Supervisor Synthesis for Deadlock Prevention of Robotic Manufacturing Cells with Uncontrollable and Unobservable Events Using Petri Nets

Hao Yue, Jianming Huang,

***Abstract*—Robotic manufacturing cells (RMCs) play an important role in manufacturing industry. Ensuring deadlock freeness remains challenging for RMCs because of the complexity of their discrete event dynamics. By using Petri nets as a formalism, this paper proposes a deadlock prevention approach for RMCs with uncontrollable and unobservable events. First, for the whole Petri net system model, a decomposition technique is developed to obtain subnets via shared transitions. Second, a liveness-enforcing supervisor is designed for each subnet by an integer linear programing (ILP). Place invariants are designed in the ILP to allow all admissible markings and prohibit all inadmissible markings. Third, the reason of causing deadlocks is analyzed when merging the subnets, and then the constraints are obtained. A constraint transformation method is proposed to convert constraints into admissible ones. Finally, the complete supervisor guaranteed to be admissible can be obtained. Our developed approach only needs to calculate the reachability graphs of subnets, therefore, it requires lower computational cost as compared to the exiting one in the recent literature. In addition, it can perform constraint transformation according to subnet constraints rather than the given constraints of the whole net. Experimental and comparative studies illustrate the effectiveness of this approach.**

***Index Terms*—Decomposition technique, discrete event systems, Petri nets, robotic manufacturing cells, supervisory control, uncontrollable and unobservable events.**

I. Introduction

In recent years, with the rapid development of information technology, manufacturing production has become increasingly unmanned and intelligent [1]–[4]. Robotic manufacturing cells (RMCs), categorized as a type of discrete event systems (DESs) from the perspective of supervisory control, are increasingly widely used in the manufacturing industry, including automotive and semiconductor manufacturing. Usually modeled with Petri nets and automata, DESs have been investigated for multiple purposes, e.g., supervisory control [5], [6], fault detection and prognosis [7], resource failures [8]–[11] , state estimation [12], and scheduling [13], [14].

For a highly automated RMC, parallel processes (such as vehicles and parts) share available resources (such as machines and robots) [15]. The competition for the limited resources by different processes may lead to deadlocks, where two or more processes are indefinitely waiting for the other to release their acquired resources. The occurrence of a deadlock can lead to unnecessary economic costs and sometimes catastrophic results.

As other formal tools such as digraphs and automata, Petri nets are often used to deal with deadlock in RMCs because of their visual and intuitive graphical representation and rigorous mathematical expressions, which are suitable for describing concurrency, conflict, and synchronization.

As a main type of way of handling deadlock problems, deadlock prevention is a static method that controls the resource requirements with an off-line mechanism to ensure that a system has no deadlock. Generally, in the framework of Petri nets, deadlock prevention is done by computing necessary control places, leading to liveness-enforcing Petri net supervisors. When evaluating and designing the liveness-enforcing supervisor for a system to be controlled, there are three very important performance criteria: behavioral permissiveness, structural complexity, and computational complexity.

Based on Petri nets, there are two main methods to prevent deadlocks: structure analysis and reachability graph analysis. In terms of structure analysis, Ezpeleta *et al*. [16] analyzed the relationship of strict minimal siphons and deadlocks in a system of simple sequential processes with resources ($\mathrm{S^3PR}$) to propose a method for adding control places. Finally, A liveness-enforcing Petri net supervisor can be obtained. Theoretically, the number of siphons increases exponentially with the size of a net, leading this approach to overly restrict the behavioral permissiveness of the controlled system. As a response, Chu and Xie [17] introduced a mixed integer programming (MIP) method to compute minimal siphons effectively which can avoid the complete siphon enumeration. Building upon MIP, Huang *et al*. [18] developed an iterative deadlock control strategy for Petri nets. In each iteration, a maximal empty siphon is computed using MIP. From this, a strict minimal siphon is extracted and a control place is added until there are no more siphons that can be unmarked. This method is proved to be practically efficient, as it negates the necessity for both complete siphons and state enumerations. Wang [19] *et al*. presented a two-stage deadlock prevention policy using MIP and iterative siphon control.

For the analysis of reachability graph, the theory of regions is adopted in [20] to derive optimal and deadlock-free supervisors for flexible manufacturing systems (FMSs) if such supervisors exist. Some improvements have been proposed in [21]–[23] to design behaviorally optimal and structurally simple supervisors. In [24], Zhong *et al*. developed a decomposition technique which can significantly reduce the calculation of

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

Manuscript received April 19, 2021; revised August 16, 2021.

reachability graphs to design liveness-enforcing supervisors for Petri nets modeling a wide class of FMSs. Huang *et al*. [14] used symbolic representations and manipulations to reduce the computational burden in the scheduling of RMCs based on Petri nets' reachability graphs. In [25], Hu *et al* not only conducted a comparison investigation but also developed some new theoretical results about structural simplification techniques. Uzam *et al*. [26] presented a divide-and-conquer method that can be applied to large Petri net models.

Many studies assume that all events in the systems are controllable and observable. However, controllability and observability are two properties that should never be taken for granted in many industrial applications in the real world, since the downtime of machine and sensor is normally inevitable.

When considering uncontrollable and unobservable events, Fig. 1 shows the process of supervisor synthesis where constraints transformation plays an important role [27]. A supervisor synthesis approach was proposed in [28] based on event separation instances, ensuring maximally permissiveness in behavior. For conventional Petri nets with uncontrollable transitions, [29] presented a bottom-up optimal supervisor synthesis approach, complemented by a linear constraint transformation method outlined in [30]. Meanwhile, for Petri nets with encompassing both uncontrollable and unobservable transitions, [31] proposed a technique that converts a given generalized mutual exclusion constraint into an admissible one by using matrix operations. Hao *et al*. [32] proposed a method to transform constraints by solving integer linear programming (ILP). Ran *et al*. [5] designed event feedback supervisors by ILP and structural analysis. Luo and Zhou [6] presented an optimal constraint transformation method for a specific class of Petri nets. You *et al*. [33] designed optimal supervisors for ordinary Petri nets whose backward observation subnets are acyclic, backward-conflict, and backward-concurrent free.

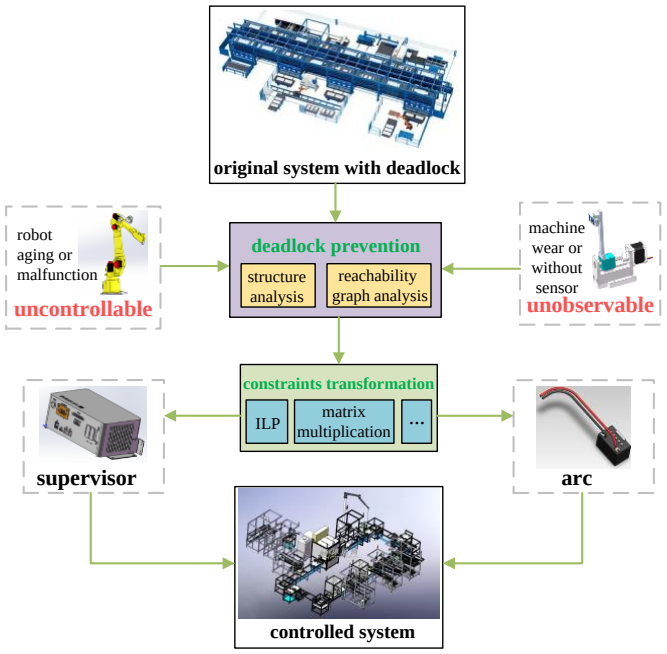


Fig. 1. Process of supervisor synthesis

However, the existing methods only apply to given constraints, while ignoring the computation of constraints. In [34], Huang *et al*. proposed a method for supervisor synthesis based on ILP, so that the supervisor is guaranteed to be admissible and structurally minimal in terms of both control places and added arcs. However, the computation of reachability graph and the enumeration of siphons suffer from high computational complexity which are exponential in the size of the system model.

In the present work, we attempt to develop a supervisor for deadlock prevention in RMCs with uncontrollable and unobservable events. The mainly contributions of this paper are as follows:

1) We develop a net decomposition and merging technique that can cope with Petri nets with uncontrollable and unobservable transitions.
2) ILP and decomposition technique are combined so that we need only compute a small amount of constraints and variables, which makes the ILP method applicable to large systems.
3) We propose a constraint transformation method for uncontrollability and unobservability using ILP.
4) An supervisor synthesis approach is proposed which ensures the control objective by the obtained supervisor and requires low computational cost.

The rest of this paper are organized as follows: The basic concept of Petri nets together with the controllability and observability are reviewed in Section II. Section III introduces some important markings of Petri net including admissible markings and first-met inadmissible markings (FIMs). Section IV shows the method of computing control places based on place invariants and ILP which can forbid all FIMs and permit all admissible markings. Section V develops our deadlock prevention approach. Section VI provides an illustrative example. Comparisons with the existing work are given in Section VII. Finally, Section VIII summarizes this paper.

II. Preliminaries

A. Petri Nets

A Petri net N is a four-tuple (P, T, F, W), where P (resp., T) is a set of places (resp., transitions) such that $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$, $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs, and $W : (P \times T) \cup (T \times P) \to \mathbb{N} \equiv \{0, 1, 2, \cdots\}$ is a mapping that assigns a weight to an arc: $W(x, y) > 0$ if $(x, y) \in F$, and $W(x, y) = 0$, otherwise.

N is pure if for all $x, y \in P \cup T$, $W(x, y) > 0$ implies $W(y, x) = 0$. If N is pure, then it can be represented by its incidence matrix $[N]$, which is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$. A P-vector is a column vector $I : P \to \mathbb{Z}$ indexed by P, where $\mathbb{Z}$ is the set of integers. A P-vector I is called a place invariant (PI) if $I \neq \mathbf{0}$ and $I^T[N] = \mathbf{0}^T$, where $\mathbf{0}$ means a vector of zeros. Given a node $x \in P \cup T$, the preset (resp., postset) of x is defined as ${}^\bullet x = \{y \in P \cup T | (y, x) \in F\}$ (resp., $x^\bullet = \{y \in P \cup T | (x, y) \in F\}$). They can be extended to sets as follows: Given $X \subseteq P \cup T$, ${}^\bullet X = \cup_{x \in X} {}^\bullet x$ and

$X^\bullet = \cup_{x\in X} x^\bullet$. A Petri net is a state machine if $W : F \to \{1\}$ and $\forall t \in T$, $|{}^\bullet t| = |t^\bullet| = 1$.

A marking M of N is a mapping $M : P \to \mathbb{N}$ such that for each $p \in P$, $M(p)$ denotes the number of tokens in place p. For economy of space, we usually describe markings and vectors by a multiset or formal sum. That is, $\sum_{p\in P} M(p)p$ denotes marking or vector M. For instance, a marking M, where 2 tokens in place p_1, 3 tokens in place p_2, and no tokens in other places, is denoted as $M = 2p_1 + 3p_2$. The initial marking is usually denoted by M_0, and the pair (N, M_0) is called a marked Petri net.

