Distributed Control of Large Scale Systems Modeled with Continuous Petri Nets

Liewei Wang, Cristian Mahulea, Jorge Júlvez, and Manuel Silva

***Abstract*— This paper addresses the distributed control of large scale systems that are modeled with timed continuous Petri nets ($ContPN$). Distributed structures are first obtained after the system is decomposed into subnets by cutting the original net through sets of places, and adding marking structurally implicit places. Then, local control laws are computed separately. Algorithms are proposed to make the locally computed laws to be compatible and fireable when the global state of the system is considered. It is proved that using the control laws computed with the proposed algorithms, the final state of the overall system can be reached in minimum time. A manufacturing system is taken as case study to illustrate the control method.**

I. INTRODUCTION

Petri Nets (PN) is a well known paradigm used for modeling, analysis, and synthesis of *discrete event systems* (DES). With strong facility to depict the sequence, concurrency, conflict and other synchronous relationships, 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 Petri nets [7], [17] are fluid approximations of classical *discrete Petri nets* obtained by removing the integrality constraints, which means 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, e.g., reachability and controllability [12], [10] problems.

Different works about control of Petri nets can be found in the literature [9], [4], [2], etc. In the context of distributed systems, distributed timed automata is discussed in [11]. In [20], the author proposed a method for the modeling and decomposition of large and complex discrete event manufacturing systems, where a Petri net based controller is distributed in machines and exchanges signals with coordinators. An architecture for distributed implementation of Petri nets in control applications is proposed in [14]. A distributed control strategy is designed in [8] for forbidden state avoidance for discrete event systems which are modeled as Petri nets.

This work has been partially supported by the European Community's Seventh Framework Programme under project DISC (Grant Agreement n. INFSO-ICT-224498) and by CICYT - FEDER projects DPI2006-15390 and TIN2007-66523.

The authors are with the Department of Computer Science and Systems Engineering, University of Zaragoza, Maria de Luna 1, 50018 Zaragoza, Spain {lwwang, cmahulea, julvez, silva@unizar.es}.

Coming back to the continuous Petri nets, in [3], a reachability control problem of timed distributed continuous Petri net systems is studied. The paper considers Petri nets composed of several subsystems that communicate through channels modeled by places. The proposed algorithm allows the subsystems to reach their respective target markings at different time instants and keep them as long as required.

In this work, the distributed control of large scale systems which are modeled with timed continuous Petri nets is addressed. As a starting point of this research topic, it is assumed that the systems we handle are modeled with marked graphs $ContPN$. This paper mainly focuses on driving the system from an initial state to a desired final state. A large scale system is first structurally decomposed into smaller subsystems, then the local control law for each subsystem is computed separately. A supervisory controller is introduced to update the locally computed control laws in order to make them admissible when considering the system globally, without knowing the detailed structures of local subsystems. With these control laws, all the local controllers work independently, and the final state can be reached in minimum time.

This paper is organized as follows: Section II briefly recalls some basic concepts. In Section IV, a structurally decomposition method for marked graphs is discussed, which is used here to obtain distributed structures. Section V proposed the approach for distributed control of large system. Section VI gives an example of manufacturing systems. The conclusions are in Section VII.

II. BASIC CONCEPTS

The reader is assumed to be familiar with basic Petri net concepts (see [7], [17] for a gentle introduction).

A. Continuous Petri Nets

***Definition** 2.1:* A continuous Petri net 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.
- $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{R}_{\geq 0}^{|P| \times |T|}$ 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, this number represents its marking. The distribution of tokens in places is denoted by $\boldsymbol{m}$. A transition $t_j \in T$ is enabled at $\boldsymbol{m}$ iff $\forall p_i \in^\bullet t_j$, $m(p_i) > 0$ and its enabling degree is given by

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

which represents the maximum amount in which t_j can fire. Transition t_j is called *k-enabled* under marking $\boldsymbol{m}$, if $enab(t, \boldsymbol{m}) = k$. 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.

If $\boldsymbol{m}$ is reachable from $\boldsymbol{m}_0$ through a finite sequence σ, the state (or fundamental) equation is satisfied: $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \vec{\boldsymbol{\sigma}}$, where $\vec{\boldsymbol{\sigma}} \in \mathbb{R}_{\geq 0}^{|T|}$ is the *firing count vector*, i.e., $\vec{\sigma}(t_j)$ is the cumulative amount of firings of t_j in the sequence σ. A vector $\vec{\boldsymbol{\sigma}}$ is said to be a fireable firing count vector, if there exist a corresponding sequence σ which can be fired.

Marked Graph (MG) is a well known subclass of Petri nets in which each place has at most one input and at most one output arc. Thus they are structurally choice-free, allow concurrency and synchronization but not decisions.

***Property** 2.1:* [5] Let $\mathcal{N}$ be a strong connected marked graph, $\mathcal{N}$ is *consistent* and its unique minimal T-*semiflow* is $\boldsymbol{x} = \mathbf{1}$, $\mathbf{1}$ is a vector with all component equal to 1.

In timed continuous Petri net($ContPN$) the state equation has an explicit dependence on time: $\boldsymbol{m}(\tau) = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \vec{\boldsymbol{\sigma}}(\tau)$ which through time differentiation becomes $\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \dot{\vec{\boldsymbol{\sigma}}}(\tau)$. The derivative of the firing sequence $\boldsymbol{f}(\tau) = \dot{\vec{\boldsymbol{\sigma}}}(\tau)$ is called the *firing flow*. Depending on how the flow is defined, many firing semantics appear, being the most used ones *infinite* and *finite* server semantics [17]. For a broad class of Petri nets it is shown that infinite server semantics offers better approximation to discrete systems than finite server semantics [13]. This paper deals with infinite server semantics for which the flow of a transition t_j at time τ is the product of the firing rate, λ_j, and the enabling degree of the transition at $\boldsymbol{m}(\tau)$

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

For the sake of clarity, τ will be omitted in the rest of the paper when there is no confusion: $f(t_j)$, $\boldsymbol{m}$ and $m(p_i)$ will be used instead of $f(t_j, \tau)$, $\boldsymbol{m}(\tau)$ and $m(p_i, \tau)$.

B. Implicit places and continuous marked graphs

A place p is called *implicit* when it is never the unique place restricting the firing of its output transitions. Hence, an implicit place can be removed without affecting the behavior of the rest of the system, i.e., the language of firing sequences of the original system is preserved [16].

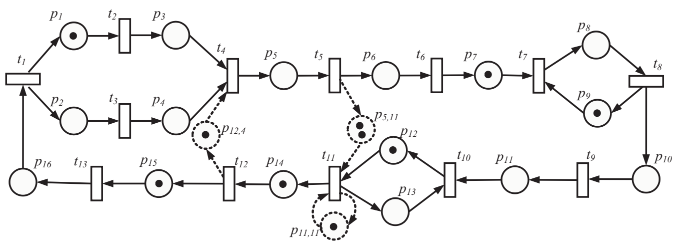


Fig. 1. Marked Graph and Marking Structurally Implicit Places, with initial marking $m_0(p_1) = m_0(p_7) = m_0(p_9) = m_0(p_{12}) = m_0(p_{14}) = m_0(p_{15}) = 1$.

