Decentralized Control of Large Scale Systems Modeled with Continuous Marked Graphs

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

***Abstract*— This paper addresses the decentralized control of large scale systems that are modeled with timed continuous Marked Graphs ($ContMG$). Decentralized structures are first obtained decomposing the system 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). 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. In the framework of Petri nets, it leads to *Continuous Petri nets* [7], [17]. An important advantage of this relaxation is that more efficient algorithms are available for the analysis, e.g., reachability and controllability problems [13], [11].

Different works about control of Petri nets can be found in the literature [9], [4], [2], etc. In the context of decentralized (distributed) control, distributed timed automata is discussed in [12]. In [10], the decentralized control of Petri nets is considered by means of the supervision based on place invariants. An architecture for distributed implementation of Petri nets in control applications is proposed in [15]. A distributed control strategy is designed in [8] for forbidden state avoidance for discrete event systems which are modeled as Petri nets.

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.

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, DPI2010-20413 and TIN2007-66523.

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

In this work, the decentralized control of large scale systems which are modeled with timed continuous Petri nets is addressed. As a starting point, it is assumed that the systems we handle are modeled with marked graphs. 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.

Compared with the approach proposed in [3], in this method, subsystems do not have to be mono T-*semiflow*, and the final states of the places used to cut the system are specified in the control problem and can also be reached in minimum time.

This paper is organized as follows: Section II briefly recalls some basic concepts. Section III states the control problem, as it is here addressed. In Section IV, a structurally decomposition method for marked graphs is discussed, which is used here to obtain decentralized structures. Section V proposes the approach for decentralized 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, $i = 1, \dots, |P|$ and $t_j, j = 1, \dots, |T|$ 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

$\boldsymbol{m}$. 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 $0 < \alpha \le enab(t_j, \boldsymbol{m})$ 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 $\vec{\boldsymbol{\sigma}} \in \mathbb{R}_{\ge 0}^{|T|}$ 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 exactly one input arc and exactly one output arc. Thus they are structurally choice-free, allow concurrency and synchronization but not decisions.

***Property** 2.2:* [5] Let $\mathcal{N}$ be a MG, $\mathcal{N}$ is *consistent* and its unique minimal T-*semiflow* is $\boldsymbol{x} = \mathbf{1}$, where $\mathbf{1}$ 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 [14]. 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 discrete system is preserved [18].

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 linearly deduced from the marking of other places is said to be *marking structurally implicit*. More formally:

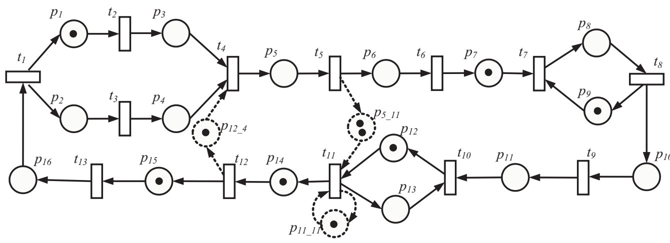


Fig. 1. Marked Graph and Marking Structurally Implicit Places, with initial marking $m_0(p_1) = m_0(p_7) = m_0(p_9) = m_0(p_{12}) = m_0(p_{14}) = m_0(p_{15}) = 1$.

***Definition** 2.3:* [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 $\boldsymbol{y} \ge 0$, such that $\boldsymbol{C}(p, T) = \boldsymbol{y} \cdot \boldsymbol{C}(P, T)$.

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

Given a marking structurally implicit place p, the minimal initial marking to make p implicit is [6]:

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

***Example** 2.4:* Fig. 1 shows a MG. It is easy to observe that from t_{12} to t_4 there exist two simple paths, $\pi_1 = \{t_{12}p_{15}t_{13}p_{16}t_1p_1t_2p_3t_4\}$ and $\pi_2 = \{t_{12}p_{15}t_{13}p_{16}t_1p_2t_3p_4t_4\}$. 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 the 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 decentralized 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 *ContMG*, the large scale system is decomposed into subsystems that are modeled with *ContMGs* 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 an initial marking $\boldsymbol{m}_0$ to a final marking $\boldsymbol{m}_f$.

