Minimum-time Decentralized Control of Choice-Free Continuous Petri Nets[*] -draft-

Liewei Wang, Cristian Mahulea, Jorge Júlvez and Manuel Silva[†]

December 23, 2014

Abstract

This paper considers the problem of reaching a desired final state from a given initial one. Models are assumed to be Choice-Free continuous Petri nets. A minimum-time decentralized control scheme is proposed. The original system is cut into disconnected *subsystems* by a set of places (*buffers*). Local control laws (*minimal firing count vectors*) 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 behaviors (firing sequences and consequently, the reachable markings) from the original ones, and 2) since the buffer places are essentially shared by more than one subsystem, there must be an agreement among the neighboring 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 *coordinator controller* is introduced, and several algorithms are proposed to reach the agreement. The coordinator design is rather simple, because it does not need to know the detailed states and structures of subsystems. Finally, by applying an ON-OFF control strategy in each subsystem, the final state is ensured to be reached in minimum-time.

Published as:
L. Wang, C. Mahulea, J. Julvez, and M. Silva, "Minimum-time Decentralized Control of Choice-Free Continuous Petri Nets," Nonlinear Analysis: Hybrid Systems, vol. 7, no. 1, pp. 39-53, February 2013. DOI: 10.1016/j.nahs.2012.07.004

[*]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.

[†]The authors are with are with the Aragón Institute of Engineering Research (I3A), University of Zaragoza, Maria de Luna 1, 50018, Spain. E-mail: {*lwwang,cmahulea,julvez,silva*}*@unizar.es*.

1 Introduction

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 *fluidization* can be used.

Continuous PN (CPN) [**?**, **?**] 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. A simple and interesting way to introduce time to CPN is to assume that time is associated to transitions.

1.1 Problem statement

Let us consider a large scale discrete event dynamic systems, e.g., a complex transportation system connecting cities from different countries. The distributed physical deployment of the system often makes it impossible to implement a centralized controller that knows the detailed structures and the current state of all subsystems. A more practical way to proceed is to have local controllers allocated in each subsystem, which is the essence of decentralized control. Besides overcoming these physical constraints, the decentralized control is fault tolerant because the control decision is performed in a decentralized way, so even when control faults occur in some subsystems, other local controllers are still able to provide suitable control laws, although possible sub-optimal. The intersections among neighboring subsystems (in our case, modeled by places) play an important role in facilitating the interaction and communication between neighboring subsystems.

In this work, the results presented in [**?**] for marked graphs are extended to Choice-Free PN. It is assumed that the original system modeled by CPN is cut into disconnected subsystems by sets of places (buffers), and the addressed problem is to compute the control law and drive the system from an initial state to a 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 behaviors from the original ones, e.g., properties like liveness or boundedness in the original system may not be preserved; and 2) since the buffer places are essentially shared by more than one subsystem, there must be an agreement among the neighboring 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 *coordinator 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 an 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.

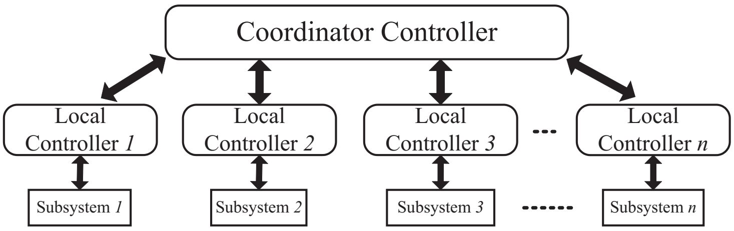


Figure 1: System Structures

1.2 Related work

Many works can be found in the literature about the control of different classes of timed CPN, e.g., [?, ?, ?]. For the kind of timed CPN under *infinite server semantics*, several control approaches have been considered. In [?], the optimal steady state control problem is studied. Model Predictive Control is used for optimal control problem in [?] assuming a discrete-time model. In [?], a Lyapunov-function-based dynamic control algorithm is studied, while in [?] a heuristics for minimum-time control is proposed. In this work, a minimum-time control problem of timed *Choice-Free CPN* under *infinite server semantics* is considered.

Decentralized control has been extensively explored in recent decades for complex dynamic systems (e.g., [?, ?, ?, ?]), in which multiple controllers may be allocated to subsystems. In the context of decentralized control on discrete PN, some approaches have been proposed. The centralized admissibility concept was extended to d-admissibility for the decentralized setting in [?]; based on the d-admissibility concept, two suboptimal methods to design decentralized supervisors are proposed. Under certain assumptions, the methods in [?] focused on global state specifications given in terms of Generalized Mutual Exclusion Constraints (GMECs) and on a control architecture without central coordinator and communication between local supervisors. In [?], 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.

