Report: Decentralized Control of Continuous Petri Nets

April 21, 2012

Abstract

Aiming to reach a desired final state from a given initial one, this paper focuses on decentralized control of systems modeled by continuous Petri nets. The general PN systems considered in the paper are composed by subsystems interconnected by places (modeling buffers). Local control laws are first computed separately in subsystems, but "incorrectly" chosen ones may cause the reachability problem of the final state. We provide two sufficient conditions that should be satisfied by the interface between subsystems and they can be verified in polynomial time. It is proved that if one of those conditions is satisfied, we can always obtain globally admissible control laws based on some limited information, without knowing the structure of each subsystem.

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 *fluidification* can be used.

Continuous PN (CPN) [6, 17] 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. Many works can be found in the literature about the control of different classes of continuous PN, e.g., [1, 8, 12]. For the kind of time interpreted systems under *infinite server semantics*, several control approaches have been considered, e.g., [13, 3, 20].

Decentralized control is extensively explored in recent decades for complex dynamic systems (e.g., [16, 22, 19, 7]), in which multiple controllers may be allocated to subsystems. In the context of decentralized control on PN, some approaches have been proposed. In [4], a decentralized approach based on overlapping decompositions was proposed. The centralized admissibility concept was extended to d-admissibility for the decentralized setting in [9]. Under certain assumptions, the methods in [5] 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.

Different from the methods in [9, 5] proposed for discrete systems, which focus on enforcing 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 [4], 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.

Two of the recent work dealing with the similar problem are presented in [2] and [21]. In the former one, mono-T-semiflow net systems are considered, where continuous models composed of 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. In [21], decentralized control methods for Marked Graphs are discussed, where the missing part of disconnected subsystems are reduced to sets of places, obtaining the complemented subsystems. Then local control laws are independently computed and a coordinator is used to reach an agreement among them.

The previous works are extended in this paper in the sense that general net structures are considered. Local control laws are separately computed in subsystems. Considering that the buffer places are shared by more than one subsystems, there must be an agreements among local control laws. In a general PN system, the "incorrectly" chosen local control laws may lead the global state to be unreachable, i.e., the agreement may never be reached. We introduce two sufficient conditions and prove that, if one of them is satisfied, we can always obtain globally admissible control laws based on the local ones using the proposed algorithms, mainly consisting of solving Liner Programming Problems (LPP) and executed in a coordinator. The coordinator only needs limited information (abstractions of local control laws and T-semiflows) from local controllers, ensuring low communication costs. When we get the admissible control laws, several control methods can be applied to drive the system to its desired state, e.g., [13, 3], but this is out of the scope of this paper.

This paper is organized as follows: Section 2 briefly recalls some basic concepts of continuous PN. Section 3 presents the system and control structures consider in this work. In section 4, two sufficient conditions are first presented, then algorithms are proposed to compute the globally admissible control laws. A manufacturing system is used to illustrate the proposed method in Section 5. The conclusions are in section 6.

2 Continuous Petri Nets and notations

The reader is assumed to be familiar with basic concepts of continuous Petri nets (see [6, 17] for a gentle introduction).

Definition 2.1 *A continuous PN 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 is a mono-T-semiflow net iff it is conservative and has a unique minimal T-semiflow whose support contains all the transitions [11].

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 of the sequence σ. A firing count vector $\boldsymbol{\sigma}$ is said to be *minimal* one driving the system to $\boldsymbol{m}$ if for any T-*semiflow* $\boldsymbol{x}$, $||\boldsymbol{x}|| \not\subseteq ||\boldsymbol{\sigma}||$.

In timed continuous PN (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 server semantics appear, being the most used ones *infinite* (or variable speed) and *finite* (or constant speed) server semantics ([6, 17]), for which a firing rate $\lambda_j \in \mathbb{R}_{>0}$ is associated to transition t_j. This paper deals with infinite server semantics for which the flow of a transition t_j at time τ is the product of its firing rate, λ_j, and its enabling degree at $\boldsymbol{m}(\tau)$:

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

In this paper 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 speed of transitions ([17]). 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 that can be applied is $u(t_j) = 0$).

3 System structures and control problems

The kind of systems we consider in this work are composed by several subsystems modeled with CPN, interconnected and communicate with sets of places, for instances, the Deterministically Synchronized Sequential Processes that cooperate through buffers ([15]).

As a simple example, let us consider a conservative and consistent PN system shown in Fig.1. It is composed by two subsystem $\mathcal{S}^1 = \langle \mathcal{N}^1, \boldsymbol{m}_0' \rangle$ and $\mathcal{S}^2 = \langle \mathcal{N}^1, \boldsymbol{m}_0^2 \rangle$, places p_{13} to p_{16} model the *buffers* that are used by subsystem for cooperating (for example, in a production/consumption schema), denoted by $B^{(1,2)}$. In each subsystem, those transitions connected with buffer places are said to be *interface transitions*, denoted by $U^1 = \{t_2, t_3, t_4, t_5, t_6, t_7\}$ and $U^2 = \{t_8, t_{10}, t_{12}, t_{13}\}$.

The control problem addressed in this work is how to compute the control law to drive a system, composed by multiple subsystems, from an initial state to a desired final state, including the state of buffer places. Similar distributed/decentralized system structures and control methods can be found in literatures, but only for limited subclasses. In [2], it is assume that the system is mono-T-semiflow and each buffer is input and output private, i.e., given a buffer b, only one subsystem i can put tokens in b, and only one subsystem j can remove tokens from b. In [21], systems are assumed to be Marked Graph.

