

## Research Article

# Supervisor Reconfiguration for Deadlock Prevention by Resources Reallocation

Miao Liu,<sup>1</sup> Shouguang Wang,<sup>2</sup> and Zhiwu Li<sup>1</sup>

<sup>1</sup> School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China

<sup>2</sup> School of Information & Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China

Correspondence should be addressed to Zhiwu Li; [systemscontrol@gmail.com](mailto:systemscontrol@gmail.com)

Received 12 December 2012; Accepted 9 March 2013

Academic Editor: Constantinos Siettos

Copyright © 2013 Miao Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Analysis and control of deadlocks play an important role in the design and operation of automated flexible manufacturing systems (FMSs). In FMS, deadlocks are highly undesirable situations, which always cause unnecessary cost. The design problem of an optimal supervisor is in general NP-hard. A computationally efficient method often ends up with a suboptimal one. This paper develops a deadlock prevention policy based on resources reallocation and supervisor reconfiguration. First, given a plant model, we reallocate the marking of each resource place to be one, obtaining a net model whose reachable states are much less than that of the original one. In this case, we find a controlled system for it by using the theory of regions. Next, the markings of the resource places in the controlled system are restored to their original ones. Without changing the structure of the obtained controlled system, we compute the markings of the monitors gradually, which can be realized by two algorithms proposed in this paper. Finally, we decide a marking for each monitor such that it makes the controlled system live with nearly optimal permissive behavior. Two FMS examples are used to illustrate the application of the proposed method and show its superior efficiency.

## 1. Introduction

Traditional mass production systems can hardly cope with intensive competition in market and rapid variation in requirements. Hence, automated flexible manufacturing systems (FMSs) arise, aiming to offer a novel production mode with a small batch and multiple product types. The analysis and control of such systems have become the hot topics in the field of manufacturing systems. Petri nets [1] are a graph-based mathematical formalism suitable to describe, model, and analyze the behavior of automated flexible manufacturing systems.

Due to the existence of shared resources, an FMS may contain deadlocks. Based on Petri nets, researchers have developed many policies to deal with the deadlock problem [2–6] in FMS. Generally, there are mainly two analysis techniques to deal with deadlock prevention in FMS: structure [7–11] and reachability graph analysis [12–16]. The former always obtains a deadlock prevention policy through special structural objects of a Petri net such as siphons and resource-transition circuits. This method can usually obtain

a computationally efficient liveness-enforcing supervisor in general but at the same time restrict a system such that a portion of permissive behavior is excluded. For the latter, the reachability graph can completely reflect the behavior of a system. Though a very highly or even maximally permissive liveness-enforcing supervisor can always be obtained, its computation is very expensive.

As stated, reachability graph analysis [17] is an important technique for deadlock control; however, it always suffers from a state explosion problem. This is due to the fact that it is impossible to achieve the enumeration of all or a part of reachable markings in practice. Based on this technique, an optimal or suboptimal supervisor with highly behavioral permissiveness can always be obtained for a small system. In [18], Uzam and Zhou develop an iterative approach to design an optimal or suboptimal supervisor. This method is easy to use if the reachable space of a system is small but cannot guarantee the optimality of the supervisor.

The theory of regions developed in [19] can be used as an effective approach to find an optimal liveness-enforcing Petri net supervisor if such a supervisor exists. However, it suffers

from computational and structural complexity problems. The work in [20] proposes a vector covering approach to improve the computation efficiency of the work in [19].

A siphon-based deadlock prevention policy is a typical application of structure analysis techniques of Petri nets. Although not optimal and even overly restrictive, this approach is computationally tractable and allows its supervisor to be reused when a system experiences new job instances [2, 21–28]. Recent effective and computationally efficient deadlock prevention policies are proposed by Piroddi et al. in [29, 30].

Nowadays, the distribution of resources in an FMS may change frequently and dynamically due to fluctuant customer demands. Such changes mean different production or service requirements for the providers. Therefore, the supervisory control system has to reconfigure rapidly in response to the changes in its physical entity and the control specifications by modifying or adjusting its plant model and controller. Once the resource configurations are changed, the supervisors are updated accordingly. Recently, the study in [31] proposes a novel deadlock prevention policy based on reconfiguration of Petri net supervisors. The method presented in [31] is near optimal, but its performance needs to be further improved.

Motivated by existing work, this paper presents an effective and computationally efficient method to design nearly optimal control places based on resources reallocation and supervisor reconfiguration. The main idea of the method is stated as follows.

- (i) The proposed method works on the premise of a class of ordinary Petri net  $(N, M_0)$  with  $N = (P^0 \cup P_A \cup P_R, T, F)$ , where for all  $i \in \{1, 2, \dots, n\}$ ,  $P^0 = \cup_{i=1}^n \{p_i^0\}$  is called a set of idle process places,  $P_A = \cup_{i=1}^n P_{A_i}$  is called a set of activity places, and  $P_R = \cup_{i=1}^n P_{R_i}$  is called a set of resource places.
- (ii) We first reallocate the initial marking of each resource place in  $(N, M_0)$  to be one, obtaining a model  $(N^1, M^1)$  where  $N^1 = N$ , for all  $p \in P^0 \cup P_A$ ,  $M^1(p) = M_0(p)$ , and for all  $r \in P_R$ ,  $M^1(r) = 1$ . In this case, we can design monitors by using the theory of regions and find a controlled system  $(N^{1c}, M^{1c})$  for  $(N^1, M^1)$  with  $N^{1c} = (P^0 \cup P_A \cup P_R \cup P_V, T^{1c}, F^{1c})$ , where for all  $p \in P^0 \cup P_A \cup P_R$ ,  $M^{1c}(p) = M^1(p)$ , for all  $i \in \{1, 2, \dots, n\}$ ,  $P_V = \cup_{i=1}^n P_{V_i}$  is called a set of control places.
- (iii) Then, the markings of the resource places in  $(N^{1c}, M^{1c})$  are restored to their original ones, obtaining a controlled system  $(N^c, M^c)$  with  $N^c = N^{1c}$ , where for all  $p \in P^0 \cup P_A \cup P_R$ ,  $M^c(p) = M_0(p)$ , and for all  $v \in P_V$ ,  $M^c(v) = M^{1c}(v)$ .
- (iv) Without changing the structure of the controlled system  $(N^c, M^c)$ , we compute the markings of the monitors gradually. Finally, we decide a marking for each monitor such that for all  $p \in P^0 \cup P_A \cup P_R$ ,  $M^c(p) = M_0(p)$ , and for all  $v \in P_V$ ,  $M^c(v)$  makes the controlled system  $(N^c, M^c)$  live, where  $(N^c, M^c)$  is a controlled system for  $(N, M_0)$  with  $N^c = N^{1c}$ .

The rest of this paper is organized as follows. Section 2 briefly reviews preliminaries used in this paper. Section 3 formulates the considered problem through a motivation example. Section 4 presents a deadlock prevention policy that is formalized by Algorithm 2. Two FMS examples are given in Section 5, showing the superiority of the proposed method. A comparison between the proposed method and a previous one is made in Section 6. Finally, Section 7 concludes this paper.

## 2. Preliminaries

*2.1. Basics of Petri Nets.* A generalized Petri net (structure) [7] is a four-tuple  $N = (P, T, F, W)$ , where  $P$  and  $T$  are finite, nonempty, and disjoint sets.  $P$  is a set of places and  $T$  is a set of transitions with  $P \cup T \neq \emptyset$  and  $P \cap T = \emptyset$ .  $F \subseteq (P \times T) \cup (T \times P)$  is called a flow relation of a net, represented by arcs with arrows from places to transitions or from transitions to places.  $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$  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, where  $x, y \in P \cup T$  and  $\mathbb{N} = \{0, 1, 2, \dots\}$ . If  $W(x, y) = 1$ , for all  $(x, y) \in F$ , the net is called an ordinary Petri net.