IV. Structural Decomposition of Marked Graphs

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

***Definition** 4.1:* Let $\mathcal{N} = \langle P \cup B, T, \boldsymbol{Pre}, \boldsymbol{Post}\rangle$ be a strongly connected MG, B 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) $P_1 \cup P_2 = P$, $P_1 \cap P_2 = \emptyset$

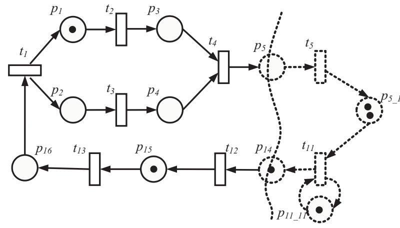
(ii) $P_1 \cup B = {}^\bullet T_1 \cup T_1{}^\bullet$, $P_2 \cup B = {}^\bullet T_2 \cup T_2{}^\bullet$

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

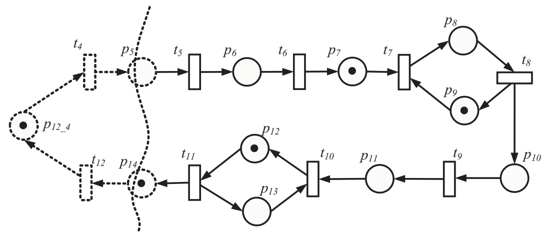
***Example** 4.2:* The non-dotted part in Fig. 2 are the subnets $\mathcal{N}_1$, $\mathcal{N}_2$ obtained from the MG in Fig. 1, which is cut by $B = \{p_5, p_{14}\}$, with the interface $U = \{t_4, t_5, t_{11}, t_{12}\}$ while $U_1 = \{t_4, t_{12}\}$, $U_2 = \{t_5, t_{11}\}$.

After cutting, the two subsystems $\mathcal{N}_1$ and $\mathcal{N}_2$ 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** 4.3:* Let $\mathcal{N} = \langle P \cup B, T, \boldsymbol{Pre}, \boldsymbol{Post}\rangle$ be a strongly connected MG, $\mathcal{N}_i = \langle P_i, T_i, \boldsymbol{Pre}_i, \boldsymbol{Post}_i\rangle$ be the subnets associated with a cut B. The *complemented subnet* $\mathcal{CN}_i$ is obtained from $\mathcal{N}_i$ by copying B and U_j, 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 $B = \{p_5, p_{14}\}$, $U_2 = \{t_5, t_{11}\}$ and adding $IP_1 = \{p_{5\_11}, p_{11\_11}\}$ to $\mathcal{N}_1$, while $\mathcal{CN}_2$ is obtained after copying $B = \{p_5, p_{14}\}$, $U_1 = \{t_4, t_{12}\}$ and adding $IP_2 = \{p_{12\_4}\}$ to $\mathcal{N}_2$. Notice that cut B and interface U are shared in both subnets.

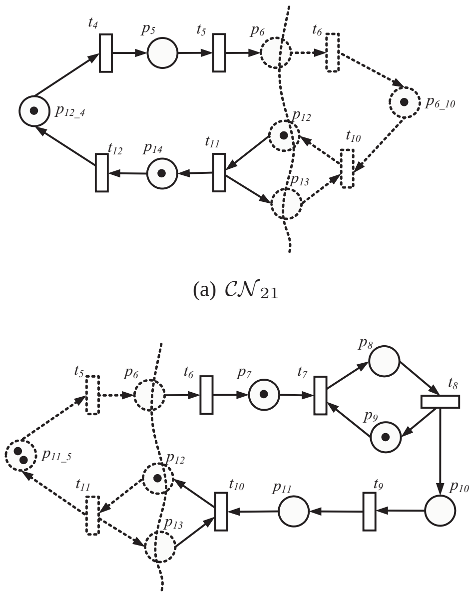
In order to calculate the initial marking of p_{e_s} that makes it implicit, we have to find out the path from t_e to t_s such that (3) is satisfied. There are some efficient algorithms which can be used, e.g., the algorithm of Floyd-Warshall [1].

Sometimes for a complex system, only one cut is not sufficient, because the complemented subsets are still difficult to handle. Therefore, the above decomposition process can 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. Decentralized Control of Large Scale Systems

The decentralized structure of a large scale system may be obtained using the decomposition method presented in section IV. In this section it shown that the *ON-OFF* controller 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 written as: $\vec{\boldsymbol{\sigma}} = \vec{\boldsymbol{\varsigma}} + k \cdot \boldsymbol{x}$, where $k > 0$, $\vec{\boldsymbol{\varsigma}}$ is a firing count vector driving the system to $\boldsymbol{m}_f$, and $\boldsymbol{x}$ is a T-*semiflow*. An *ON-OFF* controller for structurally persistent $ContPN$ is proposed in [19]: if $\vec{\boldsymbol{\sigma}}$ is



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

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

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

minimal, for any t_j, simply let it *ON* before the cumulative flow of t_j reaches $\vec{\sigma}(t_j)$, and after that let it *OFF*. $\boldsymbol{m}_f$ is reached in minimum-time using this strategy.

