Supervisory Control of Petri Nets with Decentralized Monitor Places

Francesco Basile, Alessandro Giua, Carla Seatzu

Abstract

In this paper we consider the problem of determining a set of decentralized controllers for place/transition nets to enforce a global specification on the net behavior. In particular, we assume that both the global specification and the decentralized specifications are given in terms of Generalized Mutual Exclusion Constraints (GMECs). An algorithm is given under appropriate assumptions, namely the weights of the GMECs are positive, the transitions are controllable and observable, and the support of each decentralized GMEC is a singleton. Even if such assumptions strongly limit the application of the solution to real cases, the proposed results constitute a preliminary step towards a synthesis procedure that optimizes the permissiveness of the closed loop behavior under decentralized control.

Published as: F. Basile, A. Giua, C. Seatzu, "Supervisory Control of Petri Nets with Decentralized Monitor Places," 26^{th} American Control Conference (ACC'07), New York, pp. 4957–4962, July 2007.

Francesco Basile is with Dip. Ing. dell'Informazione e Ing. Elettrica, Università di Salerno, Via Ponte don Melillo, 84084 Fisciano (Salerno), Italy. fbasile@unisa.it

Alessandro Giua, Carla Seatzu are with the Dip. di Ing. Elettrica ed Elettronica, Università di Cagliari, Piazza d'Armi, 09123 Cagliari, Italy. {giua,seatzu}@diee.unica.it

I. INTRODUCTION

Decentralized control has received a great attention in the Discrete Event Systems (DES) area in the last decade [10]. Usually, decentralized control problems in the context of supervisory control have been studied by means of formal languages approaches using automata [1], [8], [11], [12]. On the contrary, Petri Nets (PNs) have not received much attention although the compact representation which they offer may potentially help in reducing the complexity of decentralized supervisory control problems. A state predicates formulation for decentralized control has been adopted in few works. In [6] global specifications are implemented by local supervisors with communication. In [4] a central coordinator is also present but specifications are assumed to be given from the beginning in distributed form. In [7] global specifications without central coordination are considered and a sufficient condition is given for a state predicate formulated in terms of GMECs to be enforced in a decentralized setting (d-admissibility); in addition, the transformation of inadmissible decentralized constraints into admissible ones is posed either in terms of the minimization of communication costs or in terms of the transformation of the constraints into a set of more restrictive ones but d-admissible.

In this paper the attention is focused on global state specifications given in terms of GMECs and on a control architecture without central coordinator and communication between local supervisors. This choice is motivated by the following considerations. (i) It is not always possible to have communication with all plant sensors or actuators because of economic reasons or bandwidth limitations. This problem is particularly relevant for plants having a wide geographic extension or a large number of devices such as in modern communication systems. (ii) Even if centralized control is possible, the communication with a certain area of the plant can be lost. It could be useful to use a decentralized control without communication for this area until communication comes back.

In [2] we investigated the problem of determining a set of decentralized GMECs $(\boldsymbol{w}^{(i)}, k^{(i)})$ that are able to impose a specification on the net behavior given in terms of a *single* global GMEC $(\boldsymbol{w}, k)$. The set of places was partitioned in ν subsets P_i, $i = 1, \ldots, \nu$, and the support of $(\boldsymbol{w}^{(i)}, k^{(i)})$ was P_i. In particular we assumed that: (A1) $\boldsymbol{w} \geq \mathbf{0}$; (A2) all transitions are controllable and observable; (A3) $P_i = \{p_i\}$; (A4) the vectors $\boldsymbol{w}^{(i)}$'s are taken equal to the projection of $\boldsymbol{w}$ on P_i. On the basis of geometrical considerations we suggested a procedure to compute $k^{(i)}$'s as the solution of an integer programming problem.

In this paper we consider a more general problem. We assume that the global specification is given by a *set* of GMECs $(\boldsymbol{W}, \boldsymbol{k})$, where $\boldsymbol{W} = [\boldsymbol{w}_1^T, \boldsymbol{w}_2^T, ...\boldsymbol{w}_{n_c}^T]^T$ and $\boldsymbol{k} = [k_1, k_2, ...k_{n_c}]^T$.

We show that, under assumptions (A1) to (A3) (obviously assumption (A4) is meaningless now), this problem can be solved by computing an *integer inner box* $\mathcal{I}(\boldsymbol{u}) = \{\boldsymbol{m} \in \mathbb{N}^m \mid \boldsymbol{m} \leq \boldsymbol{u}\}$ included in the set of legal markings defined by the global GMEC $\mathcal{M}(\boldsymbol{W}, \boldsymbol{k}) = \{\boldsymbol{m} \in \mathbb{N}^m \mid \boldsymbol{W} \cdot \boldsymbol{m} \leq \boldsymbol{k}\}$. In particular, we consider the problem of finding a *maximal integer inner box* $\mathcal{I} \subseteq \mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$, i.e. an inner box such that there does not exist a box $\tilde{\mathcal{I}} \neq \mathcal{I}$ and $\mathcal{I} \subsetneq \tilde{\mathcal{I}} \subseteq \mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$. A solution that aims to guarantee fairness among places is proposed, that can be computed using a simple iterative algorithm.

Note that this approach constitutes a preliminary step towards the solution of a more important problem, namely that of determining the decentralized GMECs whose set of legal markings under decentralized control has maximal permissiveness. This problem requires in general a behavioral approach, since it depends on the net initial marking, and thus it is very computationally demanding.

We finally remark that, a similar problem has been recently investigated by Iordache and Antsaklis in [7]. The problem considered in [7] is set in a more rich setting that assumes the exchange of communications among decentralized controllers. In this framework a meaningful problem addressed by the authors is that of giving a sufficient condition under which decentralized supervisors result to be as much permissive as the centralized one.

The setting we consider in this paper assumes that no communication is possible. In such a case, the permissiveness of the centralized monitor can (almost) never be achieved, and in general the admissible solutions to a control problem become much more restrictive. This is why optimizing the permissiveness of the distributed controllers is a key issue in this framework.

This also makes the control problem much harder to solve and this is why in this preliminary paper we focus on a particular case that, unlike [7], assumes several restrictions: (i) each local agent can only observe and control a single place; (ii) all transitions are assumed to be observable and controllable; (iii) we address the simpler issue of determining a maximal inner box, that has not necessarily maximal cardinality, nor does it necessarily lead to a maximally permissive controller. Future research will address more general cases.