A transition  $t \in T$  is enabled at a marking  $M$  if for all  $p \in {}^*t$ ,  $M(p) \geq W(p, t)$ . This fact is denoted as  $M[t]$ . Firing it yields a new marking  $M'$  such that for all  $p \in P$ ,  $M'(p) = M(p) - W(p, t) + W(t, p)$ , as denoted by  $M[t]M'$ .  $M'$  is called an immediately reachable marking from  $M$ . Marking  $M''$  is said to be reachable from  $M$  if there exists a sequence of transitions  $\sigma = t_0 t_1 \dots t_n$  and markings  $M_1, M_2, \dots$ , and  $M_n$  such that  $M[t_0]M_1[t_1]M_2 \dots M_n[t_n]M''$  holds. The set of markings reachable from  $M$  in  $N$  is called the reachability set of Petri net  $(N, M)$  and denoted as  $R(N, M)$ .  $[N]$  is called the incidence matrix of  $N$ . It is a  $|P| \times |T|$  integer matrix with  $[N](p, t) = W(t, p) - W(p, t)$ .

A  $P$ -vector is a column vector  $I : P \rightarrow \mathbb{Z}$  index by  $P$  and a  $T$ -vector is a column vector  $J : T \rightarrow \mathbb{Z}$  index by  $T$ , where  $\mathbb{Z}$  is the set of integers.  $I$  is a  $P$ -invariant if  $I \neq \mathbf{0}$  and  $I^T[N] = \mathbf{0}^T$ . A  $P$ -invariant  $I$  is said to be a  $P$ -semiflow if every element of  $I$  is nonnegative.

A nonempty set  $S \subseteq P$  is a siphon if  ${}^*S \subseteq S^*$ . A siphon is minimal if there is no siphon contained in it as a proper subset. A minimal siphon that does not contain the support of any  $P$ -invariant is called a strict minimal siphon (SMS).

A siphon  $S$  is said to be max-marked at  $M \in R(N, M_0)$  if  $\exists p \in S$  such that  $M(p) \geq \max_{p^*}$ , where  $\max_{p^*} = \max\{W(p, t) \mid t \in p^*\}$ .  $S$  is max-controlled if it is max-marked at any reachable marking.  $(N, M_0)$  satisfies the maximal controlled-siphon (cs) property if each minimal siphon of  $N$  is max-controlled [33]. Siphon  $S$  is called uncontrolled in  $(N, M_0)$  if  $\exists M \in R(N, M_0)$ , for all  $t \in S^*$ ,  $t$  is dead at  $M$ .

A marking  $M$  of a Petri net  $N$  is a mapping from  $P$  to  $\mathbb{N}$ .  $M(p)$  denotes the number of tokens in place  $p$ . A place  $p$  is marked at a marking  $M$  if  $M(p) > 0$ . A subnet  $S \subseteq P$  is marked at a marking  $M$  if at least one place in  $S$  is marked at  $M$ . The sum of tokens of all places in  $S$  is denoted by  $M(S)$ , that is,  $M(S) = \sum_{p \in S} M(p)$ .  $S$  is said to be empty at  $M$  if  $M(S) = 0$ .  $(N, M_0)$  is called a net system or marked net and  $M_0$  is called an initial marking of  $N$ .

```

Input: an ordinary M-net  $(N, M_0)$  with  $N = (P^0 \cup P_A \cup P_R, T, F, W)$ 
Output:  $(N^{1c}, M^{1c})$ 
begin {
 $N^1 := N$ 
 $\forall p \in P^0 \cup P_A, M^1(p) := M_0(p)$ 
 $\forall r \in P_R, M^1(r) := 1$ 
if {there exists an optimal controlled system for  $(N^1, M^1)$ } then
  design a controlled system  $(N^{1c}, M^{1c})$  for  $(N^1, M^1)$  by the theory of regions
else
  design a controlled system  $(N^{1c}, M^{1c})$  for  $(N^1, M^1)$  by the method in [18]
end if
 $\forall p \in P^0 \cup P_A \cup P_R, M^{1c}(p) := M^1(p)$ 
 $N^{1c} := (P^0 \cup P_A \cup P_R \cup P_V, T, F^{1c}, W^{1c})$ 
if  $\{(N^{1c}, M^{1c})$  is an ordinary controlled system} then
  output  $(N^{1c}, M^{1c})$ 
else
  exit and stop the algorithm
end if
output  $(N^{1c}, M^{1c})$ 
} end of the algorithm

```

ALGORITHM 1: Structure design of a controlled system for  $(N, M_0)$ .

Markings and vectors are usually represented via using a multiset. As a result, vector  $M$  is denoted by  $\sum_{p \in P} M(p)p$ . For instance, a marking that puts two tokens in place  $p_1$  and three tokens in place  $p_3$  only in a net with  $P = \{p_1 - p_6\}$  is denoted as  $M = 2p_1 + 3p_3$  instead of  $(2, 0, 3, 0, 0, 0)^T$ .

Let  $x \in P \cup T$  be a node of net  $N = (P, T, F, W)$ . The preset of  $x$  is defined as  $\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$ . While the postset of  $x$  is defined as  $x \bullet = \{y \in P \cup T \mid (x, y) \in F\}$ . Furthermore, we have  $x \bullet \bullet = \cup_{y \in x \bullet} y \bullet$  and  $\bullet \bullet x = \cup_{y \in \bullet x} y$ .

Given a Petri net  $(N, M_0)$ ,  $t \in T$  is live at  $M_0$  if for all  $M \in R(N, M_0)$ ,  $\exists M' \in R(N, M)$ ,  $M'[t]$ .  $(N, M_0)$  is live if for all  $t \in T$ ,  $t$  is live at  $M_0$ .  $(N, M_0)$  is dead at  $M_0$  if  $\nexists t \in T$ ,  $M_0[t]$ .  $(N, M_0)$  is deadlock-free if for all  $M \in R(N, M_0)$ ,  $\exists t \in T$ ,  $M[t]$ .

With respect to the concepts of supervisors and controlled systems, the reader is referred to [14]. A supervisor is said to be optimal if its resulting controlled system covers all safe states of a plant and every reachable state in the controlled system is a safe state of the plant. Such a controlled system is said to be optimal.

**2.2. M-Nets.** This paper considers a class of manufacturing-oriented Petri nets, M-nets [31]. It is a generalization of the existing net classes that can model FMS. In this paper, we just focus on the research of ordinary M-nets, a type of M-nets.

**Definition 1.** An M-net denoted by  $(N, M_0)$  satisfies the following statements:

- (1)  $N = \bigcirc_{i=1}^n N_i = (P^0 \cup P_A \cup P_R, T, F, W)$  is composed of  $n$  nets  $N_1, N_2, \dots$ , and  $N_n$ , where for all  $i \in \mathbb{N}_n$ ,  $\mathbb{N}_n = \{1, 2, \dots, n\}$ ,  $N_i = (\{p_i^0\} \cup P_{A_i} \cup P_{R_i}, T_i, F_i, W_i)$  is called a subnet of  $N$ .