Because MGs is a subclass of structurally persistent nets, this *ON-OFF* strategy can be applied. In the following, it is shown how to compute the control law (minimal firing count vector) in a decentralized 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 $p \in P_i \cup B$, $m_0^i(p) = m_0(p)$, 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 $p \in P_i \cup B$, $m_f^i(p) = m_f(p)$. Every place $p \in IP_i$ belongs to different circuit in $\mathcal{CN}_i$, and since $\mathcal{CN}_i$ is a strongly connected MG, each circuit forms a P-*semiflow* [16], $m_f^i(p)$ can be easily computed.
(3) $\vec{\boldsymbol{\sigma}}$: the minimal firing count vector driving $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ to $\boldsymbol{m}_f$.
(4) $\vec{\boldsymbol{\sigma}}_{prj}^i$: the firing count vector of $\mathcal{CN}_i$ directly projected from $\vec{\boldsymbol{\sigma}}$. For every $t \in T_i$, $\vec{\sigma}_{prj}^i(t) = \vec{\sigma}(t)$.
(5) $\vec{\boldsymbol{\sigma}}_{min}^i$: the minimal firing count vector driving $\langle \mathcal{CN}_i, \boldsymbol{m}_0^i \rangle$ to $\boldsymbol{m}_f^i$.

A. Decomposition with One Cut

The most interesting point of the decomposition approach in section IV is: if proper initial markings are put into the added marking structurally implicit places to make them implicit, the projections of reachable markings and firing sequences of the original system are preserved in the complemented subnets [6], i.e., $\vec{\boldsymbol{\sigma}}_{prj}^i$ can always be fired in $\mathcal{CN}_i$ with initial marking $\boldsymbol{m}_0^i$, 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:* Let $\vec{\boldsymbol{\sigma}}^i$ be firing count vectors driving $\langle \mathcal{CN}_i, \boldsymbol{m}_0^i \rangle$ to $\boldsymbol{m}_f^i$, $i = 1, 2$. $\vec{\boldsymbol{\sigma}}^1$ 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. The *merge* of them is defined as: $\vec{\boldsymbol{\sigma}}^{12} = \vec{\boldsymbol{\sigma}}^1 \oplus \vec{\boldsymbol{\sigma}}^2$, such that $\forall t \in T_i$, $\vec{\sigma}^{12}(t) = \vec{\sigma}^i(t)$, $i = 1, 2$.

Example *5.3:* Let us consider the MG 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 $m_f^2(p_{12\_4}) = 1.5$. Observe that $\vec{\boldsymbol{\sigma}}$ is the minimal firing count vector driving $\mathcal{N}$ from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$, i.e., $\boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \vec{\boldsymbol{\sigma}}$. Its projection in $\mathcal{CN}_1$, $\vec{\boldsymbol{\sigma}}_{prj}^1 = [1.7\ 2.3\ 1.5\ 1.1\ 0.7\ 0.6\ 1.6\ 2.1]^T$ is fireable and drives $\mathcal{CN}_i$ from $\boldsymbol{m}_0^1$ to $\boldsymbol{m}_f^1$, but it is not minimal. The minimal firing count vector in this case is $\vec{\boldsymbol{\sigma}}_{min}^1 = [1.1\ 1.7\ 0.9\ 0.5\ 0.1\ 0\ 1\ 1.5]^T$. Fortunately, the projection can be obtained from the minimal one by adding 0.6 times the T-*semiflow* of $\mathcal{CN}_1$, i.e., $\vec{\boldsymbol{\sigma}}_{prj}^1 = \vec{\boldsymbol{\sigma}}_{min}^1 + 0.6 \cdot \boldsymbol{1}$.

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	$\vec{\boldsymbol{\sigma}}$	$\vec{\boldsymbol{\sigma}}_{prj}^1$ ($\vec{\boldsymbol{\sigma}}_{min}^1$)	$\vec{\boldsymbol{\sigma}}_{prj}^2$ ($\vec{\boldsymbol{\sigma}}_{min}^2$)
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 and the previous result holds for untimed system. If all transitions are controllable, a marking $\boldsymbol{m}$ is reachable in the untimed model, if it is reachable in the timed one; while if a marking $\boldsymbol{m}$ is reachable in the untimed model, then it is asymptotically reachable in the timed one [13]. Therefore, similar results can be easily extended to $ContMG$. In particular, the projections of firing count vectors and reachable markings of the original system are preserved in the complemented subnets. In the sequel, we assume the system is live.

