Monitor design with multiple self-loops for maximally permissive supervisors

YuFeng Chen a,c, ZhiWu Li b,a,* Kamel Barkaoui c, Murat Uzam d

a School of Electro-Mechanical Engineering Xidian University No. 2 South Taibai Road Xi’an 710071, China
b Institute of Systems Engineering Macau University of Science and Technology Taipa, Macau 999078, China
c Cedric Lab and Computer Science Department Conservatoire National des Arts et Métiers, Paris 75141, France
d Elektrik-Elektronik Mahendisliği Bölümü, Muhendislik-Mimarlık Fakültesi Mellaş Üniversitesi, 38280 Talas/Kayseri, Turkey

Abstract

In this paper, we improve the previous work by considering that a control place can have multiple self-loops. Then, two integer linear programming problems (ILPPs) are formulated. Based on the first ILPP, an iterative deadlock control policy is developed, where a control place is computed at each iteration to implement as many marking/transition separation instances (MTSIs) as possible. The second ILPP can find a set of control places to implement all MTSIs and the objective function is used to minimize the number of control places. It is a non-iterative deadlock control strategy since we need to solve the ILPP only once. Both ILPPs can make all legal markings reachable in the controlled system, i.e., the obtained supervisor is behaviorally optimal. Finally, we provide examples to illustrate the proposed approaches.

Keywords:
Petri net
Flexible manufacturing system (FMS)
Deadlock prevention
Supervisory control
Self-loop

1. Introduction

Petri nets [42] are an effective tool to model and control flexible manufacturing systems (FMSs) [56]. Petri nets have compact structures and can be represented in the form of matrices. Thus, they can be simply analyzed by linear algebras. Deadlocks [16] are a constant issue in FMSs [9,14,17,31,33,34,36–38,40] due to the theory of regions proposed by Ghaffari et al. [25] and Uzam [52], which can define a maximally permissive Petri net supervisor if it exists. The approach first generates the reachability graph of a net model. Then, the set of marking/transition separation instances (MTSIs) is derived. An MTSI is a pair of a marking M and a transition t, denoted as (M, t), where M is a legal marking and once t fires at M, it yields an illegal marking. For deadlock control purposes, Ghaffari et al. [25] develop an iterative approach where at each iteration, an MTSI (M, t) is singled out and a linear programming model is designed to find a control place to implement (M, t) by preventing t from firing at M. Meanwhile, all legal
markings are ensured to be reachable. The process carries out until all MTSIs are implemented. Then, a set of control places is obtained, which can make the controlled system live with all legal markings. That is to say, the obtained supervisor is behaviorally optimal. However, the approach does not consider the structural complexity since it always leads to a supervisor with too many control places.

Another representative study based on the reachability graph analysis is presented by Uzam and Zhou [53,54]. They classify a reachability graph into two parts: a live zone (LZ) and a deadlock zone (DZ), where the LZ contains all legal markings and the DZ contains all illegal markings. Then, the set of first-met bad markings (FBMs) is computed, where an FBM is an illegal marking that can be directly reached by firing a transition at a legal marking. That is to say, an FBM is within the LZ and represents the very first entry from the LZ to the DZ. Uzam and Zhou also develop an iterative approach to prevent deadlocks in a net model. An FBM is forbidden by using a place invariant (PI) based control place synthesis method [62]. Once all FBMs are forbidden, the controlled system is live. This approach is easy to use since it does not require to compute control places by solving linear programming problems. However, it cannot guarantee the behavioral optimality of the obtained supervisor. Motivated by Uzam and Zhou’s work [54], the work in [6] develops a maximally permissive deadlock control method such that a control place is computed by solving an integer linear programming problem (ILPP) while an FBM is prohibited but no legal markings are forbidden. The process cannot terminate until all FBMs are forbidden. Then, we can obtain a supervisor with all legal markings reachable in the resulting controlled net model. Meanwhile, a marking reduction approach is proposed to reduce the number of markings that need to be considered. Thus, the number of constraints in the ILPP can be reduced. In [7,8], ILPPs are proposed to design optimal supervisors with the minimal number of control places. The work [7,8] can successfully handle the behavioral permissiveness and structural complexity problems but suffers from the computational complexity due to the existence of too many constraints in the ILPPs.

All the above studies based on Petri nets are in the framework of pure net models. However, there exist some Petri net models that cannot be optimally controlled by pure net supervisors [68]. Self-loops are a classical non-pure Petri net structure. A self-loop contains two arcs (p,t) and (t,p) connecting a place p and a transition t. Self-loops are used in several papers [18,42,48,54,57] for structural reductions, systems synthesis, system modeling, etc. In the previous work [11], self-loops are used to design maximally permissive supervisors to handle deadlock problems in FMSs. Similar to the theory of regions [25,52], we prevent deadlocks by implementing all MTSIs of a net to be controlled. For an MTSI (M,t), we assume that there is a self-loop associated with t in a control place. Then, an ILPP is formulated to design the control place. The constraints in the ILPP can ensure that each control place can be designed to ensure that each MTSI is implemented by at least one control place and the objective function can minimize the number of control places. As a result, all control places can be obtained by solving only one ILPP and the obtained supervisor is structurally minimal in the sense of the number of control places. A drawback of the proposed methods is that they may fail to find a solution if the proposed ILPP for a given net model has no solution.

In summary, we reach the following contributions in this work:

1. In the case that a control place can have multiple self-loops, an ILPP is developed to design an optimal control place with a self-loop for each critical transition. The constraints in the ILPP are used to make all legal markings reachable in the controlled system and the objective function can ensure that the computed control place implements as many MTSIs as possible. Based on the proposed ILPP, an iterative deadlock prevention policy is developed, where a control place is designed at each iteration and the process carries out until all MTSIs are implemented. Hence, we can obtain an optimal supervisor with a small number of control places.

2. In order to minimize the number of control places in the obtained supervisor, we formulate an ILPP to design a set of control places with self-loops and its objective function is used to select the minimal number of control places. As a result, we can obtain an optimal supervisor with the minimal number of control places. The proposed ILPP can lead to a non-iterative deadlock control policy since it can find all control places to implement all MTSIs by solving only one ILPP.

3. The proposed approaches are applicable to all FMS-oriented classes of Petri net models in the literature, including PPN [12,60], S*PR [2], ES*PR [49], S*T*PR [50], S*T*PR [22], S*LSPR [44], S*PGR [45], and S*PMR [28].

The rest of this paper is organized as follows. Section 2 briefly recalls some basics of Petri nets. In Section 3, we propose two ILPPs to design optimal supervisors with compact structures and an illustrative example is presented to show the applications of the proposed methods in detail. Some examples from the literature are provided in Section 4 to show the control performance of the proposed methods. Finally, we conclude the paper in Section 5.

2. Preliminaries

This section only recalls the basics of Petri nets. More details can be found in [6,7,11,42].