The decentralized control structures consist of local controllers and a coordinator, the system skeleton is shown in Fig.2.

It is assume that the local controllers only know the structures of their corresponding subsystems. A high level coordinator is used to make the agreement among local control laws. The coordinator can receive some limited information from local controllers, in particular, *abstractions* of T-semiflows and minimal firing count vectors (will be explained in the next section).

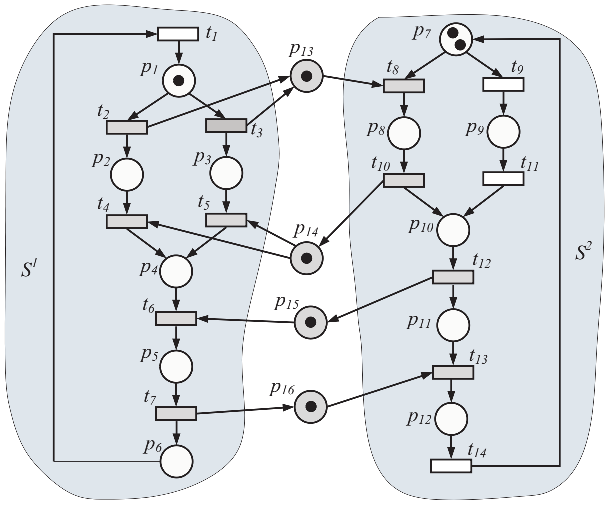

Figure 1: A Ct and Cv PN system, composed by two subsystems

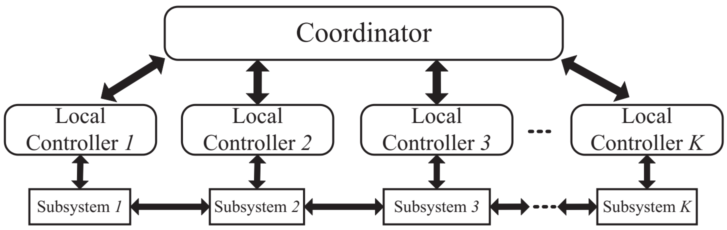

Figure 2: Decentralized Control Structures

4 Decentralized control methods

This section discusses how to compute in a decentralized way the control laws that will drive the global system from an initial state to a final state. Initial local control laws (minimal firing count vectors) are first computed independently in subsystems, then by adding some T-semiflows we try to make the agreements among them, and obtain the globally admissible ones. Unlike the mono-T-semiflow PN system, when a general CPN model is considered, minimal firing count vectors may be not unique (depending on the structure and also initial/final states). If an "incorrect" one is chosen, the final state may become unreachable (see the example shown in Ex. 4.1).

Example 4.1 *Let us consider the consistent and conservative PN system in Fig. 1. Assume we want to move the one token from* p_1 *to* p_6*, and the two tokens from* p_7 *to* p_{12}*, i.e., in the final state* $\boldsymbol{m}_f(p_6) = 1$*,* $\boldsymbol{m}_f(p_{12}) = 2$*, while keep the state of buffers unchanged. Consider subsystem* S^1*, in order to reach this state, one possible minimal firing count vector is to fire* t_2*,* t_4*,* t_6*,* t_7*, i.e., fire* $\boldsymbol{\sigma}^1_{min} = [0\ 1\ 0\ 1\ 0\ 1\ 1]^T$*. For subsystem* $\mathcal{S}^2$*, one possible minimal firing count vector driving it to its final state is to fire* t_9*,* t_{11}*,* t_{12}*,* t_{13}*, i.e.,* $\boldsymbol{\sigma}^2_{min} = [0\ 2\ 0\ 2\ 2\ 2\ 0]^T$*. Notice that, in this case, we can never maintain the states of buffers, by adding some T-semiflows to* $\boldsymbol{\sigma}^1_{min}$ *and* $\boldsymbol{\sigma}^2_{min}$*. For example, by firing* $\boldsymbol{\sigma}^1_{min}$ *and* $\boldsymbol{\sigma}^2_{min}$*,* p_{13} *will obtain one more token than expected, so a T-semiflow contains* t_8 *should be fired to remove it. But at the same time,* p_{15} *will get unnecessary tokens. To reduce the tokens of* p_{15} *by firing T-semflows in* $\mathcal{S}^1$*, we recursively put more tokens in* p_{13} *and so on. In this example, the correct minimal firing count vector should be chosen for* $\mathcal{S}^2$ *is* $[2\ 0\ 2\ 0\ 2\ 2\ 0]^T$*.*

In the following, we give two sufficient conditions and prove that if interface transitions satisfy one of them, we can always obtain the globally admissible control laws by applying the algorithm that is proposed afterwards.

