Marian Iordache, Panos J.Antsaklis, “Decentralized Control of Petri Nets with Constraint Transformations,”
Proceedings of the 2003 American Control Conference, pp. 314-319, Denver, Colorado, June 4-6, 2003.

Decentralized Control of Petri Nets with Constraint Transformations

Marian V. Iordache and Panos J. Antsaklis[1,2]

Abstract

Supervision based on place invariants (SBPI) has been effectively used for the centralized supervisory control of Petri nets. In the SBPI approach, specifications are classified as admissible or inadmissible, and inadmissible specifications are enforced by transforming them first to a (more restrictive) admissible form. This paper considers the transformation to admissible specifications in a decentralized setting. In this setting, the system can only be controlled and observed locally. The design goal is to find local supervisors, each controlling and observing a part of the system, such that a global specification is enforced. The feasibility of this problem is demonstrated with a simple integer programming approach. This approach can incorporate communication between local supervisors as well as communication constraints.

1 Introduction

The decentralized control of discrete event systems (DES) has received considerable attention in the recent years [13]. The current research effort has been focused on the automata setting, and has considered both versions of decentralized control, with communication and with no communication. This paper considers the decentralized control of Petri nets by means of the supervision based on place invariants (SBPI) [4, 10, 16].

The SBPI approach classifies the specifications as admissible and inadmissible, where the former can be directly enforced, and the latter are first transformed to an admissible form and then enforced. In the automata setting, admissibility corresponds to controllability and observability, and the transformation to an admissible form to the computation of a controllable and observable sublanguage. This paper demonstrates the feasibility of the transformation to an admissible form in the decentralized Petri net setting, using a linear integer programming technique. This approach is complementary to those we describe in the companion paper [7], effectively increasing the range of applications that can be tackled. With regard to our use of integer programming, note that while the development of alternative methods that are less computationally intensive are a direction for future research, in the automata setting it was shown that a decentralized solution cannot be found with polynomial complexity [13]. Note also that the size of the integer program depends on the size of the Petri net structure, and not on the size of its state space (i.e. the size of its equivalent automaton), which may not be finite.

The contribution of the paper is as follows. First, methods for the design of decentralized supervisors are proposed, in a decentralized setting with no communication. Second, a method for the design of decentralized supervisors with communication is proposed. Communication allows a local supervisor to observe/control transitions that are unobservable/uncontrollable in the part of the system it controls. Note that the design process generates both the local supervisors and the communication policy. Our approach allows communication constraints to be incorporated in the design process and can be used to minimize the network traffic.

The current work on the decentralized control of DES can be found in [13] and the references therein. In particular, we mention [11] for the decentralized control with no communication, and [1, 12] for decentralized control with communication. As in our paper, the communication in [12] consists of events rather than states estimates or observation strings. The existence of decentralized supervisors enforcing state predicates is studied in [15]. In a centralized framework, integer programming has been used for the computation of the optimal supervisor enforcing state predicates on a VDES (Petri net) in [8]. Literature on SBPI or closely related to it is found in [4, 10, 14] and the references therein. Finally, note that the decentralized control of DES can be used in various applications, including manufacturing [9], failure detection [2], and communication protocols [3].

The paper is organized as follows. Section 2 describes the supervisory approach. Section 3 illustrates our approach on a manufacturing example from [9]. The reader is referred to the companion paper [7] for notation and definitions.

[1]Department of Electrical Engineering, University of Notre Dame, IN 46556, USA. E-mail: iordache.1, antsaklis.1@nd.edu.

[2]The partial support of the Lockheed Martin Corporation, of the National Science Foundation (NSF ECS99-12458), and of DARPA/IXO-NEST Program (AF-F30602-01-2-0526) is gratefully acknowledged.

2 Constraint Transformations for Supervisor Design

Given a d-admissible set of constraints, a supervisor enforcing it can be easily constructed, as shown in the companion paper [7]. This section considers transformations of sets of constraints that are not d-admissible. These transformations aim to obtain (more restrictive) d-admissible constraints, in order to reduce the problem to the enforcement of d-admissible constraints. Two approaches are proposed: transformations to single sets of constraints and transformations to multiple sets of constraints. The former is a particular case of the latter, and can be done using techniques from the literature [10, 14]. As the transformation to a single set of constraints cannot deal effectively with some interesting problems, we will focus on the transformation to multiple sets of constraints. This approach will be presented in both supervisory frameworks, with communication and with no communication.

