{min}$ | $\boldsymbol{\sigma}^1_{min}$ | $\boldsymbol{\sigma}^2_{min}$ |
|---|---|---|---|---|---|---|---|
| $p_1$ | 0 (0.4) | 0 (0.4) | 0 (0.4) | $t_1$ | 1.4 | 0.9 | 1.4 |
| $p_2$ | 0 (0.3) | 0 (0.3) | 0 (0.3) | $t_2$ | 0.55 | 0.3 | 0.55 |
| $p_3$ | 0 (0.3) | | 0 (0.3) | $t_3$ | 1 | 0.5 | 1 |
| $p_4$ | 0 (0.3) | | 0 (0.3) | $t_4$ | 0.25 | | 0.25 |
| $p_5$ | 1 (1.3) | | 1 (1.3) | $t_5$ | 0.7 | | 0.7 |
| $p_6$ | 0 (0.5) | | 0 (0.5) | $t_6$ | 0 | | 0 |
| $p_7$ | 1 (0.3) | | 1 (0.3) | $t_7$ | 1.4 | | 1.4 |
| $p_8$ | 1 (0.4) | | 1 (0.4) | $t_8$ | 1 | 0.5 | 1 |
| $p_9$ | 0 (0.2) | | 0 (0.2) | $t_9$ | 0.4 | 0.23 | 0.4 |
| $p_{10}$ | 0 (0.6) | 0 (0.6) | 0 (0.6) | $t_{10}$ | 0.8 | 0.3 | 0.8 |
| $p_{11}$ | 0 (0.2) | 0 (0.2) | | $t_{11}$ | 0.6 | 0.1 | |
| $p_{12}$ | 0 (0.1) | 0 (0.1) | | $t_{12}$ | 0.7 | 0.2 | |
| $p_{13}$ | 0 (0.1) | 0 (0.1) | | $t_{13}$ | 0.35 | 0.1 | |
| $p_{14}$ | 0 (0.3) | 0 (0.3) | | $t_{14}$ | 0.5 | 0 | |
| $p_{15}$ | 0 (0.1) | 0 (0.1) | | $t_{15}$ | 0.6 | 0.1 | |
| $p_{16}$ | 0 (0.1) | 0 (0.1) | | $t_{16}$ | 0.25 | 0 | |
| $p_{17}$ | 1 (0.1) | 1 (0.1) | | | | | |
| $p_{18}$ | 1 (0.2) | 1 (0.2) | | | | | |
| $p_{19}$ | 1 (0.1) | 1 (0.1) | | | | | |
| $p_{2\_8}$ | | 2 (2.1) | | | | | |
| $p_{3\_9}$ | | 1 (0.8) | | | | | |
| $p_{10\_1}$ | | | 1 (0.4) | | | | |

state. It is assumed that the original system is decomposed into $K$ subsystems, the following notations are used:

(1) $\boldsymbol{\sigma}_{min}$: the minimal firing count vector driving $\mathcal{S}$ to $\boldsymbol{m}_f$.

(2) $\mathcal{CS}_i = \langle \mathcal{CN}_i, \boldsymbol{m}_0^i \rangle$ be the complemented subsystems with corresponding final state $\boldsymbol{m}_f^i$, $i = 1, 2, ..., K$.

(3) $B^{(k1,k2)}$: the cutting places between $\mathcal{CS}_{k1}$ and $\mathcal{CS}_{k2}$.

(4) $U^{(k1,k2)}$: the interface transitions between $\mathcal{CS}_{k1}$ and $\mathcal{CS}_{k2}$.

(5) $\boldsymbol{x}^i$: the minimal T-*semiflow* in $\mathcal{CN}_i$, $i = 1, 2, ..., K$.

(6) $\boldsymbol{\sigma}^i_{min}$: the minimal firing count vector driving $\mathcal{CS}_i$ to $\boldsymbol{m}_f^i$, $i = 1, 2, ..., K$.