- (2)  $P^0 = \cup_{i=1}^n \{p_i^0\}$  is called a set of idle process places with  $p_i^0 \neq p_j^0$ , for all  $i, j \in \mathbb{N}_n$ ,  $i \neq j$ ;  $P_A = \cup_{i=1}^n P_{A_i}$  is called a set of activity places with  $P_{A_i} \cap P_{A_j} = \emptyset$ , for all  $i, j \in \mathbb{N}_n$ ,  $i \neq j$ ; and  $P_R = \cup_{i=1}^n P_{R_i}$  is called a set of activity places.
- (3) For all  $i, j \in \mathbb{N}_n$ ,  $i \neq j$ ,  $T_i \cap T_j = \emptyset$ .
- (4) For all  $r \in P_R$ , it is associated with a minimal  $P$ -semiflow  $I_r$  such that  $I_r(r) = 1$ , for all  $p \in P_A$ ,  $I_r(p) \geq 0$ , and for all  $p \in P^0$ ,  $I_r(p) = 0$ .
- (5) For all  $p \in P_A$ ,  $p$  is associated with a minimal  $P$ -semiflow  $I_p$ , where  $\|I_p\| \subseteq P_A$ .
- (6)  $(N_i, M_{0i})$  is quasi-live, bounded, and conservative.
- (7)  $(N'_i, M'_{0i})$  with  $N'_i = (\{p_i^0\} \cup P_{A_i}, T_i, F'_i, W'_i)$  is live, bounded, and reversible, where  $N'_i$  is the resulting net from removing resource places in  $(N_i, M_{0i})$ .
- (8) Let  $(N_i, M_{0i})(i = 1, 2)$  be two subnets with  $N_i = (\{p_i^0\} \cup P_{A_i} \cup P_{R_i}, T_i, F_i, W_i)$ . Their composition denoted by  $(N_{12}, M_{12})$  with  $N_{12} = N_1 \circ N_2 = (P_{12}^0 \cup P_{A_{12}} \cup P_{R_{12}}, T_{12}, F_{12}, W_{12})$  is defined as follows:

- (i)  $P_{12}^0 = \{p_1^0\} \cup \{p_2^0\} = \{p_1^0, p_2^0\}$ ,  $P_{A_{12}} = P_{A_1} \cup P_{A_2}$ , and  $P_{R_{12}} = P_{R_1} \cup P_{R_2}$ ,

- (ii)  $T_{12} = T_1 \cup T_2$ ,

- (iii)  $F_{12} = F_1 \cup F_2$ ,

- (iv) for all  $f \in F_1$ ,  $W(f) = W_1(f)$  and for all  $f \in F_2$ ,  $W(f) = W_2(f)$ ,

- (v) for all  $p \in \{p_1^0\} \cup P_{A_1}$ ,  $M_{12}(p) = M_{01}(p)$ ; for all  $p \in \{p_2^0\} \cup P_{A_2}$ ,  $M_{12}(p) = M_{02}(p)$ ; for all  $r \in P_{R_1} \setminus P_{R_2}$ ,  $M_{12}(r) = M_{01}(r)$ ; for all  $r \in P_{R_2} \setminus P_{R_1}$ ,  $M_{12}(r) = M_{02}(r)$ ; and for all  $r \in P_{R_1} \cap P_{R_2}$ ,  $M_{12}(r) = \max\{M_{01}(r), M_{02}(r)\}$ .

```

Input: an ordinary M-net  $(N, M_0)$  with  $N = (P^0 \cup P_A \cup P_R, T, F, W)$ 
Output: controlled system  $(N^c, M^c)$ 
begin {
  design a controlled system  $(N^{1c}, M^{1c})$  for  $(N^1, M^1)$  by Algorithm 1, then
   $N^c := N^{1c}$ 
   $\forall v \in P_V, M^c(v) := M^{1c}(v)$ 
   $\forall p \in P^0 \cup P_A \cup P_R, M^c(p) := M_0(p)$ 
   $P_i := (**H(v_i)) \cap P_R, i \in \{1, 2, \dots, n\}$ 
   $\forall p \in P_i, M^c(v_i) := \sum_{p \in P_i} M(p) - 1, i \in \{1, 2, \dots, n\}$ 
   $\vec{\mu} := [M^c(v_1), M^c(v_2), \dots, M^c(v_j), \dots, M^c(v_n)]^T, j \in \{1, 2, \dots, n\}$ 
  While  $\{\vec{\mu}$  makes  $(N^c, M^c)$  not live, which can be decided by the MIP-based deadlock detection
  method in [32]} do
    for  $j = 1$  to  $n$  do
       $M^c(v_j) --;$ 
    end for
  end while
   $\vec{l} := \vec{\mu} \quad / * \vec{l}$  denotes the marking vector
   $[M_l^c(v_1), M_l^c(v_2), \dots, M_l^c(v_j), \dots, M_l^c(v_n)]^T, j \in \{1, 2, \dots, n\} * /$ 
  for  $j = 1$  to  $n$  do
     $M^c(v_j) ++;$ 
  end for
   $\vec{h} := \vec{\mu} \quad / * \vec{h}$  denotes the marking vector
   $[M_h^c(v_1), M_h^c(v_2), \dots, M_h^c(v_j), \dots, M_h^c(v_n)]^T, j \in \{1, 2, \dots, n\} * /$ 
  while (1) do

    for  $j = 1$  to  $n$  do
      if  $(M_h^c(v_j) == M_l^c(v_j))$ 
         $M_h^c(v_j) := M^c(v_j);$ 
      else
         $M_h^c(v_j) --;$ 
        break;
      end if
    end for
    if  $\{\vec{h}$  makes  $(N^c, M^c)$  live, which can be decided by the MIP-based deadlock detection
    method in [32]} do
      break;
    end if
  end while
   $\vec{\mu} := \vec{h}$ 
   $\forall p \in P^0 \cup P_A \cup P_R, M^c(p) := M_0(p)$ 
   $N^c := N^{1c}$ 
  output  $(N^c, M^c)$ 
} end of the algorithm

```

ALGORITHM 2: Controlled system design for  $(N, M_0)$ .

- (9) The net  $N$  resulting from the composition of  $n$  subnets  $N_1, N_2, \dots$ , and  $N_n$  is defined as follows: if  $n = 1$ , then  $N = N_1$ ; if  $n > 1$ , then  $N = \bigcirc_{i=1}^n N_i = (\bigcirc_{i=1}^{n-1} N_i) \circ N_n$ .
- (10) For all  $p \in P^0, M_0(p) > 0$ ; for all  $p \in P_A, M_0(p) = 0$ ; and for all  $r \in P_R, M_0(r) \geq \max\{I_r(p) \mid p \in \|I_r\|\}$ . Such a marking is said to be an admissible initial marking.
- (11) An uncontrolled siphon in  $(N, M_0)$  contains at least one resource place and one activity place but no idle process place.
- (12)  $(N, M_0)$  is live if no siphon is uncontrolled.
- (13) If  $(N, M_0)$  is not live, liveness can be enforced by adding monitors whose addition leads to a controlled system.
- (14) Let  $(N^c, M_0^c)$  be a controlled system for  $(N, M_0)$ .  $(N^c, M_0^c)$  is live if it is ordinary and no siphon is unmarked.  $(N^c, M_0^c)$  is live if it is generalized and satisfies the controlled-siphon (cs) property.
- (15) Let  $P_V$  be the set of monitors in  $(N^c, M_0^c)$ . For all  $v \in P_V$ , there exists a minimal  $P$ -semiflow  $I_v$  such that  $I_v(v) = 1$  and for all  $p \in \|I_v\| \setminus \{v\}, p \in P_A$ .

In order to make Definition 1 clear, an example is given in Appendix A. It is easy to find that M-nets are more

general than almost all manufacturing-oriented Petri nets, for example, the ones in [2, 22, 24, 32].

**2.3. An MIP-Based Deadlock Detection Method.** In this paper, by using a technique that is called the mixed integer programming (MIP) approach proposed in [32], siphons that cause deadlocks can be detected. Let  $(N, M_0)$  be an ordinary net with  $N = (P, T, F)$  and  $S$  the maximal empty siphon at  $M$ , that is, for all  $p \notin S, M(p) > 0$ . Finding  $S$  in  $N$  is the solution of a mixed integer programming problem. For all  $p \notin S$ , let  $v_p = 1$  and for all  $t \notin S^*$ , let  $z_t = 1$ .

