Research Article

Comparison and Evaluation of Deadlock Prevention Methods for Different Size Automated Manufacturing Systems

**Emad Abouel Nasr,^{1,2} Abdulaziz M. El-Tamimi,¹
Abdulrahman Al-Ahmari,¹ and Husam Kaid¹**

¹Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia

²Mechanical Engineering Department, Helwan University, Cairo 11792, Egypt

Correspondence should be addressed to Emad Abouel Nasr; eabdelghany@ksu.edu.sa

Received 13 January 2015; Revised 26 June 2015; Accepted 3 August 2015

Academic Editor: Yan-Jun Liu

Copyright © 2015 Emad Abouel Nasr 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.

In automated manufacturing systems (AMSs), deadlocks problems can arise due to limited shared resources. Petri nets are an effective tool to prevent deadlocks in AMSs. In this paper, a simulation based on existing deadlock prevention policies and different Petri net models are considered to explore whether a permissive liveness-enforcing Petri net supervisor can provide better time performance. The work of simulation is implemented as follows. (1) Assign the time to the controlled Petri net models, which leads to timed Petri nets. (2) Build the Petri net model using MATLAB software. (3) Run and simulate the model, and simulation results are analyzed to determine which existing policies are suitable for different systems. Siphons and iterative methods are used for deadlocks prevention. Finally, the computational results show that the selected deadlock policies may not imply high resource utilization and plant productivity, which have been shown theoretically in previous publications. However, for all selected AMSs, the iterative methods always lead to structurally and computationally complex liveness-enforcing net supervisors compared to the siphons methods. Moreover, they can provide better behavioral permissiveness than siphons methods for small systems. For large systems, a strict minimal siphon method leads to better behavioral permissiveness than the other methods.

1. Introduction

Effectively designing and operating an automated manufacturing system can be of assistance to manufacturers to adapt the variety in the market in order to maintain and confirm competitiveness. Different types of parts in AMS enter the system at discrete points of time and are produced concurrently; these parts have to share some common resources, such as robots, machine tools, buffers, automated guided vehicles, and fixtures. The deadlocks can occur in an AMS during its operation by shared resources competition, which are highly unwanted phenomena that often cause low use of some expensive and critical resources and long downtime and can lead to calamitous results in automated manufacturing systems. Therefore, it is required to develop a successful AMS control policy to prevent deadlocks occurrence in AMS.

There are several mathematical tools to solve deadlock problems in AMSs including automata, graph theory, and

Petri nets. Recently, Petri nets have been known as one of the most robust mathematical tools for modelling, analyzing, and controlling deadlocks in resource allocation systems including AMS [1]. To overcome the deadlocks in AMS, there are methods which have been derived from a Petri net tool either to forbid the deadlock occurrences by preventing some necessary condition or to detect and resolve a deadlock when it occurs. In general, these methods can be classified into three strategies: deadlock detection and recovery, deadlock avoidance, and deadlock prevention [1, 2].

The researchers proposed three criteria for designing and evaluating a supervisor for a manufacturing system to be controlled. These criteria include behavioral permissiveness, which leads to increase in the resources utilization of system, structural complexity that leads to design supervisor with a few number of monitors to decrease the software and hardware costs, and computational complexity that means a deadlock control policy can be implemented in huge

systems [1]. Therefore, the objectives of many researchers are to develop deadlock prevention policies which can provide liveness-enforcing supervisors with maximal permissiveness, a structural complexity, and computational complexity [1].

From previous studies, there are mainly two techniques of analysis for deadlock prevention in AMSs using Petri net: structural analysis [3, 4] and reachability graph analysis [5–7]. Oftentimes structural analysis is applied for structural objects of Petri nets, such as resource transition circuits and siphons. In this case, control steps are simple, and each empty minimal siphon requires a monitor to be added to prevent itself from being not emptied, but the disadvantages of this technique are the fact that the resulting controlled system can be suboptimal and linearly the number of control places is dependent on the size of a net [8].

The reachability graph analysis suffers from a state explosion problem because it requires listing of all or a part of reachable markings. The reachability graph of a Petri net has been classified into two zones: live zone (LZ) and deadlock zone (DZ). In DZ, there is first-met bad markings (FBMs) which are extracted from it; the FBMs are representing the very first entry from LZ to DZ. In this case, a control place is designed and added to forbid the FBM from being reached. This process requires iterations to forbid all FBMs and single out each FBM. The process is terminated if the resulting net is a live [9]. There are several methods and algorithms used for deadlocks prevention; some of them are siphon control methods, a theory of region, and iterative methods [4, 6, 9–11].

Most of the researchers have used structural analysis and reachability graph analysis to design deadlocks prevention policies for different scales of AMSs under three criteria that are maximal permissiveness, structural complexity, and computational complexity. The comparison between these policies has been done according to the three aforementioned criteria. It is taken for granted in the literature that a maximally permissive supervisor in the logical level usually leads to the better time performance in an AMS such as productivity and resource utilization, which looks intuitive, as an optimal supervisor will permit more permissive behavior. However, there is a shortage in the quantitative evaluation of time performance of these policies in terms of maximal permissiveness and productivity. Simulation is a powerful tool for performance analysis of complex manufacturing systems.

The contribution of this paper is using a simulation tool based on existing deadlock prevention policies and different types of Petri net models to explore whether a permissive liveness-enforcing Petri net supervisor can provide better time performance (maximal permissiveness).

The work of simulation is implemented in the following steps. (1) Assign the time to the controlled Petri net models, which leads to timed Petri nets. (2) Build the Petri net model using MATLAB software (Petri Net Toolbox (PN Toolbox)). (3) Run and simulate the model to obtain the system time performance. (4) Finally, the simulation results will be analyzed to determine which existing deadlock prevention policies are convenient for different types of manufacturing systems. Different control methods (siphon control methods

and iterative methods) are used for deadlocks prevention. Moreover, this paper aims to evaluate the performance of the selected methods such as utilization of resources, plant throughput, the number of monitors, the number of arcs, and the number of reachable states. This paper focuses on the performance evaluation of automated manufacturing systems by using Petri nets under different deadlock prevention policies. The considered Petri net models belong to M -nets [12] that are a general class of manufacturing-oriented net models in the literature. In an M -net, there are a number of structural properties thanks to its definition [12].

Including this introductory section, the paper is organized as follows. Section 2 provides a literature review of previous researchers works related to deadlock prevention. The basics of Petri nets are discussed in Section 3. The deadlock prevention methods and policies are presented in Section 4. Sections 5 and 6 show the analysis of the selected methods for four different automated manufacturing systems. Finally, Section 7 presents the conclusion and future work.

2. Literature Review

Deadlock prevention policies have been widely achieved for AMS and led to a huge number of results [1, 11]. In this section, deadlock prevention strategies are reviewed by using Petri nets and developed based on different techniques such as structural analysis and reachability graph analysis.

2.1. Structural-Analysis Methods. Ezpeleta et al. [13] used a class of Petri nets, called System of Simple Sequential Processes with Resources (S^3PR) and proposed an algorithm for resource allocation in flexible manufacturing systems (FMSs). The proposed algorithm added new places to the net to impose certain restrictions that forbid the presence of unmarked siphons. The works of Huang et al. [14] and Huang et al. [15] present a new deadlock prevention policy for the class of Petri nets, where deadlocks are related to unmarked siphons. Two types of control places are added to the original model for flexible manufacturing system called ordinary control place and weighted control place to prevent siphons from being unmarked.

The study by Li and Zhou [11] presents a Petri net model which consists of 26 places, 20 transitions, and 18 strict minimal siphons. In [16], elementary siphons concept is introduced to design a liveness-enforcing Petri net supervisor for the same Petri net model, and six elementary siphons are obtained; therefore, only six monitors are added to control the 18 strict minimal siphons. Nevertheless, the proposed algorithm by Ezpeleta et al. [13] adds monitors for all 18 strict minimal siphons.

A new methodology is proposed by Huang [17] to synthesize supervisors for resource allocation in FMSs; the class of Petri net, namely, S^3PR , is considered, where deadlocks are related to unmarked minimal siphons; all minimal siphons should be controlled by adding control places. In this study, the number of control places is reduced by using the concept of the elementary siphon. Based on elementary siphons and P -invariants concepts in Petri nets, a deadlock prevention

algorithm is introduced by Li and Wei for a specific class of Petri nets which can properly model several FMSs [16], siphons in a Petri net model are classified into dependent and elementary siphons, and control places are added for all elementary siphons, so that the siphons are invariant-controlled. Therefore, the controllability of dependent siphons is guaranteed by adjusting the control requirements of its related elementary siphons. A novel deadlock prevention algorithm is presented by Chen et al. [1], and the concept of siphon extraction is used to compute unmarked siphons for the Petri net model. First, a siphon extraction algorithm obtains a maximal unmarked siphon, divides the places in it, and determines a necessary siphon from the divided places; this process is carried out for all unmarked siphons. Then, the proposed deadlock prevention algorithm designs a convenient monitor for each necessary siphon to be marked until the controlled Petri net model is a live.

Most of the deadlock control algorithms have been developed for reliable automated manufacturing systems (AMSs). Liu et al. [18] proposed a variety of deadlock control algorithms for AMSs with unreliable resources, monitors and recovery subnets are designed for strict minimal siphons (emptied siphon) and unreliable resources, respectively, and two types of arcs which are normal and inhibitor are used to connect monitors with recovery subnets. In order to obtain a supervisor with structural complexity, elementary siphons extracted from all strict minimal siphons are obviously controlled. The problems of structural complexity, behavioral permissiveness, and computational complexity are considered in the study of Wang et al. [19] to design liveness-enforcing supervisors for a class of Petri nets, namely, S^3PR , without ξ -resources. They used two steps to obtain a supervisor for the model. In the first step, an algorithm is proposed to extract unmarked strict minimal siphons from a given unmarked siphon by using loop resource subsets. In the second step, a siphon-based deadlock prevention algorithm is introduced to provide a maximally permissive liveness-enforcing supervisor with small numbers and no weighted monitors.