The original system is strongly connected and consistent, all the obtained complemented subsystems preserve these properties. It has been proved that the minimal T-*semiflow* and minimal firing count vector are unique in a strongly connected and consistent CFPN ([25, 26]), therefore, $\boldsymbol{\sigma}_{min}$ and $\boldsymbol{x}^i$

are unique, any firing count vector $\boldsymbol{\sigma}^i$ driving $\mathcal{CS}_i$ to its final state can be written as the following:

$$\boldsymbol{\sigma}^i = \boldsymbol{\sigma}^i_{min} + \alpha^i \cdot \boldsymbol{x}^i, \quad \alpha^i \geq 0 \tag{2}$$

Algorithms 1 is used by the coordinator controller. Non-negative value $\alpha^1$, $\alpha^2$, ..., $\alpha^K$ are obtained by solving a simple LPP. Then these values are sent back to local controllers. It is ensured that by updating the local control law from $\boldsymbol{\sigma}^i_{min}$ to $\boldsymbol{\sigma}^i_{min} + \alpha^i \cdot \boldsymbol{x}^i$, the interface transitions are fired in the same amounts in neghboring subsystems.

---

**Algorithm 1** Coordinator

---

**Input:** $\boldsymbol{\sigma}^i_{min}$, $\boldsymbol{x}^i$, $i = 1, 2, .., K$
**Output:** $\alpha^i$, $i = 1, 2, .., K$

1: Receive $\boldsymbol{\sigma}^i_{min}$, $\boldsymbol{x}^i$ from local controllers
2: Compute $\alpha^i$ by solving LPP:

$$\begin{aligned} \min \quad & \sum_{i=1}^{K} \alpha^i \\ \text{s.t.} \quad & \boldsymbol{\sigma}^{i_1}_{min}(t_j) + \alpha^{i_1} \cdot \boldsymbol{x}^{i_1}(t_j) = \boldsymbol{\sigma}^{i_2}_{min}(t_j) + \alpha^{i_2} \cdot \boldsymbol{x}^{i_2}(t_j), \forall t_j \in U^{(i_1,i_2)} \\ & \qquad \forall i_1, i_2 \in \{1, 2, ..., K\}, \mathcal{CS}_{i_1} \text{ and } \mathcal{CS}_{i_2} \text{ are neighbors.} \\ & \alpha^i \geq 0, i = 1, 2, ..., K \end{aligned} \tag{3}$$

3: Send $\alpha_i$ to $\mathcal{CS}_i$;

---

Given a reachable final state $\boldsymbol{m}_f$, LPP (3) is feasible. Let $\boldsymbol{\sigma}$ be a firing count vector driving $\mathcal{S}$ to $\boldsymbol{m}_f$. The projections of $\boldsymbol{\sigma}$ corresponding to $\mathcal{CS}_{i_1}$ and $\mathcal{CS}_{i_2}$, denoted by $\boldsymbol{\sigma}^{i_1}$ and $\boldsymbol{\sigma}^{i_2}$, are fireable and can drive $\mathcal{CS}_{i_1}$, $\mathcal{CS}_{i_2}$ to their corresponding final states. Obviously, the transitions in $U^{(i_1,i_2)}$ are fired in the same amounts in $\boldsymbol{\sigma}^{i_1}$ and $\boldsymbol{\sigma}^{i_2}$, so there exist $\alpha^{i_1}$ and $\alpha^{i_2}$, satisfying the constraints of LPP (3).

**Proposition 4.2.** *Let $\alpha^i$ be the value obtained by using Alg.1 and $\boldsymbol{\sigma}^i = \boldsymbol{\sigma}^i_{min} + \alpha^i \cdot \boldsymbol{x}^i$, $i = 1, 2, ..., K$ be the local control laws of $\mathcal{CS}_i$. The global control law $\boldsymbol{\sigma}$ obtained by merging all the local ones, is the minimal firing count vector driving $\mathcal{S}$ to $\boldsymbol{m}_f$.*

*Proof.* It is trivial that $\boldsymbol{\sigma}$ can drive $\mathcal{S}$ to $\boldsymbol{m}_f$. If $\boldsymbol{\sigma}$ is not the minimal one, some amounts of T-*semiflow* can be subtracted, obtaining a contradiction with the objective function of LPP (3). □

Algorithms 2 is used by local controllers. The minimal firing count vector $\boldsymbol{\sigma}^i_{min}$ of subsystem $\mathcal{CS}_i$ is first computed separately in the corresponding local controller and then sent to the coordinator together with the minimal T-*semiflow*. After the updating information, $\alpha^i$, is received, the controller of $\mathcal{CS}_i$ can be implemented independently with the control law $\boldsymbol{\sigma}^i_{min} + \alpha^i \cdot \boldsymbol{x}^i$. In this work, a ON-OFF control strategy (presented in the next section) is used.