Different from the methods in [?, ?] for discrete systems, which focus on enforcing states to satisfy certain constraints (specifications), we address the problem of driving the continuous 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 [?], 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 added). On the other hand, its control structures are also strongly dependent on the desired markings.

The decentralized target marking control problem of continuous Petri nets has been considered in [?, ?]. Contribution [?] considers continuous models composed by several subsystems that communicate through buffers (modeled by places). By executing the proposed algorithm iteratively in each subsystem, their respective target markings are reached and then maintained. This work contain two significant improvements with respect to [?]. Firstly, we do not assume that subsystems are strongly connected. Secondly, the globally admissible control laws are achieved by a simple coordinator, therefore the iterative process executed in subsystems is not needed anymore. A similar and more general structure of subsystems (called *modules*), is consider in [?] where an *affine control* is applied to each subsystem, driving the system to a positive defined final state. One important difference of our method with respect to [?] is related to the communication strategy: while in [?] the coordinator needs to exchange information with subsystems during each time step, in this work, once the agreement is achieved, all the subsystems work independently, so no communication is necessary.

There exist different ways to partition a large scale system in subsystems. This may be done by partitioning the sets of places and transitions as in [?, ?], or by explicitly cutting through a set of places [?, ?] or transitions [?]. In this paper, subsystems are first obtained by cutting through a set of (buffer) places, then an abstraction and complementing process is applied. The obtained complemented subsystems have identical firing sequences to those of the original system. Thus, they can also be used to solve other interesting problems, for example, as in [?], for throughput approximations.

This paper is organized as follows: Section 2 briefly recalls some basic concepts of CPN. Section 3 introduces the decomposition method for Choice-Free PN and proposes two reduction rules in order to obtain complemented subsystems. Section 4 proposes the approach for decentralized control of Choice-Free PN system. In section 5, we illustrate the proposed methods by using a simple 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 [?, ?] 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 α, 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 σ, 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 σ. A firing count vector $\boldsymbol{\sigma}$ driving the system to $\boldsymbol{m}$ is said to be *minimal*, if for any T-*semiflow* $\boldsymbol{x}$, $||\boldsymbol{x}|| \nsubseteq ||\boldsymbol{\sigma}||$.

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 (the firing of one transition will not disable another), i.e., it is *conflict-free* [**?**, **?**]. 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, therefore 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 [**?**, **?**]. 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 [**?**]. 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 τ is the product of its firing rate, λ_j, and its enabling degree at $\boldsymbol{m}(\tau)$:

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

2.2 Gains and Weighted Markings

The gain of a directed path was introduced in [**?**] 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 π 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 π 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 π 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 π, $M(\pi, \boldsymbol{m})$ can be interpreted as the number of firings t_{in} is required to be fired to reach $\boldsymbol{m}$, in the case that π 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 $G(\pi) \cdot M(\pi, \boldsymbol{m})$.

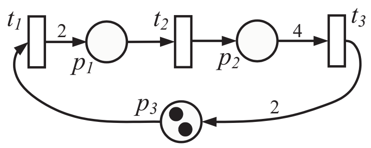


Figure 2: A simple CFPN system with $\boldsymbol{m}_0 = [0\ 0\ 2]^T$

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 π 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 π under $\boldsymbol{m}$ is $M(\pi, \boldsymbol{m}) = 1$.

Assume 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 (maximal) firing flow of transitions (defined for the *uncontrolled* systems) [**?**]. This means that 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 behavior 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 $\boldsymbol{u}(t_j) = 0$).

3 Structural Decomposition of CFPN systems

In this section, a structural decomposition approach for CFPN is introduced, obtaining subsystems that have firing sequences identical to those of the original system.

Given a large scale system, it may be divided into several parts, for example, due to its physical deployments. Here we assume 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 behaviors 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 firing sequences and consequently, the reachable markings of the original system are preserved.

3.1 Cutting

Here the structural cutting method developed in [**?**] is extended to CFPN. In order to simplify the notation, we assume that the system is cut into two parts. This is not a limitation, since each part can be further divided into two more 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) $P_1 \cup B = {}^\bullet T_1 \cup T_1{}^\bullet$, $P_2 \cup B = {}^\bullet T_2 \cup T_2{}^\bullet$

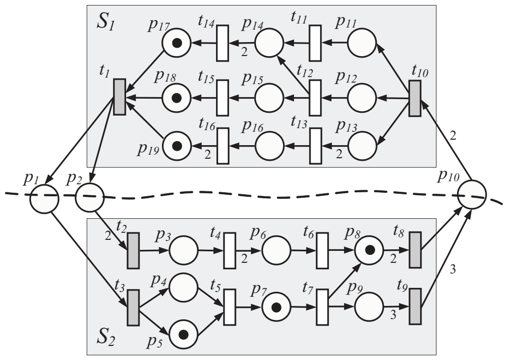
(4) $T_1 = {}^\bullet P_1 \cup P_1{}^\bullet$, $T_2 = {}^\bullet P_2 \cup P_2{}^\bullet$

