12th IFAC/IEEE Workshop on Discrete Event Systems
Cachan, France. May 14-16, 2014

Decentralized monitors design for Petri net models

Francesco Basile * **Roberto Cordone** ** **Luigi Piroddi** ***

* *Dip. di Ingegneria dell'Informazione, Ingegneria elettrica e Matematica applicata, Università di Salerno, Italy.* email: fbasile@unisa.it.
** *Dipartimento di Informatica, Università degli Studi di Milano, Milano, Italy.* email: roberto.cordone@unimi.it.
*** *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy.* email: luigi.piroddi@polimi.it.

Abstract: The problem of designing an optimal decentralized supervisor that enforces static (*e.g.*, job and resource bounds) and behavioral (*e.g.*, liveness, reversibility and controllability) constraints simultaneously on a Petri net model is here addressed. The supervisor consists of multiple local controllers assigned to different control sites, such that each control site can operate on a subset of the net transitions. A transition can be employed by multiple sites, but is not necessarily controllable by all of them. The key elements of the approach are an integer linear programming formulation that finds the decentralized supervisor that maximizes the number of allowed states among a subset of those that would be allowed by a global supervisor and a branch & bound procedure on the state set that ultimately guarantees the maximal permissiveness of the solution.

Keywords: Petri Nets, Supervisory Control, Monitor places, Decentralized control.

1. INTRODUCTION

Supervisory control (SC) deals with the problem of designing an agent (the *supervisor*) that allows only a given subset $\mathcal{L}$ (the *legal* set) of the reachable states of a discrete event system (DES). In the context of Petri nets (PNs) such supervisor is often formulated in terms of linear inequalities on the states, called Generalized Mutual Exclusion Constraints (GMECs), such that any legal state satisfies them and any other reachable state violates at least one of them (see, *e.g.*, Giua et al. (1992); Yamalidou et al. (1996)). Such indirect GMEC-based formulation of the legal set is particularly amenable to control analysis and synthesis, since the supervisor itself can be implemented as a PN consisting of places (*monitors*), suitably connected to the transitions of the PN model of the plant. Various specifications that must typically be enforced on an (open-loop) PN model can be formulated as a set of GMECs. These specifications can be divided into *static* (*e.g.*, bounds on job and resource usage) and *behavioral* (*e.g.*, deadlock prevention (DP), liveness enforcement (LE), reversibility, controllability), the former depending only on the reachable set, whereas the latter can be established only by analyzing the reachability graph of the PN.

As discussed in (Basile et al., 2013b), the separate enforcement of control specifications - as for example, in (Reveliotis and Choi, 2006) reversibility is considered - in subsequent steps is problematic for two reasons. Notice first that the enforcement of a specification determines a contraction of the reachable space, which may also result in the loss of a behavioral property enforced previously. This issue is particularly relevant if multiple behavioral properties are desired. The second reason is the risk of obtaining redundant GMECs in the overall supervisor. In the mentioned work this issue is addressed by compacting the enforcement of all required specifications in a single step, consisting in the design of the optimal GMEC-based supervisor that separates the legal set $\mathcal{L}$ of all reachable states that abide by all required specifications from the illegal one, denoted $\mathcal{U}$.

A first extension of this problem to a decentralized setting is discussed in (Basile et al., 2013a), where multiple control sites can be employed in the design, each operating on a subset of the PN transitions. The supervisor consists of multiple local controllers, each associated to a different control site. Such a control architecture, where a central coordinator is absent and local supervisors are isolated, can address situations where it is not possible to communicate with all plant sensors or actuators, due to economic reasons or bandwidth limitations (a particularly relevant issue in large scale systems deployed on a wide geographic area and involving a large number of devices).

Decentralized control problems have been more often studied in the context of formal languages and automata (Barret and Lafortune, 2000; Lin and Wonham, 1990; Rudie and Wonham, 1992), while fewer works deal with PN-based formulations (see, *e.g.*, (Guan and Holloway, 1997; Chen and Hu, 1991; Iordache and Antsaklis, 2006)). Specifically, (Iordache and Antsaklis, 2006) provides a sufficient condition for a set of GMECs to be enforced in a decentralized setting (denoted d-admissibility). In (Basile et al., 2013a), the focus is instead on finding a decentralized supervisor (if one exists) that preserves as many (globally) legal states as possible. The main difficulty lies in the fact that the states allowed by a decentralized supervisor do not necessarily configure a reachability subgraph compatible with all the given behavioral requirements. Conversely, a given legal set $\mathcal{L}$ that fully satisfies the given static and behavioral requirements may not be fully enforceable by a decentralized supervisor. This difficulty is overcome by

adopting a proposal-acceptance mechanism, where a candidate legal set $\mathcal{L}$ (by construction, included in or equal to the set of legal states that can be allowed by a global supervisor), is first selected so as to guarantee the obtainment of all the desired static and behavioral requirements, and then tested for the existence of a decentralized supervisor that can exactly enforce it. A B&B algorithm generates the candidate legal sets by subsequent reductions of the global legal state set, guaranteeing a full exploration of its subsets. The procedure proposed in this paper exploits a similar proposal-acceptance mechanism, adapted so as to deal with controllability in a decentralized setting from a *behavioral* point of view.

2. PRELIMINARIES

2.1 Petri net basics

A marked PN (Murata, 1989) is a 5-tuple $N = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}, \boldsymbol{m}_0\rangle$, where P and T are the (finite and nonempty) sets of places and transitions, respectively, with $|P| = n_p$, $|T| = n_t$, and $P \cap T = \emptyset$, $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{N}^{n_p \times n_t}$ are the input and output matrices, and $\boldsymbol{m}_0 \in \mathbb{N}^{n_p}$ is the (initial) marking vector, $\mathbb{N}$ being the set of nonnegative integers. $\boldsymbol{Pre}$ and $\boldsymbol{Post}$ represent the topology of the PN in terms of the connections between places and transitions. More precisely, $Pre_{k,j}$ [$Post_{k,j}$] is the weight of an arc going from p_k [t_j] to t_j [p_k] (it equals 0 if there is no such arc). The incidence matrix $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ provides an equivalent information in the absence of self-loops. The marking vector $\boldsymbol{m}$ defines the distribution of tokens in places.

A transition $t_j \in T$ is enabled in a marking $\boldsymbol{m}$ (denoted $\boldsymbol{m}[t_j\rangle$) iff $\boldsymbol{m} \geq \boldsymbol{Pre}\,\boldsymbol{e}_j$, where $\boldsymbol{e}_j$ is the jth versor of the $\mathbb{R}^{n_t}$ coordinate space. If $\boldsymbol{m}[t_j\rangle$, then t_j may fire at marking $\boldsymbol{m}$, yielding the marking $\boldsymbol{m}' = \boldsymbol{m} + \boldsymbol{C}\boldsymbol{e}_j$ (denoted $\boldsymbol{m}[t_j\rangle\boldsymbol{m}'$). The set of markings reachable from $\boldsymbol{m}_0$ by way of enabled transition sequences is the reachability set, denoted $R(N, \boldsymbol{m}_0)$. The reachability graph is a directed graph $RG = (V, A)$, where $V = R(N, \boldsymbol{m}_0)$ is the set of nodes and $A \subseteq (V \times V)$ the set of arcs, associated to the PN transitions through a labeling function $h : A \to T$.