Proposition 4.2 *Let $\mathcal{S} = \langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a consistent PN system, and $\boldsymbol{m}_f$ be a reachable final marking. Let $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ be firing count vectors that can drive the system to $\boldsymbol{m}_f$, i.e., $\boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}_1 = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}_2$. Let $t_a, t_b \in T$, and assume $\boldsymbol{\sigma}_1(t_j) \geq \boldsymbol{\sigma}_2(t_j), j = a, b$, if condition (C1) or (C2) is satisfied, then there exists a T-semiflow $\boldsymbol{x}$ such that*

$$\boldsymbol{\sigma}_1(t_j) = \boldsymbol{\sigma}_2(t_j) + \boldsymbol{x}(t_j), j = a, b \tag{2}$$

(C1) there exist T-semiflows $\boldsymbol{x}_1$,$\boldsymbol{x}_2$, such that $t_a \in ||\boldsymbol{x}_1||$, $t_b \notin ||\boldsymbol{x}_1||$ and $t_b \in ||\boldsymbol{x}_2||$, $t_a \notin ||\boldsymbol{x}_2||$.

(C2) there exists $\beta > 0$, such that for any T-semiflow $\boldsymbol{x}$, if $t_a \in ||\boldsymbol{x}||$ and $t_b \in ||\boldsymbol{x}||$ then $\boldsymbol{x}(t_a) = \beta \cdot \boldsymbol{x}(t_b)$.

Proof: 1) When condition (C1) is satisfied: assume $\boldsymbol{\sigma}_1(t_a) - \boldsymbol{\sigma}_2(t_a) = d_1 \geq 0$ and $\boldsymbol{\sigma}_1(t_b) - \boldsymbol{\sigma}_2(t_b) = d_2 \geq 0$, then, just let $\boldsymbol{x} = \frac{d_1}{\boldsymbol{x}_1(t_a)} \cdot \boldsymbol{x}_1 + \frac{d_2}{\boldsymbol{x}_2(t_b)} \cdot \boldsymbol{x}_2$, (2) is satisfied.

2) When condition (C2) is satisfied: Since $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ both drive the system to $\boldsymbol{m}_f$, then according the state equations, we have $\boldsymbol{C} \cdot (\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2) = 0$. Let $\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2 = \boldsymbol{\sigma}_{12}$, there may have negative elements in $\boldsymbol{\sigma}_{12}$, but since $\mathcal{N}$ is consistent, we can add a T-semiflows $\boldsymbol{x}_1 > 0$ to $\boldsymbol{\sigma}_{12}$, obtaining $\boldsymbol{\sigma}'_{12}$ that has all its elements positive value, i.e., $\boldsymbol{\sigma}'_{12} = \boldsymbol{\sigma}_{12} + \boldsymbol{x}_1 > 0$, so $\boldsymbol{\sigma}'_{12}$ is a T-semiflow. According to condition (C2), $\frac{\boldsymbol{\sigma}'_{12}(t_a)}{\boldsymbol{\sigma}'_{12}(t_b)} = \frac{\boldsymbol{x}_1(t_a)}{\boldsymbol{x}_1(t_b)} = \beta$, then $\frac{\boldsymbol{\sigma}_{12}(t_a)}{\boldsymbol{\sigma}_{12}(t_b)} = \beta$. Therefore, let $\boldsymbol{x} = \frac{\boldsymbol{\sigma}_{12}(t_j)}{\boldsymbol{\sigma}'_{12}(t_j)} \cdot \boldsymbol{\sigma}'_{12}$, $j = a$ or b, (2) is satisfied. ■

Proposition 4.2 gives the sufficient conditions to obtain a (new) firing count vector $\boldsymbol{\sigma}'$ from $\boldsymbol{\sigma}_2$ by adding a T-semiflow $\boldsymbol{x}$, i.e. $\boldsymbol{\sigma}' = \boldsymbol{\sigma}_2 + \boldsymbol{x}$, such that according to $\boldsymbol{\sigma}'$, both t_a and t_b are fired with the same amounts as in $\boldsymbol{\sigma}_1$, i.e., $\boldsymbol{\sigma}'(t_j) = \boldsymbol{\sigma}_1(t_j)$, $j = a, b$, and $\boldsymbol{\sigma}'$ also drives the system to $\boldsymbol{m}_f$. For instance, let us consider the subsystem $\mathcal{S}^2$ in Fig.1. It is obvious that t_8 and t_{10} satisfy condition (C2), because they always show in the same T-semiflows with fixed proportion. In order to move the two token from p_7 to p_{12} we may fire, for example, $\boldsymbol{\sigma}_1 = [\ 2\ 0\ 2\ 0\ 2\ 2\ 2\ 0]^T$ or $\boldsymbol{\sigma}_2 = [\ 0\ 2\ 0\ 2\ 2\ 2\ 2\ 0]^T$ and we have $\boldsymbol{\sigma}_1(t_j) > \boldsymbol{\sigma}_2(t_j)$, $j = 8, 10$. According to Proposition 4.2, we can construct $\boldsymbol{\sigma}'$ by adding $\boldsymbol{x} = [\ 2\ 0\ 2\ 0\ 2\ 2\ 2]^T$ to $\boldsymbol{\sigma}_2$, such that $\boldsymbol{\sigma}' = [\ 2\ 2\ 2\ 2\ 4\ 4\ 2]^T$, and obviously $\boldsymbol{\sigma}'(t_j) = \boldsymbol{\sigma}_1(t_j)$, $j = 8, 10$, and by firing $\boldsymbol{\sigma}'$ the same marking is reached as fring $\boldsymbol{\sigma}_1$.

Now we will show that both of these conditions can be verified in polynomial time by solving several LPPs.