II. BASIC DEFINITIONS

A. Petri nets

In this section we recall the formalism used in the paper. For more details on Petri nets we address to [9].

A Place/Transition (P/T) net is a structure $N = (P, T, \boldsymbol{Pre}, \boldsymbol{Post})$ where: P is a set of m *places* represented by circles; T is a set of n *transitions* represented by bars; $P \cap T = \emptyset$, $P \cup T \neq \emptyset$; $\boldsymbol{Pre}$ ($\boldsymbol{Post}$) is the $m \times n$ sized, natural valued, pre-(post-)incidence matrix. For instance, $\boldsymbol{Pre}(p,t) = w$ (resp., $\boldsymbol{Post}(p,t) = w$) means that there is an arc from p to t (resp., from t to p) with weight w. The incidence matrix $\boldsymbol{C}$ of the net is defined as $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$. A *marking* is a $m \times 1$ vector $\boldsymbol{m} : P \to \mathbb{N}$ that assigns to each place of a P/T net a non-negative integer number of tokens. A P/T system or net system $\langle N, \boldsymbol{m}_0 \rangle$ is a P/T net N with an initial marking $\boldsymbol{m}_0$. A transition $t \in T$ is enabled at a marking $\boldsymbol{m}$ iff $\boldsymbol{m} \geq \boldsymbol{Pre}(\cdot, t)$. If t is enabled, then it may fire yielding a new marking $\boldsymbol{m}' = \boldsymbol{m} + \boldsymbol{Post}(\cdot, t) - \boldsymbol{Pre}(\cdot, t) = \boldsymbol{m} + \boldsymbol{C}(\cdot, t)$. The notation $\boldsymbol{m}[t\rangle\boldsymbol{m}'$ means that an enabled transition t may fire at $\boldsymbol{m}$ yielding $\boldsymbol{m}'$. A *firing sequence* from $\boldsymbol{m}_0$ is a (possibly empty) sequence of transitions $\sigma = t_1, \ldots, t_k$ such that $\boldsymbol{m}_0[t_1\rangle\boldsymbol{m}_1[t_2\rangle\boldsymbol{m}_2 \ldots [t_k\rangle\boldsymbol{m}_k$. A marking $\boldsymbol{m}$ is reachable in $\langle N, \boldsymbol{m}_0 \rangle$ iff there exists a firing sequence σ such that $\boldsymbol{m}_0[\sigma\rangle\boldsymbol{m}$. Given a net system $\langle N, \boldsymbol{m}_0 \rangle$ the set of reachable markings is denoted $R(N, \boldsymbol{m}_0)$.

B. Generalized Mutual Exclusion Constraint

A *Generalized Mutual Exclusion Constraint* (GMEC) is a couple $(\boldsymbol{w}, k)$ where $\boldsymbol{w} : P \to \mathbb{Z}$ is an m dimensional row vector and $k \in \mathbb{Z}$. A GMEC defines a set of *legal markings*:

$$\mathcal{M}(\boldsymbol{w}, k) = \{\boldsymbol{m} \in \mathbb{N}^m \mid \boldsymbol{w} \cdot \boldsymbol{m} \leq k\}.$$

The markings that are not legal are called *forbidden markings*. A controlling agent, called *supervisor*, must ensure the forbidden markings will be not reached. So the set of legal markings under control is $\mathcal{M}_c(\boldsymbol{w}, k) = \mathcal{M}(\boldsymbol{w}, k) \cap R(N, \boldsymbol{m}_0)$. We call *support* of $(\boldsymbol{w}, k)$ the set $Q_{\boldsymbol{w}} = \{p \in P | \boldsymbol{w}(p) \neq 0\}$.

A set of GMECs $(\boldsymbol{W}, \boldsymbol{k})$, with

$$\boldsymbol{W} = [\boldsymbol{w}_1^T, \boldsymbol{w}_2^T, \ldots \boldsymbol{w}_{n_c}^T]^T, \quad \text{and} \quad \boldsymbol{k} = [k_1, k_2, \ldots k_{n_c}]^T,$$

defines the set of legal markings $\mathcal{M}(\boldsymbol{W},\boldsymbol{k}) = \{\boldsymbol{m} \in \mathbb{N}^m \mid \boldsymbol{W}\cdot\boldsymbol{m} \leq \boldsymbol{k}\}$. We call *support* of $(\boldsymbol{W},\boldsymbol{k})$ the set $Q_{\boldsymbol{W}} = \{p \in P \cap \left(\cup_{j=1}^{n_c} Q_{\boldsymbol{w}_j}\right)\}$.

It has been shown in [5] that a set of n_c GMECs can be enforced adding to the controlled net a set of n_c places called *monitors*, provided that the initial marking is legal. A simple rule to determine the monitors that guarantee the maximally permissiveness of the closed loop net was also given in [5], under the assumption that all transitions are controllable and observable.

C. Geometrical definitions

A *box* is a set of real vectors defined as

$$\mathcal{B}(\boldsymbol{l},\boldsymbol{u}) = \{\boldsymbol{x} \in \mathbb{R}^d | \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}\},$$

where $\boldsymbol{l}$ and $\boldsymbol{u}$ are real d-vectors.

If $\boldsymbol{x} \in \mathbb{N}^d$ we call $\mathcal{B}(\boldsymbol{0},\boldsymbol{u})$ *integer box* and we denote it simply as $\mathcal{I}(\boldsymbol{u})$.

An *hypercube* is a box such that $\boldsymbol{u} = \boldsymbol{l} + \lambda\boldsymbol{e}$, where λ is a scalar and $\boldsymbol{e}$ denotes the d-vector of ones; an *integer hypercube* is an integer box $\mathcal{I}(\boldsymbol{u})$ such that $\boldsymbol{u} = \lambda\boldsymbol{e}$ where λ is a positive integer scalar.

III. Problem statement

Let $\langle N, \boldsymbol{m}_{p0}\rangle$ be a P/T system to be controlled, where $N = (P, T, \boldsymbol{Pre}, \boldsymbol{Post})$.

Assume that a global specification is given in terms of a GMEC $(\boldsymbol{W},\boldsymbol{k})$. Without loss of generality we take $Q_{\boldsymbol{W}} = P$, i.e., all places are bounded by the constraint. If such is not the case, we can simply apply the proposed procedure to the projection on $Q_{\boldsymbol{W}}$.