A strongly connected component (SCC) of a directed graph is a maximal subgraph, such that any two of its nodes are connected by a directed path. An SCC may consist of a single vertex, if that vertex does not belong to any directed cycle. Let (S, A_S) be an SCC of RG. Then, if $|S| \geq 2$ the PN can evolve inside S for an arbitrary number of transition firings. (S, A_S) is a *terminal* SCC if there does not exist any $(\boldsymbol{m}_1, \boldsymbol{m}_2) \in A$ with $\boldsymbol{m}_1 \in S$ and $\boldsymbol{m}_2 \in V \setminus S$.

A place $p_i \in P$ is bounded iff $\exists k > 0$ s.t. $m_i \leq k$, $\forall \boldsymbol{m} \in R(N, \boldsymbol{m}_0)$. A PN is bounded iff all its places are bounded. A transition $t_j \in T$ is live iff $\forall \boldsymbol{m} \in R(N, \boldsymbol{m}_0)$, $\exists \boldsymbol{m}' \in R(N, \boldsymbol{m})$ s.t. $\boldsymbol{m}'[t_j\rangle$. N is live iff all its transitions are live. N is reversible iff $\boldsymbol{m}_0 \in R(N, \boldsymbol{m})$, $\forall \boldsymbol{m} \in R(N, \boldsymbol{m}_0)$. A marking $\boldsymbol{m} \in R(N, \boldsymbol{m}_0)$, is dead if $\nexists t_j \in T$ enabled in $\boldsymbol{m}$ (total deadlock state). It also configures a terminal SCC with a single vertex. A deadlock-free PN has no reachable dead markings.

A PN is (Fumagalli et al., 2010): i) deadlock-free if all the terminal SCCs of RG have cardinality strictly greater than 1, ii) live if for any terminal SCC (S, A_S) of RG it holds that $|S| \geq 2$ and $\{t | t = h(a), \forall a \in A_S\} = T$, and iii) reversible if RG consists of a single SCC.

2.2 GMEC enforcement by means of monitors

A GMEC is a linear marking inequality $\boldsymbol{l}\boldsymbol{m} \leq b$, with $\boldsymbol{l} \in \mathbb{N}^{n_p}$, $b \in \mathbb{N}$, completely defined by the pair $(\boldsymbol{l}, b)$, and associated to the *admissibility region* $\mathcal{M}(\boldsymbol{l}, b) = \{\boldsymbol{x} \in \mathbb{N}^{n_p} \mid \boldsymbol{l}^T\boldsymbol{x} \leq b\}$. The admissibility region of a set of GMECs $(\boldsymbol{L}, \boldsymbol{b})$, with $\boldsymbol{L} = [\boldsymbol{l}_1^T\ \boldsymbol{l}_2^T\ \ldots\ \boldsymbol{l}_{n_c}^T]^T$ and $\boldsymbol{b} = [b_1\ b_2\ \ldots\ b_{n_c}]^T$, is obtained as $\mathcal{M}(\boldsymbol{L}, \boldsymbol{b}) = \cap_{i=1}^{n_c}\mathcal{M}(\boldsymbol{l}_i, b_i)$. Let $\boldsymbol{m}_0 \in \mathcal{M}(\boldsymbol{L}, \boldsymbol{b})$. Then, a supervisor consisting of n_c monitor places connected to the existing PN transitions by way of the incidence matrix $\boldsymbol{C}_C = -\boldsymbol{L}\boldsymbol{C}$ and marked according to $\boldsymbol{m}_{C0} = \boldsymbol{b} - \boldsymbol{L}\boldsymbol{m}_0$ enforces the said constraints (Giua et al., 1992; Yamalidou et al., 1996). The designed controller prevents only transition firings leading to a violation of one of the GMECs, and is thus maximally permissive. Let $\{\boldsymbol{m}_0\} \subseteq \mathcal{L} \subseteq R(N, \boldsymbol{m}_0)$ and assume that there exists a set of GMECs $(\boldsymbol{L}, \boldsymbol{b})$ such that $\mathcal{L} \subseteq \mathcal{M}(\boldsymbol{L}, \boldsymbol{b})$ and $\mathcal{M}(\boldsymbol{L}, \boldsymbol{b})$ does not contain any other reachable marking. Then, a supervisor implementing $(\boldsymbol{L}, \boldsymbol{b})$ exactly enforces $\mathcal{L}$.

The problem of restricting the reachability set of a PN within a set of legal markings $\mathcal{L}$ becomes somewhat more involved in the presence of uncontrollable transitions. In the following, we assume that $T = T_c \cup T_{uc}$ with $T_c \cap T_{uc} = \emptyset$, where T_{uc} is the set of uncontrollable transitions (represented as black bars), and T_c is the set of controllable transitions (represented as white bars), associated to uncontrollable and controllable events, respectively.

***Definition** 2.1.* Consider a PN N with $T_c \neq \emptyset$. The sub-net N_u obtained from N eliminating every transition in T_c is denoted *uncontrollable sub-net of* N. ■

It is immediate to see that $R(N_u, \boldsymbol{m}) \subseteq R(N, \boldsymbol{m})$.

***Definition** 2.2.* A legal marking set $\mathcal{L} \subseteq \mathbb{N}^{n_p}$ is *behaviorally controllable* w.r.t. a marked PN N with initial marking $\boldsymbol{m}_0$ if $\bigcup_{\boldsymbol{m}\in\mathcal{L}} R(N_u, \boldsymbol{m}) \subseteq \mathcal{L}$, where N_u is the uncontrollable sub-net of N. ■

In other words, $\mathcal{L}$ is controllable if no forbidden marking is reachable from any marking $\boldsymbol{m} \in \mathcal{L}$ by firing a sequence containing only uncontrollable transitions.

A transition t enabled under the net marking can be disabled by way of a PN supervisor only if there is an arc from a control place to t and the control place is insufficiently marked. Therefore, to enforce a behaviorally controllable legal marking set by means of a PN controller, an arc directed from a control place to an uncontrollable transition must be avoided if there exists a reachable marking where the control place alone disables the transition, which would otherwise be enabled by way of the plant marking.