A Petri net [42] is a four-tuple N = (P, T, F, W) where P is a set of places and T is a set of transitions with P \ T = \emptyset, and F \subseteq (P \times T) \cup (T \times P) is a flow relation of the net. W : (P \times T) \cup (T \times P) \rightarrow N assigns a weight to an arc: W(x,y) > 0 if (x,y) \in F, and W(x,y) = 0, otherwise, where x,y \in P \cup T and N is the set of non-negative integers. *x = \{y \in P \cup T | (x,y) \in F\} is the preset of x and x* = \{y \in P \cup T | (x,y) \in F\} is the postset of x. A marking is a mapping M : P \rightarrow N where M(p) denotes the number of tokens in place p. A net is pure (self-loop free) if \forall(x,y) \in (P \times T) \cup (T \times P), W(x,y) > 0 implies W(y,x) = 0. The incidence matrix [N] of a net N is a |P| \times |T| integer matrix with \{N(p,t) = W(p,t) - W(t,p)\}. A transition t \in T is enabled at marking M, denoted as M(t), if \forall p \in t, M(p) \geq W(p,t). Once a transition t is enabled at M and fires, it yields a new marking M', denoted as M(t)M, where M'(p) = M(p) - W(p,t) + W(t,p). The set of reachable markings of net \{N, M_0\} is denoted by R(N, M_0) and its reachability graph is denoted as G(N, M_0). A transition t \in T is live at M_0 if
∀ M ∈ R(N, M₀), ∃ M’ ∈ R(N, M), M’(t), (N, M₀) is live if ∀ t ∈ T, t is live at M₀. It is dead at M₀ if 3t ∈ T, M₀(0).

In this paper, some notations and definitions on the reachability analysis of Petri nets and the marking reduction technique are originally from [6,7,11,53]. Specifically, they can be found in [11]. In order to focus on the main contributions of this work and also thanks to the limited space, we do not provide them repeatedly and encourage the readers to find more details in [11].

3. Simplification of supervisory structures

The work in [11] presents an approach to design an optimal supervisor where a control place has a self-loop. In this case, we develop two ILPPs to improve the previous work by simplifying the supervisory structure where a control place can have multiple self-loops.

3.1. Design of a supervisor with a small number of control places

The constraint that includes enabled conditions can be found in [41,62]. In this paper, we consider the following constraint that includes enabled conditions for multiple transitions:

\[ \sum_{i=1}^{n} l_i · μ_i + \sum_{j=1}^{m} w_j · q_j ≤ β \]  

(1)

where \( w_j \) is a nonnegative integer and \( q_j \in \{0, 1\} \) represents whether transition \( t_j \in T \) is enabled. Specially, \( q_j = 1 \) indicates that \( t_j \) is enabled and \( q_j = 0 \) means that \( t_j \) is disabled.

Eq. (1) can be considered as two parts: (1) \( \sum_{i=1}^{n} l_i · μ_i \) cannot be greater than \( β \) and (2) once \( \sum_{i=1}^{n} l_i · μ_i \) is greater than \( β - w_j \), \( t_j \) is disabled. For (1), it can be implemented by introducing a slack variable \( μ_i \) to form a PL, as shown in Section 3 of [11]. For (2), \( \sum_{i=1}^{n} l_i · μ_i > β - w_j \) implies that \( μ_i < w_j \), i.e., \( t_j \) is disabled if \( μ_i < w_j \). In this case, for each transition \( t_q \), we self-loop with a weight \( w_j \) between \( p_i \) and \( t_q \), i.e., \( W(p_i, t_q) = W(t_q, p_i) = w_j \). Then, \( t_j \) is disabled at a marking if \( M(p_i) < w \) and enabled if \( M(p_i) ≥ w \). Once \( t_j \) is enabled and fires, it does not change the tokens in \( p_i \).

By Corollary 1 in [11], we only consider markings in the minimal covering set \( M^*_t \) of legal markings (Definition 5 in [11]) to guarantee the reachability of all legal markings, i.e.,

\[ \sum_{i \in A^*_t} l_i · M_{t}(p_i) ≤ β, \quad \forall M_t \in M^*_t \]  

(2)

Now we recall some necessary results from [11] as follows.

**Definition 1** (Chen et al. [11]). Let \( t \) be a critical transition (the set of critical transitions, see Eq. (16) in [11]) and \( (M, t) \) an MTSI. \( (M, t) \) is called a \( t \)-critical MTSI. The set of \( t \)-critical MTSIs is denoted as \( Ω_t^* \).

**Theorem 1** (Chen et al. [11]). Let \( (M_1, t) \) and \( (M_2, t) \) be two MTSIs in \( Ω_t^* \) with \( M_1 ≤ A M_2 \). If \( (M_1, t) \) is implemented by a control place with a self-loop associated with \( t \), then \( (M_2, t) \) is implemented by the control place.

**Definition 2** (Chen et al. [11]). Let \( Ω^*_t \) be a subset of \( t \)-critical MTSIs. \( Ω^*_t \) is called a minimal covered set of \( t \)-critical MTSIs if the following two conditions are satisfied:

1. \( ∀ (M, t) ∈ Ω^*_t, ∃ (M’, t) ∈ Ω^*_t \) s.t. \( M’ ≤ A M \);
2. \( ∀ (M, t) ∈ Ω^*_t, ∃ (M’, t) ∈ Ω^*_t \) s.t. \( M’ ≤ A M \) and \( M ≠ M’ \).

**Corollary 1** (Chen et al. [11]). If all MTSIs in \( Ω^*_t \) are implemented by control places with self-loops associated with \( t \), then all MTSIs in \( Ω^*_t \) are implemented by the control places.

For optimal deadlock control purposes, we only consider to disable the critical transitions. Thus, we suppose that a control place \( p_i \) has a self-loop associated with each critical transition, i.e., \( W(p_i, t_q) = w_q, \forall t_q ∈ T_c \) (the set of critical transitions, see Eq. (16) in [11]). We define the minimal covered set of MTSIs as follows:

**Definition 3.** Let \( Ω^*_t \) be a subset of MTSIs. \( Ω^*_t \) is called a minimal covered set of MTSIs if the following two conditions are satisfied:

1. \( ∀ (M, t) ∈ Ω^*_t, ∃ (M’, t) ∈ Ω^*_t \) s.t. \( M’ ≤ A M \);
2. \( ∀ (M, t) ∈ Ω^*_t, ∃ (M’, t) ∈ Ω^*_t \) s.t. \( M’ ≤ A M \) and \( M ≠ M’ \).

It is obvious that \( Ω^*_t \) is the union of \( Ω^*_t \) (the minimal covered set of \( t \)-critical MTSIs, Definition 8 in [11]) with \( t \in T_c \), i.e.,

\[ Ω^*_t = \bigcup_{t ∈ T_c} Ω^*_t \]  

(3)

**Corollary 2.** If all MTSIs in \( Ω^*_t \) are implemented by a set of control places where each control place has a self-loop associated with each critical transition, then all MTSIs are implemented by the control places.