If the minimal firing count vectors of $\mathcal{CN}_1$ and $\mathcal{CN}_2$ are compatible, the merged vector is firable in $\mathcal{N}$. In the case they are not compatible, like $\vec{\boldsymbol{\sigma}}_{min}^1$ and $\vec{\boldsymbol{\sigma}}_{min}^2$ in Ex. 5.3 (because $\exists t \in U, \vec{\sigma}_{min}^1(t) \neq \vec{\sigma}_{min}^2(t)$), a T-*semiflow* can be added to one of them to make them compatible. Finally, the merged vector obtained is actually equal to $\vec{\boldsymbol{\sigma}}$.

Proposition *5.4:* Let $\langle\mathcal{N},\boldsymbol{m}_0\rangle$ be a live MG, with cut B and corresponding interface U. $\vec{\boldsymbol{\sigma}}$ is the minimal firing count vector driving $\langle\mathcal{N},\boldsymbol{m}_0\rangle$ to $\boldsymbol{m}_f$, while $\vec{\boldsymbol{\sigma}}^i_{min}$ is the minimal firing count vector driving $\langle\mathcal{CN}_i,\boldsymbol{m}^i_0\rangle$ to $\boldsymbol{m}^i_f$. There exists $k\geq 0$, such that $\vec{\boldsymbol{\sigma}}=(\vec{\boldsymbol{\sigma}}^i_{min}+k\cdot\mathbf{1})\oplus\vec{\boldsymbol{\sigma}}^j_{min}$, $i,j=1,2,i\neq j$.

Proof: We will first prove that there exists a $k\geq 0$ to make $(\vec{\boldsymbol{\sigma}}^i_{min}+k\cdot\mathbf{1})$ and $\vec{\boldsymbol{\sigma}}^j_{min}$ compatible, then by merging them, $\vec{\boldsymbol{\sigma}}$ is obtained.

Since the projection $\vec{\boldsymbol{\sigma}}^i_{prj}$ is a fireable vector in $\langle\mathcal{CN}_i,\boldsymbol{m}^i_0\rangle$ and $\boldsymbol{m}^i_f$ is reached, while $\vec{\boldsymbol{\sigma}}^i_{min}$ is the minimal firing count vector driving the system to $\boldsymbol{m}^i_f$, we have:

$$\vec{\boldsymbol{\sigma}}^i_{prj}=\vec{\boldsymbol{\sigma}}^i_{min}+\alpha_i\cdot\mathbf{1}, i=1,2,\alpha_i\geq 0$$

Without loss of generality, assume $\alpha_1\leq\alpha_2$. If only considering the common transitions in U:

$$\vec{\sigma}^i_{prj}(t)=\vec{\sigma}^i_{min}(t)+\alpha_i,\forall t\in U, i=1,2$$

Because $\vec{\sigma}^1_{prj}(t)=\vec{\sigma}^2_{prj}(t)=\vec{\sigma}(t),\forall t\in U$, we have

$$\vec{\sigma}^1_{min}(t)+\alpha_1=\vec{\sigma}^2_{min}(t)+\alpha_2,\forall t\in U$$

Therefore,

$$\vec{\sigma}^1_{min}(t)-\vec{\sigma}^2_{min}(t)=\alpha_2-\alpha_1=k\geq 0,\forall t\in U$$

Hence $(\vec{\boldsymbol{\sigma}}^2_{min}+k\cdot\mathbf{1})$ and $\vec{\boldsymbol{\sigma}}^1_{min}$ are compatible, and 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,

$$\boldsymbol{m}_0+\boldsymbol{C}\cdot(\vec{\boldsymbol{\sigma}}^1_{min}\oplus(\vec{\boldsymbol{\sigma}}^2_{min}+k\cdot\mathbf{1}))=\boldsymbol{m}_f$$

Because the system is a live MG, there always exists a sequence ς that can be fired, and its count vector $\vec{\boldsymbol{\varsigma}}$ is equal to $\vec{\boldsymbol{\sigma}}^1_{min}\oplus(\vec{\boldsymbol{\sigma}}^2_{min}+k\cdot\mathbf{1})$.