Assume that the set of places P is partitioned into ν subsets $P_1, \ldots, P_\nu$, i.e., $P_i \cap P_j = \emptyset$ if $i \neq j$, and $\cup_{i=1}^{\nu} P_i = P$.

We want to determine a set of *decentralized* GMECs $(\boldsymbol{W}^{(i)}, \boldsymbol{k}^{(i)})$ whose support is P_i, with $i = 1, \ldots, \nu$, such that

$$\cap_{i=1}^{\nu} \mathcal{M}(\boldsymbol{W}^{(i)}, \boldsymbol{k}^{(i)}) \subseteq \mathcal{M}(\boldsymbol{W},\boldsymbol{k}). \quad (1)$$

The choice of the decentralized GMECs is obviously not unique, and depends in general on the dimension $n_c^{(i)}$ of the decentralized GMECs $(\boldsymbol{W}^{(i)}, \boldsymbol{k}^{(i)})$.

In this paper we make the following three assumptions.

(A1) All weights are positive, i.e., $\boldsymbol{W} \geq \boldsymbol{0}$.

(A2) All transitions are controllable and observable.

(A3) The support of each decentralized GMEC is a singleton, thus $\nu = m$ and $P_i = \{p_i\}$, for $i = 1, \dots, m$.

By assumption (A3) it follows that $n_c^{(i)} = 1$ and the effect of each decentralized GMEC is that of imposing an upper bound on the corresponding place. Thus, the set

$$\cap_{i=1}^{\nu} \mathcal{M}(\boldsymbol{W}^{(i)}, \boldsymbol{k}^{(i)})$$

can be regarded as the integer box $\mathcal{I}(\boldsymbol{u})$ where $\boldsymbol{u}$ is an m-integer vector whose i-th component denotes the bound induced by the i-the decentralized GMEC on place p_i.

Our goal here is that of trying to determine a systematic procedure to select $\boldsymbol{u}$ in order to guarantee fairness among places.

IV. Preliminary results

In this section we present some definitions and preliminary results that will be useful in the following.

Definition 1: An integer box $\mathcal{I}(\boldsymbol{u}) \subseteq \mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$ is a *maximal integer inner box* if there does not exist an inner box $\mathcal{I}(\tilde{\boldsymbol{u}}) \neq \mathcal{I}(\boldsymbol{u})$ such that $\mathcal{I}(\boldsymbol{u}) \subsetneq \mathcal{I}(\tilde{\boldsymbol{u}}) \subseteq \mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$. ■

Note that the maximal integer inner box is in general not unique.

Proposition 2: An integer box $\mathcal{I}(\boldsymbol{u})$ is a *maximal integer inner box* in $\mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$ where $\boldsymbol{W} \geq \boldsymbol{0}$ and $\boldsymbol{k} \geq \boldsymbol{0}$, if and only if $\forall\, p \in P$:

$$\min_{i \in \{1,\dots,n_c\}} \frac{k_i - \boldsymbol{w}_i \cdot \boldsymbol{u}}{\boldsymbol{w}_i(p)} < 1.$$

Proof: *(if)* Let p be any place in P, and

$$\frac{k_{\bar{\imath}} - \boldsymbol{w}_{\bar{\imath}} \cdot \boldsymbol{u}}{\boldsymbol{w}_{\bar{\imath}}(p)} < 1 \quad \Rightarrow \quad k_{\bar{\imath}} - \boldsymbol{w}_{\bar{\imath}} \cdot \boldsymbol{u} < \boldsymbol{w}_{\bar{\imath}}(p),$$

i.e., $\boldsymbol{w}_{\bar{\imath}} \cdot \tilde{\boldsymbol{u}} > k_{\bar{\imath}}$ where $\tilde{\boldsymbol{u}} = \boldsymbol{u} + \vec{\varepsilon}_j$, and $\vec{\varepsilon}_j$ is the j-th canonical basis vector whose entry is equal to one in correspondence to place p. This means that if the bound on p is increased of one unity this would lead to the violation of the $\bar{\imath}$-th GMEC. Since this is true for any place $p \in P$, we conclude that $\mathcal{I}(\boldsymbol{u})$ is a maximal inner box.

(only if) We prove this by contradiction. Assume that $\mathcal{I}(\boldsymbol{u})$ is a maximal integer inner box and $\exists\, p \in P$ such that

$$\min_{i \in \{1,\dots,n_c\}} \frac{k_i - \boldsymbol{w}_i \cdot \boldsymbol{u}}{\boldsymbol{w}_i(p)} \geq 1.$$

This implies that $\forall i \in \{1,\ldots,n_c\}$, $k_i - \boldsymbol{w}_i \cdot \boldsymbol{u} \geq \boldsymbol{w}_i(p)$. Thus, given an arbitrary place $p \in P$, we can define the vector $\tilde{\boldsymbol{u}} = \boldsymbol{u} + \vec{\varepsilon}_j$, where $\vec{\varepsilon}_j$ is defined as in the previous statement, and $k_i - \boldsymbol{w}_i \cdot \tilde{\boldsymbol{u}} \geq 0$, or equivalently $\boldsymbol{w}_i \cdot \tilde{\boldsymbol{u}} \leq k_i$. This implies that $\tilde{\boldsymbol{u}}$ satisfies all the constraints, thus leading to a contradiction. □

Thus, an integer box $\mathcal{I}(\boldsymbol{u})$ is maximal if and only if in each direction there exists at least one constraint that is saturated.

The following proposition provides a criterion to determine the *maximal integer hypercube* in $\mathcal{M}(\boldsymbol{W},\boldsymbol{k})$.

Proposition 3: Let $\bar{\mathcal{M}} = \mathcal{M}(\boldsymbol{W},\boldsymbol{k})$ be a marking set containing the null marking ($\boldsymbol{m} = \boldsymbol{0}$), so that $\boldsymbol{k} \geq \boldsymbol{0}$. Let us denote as

$$\tau(\bar{\mathcal{M}}) = \max \{\tau \in \mathbb{N} \mid \mathcal{I}(\tau \boldsymbol{e}) \subseteq \bar{\mathcal{M}}\}.$$

It holds

$$\tau(\bar{\mathcal{M}}) = \min_{i=1,\ldots,n_c} \tau(i,\bar{\mathcal{M}})$$

where

$$\tau(i,\bar{\mathcal{M}}) = \left\lfloor \frac{k_i}{\sum_{p\in P} \boldsymbol{w}_i(p)} \right\rfloor$$

and $\lfloor\,\rfloor$ denotes the floor operator.