Let us consider the LPP (3), if it has a solution , it means that there exists a T-semiflow $\boldsymbol{x}_1$, such that $t_1 \in ||\boldsymbol{x}_1||$ and $t_2 \notin ||\boldsymbol{x}_1||$. Similarly, if at the same time, LPP (4) also has a solution, then condition (C1) is satisfied. Otherwise, we need to check LPP (5). If it does not have a solution, then condition (C2) is satisfied.

$$\begin{array}{ll} \min & \mathbf{1}^T \cdot \boldsymbol{x} \\ \text{s.t.} & \boldsymbol{C} \cdot \boldsymbol{x} = 0 \\ & \boldsymbol{x} \geq 0 \\ & \boldsymbol{x}(t_1) \geq 1 \\ & \boldsymbol{x}(t_2) = 0 \end{array} \tag{3}$$

$$\begin{array}{ll} \min & \mathbf{1}^T \cdot \boldsymbol{x} \\ \text{s.t.} & \boldsymbol{C} \cdot \boldsymbol{x} = 0 \\ & \boldsymbol{x} \geq 0 \\ & \boldsymbol{x}(t_2) \geq 1 \\ & \boldsymbol{x}(t_1) = 0 \end{array} \tag{4}$$

$$
\begin{array}{ll}
\min & \mathbf{1}^T \cdot \boldsymbol{x} \\
\text{s.t.} & \boldsymbol{C} \cdot \boldsymbol{x} = 0 \\
& \boldsymbol{x} \geq 0 \\
& ||\boldsymbol{x}(t_1) - \boldsymbol{x}(t_2) \cdot \beta||_1 \geq \epsilon \\
& \epsilon \geq 0
\end{array} \quad (5)
$$

where ϵ is a very small positive value.

In the following, it is assumed that any pair of transitions in a interface transition set U^k, satisfies condition (C1) or (C2). We will prove that under this assumption, the globally admissible control laws can always be obtained from the local ones by adding some T-semiflows. Let us first consider the system consists of two subsystems, then algorithms for a general decentralized control framework is given.

Proposition 4.3 *Let $\mathcal{S}$ be a consistent PN system, assume it is composed by two subsystems $\mathcal{S}^{k1}$ and $\mathcal{S}^{k2}$, $\boldsymbol{\sigma}^{k1}_{min}$ and $\boldsymbol{\sigma}^{k2}_{min}$ are minimal firing count vectors driving them to their corresponding final states. If each pair of interface transitions in U^{k1} and U^{k2} satisfies condition (C1) or (C2), then we can construct $\boldsymbol{\sigma}' = \begin{bmatrix} \boldsymbol{\sigma}^{k1}_{min} + \boldsymbol{x}^{k1} \\ \boldsymbol{\sigma}^{k2}_{min} + \boldsymbol{x}^{k2} \end{bmatrix}$, such that $\boldsymbol{\sigma}'$ can drive $\mathcal{S}$ to $\boldsymbol{m}_f$, where $\boldsymbol{x}^{k1}$, $\boldsymbol{x}^{k2}$ are T-semiflows of $\mathcal{S}^{k1}$, $\mathcal{S}^{k2}$ respectively.*

Proof: Since $\boldsymbol{m}_f$ is reachable and consider that $\mathcal{S}$ is consistent, we can always find a big enough $\boldsymbol{\sigma} = \begin{bmatrix} \boldsymbol{\sigma}^{k1} \\ \boldsymbol{\sigma}^{k2} \end{bmatrix}$ (by adding T-semiflows), such that $\boldsymbol{\sigma}^{k1} \geq \boldsymbol{\sigma}^{k1}_{min}$, $\boldsymbol{\sigma}^{k2} \geq \boldsymbol{\sigma}^{k2}_{min}$, and $\boldsymbol{\sigma}$ can drive $\mathcal{S}$ to $\boldsymbol{m}_f$. It is clear, $\boldsymbol{\sigma}^{k1}$ and $\boldsymbol{\sigma}^{k2}$ can drive $\mathcal{S}^{k1}$ and $\mathcal{S}^{k2}$ to their corresponding final states. According to Proposition 4.2, we can find $\boldsymbol{x}^{k1}$, such that $\varsigma^{k1} = \boldsymbol{\sigma}^{k1}_{min} + \boldsymbol{x}^{k1}$, in which the interface transitions in U^{k1} are fired with the same amount as in $\boldsymbol{\sigma}^{k1}$. Similarly, we can also find $\boldsymbol{x}^{k2}$, such that $\varsigma^{k2} = \boldsymbol{\sigma}^{k2}_{min} + \boldsymbol{x}^{k2}$, in which the interface transitions in U^{k2} are fired with the same amount as in $\boldsymbol{\sigma}^{k2}$, implying that the buffer places also reach their final states. Therefore, by firing $\boldsymbol{\sigma}'$, $\boldsymbol{m}_f$ is also reached. ■