2.1 Transformation to a single set of constraints

A possible approach to transform a set of constraints to a d-admissible set of constraints is:

1. Select a nonempty subset $\mathcal{C}$ of $\{1, 2, \ldots n\}$.
2. Transform[3] the set of constraints to a c-admissible set of constraints with respect to $(\mathcal{N}, T_{uc}, T_{uo})$, for $T_{uc} = \bigcap_{i \in \mathcal{C}} T_{uc,i}$ and $T_{uo} = \bigcup_{i \in \mathcal{C}} T_{uo,i}$.

In practice, it may not be trivial to select the “best” set $\mathcal{C}$. However, for some particular cases the choice of $\mathcal{C}$ is more obvious:

- If $T_{uo,1} = T_{uo,2} = \ldots T_{uo,n}$ (in particular, this is true when full observability is available in each subsystem: $T_{uo,i} = \emptyset\ \forall i = 1 \ldots n$), then $\mathcal{C}$ can be chosen as $\mathcal{C} = \{1, 2, \ldots n\}$, to minimize the number of transitions in T_{uc}.
- If $T_{o,i} \cap T_{o,j} = \emptyset$ for all distinct $i, j = 1 \ldots n$, then we could attempt to set $\mathcal{C}$ to each of $\{1\}$, $\{2\}$, … $\{n\}$, do in each case the transformation to admissible constraints, and then select the one yielding the least restrictive constraints.

The main drawback of this approach is that it fails for many interesting systems and constraints. For instance, it fails to provide a solution for the system of Figure 1, with $T_{uc,2} = T_{uo,2} = \{t_1, t_2\}$, $T_{uc,1} = T_{uo,1} = \{t_3, t_4\}$, initial marking as shown in figure, and specification

$$\mu_1 + \mu_3 \leq 2 \tag{1}$$

Indeed, no matter how $\mathcal{C}$ is chosen, no d-admissible inequality implying (1) is satisfied by the initial marking. However, it is possible to enforce (1) with two d-admissible inequalities

$$\mu_1 \leq 1 \tag{2}$$
$$\mu_3 \leq 1 \tag{3}$$

as shown in Figure 1. However, note that none of (2) and (3), by itself, implies (1). This example motivates the transformation to multiple constraints, which is presented next.

2.2 Transformation to multiple sets of constraints

The problem can be stated as follows: *Given a set of constraints $L\mu \leq b$ that is not d-admissible, find d-admissible sets of constraints $L_1\mu \leq b_1 \ldots L_m\mu \leq b_m$ such that*

$$(L_1\mu \leq b_1 \wedge L_2\mu \leq b_2 \wedge \ldots L_m\mu \leq b_m) \Rightarrow L\mu \leq b \tag{4}$$

Compared to the previous approach, we now use several sets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m$ to design each of the $L_1\mu \leq b_1$, $L_2\mu \leq b_2, \ldots, L_m\mu \leq b_m$, instead of a single set $\mathcal{C}$. For instance, if $T_{o,i} \cap T_{o,j} = \emptyset$ for all $i \neq j$, then we may take $\mathcal{C}_i = \{i\}$, for all i. Furthermore, note that this framework includes the case when not all constraints $L_i\mu \leq b_i$ are necessary to implement $L\mu \leq b$, by allowing $L_i = 0$ and $b_i = 0$.

In general, (4) may have many solutions, not all equally interesting. In order to have a more interesting solution, we can use a set of markings of interest $\mathcal{M}_I$, and constrain each L_i and b_i to satisfy $L_i\mu \leq b_i\ \forall \mu \in \mathcal{M}_I$. This condition can be written as

$$L_i M \leq b_i \mathbf{1}^T \tag{5}$$

where $\leq$ means that each element of L_iM is less or equal to the element of the same indices in $b_i\mathbf{1}^T$, M is a matrix whose columns are the markings of interest, and $\mathbf{1}^T$ is a row vector of appropriate dimension in which all elements are 1.