It is easy to see that any  $p$  with  $v_p = 1$  and any  $t$  with  $z_t = 1$  are removed from the net. Since  $S$  is a siphon, we have that for all  $t \in P^*$ ,  $v_p = 0$  implies  $z_t = 0$  and for all  $p \in t^*$ ,  $z_t = 1$  implies the truth of  $v_p = 1$ . This leads to

$$z_t \geq \sum_{p \in t} v_p - |t| + 1, \quad \forall t \in T \quad (1)$$

$$v_p \geq z_t, \quad \forall (t, p) \in F \quad (2)$$

$$v_p, z_t \in \{0, 1\}. \quad (3)$$

For a structurally bounded net, we have

$$v_p \geq \frac{M(p)}{SB(p)}, \quad \forall p \in P, \quad (4)$$

where  $SB(p) = \max\{M(p) \mid M = M_0 + [N]Y, M \geq 0, Y \geq 0\}$  is the structural bound of place  $p$ . Therefore, the maximal siphon unmarked at a given marking  $M$  can be determined by the following MIP problem and there exist siphons unmarked at  $M$  iff  $G^{MIP} < |P|$  [32]:

$$G^{MIP} = \text{Minimize} \sum_{p \in P} v_p \quad (5)$$

s.t. constraints (1)–(4) and

$$M = M_0 + [N]Y, \quad M \geq 0, Y \geq 0, \quad (6)$$

where  $[N]$  is the incidence matrix and  $M$  and  $Y$  are vectors of real numbers. Relation  $M = M_0 + [N]Y$  is usually called the state equation.

**Theorem 2** (see [32]). *Let  $(N, M_0)$  be a Petri net with  $N = (P, T, F)$ . There is no emptiable siphon if  $G^{MIP} = |P|$ .*

**Corollary 3.** *Let  $(N, M_0)$  be an ordinary M-net with  $N = (P, T, F)$ . There is no emptiable siphon if  $G^{MIP} = |P|$ .*

The proof of Corollary 3 is given in Appendix B.

**Theorem 4.** *Let  $(N, M_0)$  be an ordinary M-net with  $N = (P, T, F)$ . Then  $(N, M_0)$  is live if  $G^{MIP} = |P|$ .*

The proof of Theorem 4 is given in Appendix B.

### 3. Structure Design of a Petri Net Supervisor

Now, let us go briefly through the processes by using the theory of regions to design a supervisor for a Petri net model.

One first needs to generate its reachability graph. Then, the set of marking/transition separation instances should be found, whose number is in theory exponential with respect to the net size and the initial marking. Finally, for each instance, a monitor should be found by solving a linear programming problem in which the number of constraints is approximately equal to that of nodes in the reachability graph. For such a method, the size of a reachability graph is rather sensitive to the size and the initial marking of a net. These facts make it infeasible for the theory of regions to be applied to real-world problems.

With the purpose of formulating the proposed method even more clearly, we design Algorithm 1 to find a controlled system  $(N^{1c}, M^{1c})$  for  $(N^1, M^1)$  that can be obtained by reallocating the marking of each resource place in  $(N, M_0)$  to be one, with  $N^{1c} = (P^0 \cup P_A \cup P_R \cup P_V, T, F^{1c}, W^{1c})$ .

**Proposition 5.** *Let  $(N^{1c}, M^{1c})$  be the resulting net from adding monitors to an ordinary M-net  $(N^1, M^1)$  by using Algorithm 1. Then  $(N^{1c}, M^{1c})$  is ordinary and live.*

The proof of Proposition 5 is given in Appendix B.

In order to illustrate Algorithm 1, consider a small example from [31]. Figures 1(a) and 1(b) show an ordinary M-net  $(N, M_0)$  and its corresponding reachability graph with eight reachable states, respectively. Now by applying Algorithm 1 to  $(N, M_0)$ , a plant mode  $(N^1, M^1)$  can be obtained, as shown in Figure 1(c). It has the same topology structure as  $(N, M_0)$  in Figure 1(a) but its resource places have a small initial marking with  $M(p_5) = 1$  and  $M(p_6) = 1$ . Its reachability graph is shown in Figure 1(d) with five reachable states. Figure 1(e) shows a controlled system  $(N^{1c}, M^{1c})$  for  $(N^1, M^1)$ , which can be obtained by using the theory of regions [15, 19].

The principal objective of reallocating the marking of each resource place to be one is that it is more tractable by using the theory of regions to design a controlled system for  $(N^1, M^1)$  than that for  $(N, M_0)$ . We can see that the reachable states of  $(N^1, M^1)$  are five that are less than that of  $(N, M_0)$  whose reachable states are eight. One can imagine the heavy computation if the theory of regions is applied to such a net shown in Figure 1(a), with an initial marking  $M_0 = 100p_1 + 80p_2 + 50p_3$  having more than  $8 \times 10^4$  states. Algorithm 1 considers  $(N^1, M^1)$ , as shown in Figure 1(c), which has five reachable markings only. Therefore, it is easier for us to find a supervisor for  $(N^1, M^1)$  than that for  $(N, M_0)$ . Then, a controlled system  $(N^{1c}, M^{1c})$  for  $(N^1, M^1)$  can be obtained by using the theory of regions, as shown in Figure 1(e).

In this section, we propose a method to realize the structure design of a controlled system which can be obtained by Algorithm 1. That is to say, the structure of the supervisors has been found, based on which a deadlock prevention policy will be presented in the next section.

### 4. Deadlock Prevention Policy

This section proposes a deadlock prevention policy that can be carried out through the following Algorithm 2. In the previous section, a controlled system  $(N^{1c}, M^{1c})$  can be

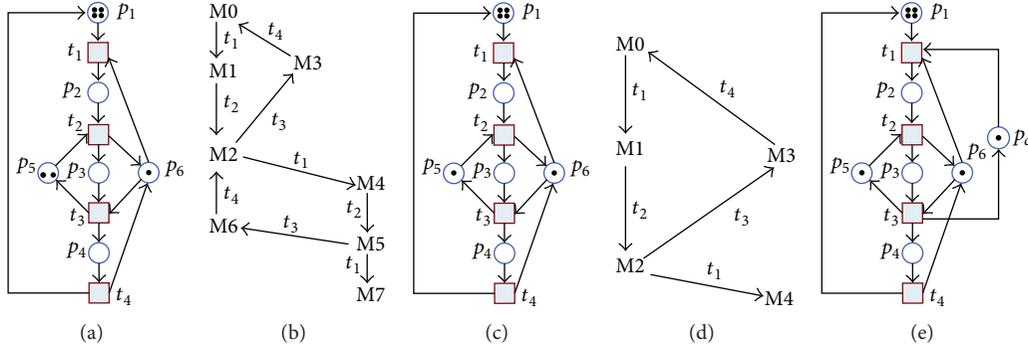


FIGURE 1: (a) A plant model  $(N, M_0)$ , (b) the reachability graph of  $(N, M_0)$ , (c) a modified model  $(N^1, M^1)$ , (d) the reachability graph of  $(N^1, M^1)$ , and (e) a controlled system  $(N^{1c}, M^{1c})$  for  $(N^1, M^1)$ .

obtained for  $(N^1, M^1)$  with  $N^{1c} = (P^0 \cup P_A \cup P_R \cup P_V, T, F^{1c}, W^{1c})$  and  $N^1 = (P^0 \cup P_A \cup P_R, T, F, W)$ . In this section, we can find a controlled system  $(N^c, M^c)$  for  $(N, M_0)$  by restoring the markings of the resource places in  $(N^{1c}, M^{1c})$  to their original ones and then computing a marking for each monitor. That is to say, even if the initial marking of the plant model changes, the structure of the controlled system obtained previously can be reused. This implies that we only need to compute the marking of each monitor in the controlled system when the markings of the resource places change.