2.3 GMEC optimization as a classification problem

A set of GMECs $(\boldsymbol{L}, \boldsymbol{b})$ configures a *linear classifier*, separating the markings in $\mathcal{M}(\boldsymbol{L}, \boldsymbol{b})$ from those outside. An illegal set $\mathcal{U}$ can be separated by a legal set $\mathcal{L}$ by a classifier of this class if the following condition holds:

***Theorem** 2.3.* (Cordone et al., 2012) Let $\mathcal{L}$ and $\mathcal{U}$ be two given (disjoint) marking sets. Then, there exists a linear classifier separating $\mathcal{U}$ from $\mathcal{L}$ *iff* there does not exist a marking $\boldsymbol{m} \in \mathcal{U}$ such that $\boldsymbol{m} \in P_{\mathcal{L}}$, where $P_{\mathcal{L}}$ is the convex hull of $\mathcal{L}$. ■

The maximal legal set $\mathcal{L}$ of states that guarantees the desired static and behavioral properties, as well as the corresponding

minimal set $\mathcal{U}$, can be determined as explained in (Cordone and Piroddi, 2011, 2013). Then, the optimal linear classifier separating $\mathcal{U}$ from $\mathcal{L}$ is obtained by searching for an optimal covering of the illegal set $\mathcal{U}$ with suitable subsets $\mathcal{U}_i$, $i = 1, \ldots, n_c$, such that for each subset there exists a GMEC that separates it from $\mathcal{L}$. All feasible coverings of the illegal set $\mathcal{U}$ can be systematically explored with the B&B method explained in (Cordone and Piroddi, 2011, 2013).

3. THE PROPOSED METHOD

3.1 BC- and DC-feasibility

Let T_i, $i = 1, \ldots, \nu$, be ν subsets of the set of transitions T, identifying ν control sites S_i such that any local supervisor is allowed to act only on the transitions of one site. Subsets T_i, $i = 1, \ldots, \nu$, do not necessarily form a partition nor a covering of T. Let also $T_i = T_{c_i} \cup T_{uc_i}$, $i = 1, \ldots, \nu$, with $T_{c_i} \cap T_{uc_i} = \emptyset$, where T_{c_i} collects all transitions associated to events whose firing can be detected and disabled from site S_i, while transitions associated to events whose firing can be only detected from site S_i (but not disabled) are collected in T_{uc_i}.

Let $\mathcal{L}$ be the maximal set of markings compatible with all the static and behavioral requirements of interest and realizable with a global supervisor (denoted *legal set* in the sequel). We assume that the static requirements include boundedness, so that $\mathcal{L}$ is a bounded set, and that the behavioral requirements include liveness, reversibility and controllability. By "global" supervisor we here denote a controller that can operate on the set of transitions $\cup_{i=1}^{\nu} T_i$ (all observable), where only the transitions in $T_c = \cup_{i=1}^{\nu} T_{c_i}$ are controllable. Note that some globally controllable transitions may become *partially* controllable in the decentralized setting, *i.e.* controllable only by a fraction of the sites which have access to them. The mentioned set $\mathcal{L}$ can be determined exactly following the approach described in (Basile et al., 2013b). Let also $\mathcal{U}$ be the corresponding set of boundary illegal states (states outside $\mathcal{L}$ that can be reached from legal states with a single transition firing), briefly referred to as *illegal set*. The boundedness of $\mathcal{L}$ automatically implies that of $\mathcal{U}$ (Basile et al., 2013b).

In the decentralized setting, only a subset $\overline{\mathcal{L}} \subseteq \mathcal{L}$ of the legal markings will generally be allowed. While this would still guarantee the obtainment of all the static specifications, that depend only on the individual states, the desired behavioral properties could be lost due to the contraction of the reachable space (which implies a modification of the behavioral characteristics of the system, as described by the reachability graph). For this reason, we will introduce the notions of BC- and DC-feasibility. Specifically, we will denote $\overline{\mathcal{L}}$ as:

(i) *BC-feasible* (behaviorally feasible) if the reachability subgraph induced by $\overline{\mathcal{L}}$ on the PN possesses all required behavioral properties (liveness, reversibility and global controllability);
(ii) *DC-feasible* (decentralization-wise feasible) if $\overline{\mathcal{L}}$ can be exactly enforced by a supervisor that abides by the decentralization constraints and such that a transition t is never disabled exclusively by monitor places belonging to control sites from which t is uncontrollable (for local controllability).

Only legal sets that are simultaneously BC- and DC-feasible result in admissible supervisors for the problem at hand. The notions of BC- and DC-feasibility extend the homologous B- and D-feasibility properties introduced in (Basile et al., 2013a) to include behavioral controllability (in (Basile et al., 2013a) only the more restrictive structural controllability condition is considered). Notice in this respect that controllability cannot be ensured in the decentralization framework by analysis of the reachability graph alone, but requires additional conditions on the structure of the supervisor, thus entering in both the BC- and DC-feasibility definitions. Such conditions do not prevent the use of arcs from monitor places to transitions that they cannot control, as long the disabling of such transitions does not occur exclusively due to an insufficient marking of such places (behavioral controllability).

***Definition** 3.1.* A set $\overline{\mathcal{L}}$ such that $\{\boldsymbol{m}_0\} \subset \overline{\mathcal{L}} \subseteq \mathcal{L}$ is denoted *BC-feasible* if a GMEC-based supervisor enforcing exactly $\overline{\mathcal{L}}$ yields a live, reversible and controllable PN. ■

Let $\overline{\mathcal{L}}_b = \{\boldsymbol{m} \in \overline{\mathcal{L}} \mid \boldsymbol{m}[t > \boldsymbol{m}', \boldsymbol{m}' \notin \overline{\mathcal{L}}, t \in T_c\}$ denote the set of *boundary* legal markings, *i.e.* the legal markings from which the illegal set can be reached in a single transition step. Notice that the described marking evolution can only occur through the firing of a controllable transition, otherwise $\boldsymbol{m}$ would not be legal. By selectively forbidding the mentioned controllable transitions enabled in markings belonging to $\overline{\mathcal{L}}_b$, no illegal marking will ever be reached. For each boundary legal marking $\boldsymbol{m} \in \overline{\mathcal{L}}_b$, let $\mathcal{D}(\boldsymbol{m}) = \{t \in T_c \mid \boldsymbol{m}[t > \boldsymbol{m}', \boldsymbol{m}' \notin \overline{\mathcal{L}}\}$ be the set of controllable transitions that must be disabled in $\boldsymbol{m}$. Finally, let $\Phi \in \{0,1\}^{\nu \times n_t}$ be a binary function defining the controllability of transitions from a certain site, *i.e.* such that $\Phi(i,t) = 1$ if t is controllable from site i, and 0 otherwise. Let also $k : \{1, \ldots, n_{dc}\} \to \{1, \ldots, \nu\}$ map the individual GMECs to the control sites ($k(p_c) = i$ indicates that the p_cth GMEC operates on site S_i, *i.e.* the corresponding monitor is connected only to transitions in T_i).