**Proof.** It follows immediately from Corollary 1. □

For each critical transition \( t_q ∈ T_c \), it should be enabled at each of its enabled good markings. Then, the enabled condition is

\[ \sum_{i \in N_A} l_i · (M_k(p_i) + N(p_i, t_q)) ≤ β - w_q, \quad ∀ M_k ∈ Ω^*_t, \quad ∀ t_q ∈ T_c \]  

(4)

where \( N \) is the incidence matrix of the net to be controlled and \( Ω^*_t \) is the minimal covering set of \( t_q \)-enabled good markings (see Definition 6 in [11]).

We introduce a set of variables \( f_{jq} \)’s (\( ∀ (M_j, t_q) ∈ Ω^*_t \)) to represent whether an MTSI \( (M_j, t_q) ∈ Ω^*_t \) is implemented by \( p_j \). The disabled condition for \( (M_j, t_q) \) is

\[ \sum_{i \in N_A} l_i · (M_k(p_i) + N(p_i, t_q)) ≥ -Q \]  

\[ · (1 - f_{jq}) + β - w_q + 1, \quad ∀ (M_j, t_q) ∈ Ω^*_t \]  

(5)

where \( f_{jq} \in \{0, 1\} \) (\( ∀ (M_j, t_q) ∈ Ω^*_t \)). In Eq. (5), \( f_{jq} = 1 \) indicates that \( (M_j, t_q) \) is implemented by \( p_j \) and \( f_{jq} = 0 \) indicates that \( (M_j, t_q) \) is not implemented.

By combining Eqs. (2), (4) and (5), the following ILPP is developed to design an optimal control place \( p_j \) with a self-loop associated with each critical transition \( t_q ∈ T_c \), which is denoted as the Implementation of Maximal Number of MTSIs (IMNM).

**IMNM:**

\[ \max f = \sum_{(M_j, t_q) ∈ Ω^*_t} f_{jq} \]

subject to

\[ \sum_{i \in N_A} l_i · M_k(p_i) ≤ β, \forall M_k ∈ Ω^*_t \]  

(6)

\[ \sum_{i \in N_A} l_i · (M_k(p_i) + N(p_i, t_q)) ≤ β - w_q, \quad ∀ M_k ∈ Ω^*_t, \quad ∀ t_q ∈ T_c \]  

(7)

\[ \sum_{i \in N_A} l_i · (M_k(p_i) + N(p_i, t_q)) ≥ -Q \]  

\[ · (1 - f_{jq}) + β - w_q + 1, \quad ∀ (M_j, t_q) ∈ Ω^*_t \]  

(8)

\[ l_i ∈ \{0, 1, 2, …\}, \quad i ∈ N_A \]

\[ β ∈ \{0, 1, 2, 3, …\} \]
The objective function \( f \) is used to maximize the number of MTSIs that are implemented by control place \( p_s \). Denote its optimal value by \( f^\ast \). If \( f^\ast = 0 \), we have \( \forall (M_j, t_q) \in \Omega^\ast, f_{j,q} = 0 \), implying that no MTSI can be implemented by \( p_s \). In this case, the proposed method cannot obtain an optimal control place with a self-loop associated with each critical transition. If \( f^\ast = |\Omega^\ast| \), we have \( \forall (M_j, t_q) \in \Omega^\ast, f_{j,q} = 1 \), implying that only one control place with a self-loop associated with \( t \) is required to implement all MTSIs. The number of constraints and variables in IMNM is summarized in Table 1.

### Algorithm 1

1. Compute the set \( M_L \) of reachable markings.
2. Add all control places \( p_s \) of MTSIs in \( M_L \), to satisfy the enabled conditions. Furthermore, we only consider the number of constraints in IMNM to compute control places.
3. \( \text{Input: Petri net model } (N, M_0) \text{ of an FMS with } N = \{p_0, p_1, ..., p_k, t_1, ..., t_l, F, W, 1\}. \)
4. \( \text{Output: an optimally controlled Petri net system } V_M(\ast) \).
5. \( \text{Compute the set } M_L \text{ of legal markings and the set } \Omega \text{ of MTSIs for } (N, M_0). \)
6. \( \text{Compute the set } T_C \text{ of critical transitions and the minimal covering set } M_L^\ast \text{ of legal markings for } (N, M_0). \)
7. \( V_M = \Omega^\ast \& V_M^\ast \text{ is used to denote the set of control places to be computed}. \)
8. \( \textbf{foreach } t \in T_C \text{ do} \)
9. \( \text{Compute the minimal covering set } \epsilon^\ast_t \text{ of } t \text{-enabled good markings and the minimal covered set } \Omega^\ast \text{ of MTSIs.} \)
10. \( \textbf{while } |\Omega^\ast| \neq \emptyset \text{ do} \)
11. \( \text{Solve IMNM. If } f^\ast = 0 \text{, exit, as no control place can be obtained.} \)
12. \( \text{Let } U/s (i \in N_k), \beta, \text{ and } w/s (t_q \in T_C) \text{ be the solution.} \)
13. \( \text{Compute control place } p_i \text{ with a self-loop associated with each critical transition.} \)
14. \( V_M = V_M \cup \{p_i\} \text{ and } \Omega^\ast = \Omega^\ast \ominus \Omega_{p_i} \text{ (the set } \Omega_{p_i} \text{ of MTSIs implemented by } p_i \text{, Definition 9 in } [11]). \)
15. \( \textbf{endwhile} \)
16. \( \text{Add all control places in } V_M \text{ to } (N, M_0) \text{ and denote the resulting net system as } (N^\ast, M_0^\ast). \)
17. \( \text{Output } (N^\ast, M_0^\ast). \)
18. \( \text{End.} \)

Now we discuss the computational complexity of Algorithm 1.

First, it is based on reachability graph analysis of a Petri net model, which suffers from the state explosion problem since the number of reachable markings increases exponentially with respect to the size of a Petri net model. Second, we need to solve ILPPs, which are NP-hard in theory. However, the computational time to solve an ILPP greatly depends on the number of constraints and variables. Thus, we propose a marking reduction technique to reduce the number of constraints in the ILPPs, aiming to reduce the computational burden of the proposed method. In summary, the computational complexity of the proposed method is still NP-hard.

#### Theorem 2

Algorithm 1 can obtain a maximally permissive net supervisor for a Petri net model of an FMS if there exists a solution that satisfies Eqs. (2), (4) and (5) for each MTSI in \( \Omega^\ast \).

#### Proof.

According to Corollary 1 in [11], Eq. (2) can ensure that each computed control place does not forbid any legal marking. By Corollary 2 in [11], if a transition \( t_k \in T_c \) is enabled at a good marking, Eq. (4) is used to ensure that the additional self-loop does not disable its firing. By Corollary 2, if all MTSIs in \( \Omega^\ast \) are implemented, all MTSIs are implemented. That is to say, the final controlled system is live and all legal markings are kept, i.e., the conclusion holds.