A transition $t \in T$ is enabled at a marking M if for each $p \in {}^\bullet t$, $M(p) \geq W(p, t)$, denoted by $M[t\rangle$. The firing of t at M yields a new marking M' such that for each $p \in P$, $M' = M(p) - W(p, t) + W(t, p)$, denoted by $M[t\rangle M'$. In this case, M' is called an immediately reachable marking from M. The set of N's markings reachable from an initial marking M_0 is called the reachability set of (N, M_0), denoted by $\mathcal{R}(N, M_0)$.

B. Controllability and Observability

In the Petri net model for modeling and simulation process of industrial application systems, the unobservable transitions, which constitute the set $T_{\widetilde{O}}$, represent the unobservable events, such as the wear, tear of internal parts of the machine, and machines without sensors installed [35]. Their firing cannot be directly detected or measured, thus, no arc should exist between the supervisor and any transitions in $T_{\widetilde{O}}$. Additionally, there also exist events that are observable but uncontrollable, such as machine aging and malfunction, which correspond to the set of observable but uncontrollable transitions denoted as $T_{O\widetilde{C}}$. The firing of these transitions cannot be restricted and controlled through external behavior. Therefore, there should not exist any arc from a supervisor to these transitions.

In summary, transitions can be divided into three types $T = T_C \cup T_{\widetilde{O}} \cup T_{O\widetilde{C}}$ such that $T_C \cap (T_{\widetilde{O}} \cup T_{O\widetilde{C}}) = \emptyset$, where T_C denotes the set of controllable transitions (i.e., controllable and observable transitions), $T_{\widetilde{O}}$ denotes the set of unobservable transitions (i.e., uncontrollable and unobservable transitions), and $T_{O\widetilde{C}}$ is the set of observable but uncontrollable transitions. Since a controllable transition is also observable, $T_O = T_C \cup T_{O\widetilde{C}}$ is the set of observable transitions. Moreover, $T_{\widetilde{C}} = T_{\widetilde{O}} \cup T_{O\widetilde{C}}$ is the set of uncontrollable transitions. The following definition was presented in [34] for modeling RMCs with uncontrollable and unobservable events.

Definition 1: $N = (P_0 \cup P_R \cup P_A, T_C \cup T_{O\widetilde{C}} \cup T_{\widetilde{O}}, F, W)$ is a Petri net with uncontrollable and unobservable transitions that satisfies the following conditions [34]:

1) The places of N consist of idle places P_0, activity (operation) places P_A, and resource places P_R such that $P_A \neq \emptyset$; $P_R \neq \emptyset$; $P_0 \cap P_A = \emptyset$; and $(P_0 \cup P_A) \cap P_R = \emptyset$.
2) $W = W_A \cup W_R$, where $W_A : ((P_A \cup P_0) \times T) \cup (T \times (P_A \cup P_0)) \to \{0, 1\}$ and $W_R : (P_R \times T) \cup (T \times P_R) \to \mathbb{N}$.
3) $\forall r \in P_R$, there exists a unique minimal P-semiflow I_r such that $||I_r|| \cap P_0 = \emptyset$ and $I_r(r) = 1$. In addition, $P_A = \bigcup_{r\in P_R}(||I_r|| \setminus r)$.
4) $\forall r \in P_R, {}^\bullet r \cap r^\bullet = \emptyset$.
5) If resource places and their related arcs are removed from N, the remainder of N forms a strongly connected state machine.
6) $P_0^{\bullet\bullet} \cap P_R = {}^{\bullet\bullet}P_0 \cap P_R = \emptyset$.
7) An initial marking M_0 is said to be acceptable for N if: $\forall p \in P_0, M_0(p) \geq 1; \forall p \in P_A, M_0(p) = 0$; and $\forall r \in P_R, M_0(r) \geq max_{p\in||I_r||} I_r(p)$.

III. Division of some important markings

A marking $M \in \mathcal{R}(N, M_0)$ is called a deadlock marking if $\nexists t \in T$, t is enabled at M. On the other hand, if M is not a deadlock marking, $M_0 \notin \mathcal{R}(N, M)$, and $\forall M' \in \mathcal{R}(N, M)$ is not a deadlock marking, then M is called a livelock marking. The reachability graph of a Petri net can be divided into two zone: live-zone (LZ) and dead-zone (DZ). The former means the markings from each of which the system can reach the initial marking, and the latter consists of deadlocks, livelocks, and bad markings (which are doomed to deadlocks or livelocks). If the DZ is forbidden by a designed supervisor, then the underlying system will never enter the DZ.

For a Petri net with uncontrollable and unobservable transitions, admissible markings and first-met inadmissible markings are crucial to prevent the deadlock.

A. Admissible Markings

The underlying system may evolve into the DZ via the firing of an uncontrollable transition. So, the designed supervisor should keep the system from reaching not only the DZ but also the markings that may enter the DZ via the firings of uncontrollable transitions. Let

$$\mathcal{M}_A = \{M \in \mathrm{LZ} | \nexists t \in T_{\widetilde{C}} : M[t\rangle M' \wedge M' \notin \mathcal{M}_A\}.$$

denote the set of admissible markings. Note that this set $\mathcal{M}_A$ is defined recursively. Clearly, all markings in $\mathcal{M}_A$ are good markings from each of which the system can reach M_0. The remaining markings in $\mathcal{R}(N, M_0)$ are inadmissible, i.e.,

$$\mathcal{M}_{\widetilde{A}} = \mathcal{R}(N, M_0) \setminus \mathcal{M}_A.$$

As a consequence, the added supervisor should forbid all inadmissible markings and permit as many admissible markings as possible to maximize the behavioral permissiveness. The controlled system run in $\mathcal{M}_A$ and never runs outside it through the firing of any uncontrollable transition. The relationships of markings are shown in Fig. 2, where $\mathcal{M}_L$ (resp., $\mathcal{M}_{\widetilde{L}}$) means that the markings are within the constraints (resp., beyond the constraints).

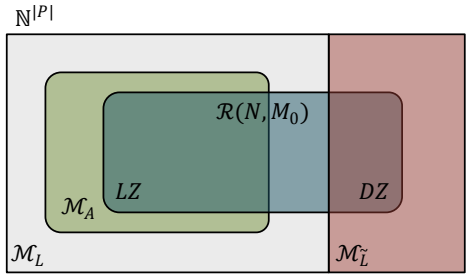


Fig. 2. Relationship of markings

B. First-Met Inadmissible Markings

In $\mathcal{R}(N, M_0)$, the first inadmissible markings from $\mathcal{M}_A$ to $\mathcal{M}_{\widetilde{A}}$ are called first-met inadmissible markings (FIMs), i.e.,

$$\mathcal{M}_{FIM} = \{M \in \mathcal{M}_{\widetilde{A}} | \exists M' \in \mathcal{M}_A, t \in T_C, s.t.\ M'[t\rangle M\}.$$

Obviously, $\mathcal{M}_{FIM} \subseteq \mathcal{M}_{\widetilde{A}}$ and any edge in the reachability graph from an admissible marking to an FIM is associated with a controllable transition. Hence, if a supervisor admits the markings in $\mathcal{M}_A$ and forbids all the markings in $\mathcal{M}_{FIM}$, then the controlled system will run in $\mathcal{M}_A$ and never enter $\mathcal{M}_{\widetilde{A}}$ via the firings of uncontrollable transitions.

As an example, consider the Petri net in Fig. 3, where $P_R = \{p_7, p_8\}$, and $P_A = \{p_1, p_2, p_5, p_6\}$, $T_{O\widetilde{C}} = \{t_7\}$, and $T_{\widetilde{O}} = \{t_3, t_8\}$. Since an unobservable transition is usually implicitly uncontrollable, we have $T_{\widetilde{C}} = \{t_3, t_7, t_8\}$. According to the definitions of $\mathcal{M}_A$ and $\mathcal{M}_{\widetilde{A}}$, all marking in $\mathcal{M}_{\widetilde{A}}$ are FIMs, i.e, $\mathcal{M}_{FIM} = \mathcal{M}_{\widetilde{A}} = 2p_1 + 2p_5$, and the remaining markings in constitute $\mathcal{M}_A$.

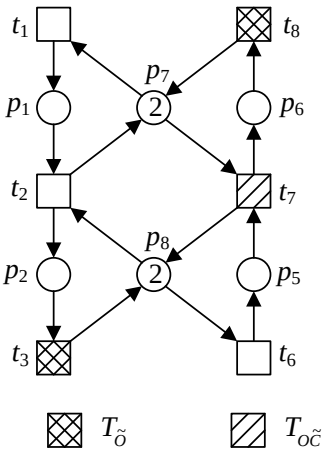


Fig. 3. A Petri net with $T_{O\widetilde{C}} = \{t_7\}$ and $T_{\widetilde{O}} = \{t_3, t_8\}$.

IV. Design of place invariants

A. Control Place Computation Based On Place Invariants

This subsection presents the method for designing supervisors which takes into account PIs [36]. The intention is to enforce the constraints by connecting control places to the Petri net. In the design phase of PIs, only activity places need to be considered [37].

Let $[N_p]$ be the incidence matrix of a plant's Petri net with n places and n' transitions, and $[N_c]$ be the incidence matrix describing the relationship between the control places and transitions. The incidence matrix $[N]$ of the controlled net can be expressed as:

$$[N] = \begin{bmatrix} N_p \\ N_c \end{bmatrix} \tag{1}$$

The control goal is to enforce the net to satisfy the following linear inequality constraint:

$$\sum_{i=1}^{n} l_i \times \mu_i \leq \beta \tag{2}$$

where μ_i denotes the marking of place p_i, l_i and β are non-negative integer constants. After a non-negative integer slack variable μ_c is introduced, constraint (2) becomes

$$\sum_{i=1}^{n} l_i \times \mu_i + \mu_c = \beta \tag{3}$$

where μ_c represents the marking of control place p_c usually called a controller. The PI defined in (3) and denoted by I must satisfy the equation $I^T[N] = \mathbf{0}^T$. For multiple constrains, the control problem can be stated in general as:

$$[L] \times \mu_p + \mu_c = b \tag{4}$$

where $[L]$ is an $n_c \times n$ integer matrix, μ_p is the marking vector of the Petri net, b is an $n_c \times 1$ integer vector, and n_c is the number of constraints of the inequality (2). Thus, the controller matrix $[N_c]$ can be computed by

$$[N_c] = -[L] \times [N_p] \tag{5}$$

The initial marking μ_{c_0} of p_c can be calculated by

$$\mu_{c_0} = b - [L] \times \mu_{p_0} \tag{6}$$

where μ_{p_0} is the initial marking. To forbid a given marking M, the following constraint must be satisfied:

$$\sum_{i \in \mathbb{N}_A} l_i \times \mu_i \leq \beta \tag{7}$$

where $\mathbb{N}_A = \{i | p_i \in P_A\}$ and

$$\beta = \sum_{i \in \mathbb{N}_A} l_i \times M(p_i) - 1 \tag{8}$$

B. Computation of Maximally Permissive Control Place

For Petri nets with uncontrollable and unobservable transitions, we need to make some improvements to the method reviewed in Section IV-A.