Proof: The above statement follows from a result presented in [3] where the problem of maximizing the volume of hypercubes included in polytopes was considered. Note however that in [3] the floor operator was not present. It is used here being $\mathcal{I}(\tau \boldsymbol{e})$ an integer hypercube. ■

Finally, we recall the following definition.

Definition 4: An integer box $\mathcal{I}(\boldsymbol{u}) \subseteq \mathcal{M}(\boldsymbol{W},\boldsymbol{k})$ is a *maximal cardinality inner box* if there does not exist an inner box $\mathcal{I}(\tilde{\boldsymbol{u}}) \neq \mathcal{I}(\boldsymbol{u})$ such that $\mathcal{I}(\tilde{\boldsymbol{u}}) \subseteq \mathcal{M}(\boldsymbol{W},\boldsymbol{k})$ and $|\mathcal{I}(\boldsymbol{u})| < |\mathcal{I}(\tilde{\boldsymbol{u}})|$. ■

V. Maximal integer inner box computation

In this section we deal with the problem of determining a maximal integer inner box $\mathcal{I}(\boldsymbol{u}^*)$ in $\mathcal{M}(\boldsymbol{W},\boldsymbol{k})$ where $\boldsymbol{W} \geq \boldsymbol{0}$ and $\boldsymbol{k} \geq \boldsymbol{0}$.

Different criteria can be used. The most trivial is briefly summarized in the following algorithm.

Algorithm 5: [Maximal integer inner box, a trivial solution]

1. Let $\boldsymbol{k}^0 = \boldsymbol{k}$, $U_0 = \{1, \ldots, m\}$.

2. For $s = 1$ to m do

2.1. let $\bar{\jmath}_s$ be an index arbitrarily chosen in U_{s-1}

2.2. let $\boldsymbol{u}^*(p_{\bar{\jmath}_s}) = \min_{i \in \{1,\ldots,n_c\}} \left\lfloor \dfrac{k_i^{s-1}}{\boldsymbol{w}_i(p_{\bar{\jmath}_s})} \right\rfloor$

2.3. for $i = 1$ to n_c do

$$\text{let } k_i^s = k_i^{s-1} - \boldsymbol{u}^*(p_{\bar{\jmath}_s}) w_i(p_{\bar{\jmath}_s}).$$

2.4. let $U_s = U_{s-1} \setminus \{\bar{\jmath}_s\}$.

In simple words the above algorithm looks at all places in an arbitrary order, and assigns them the *largest* upper bound that guarantees the satisfaction of all constraints. Obviously, in this way we are not ensuring fairness among places and major chance to saturate the constraints is given to places that are firstly considered.

Example 6: Let us consider for simplicity the case of a single GMEC: $m_1 + m_2 \leq 3$. Assume that Algorithm 5 is used to design the decentralized monitors. If $\bar{\jmath}_1 = 1$ and $\bar{\jmath}_2 = 2$, i.e., we first assign the upper bound to p_1, then we get:

$$\begin{cases} m_1 \leq 3 \\ m_2 \leq 0. \end{cases}$$

On the contrary, if $\bar{\jmath}_1 = 2$ and $\bar{\jmath}_2 = 1$, we obtain:

$$\begin{cases} m_1 \leq 0 \\ m_2 \leq 3. \end{cases}$$

■

In practical applications the solution resulting from Algorithm 5 is usually inadequate, thus in this section we look for different criteria that ensure fairness among places. In particular, we provide a first algorithm to compute an integer inner box $\mathcal{I}(\boldsymbol{u}^*) \subseteq \mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$, that can be summarized in the following items. Then we show under which assumptions such an algorithm guarantees that the resulting integer inner box is maximal. In the case that the maximality is not guaranteed we show how to modify it in order to do so.

- The algorithm is based on m iterative steps. At each step s we define a GMEC $(\boldsymbol{W}^s, \boldsymbol{k}^s)$, choosing at the initial step $(\boldsymbol{W}^0, \boldsymbol{k}^0) = (\boldsymbol{W}, \boldsymbol{k})$. We denote $\mathcal{M}_s = \mathcal{M}(\boldsymbol{W}^s, \boldsymbol{k}^s)$.

- At step s we compute the maximal integer hypercube in $\mathcal{M}_{s-1}$ using Proposition 3, and denote τ_s the corresponding edge.
- At each step we eliminate one place appropriately chosen from the support of the current GMEC, and assign to it an upper bound which coincides with the edge of the current hypercube. Thus, if $p_{\bar{\jmath}_s}$ is the place we eliminate at step s, it results $\boldsymbol{u}^*(p_{\bar{\jmath}_s}) = \tau_s$.
- The choice of the place to eliminate is essential to make sure that, at least under an appropriate condition that is discussed in the following, a maximal inner box is obtained. Assume we are considering a constraint $\boldsymbol{w}_i \cdot \boldsymbol{m} \leq k_i$ with $\boldsymbol{w}_i \geq \boldsymbol{0}$, and an integer hypercube with edge τ satisfying it. We define the *slack* of constraint i as

$$s_i = k_i - \tau \boldsymbol{w}_i \cdot \boldsymbol{1}$$

  where $\boldsymbol{1}$ is a column vector of ones, and the *relative slack* of constraint i with respect to place p_j as

$$s_{i,j} = \frac{k_i - \tau \boldsymbol{w}_i \cdot \boldsymbol{1}}{\boldsymbol{w}_i(p_j)}.$$

  We choose to eliminate at each step the place that corresponds to the smallest relative slack. We discuss in Proposition 10 under which condition this choice leads to a maximal inner box.
- A new GMEC involving all places apart from those eliminated at the previous steps is written, where the weights associated to the remaining places do not change, while $\boldsymbol{k}^s$ is updated to $k_i^s = k_i^{s-1} - \tau_s \cdot \boldsymbol{w}_i(p_{\bar{\jmath}_s})$, $i = 1, \ldots, n_c$. In fact, if we denote as U_s the set of indexes of places in the support of $\mathcal{M}_{s-1}$, the two constraints

$$\begin{cases} \sum_{j \in U_s} \boldsymbol{w}_i^s(p_j)\boldsymbol{m}(p_j) \leq \boldsymbol{k}_i^s - \boldsymbol{w}_i^s(p_{\bar{\jmath}_s})\tau_s \\ \boldsymbol{m}(p_{\bar{\jmath}_s}) \leq \tau_s \end{cases} \tag{2}$$

  guarantee that

$$\sum_{j \in U_{s-1}} \boldsymbol{w}_i^s(p_j)\boldsymbol{m}(p_j) \leq \boldsymbol{k}_i^s, \tag{3}$$

  namely, (2) guarantees the satisfaction of the GMEC at the previous step.