***Definition** 3.2.* Let $\{\boldsymbol{m}_0\} \subseteq \overline{\mathcal{L}} \subseteq \mathcal{L}$ and assume that there exists a set of GMECs $(\boldsymbol{L}, \boldsymbol{b})$, with $\boldsymbol{L} \in \mathbb{N}^{n_{dc} \times n_p}$ and $\boldsymbol{b} \in \mathbb{N}^{n_{dc}}$, that exactly enforces $\overline{\mathcal{L}}$. Set $\overline{\mathcal{L}}$ is denoted *DC-feasible* iff

(i) For each $p_c = 1, \ldots, n_{dc}$, there exists a control site S_i such that $\sum_{p=1}^{n_p} L(p_c, p) C(p,t) = 0$ for all $t \notin T_i$;
(ii) For each pair $(\boldsymbol{m}, t)$ with $\boldsymbol{m} \in \overline{\mathcal{L}}_b$ and $t \in \mathcal{D}(\boldsymbol{m})$ s.t. $\sum_{i=1}^{\nu} \Phi(i,t) \geq 1$, $(\boldsymbol{L}, \boldsymbol{b})$ satisfies the inequality

$$b(p_c) - \sum_{p=1}^{n_p} L(p_c, p) m(p) \leq \max(0, \sum_{p=1}^{n_p} \boldsymbol{L}(p_c, p) \boldsymbol{C}(p,t)) - 1 \quad (1)$$

at least for one $p_c \in \{1, \ldots, n_{dc}\}$ with $k(p_c) = i$. ■

Expression (1) requires some further explanation. Notice that the LHS of inequality (1) equals the marking of place p_c corresponding to marking $\boldsymbol{m}$ of the uncontrolled PN, while the RHS equals the weight of the arc (p_c, t) minus 1. Indeed, $m(p_c) = b(p_c) - \sum_{p=1}^{n_p} L(p_c, p) m(p)$ and $Pre_c(p_c, t) = \max(0, -C_c(p_c, t))$, where $\boldsymbol{C}_c = -\boldsymbol{L}\boldsymbol{C}$. Therefore, the inequality expresses the disabling of t by p_c.

Definition 3.2 ensures the (local) behavioral controllability of the legal set, by ensuring that an arc $a = (\boldsymbol{m}, \boldsymbol{m}') \in A$, where $\boldsymbol{m} \in \overline{\mathcal{L}}$ and $\boldsymbol{m}' \notin \overline{\mathcal{L}}$ is forbidden by a control place belonging to a control site S_i for which the associated transition is accessible and controllable, *i.e.* $h(a) \in T_{c_i}$. In doing this, the use of arcs from control places to locally uncontrollable transitions is not inhibited, as long as the latter are never disabled *only* by such control places. Consider the partially

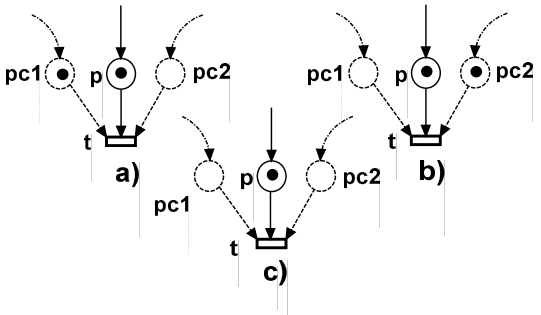

Fig. 1. Monitor places p_{c1} and p_{c2} acting on the partially controllable transition t. Monitor places and arcs are dashed.

controllable transition t in Fig.1, and two monitor places p_{c1} and p_{c2} associated respectively to site 1 and 2. Precisely, assume that t is controllable from site S_1 and uncontrollable from site S_2. From a structural point of view, the arc going from p_{c2} to t is not admissible. On the other hand, if a behavioral approach is adopted, the arc is admissible provided a marking such as the one in Fig.1a, where t would be disabled precisely by p_{c2}, is forbidden. On the other hand, the markings in Figs.1b-c are both legal since t results to be disabled from p_{c1}, for which it is controllable. In particular, in Fig.1c the fact that p_{c2} is unmarked is devoid of consequences (the disabling action is effectively performed by p_{c1}).

Stated otherwise, there always exists at least a disabling control place for which such transitions are controllable, and the presence of such arcs has no disabling effects under boundary legal markings. This additional degree of freedom in the supervisor structure may potentially increase its permissiveness, as opposed to supervisors enforcing structural controllability conditions.

3.2 The ILP core problem

As anticipated, the proposed approach operates by means of a proposal-acceptance mechanism, where a BC-feasible candidate legal set $\mathcal{L}'$ is first selected, and then tested for the existence of a decentralized supervisor that can exactly enforce it (DC-feasibility).

Let $\mathcal{L}' \subseteq \mathcal{L}$ and $\mathcal{U}$ denote the target legal and illegal state sets for the decentralized control design problem. The test is conceived as follows: we first solve an ILP problem that maximizes the number of markings in $\mathcal{L}'$ that can be allowed by a decentralized supervisor. Then, if the optimal value is equal to $|\mathcal{L}'|$, the obtained decentralized supervisor also achieves the required static and behavioral properties, thereby representing a feasible solution for the overall supervisor design problem. If the test has a negative outcome, there could still be a feasible solution allowing only a subset of markings in $\mathcal{L}'$. We use this information to find smaller target legal sets and repeat the procedure on them.

The core component of the method is the test procedure. To this aim, let $(\boldsymbol{L}, \boldsymbol{b})$, $\boldsymbol{L} \in \mathbb{N}^{n_{dc} \times n_p}$, $\boldsymbol{b} \in \mathbb{N}^{n_{dc}}$, denote the GMEC parameters to be determined (the maximum number of GMECs n_{dc} is a design parameter). Let also $\boldsymbol{\gamma} \in \{0,1\}^{|\mathcal{L}'|+|\mathcal{U}|}$ be a binary decision variable used to identify the states allowed by the decentralized supervisor ($\gamma(\boldsymbol{m}) = 1$ if $\boldsymbol{m}$ is allowed and 0 otherwise; in particular, $\gamma(\boldsymbol{m}) = 0, \forall \boldsymbol{m} \in \mathcal{U}$).

The test procedure consists in solving an ILP problem. The latter is first presented in a nonlinear form, as an Integer Programming (IP) problem, for better readability. Its reduction to an ILP problem is discussed further on.

$$\max f = \sum_{\boldsymbol{m} \in \mathcal{L}'} \gamma(\boldsymbol{m}) - \epsilon \sum_{p_c=1}^{n_{dc}} \sum_{p=1}^{n_p} L(p_c, p) - \epsilon \sum_{p_c=1}^{n_{dc}} b(p_c) \tag{2a}$$

$$\sum_{p=1}^{n_p} L(p_c, p) m(p) - b(p_c) \leq (1 - \gamma(\boldsymbol{m}))M, \quad p_c = 1, \ldots, n_{dc}, \boldsymbol{m} \in \mathcal{L}' \tag{2b}$$

$$\sum_{p=1}^{n_p} L(p_c, p) m(p) - b(p_c) \geq 1 - (1 - \delta(p_c, \boldsymbol{m}))M, \quad p_c = 1, \ldots, n_{dc}, \boldsymbol{m} \in \mathcal{L}' \cup \mathcal{U} \tag{2c}$$

$$\gamma(\boldsymbol{m}) + \sum_{p_c=1}^{n_{dc}} \delta(p_c, \boldsymbol{m}) \geq 1, \ \boldsymbol{m} \in \mathcal{L}' \cup \mathcal{U} \tag{2d}$$