Example 4.4 *Let us still consider the system shown in Ex.4.1, but remove buffer places p_{15} and p_{16}, then $U^1 = \{t_2, t_3, t_4, t_5\}$ and $U^2 = \{t_8, t_{10}\}$. It can be verified that each pair of transitions in U^1 and U_2 satisfies condition (C1) or (C2), for instance, in U^1, t_2 and t_3 satisfies condition (C1), t_2 and t_4 satisfies condition (C2). In U^2, t_8 and t_{10} satisfies condition (C2). In order to reach the final state, considering each system independently, we may choose $\boldsymbol{\sigma}^1_{min} = [\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ ]^T$ and $\boldsymbol{\sigma}^2_{min} = [\ 0\ 2\ 0\ 2\ 2\ 2\ 0\ ]^T$. In order to reach the global final state (including the buffer places), we can add $\boldsymbol{x}^2 = 0.5 \cdot [\ 2\ 0\ 2\ 0\ 2\ 2\ 2\ ]^T$ to $\boldsymbol{\sigma}^2_{min}$, then we have $\boldsymbol{\sigma}' = \begin{bmatrix} \boldsymbol{\sigma}^1_{min} \\ \boldsymbol{\sigma}^2_{min} + \boldsymbol{x}^{k2} \end{bmatrix}$. Then $\boldsymbol{\sigma}'$ can drive the global system to its final state. Notice that, depends on how the local control laws are chosen, $\boldsymbol{\sigma}'$ may not be a minimal firing count vector, because subsystems do not have the global information.*

Now let us consider a system composed by multiple subsystems. In order to have low communication costs and high efficiency, the coordinator should only exchange limited information with local controllers, in particular, the *abstractions* of local control laws and T-semiflows.

Let $\boldsymbol{\sigma}^k_{min}$ be a minimal firing count vector that can drive S^k from ${\boldsymbol{m}_0}^k$ to ${\boldsymbol{m}_f}^k$, and U^k be the set of interface transitions. The abstraction of $\boldsymbol{\sigma}^k_{min}$ corresponding to U^k is defined as $\boldsymbol{\sigma}^k_U \in \mathbb{Q}^{|U^k|}_{\geq 0}$, such that for every $t_j \in U^k$, $\boldsymbol{\sigma}^k_U(t_j) = \boldsymbol{\sigma}^k_{min}(t_j)$.

Notice that, if any pair of transition of U^k satisfies condition (C1) or (C2), then U^k can be partitioned into n_k $(1 \leq n_k \leq |U^k|)$ disjoint subsets, i.e., $U^k = U^k_1 \cup U^k_2 \cup, ..., \cup U^k_{n_k}$ and $U^k_1 \cap U^k_2 \cap ... \cap U^k_{n_k} = \emptyset$ such that the transitions in U^k_i, $i = 1, 2, ..., n_k$ are always fired in the same T-semiflows and with constant proportions. Let us denote the constant proportion between t_a and t_b by $\beta_{(a,b)}$. We will represent this proportional relation corresponding to the transitions in U^k_i by a vector $\boldsymbol{\gamma}^k_i \in \mathbb{N}^{|U^k|}$ — the abstractions of T-semiflows in S^k, such that:

(1) the greatest common divisor of the positive components of γ_i^k is one.

(2) $\forall t_a, t_b \in U_i^k$, $\gamma_i^k(t_a)/\gamma_i^k(t_b) = \beta_{(a,b)}$.

(3) $\forall t \notin U_i^k$, $\gamma_i^k(t) = 0$.

Example 4.5 *As in Ex.4.4, we consider the same system as in Fig.1 but removing buffer places p_{15} and p_{16}, then the interface transitions in $U^1 = \{t_2, t_3, t_4, t_5\}$ satisfy condition (C1) or (C2). It can then be partitioned into $U_1^1 = \{t_2, t_4\}$, $U_2^1 = \{t_3, t_5\}$: for any T-semiflow $\boldsymbol{x}$ in S^1 containing U_1^1, we have $\boldsymbol{x}(t_2)/\boldsymbol{x}(t_4) = \beta_{(2,4)} = 1$. Similarly, for any T-semiflow $\boldsymbol{x}$ containing U_2^1, we have $\boldsymbol{x}(t_3)/\boldsymbol{x}(t_5) = \beta_{(3,5)} = 1$. The abstractions of T-semiflows corresponding to U_1^1 and U_2^1 are $\gamma_1^1 = [\ 1\ 0\ 1\ 0\ ]^T$ and $\gamma_1^1 = [\ 0\ 1\ 0\ 1]^T$, respectively. Let $\boldsymbol{\sigma}_{min}^1 = [\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ ]^T$, its abstraction corresponding to U^1 is $\boldsymbol{\sigma}_U^1 = [\ 1\ 0\ 1\ 0\ ]^T$.*

Let $\boldsymbol{x}$ be a T-semiflow of S^k, and $\boldsymbol{x}_U^k$ be its part corresponding to transitions in interface U^k. It can be observed that, $\boldsymbol{x}_U^k$ can always be represented by the linear combinations of γ_i^k, $i = 1, 2, ..., n_k$, i.e., there exist $\alpha_1^k, \alpha_2^k, ..., \alpha_{n_k}^k \geq 0$ such that:

$$\boldsymbol{x}_U^k = \alpha_1^k \cdot \gamma_1^k + \alpha_2^k \cdot \gamma_2^k + ... + \alpha_{n_k}^k \cdot \gamma_{n_k}^k \qquad (6)$$

Given a system composed by K subsystems, Alg.1 gives the procedure for computing the globally admissible control laws based on the local ones. Notice that, the coordinator only needs to exchange with local controller the *abstractions* of T-semiflows and control laws, but without knowing the detailed structures of subsystems, therefore, the communications is very low.