### 3.2. Design of a supervisor with the minimal number of control places

In this section, we aim to compute an optimal supervisor with the minimal number of control places. We suppose that there are \( n_c (n_c \geq 2) \) control places to be computed, where each control place has a self-loop associated with each critical transition. Then, we propose an ILPP to select the minimal number of control places from the \( n_c \) control places to implement all MTSIs in \( \Omega^\ast \). Let \( p_{i_1}, p_{i_2}, ..., p_{i_{n_c}} \) be the \( n_c \) control places where \( \forall p_{i_j} (j \in \{1, 2, ..., n_c\}) \) and \( t_k \in T_c \), the weight on the self-loop between \( p_{i_j} \) and \( t_k \) is \( w_{\ast,j} \), i.e., \( W(p_{i_j}, t_k) = W(t_k, p_{i_j}) = w_{\ast,j} \). Meanwhile, each control place \( p_{i_j} \) is used to implement the condition:

\[
\sum_{i_\in N_4} X_{i_j} \cdot p_{i,j} \leq \beta \quad (9)
\]

In this case, the reachability condition, Eq. (2), is modified as:

\[
\sum_{i_\in N_4} X_{i_j} \cdot p_{i,j} \leq \beta, \forall (M_j, t_q) \in \Omega^\ast, \forall z \in \{1, 2, ..., n_c\} \quad (10)
\]

The enabled conditions for the markings in \( \epsilon^\ast_t (\forall t_q \in T_C) \) are presented as:

\[
\sum_{i_\in N_4} X_{i_j} \cdot (M_j(p_{i,j}) + [N(p_{i,j}, t_q)]) \leq \beta_z - w_{\ast,q}, \quad \forall M_k \in \epsilon^\ast_t, \forall t_q \in T_C, \forall z \in \{1, 2, ..., n_c\} \quad (11)
\]

We introduce a set of variables \( f_{j,z,q} \)’s (\( \forall (M_j, t_q) \in \Omega^\ast, \forall z \in \{1, 2, ..., z\} \)) to represent whether an MTSI \( (M_j, t_q) \in \Omega^\ast \) is implemented by \( p_{i_j} \). The disabled condition for \( (M_j, t_q) \) is modified as:

\[
\sum_{i_\in N_4} X_{i_j} \cdot (M_j(p_{i,j}) + [N(p_{i,j}, t_q)]) \geq -Q \cdot (1 - f_{j,z,q}) + \beta_z - w_{\ast,q} + 1, \quad \forall (M_j, t_q) \in \Omega^\ast, \forall z \in \{1, 2, ..., n_c\} \quad (12)
\]

where \( f_{j,z,q} \in \{0, 1\} (\forall (M_j, t_q) \in \Omega^\ast, \forall z \in \{1, 2, ..., z\}) \). In Eq. (12), \( f_{j,1} = 1 \) indicates that \( (M_j, t_q) \) is implemented by \( p_{i_1} \) and \( f_{j,z,q} = 0 \) indicates that \( (M_j, t_q) \) is not implemented by \( p_{i_j} \). For each MTSI \( (M_j, t_q) \in \Omega^\ast \), it should be implemented by at least one control place. Hence, we have

\[
\sum_{z=1}^{n_c} f_{j,z,q} \geq 1, \quad \forall (M_j, t_q) \in \Omega^\ast \quad (13)
\]

We introduce a binary variable \( h_{z} \in \{0, 1\} \) for each control place \( p_{i_j} \) to represent whether \( p_{i_j} \) is selected or not. The fact \( h_{z} = 1 \) indicates that \( p_{i_j} \) is selected and \( h_{z} = 0 \) implies that \( p_{i_j} \) is not selected. If a control place \( p_{i_j} \) is not selected, it cannot implement
any MTSI. Therefore, we have
\[ f_{z,j} \leq h_z, \quad \forall (M_z, t_q) \in \Omega^*, \quad \forall z \in \{1, 2, \ldots, n_z\} \quad (14) \]

Finally, an objective function is used to select the minimal number of control places to implement all MTSIs, as follows:
\[
\min h = \sum_{z=1}^{n_z} h_z
\]
(15)

Combining Eqs. (10)–(15), we can obtain an ILPP, namely the Minimization of the Number of Control Places (MNCP), as presented below:

**MNCP:**
\[
\min h = \sum_{z=1}^{n_z} h_z
\]
subject to
\[
\sum_{i \in \mathbb{N}_a} l_{z,i} \cdot M_i(p_i) \leq \beta_z, \quad \forall M_i \in \mathbb{M}_i^*, \quad \forall z \in \{1, 2, \ldots, n_z\} \quad (16)
\]
\[
\sum_{i \in \mathbb{N}_a} l_{z,j} \cdot (M_i(p_i) + [N](p_i, t_q)) \leq \beta_j - w_{z,j}^r, \quad \forall M_k \in \mathbb{C}_k^*, \forall t_q \in T_c, \quad \forall z \in \{1, 2, \ldots, n_z\} 
\]
\[
\sum_{i \in \mathbb{N}_a} l_{z,j} \cdot (M_i(p_i) + [N](p_i, t_q)) \geq -Q \cdot (1 - f_{z,j}) + \beta_j 
- w_{z,j}^r + 1, \quad \forall (M_j, t_q) \in \Omega^*, \quad \forall z \in \{1, 2, \ldots, n_z\} 
\quad \forall z \in \{1, 2, \ldots, n_z\} \quad (17)
\]
\[
\sum_{z=1}^{n_z} f_{z,j} \geq 1, \quad \forall (M_j, t_q) \in \Omega^* 
\quad \forall z \in \{1, 2, \ldots, n_z\} \quad (18)
\]
\[
f_{z,j} \leq h_z, \quad \forall (M_j, t_q) \in \Omega^*, \quad \forall z \in \{1, 2, \ldots, n_z\} 
\quad \forall z \in \{1, 2, \ldots, n_z\} \quad (19)
\]
\[
l_{z,i} \in \{0, 1, 2, \ldots\}, \quad i \in \mathbb{N}_a, \quad \forall z \in \{1, 2, \ldots, n_z\} 
\]
\[
\beta_z \in \{0, 1, 2, 3, \ldots\}, \quad \forall z \in \{1, 2, \ldots, n_z\} 
\]
\[
h_z \in \{0, 1\}, \quad \forall z \in \{1, 2, \ldots, n_z\} 
\]
\[
w_{z,q} \in \{0, 1, 2, 3, \ldots\}, \quad \forall t_q \in T_c, \quad \forall z \in \{1, 2, \ldots, n_z\} 
\]
\[
f_{z,j} \in \{0, 1\}, \quad \forall (M_j, t_q) \in \Omega^*, \quad \forall z \in \{1, 2, \ldots, n_z\} 
\]

The objective function \( h \) is used to minimize the number of control places to be computed. Denote its optimal solution as \( h^* \). If MNCP has an optimal solution, then we can claim that the given Petri net model can be optimally controlled by the minimal number \( h^* \) of control places with self-loops, as stated below.

**Theorem 3.** Given a Petri net model, if MNCP has an optimal solution \( h^* \), then we can obtain an optimal supervisor that has the minimal number \( h^* \) of control places with self-loops.