The problem is more tractable if we replace (4) with the stronger condition below:

$$\left[\left(\sum_{i=1}^{m} \alpha_i L_i\right)\mu \leq \left(\sum_{i=1}^{m} \alpha_i b_i\right)\right] \Rightarrow L\mu \leq b \tag{6}$$

where α_i are nonnegative scalars. Without loss of generality, (6) assumes that $L_1 \ldots L_m$ have the same number of rows. Again, without loss of generality, (6) can be replaced by

$$[(L_1 + L_2 + \ldots L_m)\mu \leq (b_1 + b_2 + \ldots b_m)] \Rightarrow L\mu \leq b \tag{7}$$

We further simplify our problem to

$$L_1 + L_2 + \ldots L_m = R_1 + R_2 L \tag{8}$$
$$b_1 + b_2 + \ldots b_m = R_2(b + \mathbf{1}) - \mathbf{1} \tag{9}$$

[3]Techniques that can be used to perform this transformation appear in [10, 14].

Marian Iordache, Panos J.Antsaklis, "Decentralized Control of Petri Nets with Constraint Transformations," Proceedings of the 2003 American Control Conference, pp. 314-319, Denver, Colorado, June 4-6, 2003.

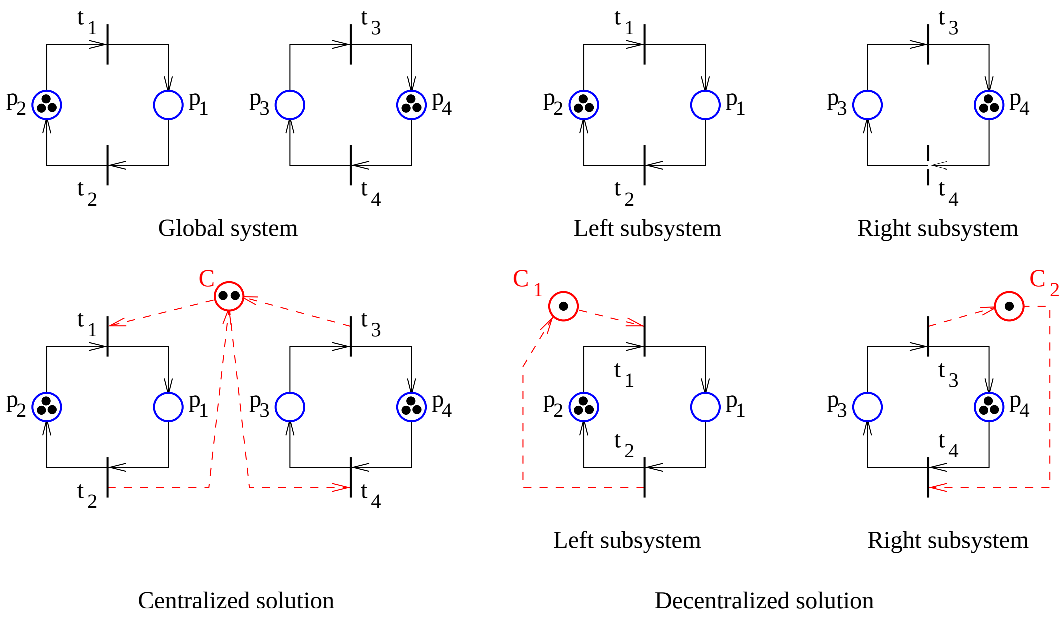

Figure 1: Decentralized control example.

for R_1 with nonnegative integer elements and R_2 diagonal with positive integer diagonal elements. Note that $[(R_1+R_2L)\mu \leq R_2(b+\mathbf{1})-\mathbf{1}] \Rightarrow L\mu \leq b$ has been proved in [10].