$$\gamma(\boldsymbol{m}_2) - \gamma(\boldsymbol{m}_1) + \sum_{p_c=1}^{n_{dc}} \sum_{i=1}^{\nu} \Phi(i,t) k(p_c, i) \delta(p_c, \boldsymbol{m}_2) \geq 0, \quad \forall (\boldsymbol{m}_1, \boldsymbol{m}_2) \in \mathcal{L}' \times (\mathcal{L}' \cup \mathcal{U}), \ s.t. \sum_{i=1}^{\nu} \Phi(i,t) \geq 1, \boldsymbol{m}_1[t > \boldsymbol{m}_2 \tag{2e}$$

$$O_c(p_c, t) - I_c(p_c, t) = -\sum_{p=1}^{n_p} L(p_c, p) C(p, t), \quad p_c = 1, \ldots, n_{dc}, t = 1, \ldots, n_t \tag{2f}$$

$$\sum_{p=1}^{n_p} L(p_c, p) \boldsymbol{m}_0(p) - b(p_c) \leq 0, \ p_c = 1, \ldots, n_{dc} \tag{2g}$$

$$O_c(p_c, t) \leq X^O(p_c, t) M, \ p_c = 1, \ldots, n_{dc}, t = 1, \ldots, n_t \tag{2h}$$

$$I_c(p_c, t) \leq X^I(p_c, t) M, \ p_c = 1, \ldots, n_{dc}, t = 1, \ldots, n_t \tag{2i}$$

$$\sum_{t \notin T_i} X^O(p_c, t) \leq (1 - k(p_c, i))M, \ p_c = 1, \ldots, n_{dc}, i = 1, \ldots, \nu \tag{2j}$$

$$\sum_{t \notin T_i} X^I(p_c, t) \leq (1 - k(p_c, i))M, \ p_c = 1, \ldots, n_{dc}, i = 1, \ldots, \nu \tag{2k}$$

$$\sum_{i=1}^{\nu} k(p_c, i) = 1, \ p_c = 1, \ldots, n_{dc} \tag{2l}$$

$$O_c(p_c, t), I_c(p_c, t) \geq 0, \ p_c = 1, \ldots, n_{dc}, t = 1, \ldots, n_t \tag{2m}$$

$$X^O(p_c, t), X^I(p_c, t) \in \{0, 1\}, \ p_c = 1, \ldots, n_{dc}, t = 1, \ldots, n_t \tag{2n}$$

$$k(p_c, i) \in \{0, 1\}, \ p_c = 1, \ldots, n_{dc}, i = 1, \ldots, \nu \tag{2o}$$

$$\gamma(\boldsymbol{m}) \in \{0, 1\}, \ \forall \boldsymbol{m} \in \mathcal{L}' \cup \mathcal{U} \tag{2p}$$

$$\gamma(\boldsymbol{m}) = 0, \ \forall \boldsymbol{m} \in \mathcal{U} \tag{2q}$$

$$\delta(p_c, \boldsymbol{m}) \in \{0, 1\}, \ p_c = 1, \ldots, n_{dc}, \forall \boldsymbol{m} \in \mathcal{L}' \cup \mathcal{U} \tag{2r}$$

The cost function (2a) is a hierarchial objective function: the primary objective is the maximization of the number of markings in $\mathcal{L}'$ that are allowed by the decentralized supervisor, while the secondary objective is to minimize the GMEC coefficients. The purpose of the latter is to prevent ill-conditioning of the optimization problem (GMECs are defined up to a multiplicative constant). To ensure that it doesn't practically affect the primary objective, it is weighted by a suitably small factor ($\epsilon = 0.01$ in the illustrated simulations). Notice, finally, that the secondary objective also ensures that, if there exists a solution with fewer monitors than n_{dc}, one or more of the obtained GMECs will have null parameters, allowing the designer to easily discard them *a posteriori*.

Constraint (2b) ensures that all the legal states with $\gamma(\boldsymbol{m}) = 1$ are allowed by all monitors, while constraint (2c) applies to the remaining states in $\mathcal{L}'$ as well as to illegal states in $\mathcal{U}$ ($\gamma(\boldsymbol{m}) = 0$), stating that $\boldsymbol{m}$ violates the GMEC associated to monitor p_c. Here, the binary variable $\delta(p_c, \boldsymbol{m})$ is used to associate each forbidden marking to the control place that forbids it. By constraint (2d) each marking in $\mathcal{L}' \cup \mathcal{U}$ is either allowed ($\gamma(\boldsymbol{m}) = 1$ and $\delta(p_c, \boldsymbol{m}) = 0$ for all control places) or

forbidden ($\gamma(\boldsymbol{m}) = 0$ implies that $\delta(p_c, \boldsymbol{m})$ must equal 1 for at least one control place). The constant M is set to a sufficiently large value (big-M parameter), so that constraint (2b) is always satisfied for $\gamma(\boldsymbol{m}) = 0$. Similarly, constraint (2c) automatically holds if $\delta(p_c, \boldsymbol{m}) = 0$. The big-M parameter is set to $M = 10$ in the examples documented in the paper.

Constraint (2e) applies to all transitions controllable at least by one control site ($\sum_{i=1}^{\nu} \Phi(i,t) \geq 1$). In detail, the constraint requires that if $\boldsymbol{m}_1$ is a legal marking ($\gamma(\boldsymbol{m}_1) = 1$) and $\boldsymbol{m}_2$ is an illegal one ($\gamma(\boldsymbol{m}_2) = 0$), the firing of a transition leading from $\boldsymbol{m}_1$ to $\boldsymbol{m}_2$ must be forbidden at least by one control place acting on a site from which t is controllable. More precisely, $\delta(p_c, \boldsymbol{m}_2)$, $k(p_c, i)$, $\Phi(i,t)$ are simultaneously equal to 1 if p_c forbids $\boldsymbol{m}_2$, and belongs to site S_i, and t is controllable from S_i. Notice that the constraint is automatically satisfied for all possible values of the decision variables $k(p_c, i)$ and $\delta(p_c, \boldsymbol{m}_2)$ if $\gamma(\boldsymbol{m}_2) = 1$ and/or $\gamma(\boldsymbol{m}_1) = 0$.

Equation (2f) calculates the supervisor net topology (weights of the arcs connecting the monitors to the PN transitions) corresponding to the GMEC parameters $\boldsymbol{L}$ and $\boldsymbol{b}$, and (2g) requires that the GMECs are satisfied in the initial marking.

Conditions (2h) and (2i) forbid the use of specific arcs, depending on the given decentralization conditions (each monitor can operate only on the transitions associated to its control site). For this purpose, if p_c belongs to site S_i (*i.e.*, $k(p_c, i) = 1$), the binary parameters $X^O(p_c, t)$ and $X^I(p_c, t)$ are set to 0 for all $t \notin T_i$ (constraints (2j) and (2k)). Condition (2l) specifies that each monitor must be assigned to one module only.

Notice, finally, that γ is preset to 0 for declaredly illegal markings ($\boldsymbol{m} \in \mathcal{U}$).