2.2. Reachability Graph-Based Approaches. Viswanadham et al. [20] presented static resource allocation methods for eliminating deadlocks; in this study, a reachability graph of the Petri net (PN) model is used to reach static resource allocation method. A deadlock prevention algorithm is implemented for a small size manufacturing system consisting of a machine and an automated guided vehicle (AGV). The authors observed that the proposed algorithm can be applied effectively only for small size manufacturing systems. The forbidden state problem of Petri nets (PN) with uncontrollable transitions and liveness requirement is addressed by Ghaffari et al. [5]. An approach is presented to compute a maximally permissive Petri net controller that consists of two steps. In the first step, an ordinary Petri net is used to build the system model. In the second step, control places are designed and added to the original model by using a theory of regions method. The work of Uzam [7, 9] proposes and improves method based on a theory of regions to design an optimal

Petri net supervisor. The size of the reachability graph of Petri net model is a major problem to apply the deadlock prevention policy to very large Petri net model. Therefore, they proposed a reduction algorithm to simplify very large Petri nets models in order to make necessary computation easily. Based on the theory of regions, Uzam and Zhou [6] developed an iterative deadlock prevention policy for FMS. In their study, the reachability graph of a Petri net model is divided into two parts: deadlock-free zone (live zone, LZ) and the deadlock zone (DZ). The live zone is developed since the maximal strongly connected component contains the initial marking. The deadlock zone contains markings from which the initial marking is unreachable. FBM is defined as a marking in the deadlock zone. The deadlocks can be eliminated by forbidding the firing of the enabled transitions at FBM. In their work, the presented approach has two problems. First, an optimal supervisor cannot be guaranteed in general even if such an optimal supervisor exists. Second, at each iteration full reachability graph computation is required to check whether the markings in the deadlock zone are reachable. This approach is easy to use and simple if the reachable graph of a system is small but cannot ensure the supervisor behavioral optimality. The redundant monitors in Petri net supervisor may exist when the supervisor is designed by an iterative siphon control approach. Therefore, Uzam et al. [21] introduced an approach to identify and eliminate the redundant monitors by computing the reachability graph of a controlled Petri net model. If the controlled Petri net model does not lose liveness when removing redundant monitors, then the redundant monitors can be eliminated from the supervisor. The reachability graph and liveness check are the major problems in designing liveness-enforcing supervisor. To overcome a full state enumeration, Li and Hu [22] proposed two approaches to eliminating control places from a Petri net supervisor. The first approach is based on the concept of implicit places [23, 24]. The implicitity of a monitor is computed by solving a linear programming problem (LPP) that can be implemented in polynomial time. The second approach is extracted from the Mixed Integer Programming- (MIP-) based deadlock detection approach. If the optimal solution to an MIP problem does not lose optimality when removing control place, then the control place is implicit and may provide permissive behavior while liveness is guaranteed. Chen et al. [1] developed a novel and computationally effective approach to design optimal monitors and an iterative approach that can compute the reachability graph to get an optimal supervisor of the flexible manufacturing system, which is demonstrated by a set of control places. The vector covering approach is used to compute the minimal sets of FBMs and legal markings. Iteratively, an FBM is selected from the minimal set of FBMs, an integer linear programming problem (ILPP) is used to design a place invariant (PI) that prevents the FBM from being reached, and all minimal sets of legal markings are not forbidden. The process is terminated if all FBMs cannot be reached and the resulting net is a live. This study is improved in its computational efficiency in [25], and the objective function of the ILPP is maximizing the number of FBMs that are prohibited by the PI; in other words, a control place is

designed such that as many FBMs as possible are forbidden and all minimal sets of legal markings are not prohibited. If the ILPP has no solution, it means that there is no maximally permissive Petri net supervisor for the Petri net model.

3. Basics of Petri Nets

Petri nets are a mathematical and graphical tool suitable for several systems. It is a major tool for studying and describing manufacturing system operations that are characterized as being asynchronous, concurrent, parallel, and distributed.

A Petri net or place/transition net can be defined as a four-tuple $N = (P, T, F, W)$, where P is a finite nonempty set of places, T is a finite nonempty set of transitions, P and T are disjoint, $P \cup T$ are called nodes with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$, and P and T are represented by circle and bar, respectively. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation or directed arcs of the net that can be represented by arcs with arrows from places (transitions) to transitions (places). $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ is a mapping that assigns a weight to an arc: if $(a, b) \in F$, $W(a, b) > 0$; otherwise, $W(a, b) = 0$, where $a, b \in P \cup T$ and $\mathbb{N} = \{0, 1, 2, \dots\}$. N is called an ordinary net, if $W(a, b) = 1, \forall (a, b) \in F$. Consider a node $a \in P \cup T$, and $\cdot a = \{b \in P \cup T \mid (b, a) \in F\}$ is a preset of a . Postset of a is $a^\bullet = \{b \in P \cup T \mid (a, b) \in F\}$. A marking M of N is a mapping from $P \rightarrow \mathbb{N}$. $M(p)$ denotes the number of tokens in place p . Usually, markings and vectors can be described using formal sum. Therefore, vector M is denoted by $\sum_{p \in P} M(p)p$. For example, a marking that puts two tokens in place p_2 , five tokens in place p_4 , and no tokens in other places in a net is denoted by $2p_2 + 5p_4$. (N, M_0) is a net system or marked net and is denoted by $N = (P, T, F, W, M_0)$, where M_0 is an initial marking of N . For a Petri net modeling M_0 represents the different raw parts that are to be synchronously processed in the system, and the state of resources, such as machines and robots. A net is self-loop free or pure if for all $a, b \in P \cup T$, $W(a, b) > 0$ and $W(b, a) = 0$. Incidence matrix $[N]$ in a net N is an integer matrix that consists of $|P|$ rows and $|T|$ columns with $[N](p, t) = W(t, p) - W(p, t)$. The enabled rule for each $t \in T$ at marking M if $\forall p \in \cdot t$, $M(p) \geq W(p, t)$. As a result, enabled rule is denoted by $M[t]$. When a transition t fires, it removes $W(p, t)$ tokens from each place $p \in \cdot t$ and deposits $W(t, p)$ tokens in each place $p \in t^\bullet$. Therefore, it yields a new marking M' , denoted by $M[t]M'$, where $M'(p) = M(p) - W(p, t) + W(t, p)$. M' is reachable from M if there exists a fireable finite transition sequence $\delta = \{t_1, t_2, t_3, \dots, t_n\}$, and markings $M_1, M_2, M_3, \dots$, and M_{n-1} so that $M[t_1]M_1[t_2]M_2[t_3]M_3, \dots, M_{n-1}[t_n]M'$, which can be denoted by $M[\delta]M'$ and satisfied the state equation $M' = M + [N]\vec{\delta}$, where $\vec{\delta} : T \rightarrow \mathbb{N}$ is a mapping t in T to the number of occurrences of t in δ , called a Parikh vector or a firing count vector. Obviously, $M_0[\cdot]$ is the set of all reachable markings of a net N from initial marking M_0 and is denoted by $R(N, M_0)$. The reachable markings $R(N, M_0)$ can be involved in the enumeration of all reachable markings of a net N and graphically expressed by a reachability tree. The reachability tree is denoted by $G(N, M_0)$, which is a directed graph and consists of nodes and arcs, nodes are markings in

the reachable markings $R(N, M_0)$, and arcs are labeled by the transitions of a net N . For instance, an arc from M_2 to M_3 is labeled by t_1 if $M_2[t_1]M_3$. A net N with initial marking M_0 is to be k -bounded if, for all $M \in R(N, M_0)$, $M(p) \leq k$ ($k = \{1, 2, 3, \dots\}$). In other words, for each reachable marking " M " from the initial marking M_0 , the number of tokens in each place " p " never exceeds a finite number " k ." A net is called safe if all its places are safe, and the number of tokens in each place " p " does not exceed one. In other words, a net is k -safe if it is 1-bounded. A Petri net with (N, M_0) is a live if all its transitions are a live at M_0 . A transition $t \in T$ is a live if, for all $M \in R(N, M_0)$, $\exists M' \in R(N, M)$, there is a firing sequence $M'[t]$. A transition is dead at M_0 if $\nexists t \in T, M_0[t]$. P -vector (place vector) and T -vector (transition vector) are column vectors. The former $I : P \rightarrow Z$ is catalogued by P and called a P -invariant or place invariant if $I \neq 0$ and $I^T \cdot [N] = 0^T$. The latter $J : T \rightarrow Z$ is catalogued by T and called a T -invariant or transition invariant if $J \neq 0$ and $[N] \cdot J = 0$, where Z is the integers set. If each element of I is nonnegative the P -invariant I is called a P -semiflow or place semiflow. Suppose that I is a place invariant of a Petri net with (N, M_0) and M is a reachable marking from the initial marking M_0 . Then, $I^T M = I^T M_0$. Let $\|I\| = \{p \mid I(p) \neq 0\}$ be a support of place invariant I and it can be classified into three parts. Firstly, $\|I\|^+$ is a positive support of place invariant I , and $\|I\|^+ = \{p \mid I(p) > 0\}$. Secondly, $\|I\|^-$ is a negative support of place invariant I , and $\|I\|^- = \{p \mid I(p) < 0\}$. Finally, I is a minimal place invariant if $\|I\|$ is not a superset of the support of any other one and its components are mutually prime. Let l_i 's be the coefficients of place invariant I if, for all $p_i \in P$, $l_i = I(p_i)$.