It is known that a sufficient condition for the c-admissibility of a set of constraints $L\mu \leq b$ is that $LD_{uc} \leq 0$ and $LD_{uo} = 0$, where D_{uc} and D_{uo} are the restrictions of the incidence matrix D to the sets of uncontrollable and unobservable transitions [10]. The admissibility requirements in our setting can then be written as

$$L_i D_{uc}^{(i)} \leq 0 \tag{10}$$
$$L_i D_{uo}^{(i)} = 0 \tag{11}$$

where $D_{uc}^{(i)}$ and $D_{uo}^{(i)}$ are the restrictions of D to the sets $T_{uc}^{(i)} = \bigcap_{i \in \mathcal{C}_i} T_{uc,i}$ and $T_{uo}^{(i)} = \bigcup_{i \in \mathcal{C}_i} T_{uo,i}$. Then our problem becomes: *find a feasible solution of (5) and (8–11).* The unknowns are R_1, R_2, L_i, and b_i, and integer programming can be used to find them. The next result is an immediate consequence of our considerations above.

Proposition 2.1 *Any sets of constraints $L_i\mu \leq b_i$ satisfying (5) and (8–11) are d-admissible and* $\bigwedge_{i=1\ldots n} [L_i\mu \leq b_i] \Rightarrow L\mu \leq b$.

Note that this approach is purely structural, as it does not take advantage of knowledge on the reachable markings. Such knowledge can be used to relax the conditions (10) and (11) (see the appendix of [5]).

2.3 Decentralized control with communication

So far, we have ignored the possibility that the local supervisors $\mathcal{S}_i$ may have the ability to communicate. We now consider the case in which the supervisors are able to communicate the firings of certain transitions. Communication is useful, as it relaxes the admissibility constraints (10) and (11) by reducing the number of uncontrollable and unobservable transitions. However, communication constraints may be present, and bandwidth limitations may encourage the minimization of the communication over the network. The analysis of this section, without being comprehensive, serves as an illustration of the fact that such problems can be approached in this framework.

Let's denote by t_j the transition whose connections appear in the j-th column of D. For each set $\mathcal{C}_i$ and transition t_j, let α_{ij} be a binary variable, where $\alpha_{ij} = 1$ if the firing of t_j is made known to the subsystems in $\mathcal{C}_i$. To simplify our presentation, assume that the firing of a transition is broadcasted over the whole network. Then we can take $\alpha_{ij} = \alpha_j$. Note that we have the following constraints:

$$\forall t_j \in T_{uo,L} : \alpha_j = 0 \tag{12}$$

where $T_{uo,L} = \bigcap_{i=1\ldots n} T_{uo,i}$ is the set of transitions that cannot be observed anywhere in the system. ($T_{uo,L}$ is the set of transitions whose firing cannot be communicated.)

Let $B_{o,L}^i$ and $B_{o,U}^i$ be lower and upper bounds of $L_i D_{uo}^{(i)}$, and α be the vector of elements α_i. Furthermore, let A_o^i be a diagonal matrix in which the diagonal is α restricted to the transitions of $T_{uo}^{(i)}$. We require

$$L_i D_{uo}^{(i)} \leq B_{o,U}^i A_o^i \tag{13}$$
$$L_i D_{uo}^{(i)} \geq B_{o,L}^i A_o^i \tag{14}$$

instead of $L_iD_{uo}^{(i)} = 0$. In this way, the admissibility requirement $L_iD_{uo}^{(i)} = 0$ is relaxed by eliminating the constraints corresponding to the transitions of $T_{uo}^{(i)}$ that are broadcasted.

Similarly, (10) can also be relaxed by communicating enabling decisions of supervisors. Naturally, for each transition t it controls, each supervisor $\mathcal{S}_i$ has two enabling decisions: *enable* and *disable*. They depend on whether all control places C of $\mathcal{S}_i$ that are connected to t satisfy $\mu_c(C) \geq W_s(C,t)$ or not. A communication policy may be that a supervisor announces a remote actuator each time its enabling decision changes. Then the actuator can determine its enabling by taking the conjunction of the decisions corresponding to all supervisors controlling it. In our setting, d-admissibility implies that the supervisors within a cluster $\mathcal{C}_i$ have always the same enabling decisions, and so only communication between clusters needs to be considered. Similarly to α_{ij}, we can introduce binary variables ε_{ij} describing the communication of enabling decisions pertaining to t_j. Thus, $\varepsilon_{ij} = 1$ if a supervisor from $\mathcal{C}_i$ communicates its enabling decisions to t_j. Then (10) becomes:

$$L_iD_{uc}^{(i)} \leq B_{c,U}^i A_c^i \tag{15}$$

where $B_{c,U}^i$ is the upper bound of $L_iD_{uc}^{(i)}$, A_c^i is a diagonal matrix of diagonal equal to the restriction of e_i to the transitions of $T_{uc}^{(i)}$, and e_i is the vector of elements ε_{ij} for $j = 1 \dots |T|$.

Communication constraints stating that certain transitions cannot be observed by communication or that certain transitions cannot be remotely controlled by communication, can be incorporated by setting coefficients α_i and ε_{ij} to zero. Constraints limiting the average network traffic can be incorporated as constraints of the form:

$$g\alpha + \sum_i h_i e_i \leq p \tag{16}$$

where g and h_i are vectors of appropriate dimensions and p is a scalar. As an example, the elements of g could reflect average firing counts of the transitions over the operation of the system.

We may also choose to minimize the amount of communication involved in the system. Then we can formulate our problem as

$$\min_{L_i,b_i,e_i,\alpha,R_1,R_2} c\alpha + \sum_i f_i e_i \tag{17}$$

subject to the constraints (5), (8–9), (12–15), and $\alpha, e_i \in \{0,1\}^{|T|}$. This problem can be solved using integer linear programming.

Obviously, an integer programming approach limits the size of the problems that can be solved. We propose ways to reduce the computational effort in the technical report [5].

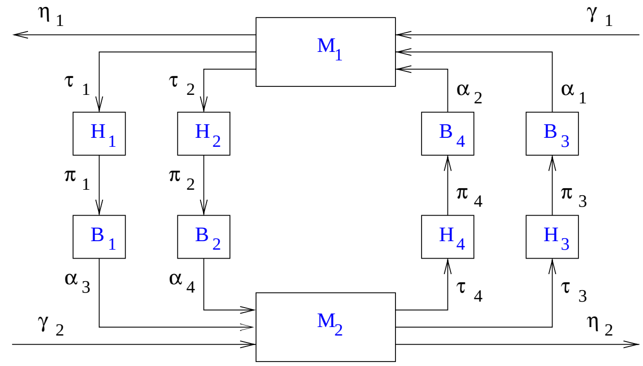

Figure 2: A manufacturing system.

2.4 Liveness Constraints

One of the difficulties encountered with this approach is that the permissivity of the generated constraints is hard or impossible to be expressed in the cost function. Moreover, the generated constraints may cause parts of the system to unavoidably deadlock. This situation can be prevented by using a special kind of constraints, that we call liveness constraints.

A liveness constraint consists of a vector x such that for all i: $L_ix \leq 0$. A possible way to obtain such constraints is described next. Given a finite firing sequence σ, let x_σ be a vector such that $x_\sigma(i)$ is the number of occurrences of the transition t_i in σ. Given the Petri net of incidence matrix D and the constraints $L\mu \leq b$, let y be a nonnegative integer vector such that $Dy \geq 0$ and $-LDy \geq 0$. A vector y satisfying these inequalities has the following property. If σ is a firing sequence such that (a) σ can be fired without violating $L\mu \leq b$ and (b) $x_\sigma = y$, then σ can be fired infinitely often without violating $L\mu \leq b$. However, if the decentralized control algorithm generates a constraint $L_i\mu \leq b_i$ such that $L_iDy \not\leq 0$, then any firing sequence σ having $x_\sigma = y$ cannot be infinitely often fired in the closed-loop. If such a situation is undesirable, the matrices L_i can be required to satisfy $L_ix \leq 0$ for $x = Dy$. An illustration will be given in the next section.