Now, let us consider the relationship between the activity places, the monitors, and the resource places. First, we can find that the tokens that will flow into the activity places can not be greater than that of their holding resource places. Similarly, the tokens in the monitor should be less than that of the resource places which the monitor's activity places hold. That can be expressed by the following: for all  $i \in \{1, 2, \dots, n\}$ ,  $v_i \in P_V$ ,  $H(v_i)$  is the set of the activity places controlled by  $v_i$ , and for all  $p \in H(v_i)$ ,  $p$  is called the monitor's activity place.  $P_i = (**H(v_i)) \cap P_R$ , for all  $p \in P_i$ ,  $M^c(v_i) \leq \sum_{p \in P_i} M(p) - 1$ . Then, an upper limit value for each marking with respect to the monitors  $v_1, v_2, \dots$ , and  $v_n$  can be obtained. Finally, we decide a marking for each monitor such that it makes the controlled system live which can be decided by the MIP-based deadlock detection method in [32]. Consequently, Algorithm 2 is designed to formulate the proposed method. Let  $(N^c, M^c)$  denote a controlled system for  $(N, M_0)$ , which has the same net structure as  $(N^{1c}, M^{1c})$  with  $N^c = (P^0 \cup P_A \cup P_R \cup P_V, T, F^{1c}, W^{1c})$ .

**Theorem 6.** Let  $(N^c, M^c)$  be a controlled system for an ordinary M-net  $(N, M_0)$  by using Algorithm 2. Then  $(N^c, M^c)$  is live.

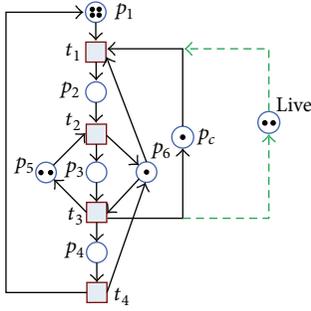
The proof of Theorem 6 is given in Appendix B.

We briefly explain Algorithm 2 as follows. First, a controlled system  $(N^{1c}, M^{1c})$  for  $(N^1, M^1)$  can be obtained by using Algorithm 1 in the previous section. Next, the markings of the resource places in  $(N^{1c}, M^{1c})$  are restored to their original ones. Then, we compute an upper limit value for the marking of each monitor and denote it by

$\vec{\mu} := [M^c(v_1), M^c(v_2), \dots, M^c(v_j), \dots, M^c(v_n)]^T$ ,  $j \in \{1, 2, \dots, n\}$ . If the marking vector  $\vec{\mu}$  makes the controlled system live, we stop the algorithm and the marking vector  $\vec{\mu}$  is the result we want to obtain. If the marking vector  $\vec{\mu}$  makes the controlled system not live, then each marking in  $\vec{\mu}$  decreases by one; repeat this step until it makes the controlled system live and denote it by  $\vec{l} = [M_l^c(v_1), M_l^c(v_2), \dots, M_l^c(v_j), \dots, M_l^c(v_n)]^T$ ,  $j \in \{1, 2, \dots, n\}$ . Then, each marking in  $\vec{l}$  increases by one that can be denoted by  $\vec{h} = [M_h^c(v_1), M_h^c(v_2), \dots, M_h^c(v_j), \dots, M_h^c(v_n)]^T$ ,  $j \in \{1, 2, \dots, n\}$ . That is to say, we obtain a new higher limit value vector  $\vec{h}$  and a lower limit value vector  $\vec{l}$  for the marking of each monitor. Finally, by using Algorithm 2, we decide a marking for each monitor such that for all  $p \in P^0 \cup P_A \cup P_R$ ,  $M^c(p) = M_0(p)$ , and for all  $v \in P_V$ ,  $M^c(v)$  makes the controlled system live which can be decided by the MIP-based deadlock detection method proposed in [32].

For example, a controlled system  $(N^c, M^c)$  can be obtained for the net in Figure 1(a) by utilizing Algorithm 2, as shown in Figure 2. In the previous section, a controlled system  $(N^{1c}, M^{1c})$  has been obtained, as shown in Figure 1(e). Then, we can find a controlled system  $(N^c, M^c)$  for  $(N, M_0)$  by restoring the markings of the resource places in  $(N^{1c}, M^{1c})$  to their original ones with  $M(p_5) = 2$  and  $M(p_6) = 1$ , and then an upper limit value for the marking of the monitor  $p_c$  can be computed. We can find that the activity places controlled by  $p_c$  are  $p_2$  and  $p_3$ . Therefore,  $H(p_c) = \{p_2, p_3\}$ ,  $\{p_5, p_6\} = (**H(p_c)) \cap P_R$ . Then, we can obtain that  $M(p_c) \leq M(p_5) + M(p_6) - 1$  implies that  $M(p_c) \leq 2$ . As shown in Figure 2, the controlled system is live with  $M(p_c) = 2$ , which can be decided by the MIP-based deadlock detection method in [32]. Consequently, the marking  $M(p_c) = 2$  is the one that we want to obtain.

Given a plant model  $(N, M_0)$ , we reallocate the marking of each resource place to be one, obtaining a net model  $(N^1, M^1)$ . By using Algorithm 1, a controlled system  $(N^{1c}, M^{1c})$  for  $(N^1, M^1)$  can be obtained. Then, the markings of the resource places in  $(N^{1c}, M^{1c})$  are restored to their original ones. On the premise of not changing the structure of the controlled system, we compute the marking of each

FIGURE 2: A controlled system  $(N^c, M^c)$  for  $(N, M_0)$ .

monitor by Algorithm 2. That is to say, even if the initial markings of the plant model change, the structure of the controlled system obtained previously can be reused. This implies that we only need to compute the marking of each monitor in the controlled system without changing the supervisor's structure. Figure 3 shows the flowchart of the proposed deadlock control strategy.

## 5. Experimental Studies

In order to show the advantage of the proposed method, this section provides two typical examples that are taken from [31]. The computational results indicate that the proposed deadlock prevention policy is nearly optimal and superior to the one in [31].

An FMS consists of two robots R1 and R2 and three machines M1–M3. Its model is shown in Figure 4(a). It is an ordinary M-net, where  $p_1$  and  $p_{10}$  are idle places,  $p_{11}$ – $p_{15}$  are resource places, and the others are activity places. As shown in Figure 4(b),  $(N^{1c}, M^{1c})$  is the controlled system for the net  $(N^1, M^1)$  with the initial marking of each resource place being one.

To illustrate Algorithm 2, consider  $(N^{1c}, M^{1c})$  shown in Figure 4(b). The activity places  $p_4$  and  $p_9$  are controlled by  $v_1$  and hold the resource places  $p_{12}$  and  $p_{14}$ , respectively. That can be expressed by the following:  $H(v_1) = \{p_4, p_9\}$ ,  $\{p_{12}, p_{14}\} = ({}^{**}H(v_1)) \cap \{p_{11}, p_{12}, p_{13}, p_{14}, p_{15}\}$ . Based on Algorithm 2, the marking of the monitor  $v_1$  should be less than the total markings of the resource places  $p_{12}$  and  $p_{14}$ . That can be expressed by an inequality:

$$M(v_1) \leq (M(p_{12}) + M(p_{14})) - 1. \quad (7)$$

Much the same can be applied to monitors  $v_2$  and  $v_3$ . The inequalities can be obtained as follows:

$$\begin{aligned} M(v_2) &\leq (M(p_{12}) + M(p_{13})) - 1, \\ M(v_3) &\leq (M(p_{13}) + M(p_{14})) - 1. \end{aligned} \quad (8)$$

Now, the net under initial marking  $4p_1 + 4p_{10} + 2p_{11} + 2p_{12} + 2p_{13} + 2p_{14} + 2p_{15}$  is used to demonstrate the proposed