Normally, implicit places are determined by the structure but also depend on their initial markings. A place p that can be made implicit (with a proper initial marking $m_0(p)$) for any initial marking of the rest of the system is called *structurally implicit place*. A structurally implicit place whose minimal initial marking can be deduced from the marking of other places is said to be *marking structurally implicit*, more formally:

***Definition** 2.2:* [18] Let $\mathcal{N} = \langle P \cup p, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$. The place p is *marking structurally implicit place*, iff there exists $\boldsymbol{y} \geq 0$, such that $\boldsymbol{C}(p, T) = \boldsymbol{y} \cdot \boldsymbol{C}(P, T)$.

For strongly connected marked graphs, a marking structurally implicit place p verifies:

$$C(p, \cdot) = \sum_{p_j \in \pi} C(p_j, \cdot) \text{ for } \forall \pi \in \mathcal{P}(t_e, t_s) \tag{2}$$

where $t_e =^\bullet p$, $t_s = p^\bullet$, $\mathcal{P}(t_e, t_s)$ is the set of simple paths (i.e., the paths without repeated nodes), from t_e to t_s [6].

It is proved in [6] that, given a marking structurally implicit place p, the minimal initial marking to make p implicit is:

$$m_0(p) = m_0^{min}(p) = \min \left\{ \sum_{p_j \in \pi} m_0(p_j) | \pi \in \mathcal{P}(t_e, t_s) \right\} \tag{3}$$

***Example** 2.1:* Fig. 1 shows a marked graph. It is easy to observe that from t_{12} to t_4 there exist two simple paths, $\pi_1 = \{t_{12}p_{15}t_{13}p_{16}t_1p_1t_2p_3t_4\}$, $\pi_2 = \{t_{12}p_{15}t_{13}p_{16}t_1p_2t_3p_4t_4\}$. Therefore, $\mathcal{P}(t_{12}, t_4) = \{\pi_1, \pi_2\}$.

With respect to $\mathcal{P}(t_{12}, t_4)$, the added place p_{12_4} is marking structurally implicit with input transition t_{12} and output transition t_4. Similarly, if considering the path from t_5 to t_{11}, $\pi_3 = \{t_5p_6t_6p_7t_7p_8t_8p_{10}t_9p_{11}t_{10}p_{12}t_{11}\}$, p_{5_11} is the corresponding marking structurally implicit place. There is a loop path from t_{11}, $\pi_4 = \{t_{11}, p_{13}, t_{10}, p_{12}, t_{11}\}$, therefore p_{11_11} is constructed.

In order to compute minimal initial marking to make p_{12_4} implicit, the sum of markings in each path from t_{12} to t_4 is considered. Because the sum of markings of places in π_1 is 2, while for π_2 it is 1, according to (3), the minimal is chosen, so one token should be put into p_{12_4}. Similarly, two tokens in p_{5_11}, and one token in p_{11_11}.

When the net system is considered as continuous, the minimal initial marking of marking structurally implicit places can also be calculated using (3).

III. Problem Statement

The classical centralized control theory has been proved inefficient for large scale distributed systems, in which the communication delay, time synchronization problems become significant. Therefore distributed or decentralized control is extensively explored in recent decades. In a distributed controlled system, normally a complex dynamic system, the controllers are not centralized in one location, but are distributed in the subsystems, while typically, each controller can only access local resources and limited information from its neighbor subsystems.

Under the framework of *ContPN*, the large scale system is decomposed into subsystems that are modeled with *ContPNs* and controlled by the local controllers. Each local controller can obtain information from its neighbor subsystems through the interface places and transitions. The problem we deal with is: how to design the control action for each local controller which works independently, and drive the system from initial marking $\boldsymbol{m}_0$ to final marking $\boldsymbol{m}_f$.

IV. Structural Decomposition of Marked Graphs

In this section we adapt the decomposition methods developed in [6]. The idea is the following: given a strongly connected marked graph $\mathcal{N}$, it is first split into two subnets $\mathcal{N}_1$ and $\mathcal{N}_2$ according to a set of places $B \subseteq P$, after that the *complemented subnets* ($\mathcal{CN}$) are derived through adding marking structurally implicit places.

***Definition** 4.1:* Let $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}\rangle$ be a strongly connected marked graph, $B \subseteq P$ is said to be a *cut* iff there exists subnets $\mathcal{N}_i = \langle P_i, T_i, \boldsymbol{Pre}_i, \boldsymbol{Post}_i\rangle$, $i = 1, 2$, such that:

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

(i) $P_1 \cup P_2 = P$, $P_1 \cap P_2 = B$

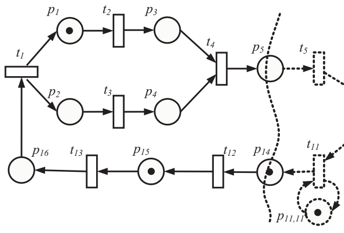
(ii) $P_1 = ^\bullet T_1 \cup T_1^\bullet$, $P_2 = ^\bullet T_2 \cup T_2^\bullet$

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

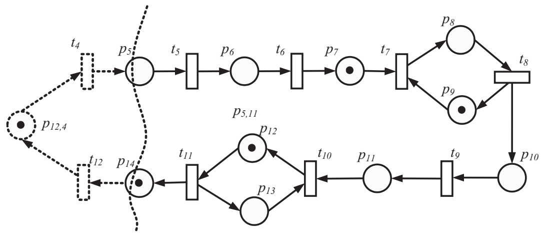
***Example** 4.1:* The non-dotted part in Fig. 2 are the subnets $\mathcal{N}_1$, $\mathcal{N}_2$ obtained from the marked graph in Fig. 1, which is cut by $B = \{p_5, p_{14}\}$, with the interface $U = \{t_4, t_5, t_{11}, t_{12}\}$ while $U_1 = \{t_4, t_{12}\}$, $U_2 = \{t_5, t_{11}\}$.

After cutting, the two subsystems $\mathcal{N}_1, \mathcal{N}_2$ are independent, because all the constraints from the rest of the system are removed. Therefore different behaviors are introduced. The *complemented subnet* is obtained after adding marking structurally implicit places as approximations of other parts of the system that are missing.

***Definition** 4.2:* Let $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}\rangle$ be a strongly connected marked graph, $\mathcal{N}_i = \langle P_i, T_i, \boldsymbol{Pre}_i, \boldsymbol{Post}_i\rangle$ be the subnets associated with a cut B. The *complemented subnet* $\mathcal{CN}_i$ is obtained from $\mathcal{N}_i$ by copying transitions in U_j and adding the marking structurally implicit places with respect to the paths $\mathcal{P}(t_e, t_s)$ in $\mathcal{N}_j$, $t_e, t_s \in U_j$, $i, j = 1, 2, i \neq j$. The set of places being added to $\mathcal{N}_i$ is denoted by IP_i .