***Remark** 3.3.* At a closer look, constraint (2e) is nonlinear, since both k and δ are decision variables. Although this is not an issue here, since in the solution approach the formulation will always be employed with k assigned from outside (thereby simplifying the problem to an ILP one), an alternative expression can be used if necessary to linearize the problem. More precisely, introducing a further binary variable $\psi(p_c, i, \boldsymbol{m})$, we can write:

$$\begin{aligned} &\psi(p_c, i, \boldsymbol{m}) \leq k(p_c, i) \\ &\psi(p_c, i, \boldsymbol{m}) \leq \delta(p_c, \boldsymbol{m}) \\ &\psi(p_c, i, \boldsymbol{m}) \geq k(p_c, i) + \delta(p_c, \boldsymbol{m}) - 1 \end{aligned}$$

for $p_c = 1, \ldots, n_{dc}$, $i = 1, \ldots, \nu$, and $\forall \boldsymbol{m} \in \mathcal{L}' \cup \mathcal{U}$. Then, product $k(p_c, i)\delta(p_c, \boldsymbol{m}_2)$ in expression (2e) can be replaced by $\psi(p_c, i, \boldsymbol{m}_2)$. ■

3.3 Conditions for BC-, and DC-feasibility and supervisor optimality

Problem (2) can be used to find DC-feasible legal state sets, as stated by the following Lemma.

***Lemma** 3.4.* A set $\overline{\mathcal{L}} \subseteq \mathcal{L}$ is DC-feasible if there exists a set of GMECs $(\boldsymbol{L}, \boldsymbol{b})$ that provides a feasible solution to problem (2), for a given n_{dc} and starting with $\mathcal{L}' = \overline{\mathcal{L}}$, and if the obtained solution has $\gamma(\boldsymbol{m}) = 1$ for each $\boldsymbol{m} \in \mathcal{L}'$.

Proof. By construction, any feasible solution of problem (2) will respect the decentralization requirements (Def. 3.2.i), thanks to constraints ((2j-2l), which enforce the condition that each monitor must be assigned to a control site, *i.e.* that it cannot have connections with the transitions not belonging to that site. The optimal solution of problem (2) will allow a subset of the states in $\mathcal{L}'$. However, since by assumption the obtained solution has $\gamma(\boldsymbol{m}) = 1$ for each $\boldsymbol{m} \in \mathcal{L}'$, the obtained decentralized supervisor enforces the whole of $\overline{\mathcal{L}}$. Constraint (2e) enforces condition (ii) of Def. 3.2. Indeed, observe that it is a non trivial constraint only if $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ are a legal and an illegal marking, respectively ($\gamma(\boldsymbol{m}_1) = 1$, $\gamma(\boldsymbol{m}_2) = 0$), such that $\boldsymbol{m}_1[t > \boldsymbol{m}_2$. This makes $\boldsymbol{m}_1$ a marking of $\overline{\mathcal{L}}_b$ and t a transition belonging to $\mathcal{D}(\boldsymbol{m}_1)$. Now, the monitor p_c forbidding $\boldsymbol{m}_2$ ($\delta(p_c, \boldsymbol{m}_2) = 1$) will be actually disabling t in $\boldsymbol{m}_1$. For this to occur, its marking in correspondence to $\boldsymbol{m}_1$ will necessarily have to be less than the weight of the arc (p_c, t), as expressed by condition (1). Also, by constraint (2e), p_c must operate on a control site S_i for which t is controllable ($\Phi(i,t) \geq 1$). ■

The following result provides a necessary and sufficient condition for BC-feasibility as well.

***Lemma** 3.5.* Let $\{\boldsymbol{m}_0\} \subset \overline{\mathcal{L}} \subseteq \mathcal{L}$ and $(\overline{\mathcal{L}}, \overline{A})$ be a subgraph of the reachability graph (V, A), such that $\overline{A} \subseteq A$ and $(\boldsymbol{m}, \boldsymbol{m}') \in \overline{A}$ iff $\boldsymbol{m}, \boldsymbol{m}' \in \overline{\mathcal{L}}$. The set $\overline{\mathcal{L}}$ is BC-feasible iff

i) $(\overline{\mathcal{L}}, \overline{A})$ has only one SCC;
ii) $\forall t_j \in T$ there exists $a \in \overline{A}$ such that $h(a) = t_j$;
iii) $\forall a = (\boldsymbol{m}, \boldsymbol{m}') \in A$ s.t. $\boldsymbol{m} \in \overline{\mathcal{L}}$ and $h(a) \notin T_c$, it holds that $\boldsymbol{m}' \in \overline{\mathcal{L}}$ as well.

Proof. Assume that there exists a GMEC-based supervisor exactly enforcing $\overline{\mathcal{L}}$. Such a supervisor will achieve reversibility, liveness, and (global) behavioral controllability. Reversibility follows immediately upon observing that the reachability graph of a (bounded) reversible PN has a unique SCC (coinciding with the entire graph itself) (Fumagalli et al., 2010), so that any state is reachable from any other. Deadlock-freeness is also automatically obtained since the PN can evolve indefinitely in a SCC with cardinality greater than 1 (as implied by assumption $\overline{\mathcal{L}} \supset \{\boldsymbol{m}_0\}$). The only additional requirement for liveness is that $\forall t_j \in T$ there exists at least an arc in the reachability graph associated to the firing of t_j (Fumagalli et al., 2010). This is ensured by condition (ii). Finally, condition (iii) implies that there cannot be an arc $a = (\boldsymbol{m}, \boldsymbol{m}') \in A$ s.t. $\boldsymbol{m} \in \overline{\mathcal{L}}$ and $\boldsymbol{m}' \notin \overline{\mathcal{L}}$, with $h(a) \notin T_c$. In other words, no illegal marking is reachable from within $\overline{\mathcal{L}}$ by firing only uncontrollable transitions, as required by Def. 2.2. ■

The BC-feasibility test does not require the explicit calculation of the set of GMECs enforcing the given set of states.

***Definition** 3.6.* A set of GMECs $(\boldsymbol{L}, \boldsymbol{b})$ results in a maximally permissive decentralized supervisor if it enforces a maximal BC- and DC-feasible subset of the set $\mathcal{L}$ of (globally) legal states. ■

The following result provides conditions for the optimality of a decentralized supervisor.

***Theorem** 3.7.* A decentralized supervisor is maximally permissive if, denoting as $\overline{\mathcal{L}}$ the set of legal states that it enforces, either $\overline{\mathcal{L}} = \mathcal{L}$ or there does not exist $\overline{\mathcal{L}}' \subseteq \mathcal{L}$ s.t.:

i) $|\overline{\mathcal{L}}'| > |\overline{\mathcal{L}}|$;
ii) Problem (2) applied with $\mathcal{L}' = \overline{\mathcal{L}}'$ yields a solution with $\gamma(\boldsymbol{m}) = 1$ for all $\boldsymbol{m} \in \mathcal{L}'$;
iii) There does not exist a (globally) uncontrollable transition enabled in a marking of $\overline{\mathcal{L}}'$ whose firing causes the exit from that set;

iv) The subgraph of the reachability graph induced by $\overline{\mathcal{L}}'$ configures a unique SCC with at least an arc labeled with each of the transitions.