3 Example

This section illustrates our approach on the manufacturing example from [9], shown in Figure 2. The system consists of two machines (M_1 and M_2), four robots (H_1 … H_4), and four buffers of finite capacity (B_1 … B_4). The events associated with the movement of the parts within the system are marked with Greek letters. There are two types of parts. The manufacturing process of the first type of parts is represented by the following sequence of events: $\gamma_1\tau_1\pi_1\alpha_3\tau_3\pi_3\alpha_1\eta_1$. The manufacturing process of the second kind of parts is represented by $\gamma_2\tau_4\pi_4\alpha_2\tau_2\pi_2\alpha_4\eta_2$. These processes can be represented by the Petri net of Figure 3. In

Marian Iordache, Panos J.Antsaklis, “Decentralized Control of Petri Nets with Constraint Transformations,”
Proceedings of the 2003 American Control Conference, pp. 314-319, Denver, Colorado, June 4-6, 2003.

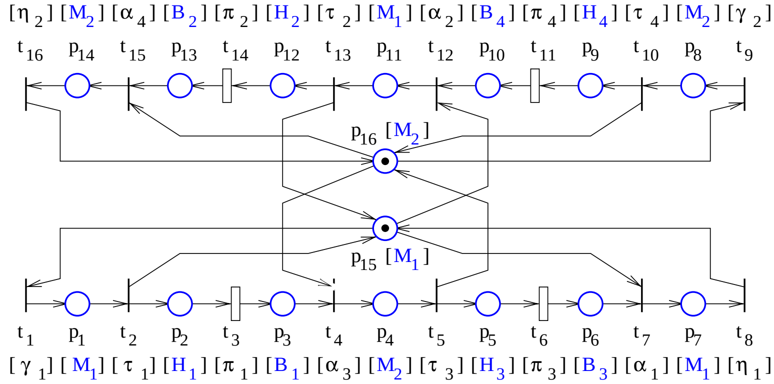

Figure 3: Petri net model of the system.

the Petri net, the transitions are labeled by the events they represent, and the places by the names of the manufacturing components. For instance, a token in p_{16} indicates that M_2 is idle, and a token in p_8 indicates that M_2 is working on a part of type 2 that has just entered the system. Furthermore, the number of parts in a buffer is the marking of the place modeling the buffer; for instance, μ_{13} represents the number of parts in B_2 at the marking μ. The number of parts the machines M_1 and M_2 can process at the same time is $\mu_1 + \mu_7 + \mu_{11} + \mu_{15} = n_1$ and $\mu_4 + \mu_8 + \mu_{14} + \mu_{16} = n_2$, respectively. In [9], $n_1 = n_2 = 1$.

The first supervisory requirements are that the buffers do not overflow. If the capacity of the buffers is k, the requirement can be written as:

$$\mu_i \leq k \text{ for } i \in \{3, 6, 10, 13\} \tag{18}$$

In [9] the capacity of the buffers is $k = 2$. Another requirement is that the number of completed parts of type 1 is about the same as the number of completed parts of type 2:

$$v_8 - v_{16} \leq u \tag{19}$$
$$v_{16} - v_8 \leq u \tag{20}$$

where v_8 and v_{16} denote the number of firings of t_8 and t_{16}, respectively. In [9], $u = 2$. Note that constraints involving the vector v can be easily represented as marking constraints in a transformed Petri net [6].

Following [9], the constraints (18) are enforced assuming that the system consists of the subsystems: $T_{c,1} = \{t_2, t_4\}$ and $T_{o,1} = \{t_2, t_3, t_4\}$, $T_{c,2} = \{t_5, t_7\}$ and $T_{o,2} = \{t_5, t_6, t_7\}$, $T_{c,3} = \{t_{10}, t_{12}\}$ and $T_{o,3} = \{t_{10}, t_{11}, t_{12}\}$, $T_{c,4} = \{t_{13}, t_{15}\}$ and $T_{o,4} = \{t_{13}, t_{14}, t_{15}\}$. We take $\mathcal{C}_i = \{i\}$ for $i = 1 \ldots 4$. Enforcing (18) results in the control places C_1, C_2, C_3, and C_4 shown in Figure 4. They correspond to the subsystems 1,2,3 and 4, respectively, and enforce $\mu_2 + \mu_3 \leq 2$, $\mu_5 + \mu_6 \leq 2$, $\mu_9 + \mu_{10} \leq 2$, and $\mu_{12} + \mu_{13} \leq 2$.