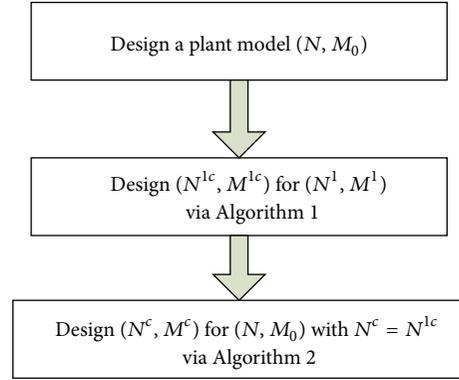


FIGURE 3: Flowchart of the deadlock prevention policy.

method. From (7) and (8), the results can be obtained as follows:

$$\begin{aligned} M(v_1) &\leq 2 + 2 - 1 = 3, \\ M(v_2) &\leq 2 + 2 - 1 = 3, \\ M(v_3) &\leq 2 + 2 - 1 = 3. \end{aligned} \quad (9)$$

Therefore, we can obtain an upper limit value for the marking of each monitor and denote it by  $\bar{\mu} = [M^c(v_1), M^c(v_2), M^c(v_3)]^T = [3, 3, 3]^T$ . Then, we find that the marking vector  $\bar{\mu}$  makes the controlled system live which can be decided by the MIP-based deadlock detection method proposed in [32]. Therefore, the marking vector  $[M^c(v_1), M^c(v_2), M^c(v_3)]^T = [3, 3, 3]^T$  is the result we want to obtain. It can be verified that the controlled model in Figure 4(b) under initial marking  $4p_1 + 4p_{10} + 2p_{11} + 2p_{12} + 2p_{13} + 2p_{14} + 2p_{15}$  with  $M(v_1) = 3$ ,  $M(v_2) = 3$ , and  $M(v_3) = 3$  obtained by the proposed method is live with 1032 reachable states. Compared the proposed method with the one in [31] with 941 states, it can be clearly seen that we have achieved a better result, what we call a near-optimal result.

Compared with the method in [31], the superiority of the proposed policy can be verified. Table 1 shows the permissive behavior of the controlled systems under different initial markings, where the markings of the monitors are decided by Algorithm 2. In this table,  $B_p$  is the number of reachable states of  $(N, M_0)$ ,  $B_L$  represents the number of states that an optimal controlled system for  $(N, M_0)$  has,  $B_c$  indicates the number of states of the controlled system  $(N^c, M^c)$ , and  $B_c/B_L$  implies the optimality degree. In order to make a comparative analysis of the proposed method and the one in [31], let Li's denotes the results in [31]. For economy of space, the detailed computational steps are omitted. From this table, we conclude that the proposed method for this example is near optimal and superior to the one in [31].

The second FMS is shown in Figure 5(a). It has two robots R1 and R2, each of which can hold one product at a time. The cell also contains four machines M1–M4, and each of them can hold one part. Parts enter FMS through two automatic loading buffers I1 and I2 and leave it through two unloading ones O1 and O2. The robots deal with the movements of

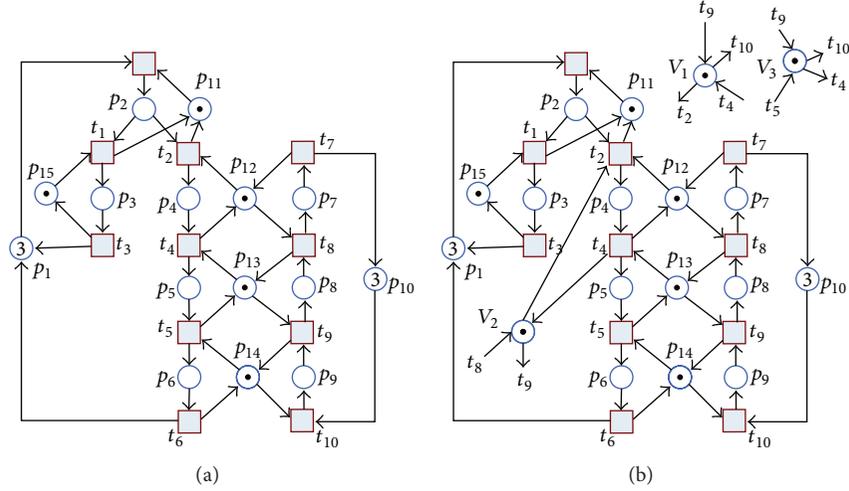


FIGURE 4: (a) An ordinary M-net  $(N, M_0)$  and (b) controlled system  $(N^{1c}, M^{1c})$ .

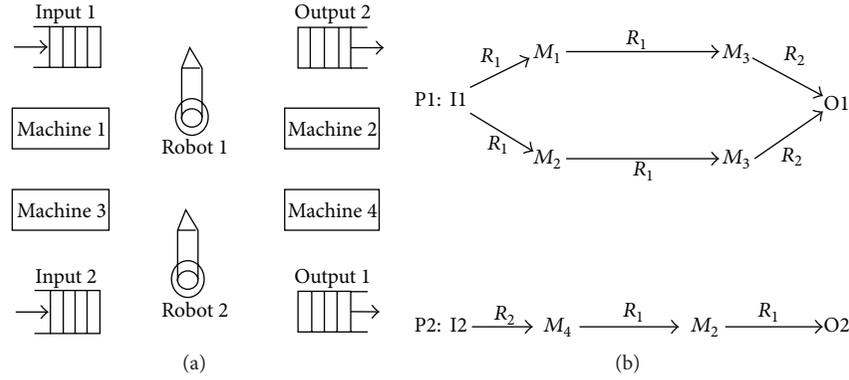


FIGURE 5: (a) Layout of an FMS and (b) routes of part types P1 and P2.

parts. Two part types P1 and P2 are produced. Their respective production routes are shown in Figure 5(b).

Figure 6(a) shows its net model that is an ordinary M-net in which  $P^0 = \{p_1, p_8\}$ ,  $P_R = \{p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\}$ , and the others are activity places. The controlled system of such a plant model is shown in Figure 6(b), which can be obtained by the theory of regions [15].

Consider the model shown in Figure 6(b). The monitor  $v_1$  controls the activity places  $p_3$ ,  $p_{11}$ , and  $p_{12}$ . The set of their corresponding resource places is  $\{p_{15}, p_{18}\}$ . Therefore, the markings of the monitor  $v_1$  should be less than the total markings of the resource places  $p_{15}$  and  $p_{18}$ . That can be expressed by an inequality:

$$M(v_1) \leq (M(p_{15}) + M(p_{18})) - 1. \quad (10)$$

The same situation can be applied to monitors  $v_2$ ,  $v_3$ ,  $v_4$ ,  $v_5$ , and  $v_6$ . The inequalities can be obtained as follows:

$$M(v_2) \leq (M(p_{14}) + M(p_{15}) + M(p_{18})) - 1,$$

$$M(v_3) \leq (M(p_{16}) + M(p_{17}) + M(p_{18}) + M(p_{19})) - 1,$$

$$M(v_4) \leq (M(p_{15}) + M(p_{17}) + M(p_{18}) + M(p_{19})) - 1$$

$$M(v_5) \leq (M(p_{15}) + M(p_{16}) + M(p_{17}) + M(p_{18}) + M(p_{19})) - 1,$$

$$M(v_6) \leq (M(p_{14}) + M(p_{15}) + M(p_{16}) + M(p_{17}) + M(p_{18}) + M(p_{19})) - 1.$$

(11)

By exploiting Algorithm 2, the markings for the monitors can be obtained, as shown in Table 2. For economy of space, the computational steps for the system are not shown in detail. Table 2 shows the performance of the controlled systems under different initial markings. From this table, it is verified that the proposed method for this example is nearly optimal and may even achieve optimality. It is obviously superior to the method in [31].