Proof. By definition $\mathcal{L}$ is the maximal set of legal states for the centralized supervisor design problem, so that if $\overline{\mathcal{L}} = \mathcal{L}$ the decentralized supervisor is as permissive as the maximally permissive optimal centralized supervisor. On the other hand, if $\overline{\mathcal{L}} \subset \mathcal{L}$ and there exists a set $\overline{\mathcal{L}}'$ for which conditions (i–iv) hold, then the supervisor is not optimal. Indeed, in that case there would be a larger subset of globally legal states (cond. (i)), both DC- (cond. (ii) implies that Lemma 3.4 holds) and BC-feasible (Lemma 3.5 applies by cond.s (iii) and (iv)). ■

3.4 A B&B approach for the design of an optimal decentralized supervisor

Based on the previous material, a given proposal subset $\overline{\mathcal{L}}$ of $\mathcal{L}$ can be tested for BC-feasibility analyzing the reachability subgraph induced by $\overline{\mathcal{L}}$ alongside Lemma 3.5, and for DC-feasibility by applying problem (2) with $\mathcal{L}' = \overline{\mathcal{L}}$ and checking for the existence of solutions with $\gamma(\boldsymbol{m}) = 1$ for all $\boldsymbol{m} \in \mathcal{L}'$ (Lemma 3.4). In practice, n_{dc} is assumed fixed and the available monitors are preassigned to the control sites, so that variables $k(p_c, i)$ are given. Therefore the DC-feasibility of $\overline{\mathcal{L}}$ is ascertained by solving a relaxed version of problem (2), which is in fact an ILP problem. An efficient systematic exploration of all possible proposal subsets of $\mathcal{L}$ can then be carried out using a B&B approach.

For this purpose, a generic node of the branching tree is associated to a specific assignment of the (globally) legal states, defined by suitable subsets of $\mathcal{L}$, $\Pi_i = \{\mathcal{L}_{in}, \mathcal{L}_{out}\}$, where $\mathcal{L}_{in} \subseteq \mathcal{L}$ identifies the legal states that must be allowed by any feasible solution of the current node, and $\mathcal{L}_{out} \subseteq (\mathcal{L} \setminus \mathcal{L}_{in})$ groups the legal markings that must not be included in the solution. The remaining legal states, $\mathcal{L}_{free} = \mathcal{L} \setminus (\mathcal{L}_{in} \cup \mathcal{L}_{out})$ are the ones to perform the actual optimization on. Branching results in more free states being assigned to $\mathcal{L}_{in}$ or $\mathcal{L}_{out}$.

To reduce the number of proposal subsets to which Lemmas 3.4-3.5 actually need to be applied, the following pre-processing can be applied at each node Π_i, previous to the solution of problem (2). Indeed, by Lemma 3.5 observe that a feasible solution can allow at most a set of legal states $\overline{\mathcal{L}} \subseteq \mathcal{L}_{in} \cup \mathcal{L}_{free}$ that configures a subgraph of the reachability graph of the PN that forms a SCC. If there exists such a component including all states in $\mathcal{L}_{in}$ and excluding all states in $\mathcal{L}_{out}$, since by definition it is unique, it allows to extend $\mathcal{L}_{out}$ with all the free states that do not belong to the component. If such a SCC does not exist, the node is unfeasible and can be eliminated.

If, after this pre-processing, the node is still potentially feasible, and $|\mathcal{L}_{in} \cup \mathcal{L}_{free}|$ is larger than the cardinality of the set of states allowed by the current best solution, problem (2) is solved presetting $\gamma(\boldsymbol{m}) = 1$ for each $\boldsymbol{m} \in \mathcal{L}_{in}$ and $\gamma(\boldsymbol{m}) = 0$ for each $\boldsymbol{m} \in \mathcal{L}_{out}$, thus restricting in practice the optimization to the remaining free legal states. By construction, the ILP will find the decentralized set of GMECs (if one exists) that allows all the states in $\mathcal{L}_{in}$ and as many states in $\mathcal{L}_{free}$ as possible, while forbidding all the states in $\mathcal{L}_{out}$. Then, if the set $\overline{\mathcal{L}}$ of states enforced by such a solution is also BC-feasible it provides a local optimum and further branching of the node is not required. In addition, if the solution is more permissive than the current best it is stored in its stead. Notice that BC-feasibility certainly holds if $\overline{\mathcal{L}} = \mathcal{L}_{in} \cup \mathcal{L}_{free}$ (after the pre-processing $\mathcal{L}_{in} \cup \mathcal{L}_{free}$ configures a strongly connected subgraph), otherwise it must be ascertained *a posteriori*. If BC-feasibility does not hold, the node is branched (a feasible solution with fewer states could exist). Finally, if the ILP terminates without finding solutions, no decentralized supervisor exists that is compatible with the given state assignments, and the node is eliminated.

For simplicity a binary branching policy is adopted. Two children nodes are generated, inheriting the state assignments from the father node. Then, a free marking not in $\overline{\mathcal{L}}$ is picked out and added to $\mathcal{L}_{out}$ for the first child node and to $\mathcal{L}_{in}$ for the second one, respectively. Notice that including in $\mathcal{L}_{in}$ a free marking that is not in $\overline{\mathcal{L}}$ will force the ILP to find a different DC-feasible solution with less allowed states, while setting it in $\mathcal{L}_{out}$ may trigger further state assignments due to pre-processing.

The branching process is initialized with a root node defined as follows, $\Pi_0 = \{\{\boldsymbol{m}_0\}, \emptyset\}$. Notice, finally, that by repeating the procedure for increasing values of n_{dc} one can also ascertain the optimality of the supervisor in terms of its size.

4. SIMULATION EXAMPLE

Consider the PN represented in Fig. 2 taken from (Ghaffari et al., 2003), for which we want to design a GMEC-based supervisor that guarantees liveness, reversibility and controllability. Resource (M_1, M_2, M_3, and R) and idle (B_1 and B_2) places are considered part of the process. The PN has 331 reachable markings, only 300 of which are included in the initial SCC.

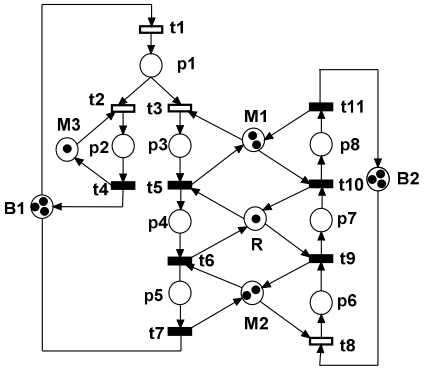

Fig. 2. Petri net of example.

Consider first the centralized supervisor design problem and assume that $T_c = \{t_1\ t_2\ t_3\ t_5\ t_8\}$ and $T_{uc} = T \setminus T_c$. The resulting optimal solution has 2 GMECs: $m_4 + m_6 \leq 2$ and $3m_3 + m_6 + m_7 \leq 6$ and allows 295 states of the 300 maximum possible. It is interesting to note that if t_5 is not assumed controllable, the optimal solution of the problem allows 280 markings only, using the following 2 GMECs: $m_4 + m_6 \leq 2$ and $3m_3 + 2m_6 + m_7 \leq 6$.