(a) $\mathcal{N}_1$(non-dotted part) and $\mathcal{CN}_1$ (with dotted part)



(b) $\mathcal{N}_2$(non-dotted part) and $\mathcal{CN}_2$ (with dotted part)

Fig. 2. Cutting of marked graph

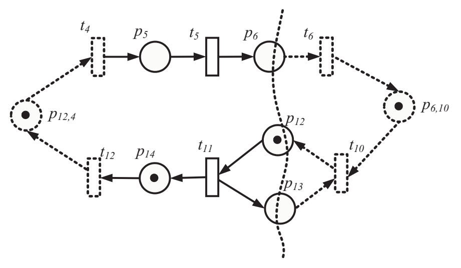
In Fig. 2, the complemented subnet $\mathcal{CN}_1$ is obtained after copying $U_2 = \{t_5, t_{11}\}$ and adding $IP_1 = \{p_{5\_11}, p_{11\_11}\}$ to $\mathcal{N}_1$, while $\mathcal{CN}_2$ is obtained after copying $U_1 = \{t_4, t_{12}\}$ and adding $IP_2 = \{p_{12\_4}\}$ to $\mathcal{N}_2$. Notice that cut B and interface U are shared in both subnets.

In order to calculate the initial marking of p_{e_s} that makes it implicit, we have to find out the path from t_e to t_s such that (3) is satisfied. There are some efficient algorithms which can be used, e.g., the algorithm of Floyd-Warshall [1] with a computation complexity of $O(|T|^3)$, where $|T|$ is the number of transitions.

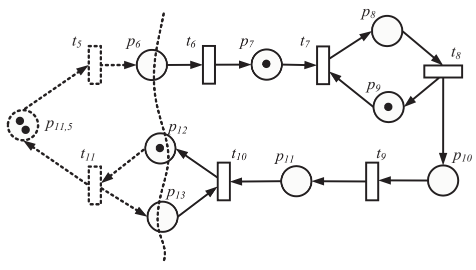
Sometimes for a complex system, only one cut is not sufficient, because the complemented subsets are still difficult to be handled. Therefore, the above decomposition process need to be executed in multiple hierarchical levels. Fig. 3 presents the complemented subnets obtained after cutting $\mathcal{CN}_2$ in Fig. 2(b) one more time, with $B = \{p_6, p_{12}, p_{13}\}$. After this two level cutting, the original system is decomposed into three: $\mathcal{CN}_1$, $\mathcal{CN}_{21}$ and $\mathcal{CN}_{22}$. It should be noticed that the order of cutting is not important, if the net in Fig. 1 is first cut by $B_1 = \{p_6, p_{12}, p_{13}\}$, then by $B_2 = \{p_5, p_{14}\}$, the exactly same subnets are obtained.

V. Distributed Control of Large Scale Systems

The distributed structure of a large scale system is obtained using the the decomposition method presented in section IV. In this section we will show that the *ON-OFF* controller



(a) $\mathcal{CN}_{21}$



(b) $\mathcal{CN}_{22}$

Fig. 3. Complemented subnets: second cut from $\mathcal{CN}_2$

developed in [19] can be applied to each subsystem, leading to the overall final state in minimum-time.

A firing count vector $\vec{\boldsymbol{\sigma}}$ driving the system to $\boldsymbol{m}_f$ is said to be *minimal* if it can not be expressed as the sum of other ones and T-*semiflows* [19] . An *ON-OFF* controller for structurally persistent $ContPN$ is proposed in [19]: if $\vec{\boldsymbol{\sigma}}$ is minimal, for any t_j, simply let it *ON* (with control input equal to 0) before the accumulated flow of t_j reaches $\vec{\sigma}(t_j)$, and after that suddenly let it *OFF* (with control input equal to $f(t_j)$). $\boldsymbol{m}_f$ is reached in minimum time using this strategy.

Because marked graphs is a subclass of structurally persistent nets, this *ON-OFF* strategy can be applied. In a distributed system, local controllers can only access limited resources, so the global control law (minimal firing count vector) cannot be obtained directly. In the following, it is shown how it is computed in a distributed way.

In the sequel, we will use the following notations:

(1) $\boldsymbol{m}_0^i$: the initial marking of $\mathcal{CN}_i$, directly projected from $\boldsymbol{m}_0$. For every $p \in P_i$, $m_0^i(p) = m_0(p)$, while for every added implicit place $p \in IP_i$, $m_0^i(p) = m_0^{min}(p)$.
(2) $\boldsymbol{m}_f^i$: the finial marking of $\mathcal{CN}_i$, directly projected from m_f. For every $p \in P_i$, $m_f^i(p) = m_f(p)$. Every place $p \in IP_i$ belongs to different circuit in $\mathcal{CN}_i$, and since $\mathcal{CN}_i$ is a strongly connected marked graph, each circuit forms a P-*semiflow* [15], $m_f^i(p)$ can be easily computed.
(3) $\vec{\boldsymbol{\sigma}}$: the minimal firing count vector driving $\mathcal{N}$ from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$.
(4) $\vec{\boldsymbol{\sigma}}^i$: the firing count vector of $\mathcal{CN}_i$ directly projected from $\vec{\boldsymbol{\sigma}}$. For every $t \in T_i$, $\vec{\sigma}^i(t) = \vec{\sigma}(t)$.
(5) $\vec{\boldsymbol{\sigma}}_i$: the firing count vector driving $\mathcal{CN}_i$ from $\boldsymbol{m}_0^i$ to $\boldsymbol{m}_f^i$. It is assumed to be minimal when there is no additional specification.

A. Decomposition with One Cut

The most interesting point of the decomposition approach in section IV is: if we put proper initial markings in the added marking structurally implicit places making them implicit, the projections of reachable markings and firing sequences of the original system are preserved in the complemented subnets [6], i.e., $\vec{\boldsymbol{\sigma}}^i$ can always be fired in $\mathcal{CN}_i$ with initial marking $\boldsymbol{m}_0^i$, and leading to $\boldsymbol{m}_f^i$. In the framework of continuous net system, this is also true, and the exactly same proof can be constructed.

***Definition** 5.1:* Firing count vectors $\vec{\boldsymbol{\sigma}}_1$ and $\vec{\boldsymbol{\sigma}}_2$ are said to be *compatible* if $\vec{\sigma}_1(t) = \vec{\sigma}_2(t)$, $\forall t \in U$.

***Definition** 5.2:* Let $\vec{\boldsymbol{\sigma}}_1$ and $\vec{\boldsymbol{\sigma}}_2$ be compatible firing count vectors. Operator $\oplus : \langle \mathbb{R}_{\geq 0}^{|T_1|}, \mathbb{R}_{\geq 0}^{|T_2|} \rangle \rightarrow \mathbb{R}_{\geq 0}^{|T_1 \cup T_2|}$ is defined to be the *merge*, such that, $\vec{\boldsymbol{\sigma}}_{12} = \vec{\boldsymbol{\sigma}}_1 \oplus \vec{\boldsymbol{\sigma}}_2$, $\forall t \in T_i$, $\vec{\sigma}_{12}(t) = \vec{\sigma}_i(t)$, $i = 1, 2$.