Let us observe that the only information required by the coordinator are the local control laws and minimal firing count vectors, therefore all computations are done locally with very low communication costs.

**Algorithm 2** Local Controller $i$

**Input:** $\mathcal{CN}_i$, $\boldsymbol{m}^i_0$, $\boldsymbol{m}^i_f$
**Output:** $\boldsymbol{\sigma}^i$

1: Compute $\boldsymbol{\sigma}^i_{min}$ driving the system to $\boldsymbol{m}^i_f$;
2: Compute the minimal T-*semiflow* $\boldsymbol{x}^i$;
3: Send $\boldsymbol{\sigma}^i_{min}$ and $\boldsymbol{x}^i$ to the coordinator;
4: Receive $\alpha^i$ from the coordinator;
5: Update $\boldsymbol{\sigma}^i \leftarrow \boldsymbol{\sigma}^i_{min} + \alpha^i \cdot \boldsymbol{x}^i$;
6: Apply ON-OFF control;

### *4.2. Minimum-time ON-OFF Controller*

The globally admissible local control laws are obtained using the method presented in Section 4.1, in this section we will discuss how to drive the system to its final state, in particular, an ON-OFF strategy will be applied which ensures the minimum-time state evolution.

The final state of a system may be reached by following different firing count vectors. In CFPN systems if the final state is reached in minimum-time, the system should follow the minimal one, which is unique in (strongly connected and consistent) CFPN.

Since in a CFPN system every place has only one output transition, the firing of one transition will not disable another. Furthermore, if two transitions $t_1$ and $t_2$ are enabled at the same time, the order of firing is not important (i.e., both sequence $t_1 t_2$ and $t_2 t_1$ are fireable). Based on these observations, it can be concluded that if there exists a transition that has not been fired with the maximal amount at one moment, certain amount of its firings may be moved ahead in order to reach the maximal amount.

**Example 4.3.** *Let us consider the trivial CFPN system in Fig. 10 and assume* $\boldsymbol{m}_f = [0.2\ 0.5\ 0.3]^T$*, the minimal firing count vector for reaching the final state is* $\boldsymbol{\sigma} = [0.8\ 0.3\ 0]^T$*. Following this vector, one firing sequence may be* $\sigma_1 = t_1(0.5)t_2(0.3)t_1(0.3)$*. It can be observed that* $t_1$ *is 1-enabled under* $\boldsymbol{m}_0$*, and the required amount that* $t_1$ *should fire is* 0.8*. Therefore, we can fire* $t_1$ *more than* 0.5 *in the beginning. In particular, the final marking is also reached by firing sequence* $\sigma_2 = t_1(0.8)t_2(0.3)$*.*

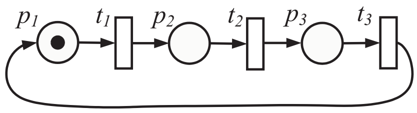



Figure 10: A CFPN system with $\boldsymbol{m}_0 = [1\ 0\ 0]$.

The ON-OFF controller is obtained if every transition is fired as fast as possible at any moment until the required minimal firing count is reached. Under the continuous time setting, the control action for transition $t_j$ at time $\tau$ is given by:

$$\boldsymbol{u}(t_j, \tau) = \begin{cases} 0 & \text{if } \int_0^{\tau^-} \boldsymbol{w}(t_j, \delta)\mathrm{d}\delta < \boldsymbol{\sigma}(t_j) \quad (a) \\ \boldsymbol{f}(t_j, \tau) & \text{if } \int_0^{\tau^-} \boldsymbol{w}(t_j, \delta)\mathrm{d}\delta = \boldsymbol{\sigma}(t_j) \quad (b) \end{cases} \tag{4}$$

where $\boldsymbol{\sigma}$ is the minimal firing count vector and $\boldsymbol{w}(t_j, \delta)$ is the controlled flow of $t_j$ at time $\delta$, $\boldsymbol{f}(t_j, \tau)$ are the uncontrolled flow at time $\tau$. (a) means that if $\boldsymbol{\sigma}(t_j)$ is not reached then $t_j$ is completely *ON*, i.e., $\boldsymbol{u}(t_j, \tau) = 0$; else (b), $t_j$ is completely *OFF*, i.e., $\boldsymbol{u}(t_j, \tau) = \boldsymbol{f}(t_j, \tau)$.