Proposition 4.6 *Let $\mathcal{S} = \langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a consistent PN system, and $\boldsymbol{m}_f$ be a reachable final marking. Assume $\mathcal{S}$ composed by K subsystems $\mathcal{S}^1$, $\mathcal{S}^2$, ... $\mathcal{S}^K$, and each pairs of transition in the interface transition U^k, $k = 1, 2, ..K$ satisfies condition (C1) or (C2). The firing count vectors computed by Alg.1, $[\boldsymbol{\sigma}^1$; $\boldsymbol{\sigma}^2$; ...; $\boldsymbol{\sigma}^K]$ drives $\mathcal{S}$ to $\boldsymbol{m}_f$.*

Proof: Since $\boldsymbol{m}_f$ is a reachable marking of $\mathcal{S}$, so the corresponding final marking in each subsystem is also reachable, therefore LPP (7) is feasible.

Since $\mathcal{S}$ is consistent and $\boldsymbol{m}_f$ is reachable, we can always find a $\boldsymbol{\sigma}$ driving $\mathcal{S}$ to $\boldsymbol{m}_f$, such that $\boldsymbol{\sigma} > [\boldsymbol{\sigma}_{min}^1;\ \boldsymbol{\sigma}_{min}^2;\ \dots\ ;\ \boldsymbol{\sigma}_{min}^k]$. According to Proposition 4.3, for any connected subsystems $\mathcal{S}^{k1}$ and $\mathcal{S}^{k2}$, there exist T-semiflows $\boldsymbol{x}^{k1}$ of $\mathcal{S}^{k1}$, and $\boldsymbol{x}^{k2}$ of $\mathcal{S}^{k2}$, such that $\boldsymbol{\sigma}^{k1} = \boldsymbol{\sigma}_{min}^{k1} + \boldsymbol{x}^{k1}$ and $\boldsymbol{\sigma}^{k1} = \boldsymbol{\sigma}_{min}^{k2} + \boldsymbol{x}^{k2}$. By firng $\boldsymbol{\sigma}^{k1}$ and $\boldsymbol{\sigma}^{k1}$, the final markings of $\mathcal{S}^{k1}$ and $\mathcal{S}^{k2}$ are reached, while the buffer places between them, $B^{(k1,k2)}$, are also in their corresponding final states. On the other side, considering (6), LPP (8) is feasible.

By solving LPP (8), the firing counts of interface transitions are obtained, LPP (9) computes the firing counts of other interior transitions, which will drive the interior places to its final states. ■

5 Case study

Let us consider the PN model in Fig.3 which models a manufacturing line that makes tables, it is adapted from the example in [15]. It consists of three subsystems that make legs, boards, or assemble a table, corresponding to $\mathcal{S}^1$, $\mathcal{S}^2$ and $\mathcal{S}^3$ respectively. Here places p_{17} to p_{20} model buffers, in which one subsystem deposits products that are later consumed by others subsystems. The interface transitions in subsystems are: $U^1 = \{t_5, t_6\}$, $U^2 = \{t_7, t_{13}\}$, $U^3 = \{t_{15}, t_{16}, t_{17}, t_{18}, t_{19}\}$.

Assume that in the initial state we have: $\boldsymbol{m}_0(p_5) = \boldsymbol{m}_0(p_{10}) = \boldsymbol{m}_0(p_{16}) = 10$, $\boldsymbol{m}_0(p_{17}) = \boldsymbol{m}_0(p_{19}) = 5$, and in the desired final state: $\boldsymbol{m}_f(p_5) = 6$, $\boldsymbol{m}_f(p_{10}) = 7$, $\boldsymbol{m}_f(p_{16}) = 5$, $\boldsymbol{m}_f(p_{17}) = 1$, $\boldsymbol{m}_f(p_{18}) = 4$, $\boldsymbol{m}_f(p_{19}) = 3$, $\boldsymbol{m}_f(p_{20}) = 2$, $\boldsymbol{m}_f(p_7) = 0$ and all the other places with one token inside.

Algorithm 1: Computing globally admissible control laws

Input: S^k, $\boldsymbol{m}_0$, $\boldsymbol{m}_f$, $k = 1, 2, ..., K$

Output: $\boldsymbol{\sigma}^k$, $k = 1, 2, ..., K$

Each local controller k **compute its minimal firing count vector** $\boldsymbol{\sigma}^k_{min}$ **by solving LPP:**

$$\begin{array}{ll} \min & \mathbf{1}^T \cdot \boldsymbol{\sigma}^k_{min} \\ \textbf{s.t.} & {\boldsymbol{m}_f}^k - {\boldsymbol{m}_0}^k = \boldsymbol{C}^k \cdot \boldsymbol{\sigma}^k_{min} \\ & \boldsymbol{\sigma}^k_{min} \geq 0 \end{array} \qquad (7)$$

Each local controller k **send the abstraction of** $\boldsymbol{\sigma}^k_{min}$, $\boldsymbol{\sigma}^k_U$ **to the coordinator ;**

Each local controller k **compute the abstractions of T-semiflows** $\gamma^k_1, \gamma^k_2, ..., \gamma^k_{n_k}$**, and send them to the coordinator;**