Specifically, a new method should be developed to forbid all FIMs and permit all admissible markings by designing some PIs. To forbid a marking $\mu' \in \mathcal{M}_{FIM}$, a PI is designed such that

$$\sum_{i \in \mathbb{N}_A} l_i \times \mu'(p_i) \geq \beta + 1 \tag{9}$$

Meanwhile, the PI should satisfies as follow to permit each marking $\forall \mu \in \mathcal{M}_A$.

$$\sum_{i \in \mathbb{N}_A} l_i \times \mu(p_i) \leq \beta \tag{10}$$

After the control place p_c is added, the designed PI can be expressed as

$$\sum_{i \in \mathbb{N}_A} l_i \times \mu_i + \mu_c = \beta \tag{11}$$

As a consequence, a marking satisfies (9), if and only if it violates (10). Therefore, we can achieve our control objective by designing the PI. By substituting (9) into (10), we obtain the following constraint that permits all admissible markings and forbids all inadmissible ones:

$$\sum_{i \in \mathbb{N}_A} l(i) \times (\mu(p_i) - \mu'(p_i)) \leq -1 \tag{12}$$

Equation 12 forbids an FIM and permits all admissible markings. However, there are often too many FIMs and admissible markings, especially for large Petri net models. In [21], the following definitions with respect to vector covering were proposed, which can minimize the markings considered in the PI design.

C. Vector Covering

Definition 2: $\forall M, M' \in \mathcal{R}(N, M_0)$, we say M covers M', denotes as $M \succcurlyeq M'$, if $\forall p \in P_A$, $M(p) \geq M'(p)$.

If $M \succcurlyeq M'$ and (9) is satisfied, indicating that the designed PI forbids M', then the PI also forbids M. Similarly, if $M \succcurlyeq M'$ and (10) is satisfied, indicating that the PI permits M, then it also permits M'.

Definition 3: Let $\mathcal{M}$ be a set of markings. $\mathcal{M}^*$ is called a minimal covered set of $\mathcal{M}$ if

1) $\mathcal{M}^* \subseteq \mathcal{M}$;
2) $\forall M \in \mathcal{M}$, $\exists M' \in \mathcal{M}^*$ such that $M \succcurlyeq M'$; and
3) $\forall M \in \mathcal{M}^*$, $\nexists M'' \in \mathcal{M}^*$ such that $M \neq M''$ and $M \succcurlyeq M''$.

Definition 4: Let $\mathcal{M}$ be a set of markings. $\mathcal{M}^*$ is called a maximal covering set of $\mathcal{M}$ if

1) $\mathcal{M}^* \subseteq \mathcal{M}$;
2) $\forall M \in \mathcal{M}$, $\exists M' \in \mathcal{M}^*$ such that $M' \succcurlyeq \mu$; and
3) $\forall M \in \mathcal{M}^*$, $\nexists M'' \in \mathcal{M}^*$ such that $M \neq M''$ and $M'' \succcurlyeq M$.

V. Deadlock Prevention Approach

The following sequence of steps provide guidelines for developing our proposed approach of deadlock prevention:

1) Decompose the system Petri net model via shared transition.
2) Design a liveness-enforcing supervisor for each subnet if necessary.
3) Explore the cause of deadlocks when merging controlled subnets.
4) Compute a liveness-enforcing supervisor for the merged net.
5) Perform constraint transformation and obtain the final supervisor.

A. Decomposition and Merging of Petri Nets

For a Petri net of Definition 1, the idle places do not use any resource place, hence they do not cause any deadlock, and can be ignored. In the following, we use $N = (P_A \cup P_R, T_C \cup T_{O\widetilde{C}} \cup T_{\widetilde{O}}, F, W)$ to represent the Petri net. In this paper, net decomposition means that a Petri net is divided into two subnets via shared transitions. We use T_s to denote the set of shared transitions of decomposed subnets.

Definition 5: Let (N, M_0) be an Petri net of Definition 1 with $N = (P_A \cup P_R, T_C \cup T_{O\widetilde{C}} \cup T_{\widetilde{O}}, F, W)$. Marked Petri nets (N^1, M_0^1) and (N^2, M_0^2) with $N^1 = (P_A^1 \cup P_R^1, T^1, F^1, W^1)$ and $N^2 = (P_A^2 \cup P_R^2, T^2, F^2, W^2)$ are two subnets of (N, M_0) with shared transitions T_s if

1) $P_A^1 \cup P_A^2 = P_A$, $P_A^1 \cap P_A^2 = \emptyset$, $P_R^1 \cup P_R^2 = P_R$, $P_R^1 \cap P_R^2 = \emptyset$, and $\forall p \in P_A^i \cup P_R^i$, $M_0^i(p) = M_0(p)$,$(i = 1, 2)$;
2) $T^1 \cup T^2 = T$ and $T^1 \cap T^2 = T_s \neq \emptyset$;
3) $F^1 \cup F^2 = F$, $F^1 \cap F^2 = \emptyset$, and $\forall f \in F^i$, $W_f^i(f) = W(f)$ $(i = 1, 2)$.
4) N^1 and N^2 are strongly connected.

where T_s, T^1 and T^2 are all in the set of $T_C \cup T_{O\widetilde{C}} \cup T_{\widetilde{O}}$. Note that, the subnet can be decomposed continuously via another shared transition if necessary.

Definition 6: Let $(N_C^1, M_0^{C_1})$ and $(N_C^2, M_0^{C_2})$ with $N_C^1 = (P_A^1 \cup P_R^1 \cup P_C^1, T^1, F_C^1, W_C^1)$ and $N_C^2 = (P_A^2 \cup P_R^2 \cup P_C^2, T^2, F_C^1, W_C^1)$ be two controlled subnets with shared transitions T_s. Marked Petri net (N^m, M_0^m) with $N^m = (P_A^m \cup P_R^m, T^m, F^m, W^m)$ is called the merged net of N_C^1 and N_C^2 via T_s if

1) $P_A^m = P_A^1 \cup P_A^2$, $P_R^m = P_R^1 \cup P_R^2 \cup P_C^1 \cup P_C^2$, and $\forall p \in P_A^i \cup P_R^i \cup P_C^i$, $M_0^m(p) = M_0^{C_i}(p)$ $(i = 1, 2)$;
2) $T^m = T^1 \cup T^2$;
3) $F^m = F_C^1 \cup F_C^2$ and $\forall f \in F_C^i$, $W^m(f) = W_f^i(f)$ $(i = 1, 2)$.

As an example, consider an RMC whose layout is depicted in Fig. 4. The system has two shared robots R_1 and R_2, one machines M_1, two loading buffers I_1 and I_2, and two unloading buffers O_1 and O_2. Two part types J_1 and J_2 are produced in the system. Specifically, J_1 is taken from I_1 by R_1, processed by M_1, held by R_2, and moved to O_1; J_2 is taken from I_2 by R_2, processed by M_1, held by R_1, and moved to O_2. That is, the production sequences are:

$$J_1 : I_1 \to R_1 \to M_1 \to R_2 \to O_1$$

$$J_2 : I_2 \to R_2 \to M_1 \to R_1 \to O_2$$

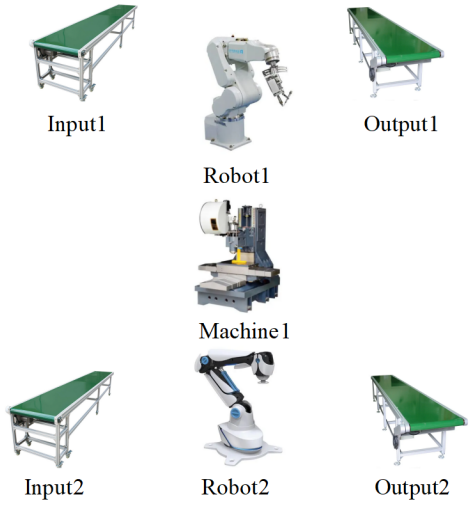


Fig. 4. The layout of an RMC.

The Petri net model of this RMC is shown in Fig. 5 with $P_A = \{p_1$ - $p_6\}$, $P_R = \{p_7, p_8, p_9\}$, and $T = \{t_1$ - $t_8\}$. Its reachability set has 203 markings including 3 deadlocks and 192 are in $\mathcal{M}_A$.

By Definition 5, this marked Petri net can be decomposed into two subnets via shared transitions. For example, we decompose it into N^1 and N^2 via $T_s = \{t_6, t_{11}\}$, as shown

in Fig. 6. As a result, we have $P_A^1 = \{p_1, p_2, p_5, p_6\}$, $P_A^2 = \{p_3, p_4\}$, $P_R^1 = \{p_7, p_8\}$, $P_R^2 = \{p_9\}$, $T_1 = \{t_1, t_2, t_3, t_6, t_7, t_8\}$, and $T_2 = \{t_3$ - $t_6\}$. In addition, N^1 has 35 reachable markings including only one deadlock and N^2 is deadlock free with 6 reachable markings.

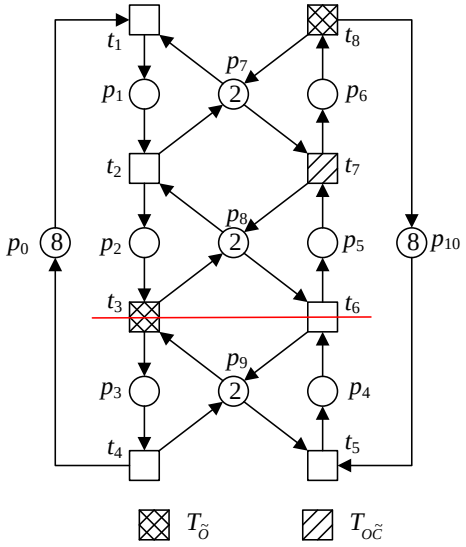
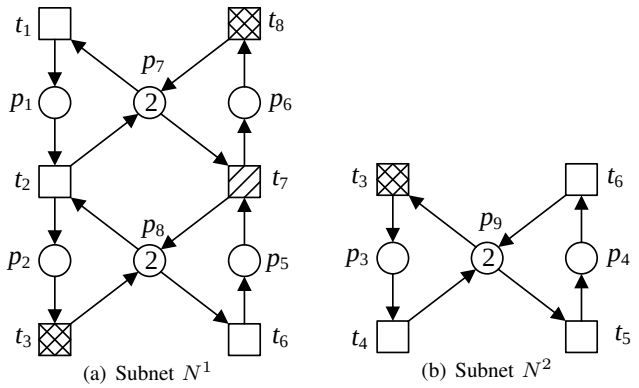


Fig. 5. The Petri net model with $T_{O\tilde{C}} = \{t_7\}$ and $T_{\tilde{O}} = \{t_3, t_8\}$.