4. Deadlock Prevention Methods and Policies

4.1. Siphon Control Methods. This section presents two methods that are motivated by Ezpeleta et al. [13] and Li and Zhou [11]. These methods are called a strict minimal siphons control method and an elementary siphons control method, respectively.

4.1.1. Strict Minimal Siphons (SMSs) Control Method. A siphon in a Petri net is a set of places $S = \{p_1, \dots, p_k\}$. The very important property of siphon is that, for a given marking, the siphon is unmarked.

Definition 1. Let N be a net. Any set $S \subseteq P$, $S \neq \emptyset$ with $\cdot S \subseteq S^\bullet$ is called a siphon. If a siphon does not properly contain another siphon is called a minimal siphon. A minimal siphon S is called a strict if $\cdot S \subsetneq S^\bullet$.

Siphons play a significant role in the liveness analysis of a Petri net, especially in ordinary ones. If a siphon S in a net has no tokens, then no transitions in $S^\bullet$ are enabled and all the transitions connected to S cannot be fired; as a result, the transitions are dead leading to the loss of liveness of the Petri net. In [13], a monitor is added to each strict minimal siphon in order to achieve the liveness of the system. The proposed approach is simple and ensures a success. Nevertheless, it leads to a complex Petri net model than the originally model because the added places are equal to the strict minimal

siphons in the target net model and the added arcs are larger than that of places added.

The SMS control based on complementary sets of siphons $[S]$ is used to design the control places. A complementary set of a siphon S is a set of operation places which are the holders of the resources in S but do not belong to S . In addition, $[S] \cup S$ is the support of a place invariant, which means the operation places in $[S]$ will compete for the resources in S with the operation places belonging to S . S will be unmarked when all tokens in S flow into $[S]$. The following notation will be used in the establishment of the control policy.

- (1) S denotes the set of strict minimal siphons which does not contain the support of any place semiflow (i.e., siphons that can be emptied).
- (2) Given the fact that S is a siphon we have $S = S_A \cup S_R$, $S_R = S \cap P_R$, $S_A = S \setminus S_R$, where S_A is the operation places and S_R is the resource places. $[S]$ denotes the following set of state places: $[S] = (U_{r \in S_R} H(r)) \setminus S_A$, where $H(r) = (\|I_r\|^+ \setminus r)$, $\|I_r\|^+ = \{p \mid I(p) > 0\}$ indicates the positive support of place invariant I , $\forall i, j \in \{1, 2, \dots, n\} \ i \neq j, H(r_i) \cap H(r_j) = \emptyset$. For a given siphon S , $[S]$ is the set of holders, corresponding to resources in S that do not belong to S . $[S]$ is called S 's complementary set.
- (3) Given the fact that $[S]$ is a complementary set of S , add a control place for $[S]$ and the initial marking of control place is $M_{0A}(V_S) = M_0(S) - 1$.

Based on the concept of strict minimal siphon, the proposed deadlock prevention algorithm developed by Ezpeleta et al. [13] is stated as follows.

Algorithm 2 (SMS based policy).

Input. Petri net model (N, M_0) of an AMS, where $N = (P^0 \cup P_A \cup P_R, T, F, W)$.

Step 1. For a given Petri net (N, M_0) , compute all SMSs.

Step 2. For each siphon SMS, add a monitor V_S such that

- (i) the V_S output arcs are connected to the source transitions leading to the sink transitions of S and all arcs weights are ones;
- (ii) the V_S input arcs are connected from the stealing places of S and all arcs weights are ones;
- (iii) $M_{0A}(V_S) = M_0(S) - 1$, where $M_{0A}(V_S)$ is an initial marking of a monitor.

Step 3. Repeat Step 2 until all existing SMSs are covered.

Step 4. Add all monitors into (N, M_0) and the resulting net is denoted by (N_1, M_1) .

Step 5. Output (N_1, M_1) .

Step 6. End.

For instance, consider the AMS shown in Figure 1(a). Its Petri net model is shown in Figure 1(b). It consists of one robot R1 that can hold a part at a time, two machines M1 and M2, where each machine can process a part at a time, two loading buffers (I1 and I2), and unloading buffers (O1 and O2). Two parts types are considered in the system: PA and PB. PA moves to M1 and PB moves to M2. The robot R1 reaches the machine that finishes its operation first and grips and loads the part to the next machine 1 or machine 2. The Petri net model consists of 11 places and 8 transitions. The places have the following set partition: $P^0 = \{P_1, P_8\}$, $P_R = \{P_9, P_{10}, P_{11}\}$, and $P_A = \{P_2, \dots, P_7\}$, where P^0 is the input places, P_R is the resource places, and P_A is the operation places. It has 8 minimal siphons, 3 of which are strict minimal siphons and 20 reachable markings. The three SMSs are $S_1 = \{P_4, P_7, P_9, P_{10}, P_{11}\}$, $S_2 = \{P_4, P_6, P_{10}, P_{11}\}$, and $S_3 = \{P_3, P_7, P_9, P_{10}\}$. Table 1 shows the required monitors using Algorithm 2 for Figure 1(b). Figure 2 illustrates the controlled system of the Petri net model in Figure 1(b) after adding monitors by using Algorithm 2.

4.1.2. Elementary Siphons Control Method. The strict minimal siphons (SMSs) are classified into elementary and dependent ones. Π is used to denote the set of strict minimal siphons while Π_E and Π_D denote the sets of elementary and dependent (redundant) ones, respectively. Otherwise, when a strict minimal one is mentioned, it can be called a siphon.

Definition 3. Let an ordinary Petri net (N, M_0) and S is a siphon, $S \subseteq P$. Place vector λ_S is said to be the characteristic place vector of S if for all $p \in S$, $\lambda_S(p) = 1$; otherwise, $\lambda_S(p) = 0$.

Definition 4. Let $N = (P, T, F)$ be an ordinary Petri net with $|P| = m$, $|T| = n$, and we suppose that N has f SMS, $S_1, S_2, \dots$, and $S_f, m, n, f \in \mathbb{N}$. Let $\lambda_{S_i}(\eta_{S_i})$ be the characteristic place (transition) vector of siphon S_i , $i \in \{1, 2, \dots, n\}$. We define $[\lambda]_{f \times m} = [\lambda_{S_1}, \lambda_{S_2}, \dots, \lambda_{S_f}]^T$ and $[\eta]_{f \times n} = [\eta]_{f \times m} \cdot [N]_{m \times n}$ are said to be the characteristic place vector and transition vector matrices of the siphons of N , respectively.

Proposition 5. Let N be an ordinary Petri net and let $[\eta]$ be the characteristic transition vector matrix of the siphons of it. The set of elementary siphons in N can be defined as a rank of $[\eta]$.

From Proposition 5, it can easily enumerate all elementary siphons in a Petri net system (N, M_0) given all siphons. First, compute matrices $[\lambda]$ and $[\eta]$. Then independent linear vectors can be determined in $[\eta]$. Finally, the siphons that correspond to these independent linear vectors are called the elementary siphons in the Petri net (N, M_0) . Li and Zhou [11] introduced some results and theorems as follows.

Theorem 6. Let N_0 be a marked ordinary Petri net and $S = \{p_{i1}, p_{i2}, \dots, p_{in}\}$ be a siphon in N_0 . Monitor V_S is added to N_0 , and the new net system can be denoted by (N_1, M_1) , such that (a) $I_1 = (\dots, 1p_{i1}, 1p_{i2}, \dots, 1p_{in}, \dots, -1V_S, \dots)^T$ is a place invariant of N_1 and (b) $M_1(V_S) = M_0(S) - \xi_S$, where ξ_S is said to

TABLE 1: Control places computations by Algorithm 2.

SMS	$\ I\ ^+$	$H(r_R)$	$[S]$	$\bullet V_{Si}$	$V_{Si} \bullet$	$M_{oA}(V_{Si})$
$S_1 = \{P_4, P_7, P_9, P_{10}, P_{11}\},$ $S_A = \{P_4, P_7\},$ $S_R = \{P_9, P_{10}, P_{11}\}.$	$\ I_{p9}\ ^+ = \{p_2, p_7, p_9\},$ $\ I_{p10}\ ^+ = \{p_3, p_6, p_{10}\},$ $\ I_{p11}\ ^+ = \{p_4, p_5, p_{11}\}.$	$H(r_9) = \{p_2, p_7\},$ $H(r_{10}) = \{p_3, p_6\},$ $H(r_{11}) = \{p_4, p_5\}.$	$[S_1] = \{p_2, p_3, p_5, p_6\}.$	t_3, t_7	t_1, t_5	2
$S_2 = \{P_4, P_6, P_{10}, P_{11}\},$ $S_A = \{P_4, P_6\},$ $S_R = \{P_{10}, P_{11}\}.$	$\ I_{p10}\ ^+ = \{p_3, p_6, p_{10}\},$ $\ I_{p11}\ ^+ = \{p_4, p_5, p_{11}\}.$	$H(r_{10}) = \{p_3, p_6\},$ $H(r_{11}) = \{p_4, p_5\}.$	$[S_2] = \{p_3, p_5\}.$	t_3, t_6	t_2, t_5	1
$S_3 = \{P_3, P_7, P_9, P_{10}\},$ $S_A = \{P_3, P_7\},$ $S_R = \{P_9, P_{10}\}.$	$\ I_{p9}\ ^+ = \{p_2, p_7, p_9\},$ $\ I_{p10}\ ^+ = \{p_3, p_6, p_{10}\}.$	$H(r_9) = \{p_2, p_7\},$ $H(r_{10}) = \{p_3, p_6\}.$	$[S_3] = \{p_2, p_6\}.$	t_2, t_7	t_1, t_5	1

FIGURE 1: (a) An AMS example and (b) Petri net model of an AMS.