***Example** 5.1:* Let us consider the marked graph in Fig. 1 and its complemented subnets obtained using cut $B = \{p_5, p_{14}\}$, and interface $U = \{t_4, t_5, t_{11}, t_{12}\}$ in Fig. 2. Table I shows their initial, final markings and the firing count vectors. The initial markings of the added marking structurally implicit places are computed from (3), while their finial markings are $m_f^1(p_{5\_11}) = 2.1$, $m_f^1(p_{11\_11}) = 1$, and $m_f^2(p_{12\_4}) = 1.5$. We can notice that the firing count vector of $\mathcal{CN}_i$ projected from $\vec{\boldsymbol{\sigma}}$ may be not minimal, e.g., in $\mathcal{CN}_1$, the minimal vector is $\vec{\boldsymbol{\sigma}}_1 = [1.1\ 1.7\ 0.9\ 0.5\ 0.1\ 0\ 1\ 1.5]^T$, while the projection $\vec{\boldsymbol{\sigma}}^1 = \vec{\boldsymbol{\sigma}}_1 + 0.6 \cdot \mathbf{1}$. For sure $\vec{\boldsymbol{\sigma}}^1$ is fireable in $\mathcal{CN}_1$ since $\mathbf{1}$ is the T-*semiflow*.

TABLE I

MARKINGS AND FIRING COUNT VECTORS

P	$\boldsymbol{m}_0$ $(\boldsymbol{m}_f)$	$\boldsymbol{m}_0^1$ $(\boldsymbol{m}_f^1)$	$\boldsymbol{m}_0^2$ $(\boldsymbol{m}_f^2)$	T	$\vec{\boldsymbol{\sigma}}$	$\vec{\boldsymbol{\sigma}}^1$ $(\vec{\boldsymbol{\sigma}}_1)$	$\vec{\boldsymbol{\sigma}}^2$ $(\vec{\boldsymbol{\sigma}}_2)$
p_1	$1(0.4)$	$1(0.4)$		t_1	1.7	1.7(1.1)	
p_2	$0(0.2)$	$0(0.2)$		t_2	2.3	2.3(1.7)	
p_3	$0(1.2)$	$0(1.2)$		t_3	1.5	1.5(0.9)	
p_4	$0(0.4)$	$0(0.4)$		t_4	1.1	1.1(0.5)	1.1(1.1)
p_5	$0(0.4)$	$0(0.4)$	$0(0.4)$	t_5	0.7	0.7(0.1)	0.7(0.7)
p_6	$0(0.2)$		$0(0.2)$	t_6	0.5		0.5(0.5)
p_7	$1(0.5)$		$1(0.5)$	t_7	1		1 (1)
p_8	$0(0.4)$		$0(0.4)$	t_8	0.6		0.6(0.6)
p_9	$1(0.6)$		$1(0.6)$	t_9	0.5		0.5(0.5)
p_{10}	$0(0.1)$		$0(0.1)$	t_{10}	0		0 (0)
p_{11}	$0(0.5)$		$0(0.5)$	t_{11}	0.6	0.6(0)	0.6(0.6)
p_{12}	$1(0.4)$		$1(0.4)$	t_{12}	1.6	1.6(1)	1.6(1.6)
p_{13}	$0(0.6)$		$0(0.6)$	t_{13}	2.1	2.1(1.5)	
p_{14}	$1(0)$	$1(0)$	$1(0)$				
p_{15}	$1(0.5)$	$1(0.5)$					
p_{16}	$0(0.4)$	$0(0.4)$					
$p_{5,11}$		$2(2.1)$					
$p_{11,11}$		$1(1)$					
$p_{12,4}$			$1(1.5)$				

Until now the time has been ignored. If all the transition are controllable, a marking $\boldsymbol{m}$ is reachable in the timed model, it is also reachable in the untimed one; while if a marking $\boldsymbol{m}$ is reachable in the untimed model, then it is asymptotically reachable in the timed one [12]. Therefore, similar results can be easily extended to $ContPN$. In particular, the projections of firing count vectors and reachable markings of the original system are preserved in the complemented subnets. In the following parts, we assume the system is live.

If the minimal firing count vectors of $\mathcal{CN}_1$ and $\mathcal{CN}_2$ are compatible, we will prove the merged vector is firable in $\mathcal{N}$. In the case they are not compatible, like $\vec{\boldsymbol{\sigma}}_1$ and $\vec{\boldsymbol{\sigma}}_2$ in Ex. 5.1 (because $\forall t \in U, \vec{\sigma}_1(t) \neq \vec{\sigma}_2(t)$), a T-*semiflow* can be added to make them compatible. Finally, the merged vector obtained is actually equal to $\vec{\boldsymbol{\sigma}}$.

***Proposition** 5.1:* Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a live marked graph, with cut B and corresponding interface U. Let $\vec{\boldsymbol{\sigma}}_i$ be the firing count vector driving $\mathcal{CN}_i$ from $\boldsymbol{m}_0^i$ to $\boldsymbol{m}_f^i$, $i = 1, 2$. If $\vec{\boldsymbol{\sigma}}_1$, $\vec{\boldsymbol{\sigma}}_2$ are compatible, then there exists a firing sequence σ_{12} with $\vec{\boldsymbol{\sigma}}_{12} = \vec{\boldsymbol{\sigma}}_1 \oplus \vec{\boldsymbol{\sigma}}_2$ that can be fired in $\mathcal{N}$ $\boldsymbol{m}_f$ is reached.

Proof: Since $\vec{\boldsymbol{\sigma}}_1$ and $\vec{\boldsymbol{\sigma}}_2$ are compatible, all the common transitions ($t \in U$) have the same firing counts. On the other side B are the common places of $\mathcal{CN}_1$ and $\mathcal{CN}_2$ and ${}^\bullet B \cup B^\bullet = U$, therefore,

$$\boldsymbol{m}_0 + \boldsymbol{C} \cdot \vec{\boldsymbol{\sigma}}_{12} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot (\vec{\boldsymbol{\sigma}}_1 \oplus \vec{\boldsymbol{\sigma}}_2) = \boldsymbol{m}_f$$

Because the system is a live marked graph, there always exists a sequence σ_{12} can be fired. ■

***Property** 5.1:* Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a live marked graph, with cut B and corresponding interface U. There exists $k \geq 0$, such that $\vec{\boldsymbol{\sigma}} = (k \cdot \mathbf{1} + \vec{\boldsymbol{\sigma}}_i) \oplus \vec{\boldsymbol{\sigma}}_j$, $i, j = 1, 2, i \neq j$.