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

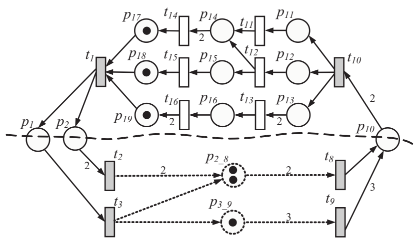
Example 3.2. *Fig. 3(a) shows a CFPN system. The set of places $B = \{p_1, p_2, p_{10}\}$ is a cut decomposing the original system into two subsystems, $\mathcal{S}_1$ and $\mathcal{S}_2$, where the interface transitions are $U_1 = \{t_1, t_{10}\}$ and $U_2 = \{t_2, t_3, t_8, t_9\}$.*

3.2 Complemented Subsystems

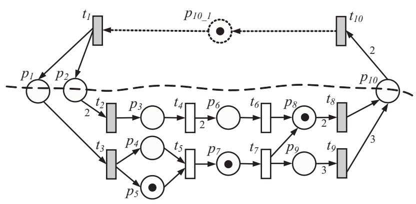
Due to the cut, different behaviors can be introduced, because subsystems become disconnected from the remaining parts. For instance, the net system in Fig.3(a) is live and bounded. After cutting by $B = \{p_1, p_2, p_{10}\}$, both obtained subsystems $\mathcal{S}_1$ and $\mathcal{S}_2$ become unbounded.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



(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 $\mathcal{CS}_2$

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 paths in $\mathcal{S}_1$ between interface transitions 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, it is 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 equivalent to strongly connected and consistent [**?**]).

Transition Reduction Rule (T-RR). *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 $t_{in} \in {}^\bullet P_{in} \cup$

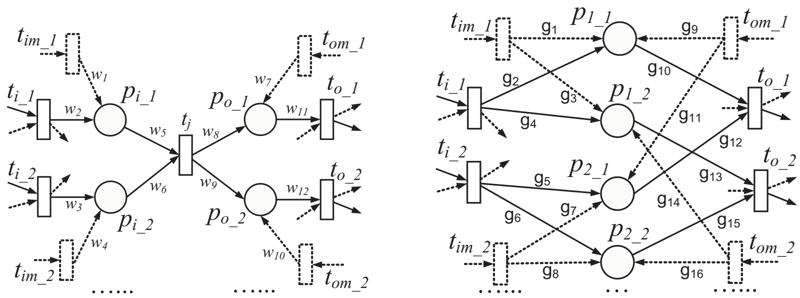
${}^\bullet P_{out}$, $t_{out} \in P_{out}{}^\bullet$, $\pi(t_{in}, t_{out})$ and $\pi'(t_{in}, t_{out})$ are the paths from t_{in} to t_{out}, in $\mathcal{S}$ and $\mathcal{S}'$ respectively.*

(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 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 behaviors 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 T-RR, the weights on the arcs connecting with p, are constrained to naturals, and have the greatest common divisor equal to one. In this way, the obtained system is uniquely (structurally) determined.

Example 3.4. *Consider the CFPN system $\mathcal{S}$ in Fig. 4(a), it is shown how to reduce t_j by applying T-RR. 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: Transition reduction rule (T-RR): reducing t_j

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 paths 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, σ be a firing sequence in $\mathcal{S}$. Sequence ς is said to be the projection of σ from $\mathcal{S}$ to $\mathcal{S}'$ when ς is obtained from σ 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 T-RR, removing transition t_j. Assume σ is a firing sequence of $\mathcal{S}$, and ς is its projection to $\mathcal{S}'$. Then σ is fireable in $\mathcal{S}$ if and only if ς is fireable in $\mathcal{S}'$.*

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

Consider T-RR applied in Fig. 4 to reduce transition t_j and its input/output places. Let us assume, without loss of generality that, the firing sequence in $\mathcal{S}$ is $\sigma = t_{i\_1}(\alpha_1)t_{i\_2}(\alpha_2)\ t_j(\beta)t_{im\_1}(\alpha_3)t_{o\_1}(\alpha_4)$, and its projection to the reduced system $\mathcal{S}'$ is $\varsigma = t_{i\_1}(\alpha_1)\ t_{i\_2}(\alpha_2)\ t_{im\_1}(\alpha_3)t_{o\_1}(\alpha_4)$.

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 π_1', π_2' and π_3' be the paths corresponding to the same transitions as π_1, π_2 and π_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 σ, $\sigma_1 = t_{i\_1}(\alpha_1)t_{i\_2}(\alpha_2)t_j(\beta)t_{im\_1}(\alpha_3)$, 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, σ_1 is fireable in $\mathcal{S}$ iff ς_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 σ_1, t_{o_1} will get the maximal enabling degree. Therefore by firing σ_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 σ_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 ς_1, it is increased by the same amount ϕ as in $\mathcal{S}$, because $G(\pi_i) = G(\pi_i')$, $i = 1, 2, 3$.