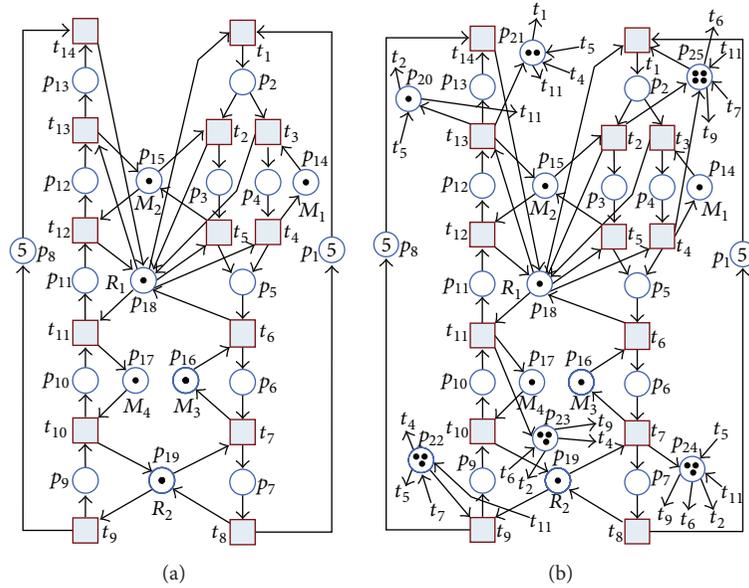


FIGURE 6: (a) Petri net model of an FMS and (b) structure of the controlled system.

### 6. Comparison of Computational Efficiency

The deadlock prevention method proposed in [31] needs to calculate all SMS in the controlled system and infer algebraic inequalities. An improved method of avoiding this problem is presented in this paper, which is simple and practicable. To illustrate the application of the proposed method, two FMS examples are used in this paper. A comparison between the proposed method and the one in [31] is shown in Tables 1 and 2. The superiority of the proposed one is obvious. For a class of FMS considering resource allocation, this paper proposes a deadlock prevention policy by resources reallocation and supervisor reconfiguration, which can make a good tradeoff between optimality and computational tractability for a class of ordinary Petri nets.

### 7. Conclusion

The deadlock prevention policy is a static strategy that imposes restrictions on the interactions among resources and processes such that resource requests that may lead to deadlocks are prevented. Behavioral permissiveness is very important in designing a liveness-enforcing supervisor for a system to be controlled. An optimal liveness-enforcing supervisor can lead to high utilization of system resources. This paper proposes a deadlock prevention policy by resources reallocation and supervisor reconfiguration. Given a plant model, we first reallocate the marking of each resource place to be one, and then find a controlled system by using Algorithm 1. Next, the markings of the resource places in the controlled system are restored to their original ones. Without changing the structure of the controlled system, we compute the markings of the monitors. Finally, we decide a marking for each monitor such that it makes the controlled system live which can be realized by Algorithms 2 proposed

in this paper. For a fixed net structure with different initial marking, the theory of regions is used once only. That is to say, the supervisory control system can be reconfigured rapidly in response to the changes in the initial markings of the plant model. Two FMS examples are used to illustrate the application of the proposed method and show its superior efficiency.

However, the proposed method suffers from the computational complexity problem due to the theory of regions. In theory, it suffers from the exponential complexity. Future efforts will be made to a near-optimal supervisor with low computational costs. In addition, the proposed method is applicable to ordinary M-nets only. Therefore, our future work will extend this method to more general classes of Petri nets, for example, the ones in [34, 35].

### Appendices

#### A. An Example for Definition 1

As the net shown in Figure 7, it is an M-net, where  $p_1$  is an idle process place,  $p_2, p_3,$  and  $p_4$  are activity places, and  $p_5$  and  $p_6$  are resource places. It is quasi-live, bounded, and conservative. It is live if no siphon is uncontrolled.

#### B. Proofs for Corollary 3, Theorems 4, 6, and Proposition 5

The proof of Corollary 3.

*Proof.* It follows immediately from the definition of an ordinary M-net that it is a class of ordinary Petri nets. According to Theorem 2, the result is true.  $\square$

The proof of Theorem 4.

TABLE 1: Behavior permissiveness of the proposed deadlock prevention policy.

$P_1, P_{10}, P_{11}-P_{15}$	$v_1, v_2, v_3/Li$	$B_p$	$B_L$	$B_c/Li's$	$B_c/B_L/Li's$
(1) 3, 3, 1, 1, 1, 1, 1	$[1, 1, 1]^T/[1, 1, 1]^T$	73	54	54/54	100%/100%
(2) 4, 4, 2, 2, 2, 2, 2	$[3, 3, 3]^T/[3, 3, 2]^T$	1093	1047	1032/941	94.419%/89.876%
(3) 5, 5, 3, 3, 3, 3, 3	$[5, 5, 5]^T/[5, 5, 3]^T$	5767	5705	5680/5151	99.562%/90.290%
(4) 6, 6, 4, 4, 4, 4, 4	$[7, 7, 7]^T/[7, 7, 3]^T$	20324	20263	20234/18517	99.857%/91.383%
(5) 7, 7, 5, 5, 5, 5, 5	$[9, 9, 9]^T/[9, 9, 5]^T$	57450	57390	57360/52995	99.948%/92.342%
(6) 8, 8, 6, 6, 6, 6, 6	$[11, 11, 11]^T/[11, 11, 6]^T$	140703	140643	140613/13100	99.977%/93.144%
(7) 9, 9, 7, 7, 7, 7, 7	$[13, 13, 13]^T/[13, 13, 7]^T$	310783	310723	310693/292363	99.990%/93.770%
(8) 10, 10, 8, 8, 8, 8, 8	$[15, 15, 15]^T/[15, 15, 8]^T$	634173	634113	634083/597853	99.995%/94.282%
(9) 11, 11, 9, 9, 9, 9, 9	$[17, 17, 17]^T/[17, 17, 9]^T$	1214679	1214619	1214589/1150189	99.998%/94.695%
(10) 12, 12, 10, 10, 10, 10, 10	$[19, 19, 19]^T/[19, 19, 10]^T$	2208445	2208385	2208355/2098887	99.990%/95.042%

TABLE 2: Behavior permissiveness of the proposed deadlock prevention policy.

$P_1, P_8, P_{14}-P_{19}$	$v_1, v_2, v_3, v_4, v_5, v_6$	$B_p$	$B_L$	$B_c/Li's$	$B_c/B_L/Li's$
(1) 3, 3, 1, 3, 1, 2, 3, 2	$[5, 6, 7, 9, 10, 11]^T$	2946	2945	2945/2842	100%/96.500%
(2) 3, 4, 1, 4, 1, 1, 3, 3	$[6, 7, 7, 10, 11, 12]^T$	5235	5234	5234/4730	100%/90.388%
(3) 4, 4, 1, 5, 1, 1, 2, 4	$[6, 7, 7, 12, 13, 14]^T$	6877	6868	6868/6861	100%/99.898%
(4) 4, 5, 1, 5, 2, 2, 4, 5	$[8, 9, 12, 15, 17, 18]^T$	31759	31758	37758/29129	100%/91.722%
(5) 5, 5, 1, 6, 1, 1, 3, 5	$[8, 9, 9, 14, 15, 16]^T$	28243	28233	28233/28177	100%/99.802%
(6) 5, 6, 1, 7, 1, 6, 3, 1	$[9, 10, 10, 16, 17, 18]^T$	24448	24438	24438/24384	100%/99.779%
(7) 6, 6, 3, 7, 3, 3, 5, 7	$[11, 14, 17, 21, 24, 27]^T$	298725	298724	298724/290187	100%/97.142%

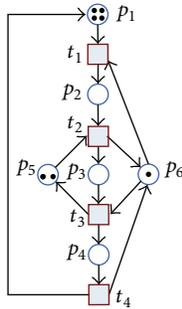


FIGURE 7: An M-net model.

*Proof.* Let  $(N, M_0)$  be an ordinary M-net. There is no empty siphon if  $G^{\text{MIP}} = |P|$  from Corollary 3. By the definition of M-nets, if  $(N, M_0)$  is ordinary and no empty siphon, that is, uncontrolled siphon in  $(N, M_0)$ ,  $(N, M_0)$  is live.  $\square$

The proof of Proposition 5.