Since $\vec{\boldsymbol{\sigma}}^1_{min}$, $\vec{\boldsymbol{\sigma}}^2_{min}$ are minimal, $\vec{\boldsymbol{\varsigma}}$ is also minimal. Because the minimal firing count vector is unique in a live MG [19], $\vec{\boldsymbol{\varsigma}}=\vec{\boldsymbol{\sigma}}^1_{min}\oplus(\vec{\boldsymbol{\sigma}}^2_{min}+k\cdot\mathbf{1})=\vec{\boldsymbol{\sigma}}$. ∎

Example *5.5:* In Ex. 5.3, $\vec{\boldsymbol{\sigma}}^1_{min}$ and $\vec{\boldsymbol{\sigma}}^2_{min}$ are not compatible. Observe that $\forall t\in U,\vec{\sigma}^2_{min}(t)-\vec{\sigma}^1_{min}(t)=0.6$. Clearly, after adding $0.6\cdot\mathbf{1}$ to $\vec{\boldsymbol{\sigma}}^1_{min}$, they are compatible and can be merged: $\vec{\boldsymbol{\sigma}}=(\vec{\boldsymbol{\sigma}}^1_{min}+0.6\cdot\mathbf{1})\oplus\vec{\boldsymbol{\sigma}}^2_{min}$.

Notice that, $\vec{\boldsymbol{\sigma}}^1_{min}$ is different from the direct projection from $\vec{\boldsymbol{\sigma}}$, while $\vec{\boldsymbol{\sigma}}^2_{min}$ is equal to $\vec{\boldsymbol{\sigma}}^2_{prj}$. In fact, if $\vec{\boldsymbol{\sigma}}=(\vec{\boldsymbol{\sigma}}^j_{min}+k\cdot\mathbf{1})\oplus\vec{\boldsymbol{\sigma}}^i_{min}$, then $\forall t\in T_i,\vec{\boldsymbol{\sigma}}(t)=\vec{\boldsymbol{\sigma}}^i_{min}(t)$, therefore the minimal firing count count vector and the projection are equal, i.e., $\vec{\boldsymbol{\sigma}}^i_{min}=\vec{\boldsymbol{\sigma}}^i_{prj}$.

Definition *5.6:* A complemented subnet $\mathcal{CN}_i$ is said to be *critical*, if $\vec{\boldsymbol{\sigma}}^i_{min}=\vec{\boldsymbol{\sigma}}^i_{prj}$, i.e., $\forall t\in T_i,\vec{\boldsymbol{\sigma}}(t)=\vec{\boldsymbol{\sigma}}^i_{min}(t)$.

B. Decomposition with Hierarchical Cut

Let us now assume that $\mathcal{N}$ has been cut into two $\mathcal{CN}_1$ and $\mathcal{CN}_2$ with $\vec{\boldsymbol{\sigma}}=(\vec{\boldsymbol{\sigma}}^1_{min}+k_1\cdot\mathbf{1})\oplus\vec{\boldsymbol{\sigma}}^2_{min}$, then $\vec{\boldsymbol{\sigma}}^2_{min}=\vec{\boldsymbol{\sigma}}^2_{prj}$, $\mathcal{CN}_2$ is critical. If $\mathcal{CN}_2$ is cut one more time into $\mathcal{CN}_{21}$, $\mathcal{CN}_{22}$ and suppose $\vec{\boldsymbol{\sigma}}^2_{min}=(\vec{\boldsymbol{\sigma}}^{21}_{min}+k_2\cdot\mathbf{1})\oplus\vec{\boldsymbol{\sigma}}^{22}_{min}$, then obviously $\mathcal{CN}_{22}$ is critical. The same result can be obtained when $\mathcal{CN}_{22}$ is cut again: there exists at least one subnet $\mathcal{CN}_i$ that is *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 *5.7:* Let $\langle\mathcal{N},\boldsymbol{m}_0\rangle$ be a live MG that is decomposed into n subnets. Assuming $\mathcal{CN}_q$, $1\leq q\leq n$ is a critical complemented subnet, then there exist $\alpha_i, i=1,2,...,n$ such that:

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

where $\alpha_i\geq 0$, $\alpha_q=0$.

Proof: Since all complemented subnets are live MGs, $\vec{\boldsymbol{\sigma}}^i_{min}+\alpha_i\cdot\mathbf{1}$ is also fireable in $\mathcal{CN}_i$. For any two neighbor subnets $\mathcal{CN}_i$ and $\mathcal{CN}_j$, $\alpha_i,\alpha_j\geq 0$ can be found such that $\vec{\boldsymbol{\sigma}}^i_{min}+\alpha_i\cdot\mathbf{1}$ and $\vec{\boldsymbol{\sigma}}^j_{min}+\alpha_j\cdot\mathbf{1}$ become compatible and can be merged. According to Proposition 5.4, after merging all the firing count vectors, $\vec{\boldsymbol{\sigma}}'=\bigoplus_{i=1}^{n}(\vec{\boldsymbol{\sigma}}^i_{min}+\alpha_i\cdot\mathbf{1})$ is obtained, which can be fired in $\langle\mathcal{N},\boldsymbol{m}_0\rangle$, reaching $\boldsymbol{m}_f$.