It is proved in Appendix A that by applying this ON-OFF controller to a CFPN system, the final state can always be reached in minimum-time. It should be noticed that for continuous timed system under infinite server semantics, once a place is marked it will take infinite time to be emptied (like the discharging of a capacitor in an electrical RC-circuit). Therefore, if there exist places that are emptied during the trajectory to $\boldsymbol{m}_f$, the final marking is reached at the limit, i.e., in infinite time. Obviously, if $\boldsymbol{m}_f > 0$, this situation does not happen.

The advantage of this ON-OFF control strategy is its low computational complexity: only the minimal firing count vector needs to be computed, and it is polynomial time. On the other side, the the minimum-time state evolution is guaranteed.

## 5. Case Study

The CPN in Fig.11 shows the model of a manufacturing system that consists of three work stations: WS_1 and WS_2 and WS_3. There are two type of raw materials A and B, processed by WS_1 and WS_2 respectively. The obtained semi-products are deposited in buffers and will be finally assembled in WS_3 to make the final products.

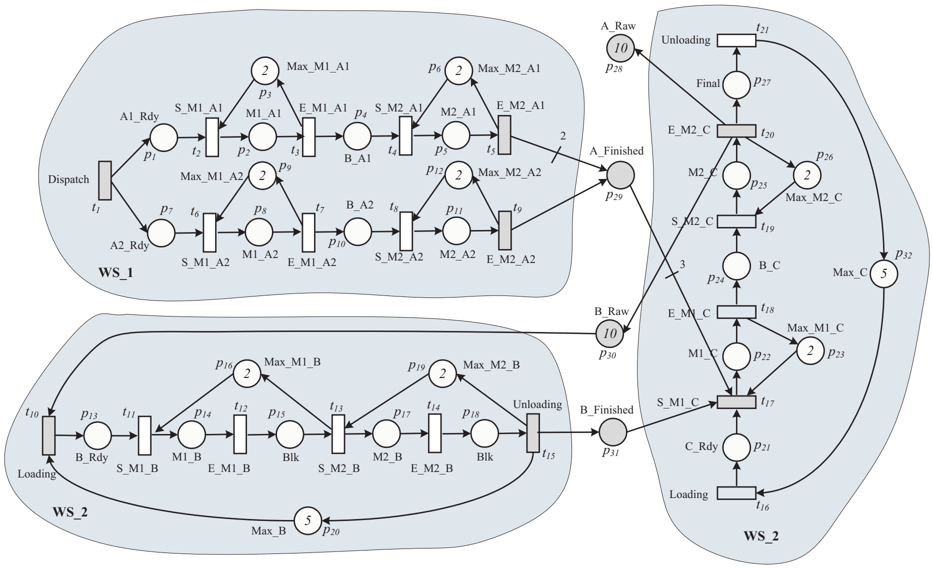



Figure 11: A manufacturing system with three work stations.

Tab.2 gives the interpretations of the model:

It is assumed that both types of materials have quantities equal to 10, while two machines are available for any processing, production lines in WS_2 and WS_3 have maximal capabilities equal to 5. The firing rates are: $\lambda_7 = \lambda_9 = 1/2$, $\lambda_{13} = \lambda_{15} = 1/3$, $\lambda_{18} = \lambda_{20} = 1/4$ and for other transitions, all equal to 1. Under this setting, the maximal throughput of transition E_M2_C ($t_{20}$, which models the machine that produces the final product) in the steady state is 0.29 ([3]). We will design the control laws for reaching this state using the proposed decentralized control method.