**Proof.** According to Eq. (16), every computed control place does not forbid any legal marking. For each critical transition, Eq. (17) ensures that it is enabled at all its enabled good markings. By Eqs. (18) and (19), every MTSI in \( \Omega^* \) is implemented by at least one selected control place. Thus, all MTSIs are accordingly implemented. Eq. (20) ensures that only the selected control places are used to implement MTSIs. The objective function ensures that the minimal number of control places is selected, where the number of control places in the obtained supervisor is equal to the optimal solution \( h^* \). In summary, the conclusion holds.

If the given \( n_z \) is too small, i.e., it is smaller than the minimal number of control places for the given net model, MNCP has no solution. In this case, we can rewrite MNCP by increasing \( n_z \). In fact, the greatest number of control places is such that we design a control place for each MTSI in \( \Omega^* \). Thus, we have \( n_z \leq |\Omega^*| \). If MNCP has no solution with \( n_z = |\Omega^*| \), it fails to design an optimal supervisor with self-loops for the given Petri net model.

Next, we discuss MNCP from the viewpoint of the numbers of constraints and variables, as shown in Table 2. It can be seen that both instances greatly depend on \( n_z \). Suppose that there exists an optimal supervisor with the minimal number of control places for a given Petri net model and denote the number as \( n_z^* \). If \( n_z \) is too small, i.e., \( n_z < n_z^* \), MNCP has no solution. In this case, we have to increase \( n_z \) and solve MNCP again. Therefore, the proper range for \( n_z \) is from \( n_z^* \) to \( |\Omega^*| \). The best case is that \( n_z \) initially happens to be \( n_z^* \). Then an optimal supervisor with a minimal structure can be obtained and MNCP has a small number of constraints and variables. However, given a net model, we do not know the exact value of \( n_z^* \) before solving MNCP. That is to say, it is hard to set \( n_z \) initially. In fact, the closer the given \( n_z \) is to \( n_z^* \), the more efficient the proposed method is. In the following, we provide three methods to give an initial value for \( n_z \).

First, we can initially assign \( n_z \) as 1. If \( n_z < n_z^* \), MNCP has no solution. Then, we increase \( n_z \) and solve MNCP again. The process carries out until \( n_z = n_z^* \). Then, MNCP has an optimal solution. By using this method, the number of iterations is \( n_z^* \).

Second, we can propose an initial value by other existing effective methods and make it in the range from \( n_z^* \) to \( |\Omega^*| \). For instance, in [11], we propose an iterative approach to obtain an optimal supervisor with self-loops. The approach is rather efficient though it cannot guarantee the minimality of the supervisory structure. Suppose that the method proposed in [11] obtains a supervisor with \( n_z \) control places. Then, we have \( n_z^* \leq n_z^* \). Just as stated in [11], \( n_z^* \) is very close to \( n_z^* \). Thus, we can initially set \( n_z = n_z^* \).

Third, we can assign \( n_z = |T_c| \). The work in [11] shows that the number of control places is equal to \( |T_c| \) in the best case. Thus, without the solution of any other work, we can initially make \( n_z = |T_c| \). If \( n_z^* \leq |T_c| \), MNCP has a solution with \( n_z = |T_c| \). Otherwise, we can increase \( n_z \) and solve MNCP again. In this case, the number of iteration is greatly reduced and MNCP becomes effective and efficient.

### 3.3. An illustrative example

In this section, the Petri net model of an FMS is used to illustrate the proposed approaches. In the following, for the sake of simplicity, we introduce a multiset formalism to represent the markings in \( \mathbb{M}_i^* \) and \( \mathbb{C}_k^* \). Since only operation places are considered, we use \( \sum_{i \in \mathbb{N}_o} M_i(p_i) \) to denote marking \( M \) in \( \mathbb{M}_i^* \) or \( \mathbb{C}_k^* \).
Fig. 1 shows the Petri net model of an FMS. The places have the following set partitions: \( P^0 = \{p_1, p_3\} \), \( P_k = \{p_0 - p_1\} \), and \( P_5 = \{p_2 - p_3\} \). The model has 39 reachable markings, 23 of which are legal. Its reachability graph is shown in Fig. 2. By using the marking reduction approach, we have \( \mathcal{M}_1 = \{2p_5 + p_6 + p_7, 2p_3 + p_3 + p_3\} \). It has two critical transitions \( t_1 \) and \( t_6 \), i.e., \( T_c = \{t_1, t_5\} \), and 10 MTSIs that are classified into two sets: \( \Omega_c = \{(M_2, t_1), (M_6, t_1), (M_7, t_1), (M_{14}, t_1), (M_{22}, t_1)\} \) and \( \Omega_m = \{(M_1, t_5), (M_6, t_5), (M_7, t_5), (M_{15}, t_5), (M_{16}, t_5)\} \). By using the reduction technique, the two sets are reduced to be \( \Omega_c^* = \{(M_2, t_1), (M_7, t_1)\} \) and \( \Omega_m^* = \{(M_1, t_5), (M_6, t_5)\} \). Thus, the minimal covered set \( \Omega^* \) of MTSIs is \( \Omega^* = \{(M_2, t_1), (M_7, t_1), (M_1, t_5), (M_{15}, t_5)\} \). The set of \( t_1 \)-enabled good markings is \( \mathcal{E}_{t_1} = \{(M_0, M_1, M_4, M_8, M_{10}, M_{17}, M_{22}, M_{33})\} \) and the set of \( t_5 \)-enabled good markings is \( \mathcal{E}_{t_5} = \{(M_0, M_2, M_7, M_{14}, M_{24}, M_{32}, M_{36})\} \). By using the reduction technique, we obtain \( \mathcal{E}_{t_1}^* = \{(M_{23})\} \) and \( \mathcal{E}_{t_5}^* = \{(M_{35})\} \). Specially, we have \( \mathcal{E}_{t_1}^* = \{(M_{35})\} \) and \( \mathcal{E}_{t_5}^* = \{(M_{35})\} \). The set of \( t_1 \)-enabled safe markings is \( \mathcal{S}_{t_1} = \{(M_0, M_1, M_4, M_8, M_{10}, M_{17}, M_{22}, M_{33})\} \) and the set of \( t_5 \)-enabled safe markings is \( \mathcal{S}_{t_5} = \{(M_0, M_2, M_7, M_{14}, M_{24}, M_{32}, M_{36})\} \). By using the reduction technique, we obtain \( \mathcal{S}_{t_1}^* = \{(M_{23})\} \) and \( \mathcal{S}_{t_5}^* = \{(M_{35})\} \). Specially, we have \( \mathcal{S}_{t_1}^* = \{(M_{35})\} \) and \( \mathcal{S}_{t_5}^* = \{(M_{35})\} \).