If all $\alpha_i>0$, $\vec{\boldsymbol{\sigma}}'$ is not the minimal firing count vector, then certain amount of T-*semiflow* can be subtracted from $\vec{\boldsymbol{\sigma}}'$ until $\vec{\boldsymbol{\sigma}}=\vec{\boldsymbol{\sigma}}'$. ∎

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

Example *5.8:* Let us examine the MG 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 there are presented the corresponding minimal firing count vectors. It is obtained:

$$\vec{\boldsymbol{\sigma}}=(\vec{\boldsymbol{\sigma}}^1_{min}+0.6\cdot\mathbf{1})\oplus\vec{\boldsymbol{\sigma}}^2_{min}\oplus\vec{\boldsymbol{\sigma}}^3_{min}$$

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

The rest of this section devotes to design an effective algorithm to search a critical subnet, and compute corresponding α_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 in V represents a subnet, there are arcs between nodes 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 given by $w(v_i,v_j)=\vec{\sigma}^i_{min}(t)-\vec{\sigma}^j_{min}(t),\forall t\in U$, negative weight is also allowed here. So in the corresponding graph G (Fig. 4), $w(v_2,v_1)=0.6$, $w(v_1,v_2)=-0.6$, while $w(v_2,v_3)=w(v_3,v_2)=0$. Let us denote by $W(v_i,v_j)$ the sum of the weights on the simple path from v_i to v_j.

TABLE II

MINIMAL FIRING COUNT VECTORS

T	$\vec{\boldsymbol{\sigma}}(\mathcal{N})$	$\vec{\boldsymbol{\sigma}}^1_{min}(\mathcal{CN}_1)$	$\vec{\boldsymbol{\sigma}}^2_{min}(\mathcal{CN}_{21})$	$\vec{\boldsymbol{\sigma}}^3_{min}(\mathcal{CN}_{22})$
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		

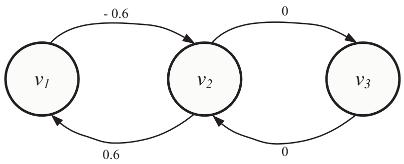


Fig. 4. The graph $G = \langle V, W\rangle$ constructed from the three complemented subnets in Ex. 5.8

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

It can be observed that, the sum of weights in the path from v_i to v_j reflects the *relative difference* of $\vec{\boldsymbol{\sigma}}^i_{min}$ to $\vec{\boldsymbol{\sigma}}^j_{min}$. In Ex. 5.8, 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 $W(v_3, v_1) = w(v_3, v_2) + w(v_2, v_1) = 0.6$. Obviously we have $\vec{\boldsymbol{\sigma}} = (\vec{\boldsymbol{\sigma}}^1_{min} + 0.6 \cdot \mathbf{1}) \oplus (\vec{\boldsymbol{\sigma}}^2_{min} + 0 \cdot \mathbf{1}) \oplus \vec{\boldsymbol{\sigma}}^3_{min}$. Actually, the non-negative value α_i in (4) is equal to $W(v_q, v_i)$.

***Property** 5.9:* Let $v_q \in V$. 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 $\alpha_i = W(v_q, v_i)$, $\vec{\boldsymbol{\sigma}} = \oplus_{i=1}^n (\vec{\boldsymbol{\sigma}}^i_{min} + W(v_q, v_i) \cdot \mathbf{1})$. Since $\alpha_q = W(v_q, v_q) = 0$, $\vec{\boldsymbol{\sigma}}^q_{min} = \vec{\boldsymbol{\sigma}}^q_{prj}$. Therefore $\mathcal{CN}_q$ is critical. ∎

Algorithm 1 searches a critical subnet based on the graph G. First, all nodes are labeled as *new*. Then for each node v_i labeled as *new*, the relative differences from v_i to all others nodes v_j, $W(v_i, v_j)$ is computed. If one is found to be negative then v_i is not critical and it is labeled as *old*. If it is positive then v_j is not critical because $W(v_j, v_i)$ must be negative, and v_j is labeled as *old*. When a node with all relative differences non-negative is found, or there is only one node labeled as *new* is left, the program finishes. When computing the sum of weights, of course the intermediate values that have been computed before should be reused. In the worst case, the 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 $v_q \in V$

Label all the nodes in V as *new*;
while more than one node in V is labeled as *new* **do**
  Choose a node v_i from V which is labeled as *new*;
  for $j = 1$ to n **do**
    if $W(j, i)$ has not been computed **then**
      compute $W(i, j)$;
      if $W(i, j) > 0$ **then**
        label v_j as *old*;
      else if $W(i, j) < 0$ **then**
        label v_i as *old*;
        break;
      end if
    end if
  end for
  if $j = n$ and v_i is labeled as *new* **then**
    return v_i;
  end if
end while
return The last node in V that is labeled as *new*

C. Control Structures

There are two kinds of controllers in the decentralized control system: local controllers and a supervisory controller.