In [9], (19–20) are assumed to be enforced at a higher

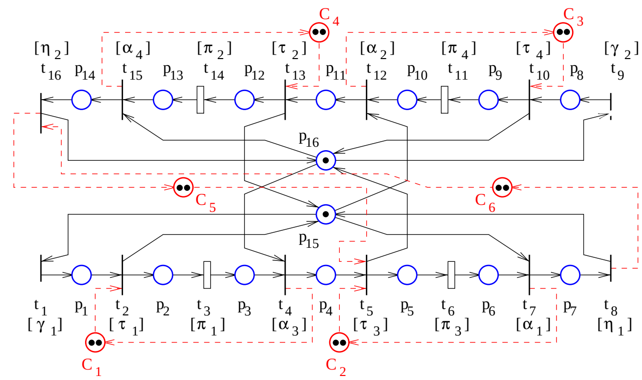

Figure 4: Decentralized supervision.

hierarchical level at which all transitions are observable and controllable, except for t_7 and t_8 which are uncontrollable due to communication problems. In this context, (19) is inadmissible and (20) is admissible. In our computer implementation of the decentralized algorithms, enforcing (19) results in the constraint $\mu_5 + \mu_6 + \mu_7 + v_8 \leq 2$. This is clearly an unacceptable constraint, as it causes the closed-loop to unavoidably deadlock. As discussed in section 2.4, a remedy is to add liveness constraints. So we added the liveness constraint $L_i x \leq 0$ for $x = Dy$ and $y = [1, 1, \ldots 1]^T$. This is to prevent the constraints generated by the algorithm from blocking the firing sequence $t_1 t_2 \ldots t_{16}$ to occur infinitely often. Then, the generated constraints for (19–20) are

$$\mu_5 + \mu_6 + \mu_7 + v_8 - v_{16} \leq 2 \tag{21}$$
$$v_{16} - v_8 \leq 2 \tag{22}$$

They are enforced by the control places C_5 and C_6 in Figure 4. Note that compared to the solution of [9], our solution is equivalent. However, in our case the supervisor can be reused for other values of n_1, n_2, k and u, by changing accordingly the initial markings of $C_1 \ldots C_6$.

Assuming that the higher level supervisor implementing C_5 and C_6 uses direct links to access each transition, the communication cost depends only on the number of links, that is, the number of transitions it controls and/or observes. Figure 4 shows that the communication between the plant and C_5 and C_6 involves t_5, t_8 and t_{16}. Is three the minimal number of transitions? While our approach is suboptimal, the minimization it employs could be used to find a solution with communication that involves less transitions. To do so, we could attempt to design a supervisor for (19-20) minimizing communication. The setting is as follows. At the higher level, no transition is controllable or observable. Communication can make all transitions observable and controllable, except for the transitions of $T_{uc,L} = \{t_3, t_6, t_{11}, t_{14}, t_7, t_8\}$, which cannot be controlled. With the notation of section 2.3, we have $c = \mathbf{1}^T$, $f_i = \mathbf{0}^T$ (here $i = 1$), and we constrain ε_{ij} to $\varepsilon_{ij} = 0\ \forall t_j \in T_{uc,L}$ and $\varepsilon_{ij} = \alpha_j\ \forall t_j \notin T_{uc,L}$. By solving the integer program the following constraints were obtained

$$\begin{aligned} \mu_1 + \mu_2 + \mu_3 + 2\mu_4 + \mu_5 + \mu_6 + & \\ + \mu_7 + \mu_8 + \mu_{16} + v_8 - v_{16} &\leq 2 \qquad (23) \\ \mu_{14} + v_{16} - v_8 &\leq 2 \qquad (24) \end{aligned}$$