FIGURE 2: Controlled system of the Petri net in Figure 1(b) by Algorithm 2.

be the control depth variable of siphon S , which states the least number of tokens that siphon can hold. Evidently is greater than or equal to 1 to fulfil purpose of control $1 \leq \xi_S \leq M_0(S) - 1$, and (c) for all $p \in P_0$, $M_1(p) = M_0(p)$, where P_0 is the set of operation places of N_0 , such that S can be called invariant-controlled.

Theorem 7. Let (N, M_0) be an ordinary Petri net and let S_0 be a generally dependent siphon on elementary

siphons $S_1, S_2, \dots, S_n, S_{n+1}, S_{n+2}, \dots$ and S_{n+m} that means $\eta_{S_0} = \sum_{i=1}^n a_i \cdot \eta_{S_i} - \sum_{j=n+1}^{n+m} a_j \cdot \eta_{S_j}$. If $S_1, S_2, \dots, S_n, S_{n+1}, S_{n+2}, \dots$ and S_{n+m} are invariant-controlled by adding monitors $V_{S_1}, V_{S_2}, \dots, V_{S_{n+m}}, \dots$ and $M_0(S_0) > \sum_{i=1}^n a_i \cdot M_0(S_i) - \sum_{j=n+1}^{n+m} a_j \cdot \xi_{S_j}$, where $1 \leq \xi_{S_i} \leq M_0(S_i) - 1$, S_0 is called controlled. Based on the concept of elementary siphons, the proposed deadlock prevention algorithm developed by Li and Zhou [11] is stated as follows.

Algorithm 8 (elementary siphon-based policy).

Input. Petri net model (N, M_0) of an AMS, where $N = (P^0 \cup P_A \cup P_R, T, F, W)$.

Step 1. For a given Petri net (N, M_0) , compute all SMSs.

Step 2. Compute the transition vector matrix $[\eta]$ of the SMS.

Step 3. Compute the elementary siphons of Petri net N . The remaining are the dependent siphons.

Step 4. For each siphon elementary siphon, add a monitor V_S such that

- (i) the V_S output arcs are connected to the source transitions leading to the sink transitions of S and all arcs weights are ones;

TABLE 2: Marking relationships between dependent and elementary siphons.

Dependent	η relationship	Initial marking relationships, $M_0(S_0) > \sum_{i=1}^n a_i \cdot M_0(S_i) - \sum_{i=1}^n a_i \cdot \xi_{S_i}$	Controlled
S_1	$\eta_1 = \eta_2 + \eta_3$	$M_0(S_1) > (M_0(S_2) + M_0(S_3)) - (\xi_{S_2} + \xi_{S_3})$ $3 > (2 + 2) - (1 + 1), 3 > 2$	Yes

(ii) the V_S input arcs are connected from the stealing places of S and all arcs weights are ones;

(iii) $M_{0A}(V_S) = M_0(S) - \xi_S$, where $M_{0A}(V_S)$ is an initial marking of a monitor and $1 \leq \xi_S \leq M_0(S) - 1$.

Step 5. Repeat Step 4 until all elementary siphons are covered.

Step 6. Adjust ξ_i , such that each dependent siphon is controlled.

Step 7. Add all monitors to (N, M_0) and the resulting net is denoted by (N_1, M_1) .

Step 8. Output (N_1, M_1) .

Step 9. End.

For example, there are three SMSs in the Petri net model shown in Figure 1(b), which are $S_1 = \{p_4, p_7, p_9, p_{10}, p_{11}\}$, $S_2 = \{p_4, p_6, p_{10}, p_{11}\}$, and $S_3 = \{p_3, p_7, p_9, p_{10}\}$. One can obtain that $\lambda_{S_1} = (0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1)^T$, $\lambda_{S_2} = (0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1)^T$, and $\lambda_{S_3} = (0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0)^T$. Hence, $\eta_{S_1} = (-1, 0, 1, 0, -1, 0, 1, 0)^T$, $\eta_{S_2} = (0, -1, 1, 0, -1, 1, 0, 0)^T$, and $\eta_{S_3} = (-1, 1, 0, 0, 0, -1, 1, 0)^T$. The transition vector matrix $[\eta]$ for the Petri net model in Figure 1(b) is as follows. Obviously, the rank of $[\eta]$ is 2 since the first row η_{S_1} can be linearly represented by the second and third rows, and we can see that $\eta_{S_1} = \eta_{S_2} + \eta_{S_3}$. As a result, the siphons that correspond to the second and third rows S_2 and S_3 are elementary siphons. Thus, S_1 is a dependent siphon:

$$[\eta] = \begin{pmatrix} -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 \end{pmatrix}. \quad (1)$$

The dependent siphons and their elementary siphons, the initial marking relationships between dependent and elementary siphons, and the controllability of dependent siphons due to Theorem 7 are shown in Table 2. Table 3 shows the required monitors using Algorithm 8 for Figure 1(b). Figure 3 presents the controlled system of the Petri net model in Figure 1(b) after adding monitors by using Algorithm 8.

4.2. Iterative Methods. This section presents two techniques that are motivated and developed by [25]. These techniques are used to design a place invariant (PI) that can prohibit as many forbidden bad markings (FBMs) as possible. First, the vector covering method is used to find the minimal covering sets of legal markings and FBMs (see Chen et al. [1]). Then, solving the integer linear programming problem (ILPP) can obtain the coefficients of the place invariant and monitors. An

TABLE 3: Required monitors using Algorithm 8.

Siphon	$\bullet V_{S_i}$	$V_{S_i} \bullet$	$M_{0A}(V_{S_i})$
S_2	t_3, t_6	t_2, t_5	1
S_3	t_2, t_7	t_1, t_5	1

FIGURE 3: Controlled system of the Petri net in Figure 1(b) by Algorithm 8.

iterative monitor design process is executed. At each iteration, a place invariant is designed to prohibit as many FBMs as possible. All FBMs that are prohibited by the place invariant are eliminated from the minimal covered set of FBMs. This process is terminated when all the minimal covered sets of FBMs are forbidden.

4.2.1. Maximal Number of Forbidding FBM Problem 1. The Maximal number of Forbidding FBM Problem 1 (MFFP1) is a technique used to design a monitor synthesis for FBMs. This technique used ILPP to obtain the coefficients of the place invariant that prohibit as many FBMs as possible. The MFFP1 is shown in the following mathematical model.

MFFP1 is

$$\begin{aligned} \max \quad & f = \sum_{k \in N_{\text{FBM}}} f_k \\ \text{subject to} \quad & \sum_{i \in N_A} l_i \cdot M_l(p_i) \leq \beta, \quad \forall M_l \in M_L^* \\ & \sum_{i \in N_A} l_i \cdot M_k(p_i) \geq \beta + 1 - Q \cdot (1 - f_k), \\ & \forall M_k \in M_{\text{FBM}}^* \\ & l_i \in \{0, 1, 2, \dots\}, \quad \forall i \in \mathbb{N} \\ & \beta \in \{1, 2, \dots\} \\ & f_k \in \{0, 1\}, \quad \forall k \in N_{\text{FBM}}^* \end{aligned} \quad (2)$$

where M_L^* and M_{FBM}^* are the minimal covering sets of legal markings and FBMs, respectively. The objective function f is used to maximize the number of FBMs that are forbidden by place invariant. If $f = 0$, we have $f_k = 0$, for all $k \in N_{FBM}^*$ (N_{FBM}^* is used to denote $\{i \mid M_i \in M_{FBM}^*\}$): that is to say, no FBMs in M_{FBM}^* can be prohibited by place invariant. In (2), let I be a place invariant, where l_i 's ($i \in \mathbb{N}$) are the coefficients of I , β is a positive integer variable, and $M_i(p_i)$ ($\forall M_i \in M_L^*$) are the number of tokens in minimal covering set of legal markings.

All legal markings should be kept after a monitor is added, implying that every marking $M_l \in M_L^*$ cannot be forbidden from being reached, and coefficients l_i 's ($i \in \mathbb{N}$) should satisfy (2) that is called the reachability condition.

In (3), Q is a positive integer constant number that must be large enough, $M_k(p_i)$ ($\forall M_k \in M_{FBM}^*$) are the number of tokens in minimal covering set of FBMs, and f_k 's ($f_k \in \{0, 1\}$, $k = 1, 2, \dots$) are a set of M_{FBM}^* variables. Moreover, in (3), $f_k = 1$ implies that M_k is prohibited by I and $f_k = 0$ implies that this constraint is redundant and M_k cannot be prohibited by I .

Based on the concept of MFFP1, the proposed deadlock prevention algorithm developed in [25] is stated as follows.

Algorithm 9 (MFFP1 based policy).

Input. Petri net model (N, M_0) of an AMS, where $N = (P^0 \cup P_A \cup P_R, T, F, W)$.

Step 1. For a given Petri net (N, M_0) , find the sets of FBMs M_{FBM}^* and legal markings M_L^* .

Step 2. Find the minimal covered sets of FBMs M_{FBM}^* and legal markings M_L^* .

Step 3. $V_M := \emptyset$. $/^*V_M$ is used to indicate the set of monitors to be found. $^*/$

Step 4. While $\{M_{FBM}^* \neq \emptyset\}$, do the following.

- (i) Design MFFP1.
- (ii) Solve MFFP1. Let l_i 's ($i \in \mathbb{N}$) and β be the solution if $f^* \neq 0$.
Otherwise, exit, as the solution does not exist.
- (iii) Design a place invariant I and a monitor V_S .
- (iv) $V_M := V_M \cup \{V_S\}$ and $M_{FBM}^* = M_{FBM}^* - F_I$. $/^*F_I$ is covered M_{FBM}^*
endwhile.