Local controllers know only the structures of the local subsystem. The local control law (minimal firing count vector) $\vec{\boldsymbol{\sigma}}^i_{min}$ of subsystem $\mathcal{CN}_i$ is first computed independently in the corresponding controller. Since this control law may be not globally applicable, its value is sent to the supervisory controller. After the updating information α_i is received from the supervisory controller, the controller of $\mathcal{CN}_i$ can be implemented independently with the control law $\vec{\boldsymbol{\sigma}}^i_{min} + \alpha_i \cdot \mathbf{1}$.

The supervisory controller is mainly used to update the locally computed control laws in order to make them globally admissible. Based on the local control laws of subsystems, graph G is first constructed, and Algorithm 1 is applied to find a critical subnets $\mathcal{CN}_q$. Then the relative difference α_i is computed and sent to $\mathcal{CN}_i$. Let us observe that the only information required by the supervisory controller are the local control laws, therefore all computations are done locally, so the communication cost is very low.

Algorithm 2, 3 are used by supervisory controller, local controller respectively.

VI. CASE STUDY

Let us consider the $ContMG$ system in Fig. 5 which models a manufacturing system with three types of product lines which are assembled for one final product. The system

Algorithm 2 Supervisory Controller

Input: $\vec{\boldsymbol{\sigma}}^i_{min}$

Output: $\boldsymbol{\alpha}$

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

Algorithm 3 Local Controller i

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

Output: $\vec{\boldsymbol{\sigma}}^i_{min}$

1: Compute $\vec{\boldsymbol{\sigma}}^i_{min}$ driving the system to $\boldsymbol{m}^i_f$;
2: Send $\vec{\boldsymbol{\sigma}}^i_{min}$ to the supervisory controller;
3: Receive α_i from the supervisory controller;
4: Update $\vec{\boldsymbol{\sigma}}^i_{min} \leftarrow \vec{\boldsymbol{\sigma}}^i_{min} + \alpha_i \cdot \mathbf{1}$;
5: Apply *ON-OFF* control;

is cut into four subsystems through buffers ($B_1 = \{p_1, p_{12}\}$, $B_2 = \{p_{13}, p_{23}\}$, $B_3 = \{p_{24}, p_{38}\}$) 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.

Assuming the initial and desired final marking are listed in Table III. The corresponding minimal firing count vectors are easily computed, and the result is 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)

Graph G (presented in 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 Algorithm 1 is applied, $\mathcal{CN}_4$ is found as the critical subnet.

The relative differences of $\mathcal{CN}_4$ to all the other subnets can be computed, which in this case is very straightforward: $\alpha_1 = \alpha_2 = 2.8$, $\alpha_3 = 2.2$. Therefore, the minimal firing count vector is generated as: $\vec{\boldsymbol{\sigma}} = (\vec{\boldsymbol{\sigma}}^1_{min} + 2.8 \cdot \mathbf{1}) \oplus (\vec{\boldsymbol{\sigma}}^2_{min} + 2.8 \cdot \mathbf{1}) \oplus (\vec{\boldsymbol{\sigma}}^3_{min} + 2.2 \cdot \mathbf{1}) \oplus \vec{\boldsymbol{\sigma}}^4_{min}$.