which are enforced by control places C_5' and C_6' such that $C_5'\bullet = \{t_1\}$, $\bullet C_5' = \{t_{15}\}$, $C_6'\bullet = \{t_{15}\}$ and $\bullet C_6' = \{t_8\}$. In this solution, the communication involves t_1, t_8 and t_{15}. While this solution is obviously less permissive than that of (21–22), it shows that our approach cannot find a solution involving the communication of less transitions. In this sense, (21–22) are optimal. Finally, (23–24) illustrate once more that the permissivity of the solutions is hard to control. However, in this particular case, a second integer program can be used to improve the permissivity, by minimizing the sum of the positive coefficients in (23–24), while requiring the other coefficients to stay less or equal to zero (the integer program is also subject to the constraints of the previous integer program and to $\sum \alpha_i = 3$, which constrains the communication cost to the minimum value).

4 Final Remarks

This paper has introduced a simple linear integer programming technique for the design of decentralized supervisors of Petri nets. This technique transforms a global specification into local specifications that can be implemented by local supervisors. In the case when communication between the local supervisors is possible, both the supervisors and their communication policy is designed. In particular, the supervisors can be designed to minimize their communication. The approach is suboptimal, as it may not produce the least restrictive solution, when it exists. Future work may explore possibilities to increase the computational efficiency of the supervisor design. Another direction to be considered is decentralized deadlock prevention, to avoid the deadlock possibilities existing in the system and those caused by supervision.

References

[1] G. Barrett and S. Lafortune. Decentralized supervisory control with communicating controllers. *IEEE Trans. Aut. Contr.*, 45(9):1620–1638, 2000.

[2] R. Boel and J. van Schuppen. Decentralized failure diagnosis with costly communication between diagnosers. *Proc. WODES'02*, pp. 175–181, 2002.

[3] R. Cieslak, C. Desclaux, A. Fawaz, P. Varayia. Supervisory control of discrete-event processes with partial observations. *IEEE Trans. Aut. Contr.*, 33(3):249–260, 1988.

[4] A. Giua, F. DiCesare, and M. Silva. Generalized mutual exclusion constraints on nets with uncontrollable transitions. In *Proc. IEEE Conf. Syst., Man, Cybern.*, pp. 974–979, 1992.

[5] M. V. Iordache and P. J. Antsaklis. Decentralized control of DES using Petri nets. Technical report isis-2002-005, University of Notre Dame, 2002. Available at www.nd.edu/~isis/tech.html.

[6] M. V. Iordache and P. J. Antsaklis. Synthesis of supervisors enforcing general linear vector constraints in Petri nets. In *Proc. 2002 ACC.*, pp. 154–159, 2002.

[7] M. V. Iordache and P. J. Antsaklis. Admissible decentralized control of Petri nets. In *Proc. 2003 ACC.*

[8] Y. Li and W. Wonham. Control of Vector Discrete-Event Systems II - Controller Synthesis. *IEEE Trans. Aut. Contr.*, 39(3):512–530, March 1994.

[9] F. Lin and W. Wonham. Decentralized control and coordination of discrete-event systems with partial observation. *IEEE Trans. Aut. Contr.*, 35(12):1330–1337, 1990.

[10] J. O. Moody and P. J. Antsaklis. Petri net supervisors for DES with uncontrollable and unobservable transitions. *IEEE Trans. Automat. Contr.*, 45(3):462–476, 2000.

[11] K. Rudie and W. Wonham. Think globally, act locally: Decentralized supervisory control. *IEEE Trans. Automat. Contr.*, 37(11):1692–1708, 1992.

[12] K. Rudie, S. Lafortune, and F. Lin. Minimal Communication in a Distributed Discrete-Event System In *Proc. 1999 ACC.*, pp. 1965–1970.

[13] K. Rudie. The current state of decentralized discrete-event control systems. In *Proc. of the 10th IEEE Mediterranean Conf. Contr. Automat.*, 2002.

[14] G. Stremersch. *Supervision of Petri Nets.* Kluwer Academic Publishers, 2001.

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

[16] E. Yamalidou, J. O. Moody, P. J. Antsaklis, and M. D. Lemmon. Feedback control of Petri nets based on place invariants. *Automatica*, 32(1):15–28, January 1996.