The coordinator solve LPP:

$$\begin{array}{ll} \min & \sum\limits_{k=1}^{K} \alpha^k_1 + \alpha^k_2 + ... + \alpha^k_{n_k} \\ \textbf{s.t.} & {\boldsymbol{m}_f}^{B^{(k1,k2)}} - {\boldsymbol{m}_0}^{B^{(k1,k2)}} = \boldsymbol{C}^{(k1,k2)} \cdot \begin{bmatrix} \boldsymbol{\sigma}^{k1}_U + \alpha^{k1}_1 \cdot \boldsymbol{\gamma}^{k1}_1 + \alpha^{k1}_2 \cdot \boldsymbol{\gamma}^{k1}_2 + ... + \alpha^{k1}_{n_{k1}} \cdot \boldsymbol{\gamma}^{k1}_{n_{k1}} \\ \boldsymbol{\sigma}^{k2}_U + \alpha^{k2}_1 \cdot \boldsymbol{\gamma}^{k2}_1 + \alpha^{k2}_2 \cdot \boldsymbol{\gamma}^{k2}_2 + ... + \alpha^{k2}_{n_{k2}} \cdot \boldsymbol{\gamma}^{k2}_{n_{k2}} \end{bmatrix}, \\ & \forall k1, k2 \in \{1, 2, ..., K\}, S^{k1} \textbf{ and } S^{k1} \textbf{ are neighboring subsystems} \\ & \alpha^k_1, \alpha^k_2, ..., \alpha^k_{n_k} \geq 0, \quad k = 1, 2, ..., K \end{array} \qquad (8)$$

where $\boldsymbol{C}^{(k1,k2)} \in \mathbb{Q}^{|B^{(k1,k2)}| \times (|U^{k1}| + |U^{k2}|)}_{\geq 0}$ *is the flow matrix corresponding to* $B^{(k1,k2)}$ **;**

The coordinator send α^k_1, α^k_2, **...**, $\alpha^k_{n_k}$ **to subsystem** S^k **;**

Each local controller k **solves LPP:**

$$\begin{array}{ll} \min & \mathbf{1}^T \cdot \boldsymbol{\sigma} \\ \textbf{s.t.} & {\boldsymbol{m}_f}^k - {\boldsymbol{m}_0}^k = \boldsymbol{C}^k \cdot \boldsymbol{\sigma} \\ & \boldsymbol{\sigma}(t_j) = \boldsymbol{\sigma}^k_{min}(t_j) + \sum\limits_{i=1}^{n_k} \alpha^k_i \cdot \boldsymbol{\gamma}^k_i(t_j), \quad \forall t_j \in U^k \\ & \boldsymbol{\sigma} \geq 0 \end{array} \qquad (9)$$

where $\boldsymbol{C}^k$ *is the flow matrix of* $\mathcal{S}^k$**.**

updates control laws: $\boldsymbol{\sigma}^k \leftarrow \boldsymbol{\sigma}$ **;**

return $\boldsymbol{\sigma}^k$**,** $k = 1, 2, ..., K$

If we apply Alg.1, minimal firing count vectors driving subsystems to their corresponding final states are first computed independently, so we will obtain: in $\mathcal{S}^1$, $\boldsymbol{\sigma}^1_{min} = [2\ 2\ 1\ 1\ 0\ 0]^T$; in $\mathcal{S}^2$, $\boldsymbol{\sigma}^2_{min} = [3\ 0\ 0\ 1\ 1\ 0\ 0]^T$; in $\boldsymbol{\sigma}^3_{min} = [0\ 1\ 2\ 3\ 4\ 5]^T$. It is easy to obtain their abstractions corresponding to interface transitions: $\boldsymbol{\sigma}^1_U = [0\ 0]^T$, $\boldsymbol{\sigma}^2_U = [3\ 0]^T$, $\boldsymbol{\sigma}^3_U = [1\ 2\ 3\ 4\ 5]^T$.

It can be verified that in $\mathcal{S}^1$, t_5 and t_6 satisfy condition (C1), so U^1 can be partitioned into two $U^1_1 = \{t_5\}$ and $u^1_2 = \{t_6\}$, therefore, the abstractions of T-semfilows in $\mathcal{S}^1$ corresponding to the interface transitions is $\boldsymbol{\gamma}^1_1 = [1\ 0]^T$ and $\boldsymbol{\gamma}^1_2 = [0\ 1]^T$. In $\mathcal{S}^2$, interface transitions t_7 and t_{13} satisfy condition (C2), firing the same amount in any T-semiflow, so we have $\boldsymbol{\gamma}^2_1 = [1\ 1]^T$. Similarly, in $\mathcal{S}^3$, we have $\boldsymbol{\gamma}^3_1 = [1\ 1\ 1\ 1\ 1]^T$.