It is natural to cut the original system into three subsystems $\mathcal{CS}_1$ to $\mathcal{CS}_3$ corresponding to work stations WS_1 to WS_3. The buffer places are $B^{(1,2)} = \{p_{28}, p_{29}\}$, $B^{(2,3)} = \{p_{30}, p_{31}\}$ and the interface transition are $U^{(1,2)} = \{t_1, t_5, t_9, t_{17}, t_{20}, \}$, $B^{(2,3)} = \{t_{10}, t_{15}, t_{17}, t_{20}\}$. The complemented subsystems

Table 2: The interpretation of the PN model in Fig.5

| **Labels** | **Interpretation** |
|---|---|
| x_Rdy | material x is ready |
| Mx_y | machine x processing y |
| Max_Mx_y | the free machine x processing y |
| Blk | blocked |
| B_x | the buffer of semi-product x |
| Max_x | the maximal allowed capacity of x |
| final | the final product |
| x_Raw | raw material x |
| x_finish | the semi-product x finished |
| S_Mx_y | machine x starts to process y |
| E_Mx_y | machine x finishes the process of y |
| Dispatch | dispatch raw materials to producing lines |

are shown in Fig.12, the final states of subsystems and their corresponding minimal firing count vectors are shown in Tab.3.

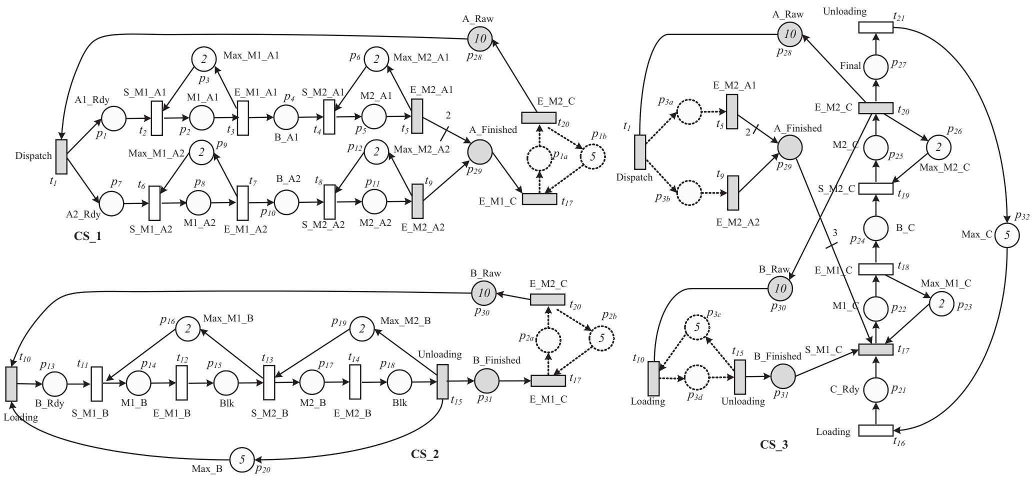



Figure 12: The complemented subsystems obtained from the model in Fig.??

In this specific example, the T-*semiflows* of subsystems are unit vectors $\boldsymbol{I}$, by applying Alg.1, the solution is quite straightforward: $\alpha^1 = \alpha^3 = 0$ and $\alpha^2 = 0.29$. So the control law of $CS^2$ should be updated to $\boldsymbol{\sigma}^2_{min} + 0.29 \cdot \boldsymbol{I}$, the control laws of $CS^1$ and $CS^3$ are $\boldsymbol{\sigma}^1_{min}$ and $\boldsymbol{\sigma}^3_{min}$, respectively. By applying