Formally, the algorithm can be written as follows.

Algorithm 7: [Inner box computation]

1. Let $\tau_0 = 0$, $\boldsymbol{W}^0 = \boldsymbol{W}$, $\boldsymbol{k}^0 = \boldsymbol{k}$, $\boldsymbol{u}_0 = \boldsymbol{0}$,
$U_0 = \{1, \ldots, m\}$.

2. For $s = 1$ to m do

2.1. let $\mathcal{M}_{s-1} = \mathcal{M}(\boldsymbol{W}^{s-1}, \boldsymbol{k}^{s-1})$

2.2. let $\tau_s = \tau(\mathcal{M}_{s-1})$ (see Proposition 3)

2.3. let $\bar{\jmath}_s$ be an index arbitrarily chosen in

$$J_s = \left\{ \bar{\jmath} \in \mathbb{N} \ \mid \ s_{\bar{\imath},\bar{\jmath}}^{s-1} = \min_{\substack{j \in U_{s-1} \\ i \in \{1,\dots,n_c\}}} s_{i,j}^{s-1} \right\}$$

where $s_{i,j}^{s-1} = \dfrac{k_i^{s-1} - \tau \boldsymbol{w}_i^{s-1} \cdot \mathbf{1}}{\boldsymbol{w}_i(p_j)}$

2.4. for $i = 1$ to n_c do

let $\boldsymbol{w}_i^s(p_j) = \begin{cases} 0 & j = \bar{\jmath}_s \\ \boldsymbol{w}_i^{s-1}(p_j) & \text{otherwise} \end{cases}$

let $k_i^s = k_i^{s-1} - \tau_s \cdot \boldsymbol{w}_i(p_{\bar{\jmath}_s})$

2.5. let $\boldsymbol{u}_s(p_j) = \begin{cases} \tau_s & j = \bar{\jmath}_s \\ \boldsymbol{u}_{s-1}(p_j) & \text{otherwise} \end{cases}$

2.6. let $U_s = U_{s-1} \setminus \{\bar{\jmath}_s\}$

3. let $\boldsymbol{u}^* = \boldsymbol{u}^m$. ■

We now formally prove a rather intuitive result that will be used in the following.

Proposition 8: At each step of the previous algorithm it results $\tau_s \geq \tau_{s-1}$.

Proof: By definition, at the iteration $s+1$, it holds

$$\begin{aligned}
\tau_{s+1} &= \min_{i=1,\dots,n_c} \left\lfloor \frac{k_i^s}{\sum_{j \in U_s} w_i(p_j)} \right\rfloor \\
&= \min_{i=1,\dots,n_c} \left\lfloor \frac{k_i^{s-1} - \tau_s w_i(p_{\bar{\jmath}_s})}{\sum_{j \in U_s} w_i(p_j) + w_i(p_{\bar{\jmath}_s}) - w_i(p_{\bar{\jmath}_s})} \right\rfloor \\
&\geq \min_{i=1,\dots,n_c} \left\lfloor \frac{\tau_s(\sum_{j \in U_{s-1}} w_i(p_j) - w_i(p_{\bar{\jmath}_s}))}{\sum_{j \in U_{s-1}} w_i(p_j) - w_i(p_{\bar{\jmath}_s})} \right\rfloor \\
&= \tau_s
\end{aligned}$$

where the inequality follows from the obvious observation that by definition of τ_s, $\sum_{j \in U_{s-1}} w_i(p_j)\tau_s \leq k_i^{s-1}$. □

We can now prove the following results.

Proposition 9: Let $(\boldsymbol{W}, \boldsymbol{k})$ be a centralized constraint with $\boldsymbol{W} \geq \mathbf{0}$ and $\boldsymbol{k} \geq \mathbf{0}$, and $\boldsymbol{u}^*$ be the upper bound vector determined by Algorithm 7. Then $\mathcal{I}(\boldsymbol{u}^*) \subseteq \mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$.

Proof: We first observe that for all $i = 1, \dots, n_c$, it holds:

$$\begin{aligned} 0 \le \bar{k}_i^m &= k_i - \textstyle\sum_{s=1}^m \boldsymbol{w}_i(p_{\bar{\jmath}_s})\tau_s \\ &= k_i - \textstyle\sum_{s=1}^m \boldsymbol{w}_i(p_{\bar{\jmath}_s})\boldsymbol{u}_s(p_{\bar{\jmath}_s}) \\ &= k_i - \textstyle\sum_{s=1}^m \boldsymbol{w}_i(p_{\bar{\jmath}_s})\boldsymbol{u}^*(p_{\bar{\jmath}_s}) = k_i - \boldsymbol{w}_i \cdot \boldsymbol{u}^* \end{aligned}$$

Since for all $\boldsymbol{m} \in \mathcal{I}(\boldsymbol{u}^*)$ it is possible to write $\boldsymbol{w}_i \cdot \boldsymbol{m} \le \boldsymbol{w}_i \cdot \boldsymbol{u}^* = k_i - \bar{k}_i^m \le k_i$, it follows that $\boldsymbol{m} \in \mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$. □

In Proposition 8 we have shown that the sequence of edges τ_i of the maximal integer hypercubes determined by Algorithm 7 is nondecreasing. Next proposition shows that if this sequence is *strictly* increasing (with the possible exception of the tail of the sequence that may remain constant) a maximal inner box is obtained.

Proposition 10: Let $(\boldsymbol{W}, \boldsymbol{k})$ be a centralized constraint and $\boldsymbol{u}^*$ be the final upper bound vector computed by Algorithm 7. If there exists an index $\mu \le m$ such that the sequence of τ's computed by Algorithm 7 satisfies the condition

$$\tau_1 < \tau_2 < \cdots < \tau_\mu = \tau_{\mu+1} = \cdots = \tau_m \tag{4}$$

then $\mathcal{I}(\boldsymbol{u}^*)$ is a *maximal inner box* included in $\mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$.