*Proof.* It follows immediately from the theory of regions that  $(N^{1c}, M^{1c})$  is live.  $(N^{1c}, M^{1c})$  is excluded by Algorithm 1 if it is a generalized net. Therefore, the result is true.  $\square$

The proof of Theorem 6.

*Proof.* According to Proposition 5,  $(N^{1c}, M^{1c})$  is ordinary and live. The structure of the controlled system  $(N^c, M^c)$  is the same as that of the  $(N^{1c}, M^{1c})$ . That is to say,  $(N^c, M^c)$  is an ordinary controlled system as well as an M-net according to

Definition 1. Then, from Theorem 4,  $(N^c, M^c)$  is an ordinary M-net,  $(N^c, M^c)$  is live if  $G^{\text{MIP}} = |P|$ . Therefore, the result is true.  $\square$

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants no. 61074035 and 61100056, the Fundamental Research Funds for the Central Universities under Grant no. 72103326, the Zhejiang Provincial Natural Science Foundation of China under Grant no. LY12F03020, the Zhejiang Provincial Education Department Foundation under Grant no. Y201018216, and the Opening Project of Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing, under Grant no. MCCSE2012A05.

## References

- [1] T. Murata, "Petri nets: properties, analysis and application," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
- [2] J. Ezpeleta, J. M. 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.
- [3] Z. W. Li and M. C. Zhou, "Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Transactions on Industrial Informatics*, vol. 2, no. 4, pp. 313–325, 2006.

- [4] Z. W. Li, N. Q. Wu, and M. C. Zhou, "Deadlock control of automated manufacturing systems based on Petri nets—a literature review," *IEEE Transactions on Systems, Man, and Cybernetics C*, vol. 42, no. 4, pp. 437–462, 2012.
- [5] S. G. Wang, C. Y. Wang, and Y. P. Yu, "Comments on 'siphon-based deadlock prevention policy for flexible manufacturing systems,'" *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 41, no. 2, pp. 338–340, 2011.
- [6] S. G. Wang, C. Y. Wang, and M. C. Zhou, "Controllability conditions of resultant siphons in a class of Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 42, no. 5, pp. 1206–1215, 2012.
- [7] B. Hruz and M. C. Zhou, *Modeling and Control of Discrete Event Dynamic Systems*, Springer, London, UK, 2007.
- [8] M. D. Jeng and X. L. Xie, "Deadlock detection and prevention of automated manufacturing systems using Petri nets and siphons," in *Deadlock Resolution in Computer-Integrated Systems*, M. C. Zhou and M. P. Fanti, Eds., pp. 233–281, Marcel-Dekker, New York, NY, USA, 2005.
- [9] T. K. Kumaran, W. Chang, H. Cho, and A. Wysk, "A structured approach to deadlock detection, avoidance and resolution in flexible manufacturing systems," *International Journal of Production Research*, vol. 32, no. 10, pp. 2361–2379, 1994.
- [10] M. Silva and J. M. Colom, "On the computation of structural synchronic invariants in P/T nets," in *Advances in Petri Nets 1988*, Springer, New York, NY, USA, 1988.
- [11] S. G. Wang, C. Y. Wang, M. C. Zhou, and Z. W. Li, "A method to compute strict minimal siphons in a class of Petri nets based on loop resource subsets," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 42, no. 1, pp. 226–237, 2012.
- [12] Y. F. Chen and Z. W. Li, "On structural minimality of optimal supervisors for flexible manufacturing systems," *Automatica*, vol. 48, no. 10, pp. 2647–2656, 2012.
- [13] Y. F. Chen and Z. W. Li, *Optimal Supervisory Control of Automated Manufacturing Systems*, CRC Press, Taylor & Francis Group, New York, NY, USA, 2013.
- [14] Z. W. Li and M. C. Zhou, *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*, Springer, London, UK, 2009.
- [15] 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, no. 3, pp. 192–208, 2002.
- [16] M. Uzam and M. C. Zhou, "An improved iterative synthesis method for liveness enforcing supervisors of flexible manufacturing systems," *International Journal of Production Research*, vol. 44, no. 10, pp. 1987–2030, 2006.
- [17] P. K. Mishra, "Lower and upper bounds of shortest paths in reachability graphs," *International Journal of Mathematics and Mathematical Sciences*, no. 57, pp. 3023–3036, 2004.
- [18] M. Uzam and M. C. Zhou, "An iterative synthesis approach to Petri net-based deadlock prevention policy for flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 37, no. 3, pp. 362–371, 2007.
- [19] 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. 137–142, 2003.
- [20] Y. F. Chen, Z. W. 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.
- [21] B. Barzegar, H. Motameni, and H. Bozorgi, "Solving flexible job-shop scheduling problem using gravitational search algorithm and colored Petri net," *Journal of Applied Mathematics*, vol. 2012, Article ID 651310, 20 pages, 2012.
- [22] J. Ezpeleta, F. Garcia-Valls, and J. M. Colom, "A class of well structured Petri nets for flexible manufacturing systems," in *Proceedings of the 19th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency*, J. Desel and M. Silva, Eds., vol. 1420 of *Lecture Notes in Computer Science*, pp. 64–83, 1998.
- [23] Y. S. Huang, M. D. Jeng, X. L. Xie, and S. L. Chung, "A deadlock prevention policy for flexible manufacturing systems using siphons," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '01)*, pp. 541–546, May 2001.
- [24] Y. S. Huang, M. D. Jeng, X. L. Xie, and D. H. Chung, "Siphon-based deadlock prevention policy for flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 36, no. 6, pp. 1248–1256, 2006.
- [25] Z. W. Li and M. C. Zhou, "Elementary siphons of petri nets and their application to deadlock prevention in flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 34, no. 1, pp. 38–51, 2004.
- [26] F. Tricas, F. Garcia-Valls, J. M. Colom, and J. Ezpeleta, "An iterative method for deadlock prevention in FMSs," in *Proceedings of the 5th Workshop on Discrete Event Systems*, R. Boel and G. Stremersch, Eds., pp. 139–148, 2000.
- [27] A. R. Wang, Z. W. Li, J. Y. Jia, and M. C. Zhou, "An effective algorithm to find elementary siphons in a class of Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 39, no. 4, pp. 912–923, 2009.
- [28] K. Y. Xing, M. C. Zhou, F. Wang, H. X. Liu, and F. Tian, "Resource-transition circuits and siphons for deadlock control of automated manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 41, no. 1, pp. 74–84, 2011.
- [29] L. Piroddi, R. Cordone, and I. Fumagalli, "Selective siphon control for deadlock prevention in Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 38, no. 6, pp. 1337–1348, 2008.
- [30] L. Piroddi, R. Cordone, and I. Fumagalli, "Combined siphon and marking generation for deadlock prevention in Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 39, no. 3, pp. 650–661, 2009.
- [31] Z. W. Li, G. Y. Liu, H. M. Hanisch, and M. C. Zhou, "Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 42, no. 1, pp. 178–191, 2012.
- [32] F. Chu and X. L. 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.
- [33] K. Barkaoui and J. F. Pradat-Peyre, "On liveness and controlled siphons in Petri nets," in *Proceedings of the 17th International Conference on Applications and Theory of Petri Nets*, vol. 1091 of *Lecture Notes in Computer Science*, pp. 57–72, 1996.
- [34] H. S. Hu, M. C. Zhou, and Z. W. Li, "Supervisor design to enforce production ratio and absence of deadlock in automated manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 41, no. 2, pp. 201–212, 2010.
- [35] D. Liu, Z. W. Li, and M. C. Zhou, "Liveness of an extended  $S^3PR$ ," *Automatica*, vol. 46, no. 6, pp. 1008–1018, 2010.