Therefore, if σ is fireable in $\mathcal{S}$, ς is for sure fireable in $\mathcal{S}'$. The other direction, if ς is fireable in $\mathcal{S}'$, σ is fireable in $\mathcal{S}$ when the intermediate transition t_j is fired in the maximal amount.

A 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. ◻

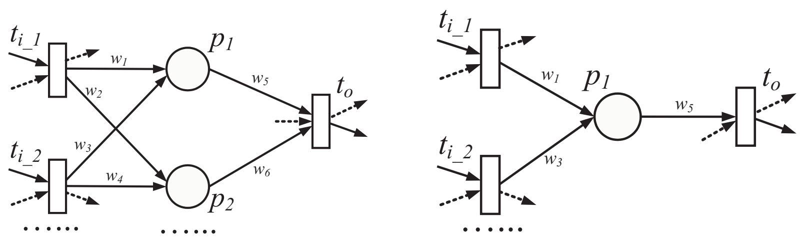
Remark 3.6. *Transition Reduction rule (T-RR) is a generalization of the methods discussed in [**?**] for continuous CFPN systems. For instance, in [**?**], only ordinary nets are considered; on the other hand, a transition that has multiple inputs or outputs while its output places have multiple inputs, might not be reduced.*

It can be observed that, each time T-RR 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 T-RR results in a set of places between T_{in} and T_{out} but no transition.

Place Reduction Rule (P-RR). *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 P-RR, $G(\pi_a) = G(\pi_b)$ 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 P-RR, the weights of arcs should satisfy $\frac{w_1}{w_5} = \frac{w_2}{w_6}$, and $\frac{w_3}{w_5} = \frac{w_4}{w_6}$. Assume $\frac{\boldsymbol{m}_0(p_1)}{w_5} \leq \frac{\boldsymbol{m}_0(p_2)}{w_6}$, then by removing p_2, the reduced system is shown in Fig. 5(b).*



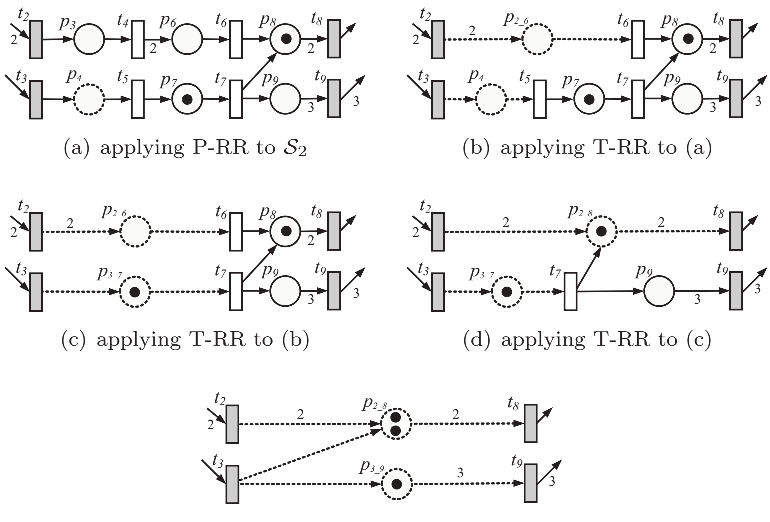
(a) Original subsystem $\mathcal{S}$ (b) Reduced subsystem $\mathcal{S}'$

Figure 5: Place Reduction Rule (P-RR): reducing p_2

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

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

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 P-RR to remove place p_5. By applying T-RR to the path between t_2 and t_6, p_{2_6} is obtained (Fig.6(b)). Similarly, the application of T-RR to the path between t_3 and t_7 in Fig.6(b),* *removes* t_5 *and obtains* p_{3_7} *(Fig.6(c)). The application of T-RR to the path between* t_2 *and* t_8 *in Fig.6(c), removes* t_6 *and obtains* p_{2_8} *(Fig.6(d)). The application of T-RR to the path between* t_3 *and* t_9 *in Fig.6(d), removes* t_7 *and obtains* p_{3_9} *(Fig.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* $\mathcal{S}_2$.



(a) applying P-RR to $\mathcal{S}_2$ (b) applying T-RR to (a)

(c) applying T-RR to (b) (d) applying T-RR to (c)

(e) applying T-RR to (d)

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

Assume that, using T-RR and P-RR, 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.