Proof: Proposition 10 has already shown that the box is included in $\mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$. We will prove by contradiction that it is also maximal if condition (4) holds. Suppose that there exists an inner box $\mathcal{I}(\tilde{\boldsymbol{u}})$ such that $\mathcal{I}(\boldsymbol{u}^*) \subsetneq \mathcal{I}(\tilde{\boldsymbol{u}}) \subseteq \mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$, i.e., such that $\tilde{\boldsymbol{u}} \gneq \boldsymbol{u}^*$. Then, there must exist an index h such that $\tilde{\boldsymbol{u}}(p_h) > \boldsymbol{u}^*(p_h)$. Assume, without loss of generality that $\tilde{\boldsymbol{u}}(p_h) = \boldsymbol{u}^*(p_h)+1$ and $\tilde{\boldsymbol{u}}(p_j) = \boldsymbol{u}^*(p_j), \forall j \ne h$. Suppose that $\boldsymbol{u}^*(p_h)$ has been fixed at l-th step of Algorithm 7, i.e., $\boldsymbol{u}^*(p_h) = \tau_l$.

Furthermore, since $\mathcal{I}(\tilde{\boldsymbol{u}}) \subseteq \mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$ it holds for all $i = 1, \dots, n_c$,

$$\begin{aligned} \sum_{j \in U_{l-1}} \boldsymbol{w}_i(p_j)\tilde{\boldsymbol{u}}(p_j) &\le k_i - \sum_{j \in U_0 \setminus \{U_{l-1}\}} \boldsymbol{w}_i(p_j)\tilde{\boldsymbol{u}}(p_j) \\ &= k_i - \sum_{j \in U_0 \setminus \{U_{l-1}\}} \boldsymbol{w}_i(p_j)\boldsymbol{u}^*(p_j) \\ &= k_i^{l-1} \end{aligned} \tag{5}$$

where $U_0 \setminus \{U_{l-1}\}$ contains the indexes of the places eliminated in the first $l-1$ iterations of the algorithm.

We consider two cases.

Case I: $l < \mu$.

Condition (4) implies that $\forall j \in U_l$ it holds $\tilde{\boldsymbol{u}}(p_j) = \boldsymbol{u}^*(p_j) \ge \boldsymbol{u}^*(p_h) + 1 = \tau_l + 1$. Since $\tilde{\boldsymbol{u}}(p_h) = \tau_l + 1$, we can also conclude that for all $j \in U_{l-1} = U_l \cup \{h\}$ it holds $\tilde{\boldsymbol{u}}(p_j) \ge \tau_l + 1$, i.e, from (5) we have that for all $i = 1, \dots, n_c$,

$$\sum_{j \in U_{l-1}} \boldsymbol{w}_i(p_j)(\tau_l + 1) \le \sum_{j \in U_{l-1}} \boldsymbol{w}_i(p_j)\tilde{\boldsymbol{u}}(p_j) \le k_i^{l-1}.$$

This means that at step l an hypercube with edge τ_l+1 should have been chosen by the algorithm. Clearly this leads to a contradiction.

Case II: $l \geq \mu$.

First we note that in this case for all $j \in U_{l-1}$ it holds $\boldsymbol{u}^*(p_j) = \tau_l = \tau_m = \tau$ and, for all $i = 1, \ldots, n_c$, we can rewrite (5) as

$$\boldsymbol{w}_i(p_h) + \sum_{j \in U_{l-1}} \boldsymbol{w}_i(p_j)\tau \leq k_i^{l-1}. \tag{6}$$

Then, using the fact that the algorithm eliminates at each step the place with minimal relative slack, we prove that it also holds

$$\boldsymbol{w}_i(p_{h'}) + \sum_{j \in U_{l-1}} \boldsymbol{w}_i(p_j)\tau \leq k_i^{l-1}, \tag{7}$$

where $p_{h'}$ is the place removed at step m of the algorithm. In fact, it is not difficult to see that (6) implies that the relative slacks of places p_h satisfy, for all $i = 1, \ldots, n_c$, $s_{i,h}^{l-1} \geq 1$ and since it also holds $s_{i,h'}^{l-1} \geq \min_{i=1,\ldots,n_c} s_{i,h}^{l-1} \geq 1$ we obtain (7), that in turn can be rewritten, for all $i = 1, \ldots, n_c$,

$$\boldsymbol{w}_i(p_{h'})(\tau+1) \leq k_i^{l-1} - \sum_{j \in U_{l-1} \setminus \{h'\}} \boldsymbol{w}_i(p_j)\tau = k_i^{m-1}.$$

Hence we observe that at the last step the algorithm should have assigned to place $p_{h'}$ a bound $\boldsymbol{u}^*(p_{h'}) = \tau + 1$, thus reaching a contradiction. □

Let us now discuss a numerical example that clearly shows that the inner box $\mathcal{I}(\boldsymbol{u}^*)$ computed using Algorithm 7 may be not maximal if condition (4) does not hold.

Example 11: Let

$$\begin{aligned} \mathcal{M}(\boldsymbol{W}, \boldsymbol{k}) = \{ \quad & \boldsymbol{m} \in \mathbb{N}^3 \quad | \\ & 20\boldsymbol{m}(p_1) + 19\boldsymbol{m}(p_2) + \boldsymbol{m}(p_3) \leq 61 \\ & \boldsymbol{m}(p_1) + 22\boldsymbol{m}(p_2) + 21\boldsymbol{m}(p_3) \leq 84\}. \end{aligned}$$

Applying Algorithm 7 the following results are obtained.