Step 5. Add all monitors in V_M to (N, M_0) and the resulting net is denoted by (N_1, M_1) .

Step 6. Output (N_1, M_1) .

Step 7. End.

For the Petri net model shown in Figure 1(b), we have $M_{FBM}^* = \{p_2 + p_5, p_3 + p_5, p_2 + p_6\}$ and $M_L^* = \{p_2 + p_3 + p_4, p_5 + p_6 + p_7\}$. Algorithm 9 is considered. At the first iteration, let

I_1 be the place invariant to be calculated. I_1 should satisfy (2) for the two legal markings in M_L^* : that is, $l_2 \cdot 1 + l_3 \cdot 1 + l_4 \cdot 1 \leq \beta$ and $l_5 \cdot 1 + l_6 \cdot 1 + l_7 \cdot 1 \leq \beta$. Therefore, we have the following two constraints:

$$\begin{aligned} l_2 + l_3 + l_4 &\leq \beta, \\ l_5 + l_6 + l_7 &\leq \beta. \end{aligned} \quad (4)$$

We introduce three variables f_1, f_2 , and f_3 ($f_1, f_2, f_3 \in \{0, 1\}$) to represent whether I_1 forbids $FBM_1 = p_2 + p_5$, $FBM_2 = p_3 + p_5$, and $FBM_3 = p_2 + p_6$, respectively. Therefore, we have the following three constraints: that is,

$$\begin{aligned} l_2 + l_5 &\geq \beta + 1 - Q \cdot (1 - f_1), \\ l_3 + l_5 &\geq \beta + 1 - Q \cdot (1 - f_2), \\ l_2 + l_6 &\geq \beta + 1 - Q \cdot (1 - f_3). \end{aligned} \quad (5)$$

Finally, MFFP1 is defined as follows.

MFFP1 is

$$\begin{aligned} \max \quad & f = f_1 + f_2 + f_3 \\ \text{subject to} \quad & l_2 + l_3 + l_4 \leq \beta \\ & l_5 + l_6 + l_7 \leq \beta \\ & l_2 + l_5 \geq \beta + 1 - Q \cdot (1 - f_1) \\ & l_3 + l_5 \geq \beta + 1 - Q \cdot (1 - f_2) \\ & l_2 + l_6 \geq \beta + 1 - Q \cdot (1 - f_3) \\ & l_i \in \{0, 1, 2, \dots\}, \quad \forall i \in \{2, 3, 4, 5, 6, 7\} \\ & \beta \in \{1, 2, \dots\} \\ & f_k \in \{0, 1\}, \quad \forall k \in \{1, 2, 3\}. \end{aligned} \quad (6)$$

The optimal solution of the above MFFP1 is $l_2 = 2, l_5 = 1, l_6 = 1, \beta = 2, f_1 = 1$, and $f_3 = 1$, and all the remaining variables are equal to zero. Then, a monitor V_{S1} is designed for $I_1: 2\mu_2 + \mu_5 + \mu_6 + \mu_{VS1} = 2$. As a result, we have $M_{0A}(V_{S1}) = \beta = 2$, $^*V_{S1} = \{2t_2, t_7\}$, and $V_{S1}^* = \{2t_1, t_5\}$. FBM_1 and FBM_3 are prohibited by I_1 : that is, $F_{I1} = \{FBM_1, FBM_3\}$. Therefore, we have $M_{FBM}^* := M_{FBM}^* - F_{I1}$: that is, $M_{FBM}^* = FBM_2 = \{p_3 + p_5\}$.

At the next iteration, let I_2 place invariant be calculated. The new MFFP1 can be easily obtained by eliminating constraints that related to FBM_1 and FBM_3 from the first MFFP1. Therefore, the new MFFP1 is stated as follows.

MFFP1 is

$$\begin{aligned} \max \quad & f = f_2 \\ \text{subject to} \quad & l_2 + l_3 + l_4 \leq \beta \\ & l_5 + l_6 + l_7 \leq \beta \\ & l_3 + l_5 \geq \beta + 1 - Q \cdot (1 - f_2) \end{aligned}$$

$$\begin{aligned}
l_i &\in \{0, 1, 2, \dots\}, \quad \forall i \in \{2, 3, 4, 5, 6, 7\} \\
\beta &\in \{1, 2, \dots\} \\
f_2 &\in \{0, 1\}.
\end{aligned} \tag{7}$$

The optimal solution of the above MFFP1 is $l_3 = 1, l_5 = 1, \beta = 1$, and $f_2 = 1$ and all the remaining variables are equal to zero. Then, a monitor V_{S2} is designed for $I_2: \mu_3 + \mu_5 + \mu_{VS2} = 1$. As a result, we have $M_{0A}(V_{S2}) = \beta = 1, {}^*V_{S1} = \{t_3, t_6\}$, and $V_{S1}^* = \{t_2, t_5\}$. Only FBM_2 is prohibited by I_2 : that is, $F_{I2} = \{FBM_2\}$. Therefore, we have $M_{FBM}^* := M_{FBM}^* - F_{I2}$: that is, $M_{FBM}^* = \emptyset$.

Algorithm 9 is terminated. Totally, there are two monitors computed for this net. Adding the two monitors to the original Petri net model, the controlled system after addition is as shown in Figure 4. Table 4 shows the two required monitors by using Algorithm 9.

4.2.2. Maximal Number of Forbidding FBM Problem 2. The Maximal number of Forbidding FBM Problem 1 has been modified to improve its efficiency and its number of variables and constraints are slightly reduced. The modified technique is called a Maximal number of Forbidding FBM Problem 2 (MFFP2). The MFFP1 is shown in the following mathematical model.

MFFP2 is

$$\begin{aligned}
\max \quad & f = \sum_{k \in N_{FBM}^*, k \neq j} f_k \\
\text{subject to} \quad & \sum_{i \in N_A} l_i \cdot (M_l(p_i) - M_j(p_i)) \leq -1, \\
& \forall M_l \in M_L^* \tag{8}
\end{aligned}$$

$$\begin{aligned}
& \sum_{i \in N_A} l_i \cdot (M_k(p_i) - M_j(p_i)) \\
& \geq -Q \cdot (1 - f_k), \\
& \forall M_k \in M_{FBM}^*, k \neq j \\
& l_i \in \{0, 1, 2, \dots\}, \quad \forall i \in \mathbb{N} \\
& f_k \in \{0, 1\}, \quad \forall k \in N_{FBM}^*, k \neq j.
\end{aligned} \tag{9}$$

The objective function is maximizing f the number of FBMs that are prohibited by place invariant. If no solution has been obtained from MFFP2 for an FBM, then there are not any FBMs in M_{FBM}^* that can be forbidden by that place invariant.

In (8), l_i 's ($i \in \mathbb{N}$) are the coefficients of I , $M_l(p_i)$ ($\forall M_l \in M_L^*$) are the number of tokens in minimal covering set of legal markings, and $M_j(p_i)$ are the number of tokens in selected FBM $_j$. In (9), Q is a positive integer constant number that must be large enough, $M_k(p_i)$ ($\forall M_k \in M_{FBM}^*$) are the number of tokens in minimal covering set of FBMs, and f_k 's ($f_k \in \{0, 1\}, k = 1, 2, \dots$) are a set of M_{FBM}^* variables. In (9), $f_k = 1$ implies that M_k is prohibited by I and $f_k = 0$ implies that this constraint is redundant and M_k cannot be prohibited by I .

TABLE 4: Required monitors using Algorithm 9.

i	${}^*V_{Si}$	V_{Si}^*	$M_{0A}(V_{Si})$
1	$2t_2, t_7$	$2t_1, t_5$	2
2	t_3, t_6	t_2, t_5	1

FIGURE 4: Controlled system of the Petri net in Figure 1(b) by Algorithm 9.

Based on the concept of MFFP2, the proposed deadlock prevention algorithm developed in [25] is stated as follows.

Algorithm 10 (MFFP2 based policy).

Input. Petri net model (N, M_0) of an AMS, where $N = (P^0 \cup P_A \cup P_R, T, F, W)$.

Step 1. For a given Petri net (N, M_0) , find the sets of FBMs M_{FBM} and legal markings M_L .

Step 2. Find the minimal covered sets of FBMs M_{FBM}^* and legal markings M_L^* .

Step 3. $V_M := \emptyset$. $/^*V_M$ is used to indicate the set of monitors to be found. $*/$

Step 4. While $\{M_{FBM}^* \neq \emptyset\}$, do the following.

- (i) Design MFF2.
- (ii) Solve MFFP2. Let l_i 's ($i \in N_A$) and β be the solution if $f^* \neq 0$. Otherwise, exit, as the solution does not exist.
- (iii) Design a place invariant I and a monitor V_S . $V_M := V_M \cup \{V_S\}$ and $M_{FBM}^* = M_{FBM}^* - F_I$. $/^*F_I$ is covered M_{FBM}^* endwhile.

Step 5. Add all monitors in V_M to (N, M_0) and the resulting net is denoted by (N_1, M_1) .

Step 6. Output (N_1, M_1) .

Step 7. End.