First, we consider Algorithm 1. Suppose that there is a control place \( p_{\text{s}1} \) with a self-loop associated with each critical transition, i.e., \( W(p_{\text{s}1}, t_1) = W(t_1, p_{\text{s}1}) = w_1 \) and \( W(p_{\text{s}1}, t_5) = W(t_5, p_{\text{s}1}) = w_2 \). No legal marking is forbidden by \( p_{\text{s}1} \). Therefore, considering the two legal markings in \( \mathcal{M}_1^* \), we have
\[
2l_2 + l_3 + l_4 \leq \beta \quad \text{and} \\
2l_5 + l_6 + l_7 \leq \beta.
\]

Control place \( p_{\text{s}1} \) should not disable the firing of \( t_1 \) at its enabled good markings. By considering \( \sum_{i=2}^{4} l_1 \cdot N(p_{\text{s}1}, t_1) = l_2 \), for
the marking in $E_t^*$, we have
\[ 2l_2 + l_3 + l_4 \leq \beta - w_1. \]

Similarly, for $t_5$, we have $\sum_{i=2}^7 l_i \cdot (N(p_i, t_5) = l_5$. Hence, the enabled condition for the marking in $E_t^*$ is:
\[ 2l_5 + l_6 + l_7 \leq \beta - w_5. \]

Four binary variables $f_{2.1}, f_{7.1}, f_{1.5},$ and $f_{4.5}$ are introduced to represent whether $p_{i}$ implements the four MTSIs in $\Omega^*$, i.e., $(M_2, t_1), (M_7, t_1), (M_1, t_5)$, and $(M_4, t_5)$, respectively. Hence, we have
\[ l_2 + l_5 \geq -Q \cdot (1 - f_{2.1}) + \beta - w_1 + 1, \]
\[ l_2 + l_6 \geq -Q \cdot (1 - f_{7.1}) + \beta - w_1 + 1, \]
\[ l_2 + l_3 \geq -Q \cdot (1 - f_{1.5}) + \beta - w_5 + 1, \quad \text{and} \]
\[ l_2 + l_4 \geq -Q \cdot (1 - f_{4.5}) + \beta - w_5 + 1. \]

An objective function is used to maximize the number of MTSIs that are implemented by $p_{i}$, as presented below:
\[ \text{max } f = f_{2.1} + f_{7.1} + f_{1.5} + f_{4.5}. \]

By summarizing the above constraints, we formulate an ILPP, IMNM, as follows:
\[ \text{IMNM:} \]
\[ \text{max } f = f_{2.1} + f_{7.1} + f_{1.5} + f_{4.5} \]
subject to
\[ 2l_2 + l_3 + l_4 \leq \beta \]
\[ 2l_5 + l_6 + l_7 \leq \beta \]
\[ 2l_2 + l_3 + l_4 \leq \beta - w_1 \]
\[ 2l_5 + l_6 + l_7 \leq \beta - w_5 \]
\[ l_2 + l_5 \geq -Q \cdot (1 - f_{2.1}) + \beta - w_1 + 1 \]
\[ l_2 + l_6 \geq -Q \cdot (1 - f_{7.1}) + \beta - w_1 + 1 \]
\[ l_2 + l_3 \geq -Q \cdot (1 - f_{1.5}) + \beta - w_5 + 1 \]
\[ l_2 + l_4 \geq -Q \cdot (1 - f_{4.5}) + \beta - w_5 + 1 \]
\[ l_2, l_3, l_4, l_5, l_6, l_7 \in \{0, 1, 2, \ldots\} \]
\[ \beta \in \{0, 1, 2, 3, \ldots\} \]
\[ w_1, w_5 \in \{0, 1, 2, 3, \ldots\} \]
\[ f_{2.1}, f_{7.1}, f_{1.5}, f_{4.5} \in \{0, 1\} \]

The above ILPP has an optimal solution with $l_2 = l_5 = 1$, \( \beta = w_5 = 3 \), and $f^* = 2$. Since $w_1 = 0$ in the solution, there is no self-loop associated with $t_3$. Then, we can design a control place $p_{i}$ with a self-loop associated with $t_5$, where $p_{i}$ satisfies constraint: $\mu_{l_5} + \mu_5 \leq 3$ and the weight of the self-loop is 3. That is to say, \(*p_{i} = \{t_3, 3t_1\}, \mu_{l_5} = \{3t_5, t_6\}, \text{ and } M_0(p_{i}) = 3$. Removing the MTSIs implemented by $p_{i}$, we have $\Omega^* = (M_1, t_5), (M_4, t_5)).$

At the second iteration, only $t_5$ is required to be disabled. Thus, we consider a control place $p_{i}$ with a self-loop associated with $t_5$. Then, IMNM can be easily obtained by removing the enabled conditions for $E_t^*$ and the disabled conditions for MTSIs implemented by $p_{i}$ from the IMNM at the first iteration, as presented below:
\[ \text{IMNM:} \]
\[ \text{max } f = f_{1.5} + f_{4.5} \]
subject to
\[ 2l_2 + l_3 + l_4 \leq \beta \]
\[ 2l_5 + l_6 + l_7 \leq \beta \]
\[ 2l_5 + l_6 + l_7 \leq \beta - w_5 \]
\[ l_2 + l_5 \geq -Q \cdot (1 - f_{1.5}) + \beta - w_5 + 1 \]
\[ l_3 + l_5 \geq -Q \cdot (1 - f_{4.5}) + \beta - w_5 + 1 \]
\[ l_2, l_3, l_4, l_5, l_6, l_7 \in \{0, 1, 2, 3, \ldots\} \]
\[ \beta, w_5 \in \{0, 1, 2, 3, \ldots\} \]
\[ f_{1.5}, f_{4.5} \in \{0, 1\} \]

The above ILPP has an optimal solution with $l_3 = l_7 = 1$, \( \beta = w_5 = 3 \), and $f^* = 2$. Then we can design a control place $p_{i}$ with a self-loop associated with $t_5$, where $p_{i}$ satisfies constraints: $\mu_2 + \mu_5 \leq 3$ and hence the weight of the self-loop is 3. That is to say, \(*p_{i} = \{t_3, 3t_5\}, \mu_{l_5} = \{3t_5, t_6\}, \text{ and } M_0(p_{i}) = 3$. Finally, two control places are obtained, which can make the net model live with all 23 legal markings. The results are shown in Table 3, where $i$ is the iteration number, $N_{\text{con}}$ and $N_{\text{var}}$ indicate the numbers of constraints and variables in IMNM, respectively, $l_i$ is the computed PI, $w_5$ indicates the weight of the self-loop associated with $t_5$, and $f_j, f_j^*$ in the seventh to ninth columns show the preset, poset, and initial marking of the computed control place $p_{i}$, respectively. The final optimally controlled system is shown in Fig. 3.

Next, we show MNCP for this example. Suppose that $n_c = |T_c| = 2$, then MNCP is:
\[ \text{MNCP:} \]
\[ \text{min } h = h_1 + h_2 \]
subject to
\[ 2h_2 + h_3 + h_4 \leq \beta_1 \]
\[ 2h_5 + h_6 + h_7 \leq \beta_1 \]
\[ 2h_2 + h_3 + h_4 \leq \beta_2 \]
weight of the self-loop between \( p_0 \) and \( t_0 \), and \( *p_0, p_0^* \), and \( M_0(p_0) \) in the fourth to sixth columns show the preset, poset, and initial marking of the controlled place \( p_0 \), respectively. Note that MNCP obtains the same supervisor as Algorithm 1 does.