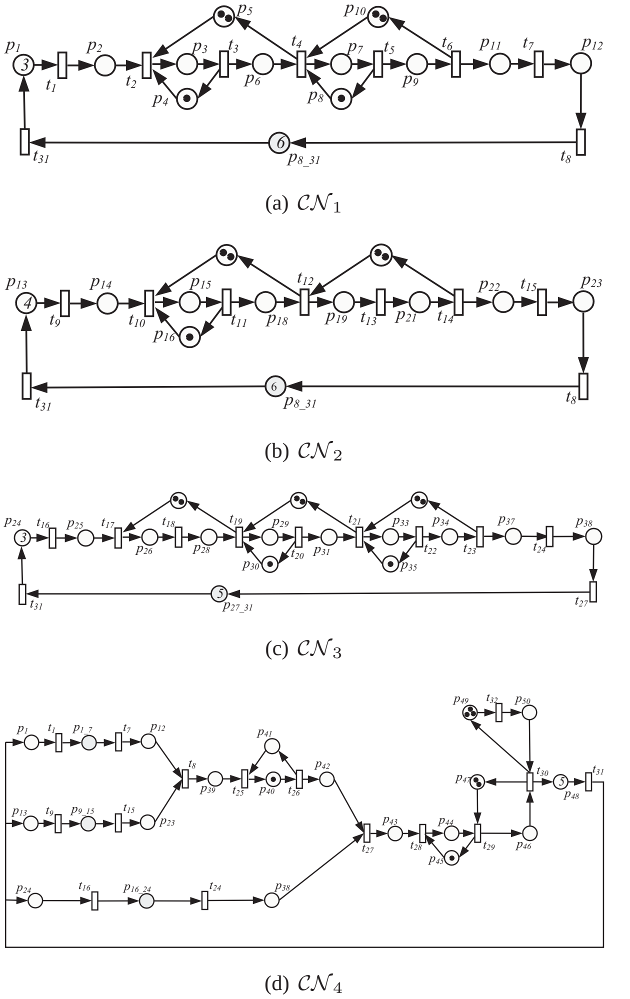


Fig. 6. Complemented subnets from the system model in Fig. 5, with cut $B_1 = \{p_1, p_{12}\}$, $B_2 = \{p_{13}, p_{23}\}$, $B_3 = \{p_{24}, p_{38}\}$

Finally, the local controllers can apply their control lows using an *ON-OFF* strategy. The global final marking is reached in 13.24 time units, that is the minimum time [19].

VII. CONCLUSIONS

Decentralized control could be a solution of controlling systems that are too complex to be handled with centralized controllers. This work focuses on decentralized control of large scale systems that are modeled with timed continuous MGs, aiming to drive the system from an initial marking to a desired final marking. The model is first decomposed into subnets with sets of places, then control laws are computed in a decentralized way. A supervisory control is introduced to make the locally computed laws globally applicable. After that, an *ON-OFF* strategy can be applied in each subnet, and final marking is reached in minimum time.

As a future work, we plan to investigate the possibility of developing an automatic system cutting procedure and applying this control method to more general nets structures.

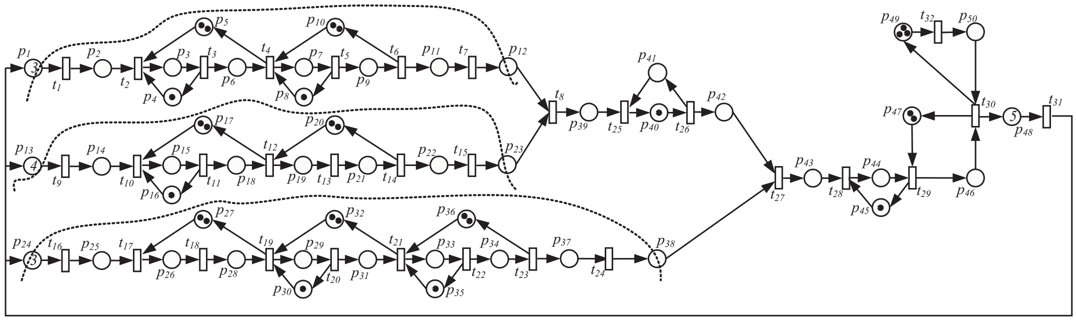


Fig. 5. A manufacturing system model

TABLE IV

MINIMAL FIRING COUNT VECTORS

$\mathcal{CN}_1$		$\mathcal{CN}_2$		$\mathcal{CN}_3$		$\mathcal{CN}_4$	
T	$\vec{\sigma}^1_{min}$	T	$\vec{\sigma}^2_{min}$	T	$\vec{\sigma}^3_{min}$	T	$\vec{\sigma}^4_{min}$
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

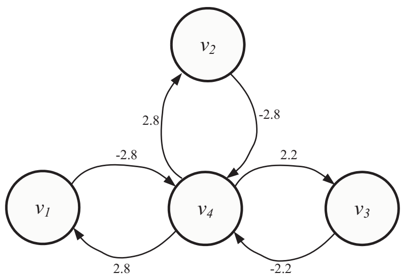


Fig. 7. The graph constructed from Table. IV

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.

[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. *IEEE Transactions on Software Engineering*, 20(7):526–535, 1994.

[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] M. V. Iordache and P. J. Antsaklis. Decentralized control of Petri nets with constraint transformations. In *Proceedings of the American Control Conference*, pages 314 – 319, 2003.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

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

[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.