Consider the Petri net model shown in Figure 1(b) by Algorithm 10. At the first iteration, $FBM_1 = p_2 + p_5$ is chosen. Let I_1 be the place invariant to prohibit FBM_1 , whose coefficients are l_i 's ($i \in \{2, 3, 4, 5, 6, 7\}$). I_1 should satisfy

(8) that is called the reachability condition and does not prohibit any marking in M_L^* . Therefore, I_1 should satisfy two constraints:

$$\begin{aligned} & l_2 \cdot (1 - 1) + l_3 \cdot (1 - 0) + l_4 \cdot (1 - 0) + l_5 \cdot (0 - 1) + l_6 \\ & \cdot (0 - 0) + l_7 \cdot (0 - 0) \leq -1, \\ & l_2 \cdot (0 - 1) + l_3 \cdot (0 - 0) + l_4 \cdot (0 - 0) + l_5 \cdot (1 - 1) + l_6 \\ & \cdot (1 - 0) + l_7 \cdot (1 - 0) \leq -1 \end{aligned} \quad (10)$$

for the two legal markings $p_2 + p_3 + p_4$ and $p_5 + p_6 + p_7$ in M_L^* , respectively. After simplifying the two constraints, we have

$$\begin{aligned} & l_3 + l_4 - l_5 \leq -1, \\ & -l_2 + l_6 + l_7 \leq -1. \end{aligned} \quad (11)$$

Two variables are introduced f_2 and f_3 to represent whether I_1 forbids $\text{FBM}_2 = p_3 + p_5$ and $\text{FBM}_3 = p_2 + p_6$, respectively. Therefore, we have the two constraints: that is, $l_2 \cdot (0 - 1) + l_3 \cdot (1 - 0) + l_4 \cdot (0 - 0) + l_5 \cdot (1 - 1) + l_6 \cdot (0 - 0) + l_7 \cdot (0 - 0) \geq -Q \cdot (1 - f_2)$ and $l_2 \cdot (1 - 1) + l_3 \cdot (0 - 0) + l_4 \cdot (0 - 0) + l_5 \cdot (0 - 1) + l_6 \cdot (1 - 0) + l_7 \cdot (0 - 0) \geq -Q \cdot (1 - f_3)$, where Q is a positive integer constant that must be big enough and $f_2 = 1$ ($f_3 = 1$) implies that FBM_2 (FBM_3) is forbidden by I_1 and $f_2 = 0$ ($f_3 = 0$) implies that FBM_2 (FBM_3) cannot be forbidden by I_1 . After simplifying the two constraints, we have

$$\begin{aligned} & -l_2 + l_3 \geq -Q \cdot (1 - f_2), \\ & -l_5 + l_6 \geq -Q \cdot (1 - f_3). \end{aligned} \quad (12)$$

Finally, from the above constraints MFFP2 for FBM_1 is stated as follows.

MFFP2 is

$$\begin{aligned} & \max \quad f = f_2 + f_3 \\ & \text{subject to} \quad l_3 + l_4 - l_5 \leq -1 \\ & \quad -l_2 + l_6 + l_7 \leq -1 \\ & \quad -l_2 + l_3 \geq -Q \cdot (1 - f_2) \\ & \quad -l_5 + l_6 \geq -Q \cdot (1 - f_3) \\ & \quad l_i \in \{0, 1, 2, \dots\}, \quad \forall i \in \{2, 3, 4, 5, 6, 7\} \\ & \quad f_k \in \{0, 1\}, \quad \forall k \in \{2, 3\}. \end{aligned} \quad (13)$$

The optimal solution of the above MFFP2 is $l_2 = 1$, $l_3 = 1$, $l_5 = 2$, and $f_2 = 1$, and all the remaining variables are equal to zero. Then, a monitor V_{S1} is designed for I_1 : $\mu_2 + \mu_3 + 2\mu_5 + \mu_{V_{S1}} = 2$. As a result, we have $M_{0A}(V_{S1}) = 2$, $\bullet V_{S1} = \{t_3, 2t_6\}$, and $V_{S1}^\bullet = \{t_1, 2t_5\}$. FBM_1 and FBM_2 are prohibited by I_1 : that is, $F_{I1} = \{\text{FBM}_1, \text{FBM}_2\}$. Therefore, we have $M_{\text{FBM}}^* := M_{\text{FBM}}^* - F_{I1}$: that is, $M_{\text{FBM}}^* = \{p_2 + p_6\}$. At the second iteration, the last FBM is considered, that is, FBM_3 . Let I_2 be the place invariant to prohibit FBM_3 . First, the coefficients of I_2 should fulfill the reachability conditions: that is,

$$\begin{aligned} & l_3 + l_4 - l_6 \leq -1, \\ & -l_2 + l_5 + l_7 \leq -1. \end{aligned} \quad (14)$$

TABLE 5: Required monitors using Algorithm 10.

i	$\bullet V_{Si}$	$V_{Si}^\bullet$	$M_{0A}(V_{Si})$
1	$t_3, 2t_6$	$t_1, 2t_5$	2
2	t_2, t_7	t_1, t_6	1

FIGURE 5: Controlled system of the Petri net in Figure 1(b) by Algorithm 10.

Since FBM_3 is the last marking in M_{FBM}^* , no other FBMs in the set require to be prohibited by I_2 . By solving the above integer linear system, we have a solution with $l_2 = 1$, $l_6 = 1$, and all the remaining variables are equal to zero. Then, a monitor V_{S2} is designed for I_2 : $\mu_2 + \mu_6 + \mu_{V_{S2}} = 2$. As a result, we have $M_{0A}(V_{S2}) = 1$, $\bullet V_{S2} = \{t_2, t_7\}$, and $V_{S2}^\bullet = \{t_1, t_6\}$. Only FBM_3 is prohibited by I_2 : that is, $F_{I2} = \{\text{FBM}_3\}$. Therefore, we have $M_{\text{FBM}}^* := M_{\text{FBM}}^* - F_{I1}$: that is, $M_{\text{FBM}}^* = \emptyset$.

Algorithm 10 is terminated. Totally, there are two monitors computed for this net. We add the two monitors to the original Petri net model. The controlled system after addition is as shown in Figure 5. Table 5 shows the two required monitors by using Algorithm 10.

5. Case Studies

This section presents four case studies and the applications of the proposed applied algorithms.

(a) The production sequence of the first AMS case study is shown in Figure 6(a). It consists of two robots (R1 and R2; each one can hold a part at a time) and five machines (M1, M2, M3, M4, and M5; each one can process one part at a time). There are three loading buffers I1–3 and three unloading buffers O1–3. Three types of parts are considered in the system: PA, PB, and PC. The model of Petri net for this case study is shown in Figure 6(b).

(b) The production sequence of the second AMS case study is shown in Figure 7(a) and several studies have been done in this case study (see Elzepeleta et al. [13] and Chen et al. [25]). The system consists of three robots (R1, R2, and R3; each one can hold a part at a time) and four machines (M1, M2, M3, and M4; each machine can process two parts at a time). There are three loading buffers I1–3 and three unloading buffers O1–3. Three types of parts are considered in the system: PA, PB, and PC. The model of Petri net for this case study is shown in Figure 7(b).

FIGURE 6: (a) Production sequence and (b) Petri net model of case study 1.

FIGURE 7: (a) Production sequence and (b) Petri net model of case study 2.

(c) Next, the production sequence of the third AMS case study is shown in Figure 8(a). The cell consists of two robots (R1 and R2; each one can hold a part at a time) and four machines (M1, M2, and M3; each machine can process a part at a time; M4 can process two products at

a time) and, for loading and unloading the cell, two loading buffers (I1 and I2) and two unloading buffers (O1 and O2). Two types of parts are considered in the cell: PA and PB. The model of Petri net for this case study is illustrated in Figure 8(b).

FIGURE 8: (a) Production sequence and (b) Petri net model of case study 3.

(d) Finally, the production sequence of the fourth AMS case study is shown in Figure 9(a) and several studies have been achieved in this case study (see Uzam [9], Chen et al. [25], Piroddi et al. [26], and Piroddi et al. [27]). The system is composed of two robots (R1 and R2; each robot can hold a part at a time) and four machines (M1, M2, M3, and M4; each machine can process one part at a time). There are two loading buffers (I1 and I2) and two unloading buffers (O1 and O2). PA and PB are considered in the system. The model of Petri net for this case study is shown in Figure 9(b).

The main parameters of all case studies are presented in Table 6 such as number of places, number of transitions, set partition, SMS, dependent siphons, elementary siphons, reachable markings, FBMs, legal markings, M_{FBM}^* , and M_L^* . Table 7 illustrates the supervisor performance of applied algorithms for all case studies with respect to the numbers of monitors, arcs, and reachable states.

From Table 7, Algorithms 9 and 10 obtain fewer numbers of monitors than Algorithms 2 and 8 for all case studies while Algorithm 9 provides less or equal number of arcs than the other algorithms in case studies 1, 2, and 3. In case study 4, Algorithm 10 obtains less number of arcs than the other algorithms. As for the reachable states, all algorithms provide the same number of reachable states in case study 1, which is 1072 while both Algorithms 2 and 8 have the same number of reachable states, which is 6287 in case study 2. Moreover, in case study 2, both Algorithms 9 and 10 provide the same number of reachable states, which is 21581. Moreover, in case

study 3, both Algorithms 9 and 10 outperform Algorithms 2 and 8 in number of reachable states, which have 656 reachable states. Finally, in case study 4, Algorithms 8, 9, and 10 provide better reachable states than Algorithm 2, which is 205. The required monitors using applied algorithms for all case studied are shown in Tables 8, 9, 10, and 11.

6. Computational Results and Analysis

This section presents that the computational results and analysis of the applied methods are conducted for four AMSs case studies. All cases have been modeled using MATLAB software. The simulation is done for 24 time units. After running and simulating the case studies, the results of MATLAB simulation in terms of the utilization resources and throughput can be summarized as follows. For case study 1, Figures 10 and 11 show the resources utilization and throughput of case study 1, respectively. From the figures, it can be found that all applied Algorithms 8, 9, and 10 obtain the same utilization for all resources, while Algorithm 2 obtains utilization better than Algorithms 8, 9, and 10 at M1, M2, M5, and R1, and same utilization at M4 and R2. For the throughput, Algorithm 2 obtains greater produced number of part A and part C than Algorithms 8, 9, and 10. Moreover, Algorithms 8, 9, and 10 provide the same throughput for all part types. For case study 2, Figures 12 and 13 show the resources utilization and throughput of the case study 2, respectively.