(a) Subnet N^1 (b) Subnet N^2

Fig. 6. Subnets of the Petri net model in Fig. 5.

B. Deadlock Prevention for Subnets

This subsection presents the method originally developed in [34] to synthesize an admissible and structure-minimal supervisor for each subnet by solving a multiobjective ILP.

The first objective of the supervisor synthesis is to minimize the number of control places. Each control place of the supervisor is associated with a designed PI. Furthermore, each designed PI permits all admissible markings and may forbid more than one FIM in $\mathcal{M}_{FIM}^*$. Thus, the number of control places, each of which corresponds to a designed PI, can be minimized as follows. Let I_j be the PI that forbids the jth marking in $\mathcal{M}_{FIM}^*$, where $j \in \mathbb{N}_{FIM}^* = \{i | \mu_i \in \mathcal{M}_{FIM}^*\}$. We use $q_j \in \{0, 1\}$ to indicate whether or not I_j is selected to construct a control place, so that $q_j = 1$ represents the selection of I_j and $q_j = 0$ represents it is not selected. Then, the following objective can be used to produce the fewest control places in the supervisor.

$$mine_1 = \sum_{j \in \mathbb{N}_{FIM}^*} q_j$$

The second objective is to minimize the number of arcs of the supervisor. For the jth $(j = 1, 2, \cdots)$ control places, we use $u_{j,y} \in \{0, 1\}$ to denote whether or not there is an arc that connects a control place p_{c_j} to a transition $t_y \in T$ in the controlled net. Let $[N_c](j, y) = W(t_y, p_{c_j}) - W(p_{c_j}, t_y)$. According to [36], $[N_c] = -[L] \times [N_p]$ where $[L]$ is an $n_c \times |P|$ nonnegative integer matrix that denotes the coefficients of the m_c designed PIs. Then we have $[N_c](j, y) = -\sum_{i \in \mathbb{N}_A} l_j(i) \times [N_p](i, y)$. Thus, the inequality between $[N_c]$ and the arc can be given as:

$$-\sum_{i \in \mathbb{N}_A} l_j(i) \times [N_p](i, y) \geq -\Gamma \times u_{j,y} - \Gamma \times (1 - q_j),$$

$$\forall j \in \mathbb{N}_{FIM}^*, \forall y \in \mathbb{N}_T$$

where $\mathbb{N}_T = \{i | t_i \in T\}$, and Γ is a positive integer that is chosen big enough. This constraint denotes that if $q_j = 1$ and $[N_c](j, y) \leq -1$, then $u_{j,y} = 1$ which implies an arc from p_{c_j} to t_y. Similarly, a set of variables $v_{j,y} \in \{0, 1\}$ is used to denote whether or not an arc exists from t_y to p_{c_j}, and then we have

$$-\sum_{i \in \mathbb{N}_A} l_j(i) \times [N_p](i, y) \geq -\Gamma \times v_{j,y} - \Gamma \times (1 - q_j),$$

$$\forall j \in \mathbb{N}_{FIM}^*, \forall y \in \mathbb{N}_T$$

It means that if $q_j = 1$ and $[N_c](j, y) \geq 1$, then $v_{j,y} = 1$, which implies an arc from t_y to p_{c_j}. Therefore, the following objective produces the fewest arcs in the supervisor.

$$mine_2 = \sum_{j \in \mathbb{N}_{FIM}^*} \sum_{y \in \mathbb{N}_T} (u_{j,y} + v_{j,y})$$

Furthermore, due to the characteristics of unobservable transitions, there should not exist any arc between the supervisor and unobservable transitions. Otherwise, the firings of the unobservable transitions will change the tokens in the supervisor and the unobservable events will be detected or measured by the device that implement the supervisor. Thus, we have

$$u_{j,y} + v_{j,y} = 0, \forall j \in \mathbb{N}_{FIM}^*, \forall y \in \mathbb{N}_{T_{\tilde{O}}}$$

where $\mathbb{N}_{T_{\tilde{O}}} = \{i | T_i \in T_{\tilde{O}}\}$. Similarly, there should not exist any arc from supervisor to uncontrollable transitions; Thus, we have

$$u_{j,y} = 0, \forall j \in \mathbb{N}_{FIM}^*, \forall y \in \mathbb{N}_{T_{\tilde{C}}}$$

By combining them, we obtain the following ILP formulation, denoted as a supervisor synthesis with uncontrollable and unobservable transitions ($\mathrm{S^2U^2T}$) problem, so as to design admissible and structurally minimal supervisors for each subnet:

$$\mathrm{S^2U^2T}:$$

$$lex\ min\{e_1, e_2\}$$

$$s.t.$$

$$\sum_{i\in\mathbb{N}_A} l(i)\times(\mu_l(p_i)-\mu_j(p_i))\le -1, \forall \mu_j\in\mathcal{M}^*_{FIM}, \forall \mu_l\in\mathcal{M}^*_A \tag{13}$$

$$-\sum_{i\in\mathbb{N}_A} l_j(i)\times[N_p](i,y)\ge -\Gamma\times u_{j,y}-\Gamma\times(1-q_j), \forall j\in\mathbb{N}^*_{FIM}, \forall y\in\mathbb{N}_T \tag{14}$$

$$\sum_{i\in\mathbb{N}_A} l_j(i)\times[N_p](i,y)\ge -\Gamma\times v_{j,y}-\Gamma\times(1-q_j), \forall j\in\mathbb{N}^*_{FIM}, \forall y\in\mathbb{N}_T \tag{15}$$

$$u_{j,y}+v_{j,y}=0, \forall j\in\mathbb{N}^*_{FIM}, \forall y\in\mathbb{N}_{T_{\tilde{O}}} \tag{16}$$

$$u_{j,y}=0, \forall j\in\mathbb{N}^*_{FIM}, \forall y\in\mathbb{N}_{T_{\tilde{C}}} \tag{17}$$

$$\sum_{i\in\mathbb{N}_A} l_j(i)\times((\mu_k(p_i)-\mu_j(p_i))\ge -\Gamma\times(1-f_{j,k}), \forall \mu_j,\mu_k\in\mathcal{M}^*_{FIM}, j\ne k \tag{18}$$

$$f_{j,k}\le q_j, \forall j,k\in\mathbb{N}^*_{FIM}, j\ne k \tag{19}$$

$$q_j+\sum_{k\in\mathbb{N}^*_{FIM},k\ne j} f_{k,j}\ge 1, \forall\in\mathbb{N}^*_{FIM} \tag{20}$$

$$l_j(i)\in\{0,1,2,...\}, \forall i\in\mathbb{N}_A, \forall j\in\mathbb{N}^*_{FIM}$$

$$f_{j,k}\in\{0,1\}, \forall j,k\in\mathbb{N}^*_{FIM}, j\ne k$$

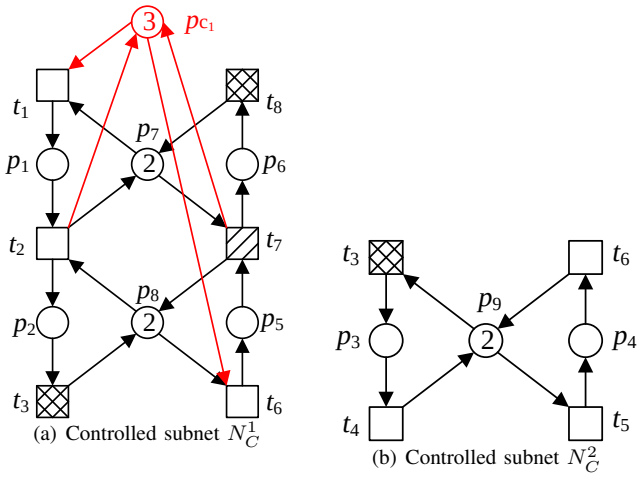
$$q_j\in\{0,1\}, \forall j\in\mathbb{N}^*_{FIM}$$

$$u_{j,y},v_{j,y}\in\{0,1\}, \forall j\in\mathbb{N}^*_{FIM}, \forall y\in\mathbb{N}_T$$

The above $\mathrm{S^2U^2T}$ has $|\mathcal{M}^*_{FIM}|\times(2|\mathcal{M}^*_{FIM}|+2|\mathbb{N}_T|+|\mathcal{M}^*_A|+|\mathbb{N}_{T_{\tilde{O}}}|+|\mathbb{N}_{T_{\tilde{C}}}|-1)$ constraints and $|M^*_{FIM}|\times(|M^*_{FIM}|+2|\mathbb{N}_T|+|\mathbb{N}_A|)$ variables. In the ILP, equations (18)-(20) are used to calculate all needed PIs [38]. The coefficients $l_j(i)$ of each designed PI are nonnegative, i.e., all designed PIs are P-semiflows, and $\mathrm{S^2U^2T}$ minimizes its objectives under this restriction. To summarize, we can design a liveness-enforcing Petri net supervisor, which is admissible and structurally minimal for each subnet using $\mathrm{S^2U^2T}$.

As an example, reconsider the subnets N^1 in Fig. 6, $T_{\tilde{C}}=\{t_7\}$ and $T_{\tilde{O}}=\{t_3,t_8\}$. As can be seen from the results in section V-A, its reachability graph contains 35 reachable markings, 34 of which are admissible markings, and the remaining one are FIMs. In addition, by Definition 3 and Definition 4, $\mathcal{M}^*_A=\{M_{23},M_{26},M_{28},M_{30},M_{31},M_{33},M_{34}\}$ and $\mathcal{M}^*_{FIM}=\{M_{17}\}$.

Then, an $\mathrm{S^2U^2T}$ with $1\times(2\times1+2\times6+7+2+1-1)=23$ constraints and $1\times(1+2\times6+4)=17$ variables is formulated. By solving the ILP, a PI that forbids M_{17} in $\mathcal{M}^*_{FIM}$ is selected to construct a supervisor consisting of a control place p_{c_1} and 4 arcs such that ${}^\bullet p_{c_1}=\{t_1,t_6\}, p_{c_1}^\bullet=\{t_2,t_7\}$, and $M_0(p_{c_1})=3$. After adding the supervisor to the plant net, we obtain a controlled net as shown in Fig. 7. The supervisor is admissible and deadlock free since all admissible makings are reachable by the controlled net system and all inadmissible markings, including one deadlock are prohibited. In addition, the supervisor is guaranteed to have the fewest control places and arcs, (i.e., the supervisor is structurally minimal). Subnet N^2 is deadlock free. Therefore, we do not need to add any control place for it.



(a) Controlled subnet N^1_C (b) Controlled subnet N^2_C

Fig. 7. Controlled subnets of N^1 and N^2 in Fig. 6.