Proof: Since $\vec{\boldsymbol{\sigma}}^i$ is a fireable vector in $\mathcal{CN}_i$ reaching $\boldsymbol{m}_f^i$, and $\vec{\boldsymbol{\sigma}}_i$ is the corresponding minimal firing count vector, considering $\mathcal{CN}_i$ is live marked graph with the unit vector as the unique T-*semiflow*, we have $\vec{\boldsymbol{\sigma}}^i = \vec{\boldsymbol{\sigma}}_i + \alpha_i \cdot \mathbf{1}$, $\alpha_i \geq 0, i = 1, 2$, without loss of generality, assume $\alpha_1 \leq \alpha_2$. In particular, for any $t \in U$, $\vec{\sigma}^i(t) = \vec{\sigma}_i(t) + \alpha_i$, because $\vec{\sigma}^1(t) = \vec{\sigma}^2(t) = \vec{\sigma}(t)$, $\vec{\sigma}_1(t) - \vec{\sigma}_2(t) = \alpha_2 - \alpha_1 = k \geq 0$.

Therefore $\vec{\boldsymbol{\sigma}}_1$ and $\vec{\boldsymbol{\sigma}}_2 + k \cdot \mathbf{1}$ are compatible, according to Proposition 5.1 $\vec{\boldsymbol{\sigma}}_{12} = (k \cdot \mathbf{1} + \vec{\boldsymbol{\sigma}}_2) \oplus \vec{\boldsymbol{\sigma}}_1$ is fireable in $\mathcal{CN}$ and $\boldsymbol{m}_f$ is reached. Since $\vec{\boldsymbol{\sigma}}_1$, $\vec{\boldsymbol{\sigma}}_2$ are minimal, $\vec{\boldsymbol{\sigma}}_{12}$ is also minimal. Because the minimal firing count vector is unique in live marked graph [19], $\vec{\boldsymbol{\sigma}}_{12} = \vec{\boldsymbol{\sigma}}$. ■

***Example** 5.2:* In Ex. 5.1, $\vec{\boldsymbol{\sigma}}_1$ and $\vec{\boldsymbol{\sigma}}_2$ are not compatible, with difference $k = 0.6$ (i.e., $\forall t \in U, \vec{\sigma}_2(t) - \vec{\sigma}_1(t) = 0.6$). Clearly, after adding $0.6 \cdot \mathbf{1}$ to $\vec{\boldsymbol{\sigma}}_1$, they can be merged, and $\vec{\boldsymbol{\sigma}}$ is obtained, i.e., $\vec{\boldsymbol{\sigma}} = (\vec{\boldsymbol{\sigma}}_1 + 0.6 \cdot \mathbf{1}) \oplus \vec{\boldsymbol{\sigma}}_2$.

Notice that in the example, $\vec{\boldsymbol{\sigma}}_1$ is different from the direct projection from $\vec{\boldsymbol{\sigma}}$, while $\vec{\boldsymbol{\sigma}}_2$ is equal to $\vec{\boldsymbol{\sigma}}^2$. In fact, if $\vec{\boldsymbol{\sigma}} = (k \cdot \mathbf{1} + \vec{\boldsymbol{\sigma}}_i) \oplus \vec{\boldsymbol{\sigma}}_j$, then $\vec{\sigma}(t) = \vec{\sigma}_j(t), t \in T_j$.

B. Decomposition with Hierarchical Cut

Let us consider the case when the system is decomposed hierarchically. When cutting a net into two parts $\mathcal{CN}_1, \mathcal{CN}_2$, and suppose $\vec{\boldsymbol{\sigma}} = (k_1 \cdot \mathbf{1} + \vec{\boldsymbol{\sigma}}_1) \oplus \vec{\boldsymbol{\sigma}}_2$, then $\vec{\sigma}(t) = \vec{\sigma}_2(t), t \in T_2$. If cutting $\mathcal{CN}_2$ one more time into $\mathcal{CN}_{21}$ and $\mathcal{CN}_{22}$, and suppose $\vec{\boldsymbol{\sigma}}_2 = (k_2 \cdot \mathbf{1} + \vec{\boldsymbol{\sigma}}_{21}) \oplus \vec{\boldsymbol{\sigma}}_{22}$, then $\vec{\sigma}_2(t) = \vec{\sigma}_{22}(t), \forall t \in T_{22}$. Therefore we have $\vec{\sigma}(t) = \vec{\sigma}_{22}(t), \forall t \in T_{22}$. The same result can be obtained when $\mathcal{CN}_{22}$ is cut again, hence it can be concluded: there always exists at least one complemented subnet $\mathcal{CN}_i$, such that $\vec{\sigma}(t) = \vec{\sigma}_i(t), \forall t \in T_i$, and $\mathcal{CN}_i$ is said to be *critical*.

Two complemented subnets are neighbors if they share a cut. Because every time we split one net into two, each subnets has at least one neighbor. We will prove it is possible to make pairs of minimal firing vectors of neighbors to be compatible and obtain $\vec{\boldsymbol{\sigma}}$ after merging all of them.

***Proposition** 5.2:* Let $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ be a live marked graph that is decomposed into n subnets. Assuming $\mathcal{CN}_q$, $1 \leq q \leq n$ is a critical complemented subnet, then there exist $x_i, i = 1, 2, ..., n$ such that:

$$\vec{\boldsymbol{\sigma}} = \bigoplus_{i=1}^{n} (\vec{\boldsymbol{\sigma}}_i + x_i \cdot \mathbf{1}) \tag{4}$$

and if $i = q, x_i = 0$, else $x_i \geq 0$.

Proof: Since all the complemented subnets are still live marked graphs, $\vec{\boldsymbol{\sigma}}_i + x_i \cdot \mathbf{1}$ is also fireable in $\mathcal{CN}_i$. For any two neighbor subnets $\mathcal{CN}_i$, $\mathcal{CN}_j$, $x_i, x_j \geq 0$ can be found such that $\vec{\boldsymbol{\sigma}}_i + x_i \cdot \mathbf{1}$ and $\vec{\boldsymbol{\sigma}}_j + x_j \cdot \mathbf{1}$ are compatible. According to Proposition 5.1, $(\vec{\boldsymbol{\sigma}}_i + x_i \cdot \mathbf{1}) \oplus (\vec{\boldsymbol{\sigma}}_j + x_j \cdot \mathbf{1})$ is fireable in the net composed by $\mathcal{CN}_i$ and $\mathcal{CN}_j$. Therefore after merging all the firing count vectors, $\vec{\boldsymbol{\sigma}}' = \bigoplus_{i=1}^{n} (\vec{\boldsymbol{\sigma}}_i + x_i \cdot \mathbf{1})$ is obtained, which can be fired in $\mathcal{N}$, and reach the final marking.