FIGURE 9: (a) Production sequence and (b) Petri net model of case study 4.

TABLE 6: Main parameters of all case studies.

Parameter	Case study			
	1	2	3	4
Number of places	26	26	20	19
Number of transitions	20	20	16	14
Set partition	$P_0 = \{p_1, p_{25}, p_{26}\}$ $P_R = \{p_{18}-p_{24}\}$ $P_A = \{p_2, \dots, p_{17}\}$	$P_0 = \{p_1, p_{10}, p_{14}\}$ $P_R = \{p_{20}, \dots, p_{26}\}$ $P_A = \{p_2, \dots, p_4, p_6, \dots, p_{13}, p_{15}, \dots, p_{19}\}$	$P_0 = \{p_1, p_{20}\}$ $P_R = \{p_{14}, \dots, p_{19}\}$ $P_A = \{p_2, \dots, p_{13}\}$	$P_0 = \{p_1, p_{19}\}$ $P_R = \{p_{13}, \dots, p_{18}\}$ $P_A = \{p_2, \dots, p_{12}\}$
SMS	11	18	12	5
Dependent siphons	6	9	7	2
Elementary siphons	5	9	5	3
Reachable markings	1492	26750	836	282
FBMs	388	4211	156	54
Legal markings	1072	21581	656	205
M_{FBM}^*	9	34	7	8
M_L^*	92	393	56	26

TABLE 7: Supervisor performance of applied algorithms for all case studies.

Case study	Parameter	Algorithm 2	Algorithm 8	Algorithm 9	Algorithm 10
1	Monitors number	11	5	3	3
	Arcs number	66	23	14	31
	Reachable states number	1072	1072	1072	1072
2	Monitors number	18	9	6	6
	Arcs number	106	50	45	59
	Reachable states number	6287	6287	21581	21581
3	Monitors number	12	5	4	4
	Arcs number	61	25	21	21
	Reachable states number	436	489	656	656
4	Monitors number	5	3	2	2
	Arcs number	21	13	15	12
	Reachable states number	182	205	205	205

TABLE 8: Required monitors using applied algorithms for case study 1.

Algorithm	i	Siphon	$\bullet V_{Si}$	$V_{Si} \bullet$	$M_{oA}(V_{Si})$
Algorithm 2	1	S_1	$t_5, t_{11}, t_{18}, t_{19}$	t_1, t_7, t_{13}	6
	2	S_2	$t_5, t_{11}, t_{15}, t_{18}$	t_1, t_7, t_{13}, t_{16}	5
	3	S_3	t_5, t_{11}, t_{15}	t_1, t_9, t_{13}	4
	4	S_4	t_3, t_9, t_{18}, t_{19}	t_1, t_7, t_{15}	3
	5	S_5	$t_5, t_{11}, t_{18}, t_{19}$	t_3, t_7, t_{13}	5
	6	S_6	$t_5, t_{11}, t_{15}, t_{18}$	t_3, t_7, t_{13}, t_{16}	4
	7	S_7	t_5, t_{11}, t_{15}	t_3, t_9, t_{13}	3
	8	S_8	t_9, t_{18}, t_{19}	t_7, t_{15}	2
	9	S_9	t_3, t_9, t_{18}	t_1, t_7, t_{16}	2
	10	S_{10}	t_3	t_1	1
	11	S_{11}	t_9, t_{18}	t_7, t_{16}	1
Algorithm 8	1	S_3	t_5, t_{11}, t_{15}	t_1, t_9, t_{13}	4
	2	S_7	t_5, t_{11}, t_{15}	t_3, t_9, t_{13}	3
	3	S_8	t_9, t_{18}, t_{19}	t_7, t_{15}	2
	4	S_{10}	t_3	t_1	1
	5	S_{11}	t_9, t_{18}	t_7, t_{16}	1
Algorithm 9	1		t_5, t_{11}, t_{15}	t_3, t_9, t_{13}	3
	2		$2t_9, t_{19}, 2t_{18}$	$2t_7, t_{15}, t_{16}$	3
	3		t_3	t_1	1
Algorithm 10	1		$t_5, 5t_{10}, t_{11}, 6t_{18}, 3t_{19}$	$t_4, 6t_7, t_{14}, 2t_{15}, 3t_{16}$	11
	2		$t_2, 11t_4, t_5, t_8, t_{10}, t_{11}, 4t_{14}, t_{15}, t_{16}, t_{17}$	$11t_1, 2t_3, t_7, 2t_9, 6t_{13}$	20
	3		t_5, t_{11}, t_{15}	t_4, t_9, t_{13}	3

The figures indicate that both applied Algorithms 9 and 10 have the same utilization for all resources and have better results than Algorithms 2 and 8 at M1, M2, M4, R1, and R2. Algorithm 2 obtains utilization better than Algorithms 8, 9, and 10 at M3 and R3. For the throughput, Algorithm 2 obtains greater produced number of part C than Algorithms 8, 9, and 10, while Algorithms 9 and 10 have greater produced number of part A than Algorithms 2 and 8. Moreover, Algorithm 8 obtains a greater produced number of part B than Algorithms 2, 9, and 10. For case study 3, Figures 14 and 15 show the resources utilization and throughput of case study 3, respectively. The figures indicate that both applied Algorithms 9 and 10 have the same utilization of all resources

and obtain better results than Algorithms 2 and 8 at M1, M3, and R2. Both Algorithms 2 and 8 obtain the same utilization for all resources and provide utilization better than Algorithms 9 and 10 at M2, M4, and R1. For the throughput, Algorithms 2 and 8 have greater produced number of part B than Algorithms 9 and 10, while Algorithms 9 and 10 obtain greater produced number of part A than Algorithms 2 and 8. Finally, for case study 4, Figures 16 and 17 show the resources utilization and throughput of case study 4, respectively. From the figures, it can be found that both applied Algorithms 9 and 10 obtain the same utilization of all resources and obtain better results than Algorithms 2 and 8 at M1, M2, R1, and R2. Both Algorithms 2 and 8 obtain the same utilization of

TABLE 9: Required monitors using applied algorithms for case study 2.

Algorithm	i	Siphon	$\bullet V_{Si}$	$V_{Si} \bullet$	$M_{oA}(V_{Si})$
Algorithm 2	1	S_1	$t_5, t_{10}, t_{13}, t_{19}$	t_1, t_{11}, t_{15}	10
	2	S_2	$t_5, t_{10}, t_{13}, t_{18}$	t_3, t_7, t_{11}, t_{15}	7
	3	S_3	$t_5, t_{10}, t_{13}, t_{17}$	t_3, t_8, t_{11}, t_{15}	5
	4	S_4	t_2, t_{10}, t_{16}	t_9, t_{15}	2
	5	S_5	t_4, t_9, t_{13}, t_{19}	t_1, t_{11}, t_{16}	9
	6	S_6	t_4, t_9, t_{13}, t_{18}	t_3, t_7, t_{11}, t_{16}	6
	7	S_7	t_4, t_9, t_{13}, t_{17}	t_3, t_8, t_{11}, t_{16}	4
	8	S_8	t_4, t_8, t_{13}, t_{19}	t_1, t_{11}, t_{17}	7
	9	S_9	t_4, t_8, t_{13}, t_{18}	t_3, t_7, t_{11}, t_{17}	4
	10	S_{10}	t_4, t_7, t_{13}	t_3, t_{11}	2
	11	S_{11}	t_3, t_{10}, t_{19}	t_1, t_{15}	8
	12	S_{12}	t_2, t_{10}, t_{18}	t_7, t_{15}	5
	13	S_{13}	t_2, t_{10}, t_{17}	t_8, t_{15}	3
	14	S_{14}	t_3, t_8, t_{19}	t_1, t_{17}	5
	15	S_{15}	t_2, t_8, t_{18}	t_7, t_{17}	2
	16	S_{16}	t_3, t_9, t_{19}	t_1, t_{15}	7
	17	S_{17}	t_2, t_9, t_{18}	t_7, t_{16}	4
	18	S_{18}	t_2, t_9, t_{17}	t_8, t_{16}	2
Algorithm 8	1	S_1	$t_5, t_{10}, t_{13}, t_{19}$	t_1, t_{11}, t_{15}	10
	2	S_3	$t_5, t_{10}, t_{13}, t_{17}$	t_3, t_8, t_{11}, t_{15}	5
	3	S_4	t_2, t_{10}, t_{16}	t_9, t_{15}	1
	4	S_{10}	t_4, t_7, t_{13}	t_3, t_{11}	2
	5	S_{11}	t_3, t_{10}, t_{19}	t_1, t_{15}	8
	6	S_{14}	t_3, t_8, t_{19}	t_1, t_{17}	5
	7	S_{15}	t_2, t_8, t_{18}	t_7, t_{17}	1
	8	S_{17}	t_2, t_9, t_{18}	t_7, t_{16}	4
	9	S_{18}	t_2, t_9, t_{17}	t_8, t_{16}	1
Algorithm 9	1		$17t_5, 50t_{10}, 15t_{13}, 49t_{17}, 2t_{19}$	$3t_1, 14t_3, 44t_8, 3t_9, 15t_{11}, 50t_{15}, t_{18}$	196
	2		$t_5, 24t_8, 3t_{10}, t_{13}, 18t_{18}$	$t_3, 27t_7, t_{11}, 6t_{15}, 3t_{16}, 9t_{17}$	71
	3		$6t_3, t_5, 5t_8, t_9, 6t_{19}$	$6t_1, t_4, 5t_{15}, 4t_{17}, t_{18}$	34
	4		t_4, t_{13}	t_3, t_{11}	2
	5		t_9, t_{17}	t_8, t_{16}	2
	6		t_{10}, t_{16}	t_9, t_{15}	2
Algorithm 10	1		$9t_5, 50t_8, 25t_{10}, 8t_{13}, 74t_{17}, t_{19}$	$2t_1, 7t_3, 73t_7, 8t_{11}, 50t_{15}, 25t_{16}$	196
	2		$7t_5, 20t_9, 19t_{10}, 6t_{13}, 39t_{17}, t_{19}$	$t_1, 6t_3, 38t_8, 6t_{11}, 20t_{15}, 19t_{16}, t_{18}$	71
	3		$7t_5, t_8, 27t_{10}, 6t_{13}, 19t_{16}, 8t_{17}, t_{19}$	$t_1, 6t_3, 27t_9, 6t_{11}, 27t_{15}, t_{18}$	34
	4		t_4, t_{13}	t_3, t_{11}	2
	5		$6t_3, t_5, 23t_8, t_{10}, 18t_{18}, 5t_{19}$	$6t_1, t_4, 17t_7, t_9, t_{15}, 22t_{17}$	2
	6		t_3, t_7, t_{19}	t_1, t_{17}	2