For $i=1,2$, we use $N^i_C=(P^i_A\cup P^i_R\cup P^i_C,T_i,F^i_C,W^i_C)$ to denote the controlled subnet of N^i, where P^i_C is the set of control places added to subnet N^i. For convenience, a subnet N^i ($i\in\{1,2\}$) is also denoted as N^i_C if it is deadlock free by itselt. For example, as shown in Fig. 7, the controlled subnets of N^1 and N^2 are denoted as N^1_C and N^2_C, respectively.

C. Subnet Merging Deadlock Analysis

Unfortunately, the deadlock freeness of both controlled subsets cannot guarantee that the merged net is also deadlock free. For example, the merged net N^m has two deadlock markings $M_{d_1}=2p_1+p_2+2p_4+p_5$ and $M_{d_2}=2p_1+p_2+p_4+p_{c_1}$ with $M_{d_1},M_{d_2}\in\mathcal{R}(N^m,M^m_0)$.

In the following, we analyze the root of generating deadlocks when merging two controlled subnets.

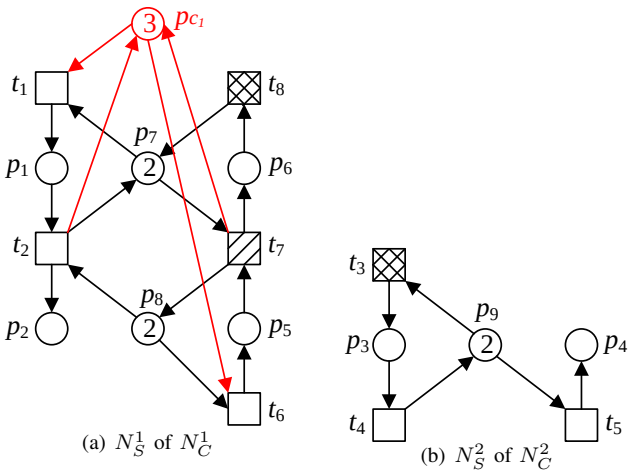
Definition 7: Let T^i_{out} be the set of output transitions of a controlled subnet $N^i_C=(P^i_C,T^i,F^i_C,W^i_C)$, $i=1,2$. A marking $M'\in\mathcal{R}(N^i_C,M_0^{C_i})$ is called a lethal state (LS) if there does not exist a transition $t\in T^i\backslash T^i_{out}$ such that $M'[t\rangle$ holds, $i=1,2$.

By Definition 7, if M' is an LS, then only $t\in T^i_{out}$ might be enabled at M'. Reconsider Fig. 7 which shows two controlled subnets N^1_C and N^2_C. For N^1_C, $T^1_{out}=\{t_3\}$. Markings $M_1=2p_1+p_2+p_5$ and $M_2=2p_1+2p_2+p_{c_1}$ are two LSs of N^1_C, and only t_3 is enabled either at M_1 or at M_2. Similarly, for N^2_C, $T^2_{out}=\{t_6\}$. The marking $M_3=2p_4$ is an LS, at which only t_6 is enabled.

During the synthesis process of the subnet, the output transitions are prohibited which lead to the deadlock, so we need to identify all LSs and the markings that will definitely cause LS. Thus, we can remove the output transitions T^i_{out} from N^i_C to find LSs and the markings from which an LS is definitely reachable.

Definition 8: The net (N^i_S,M^S_0) with $N^i_S=(P^i_C,T^i\setminus T^i_{out},F^i_C,W^i_C)$ is called a single-direction subnet of N^i_C ($i=1,2$).

The net N^i_S is the resulting net by removing the output transitions and the related arcs from N^i_C. Hence the dead markings of N^i_S are LSs of N^i_C. Fig. 8 shows N^1_S and N^2_S of N^1_C and N^2_C.



(a) N_S^1 of N_C^1 (b) N_S^2 of N_C^2

Fig. 8. Single-direction subnet of N_C^1 and N_C^2.

Definition 9: If there exists a sequence of transitions $\sigma = t_1, t_2 \cdots t_n$ and markings $M_1, M_2, \cdots, M_n$ such that $M_1[t_1\rangle M_2 \cdots M_n[t_n\rangle M_1$ holds, i.e., $M[\sigma\rangle M'$, M_1 is called a cycle marking and $t_0 t_1 \cdots t_n$ are called cycle transitions of M_1. For a cycle marking M, the set of its cycle transitions is denoted by $T_c(M)$.

Proposition 1: Let N_S^i be the single-direction subnet of a controlled subnet N_C^i. A marking $M' \in \mathcal{R}(N_S^i, M_0^S)$ satisfies one of the following statements:

1) $M_0^S \in \mathcal{R}(N_S^i, M')$;
2) $M_0^S \notin \mathcal{R}(N_S^i, M')$, $\exists M'' \in \mathcal{R}(N_S^i, M')$, $T_c(M'') \neq \emptyset$ and $T_{in}^i \subseteq T_c(M'')$;
3) $M_0^S \notin \mathcal{R}(N_S^i, M')$, $\exists \mathcal{C} = \{M | M \in \mathcal{R}(N_S^i, M'), T_c(M) \neq \emptyset, \forall M'' \in \mathcal{C}, T_{in}^i \nsubseteq T_c(M'')\}$; and
4) $M_0^S \notin \mathcal{R}(N_S^i, M')$, $\forall M'' \in \mathcal{R}(N_S^i, M'), T_c(M'') = \emptyset$.

For Proposition 1, a marking M satisfying Statement 1) can reach the initial marking; a marking M satisfying Statement 2) cannot reach the initial marking and at least one of its successors is a cycle marking whose cycle transitions contain all the input transitions; a marking M satisfying Statement 3) cannot reach the initial marking and at least one of its successors is a cycle marking but whose cycle transitions do not contain all the input transitions; and a marking M satisfying Statement 4) cannot reach the initial marking and all its successors are not cycle markings.

These four types of markings are collectively referred to as single direction marking (SDM). For $i = 1, 2$ and $j = 1$ - 4, the jth set of SDMs of N_C^i is denoted by $\mathcal{M}_{SDM_j}^i$. The marking $M \in \mathcal{M}_{SDM_{1,2}}^i$ satisfies a property that when M meets any SDM of the other controlled subnet, the shared transitions are live. Therefore, it is still live when merging the two controlled subnets. The marking $M \in \mathcal{M}_{SDM_3}^i$ leads to that there exists $t \in T_{in}^i$ such that t is always disabled at $M' \in \mathcal{R}(N_S^i, M)$; hence it may be stuck in an endless loop when it meets an $M \in \mathcal{M}_{SDM_{3,4}}^i$ of the other subnet in the merged net, i.e., a livelock in the merged net. The marking $M \in \mathcal{M}_{SDM_4}^i$ means that it will definitely reach an LS. Hence, it will lead to deadlocks when it meets an $M \in \mathcal{M}_{SDM_{3,4}}^i$ of the other controlled subnet. An algorithm to find the SDMs is as follows.

Algorithm 1 Calculation of SDM.

Input: A controlled Petri net(N, M_0).
Output: The set of single direction markings $\mathcal{M}_{SDM_1}$, $\mathcal{M}_{SDM_2}, \mathcal{M}_{SDM_3}, \mathcal{M}_{SDM_4}$.
$E := \emptyset \cup \{M_0\}(E$ is a set of markings)
$\mathcal{M}_{SDM_1}$,$\mathcal{M}_{SDM_2}$,$\mathcal{M}_{SDM_3}$,$\mathcal{M}_{SDM_4} := \emptyset$
for $(i = 0; i < |E|; i++)$ **do**
  for $(j = 0; j < |T|; j++)$ **do**
    if $(M_i[t_j\rangle)$ **then**
      $M_i[t_j\rangle M$;
      if $(M \notin E)$ **then**
        $(E := E \cup \{M\})$;
        if (M can reach the initial marking) **then**
          $(\mathcal{M}_{SDM_1} := \mathcal{M}_{SDM_1} \cup \{M\})$;
        else if ($M's$ successors are cycle markings) **then**
          if (cycle markings contain all the input transitions) **then**
            $\mathcal{M}_{SDM_2} := \mathcal{M}_{SDM_2} \cup \{M\}$;
          else
            $\mathcal{M}_{SDM_3} := \mathcal{M}_{SDM_3} \cup \{M\}$;
          end if
        else
          $\mathcal{M}_{SDM_4} := \mathcal{M}_{SDM_4} \cup \{M\}$;
        end if
      end if
    end if
  end for
end for

For instance, in Fig. 9, the number k represents the kth marking. Furthermore, the green circle, blue circle, and red circle represent the set of $\mathcal{M}_{SDM_1}^i$, $\mathcal{M}_{SDM_2}^i$, and $\mathcal{M}_{SDM_4}^i$, respectively (note that in this example, there is no marking belongs to $\mathcal{M}_{SDM_3}^i$). Thus, we can get that $\mathcal{M}_{SDM_1}^1 = \{M_0, M_2, M_6, M_7, M_{13}, M_{19}, M_{20}, M_{26}, M_{31}\}$. The markings in it can reach the initial markings M_0^1 and will not lead to deadlock. $\mathcal{M}_{SDM_2}^1 = \{M_1, M_3, M_4, M_5, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{16}, M_{17}, M_{18}, M_{23}, M_{24}, M_{25}, M_{29}, M_{30}, M_{33}\}$. The markings in it or their successors are cycle markings whose cycle transitions contain all input transitions $T_{in}^1 = \{t_6\}$. Therefore, all transition of T_{out}^2 $(T_{out}^2 = T_{in}^1)$ of N_C^2 can fire when an SDM of N_C^2 meets these markings. Hence, these markings do not lead to deadlocks. Specifically, markings M_3 and M_9 are not cycle markings but their successors contain cycle markings whose cycle transitions contain all input transitions T_{in}^1. The set of markings satisfying Statement 3) in Proposition 1 does not exist in this Petri net, but may exist in some Petri net models. These markings or their successors are cycle markings but their cycle transitions do not contain all input transitions T_{in}^1. Thus, deadlocks may appear when merging subnets. $\mathcal{M}_{SDM_4}^1 = \{M_{14}, M_{15}, M_{21}, M_{22}, M_{27}, M_{28}, M_{32}\}$. The markings in it are LSs or they will definitely reach LSs. Clearly, M_{22} and M_{27} are LSs, and markings $M_{14}, M_{15}, M_{21}, M_{28}, M_{32}$ will definitely reach M_{22} or M_{27}.

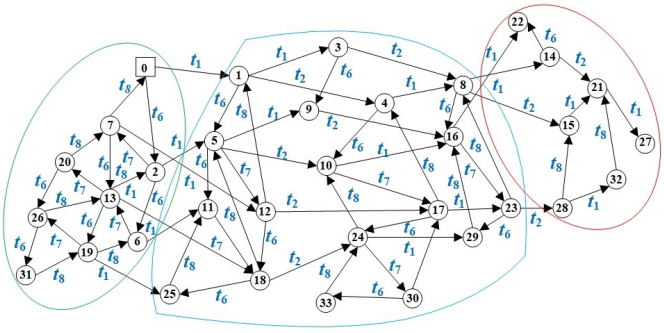


Fig. 9. Reachability graph of N_S^1.