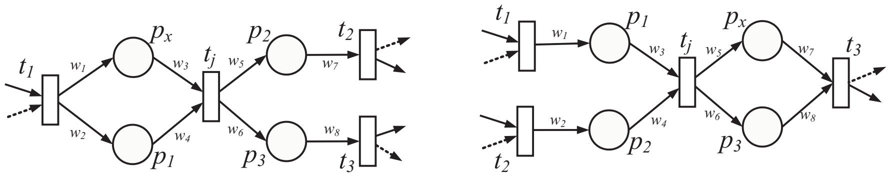
Property 3.10. *Any arbitrary and interleaved application of T-RR and P-RR until none of them can be applied produces the same reduced system.*

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 or BA, the same system is obtained. 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 T-RR (or P-RR), it is trivial.

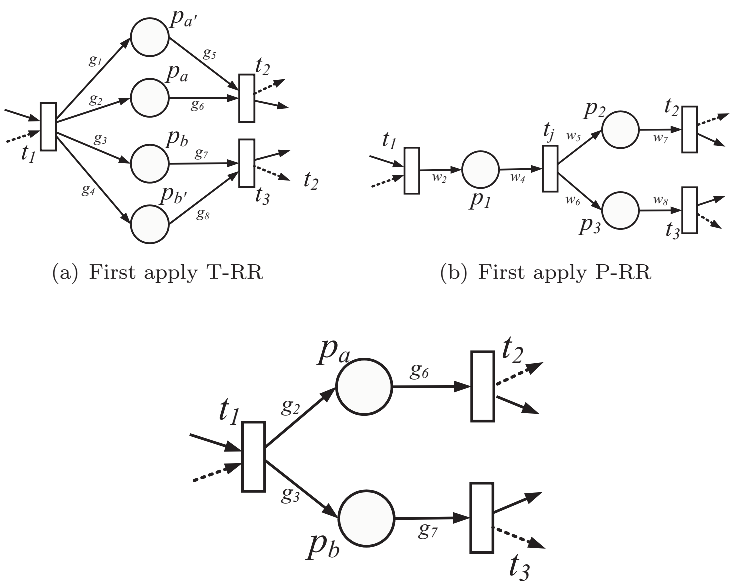
2) if A and B are instances of different rules. Without loss of generality, assume A is an instance of T-RR, removing a transition t_j and B is an instance of P-RR, removing an implicit place p_x. Obviously, if $t_j \notin {}^\bullet p_x \cup p_x{}^\bullet$, A and B are independent, so the system obtained 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 T-RR, at the same time, its input or/and output places can be reduced by using P-RR. Its extension to more general structures is quite straightforward.

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.



(a) (b)

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



(a) First apply T-RR

(b) First apply P-RR

(c) Fully reduced

Figure 8: Reducing by applying rules in different order

Since p_x can be removed by using P-RR, then $w_1/w_3 = w_2/w_4$ and in the initial state $\boldsymbol{m}_0(p_x)/w_3 \geq \boldsymbol{m}_0(p_1)/w_4$. 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 T-RR 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 P-RR. 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 P-RR has been applied to remove p_x, the system in Fig.8(b) is obtained. Then by applying T-RR, t_j is removed, it is clear that the same reduced system in Fig.8(c) is achieved.

Now we know that the order of applying reduction rules is not important. Let Γ_1 and Γ_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 T-RR is applied in Γ_1 and Γ_2 (because applying T-RR once, one transition between T_{in} and T_{out} is removed). From 1) and 2), we can transform the sequence Γ_1 to Γ'_1 by interchanging the order of adjacent rules, until all the instances of T-RR are moved ahead of instances of P-RR. Assume that by applying all the instances of T-RR, the obtained system is $\mathcal{S}'_1$. On the other hand, we can also transform the sequence Γ_2 to Γ'_2 by doing the same interchanging and assume that by applying all the instances of T-RR, the obtained system is $\mathcal{S}'_2$. Obviously, $\mathcal{S}'_1$ and $\mathcal{S}'_2$ are equivalent, and there are only places (but no transition) left between T_{in} and T_{out}. After that, the instances of P-RR 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 T-RR, we should first apply P-RR (or other possible methods) as much as possible, to remove the redundant implicit places, e.g., in Ex.3.9, P-RR 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 the subsystems obtained by cutting through 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. □

The decomposition method can be easily extended to a large scale discrete event system that is decomposed into K subsystems, by given sets of cutting places. The complemented subsystems are constructed in two steps: first, each subsystem builds its abstraction (the reduced subsystem respect to its interface transitions); then, each subsystem constructs its complementing parts based on the abstractions of the rest of the system that have been built in the first step. In this way, each subsystem does not need the detailed structures and states of other parts of the system, but only their abstractions.