This net model has no optimal pure Petri net supervisor. Thus, we cannot find an optimal Petri net supervisor by the work in [25, 52, 6, 7]. However, both the proposed approaches and the previous work [11] can provide a non-pure optimal supervisor with self-loops, which has the minimal number of control places. The final controlled system has all 23 legal markings. Thus, it can be seen that Petri net supervisors with self-loops are more powerful than pure net supervisors. In next section, some examples are

\[
2l_{2,5} + l_{2,6} + l_{2,7} \leq \beta_2 \\
2l_{1,2} + l_{1,3} + l_{1,4} \leq \beta_1 - w_{1,1} \\
2l_{1,5} + l_{1,6} + l_{1,7} \leq \beta_1 - w_{1,5} \\
2l_{2,2} + l_{2,3} + l_{2,4} \leq \beta_2 - w_{2,1} \\
2l_{2,5} + l_{2,6} + l_{2,7} \leq \beta_2 - w_{2,5} \\
l_{1,2} + l_{1,5} \geq -Q \cdot (1 - f_{1,2,1}) + \beta_1 - w_{1,1} + 1 \\
l_{1,2} + l_{1,6} \geq -Q \cdot (1 - f_{1,7,1}) + \beta_1 - w_{1,1} + 1 \\
l_{1,2} + l_{5} \geq -Q \cdot (1 - f_{1,1,5}) + \beta_1 - w_{1,5} + 1 \\
l_{1,3} + l_{5} \geq -Q \cdot (1 - f_{1,4,5}) + \beta_1 - w_{1,5} + 1 \\
l_{2,2} + l_{5} \geq -Q \cdot (1 - f_{2,2,1}) + \beta_2 - w_{2,1} + 1 \\
l_{2,2} + l_{6} \geq -Q \cdot (1 - f_{2,7,1}) + \beta_2 - w_{2,1} + 1 \\
l_{2,2} + l_{5} \geq -Q \cdot (1 - f_{2,1,5}) + \beta_2 - w_{2,5} + 1 \\
l_{2,3} + l_{5} \geq -Q \cdot (1 - f_{2,4,5}) + \beta_2 - w_{2,5} + 1
\]

\[l_{1,2,1} \leq h_1\]
\[l_{1,7,1} \leq h_1\]
\[l_{1,1,5} \leq h_1\]
\[l_{1,4,5} \leq h_1\]
\[l_{2,2,1} \leq h_2\]
\[l_{2,7,1} \leq h_2\]
\[l_{2,1,5} \leq h_2\]
\[l_{2,4,5} \leq h_2\]
\[l_{1,2,1} + f_{2,2,1} \geq 1\]
\[l_{1,7,1} + f_{2,7,1} \geq 1\]
\[l_{1,1,5} + f_{2,1,5} \geq 1\]
\[l_{1,4,5} + f_{2,4,5} \geq 1\]
\[l_{1,2,1,1,3,1,4,1,5,1,6,1,7} \in \{0, 1, \ldots\}\]
\[l_{2,2,1,1,2,3,1,4,2,5,1,6,2,7} \in \{0, 1, \ldots\}\]
\[\beta_1, \beta_2 \in \{0, 1, \ldots\}\]
\[w_{1,1,1,1,2,2,1,2,5} \leq \{0, 1, \ldots\}\]
\[f_{1,2,1,1,3,1,4,1,5,1,6,1,7} \in \{0, 1, \ldots\}\]

In MNCP, Eqs. (21)–(24) are reachability conditions, Eqs. (25)–(28) represent enabled conditions, and Eqs. (29)–(36) imply disabled conditions. Eqs. (37)–(44) indicate that a control place does not implement any MTSI if it is not selected. Eqs. (45)–(48) ensure that any MTSI must be implemented by at least one control place.

The above MNCP has 28 constraints and 26 variables. By solving it, we can obtain an optimal solution with \( l_{1,5} = l_{1,6} = l_{2,2} = l_{2,3} = 1, \beta_1 = \beta_2 = w_{1,1,1} = w_{2,5} = 3, \) and \( h^* = 2. \) Thus, we can obtain two control places \( p_0 \) and \( p_5. \) Specifically, we have \( *p_0 = (3t_1, t_1), p_0^* = (3t_3, t_3), M_0(p_0) = 3, *p_5 = (t_3, 3t_5), p_5^* = (t_1, 3t_5), \) and \( M_0(p_5) = 3. \) The results by MNCP are shown in Table 4, where \( z \) indicates the \( z \)-th control place, \( l_z \) is the \( z \)-th PI, \( w_{z,q} \) indicates the

| No. monitors | 4 | 3 | 3 | 2 | 2 |
| No. arcs | 13 | 10 | 10 | 9 | 9 |
| No. markings | 10 | 11 | 13 | 13 | 13 |
provided to show that both Algorithm 1 and MNCP can find an optimal supervisor with less number of control places than that proposed in the previous work [11].

4. Experimental results

In this section, two Petri net models of FMSs are used to illustrate the proposed methods. First, we consider an SₚPR from [51], as shown in Fig. 4. There are nine places and seven transitions and the places have the following set partitions: \( I^0 = \{ P_{10}, P_{20} \} \), \( I^1 = \{ P_{31}, P_{32} \} \), and \( I^2 = \{ P_{11}, P_{12}, P_{13}, P_{21}, P_{22} \} \). It has 20 reachable markings, 13 of which are legal. By using the vector covering approach, we have the following set partitions:

\[ \mathcal{M}_n^* = \{ 21 \} \]

The net has seven MTSIs and three critical transitions \( t_1, t_2, t_5 \), where \( |E_{r_1}| = 5 \), \( |E_{r_2}| = 2 \), and \( |E_{r_5}| = 4 \). By using the marking reduction approach, we have \( |E_{r_{11}}^*| = 2 \), \( |E_{r_{12}}^*| = 1 \), \( |E_{r_{13}}^*| = 1 \), and \( |E_{r_{14}}^*| = 4 \). Tables 5 and 6 show the applications of Algorithm 1 and MNCP, respectively. For this net model, MNCP with \( n_{c} = 2 \) has 34 constraints and 28 variables.

The net model does not have an optimal pure Petri net supervisor by using the theory of regions in [25,52] or the methods proposed in the previous work [6,7]. In the previous work [12], we propose an optimal supervisor to enforce disjunctive constraints. However, the supervisory structure in [12] includes four added transitions, which lead to six extra markings in the controlled system. The previous work [11] and the proposed methods in this paper can find optimal supervisors with

Table 8
Experimental results for the net shown in Fig. 5 by Algorithm 1.

<table>
<thead>
<tr>
<th>( i )</th>
<th>( N_{mon} )</th>
<th>( N_{var} )</th>
<th>( h^* )</th>
<th>( l_i )</th>
<th>( w_q )</th>
<th>( p^*_q )</th>
<th>( p^*_i )</th>
<th>( M_0(p^*_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>20</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>18</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 9
Experimental results for the net shown in Fig. 5 by MNCP.