D. Deadlock Prevention for Merged Net

The cause of deadlocks in the merged net is the meeting of $M \in \mathcal{M}^i_{SDM_{3,4}}$ in the two controlled subnets. Thus, if we can prevent $M \in \mathcal{M}^i_{SDM_{3,4}}$ of the two controlled subnets from meeting each other, we can prevent the deadlocks from occurring. In this subsection, a way of preventing the merging-induced deadlocks will be reported.

Proposition 2: Let $M', M'' \in \mathcal{M}^i_{SDM_{3,4}}$ in controlled subnets N_C^1 and N_C^2, respectively. If $M' + M'' \in \mathcal{R}(N^m, M_0^m)$, $M' + M''$ can be prohibited by a PI if $\sum_{i\in\mathbb{N}'_A} \mu_i \leq \sum_{i\in\mathbb{N}'_A} (M'(p_i) + M''(p_i)) - 1$, where $\mathbb{N}'_A = \{i | p_i \in (p_A^1 \cap |M'|) \cup (p_A^2 \cap |M''|)\}$.

Corollary 1: Let $\mathcal{M}^{1*}_{SDM_{3,4}}$ and $\mathcal{M}^{2*}_{SDM_{3,4}}$ be the minimal covered sets of $\mathcal{M}^1_{SDM_{3,4}}$ and $\mathcal{M}^2_{SDM_{3,4}}$, respectively. If all markings $M_{i,j} \in \{M_i + M_j | M_i \in \mathcal{M}^{1*}_{SDM_{3,4}}, M_j \in \mathcal{M}^{2*}_{SDM_{3,4}}\}$ are prohibited, any markings in $\mathcal{M}_{SDM_3}$ and $\mathcal{M}_{SDM_4}$ of N_C^1 and N_C^2 are prohibited when merging N_C^1 and N_C^2.

By using decomposition technique, we decompose a net into two subnets. Evidently, the computation of the reachability graph of each subnet is much smaller than that of the whole one. Therefore, the computational cost is greatly reduced since we need only compute the reachability graphs of the two subnets so that we can use ILP to add supervisor even for a large net. It is noted that the number of reachable markings increases exponentially with respect to the size of a net, which always suffers from the state explosion problem. In addition, a minimal covered set of $\mathcal{M}^i_{SDM_{3,4}}$ can be obtained by using the vector covering method [21], which can remove the redundant markings. This technique can reduce the computational cost for solving inequality constraints and avoid the redundant control places. Thus, this method is computationally competitive.

In the two controlled subnets shown in Fig. 7, by Definition 3, we can get their respective minimal covered sets of $\mathcal{M}^i_{SDM_{3,4}}$, $\mathcal{M}^{1*}_{SDM_{3,4}} = \{2p_1+p_2+p_8+p_{c_1}, 2p_2+2p_7+3p_{c_1}\}$ and $\mathcal{M}^{2*}_{SDM_{3,4}} = \{2p_4\}$.

By Proposition 2 and Corollary 1, to prohibits markings in $\mathcal{M}^i_{SDM_{3,4}}$ of N_C^1 and N_C^2, the following constraints must be satisfied:

$$\mu_1 + \mu_2 + \mu_4 \leq 4;$$

$$\mu_2 + \mu_4 \leq 3;$$

As a consequence, we can get two constraints, $(w_1, k_1) = ((1,1,0,1,0,0),4)$ and $(w_2, k_2) = ((0,1,0,1,0,0),3)$.

Based on the above constraints, we can calculate the controller using the PIs. However, the arc of the directly calculated controller may be connected to uncontrollable and unobservable transitions. Therefore, we need to convert the constraints into admissible ones, which does not need to calculate the reachability graph of the entire network.

E. Constraint Transformation

Proposition 3: The constraint (w, k) is admissible if and only if $\forall t \in T_{\widetilde{C}}, \lambda(t) \leq 0$ [32].

Proposition 4: The constraint (w, k) is admissible if and only if $\forall t \in T_{\widetilde{O}}, \lambda(t) = 0$.

In which, $\lambda(t) = w \times M' - w \times M$, $M[t\rangle M'$.

Proposition 5: For a Petri net and an inadmissible constraint (w, k), if the constraint (w^*, k) is admissible, then $\forall p \in P, w^*(p) \geq w(p)$.

Proposition 6: Given 2 admissible constraints (w_1, k) and (w_2, k). If $\forall p \in P, (w_1, k) \geq (w_2, k)$, then $(w_1, k) \subseteq (w_2, k)$.

Usually, the admissible linear constraints that meet the above conditions are not unique. In order to maximize the permissiveness of system behavior (i.e., ensure that the number of markings defined by the admissible linear constraints is as large as possible), according to Proposition 3, the objective function is taken as:

$$min \sum_{i=1}^{|P|} w^*(p_i)$$

.

By using the above Propositions 3-6, the forbidden state problem can be transformed into an ILP problem. By solving this ILP problem, we obtain the admissible linear constraints. The constraint transformation process is shown in Algorithm 2.

Algorithm 2 Constraint transformation

1: **Input:** A Petri net(N, M_0) and constraint (w, k);
2: **Output:** Allowed constraint (w^*, k);
3: If $\forall t \in T_{O\widetilde{C}}$, $\lambda(t) \leq 0$, or $\forall t \in T_{\widetilde{O}}$, $\lambda(t) = 0$, then $w^* = w$ and turn to Step 7;
4: Abstract the constraint transformation problem into a linear programming problem;
5: Obtain the objective function based on Proposition 3 $min \sum_{i=1}^{|P|} w^*(p_i)$;
6: Obtain constraint conditions by Propositions 1 and 2:

$$\begin{cases} \forall p \in P, w^*(p) \geq w(p) \\ \forall t \in T_{O\widetilde{C}}, \lambda(t) = \sum_{p\in t^\bullet} w^*(p) - \sum_{p\in {}^\bullet t} w^*(p) \leq 0 \\ \forall t \in T_{\widetilde{O}}, \lambda(t) = \sum_{p\in t^\bullet} w^*(p) - \sum_{p\in {}^\bullet t} w^*(p) = 0 \end{cases}$$

7: Calculate the linear programming problem, get the constraint (w^*, k);

According to the Propositions 3 and 4, it is easy to know that both constraints are inadmissible ones. For marking $[2,1,0,2,0,0]^T$, $\lambda(t_3) = w_1([2,0,1,2,0,0]^T) -$

$w_1([2,1,0,2,0,0]^T) = 4-5 \neq 0$, For marking $[0,2,0,2,0,0]^T$, $\lambda(t_3) = w_2([0,1,1,2,0,0]^T) - w_2([0,2,0,2,0,0]^T) = 3-4 \neq 0$. By converting the constraint transformation problem into an ILP problem, we perform Step 6 in Algorithm 2, and can obtain the following constraint conditions:

$$min \sum_{i=1}^{6} w^*(p_i)$$

$$\begin{cases} w_1^*(p_1) \geq 1 \\ w_1^*(p_2) \geq 1 \\ w_1^*(p_3) \geq 0 \\ w_1^*(p_4) \geq 1 \\ w_1^*(p_5) \geq 0 \\ w_1^*(p_6) \geq 0 \\ (t_6) = w_1^*(p_3) - w_1^*(p_2) = 0 \end{cases} \quad \begin{cases} w_2^*(p_1) \geq 0 \\ w_2^*(p_2) \geq 1 \\ w_2^*(p_3) \geq 0 \\ w_2^*(p_4) \geq 1 \\ w_2^*(p_5) \geq 0 \\ w_2^*(p_6) \geq 0 \\ \lambda(t_6) = w_2^*(p_3) - w_2^*(p_2) = 0 \end{cases}$$

TABLE I
CONTROL PLACE FOR THE MERGED NET N^m

i	PI_i	${}^\bullet P_{c_i}$	$P_{c_i}^\bullet$	$M_0(P_{c_i})$
2	$\mu_1+\mu_2+\mu_4 \leq 4$	t_3, t_6	t_1, t_5	4
3	$\mu_2+\mu_4 \leq 3$	t_3, t_6	t_2, t_5	3

LINGO is adopted to apply the algorithm of constraint transformation, we get 2 admissible constraints: $((1,1,1,1,0,0),4)$ and $((0,1,1,1,0,0),3)$. Compared with the previous constraints, the new constraints are more strict with the system since the markings satisfy these constraints must satisfy the previous constraints. After adding the supervisor, the Petri net is deadlock free and has 175 reachable markings.

Algorithm 3 Deadlock prevention approach

1: **Input:** A Petri net with uncontrollable and unobservable transitions $N = (P_0 \cup P_R \cup P_A,\ T_C \cup T_{O\widetilde{C}} \cup T_{\widetilde{O}}, F, W)$;
2: **Output:** A deadlock-free controlled net with an admissible supervisor;
3: Decompose Petri net N into two subnets $N^1 = (P_A^1 \cup P_R^1, T^1, F^1)$ and $N^2 = (P_A^2 \cup P_R^2, T^2, F^2)$ according to Definition 5;
4: Design a liveness-enforcing supervisor for each subnet using S^2U^2T;
5: Calculate $\mathcal{M}_{SDM_{3,4}}^i$ for N_C^1 and N_C^2 by **Algorithm 1**;
6: Select the minimal cover set of $\mathcal{M}_{SDM_i}^i$ for N_C^1 and N_C^2 according to Definition 3;
7: Get constraints according to $\mathcal{M}_{SDM_i}^{i*}$;
8: Transform constraints into admissible ones by **Algorithm 2**;
9: Add control places according to the above constraints.

VI. AN EXPERIMENTAL EXAMPLE

This section tests the approach proposed in the present work with an experimental example adapted from the literature [34]. As shown in Fig. 10, the RMC has two robots R_1 - R_2, four machines M_1 - M_4, two loading buffers I_1 - I_2, and two unloading buffers O_1 - O_2. There are two kinds of parts J_1 - J_2 to be processed with the following production sequences:

$$J_1 : I_1 \to R_1 \to M_1(\text{or } M_2) \to R_1 \to M_3 \to R_2 \to O_1$$

$$J_2 : I_2 \to R_2 \to M_4 \to R_1 \to M_2 \to R_1 \to O_2$$

Fig. 10. The layout of an RMC in [34].

Fig. 11 shows the Petri net which has 14 transitions and 19 places in which $P_0 = \{p_1, p_8\}$, $P_R = \{p_{14}$ - $p_{19}\}$, $P_A = \{p_2$ - p_7, p_9 - $p_{13}\}$, $T_{\widetilde{O}} = \{t_3, t_{12}\}$, and $T_{O\widetilde{C}} = \{t_3, t_7, t_8, t_{10}, t_{13}, t_{14}\}$.