If every $x_i > 0$, $\vec{\boldsymbol{\sigma}}'$ is not a minimal firing count vector, then certain amount of T-*semiflow* can be subtracted from $\vec{\boldsymbol{\sigma}}'$ until $\vec{\boldsymbol{\sigma}} = \vec{\boldsymbol{\sigma}}'$. Since $\mathcal{CN}_q$ is critical, $x_q = 0$. ■

Let us observe that it is possible to have more than one critical subnet, but considering there is unique minimal firing count vector in a live marked graph, given any one of the critical subnets, the same $\vec{\boldsymbol{\sigma}}$ is constructed.

***Example** 5.3:* Let's examine the marked graph in Fig. 1 with the initial and final markings as listed in Table. I. After cutting with $B_1 = \{p_5, p_{14}\}$ and $B_2 = \{p_6, p_{12}, p_{13}\}$, we get three complemented subnets $\mathcal{CN}_1$ (Fig. 2(a)), $\mathcal{CN}_{21}$, $\mathcal{CN}_{22}$ (Fig. 3). $\mathcal{CN}_1$ and $\mathcal{CN}_{21}$ are neighbors sharing cutting B_1, $\mathcal{CN}_{21}$ and $\mathcal{CN}_{22}$ are neighbors sharing B_2. In Table II is the minimal firing count vectors for reaching corresponding final markings. It is obtained:

$$\vec{\boldsymbol{\sigma}} = (\vec{\boldsymbol{\sigma}}_1 + 0.6 \cdot \mathbf{1}) \oplus \vec{\boldsymbol{\sigma}}_2 \oplus \vec{\boldsymbol{\sigma}}_3$$

Here $\mathcal{CN}_{21}, \mathcal{CN}_{22}$ are critical subnets.

The rest of this section devotes to design an effective algorithm to search a critical subnet, and calculating corresponding x_i to generate $\vec{\boldsymbol{\sigma}}$.

In order to make it more understandable, let us construct a graph $G = \langle V, W \rangle$ to depict the relations among complemented subnets. Each node $v \in V$ represents a subnet, there are arcs between v_i and v_j if the corresponding subnets $\mathcal{CN}_i$ and $\mathcal{CN}_j$ are neighbors. The weight of the arc from v_i to v_j is $w(v_i, v_j) = \vec{\sigma}_i(t) - \vec{\sigma}_j(t), t \in U$, negative weight is also allowed here. So in the corresponding graph G (Fig. 4) the weight $w(v_2, v_1) = 0.6$, $w(v_1, v_2) = -0.6$, while

TABLE II

MINIMAL FIRING COUNT VECTORS

T	$\vec{\boldsymbol{\sigma}}(\mathcal{N})$	$\vec{\boldsymbol{\sigma}}_1(\mathcal{CN}_1)$	$\vec{\boldsymbol{\sigma}}_2(\mathcal{CN}_{21})$	$\vec{\boldsymbol{\sigma}}_3(\mathcal{CN}_{22})$
t_1	1.7	1.1		
t_2	2.3	1.7		
t_3	1.5	0.9		
t_4	1.1	0.5	1.1	
t_5	0.7	0.1	0.7	0.7
t_6	0.5		0.5	0.5
t_7	1			1
t_8	0.6			0.6
t_9	0.5			0.5
t_{10}	0		0	0
t_{11}	0.6	0	0.6	0.6
t_{12}	1.6	1	1.6	
t_{13}	2.1	1.5		

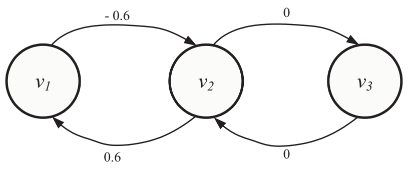


Fig. 4. The graph $G = \langle V, W \rangle$ constructed from the three complemented subnets in Ex. 5.3

$w(v_2, v_3) = w(v_3, v_2) = 0$. Denote $W(v_i, v_j)$ the sum of the weights on the simple path from v_i to v_j.

Since a cut splits a net into two subnets, in graph G there only exists one directed simple path from nodes v_i to v_j (also from v_j, v_i), and obviously $W(v_i, v_j) = -W(v_j, v_i)$.

It can be observed that, the sum of weight in the path $\langle v_i, v_j \rangle$ reflects the *relative difference* of $\vec{\boldsymbol{\sigma}}_i$ to $\vec{\boldsymbol{\sigma}}_j$. In Ex. 5.3, the relative difference of v_3 to v_2 is $w(v_3, v_2) = 0$, while the one of v_3 to v_1 is $w(v_3, v_2) + w(v_2, v_1) = 0.6$. Obviously we have $\vec{\boldsymbol{\sigma}} = (\vec{\boldsymbol{\sigma}}_1 + 0.6 \cdot \mathbf{1}) \oplus (\vec{\boldsymbol{\sigma}}_2 + 0 \cdot \mathbf{1}) \oplus \vec{\boldsymbol{\sigma}}_3$. Actually, the non negative value x_i in (4) is equal to $W(v_q, v_i)$.

***Property** 5.2:* If for any node $v_i \in V$, $W(v_q, v_i) \geq 0$, then $\mathcal{CN}_q$ is a critical subnet.

Proof: If $W(v_q, v_i) \geq 0$, then let $x_i = W(v_q, v_i)$, $\vec{\boldsymbol{\sigma}}$ can be constructed. Therefore $x_q = W(v_q, v_q) = 0$, $\mathcal{CN}_q$ is critical.

■

Algorithm 1 shows how to search a critial subnet based on the graph constructed. In the beginning all the nodes are labeled as *new*, then every time a *new* labeled node, denoted by v_i is chosen, the relative differences from v_i to others nodes v_j, $W(v_i, v_j)$ is calculated (if it is not done before). If it is negative then v_i is not critical and labeled as *old*. If it is positive then v_j is not critical because $W(v_j, v_i)$ must be negative, and we don't need to check v_j in the future. When a node with all relative differences non-negative is found, or there is only one node left which is labeled as *new*, the program finishes. When calculating the sum of weights, of course the intermediate value that has been calculated before should be reused. In the worst case, the computing complexity is $O(\frac{n(n-1)}{2})$, where n is the number of complemented subnets.

Algorithm 1 Search a critical subnet

Input: $G = \langle V, W \rangle$

Output: A node $v_q \in V$

```
1:  Label all the nodes in $V$ as new;
2:  while more than one node in $V$ is labeled as new do
3:    Choose a node $v_i$ from $V$ which is labeled as new;
4:    for j = 1 to n do
5:      if $W(j, i)$ has not been calculated then
6:        calculate $W(i, j)$;
7:        if $W(i, j) > 0$ then
8:          label $v_j$ as old;
9:        else if $W(i, j) < 0$ then
10:         label $v_i$ as old;
11:         break;
12:       end if
13:     end if
14:   end for
15:   if $j = n$ and $v_i$ is labeled as new then
16:     return $v_i$;
17:   end if
18: end while
19: return The last node in $V$ that is labeled as new
```

We will assume there is a supervisory controller in our system structure, whose work is to search a critical subnet, and send the relative difference x_i to the local controller of $\mathcal{CN}_i$. The local controller receives this value then updates the local control law. Algorithm 2, 3 are for supervisory controller, local controller respectively.