We initially set

$$\begin{aligned} \boldsymbol{w}_1^0 &= \begin{bmatrix} 20 & 19 & 1 \end{bmatrix}, \; k_1^0 = 61 \\ \boldsymbol{w}_2^0 &= \begin{bmatrix} 1 & 22 & 21 \end{bmatrix}, \; k_2^0 = 84, \; U_0 = \{1, 2, 3\}. \end{aligned}$$

By applying iteratively step 2

$s = 1$

$$\begin{aligned} &\tau_1 = \lfloor \tfrac{61}{40} \rfloor = 1, \; J_1 = \{1\} \\ &\boldsymbol{w}_1^1 = \begin{bmatrix} 0 & 19 & 1 \end{bmatrix}, \; k_1^1 = 61 - 20\tau_1 = 41 \\ &\boldsymbol{w}_2^1 = \begin{bmatrix} 0 & 22 & 21 \end{bmatrix}, \; k_2^1 = 84 - 1\tau_1 = 83 \\ &\boldsymbol{u}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \; U_1 = \{2, 3\}. \end{aligned}$$

$s = 2$

$$\tau_2 = \lfloor \tfrac{21}{19} \rfloor = 1, \; J_2 = \{2\}$$
$$\boldsymbol{w}_1^2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \; k_1^2 = 41 - 19\tau_2 = 22$$
$$\boldsymbol{w}_2^2 = \begin{bmatrix} 0 & 0 & 21 \end{bmatrix}, \; k_2^2 = 83 - 22\tau_2 = 61$$
$$\boldsymbol{u}_2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}, \; U_2 = \{3\}.$$

$s = 3$

$$\tau^3 = \lfloor \tfrac{61}{21} \rfloor = 2, \; J^3 = \{3\}$$
$$\boldsymbol{w}_1^3 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \; k_1^3 = 22 - \tau_3 = 21$$
$$\boldsymbol{w}_2^3 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \; k_2^3 = 61 - 21\tau_3 = 19$$
$$\boldsymbol{u}^* = \boldsymbol{u}_3 = \begin{bmatrix} 1 & 1 & 2 \end{bmatrix}, \; U_3 = \emptyset.$$

The resulting inner box is $\mathcal{I}(\boldsymbol{u}^*)$ with $\boldsymbol{u}^* = [1 \ 1 \ 2]^T$.

The sequence of maximal edges computed by the algorithm is $\tau_1 = 1, \tau_2 = 1, \tau_3 = 2$ that does not satisfy condition (4). Inner box $\mathcal{I}(\boldsymbol{u}^*)$ is not maximal. In fact, if we consider $\tilde{\boldsymbol{u}} = [2 \ 1 \ 2]^T$, it holds that $\mathcal{I}(\boldsymbol{u}^*) \subsetneq \mathcal{I}(\tilde{\boldsymbol{u}}) \subseteq \mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$. ■

However, Algorithm 7 may be easily modified in order to guarantee that the resulting inner box is maximal.

Algorithm 12: [Maximal inner box computation]

1. Run Algorithm 7. Assume that the sequence of τ's is $\tau_1 \leq \tau_2 \cdots \leq \tau_\mu = \cdots = \tau_m$.

2. Let $\bar{\boldsymbol{u}}^0 = \boldsymbol{u}^m$.

3. For $s = 1$ to $\mu - 1$ do

3.1. for $j = 1, \ldots, m, \; j \neq \bar{\jmath}_s$

let $\bar{\boldsymbol{u}}^s(p_j) = \bar{\boldsymbol{u}}^{s-1}(p_j)$

3.2. let

$$\bar{\boldsymbol{u}}^s(p_{\bar{\jmath}_s}) = \begin{cases} \bar{\boldsymbol{u}}^{s-1}(p_{\bar{\jmath}_s}) & \text{if } \tau_s < \tau_{s+1} \\ \bar{\boldsymbol{u}}^{s-1}(p_{\bar{\jmath}_s}) + & \\ \quad + \min\limits_{i \in \{1,\ldots,n_c\}} \left\lfloor \dfrac{k_i - \boldsymbol{w}_i \cdot \bar{\boldsymbol{u}}^{s-1}}{\boldsymbol{w}_i(p_{\bar{\jmath}_s})} \right\rfloor & \\ & \text{if } \tau_s = \tau_{s+1} \end{cases}$$

4. let $\boldsymbol{u}^* = \bar{\boldsymbol{u}}^{\mu-1}$. ■

The main idea behind the new steps of the algorithm is the following. The solution computed using Algorithm 7 provides a maximal inner box when the sequence of τ's is strictly increasing, apart from the tail of the sequence that may keep constant. On the contrary, no guarantee is given if two or more τ's that are not in the tail are equal. Therefore, we look for all variables to which it corresponds the same upper bound that is different from τ_m, and we verify if their upper bounds may be further increased. If so, we increase them as much as possible in accordance with the given constraints, and go further with our exploration.

Note that, by Proposition 8, at step 1 of Algorithm 12, only places whose bound has been assigned in consecutive steps may have equal upper bounds.

Proposition 13: Let $(\boldsymbol{W}, \boldsymbol{k})$ be a centralized constraint with $\boldsymbol{W} \geq \boldsymbol{0}$ and $\boldsymbol{k} \geq 0$. Let $\boldsymbol{u}^*$ be the upper bound vector determined by Algorithm 12. Then $\mathcal{I}(\boldsymbol{u}^*)$ is a *maximal inner box* included in $\mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$.

Proof: If $\tau_s < \tau_{s+1}$ for all $s = 1, \ldots, \mu - 1$, and $\tau_s = \tau_{s+1}$ for all $s = \mu, \ldots, m-1$, the solution provided by Algorithm 12 coincides with that of Algorithm 7 and the result follows from Proposition 10.

Now, let r be the smallest value of $s \in \{1, \ldots, \mu - 1\}$ such that $\tau_s = \tau_{s+1}$. Then, at step 4.2 we impose:

$$\begin{aligned}
\bar{\boldsymbol{u}}^r(p_{\bar{\jmath}_r}) &= \bar{\boldsymbol{u}}^{r-1}(p_{\bar{\jmath}_r}) + \min_{i \in \{1,\ldots,n_c\}} \left\lfloor \frac{k_i - \boldsymbol{w}_i \cdot \bar{\boldsymbol{u}}^{r-1}}{\boldsymbol{w}_i(p_{\bar{\jmath}_r})} \right\rfloor \\
&= \bar{\boldsymbol{u}}^{r-1}(p_{\bar{\jmath}_r}) + \min_{i \in \{1,\ldots,n_c\}} \left\lfloor \frac{k_i - \boldsymbol{w}_i \cdot \bar{\boldsymbol{u}}^{0}}{\boldsymbol{w}_i(p_{\bar{\jmath}_r})} \right\rfloor
\end{aligned}$$

while $\bar{\boldsymbol{u}}^r(p_j) = \bar{\boldsymbol{u}}^0(p_j)$ for all $j \neq \bar{\jmath}_r$.