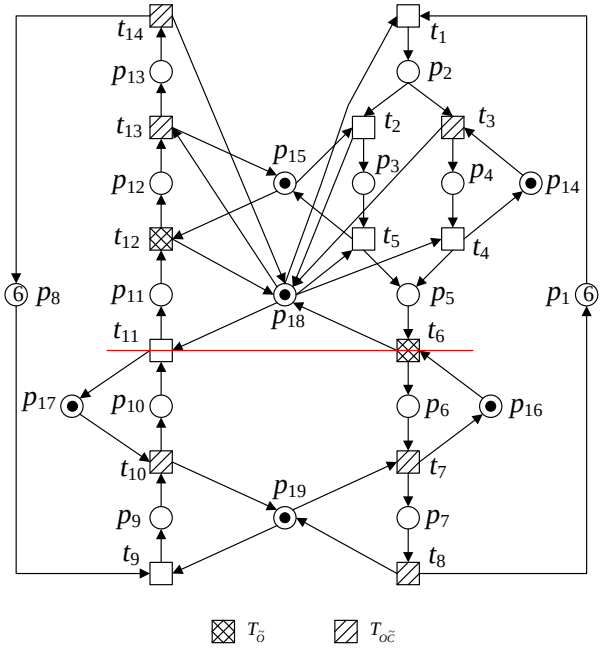
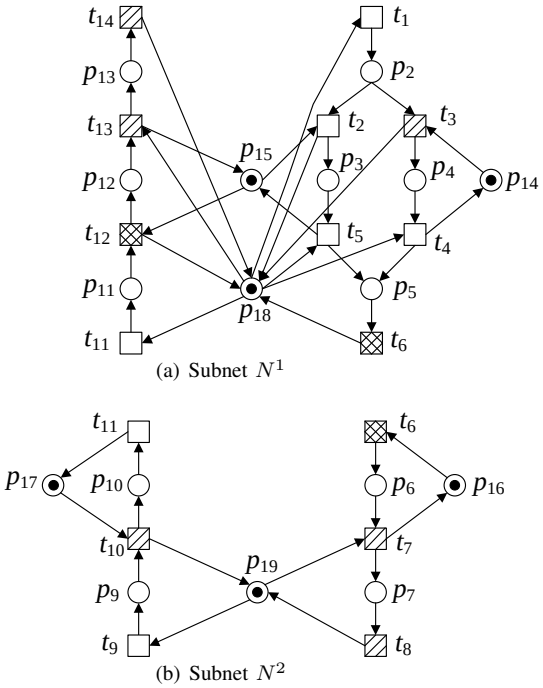


Fig. 11. The Petri net with $T_{O\widetilde{C}} = \{t_3, t_7, t_8, t_{10}, t_{13}, t_{14}\}$ and $T_{\widetilde{O}} = \{t_6, t_{12}\}$

.

First, we develop a Java program to calculate the LZ, DZ, SDM, $\mathcal{M}_A$, $\mathcal{M}_{FIM}$, $\mathcal{M}_A^*$, and $\mathcal{M}_{FIM}^*$. Next, LINGO is adopted to solve S^2U^2T and the Algorithm 2.

Then its reachability graph has 282 reachable markings in which |LZ| = 205 and $|\mathcal{M}_{FIM}| = 54$. For this net, all markings in LZ are admissible, i.e., $|\mathcal{M}_A| = |\text{LZ}|$. $\mathcal{M}_A^*$ and $\mathcal{M}_{FIM}^*$ have 26 and 8 markings, respectively. The formulated $\text{S}^2\text{U}^2\text{T}$ has $8\times(2\times8+2\times14+26+2+6-1) = 616$ constraints and $8 \times (8 + 2 \times 14 + 11) = 376$ variables. By Definition 5, the Petri net can be decomposed into two subnets via shared transitions. For example, we decompose the net into N^1 and N^2 via $T_s = \{t_6, t_{11}\}$, as shown in Fig. 12. It is easy to know that N^1 has 24 reachable markings including 18 admissible markings and 6 deadlocks which constitute the entire set of $\mathcal{M}_{FIM}$. After using vector covering, $\mathcal{M}_A^*$ and $\mathcal{M}_{FIM}^*$ have 10 and 4 markings, respectively. The subset N^2 is deadlock free with 12 reachable markings, all of which are in $\mathcal{M}_A$.



(a) Subnet N^1

(b) Subnet N^2

Fig. 12. Subnets of Fig. 11

By $\text{S}^2\text{U}^2\text{T}$, subnet N^1 becomes deadlock free after adding one control place p_{c_1} with ${}^\bullet p_{c_1} = \{t_4, 2t_5, 2t_{13}\}$, $p_{c_1}^\bullet = \{t_1, t_2, 2t_{11}\}$, and $M_0(p_{c_1}) = 3$, as shown in Fig. 13. The controlled net N_C^1 that has 18 reachable markings, and all of them are in $\mathcal{M}_A$.

Subnet N^2, also the N_C^2, since it is deadlock free, thus we do not need to add any control place for it.

It must be noted that for N_1 and N_2, we only need to solve the $\text{S}^2\text{U}^2\text{T}$ with $4 \times (2 \times 4 + 2 \times 10 + 10 + 2 + 3 - 1) = 168$ constraints and $4 \times (4 + 2 \times 10 + 7) = 124$ variables instead of 616 and 376. At present, if we compose two subnets, and the merged net has 214 markings, including 205 in $\mathcal{M}_A$ and 9 in $\mathcal{M}_{FIM}$.

During the merging process of the subnet, the output transitions are prohibited which lead to the deadlocks, so we need to identify all LSs and the markings that will definitely cause LSs. Thus, we can remove the output transitions T_{out}^i from N_C^i to find LSs and the bad states from which an LS is definitely reachable.

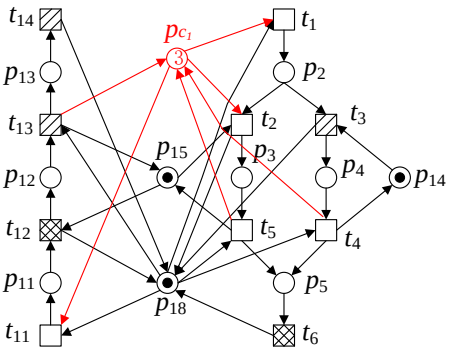


Fig. 13. The controlled net N_C^1

By Algorithm 1, all SDMs of N_C^1 and N_C^2 are computed. The selected minimal covered set of $\mathcal{M}_{SDM_{3,4}}^i$ is computed according to Definition 3. Then all the markings in $\mathcal{M}_{SDM_{3,4}}^i$ of N_C^1 and N_C^2 are prevented from meeting according to Proposition 2. Therefore, all deadlocks are prevented. For two controlled subnets N_C^1 and N_C^2, their minimal covered set of $\mathcal{M}_{SDM_{3,4}}^i$ is $\mathcal{M}_{SDM_{3,4}}^{1*} = \{p_5+3p_{c_1}, p_3+p_{c_1}, p_2+p_4+p_{c_1}\}$ and $\mathcal{M}_{SDM_{3,4}}^{2*} = \{p_9 + p_{10}\}$, respectively.

By Proposition 2 and Corollary 1, to prohibits $\mathcal{M}_{SDM_{3,4}}^i$ of N_C^1 and N_C^2, the following constraints must be satisfied:

$$\mu_5 + \mu_9 + \mu_{10} \leq 3;$$

$$\mu_3 + \mu_9 + \mu_{10} \leq 2;$$

$$\mu_2 + \mu_4 + \mu_9 + \mu_{10} \leq 3;$$

It is easy to check that the above first constraint is inadmissible, since $\lambda(t_6) = w(p_6) - w(p_5) = -1 \neq 0$. We solve the follow constraints by LINGO:

$$min \sum_{i=2}^{13} w^*(p_i)$$

$$\begin{cases} w^*(p_2) \geq 0 \\ w^*(p_3) \geq 0 \\ w^*(p_4) \geq 0 \\ w^*(p_5) \geq 1 \\ w^*(p_6) \geq 0 \\ w^*(p_7) \geq 0 \\ w^*(p_9) \geq 1 \\ w^*(p_{10}) \geq 1 \\ w^*(p_{11}) \geq 0 \\ w^*(p_{12}) \geq 0 \\ w^*(p_{13}) \geq 0 \\ \lambda(t_6) = w^*(p_6) - w^*(p_5) \leq 0 \end{cases}$$

We can get that $(w^*, k) = ((0,0,0,1,1,0,1,1,0,0,0),3)$, i.e., $\mu_5 + \mu_6 + \mu_9 + \mu_{10} \leq 3$. As a result, the following constraints must be satisfied to prohibit $\mathcal{M}_{SDM_{3,4}}^i$ of N_1^C and N_2^C:

$$\mu_5 + \mu_6 + \mu_9 + \mu_{10} \leq 3;$$

$$\mu_3 + \mu_9 + \mu_{10} \leq 2;$$

$$\mu_2 + \mu_4 + \mu_9 + \mu_{10} \leq 3;$$

The information of the control places $p_{c_2} - p_{c_4}$ for the merged net is shown in Table II. Finally, a liveness-enforcing supervisor for the whole Petri net system model in Fig. 11 is obtained. The detailed comparison results with [34] are shown in Table III. It is clear that our approach can greatly reduce the number of constraints and variables, which corresponds to the expected amount of computation of ILP. Therefore, our proposed approach is superior in the aspect of computation cost reduction. Although the finally obtained supervisor in the present work cannot guarantee to be structurally minimal due to subnets merging, its scale in terms of control places and arcs is very small since the supervisor for each subnet is structurally minimal.

TABLE II
CONTROL PLACES FOR PETRI NET IN FIG. 11

i	PI_i	${}^\bullet P_{c_i}$	$P_{c_i}^\bullet$	$M_0(P_{c_i})$
2	$\mu_5 + \mu_6 + \mu_9 + \mu_{10} \leq 3$	t_5, t_{11}	t_2, t_9	3
3	$\mu_3 + \mu_9 + \mu_{10} \leq 3$	t_3, t_6	t_2, t_5	2
4	$\mu_2 + \mu_4 + \mu_9 + \mu_{10} \leq 3$	t_2, t_4, t_{11}	t_1, t_9	3

TABLE III
COMPARISON WITH THE METHOD IN [34]

Method	Constraints	Variables	Control places	Initial markings	Permissive behaviors
[34]	616	376	2	28	205
this work	168	124	4	11	201

VII. COMPARISON

As shown in Table IV, we list some classical or latest work concerning Petri nets with uncontrollable and unobservable transitions. We conduct a comparison of these studies to highlight their different properties and characteristics, particularly the significant difference between the present work and the others.