Algorithm 2 Supervisory Controller

Input: $\vec{\boldsymbol{\sigma}}_i$

Output: $\boldsymbol{x}$

```
1: Construct the graph $G = \langle V, W \rangle$;
2: Find out a critical subnet $\mathcal{CN}_q$ using Algorithm 1;
3: Compute $\boldsymbol{x}(i)$: the relative difference of $\mathcal{CN}_q$ to $\mathcal{CN}_i$;
4: Send $\boldsymbol{x}(i)$ to subnet $\mathcal{CN}_i$, $i = 1, 2, ..., n$;
```

VI. CASE STUDY

Let us consider the $ContPN$ system in Fig. 5 which models a manufacturing system with three types of product lines which are assembled for one final product. The system is cut into four subsystems through the buffer places ($B_1 = \{p_1, p_{12}\}$, $B_2 = \{p_{13}, p_{23}\}$, $B_3 = \{p_{24}, p_{38}\}$) of each product line, as shown in Fig. 6, where $p_{8,31}$, $p_{27,31}$, $p_{1,7}$, $p_{9,15}$ and $p_{16,24}$ are the added marking structurally implicit places.

Algorithm 3 Local Controller i

```
Input: $\mathcal{CN}_i$, $\boldsymbol{m}_0^i$, $\boldsymbol{m}_f^i$
Output: $\vec{\boldsymbol{\sigma}}_i$
1: Calculate the minimal firing count $\vec{\boldsymbol{\sigma}}_i$ to reach $\boldsymbol{m}_f^i$;
2: Send $\vec{\boldsymbol{\sigma}}_i$ to the supervisory controller;
3: Receive $\boldsymbol{x}(i)$ from the supervisory controller;
4: Update $\vec{\boldsymbol{\sigma}}_i$ to $\vec{\boldsymbol{\sigma}}_i + \boldsymbol{x}(i) \cdot \mathbf{1}$;
5: Apply ON-OFF control;
```

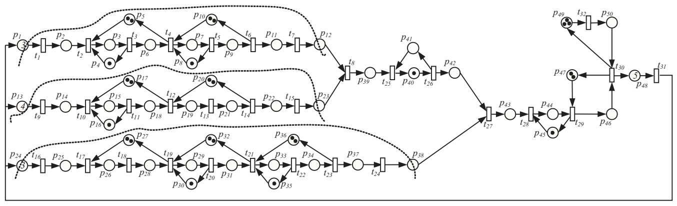


Fig. 5. A manufacturing system model

Assuming the initial and desired final marking are listed in Table III. The corresponding minimal firing count vector are easy to calculate, shown in Table IV.

TABLE III

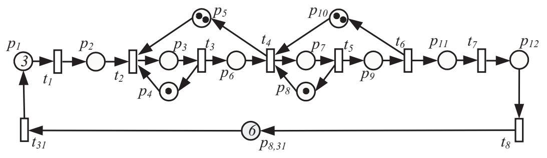
INITIAL AND FINAL MARKINGS

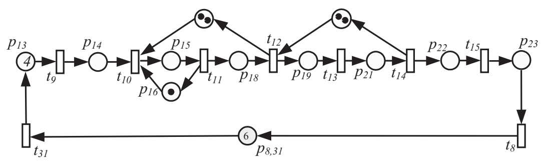
$\mathcal{CN}_1$		$\mathcal{CN}_2$		$\mathcal{CN}_3$		$\mathcal{CN}_4$	
P	$\boldsymbol{m}_0$ $(\boldsymbol{m}_f)$	P	$\boldsymbol{m}_0$ $(\boldsymbol{m}_f)$	P	$\boldsymbol{m}_0$ $(\boldsymbol{m}_f)$	P	$\boldsymbol{m}_0$ $(\boldsymbol{m}_f)$
p_1	3(0.6)	p_{13}	4(1.9)	p_{24}	3(0.3)	p_1	3(0.6)
p_2	0(0.4)	p_{14}	0(0.3)	p_{25}	0(0.2)	p_{12}	0(0.4)
p_3	0(0.8)	p_{15}	0(0.9)	p_{26}	0(0.7)	p_{13}	4(1.9)
p_4	1(0.2)	p_{16}	1(0.1)	p_{27}	2(0.5)	p_{23}	0(0.3)
p_5	2(0.5)	p_{17}	2(0.5)	p_{28}	0(0.8)	p_{24}	3(0.3)
p_6	0(0.7)	p_{18}	0(0.6)	p_{29}	0(0.9)	p_{38}	0(0.2)
p_7	0(0.8)	p_{19}	0(0.9)	p_{30}	1(0.1)	p_{39}	0(0.6)
p_8	1(0.2)	p_{20}	2(0.5)	p_{31}	0(0.6)	p_{40}	1(0.8)
p_9	0(0.7)	p_{21}	0(0.6)	p_{32}	2(0.5)	p_{41}	0(0.2)
p_{10}	2(0.5)	p_{22}	0(0.3)	p_{33}	0(0.9)	p_{42}	0(0.2)
p_{11}	0(0.4)	p_{23}	0(0.3)	p_{34}	0(0.6)	p_{43}	0(0.4)
p_{12}	0(0.4)			p_{35}	1(0.1)	p_{44}	0(0.8)
				p_{36}	2(0.5)	p_{45}	1(0.2)
				p_{37}	0(0.2)	p_{46}	0(1.0)
				p_{38}	0(0.2)	p_{47}	2(1.0)
						p_{48}	5(0.4)
						p_{49}	3(1.5)
						p_{50}	0(1.5)
$p_{8,31}$	6(4.2)	$p_{8,31}$	6(4.2)	$p_{27,31}$	5(2.6)	$p_{1,7}$	0(3.8)
						$p_{9,15}$	0(3.6)
						$p_{16,24}$	0(4.9)

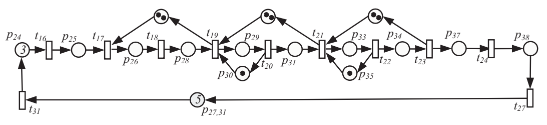
Based on Table. IV, graph G (Fig. 7) is constructed, in which $\mathcal{CN}_4$ is neighbor to all the other subnets with weight $w(v_4, v_1) = w(v_4, v_2) = 2.8$, $w(v_4, v_3) = 2.2$. If applying Algorithm 1, $\mathcal{CN}_4$ is selected.

The relative differences of $\mathcal{CN}_4$ to all the other subnets can be computed, which in this case is very straightforward: $x_1 = x_2 = 2.8$, $x_3 = 2.2$. Therefor the minimal firing count vector is generated as: $\vec{\boldsymbol{\sigma}} = (\vec{\boldsymbol{\sigma}}_1 + 2.8 \cdot \mathbf{1}) \oplus (\vec{\boldsymbol{\sigma}}_2 + 2.8 \cdot \mathbf{1}) \oplus (\vec{\boldsymbol{\sigma}}_3 + 2.2 \cdot \mathbf{1}) \oplus \vec{\boldsymbol{\sigma}}_4$.