Let $\bar{k}_i^0 = k_i - \boldsymbol{w}_i \cdot \bar{\boldsymbol{u}}^0$ and $\bar{k}_i^r = k_i - \boldsymbol{w}_i \cdot \bar{\boldsymbol{u}}^r$, thus

$$\begin{aligned}
\bar{k}_i^r &= k_i - \boldsymbol{w}_i \cdot \bar{\boldsymbol{u}}^0 - \boldsymbol{w}_i(p_{\bar{\jmath}_r}) \min_{i \in \{1,\ldots,n_c\}} \left\lfloor \frac{k_i - \boldsymbol{w}_i \cdot \bar{\boldsymbol{u}}^{0}}{\boldsymbol{w}_i(p_{\bar{\jmath}_r})} \right\rfloor \\
&= \bar{k}_i^0 - \boldsymbol{w}_i(p_{\bar{\jmath}_r}) \min_{i \in \{1,\ldots,n_c\}} \left\lfloor \frac{\bar{k}_i^0}{\boldsymbol{w}_i(p_{\bar{\jmath}_r})} \right\rfloor
\end{aligned}$$

and

$$\frac{\bar{k}_i^r}{\boldsymbol{w}_i(p_{\bar{\jmath}_r})} = \frac{\bar{k}_i^0}{\boldsymbol{w}_i(p_{\bar{\jmath}_r})} - \min_{i \in \{1,\ldots,n_c\}} \left\lfloor \frac{\bar{k}_i^0}{\boldsymbol{w}_i(p_{\bar{\jmath}_r})} \right\rfloor < 1.$$

Therefore, by Proposition 2, $\mathcal{I}(\bar{u}_r)$ is a maximal inner box.

Similarly, if we denote as q the smallest value of $s \in \{r+1, \ldots \mu - 1\}$ such that $\tau_s = \tau_{s+1}$, we can prove that

$$\frac{\bar{k}_i^q}{\boldsymbol{w}_i(p_{\bar{\jmath}_q})} < 1,$$

thus, iteratively repeating the same reasoning until all places have been considered, we conclude that $\mathcal{I}(\bar{\boldsymbol{u}}^{\mu-1})$ is a maximal inner box. □

An important remark needs to be done.

Remark 14: For simplicity of notation, in Algorithm 12 we have assumed that the upper bounds are increased, when possible, following the same order in which they have been assigned in step 2. Clearly, this is not the only admissible solution. Variables that share the same upper bound may be examined in any order, and this in general provides different decentralized constraints. In any case Proposition 13 still applies, and all the resulting solutions are maximal inner boxes. ■

Example 15: Let us consider again the GMEC in Example 11. As already discussed above Algorithm 7 provides the inner box $\mathcal{I}(\boldsymbol{u}^*)$ with $\boldsymbol{u}^* = [1\ 1\ 2]^T$, that is not maximal.

We now apply Algorithm 12. In this case the steps where the same bound is assigned to more than one place are the first and the second one. Here the bound 1 is initially assigned to places p_1 and p_2, respectively. Thus we can be sure that the only variable whose bound may be increased is p_1. In particular, it holds

$$\begin{aligned} k_1 - \boldsymbol{w}_1 \cdot \boldsymbol{u}^* = 21 \\ k_2 - \boldsymbol{w}_2 \cdot \boldsymbol{u}^* = 19 \end{aligned}$$

thus

$$\boldsymbol{u}^*(p_1) = \tau_1 + \min_{i \in \{1,2\}} \left\lfloor \frac{k_i - \boldsymbol{w}_i \cdot \boldsymbol{u}^*}{\boldsymbol{w}_i(p_1)} \right\rfloor = 1 + 1 = 2.$$

■

VI. Conclusions and Future Works

In this paper we have investigated the problem of determining a set of decentralized GMECs that are able to impose a specification on the net behavior given in terms of a global set of n_c GMECs. In particular we assumed that the support of each decentralized GMEC is a singleton, thus the effect of each decentralized GMEC is that of imposing an upper bound on the marking of the corresponding place. An iterative algorithm is given to compute appropriate bounds that guarantee a satisfactory solution in terms of permissiveness and fairness among places.

Our future work will be that of removing the assumption that the support of each decentralized GMEC is a singleton and investigating how to determine the maximal cardinality inner box.

References

[1] G. Barret and S. Lafortune. Decentralized supervisory control with communicating controllers. *IEEE Transaction on Automatic Control*, 45(9):1620–1638, 2000.

[2] F. Basile, A. Giua, and C. Seatzu. Decentralized supervisory control of Petri nets with monitor places. 10^{th} *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'05), Catania, Italy*, Settembre 2005.

[3] A. Bemporad, C. Filippi, and F.D. Torrisi. Inner and outer approximations of polytopes using boxes. *Computational Geometry*, 27:151–178, 2004.

[4] H. Chen and H. Baosheng. Distributed control of discrete event systems described by a class of controlled Petri nets. *IFAC International Symposium on Distributed Intelligence Systems*, 1991.

[5] A. Giua, F. DiCesare, and M. Silva. Generalized mutual exclusion constraints on nets with uncontrollable transitions. *1992 IEEE Int. Conf. on Systems, Man, and Cybernetics (Chigago, Illinois)*, pages 974–979, October 1992.

[6] X. Guan and L.E. Holloway. Control of distributed discrete event systems modeled as Petri nets. *Proc. 1997 American Control Conference, Albuquerque, New Mexico*, pages 2342–2347, June 1997.

[7] M.V. Iordache and P.J. Antsaklis. Decentralized control of Petri nets with constraint tansformation. *IEEE Transaction on Automatic Control*, 51(2):376–381, February 2006.

[8] F. Lin and W.M. Wonham. Decentralized control and coordination of discrete-event systems with partial observation. *IEEE Transaction on Automatic Control*, 35(12):1330–1337, 1990.

[9] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of IEEE*, 77(4):541–580, April 1989.

[10] K. Rudie. The current state of decentralized discrete-event control systems. *Proc. 2002 IEEE mediterranean Conference on Control and Automation, Lisbon, Portugal*, June 2002.

[11] K. Rudie and W.M. Wonham. Think globally, act locally: Decentralized supervisory control. *IEEE Transaction on Automatic Control*, 37(11):1692–1708, 1992.

[12] S. Takai, S. Kodama, and T. Ushio. Decentralized state feedback control of discrete event systems. *Systems and Control Letters*, 22(5):369–375, 1994.