When the coordinate obtains the abstractions from subsystems, by solving LPP (8), it is obtained:

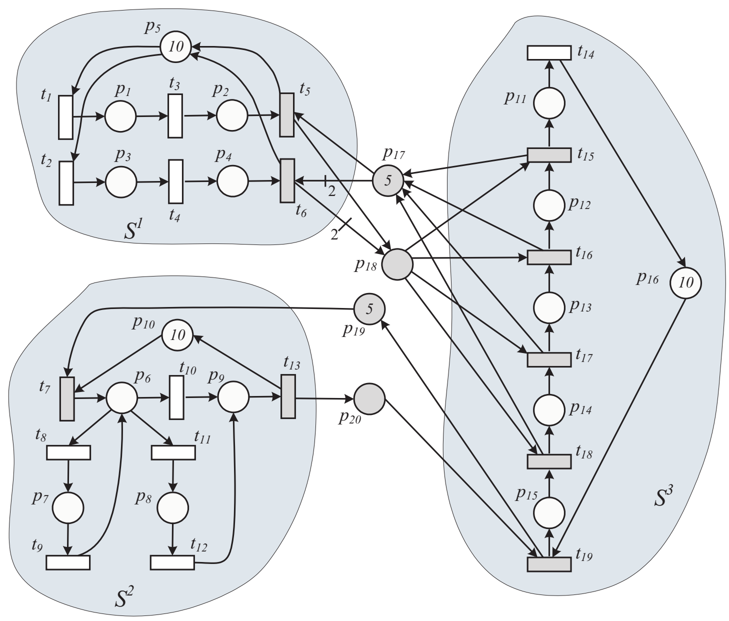

Figure 3: A PN model of a manufacturing line consists of three subsystems

$\alpha_1^1 = 0$, $\alpha_2^1 = 7$, $\alpha_1^2 = 7$ and $\alpha_1^3 = 0$. Finally, by solving LPP (9), we can obtain the globally admissible control laws in each subsystem: $\boldsymbol{\sigma}^1 = [2\ 9\ 1\ 8\ 0\ 7\ ]^T$, $\boldsymbol{\sigma}^2 = [10\ 0\ 0\ 8\ 1\ 0\ 7\ ]^T$, $\boldsymbol{\sigma}^3 = \boldsymbol{\sigma}^3_{min}$. It implies that in addition to the firing amount given by the minimal firing count vector, we need to fire T-semflow $\boldsymbol{x}^1 = [0\ 7\ 0\ 7\ 0\ 7]^T$ in $\mathcal{S}^1$ and $\boldsymbol{x}^2 = [7\ 0\ 0\ 7\ 0\ 0\ 7]^T$ in $\mathcal{S}^2$.

6 Conclusions

In this paper we address the decentralized control of general CPN system, driving the system from an initial state to a desired final state. The system composed by several subsystems interconnected with sets of buffer places are considered. Local control laws are computed separately, but they may not globally admissible and cause the reachability problem of the final state. We propose two sufficient conditions and prove that the globally admissible control laws can be achieved by using the proposed algorithms if one of them is satisfied. The conditions can be verified in polynomial time, and the algorithm is also efficient because only several LPPs should be solved. At the same time, since only limited information should be exchanged between local controllers and the coordinator, the communication costs are also low.

References

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

[2] H. Apaydin-Ozkan, J. Julvez, C. Mahulea, and M. Silva. A Control Method for Timed Distributed Continuous Petri nets. In *2010 American Control Conference*, Baltimore, USA, June 2010.

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

[4] Aybar Aydin. Decentralized structural control approach for Petri nets. *Control and Cybernetics*, 36(1):143–159, 2007.

[5] Francesco Basile, Alessandro Giua, and Carla Seatzu. Supervisory Control of Petri Nets with Decentralized Monitor Places. In *2007 American Control Conference*, pages 4957 – 4962, New York, USA, 2007.

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

[7] Zhisheng Duan, Jinzhi Wang, Guanrong Chen, and Lin Huang. Stability analysis and decentralized control of a class of complex dynamical networks. *Automatica*, 44(4):1028 1035, 2008.

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

[9] Marian V. Iordache and Panos J. Antsaklis. Decentralized Supervision of Petri Nets. *IEEE Transactions on Automatic Control*, 51(2):376–391, 2006.

[10] J. Júlvez, L. Recalde, and M. Silva. On reachability in autonomous continuous Petri net systems. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, *Applications and Theory of Petri Nets 2003*, volume 2679, pages 221–240, Eindhoven, The Netherlands, June 2003.

[11] J. Júlvez, L. Recalde, and M. Silva. Steady–state performance evaluation of continuous mono-T-semiflow Petri nets. *Automatica*, 41(4):605–616, 2005.

[12] Weizhi Lia and Tianlong Gu. Optimization and control of production system based on interval speed continuous Petri nets. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 1212 – 1217, Oct. 2005.

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

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

[15] Laura Recalde, Enrique Teruel, and Manuel Silva. Modeling and Analysis of Sequential Processes that Cooperate Through Buffers. *IEEE Transactions on Robotics and Automation*, 14(2):267–276, 1998.

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

[17] M. Silva, J. Júlvez, C. Mahulea, and C. Renato Vázquez. On fluidization of discrete event models: observation and control of continuous Petri nets. *Discrete Event Dynamic Systems: Theory and Applications*, 21(4):427–497, 2011.

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

[19] Shaocheng Tong, Han-Xiong Li, and Guanrong 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):770 – 775, 2004.

[20] LW. Wang, C. Mahulea, J. Júlvez, and M. Silva. Minimum-time Control for Structurally Persistent Continuous Petri Nets. In *49th IEEE Conference on Decision and Control*, pages 2771–2776, Atlanta, Georgia USA, December 2010.

[21] LW. Wang, C. Mahulea, J. Júlvez, and M. Silva. Decentralized Control of Large Scale Systems Modeled with Continuous Marked Graphs. In *18th IFAC World Congress*, Milan, Italy, Aug. 2011.

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