In Table IV, the sign "$\surd$ "(resp., "×") stands for the fact that an approach does (resp., does not) have the corresponding specification. The sign "$-$" means that the approach does not consider such item or term. It should be noted that most studies handle uncontrollable and unobservable transitions according to given constraints. That is, they only need to transform the given constraints of the whole net. By contrast, we transform the subnet constraints obtained by the merging of subnets. In addition, our proposed approach is the only one that deals only with the reachability graph of subnets and is promisingly suitable for large-scale Petri net system models. Although both the work in [24] and the present work use decomposition technique, they differ from each other in the following three aspects.

1) The present work considers uncontrollable and unobservable transitions while the work in [24] does not.
2) We control the subnets by solving $\mathrm{S^2U^2T}$, which can obtain admissible and structure-minimal controllers.
3) Because of the existence of uncontrollable and unobservable transitions, constraint transformation has to be conducted in the present work.

VIII. CONCLUSION

By using Petri nets as a formalism, this paper proposes a deadlock prevention approach based on decomposition technique and ILP to design behaviorally admissible supervisors for RMCs with uncontrollable and unobservable events. We develop a new decomposition technique considering the uncontrollable and unobservable transitions with which the Petri nets are more suitable to model pratical RMCs. It can significantly reduce the number of constraints and variables in ILP, so as to alleviate the problem of state explosion. Furthermore, a method of constraint transformation for unobservable transitions based on ILP are developed so that we can add control places to prevent markings in $\mathcal{M}^i_{SDM_{3,4}}$ when merging subnets.

As for behavioral permissiveness, our proposed approach cannot obtain a maximally permissive liveness-enforcing supervisor in general. Therefore, the major directions of future work are to find the optimal supervisor or to explore the

TABLE IV
COMPARISON WITH SOME REPRESENTATIVE WORK

Work	Huang *et al.* [34]	Moody and Antsaklis [31]	Luo and Zhou [6]	You *et al.* [33]	Wang *et al.* [29]	Ran *et al.* [5]	Qin *et al.* [39]	ours
Method description	ILP	matrix multiplication	dynamic constraint	fuse B-MSC	transform weight bottom-up	ILP / structural analysis	iterativeILP	decompose + ILP
Permissive behavior	optimal	high	optimal	optimal	optimal	optimal	high	high
Whole reachability	$\surd$	$-$	$-$	$-$	$-$	$-$	minimal siphon	subnets reachability
Unobservable?	$\surd$	$\surd$	$\surd$	×	×	$\surd$	$\surd$	$\surd$
Uncontrollable?	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
Given constraints?	×	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	×	×
Whole constraints?	$-$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	subnets constraints
Low computational burden?	×	×	×	×	×	×	×	$\surd$

situation in which the obtained supervisors can be maximally permissive.

REFERENCES

[1] C. G. Cassandras and S. Lafortune, Eds., *"Petri net" in Introduction to Discrete Event Systems*, 4th ed. Germany, Berlin: Springer, 2021, pp. 259–302.

[2] F. Basile and L. Ferrara, "Validation of industrial automation systems using a timed model of system requirements," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 1, pp. 130–143, 2023.

[3] B. Lennartson, K. Bengtsson, O. Wigström, and S. Riazi, "Modeling and optimization of hybrid systems for the tweeting factory," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 191–205, 2016.

[4] A. Giua and M. Silva, "Petri nets and automatic control: A historical perspective," *Annual Reviews in Control*, vol. 45, pp. 223–239, 2018.

[5] N. Ran, T. Li, S. Wang, and Z. He, "Supervisor synthesis for Petri nets with uncontrollable and unobservable transitions," *IEEE Transactions on Automation Science and Engineering*, 2023.

[6] J. Luo and M. Zhou, "Petri-net controller synthesis for partially controllable and observable discrete event systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1301–1313, 2017.

[7] A. C. Bonafin, F. G. Cabral, and M. V. Moreira, "An effective approach for fault diagnosis of discrete-event systems modeled as safe labeled petri nets," *Control Engineering Practice*, vol. 123, p. 105168, 2022.

[8] H. Liu, Y. Feng, J. Li, and J. Luo, "Robust Petri net controllers for flexible manufacturing systems with multitype and multiunit unreliable resources," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 3, pp. 1431–1444, 2023.

[9] G. Liu, Y. Liu, and Z. Li, "Robust liveness-enforcing supervisor for Petri nets with unreliable resources based on mixed integer programming," *Soft Computing*, vol. 26, no. 8, pp. 4019–4032, 2022.

[10] J. Luo, Z. Liu, S. Wang, and K. Xing, "Robust deadlock avoidance policy for automated manufacturing system with multiple unreliable resources," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, pp. 812–821, 2020.

[11] Y. Feng, K. Xing, M. Zhou, X. Wang, and H. Liu, "Robust deadlock prevention for automated manufacturing systems with unreliable resources by using general Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 10, pp. 3515–3527, 2020.

[12] D. Lefebvre, C. Seatzu, C. N. Hadjicostis, and A. Giua, "Probabilistic state estimation for labeled continuous time markov models with applications to attack detection," *Discrete Event Dynamic Systems*, vol. 32, no. 1, pp. 65–88, 2022.

[13] Y. Qiao, M. Zhou, N. Wu, and Q. Zhu, "Scheduling and control of startup process for single-arm cluster tools with residency time constraints," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1243–1256, 2017.

[14] B. Huang and M. Zhou, "Symbolic scheduling of robotic cellular manufacturing systems with timed Petri nets," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 5, pp. 1876–1887, 2022.

[15] T. Miyano, J. Romberg, and M. Egerstedt, "Globally optimal assignment algorithm for collective object transport using air–ground multirobot teams," *IEEE Transactions on Control Systems Technology*, pp. 1–8, 2023.

[16] J. Ezpeleta, J. Colom, and J. Martinez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 173–184, 1995.

[17] F. Chu and X. Xie, "Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 6, pp. 793–804, 1997.

[18] Y. Huang, M. Jeng, X. Xie, and S. Chung, "Deadlock prevention policy based on Petri nets and siphons," *International Journal of Production Research*, vol. 39, no. 2, pp. 283–305, 2001.

[19] S. Wang, X. Guo, O. Karoui, M. Zhou, D. You, and A. Abusorrah, "A refined siphon-based deadlock prevention policy for a class of Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 1, pp. 191–203, 2023.

[20] M. Uzam, "An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions," *International Journal of Advanced Manufacturing Technology*, vol. 19, pp. 192–208, 2002.

[21] Y. Chen, Z. Li, M. Khalgui, and O. Mosbahi, "Design of a maximally permissive liveness- enforcing Petri net supervisor for flexible manufacturing systems," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 2, pp. 374–393, 2011.

[22] Y. Huang, Y. Pan, and M. Zhou, "Computationally improved optimal deadlock control policy for flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 2, pp. 404–415, 2012.

[23] B. Huang, M. Zhou, Y. Huang, and Y. Yang, "Supervisor synthesis for fms based on critical activity places," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 5, pp. 881–890, 2019.

[24] C. Zhong, W. He, Z. Li, N. Wu, and T. Qu, "Deadlock analysis and control using Petri net decomposition techniques," *Information Sciences*, vol. 482, pp. 440–456, 2019.

[25] H. Hu, Y. Liu, and L. Yuan, "Supervisor simplification in FMSs: Comparative studies and new results using Petri nets," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 1, pp. 81–95, 2016.

[26] M. Uzam, Z. Li, G. Gelen, and R. S. Zakariyya, "A divide-and-conquer-method for the synthesis of liveness enforcing supervisors for flexible manufacturing systems," *Journal of Intelligent Manufacturing*, vol. 27, no. 5, pp. 1111–1129, 2016.

[27] X. Jin, S.-L. Dai, J. Liang, and D. Guo, "Multirobot system formation control with multiple performance and feasibility constraints," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 4, pp. 1766–1773, 2022.

[28] A. Ghaffari, N. Rezg, and X. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 137–141, 2003.

[29] S. Wang, D. You, and C. Wang, "Optimal supervisor synthesis for Petri nets with uncontrollable transitions: A bottom-up algorithm," *Information Sciences*, vol. 363, pp. 261–273, 2016.

[30] D. You, S. Wang, Z. Li, and C. Wang, "Computation of an optimal transformed linear constraint in a class of Petri nets with uncontrollable transitions," *IEEE Access*, vol. 5, pp. 6780–6790, 2017.

[31] J. Moody and P. Antsaklis, "Petri net supervisors for des with uncontrollable and unobservable transitions," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 462–476, 2000.

[32] Z. Hao, S. Lei, J. Liu, and J. Hao, "Supervisor synthesis for a class of Petri nets," *Journal of Electronic Measurement and Instrument*, vol. 36, no. 1, pp. 180–187, 2022.

[33] D. You, S. Wang, and C. Seatzu, "Supervisory control of a class of Petri nets with unobservable and uncontrollable transitions," *Information Sciences*, vol. 501, pp. 635–654, 2019.

[34] B. Huang, M. Zhou, C. Wang, A. Abusorrah, and Y. Al-Turki, "Deadlock-free supervisor design for robotic manufacturing cells with uncontrollable and unobservable events," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 3, pp. 597–605, 2021.

[35] J. O. Moody and P. J. Antsaklis, "Supervisory control of discrete event systems using Petri nets," 1998.

[36] K. Yamalidou, J. Moody, M. Lemmon, and P. Antsaklis, "Feedback control of Petri nets based on place invariants," *Automatica*, vol. 32, no. 1, pp. 15–28, 1996.

[37] M. Uzam and M. Zhou, "An iterative synthesis approach to Petri net-based deadlock prevention policy for flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 3, pp. 362–371, 2007.

[38] Y. Chen and Z. Li, "Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems," *Automatica*, vol. 47, no. 5, pp. 1028–1034, 2011.

[39] M. Qin, Z. Li, M. Zhou, M. Khalgui, and O. Mosbahi, "Deadlock prevention for a class of Petri nets with uncontrollable and unobservable transitions," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 3, pp. 727–738, 2012.

IX. SIMPLE REFERENCES

You can manually copy in the resultant. bbl file and set second argument of `\begin` to the number of references (used to reserve space for the reference number labels box).

X. BIOGRAPHY SECTION

If you have an EPS/PDF photo (graphicx package needed), extra braces are needed around the contents of the optional argument to biography to prevent the LaTeX parser from getting confused when it sees the complicated `\includegraphics`

command within an optional argument. (You can create your own custom macro containing the `\includegraphics` command to make things simpler here.)

If you include a photo:

Michael Shell Use `\begin{IEEEbiography}` and then for the 1st argument use `\includegraphics` to declare and link the author photo. Use the author name as the 3rd argument followed by the biography text.

If you will not include a photo:

John Doe Use `\begin{IEEEbiographynophoto}` and the author name as the argument followed by the biography text.