Then, for instance, in order to control the transitions in $\mathcal{CN}_1$, the *ON-OFF* can be applied with control law $\vec{\boldsymbol{\sigma}}'_1 = \vec{\boldsymbol{\sigma}}_1 + 2.8 \cdot \mathbf{1}$, i.e., transition $t_j \in T_1$ is *ON* when the accumulated flow of t_j is less than $\vec{\sigma}'_1(t_j)$, else t_j is


(a) $\mathcal{CN}_1$


(b) $\mathcal{CN}_2$


(c) $\mathcal{CN}_3$

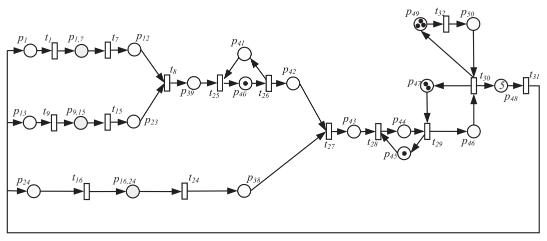

(d) $\mathcal{CN}_4$

Fig. 6. Complemented subnets from the system model in Fig. 5, with cut $B_1 = \{p_1, p_{12}\}$, $B_2 = \{p_{13}, p_{23}\}$, $B_3 = \{p_{24}, p_{38}\}$

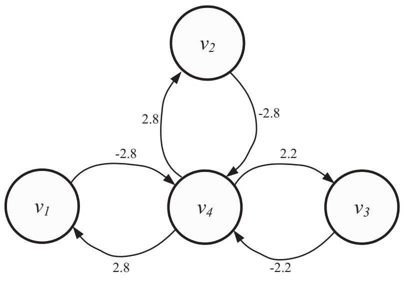


Fig. 7. The graph constructed from Table. IV

OFF. The global final marking is reached in 13.24 time units, which is minimal time as what has been proved in [19].

VII. CONCLUSIONS

Distributed control could be a solutions of controlling systems that are too complex to handle with centralized controller or the deployments of systems are physically distributed. This work focus on distributed control of large scale systems that are modeled with timed continuous Petri nets, aiming to drive the system from initial marking to desired final marking. The model is first decomposed into subnets with sets of places, then making structurally implicit

TABLE IV

MINIMAL FIRING COUNT VECTORS

$\mathcal{CN}_1$		$\mathcal{CN}_2$		$\mathcal{CN}_3$		$\mathcal{CN}_4$	
T	$\vec{\boldsymbol{\sigma}}_1$	T	$\vec{\boldsymbol{\sigma}}_2$	T	$\vec{\boldsymbol{\sigma}}_3$	T	$\vec{\boldsymbol{\sigma}}_4$
t_1	4.2	t_8	0	t_{16}	5.1	t_1	7
t_2	3.8	t_9	3.9	t_{17}	4.9	t_7	3.2
t_3	3	t_{10}	3.6	t_{18}	4.2	t_8	2.8
t_4	2.3	t_{11}	2.7	t_{19}	3.4	t_9	6.7
t_5	1.5	t_{12}	2.1	t_{20}	2.5	t_{15}	3.1
t_6	0.8	t_{13}	1.2	t_{21}	1.9	t_{16}	7.3
t_7	0.4	t_{14}	0.6	t_{22}	1	t_{24}	2.4
t_8	0	t_{15}	0.3	t_{23}	0.4	t_{25}	2.2
t_{31}	1.8	t_{31}	1.8	t_{24}	0.2	t_{26}	2.4
				t_{27}	0	t_{27}	2.2
				t_{31}	2.4	t_{28}	1.8
						t_{29}	1
						t_{30}	0
						t_{31}	4.6
						t_{32}	1.5

places are introduced to obtain complemented subnets and control laws can be computed in a distributed way. After that, *ON-OFF* control is applied in each subnet, and final marking is reached in minimum time.

Since the obtained subnets depend on the selection of cuts, an improper cut may lead to subnets with big size which are still difficult to handle. Therefore a qualified automatic cutting algorithm is worth to be investigated. In principle, a proper cut should generate subnets with similar size and the corresponding set of interface transitions should be small.

Another future work will be applying this control method to more general nets structures, for example structurally persistent nets, where the decomposition method and approximation strategy should be reconsidered.

REFERENCES

[1] A. V Aho, J. D Ullman, and J. E Hopcroft. *Data Structures and Algorithms*. Addison Wesley, 1983.

[2] 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.

[3] 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. to appear.

[4] F. Balduzzi, A. Giua, and C. Seatzu. Hybrid Control of Production Systems with Local Optimization. In *Proc. of the 7th IEEE Int. Conf. on Emerging Technologies and Factory Automation*, pages 1531–1540, Barcelona, Spain, October 1999.

[5] J. Campos, G. Chiola, J. M. Colom, and M. Silva. Properties and Performance Bounds for Timed Marked Graphs. *IEEE Transactions on Circuits and Systems I Fundamental Theory and Applications*, 39(5):386–401, 1992.

[6] J. Campos, J.M. Colom, H. Jungnitz, and M. Silva. *Approximate Throughput Computation of Stochastic Marked Graphs*. Springer, 1995.

[7] R. David and H.Alla. *Autonomous and timed continuous Petri nets*. Springer, Berlin, 2010.

[8] X. Guan and L. E. Holloway. Control of Distributed Discrete Event Systems Modeled as Petri Nets. In *Proceedings of the American Control Conference*, pages 2342–2347, Albuquerque, New Mexico, June 1997.

[9] L. E. Holloway, B. H. Krogh, and A. Giua. A Survey of Petri Net Methods for Controlled Discrete Event Systems. *Discrete Event Dynamic Systems*, 7(2):151–190, 1997.

[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. Springer Berlin / Heidelberg.

[11] P. Krishnan. Distributed timed automata. In *WDS'99, Workshop on Distributed Systems (A satellite workshop to FCT'99)*, volume 28 of *Electronic Notes in Theoretical Computer Science*, pages 5 – 21, 2000.

[12] 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.

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

[14] R. P. Moreno, D. Tardioli, and J.l.V. Salcedo. Distributed Implementation of Discrete Event Control Systems based on Petri nets. In *ISIE08: IEEE International Symposium on Industrial Electronics*, pages 1738–1745, June 2008.

[15] T. Murata. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77 No:4:541–580, 1989.

[16] M. Silva. *Las Redes de Petri en La Automática y la Informática*. Editorial AC, Madrid, 1985.

[17] 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.

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

[19] L. 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*, Atlanta, Georgia USA, December 2010. to appear.

[20] G. Yasuda. Design and implementation of Petri net based distributed control architecture for robotic manufacturing systems. In *MICAI'07: Proceedings of the artificial intelligence 6th Mexican international conference on Advances in artificial intelligence*, pages 1151–1161, Berlin, Heidelberg, 2007. Springer-Verlag.