4 Decentralized Control of CFPN Systems

4.1 Computing the Control Laws

A structural decomposition method has been proposed in Section 3. By using this method, we obtain a set of complemented subsystems. A interesting result is that the firing sequences of the obtained subsystems are identical to those of the original system. Thus, the local control laws can be computed separately, driving all subsystems to their corresponding final states. Considering 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 same 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 based 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 Table 1. As for the subsystems, 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 Table 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,* $\boldsymbol{\sigma}^1_{min}(t_1) \neq \boldsymbol{\sigma}^2_{min}(t_1)$.

Table 1: Markings and firing count vectors

P	$\boldsymbol{m}_0$ ($\boldsymbol{m}_f$)	$\boldsymbol{m}^1_0$ ($\boldsymbol{m}^1_f$)	$\boldsymbol{m}^2_0$ ($\boldsymbol{m}^2_f$)	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)				

Let $\mathcal{S} = \langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be the original system, with $\boldsymbol{m}_f > 0$ the desired final state. It is assumed that $\mathcal{S}$ 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) $\mathcal{CS}_i = \langle \mathcal{CN}_i, \boldsymbol{m}^i_0 \rangle$: the complemented subsystems with corresponding final state $\boldsymbol{m}^i_f$, $i = 1, 2, ..., K$.

(3) $B^{(i_1,i_2)}$: the buffer cutting places between $\mathcal{CS}_{i_1}$ and $\mathcal{CS}_{i_2}$.

(4) $U^{(i_1,i_2)}$: the interface transitions between $\mathcal{CS}_{i_1}$ and $\mathcal{CS}_{i_2}$.

(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 $\boldsymbol{m}^i_f$, $i = 1, 2, ..., K$.

According to the decomposition process and to the fact that firing sequences of the original system are preserved, the complemented subsystems are also strongly connected and consistent CFPN. Therefore, the minimal T-*semiflow* and minimal firing count vector are unique [**?**, **?**], i.e., $\boldsymbol{x}^i$ and $\boldsymbol{\sigma}^i_{min}$ are unique. So, any firing count vector $\boldsymbol{\sigma}^i$ driving $\mathcal{CS}_i$ to its final state can be written as follows

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

Algorithm 1 is used by the coordinator controller. Non-negative value α^1, α^2, ..., α^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 corresponding neghboring subsystems.

Algorithm 1 Coordinator

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

1: Receive $\boldsymbol{\sigma}^i_{min}$, $\boldsymbol{x}^i$ from local controllers
2: Compute α^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 α_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 $\boldsymbol{m}_f$, and denote by $\boldsymbol{\sigma}^{i_1}$ and $\boldsymbol{\sigma}^{i_2}$ the projections of $\boldsymbol{\sigma}$, corresponding to $\mathcal{CS}_{i_1}$ and $\mathcal{CS}_{i_2}$. By firing $\boldsymbol{\sigma}^{i_1}$ and $\boldsymbol{\sigma}^{i_2}$ in $\mathcal{CS}_{i_1}$ and $\mathcal{CS}_{i_2}$, markings $\boldsymbol{m}^{i_1}_f$, $\boldsymbol{m}^{i_2}_f$ are reached. 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 α^{i_1} and α^{i_2}, satisfying the constraints of LPP (3).

Proposition 4.2. *Let α^i be the value obtained by using Algorithm 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 (unique) 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). □

Notice that the minimal firing count vector is unique, implying that the solution of LPP (3) is also unique.

Algorithm 2 is used by the local controllers. In the first step, the minimal firing count vector $\boldsymbol{\sigma}^i_{min}$ of each subsystem $\mathcal{CS}_i$ is computed separately by the local controller. Then, every subsystem $\mathcal{CS}_i$ sends to the coordinator $\boldsymbol{\sigma}^i_{min}$, together with its corresponding minimal T-*semiflow*. After α^i is received from the coordinator, the controller of $\mathcal{CS}_i$ can be implemented independently by considering $\boldsymbol{\sigma}^i_{min} + \alpha^i \cdot \boldsymbol{x}^i$. In this work, an ON-OFF control strategy (presented in the next section) is used.

Recall that the coordinator does not know the detailed structures of subsystems, but only the interface transitions. The limited information required by the coordinator are the local control laws and the minimal T-*semiflows*, therefore all computations are done locally with very low communication costs. When the agreement is obtained, all the subsystems work independently.

Algorithm 2 Local Controller i

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

```
Compute σ^i_min driving the system to m^i_f;
Compute the minimal T-semiflow x^i;
Send σ^i_min and x^i to the coordinator;
Receive α^i from the coordinator;
Update σ^i ← σ^i_min + α^i · x^i;
Apply ON-OFF control;
```

4.2 Minimum-time ON-OFF Controller

Using the method presented in Section 4.1, the globally admissible local control laws are obtained. In this section we will discuss how to drive the system to its final state. An ON-OFF strategy is applied ensuring that the final marking is reached in minimum-time.