<table>
<thead>
<tr>
<th>( z )</th>
<th>( l_z )</th>
<th>( w_{z,q} )</th>
<th>( p^*_q )</th>
<th>( p^*_i )</th>
<th>( M_0(p^*_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \mu_z + 2 \mu_1 + 2 \mu_2 + 2 \mu_3 + 3 \mu_9 + 3 \mu_{10} \leq 9 )</td>
<td>( w_{1,z} = 1, w_{2,z} = 2 )</td>
<td>( 2r_1, 3r_1, 2r_2, 2r_3, 2r_4, 2r_5 )</td>
<td>( 2r_1, 2r_2, 2r_3, 2r_4, 2r_5 )</td>
<td>( 9 )</td>
</tr>
<tr>
<td>2</td>
<td>( \mu_z + 2 \mu_1 + 2 \mu_2 + 2 \mu_3 + 3 \mu_9 + 3 \mu_{10} \leq 9 )</td>
<td>( w_{2,z} = 2 )</td>
<td>( 2r_1, 2r_2, 2r_3, 2r_4, 2r_5 )</td>
<td>( 2r_1, 2r_2, 2r_3, 2r_4, 2r_5 )</td>
<td>( 3 )</td>
</tr>
</tbody>
</table>
Table 11
Experimental results for the net shown in Fig. 6 by Algorithm 1.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$N_{ran}$</th>
<th>$N_{inv}$</th>
<th>$h^*$</th>
<th>$l_i$</th>
<th>$w_i$</th>
<th>$\mu_p$</th>
<th>$\mu_0$</th>
<th>$\text{Mo}(\mu_p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>47</td>
<td>260</td>
<td>18</td>
<td>$p_3 + 3p12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>204</td>
<td>29</td>
<td>3</td>
<td>$\mu_3 + 3\mu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>133</td>
<td>32</td>
<td>2</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>97</td>
<td>19</td>
<td>1</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td></td>
</tr>
</tbody>
</table>

Table 12
Experimental results for the net shown in Fig. 6 by MNCP.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$L_i$</th>
<th>$w_i$</th>
<th>$\mu_p$</th>
<th>$\mu_0$</th>
<th>$\text{Mo}(\mu_p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\mu_3 + 3\mu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td>$\nu_3 + 3\nu_12 \leq 5$</td>
<td></td>
</tr>
</tbody>
</table>

self-loops. Table 7 shows the performance comparison of some policies for this example. The work in [11] and the proposed approaches can lead to optimal supervisors to make all 13 legal markings reachable. Note that in [11], the optimal supervisor has three control places and 10 arcs. However, considering our solutions to this problem, both Algorithm 1 and MNCP can obtain an optimal supervisor with only two control places and nine arcs, which is the simplest supervisory structure.

Next, we consider a well-known example as shown in Fig. 5. It is studied in several papers (see [35,46,47,52]). There are 19 places and 14 transitions. The places have the following set partitions: $P^0 = \{p_1, p_3\}$, $P_2 = \{p_4-p_{10}\}$, and $P_3 = \{p_2-p_7, p_8-p_{13}\}$. It has 282 reachable markings, 205 of which are legal ones. By using a vector covering approach, $M_t^*$ has only 26 markings. The net has 59 MTSIs and five critical transitions, $t_1$, $t_2$, $t_4$, $t_6$, and $t_8$, where $|E_{t_1}| = 46$, $|E_{t_2}| = 22$, $|E_{t_4}| = 32$, $|E_{t_6}| = 63$, and $|E_{t_8}| = 12$. By using the marking reduction approach, we have $|E_{t_1}| = 8$, $|E_{t_2}| = 3$, $|E_{t_4}| = 5$, $|E_{t_6}| = 15$, $|E_{t_8}| = 2$, and $|\Omega^*| = 12$. Tables 8 and 9 show the applications of Algorithm 1 and MNCP, respectively. For this net model, MNCP with $n_\text{c} = 4$ has 1352 constraints and 192 variables. Note that in the first row of Table 9, we obtain $w_{t_4} = \emptyset$, $\forall t \in E_{t_4}$, i.e., no self-loop is added for control place $p_7$.

Table 10 shows the performance comparison of some policies for this example. It can be seen that the previous work [11] obtains an optimal solution with $|T_c| = 5$ control places, which is bounded by $|T|$ However, both Algorithm 1 and MNCP can find an optimal supervisor with only two control places, which is three less than that in the previous work [11] and the same as the minimal structure proposed in [66]. However, the proposed methods lead to more arcs in the obtained supervisors. Note that in [14], we propose an approach to find an optimal supervisor with the lowest implementation cost, which also leads to the minimal supervisory structure with two control places and 12 arcs.

Consider another Petri net model of an FMS, as shown in Fig. 6. It is a WS$^3$PR that is a weighted version of an S$^3$PR in [21]. Comparing with S$^3$PR, both operation places $p_{13}$ and $p_{20}$ in this model have weighted resource requirements. There are 26 places and 20 transitions and the places have the following set partitions: $P^0 = \{p_1, p_5, p_{14}\}$, $P_2 = \{p_20-p_{26}\}$, and $P_3 = \{p_2-p_4, p_8-p_{13}, p_{15}-p_{19}\}$. It has 9572 reachable markings, 5395 of which are legal ones. By using a vector covering approach, $M_t^*$ has only 5977 markings. The net has 1915 MTSIs and six critical transitions $t_1$, $t_2$, $t_4$, $t_{11}$, $t_{15}$, and $t_7$, where $|E_{t_1}| = 2040$.

Table 13
Performance comparison of some deadlock control policies for the net shown in Fig. 6.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No. monitors</td>
<td>12</td>
<td>8</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>No. arcs</td>
<td>57</td>
<td>42</td>
<td>33</td>
<td>39</td>
</tr>
<tr>
<td>No. markings</td>
<td>4795</td>
<td>5395</td>
<td>5395</td>
<td>5395</td>
</tr>
</tbody>
</table>
models that cannot be optimally controlled by pure Petri net supervisors. Compared with the work in [12], the obtained supervisor in this work does not include added transitions and no extra markings generated in the controlled system. Compared with the previous work [11], this research can obtain optimal supervisors with less control places and MNCP can even find the minimal number of control places.

Though this work can handle the behavioral permissiveness- and structural complexity problems, it suffers from the computational complexity problem. It requires to generate the reachable graph of a Petri net model and solve ILPPs that are NP-hard in theory. Although we use a marking reduction technique to reduce the numbers of constraints and variables in the ILPPs, which can greatly reduce the computational burden of the proposed methods, their computational complexity is still NP-hard. In the future, we will focus on avoiding the full enumeration of reachable markings [43] and reduce the number of constraints in the proposed ILPPs.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant nos. 61203038, 61374068, 61304050, 61304051, and 61364004, the Fundamental Research Funds for the Central Universities under Grant no. JB140402, the Recruitment Program of Global Experts, the Science and Technology Development Fund, MSAR, under Grant no. 066/2013/A2, and the Scientific and Technological Research Council of Turkey (Türkiye Bilimsel ve Teknolojik Araştırma Kurumu – TÜBİTAK) under Grant no. TÜBİTAK-112M229.

References