Table 3: Final states and minimal firing count vectors

| $\mathcal{CS}_1$ (WS_1) | | | | $\mathcal{CS}_2$(WS_2) | | | | $\mathcal{CS}_3$(WS_3) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $p$ | $\boldsymbol{m}_f^1$ | $t$ | $\boldsymbol{\sigma}_{min}^1$ | $p$ | $\boldsymbol{m}_f^1$ | $t$ | $\boldsymbol{\sigma}_{min}^1$ | $p$ | $\boldsymbol{m}_f^1$ | $t$ | $\boldsymbol{\sigma}_{min}^1$ |
| $p_1$ | 0.29 | $t_1$ | 5.86 | $p_{13}$ | 0.29 | $t_{10}$ | 5.42 | $p_{21}$ | 0.29 | $t_{16}$ | 3.14 |
| $p_2$ | 0.29 | $t_2$ | 5.57 | $p_{14}$ | 0.29 | $t_{11}$ | 5.14 | $p_{22}$ | 1.14 | $t_{17}$ | 2.86 |
| $p_3$ | 1.71 | $t_3$ | 5.29 | $p_{15}$ | 0.86 | $t_{12}$ | 4.85 | $p_{23}$ | 0.86 | $t_{18}$ | 1.71 |
| $p_4$ | 0.29 | $t_4$ | 5.00 | $p_{16}$ | 0.86 | $t_{13}$ | 4.00 | $p_{24}$ | 0.29 | $t_{19}$ | 1.43 |
| $p_5$ | 0.29 | $t_5$ | 4.71 | $p_{17}$ | 0.29 | $t_{14}$ | 3.71 | $p_{25}$ | 1.14 | $t_{20}$ | 0.29 |
| $p_6$ | 1.71 | $t_6$ | 1.43 | $p_{18}$ | 0.86 | $t_{15}$ | 2.85 | $p_{26}$ | 0.86 | $t_{21}$ | 0.00 |
| $p_7$ | 4.43 | $t_7$ | 0.86 | $p_{19}$ | 0.86 | $t_{17}$ | 2.57 | $p_{27}$ | 0.29 | $t_1$ | 5.86 |
| $p_8$ | 0.57 | $t_8$ | 0.57 | $p_{30}$ | 4.57 | $t_{20}$ | 0.00 | $p_{28}$ | 4.43 | $t_5$ | 4.71 |
| $p_9$ | 1.43 | $t_9$ | 0.00 | $p_{31}$ | 0.29 | | | $p_{29}$ | 0.86 | $t_9$ | 0.00 |
| $p_{10}$ | 0.29 | $t_{17}$ | 2.86 | $p_{2a}$ | 2.57 | | | $p_{30}$ | 4.57 | $t_{10}$ | 5.71 |
| $p_{11}$ | 0.57 | $t_{20}$ | 0.29 | $p_{2b}$ | 2.43 | | | $p_{31}$ | 0.29 | $t_{15}$ | 3.14 |
| $p_{12}$ | 1.43 | | | | | | | $p_{32}$ | 1.86 | | |
| $p_{28}$ | 4.43 | | | | | | | $p_{3a}$ | 1.14 | | |
| $p_{29}$ | 0.86 | | | | | | | $p_{3b}$ | 5.86 | | |
| $p_{1a}$ | 2.57 | | | | | | | $p_{3c}$ | 2.43 | | |
| $p_{1b}$ | 2.43 | | | | | | | $p_{3d}$ | 2.57 | | |

the ON-OFF controller using these laws, the final state of the system is reached in 15.17 time unites, which is the minimum-time.

## 6. Conclusion

This paper focuses on the minimum-time decentralized control of Choice-Free continuous Petri nets. The addressed problem is how to drive the system from an initial state to a desired final one.

We assume the original system can be viewed as divided by given sets of places. It should be noticed that the number of interface transitions varies, depending on how the cutting places are chosen. It may further influence the computational complexity, because the size of complemented subsystems are bigger if we use a cut that introduces many interface transitions. Two rules are proposed to reduce subsystems, more specifically, the paths between interface transitions can be reduced to some places. In the worst case, the number of places may not be reduced, but since all intermediate transitions in the paths are removed, the subsystems are still highly simplified in general, obtaining their abstractions. A coordinator is introduced to reach the agreement among neighboring subsystems, by solving a simple LPP. The coordinator does not need to know the detailed structures of subsystems: only limited information — the minimal firing count vector and minimal T-*semiflow*, are exchanged, ensuring the low communication cost. Applying the ON-OFF strategy in each subsystem, the global final state is reached in minimum-time.

As a future work, this decentralized control framework will be considered to general PN systems. Then two potential problems need to be addressed: 1) the minimal firing count vectors may not be unique, and some of them can cause the reachability problem of the final state; 2) the ON-OFF controller may not be directly applied, because in some cases systems may get a deadlock when pure ON-OFF strategy is used.