Usually, particularly when choices exist, the final state of a system may be reached firing different firing count vectors. For CFPN, 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 [**?**].

In a CFPN system, 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_1t_2 and t_2t_1 are fireable). Based on these observations, 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. 9 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* $\sigma_2 = t_1(0.8)t_2(0.3)$*.*

The strategy of the ON-OFF controller is quite simple: 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

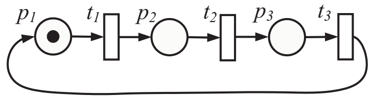


Figure 9: A CFPN system with $\boldsymbol{m}_0 = [1\ 0\ 0]$.

t_j at time τ 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 δ, $\boldsymbol{f}(t_j,\tau)$ are the uncontrolled flow at time τ. (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 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 systems 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 $\boldsymbol{m}_f > 0$ and use the proposed control method, 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 hand, the the minimum-time state evolution is guaranteed.

5 Case Study

In order to illustrate the developed approach, let us consider the Choice-Free CPN in Fig.10. It is adapted from the model of a simple manufacturing line that makes tables [**?**]. It consists of three work stations: WS_1 and WS_2 and WS_3. Two types of raw materials A and B are 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.

Let us assume that the three work stations are located in three different cities, while the cost for having a complex central controller that knows the structures and states of the whole system is prohibitive. We will apply to this system the proposed decentralized control method.

Table 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: $\lambda_8 = \lambda_{10} = 1/2$, $\lambda_{15} = \lambda_{16} = 1/3$, $\lambda_{20} = \lambda_{22} = 1/4$ and for other transitions, all equal to 1. Under this setting, the maximal throughput of transition E_M2_C (t_{22}, which models the machine that produces the final product) in the steady state is 0.33 ([**?**]).

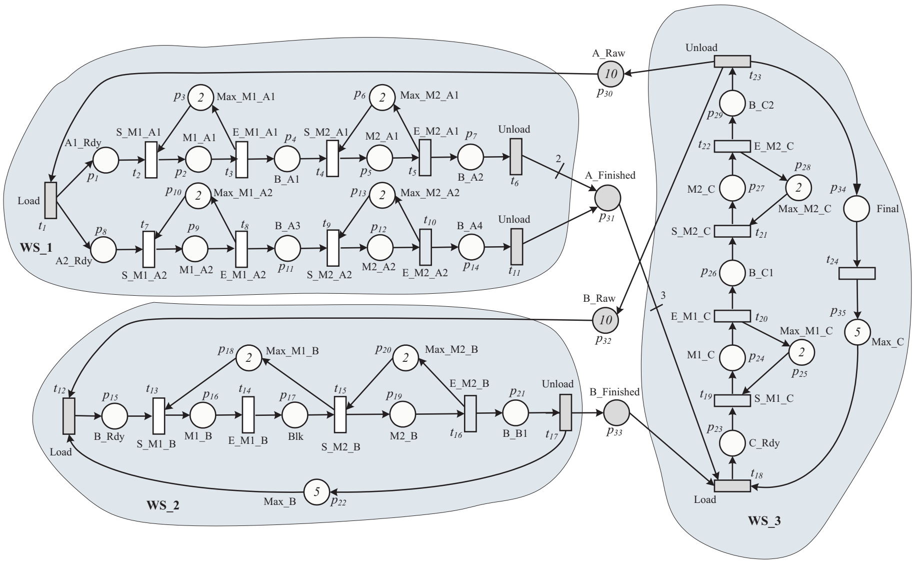


Figure 10: A manufacturing system with three work stations.

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 $B^{(1,3)} = \{p_{30}, p_{31}\}$, $B^{(2,3)} = \{p_{32}, p_{33}\}$ and the interface transition are $U^{(1,3)} = \{t_1, t_6, t_{11}, t_{18}, t_{23}, \}$, $U^{(2,3)} = \{t_{12}, t_{17}, t_{18}, t_{23}\}$. The complemented subsystems are shown in Fig.11, the final states of subsystems and their corresponding minimal firing count vectors are shown in Table 3.

In this specific example, the T-*semiflows* of subsystems are unit vectors $\mathbf{1}$, by applying Algorithm 1, the solution is quite straightforward: $\alpha^1 = \alpha^3 = 0$ and $\alpha^2 = 0.33$. So the control law of $\mathcal{CS}_2$ should be updated to $\boldsymbol{\sigma}^2_{min} + 0.33 \cdot \mathbf{1}$, the control laws of $\mathcal{CS}_1$ and $\mathcal{CS}_3$ are $\boldsymbol{\sigma}^1_{min}$ and $\boldsymbol{\sigma}^3_{min}$, respectively. By applying the ON-OFF controller using these laws, the final state of the system is reached in 17.66 time units, which is the minimum-time.