The first monitor has an arc towards an uncontrollable transition (t_5), but this is fine, since it is never exclusively responsible of its disabling. Indeed, there are 45 markings in which two or more places (among which the mentioned monitor) disable t_5, but no marking is disabled by the monitor alone. In both cases, the optimal solution is found already at the first node of the B&B procedure. This can be justified by recalling that the optimal solution of problem (2) is the maximal subset of $\mathcal{L}$ for which a correct decentralized supervisor exists. So, even if it

is smaller than $\mathcal{L}$, it provides the optimal solution if – tested *a posteriori* – it is found to be BC-feasible as well.

Now, consider the same problem in a decentralized setting, where we can employ monitors of two control sites, defined in 3 alternative scenarios (differing only for the role of t_5) as follows:

case a) $S_1 : T_1 = [t_5\ t_6\ t_8\ t_9]$, with $T_{c_1} = [t_8]$,
$S_2 : T_2 = [t_3\ t_5\ t_7\ t_8\ t_9\ t_{10}]$, with $T_{c_2} = [t_3\ t_8]$,
case b) $S_1 : T_1 = [t_5\ t_6\ t_8\ t_9]$, with $T_{c_1} = [t_5\ t_8]$,
$S_2 : T_2 = [t_3\ t_5\ t_7\ t_8\ t_9\ t_{10}]$, with $T_{c_2} = [t_3\ t_8]$,
case c) $S_1 : T_1 = [t_5\ t_6\ t_8\ t_9]$, with $T_{c_1} = [t_8]$,
$S_2 : T_2 = [t_3\ t_5\ t_7\ t_8\ t_9\ t_{10}]$, with $T_{c_2} = [t_3\ t_5\ t_8]$.

In all three cases the optimal supervisor has one GMEC per control site (increasing n_{dc} does not provide different solutions), but the number of allowed markings is different (280 in cases (a) and (c), and 295 in case (b)).

An even more interesting case unfolds if one removes place R from the PN, and adds the corresponding static constraint:

$$m_4 + m_7 \leq 1 \tag{3}$$

to the supervisor design requirements. In other words, our aim here is to evaluate the cost of imposing the requirement corresponding to place R, which was previously centralized, in a decentralized way. This time the optimal solution requires one monitor of control site S_1, and 2 belonging to S_2 in all three studied scenarios. Again, increasing n_{dc} to allow more GMECs per control site does not provide solution improvements. The number of allowed states is much smaller (128, 138 and 142 markings, respectively), implying that constraint (3) is not easily implemented with the given decentralization conditions. Also, the B&B is not solved at the initial node or immediately after, but requires some non-trivial processing (25, 67, and 299 nodes, respectively).

Consider, *e.g.*, the solution relative to case (b):

$$S_1 : m_4 + m_6 \leq 1 \tag{4a}$$
$$S_2 : m_3 + 2m_6 + 2m_7 \leq 2 \tag{4b}$$
$$m_4 + m_5 + 3m_7 \leq 3 \tag{4c}$$

The first monitor of site S_2 has an arc towards t_5 which is uncontrollable from S_2. There are 23 markings where such monitor disables t_5, always in combination with a simultaneous disabling by place p_3, and twice even by the unique monitor of S_1 (for which t_5 is controllable).

5. CONCLUSIONS

A new approach to the synthesis of decentralized monitors in the presence of static and behavioral (including liveness, reversibility, and controllability) specifications has been presented. Starting from the legal and illegal sets of the centralized control case, an ILP problem is formulated that aims at maximizing the number of legal states allowed by the decentralized monitors. Even assuming that a feasible solution to such ILP problem exists, there is no guarantee that the subset of legal states enforced by the decentralized supervisor achieves the required behavioral specifications. A branch & bound (B&B) method has been developed to systematically and efficiently explore all possible subsets of the legal states of the centralized case to find the largest one that meets the constraints, *i.e.* that can be implemented in a decentralized form and for which all the (static and) behavioral specifications hold.

REFERENCES

Barret, G. and Lafortune, S. (2000). Decentralized supervisory control with communicating controllers. *IEEE Trans. on Aut. Control*, 45(9), 1620–1638.

Basile, F., Cordone, R., and Piroddi, L. (2013a). Compact and decentralized supervisors for general constraint enforcement in Petri net models. In *52nd IEEE Conference on Decision and Control (CDC'13)*. Florence, Italy.

Basile, F., Cordone, R., and Piroddi, L. (2013b). Integrated design of optimal supervisors for the enforcement of static and behavioral specifications in Petri net models. *Automatica*, 49, 3432–3439.

Chen, H. and Hu, B. (1991). Distributed control of discrete event systems described by a class of controlled Petri nets. In *IFAC Int. Symposium on Distributed Intelligence Systems*.

Cordone, R., Nazeem, A., Piroddi, L., and Reveliotis, S. (2012). Maximally permissive deadlock avoidance for sequential resource allocation systems using disjunctions of linear classifiers. In 51^{st} *IEEE Conf. on Decision and Control*. Maui (HI), USA.

Cordone, R. and Piroddi, L. (2011). Monitor optimization in Petri net control. In 7^{th} *IEEE Conf. on Automation Science and Engineering*, 413–418. Trieste, Italy.

Cordone, R. and Piroddi, L. (2013). Parsimonious monitor control of Petri net models of FMS. *IEEE Trans. Syst. Man Cybern., Part A Syst. Humans*, 43(1), 215–221.

Fumagalli, I., Piroddi, L., and Cordone, R. (2010). A reachability graph partitioning technique for the analysis of deadlock prevention methods in bounded Petri nets. In *American Control Conference, ACC2010*, 3365–3370. Baltimore (MD), USA.

Ghaffari, A., Rezg, N., and Xie, X. (2003). Design of a live and maximally permissive Petri net controller using the theory of regions. *IEEE Trans. on Robotics and Automation*, 19(1), 137–141.

Giua, A., DiCesare, F., and Silva, M. (1992). Generalized mutual exclusion constraints on nets with uncontrollable transitions. In *IEEE Int. Conf. on Systems, Man and Cybernetics*, 974–979. Chicago (IL), USA.

Guan, X. and Holloway, L. (1997). Control of distributed discrete event systems modeled as Petri nets. In *1997 American Control Conference*, 2342–2347. Albuquerque (NM), USA.

Iordache, M. and Antsaklis, P. (2006). Decentralized control of Petri nets with constraint tansformation. *IEEE Trans. on Aut. Control*, 51(2), 376–381.

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

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proc. of the IEEE*, 77(4), 541–580.

Reveliotis, S.A. and Choi, J.Y. (2006). Designing reversibility-enforcing supervisors of polynomial complexity for bounded petri nets through the theory of regions. In *Intl. Conf. on the Application and Theory of Petri Nets and Other Models of Concurrency (ATPN 2006)*, 322–341. Turku, Finland.

Rudie, K. and Wonham, W. (1992). Think globally, act locally: Decentralized supervisory control. *IEEE Trans. on Aut. Control*, 37(11), 1692–1708.

Yamalidou, K., Moody, J., Lemmon, M., and Antsaklis, P. (1996). Feedback control of Petri nets based on place invariants. *Automatica*, 32(1), 15–28.