## Appendix A.

By sampling the continuous-time TCPN system with a *sampling period* $\Theta$, we obtain the discrete-time TCPN ([8]) given by:

$$\begin{aligned} &\boldsymbol{m}_{k+1} = \boldsymbol{m}_k + \boldsymbol{C} \cdot \boldsymbol{w}_k \cdot \Theta \\ &0 \leq \boldsymbol{w}_k \leq \boldsymbol{f}_k \end{aligned} \tag{A.1}$$

Here $\boldsymbol{m}_k$ and $\boldsymbol{w}_k$ are the marking and controlled flow at sampling instant $k$, i.e., at $\tau = k \cdot \Theta$.

It is proved in [8] that if the sampling period satisfies (A.2), the reachability spaces of discrete-time and continuous-time PN systems are the same.

$$\forall p \in P : \sum_{t_j \in p^{\bullet}} \lambda_j \cdot \Theta < 1 \tag{A.2}$$

It is assumed that the sampling period $\Theta$ is small enough to satisfy (A.2), and the detailed proof is given in the setting of discrete-time. It can be naturally extended to continuous-time systems.

**Proposition Appendix A.1.** *Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \Theta, \boldsymbol{m}_0 \rangle$ be a discrete-time continuous CFPN system and $\boldsymbol{m}_f$ be a reachable final marking with the corresponding minimal firing count vector $\boldsymbol{\sigma}$. The ON-OFF controller is the minimum-time controller driving the system to $\boldsymbol{m}_f$.*

*Proof.* We will prove that whenever there exists a controller $\boldsymbol{G}$ driving the system to $\boldsymbol{m}_f$, it consumes at least the time of the ON-OFF controller. This will imply that the ON-OFF controller is the minimum-time controller.

Assume a non ON-OFF controller $\boldsymbol{G}$. Hence, there exists a transition $t_j$ that is not *sufficiently fired*, i.e., not fired as much as possible, in a sampling period $k$. In other words, $t_j$ has to be fired later in a sampling period $l$, $l > k$. Let us assume, without loss of generality, that $t_j$ is not fired between the $k^{th}$ and the $l^{th}$ sampling period. It is always possible to "move" some amounts of firings from the $l^{th}$ sampling period to the $k^{th}$ one until $t_j$ becomes sufficiently fired in $k$. According to Property 2.2 this move does not affect the fireability of the other transitions. Iterating the procedure, all transitions can be made sufficiently fired in all sampling periods and the obtained controller is an ON-OFF one.

Obviously, the number of discrete-time periods necessary to reach the final marking after moving firings from a sampling period $l$ to another one $k$ with $k \leq l$ is at least the same. Hence the number of sampling steps is not higher with the ON-OFF controller. □

If we take sampling period $\Theta$ tending to 0, the ON-OFF controller for continuous-time system (shown in (4.2)) is obtained. According to Proposition Appendix A.1, this is the minimum-time controller.

## References


[1] R. David, H.Alla, Discrete, Continuous and Hybrid Petri Nets, Springer, Berlin, 2010, 2nd edition.

[2] M. Silva, L. Recalde, On fluidification of Petri net models: from discrete to hybrid and continuous models, Annual Reviews in Control 28 (2) (2004) 253–266.

[3] C. Mahulea, A. Ramirez, L. Recalde, M.Silva, Steady state control reference and token conservation laws in continuous petri net systems, IEEE Transactions on Automation Science and Engineering 5 (2) (2008) 307–320.

[4] J. Júlvez, L. Recalde, M. Silva, On reachability in autonomous continuous Petri net systems, in: G. Goos, J. Hartmanis, J. van Leeuwen (Eds.), Applications and Theory of Petri Nets 2003, Vol. 2679, Eindhoven, The Netherlands, 2003, pp. 221–240.

[5] A. Amrah, N. Zerhouni, A. E. Moudni, On the control of manufacturing lines modelled by controlled continuous Petri nets, Journal of Systems Science 29 (2) (1998) 127–137.

[6] S. Hennequin, D. Lefebvre, A. El-Moudni, Fuzzy control of variable speed continuous Petri nets, in: 38th IEEE Conference on Decision and Control, Phoenix, USA, 1999.