Table 2: The interpretation of the PN model in Fig.10

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

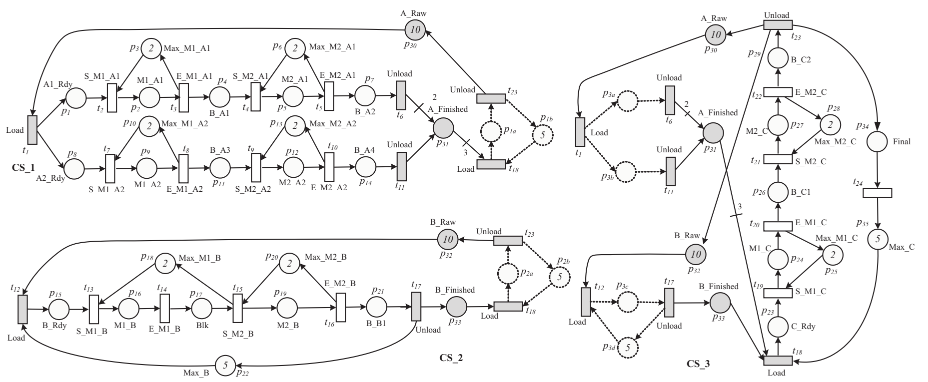


Figure 11: The complemented subsystems obtained from the model in Fig.10

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	0.33	t_1	8.17	p_{15}	0.33	t_{12}	7.00	p_{23}	0.33	t_1	8.17
p_2	0.33	t_2	7.83	p_{16}	0.33	t_{13}	6.67	p_{24}	1.33	t_6	6.50
p_3	1.67	t_3	7.50	p_{17}	1.00	t_{14}	6.34	p_{25}	0.67	t_{11}	0.00
p_4	0.33	t_4	7.17	p_{18}	0.67	t_{15}	5.34	p_{26}	0.33	t_{12}	7.33
p_5	0.33	t_5	6.83	p_{19}	1.00	t_{16}	4.34	p_{27}	1.33	t_{17}	4.33
p_6	1.67	t_6	6.50	p_{20}	1.00	t_{17}	4.00	p_{28}	0.67	t_{18}	4.00
p_7	0.33	t_7	2.00	p_{21}	0.33	t_{18}	3.67	p_{29}	0.33	t_{19}	3.67
p_8	6.17	t_8	1.33	p_{22}	2.00	t_{23}	0.00	p_{30}	2.17	t_{20}	2.33
p_9	0.67	t_9	1.00	p_{32}	3.00			p_{31}	1.00	t_{21}	2.00
p_{10}	1.33	t_{10}	0.33	p_{33}	0.33			p_{32}	3.00	t_{22}	0.67
p_{11}	0.33	t_{11}	0.00	p_{2a}	3.67			p_{33}	0.33	t_{23}	0.33
p_{12}	0.67	t_{18}	4.00	p_{2b}	1.33			p_{34}	0.33	t_{24}	0.00
p_{13}	1.33	t_{23}	0.33					p_{35}	1.00		
p_{14}	0.33							p_{3a}	1.67		
p_{30}	2.17							p_{3b}	8.17		
p_{31}	1.00							p_{3c}	3.00		
p_{1a}	3.67							p_{3d}	2.00		
p_{1b}	1.33										

6 Conclusions

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

We assume that 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 those cutting places are chosen. This may further influence the computational complexity, because the size of complemented subsystems is larger 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 the control laws of 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 a low communication cost. By applying the ON-OFF strategy in each subsystem, the global final state is reached in minimum-time.

As future work, this decentralized control framework will be considered for more general PN classes. In this case, we need to address the reachability problem of the final state, considering that 1) the minimal firing count vectors may not be unique, if some "incorrect" ones are chosen, we may never reach an agreement, and that 2) in some situations, systems may "deadlock" when pure ON-OFF strategy is used. On the other hand, since all the transitions are assumed to be controllable in the present work, another line of extension is to consider partially controllable systems.

A

By sampling the continuous-time TCPN system with a *sampling period* Θ, we obtain the discrete-time TCPN ([**?**]) given by:

$$\begin{aligned} &\boldsymbol{m}_{k+1} = \boldsymbol{m}_k + \boldsymbol{C} \cdot \boldsymbol{w}_k \cdot \Theta \\ &0 \leq \boldsymbol{w}_k \leq \boldsymbol{f}_k \end{aligned} \tag{5}$$

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 [**?**] that if the sampling period satisfies (6), the reachability spaces of discrete-time and continuous-time PN systems are the same.

$$\forall p \in P : \sum_{t_j \in p^\bullet} \lambda_j \cdot \Theta < 1 \tag{6}$$

It is assumed that the sampling period Θ is small enough to satisfy (6), and the detailed proof is given in the setting of discrete-time. It can be naturally extended to continuous-time systems.

Proposition 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. $\square$

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