[7] W. Lia, T. Gu, Optimization and control of production system based on interval speed continuous Petri nets, in: IEEE International Conference on Systems, Man and Cybernetics, 2005, pp. 1212 – 1217.

[8] C. Mahulea, A. Giua, L. Recalde, C. Seatzu, M. Silva, Optimal model predictive control of Timed Continuous Petri nets, IEEE Trans. on Automatic Control 53 (7) (2008) 1731 – 1735.

[9] J. Xu, L. Recalde, M. Silva, Tracking control of join-free timed continuous Petri net systems under infinite server semantics, Discrete Event Dynamic Systems 18 (2) (2008) 263–288.

[10] H. Apaydin-Ozkan, J. Júlvez, C. Mahulea, M. Silva, Approaching Minimum Time Control of Timed Continuous Petri nets, Nonlinear Analysis: Hybrid Systems 5 (2) (2011) 136–148.

[11] J. Sandell, N., P. Varaiya, M. Athans, M. Safonov, Survey of decentralized control methods for large scale systems, IEEE Transactions on Automatic Control 23 (2) (1978) 108 – 128.

[12] T. Yoo, S. Lafortune, A General Architecture for Decentralized Supervisory Control of Discrete-Event Systems, Discrete Event Dynamic Systems-Theory and Applications 12 (3) (2002) 335–377.

[13] S. Tong, H.-X. Li, G. Chen, Adaptive fuzzy decentralized control for a class of large-scale nonlinear systems, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34 (1) (2004) 770 – 775.

[14] Z. Duan, J. Wang, G. Chen, L. Huang, Stability analysis and decentralized control of a class of complex dynamical networks, Automatica 44 (4) (2008) 1028 1035.

[15] M. V. Iordache, P. J. Antsaklis, Decentralized Supervision of Petri Nets, IEEE Transactions on Automatic Control 51 (2) (2006) 376–391.

[16] F. Basile, A. Giua, C. Seatzu, Supervisory Control of Petri Nets with Decentralized Monitor Places, in: 2007 American Control Conference, New York, USA, 2007, pp. 4957 – 4962.

[17] A. Aydin, Decentralized structural control approach for Petri nets, Control and Cybernetics 36 (1) (2007) 143–159.

[18] H. Apaydin-Ozkan, J. Julvez, C. Mahulea, M. Silva, A Control Method for Timed Distributed Continuous Petri nets, in: 2010 American Control Conference, Baltimore, USA, 2010.

[19] L. H. Landweber, E. L. Robertson, Properties of conflict-free and persistent Petri nets, J. ACM 25 (3) (1978) 352–364.

[20] C. Mahulea, L. Recalde, M. Silva, Basic Server Semantics and Performance Monotonicity of Continuous Petri Nets, Discrete Event Dynamic Systems: Theory and Applications 19 (2) (2009) 189 – 212.

[21] E. Teruel, P. Chrzastowski-Wachtel, J. Colom, M. Silva, On weighted T-systems, in: K. Jensen (Ed.), Application and Theory of Petri nets. LNCS, Vol. 616, Berlin, Germany, 1992, pp. 348–367.

[22] J. Campos, J. Colom, H. Jungnitz, M. Silva, Approximate Throughput Computation of Stochastic Marked Graphs, IEEE Transactions on Software Engineering 20 (7) (1994) 526–535.

[23] G. Berthelot, Lri-Iie, Checking properties of nets using transformations, Advances in Petri Nets 1985: Lecture Notes in Computer Science 222 (1986) 19–40.

[24] M. Silva, E. Teruel, J. Colom, Linear algebraic and linear programming techniques for the analysis of P/T net systems, LCNS 1 (1941) (1998) 309–373.

[25] E. Teruel, J. M. Colom, M. Silva, Choice-free Petri nets: A model for deterministic concurrent systems with bulk services and arrivals, IEEE Trans. on Systems, Man, and Cybernetics 27 (1) (1997) 73–83.

[26] L. Wang, C. Mahulea, J. Júlvez, M. Silva, Minimum-time Control for Structurally Persistent Continuous Petri Nets, in: 49th IEEE Conference on Decision and Control, Atlanta, Georgia USA, 2010, pp. 2771–2776.