all resources and obtain utilization better than Algorithms 9 and 10 at M3 and M4. For the throughput, Algorithms 9 and 10 obtain greater produced number of parts A and B than Algorithms 2 and 8.

Based on the above computational results for all case studies, it is shown that four applied deadlock prevention policies may not imply high resource utilization and plant productivity which has been shown theoretically in previous publications, since the utilization of any resource at case studies does not exceed 50% and there are several resources that have utilization less than 20%. Moreover, it is found that the

resource utilization and plant productivity are different for case studies by using applied policies. The main reason of the occurrence of this difference is a conflict situation in places. When a control place enables more than one transition at the same time, only one transition can fire. For instance, in case study 1, the number of control places using Algorithm 2 are 11 and more than one transition is enabled at the same time; that is, several transitions can be fired. While Algorithm 10 reduces the control places into 3 places and enables more than one transition at the same time, few transitions can be fired compared with Algorithm 2. Therefore, the performance

TABLE 10: Required monitors using applied algorithms for case study 3.

Algorithm	i	Siphon	$\bullet V_{Si}$	$V_{Si} \bullet$	$M_{oA}(V_{Si})$
Algorithm 2	1	S_1	t_7, t_{14}, t_{15}	t_1, t_9	6
	2	S_2	t_6, t_{14}, t_{15}	t_1, t_9	5
	3	S_3	t_7, t_{11}, t_{14}	t_1, t_9, t_{12}	5
	4	S_4	t_6, t_{11}, t_{14}	t_1, t_9, t_{12}	4
	5	S_5	t_4, t_5, t_{14}, t_{15}	t_1, t_9	3
	6	S_6	t_4, t_5, t_{14}	t_1, t_{12}	2
	7	S_7	t_7, t_{14}, t_{15}	t_1, t_9	5
	8	S_8	t_7, t_{11}	t_1, t_9	3
	9	S_9	t_7, t_{10}	t_6, t_9	2
	10	S_{10}	t_6, t_{14}, t_{15}	t_1, t_9	4
	11	S_{11}	t_6, t_{11}	t_4, t_5, t_9	2
	12	S_{12}	t_4, t_{14}, t_{15}	t_2, t_{11}	2
Algorithm 8	1	S_5	t_4, t_5, t_{14}, t_{15}	t_1, t_9	3
	2	S_6	t_4, t_5, t_{14}	t_1, t_{12}	2
	3	S_9	t_7, t_{10}	t_6, t_9	2
	4	S_{11}	t_6, t_{11}	t_4, t_5, t_9	2
	5	S_{12}	t_4, t_{14}, t_{15}	t_2, t_{11}	2
Algorithm 9	1		t_4, t_{14}, t_{15}	t_2, t_{11}	2
	2		t_6, t_{14}	t_1, t_{12}	2
	3		$2t_7, t_{10}, t_{11}$	$t_4, t_5, t_6, 2t_9$	5
	4		t_6, t_{11}	t_4, t_5, t_{10}	2
Algorithm 10	1		t_4, t_5, t_{14}	t_1, t_{12}	2
	2		$t_6, t_7, 2t_{11}$	$2t_4, 2t_5, t_9, t_{10}$	5
	3		t_4, t_{14}, t_{15}	t_2, t_{11}	2
	4		t_7, t_{10}	t_6, t_9	2

TABLE 11: Required monitors using applied algorithms for case study 4.

Algorithm	i	Siphon	$\bullet V_{Si}$	$V_{Si} \bullet$	$M_{oA}(V_{Si})$
Algorithm 2	1	S_1	t_7, t_{13}	t_1, t_9	5
	2	S_2	t_4, t_5, t_{13}	t_1, t_{11}	2
	3	S_3	t_3, t_7, t_{13}	t_1, t_9	4
	4	S_4	t_7, t_{11}	t_1, t_9	3
	5	S_5	t_4, t_{13}	t_2, t_{11}	1
Algorithm 8	1	S_2	t_4, t_5, t_{13}	t_1, t_{11}	2
	2	S_4	t_7, t_{11}	t_1, t_9	3
	3	S_5	t_4, t_{13}	t_2, t_{11}	1
Algorithm 9	1		$3t_4, 5t_6, t_{12}, 7t_{13}$	$4t_1, 4t_2, t_5, t_9, 7t_{11}$	14
	2		$2t_7, 3t_{11}$	$t_1, t_2, t_5, 3t_9$	9
Algorithm 10	1		$t_4, 2t_4, 2t_{13}$	$t_1, t_{12}, 2t_{11}$	3
	2		$2t_7, 3t_{11}$	$t_1, t_2, t_5, 3t_9$	9

is different for different cases by using applied policies. However, we can introduce the comparison between applied policies based on the above computational results, from the viewpoint of behavior permissiveness. Both Algorithms 9

Utilization comparison

FIGURE 10: Utilization of resources for the Petri net model in case study 1.

Throughput comparison

FIGURE 11: Throughput of the system for the Petri net model in case study 1.

Utilization comparison

FIGURE 12: Utilization of resources for the Petri net model in case study 2.

and 10 can provide better behavioral permissiveness than Algorithms 2 and 8 for small size systems (case studies 3 and 4). For the large systems size (case studies 1 and 2), Algorithm 2 leads to better behavioral permissiveness than the other algorithms. The supervisor performance of applied algorithms such as the number of monitors, arcs, and reachable states as averages values is summarized. Figures 18, 19, and 20 show the summarized supervisor performance of all algorithms. From the figures, both Algorithms 9 and 10 obtain a supervisor with structural complexity and computational complexity compared to Algorithms 2 and 8.

FIGURE 13: Throughput of the system for the Petri net model in case study 2.

FIGURE 14: Utilization of resources for the Petri net model in case study 3.

FIGURE 15: Throughput of the system for the Petri net model in case study 3.

7. Conclusion and Future Work

This paper presents four applied deadlock prevention policies, two and two of which are iterative methods and siphon control methods, respectively. Four applied deadlock control methods are used to design a liveness-enforcing supervisor for four automated manufacturing system case studies. The controlled case studies are modeled and simulated using

FIGURE 16: Utilization of resources for the Petri net model in case study 4.

FIGURE 17: Throughput of the system for the Petri net model in case study 4.

FIGURE 18: Average of monitors for all cases.

MATLAB software (Petri net toolbox) to analyze and evaluate the performance of selected methods such as utilization of resources and throughput; the computational results can be summarized as follows.

The applied deadlock prevention policies may not imply high resource utilization and plant productivity which has been shown theoretically in previous publications. Moreover, it is found that the resource utilization and plant productivity

FIGURE 19: Average of arcs for all cases.

FIGURE 20: Average of reachable states for all cases.

are different for case studies by using applied policies. The main reason for this difference is a conflict situation in places. When a control place enables more than one transition at the same time, only one transition can fire. For instance, in case study 1, the number of control places using Algorithm 2 are 11 and more than one transition is enabled at the same time; that is, several transitions can be fired. While Algorithm 10 reduces the control places into 3 places and enables more than one transition at the same time, few transitions can be fired compared with Algorithm 2. Therefore, the performance is different for different cases by using applied policies. However, for all automated manufacturing systems, with respect to structural complexity and computational complexity, the selected iterative methods can always lead to structurally complicated and computationally expensive liveness-enforcing net supervisors compared to the siphon control methods. From the aspect of behavioral permissiveness, the selected iterative methods can provide better behavioral permissiveness than siphon control methods for small size systems. For large systems size, strict minimal siphons control method leads to better behavioral permissiveness than the other methods.

Real-world AMSs can suffer from impacts of unreliable resources (resources with failures) under the supervisory control of deadlocks. Therefore, the need for new effective deadlock control policies based on Petri nets becomes

increasingly more important. Thus, the future research direction involves designing new effective deadlock prevention policies for automated manufacturing systems with unreliable resources [8]. Then, a simulation will be used to measure the performance of these policies. Different and complex automated manufacturing systems are required and should be considered as a benchmark for these deadlock prevention policies under a variety of control specifications. Optimal deadlock-free scheduling with high time performance will also be interesting.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was funded by the National Plan for Science, Technology and Innovation (MARRIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, Award no. 12-ELE2506-02.

References

- [1] Y. Chen, Z. Li, M. Khalgui, and O. Mosbahi, "Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 2, pp. 374–393, 2011.
- [2] R. A. Wysk, N. S. Yang, and S. Joshi, "Detection of deadlocks in flexible manufacturing cells," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 853–859, 1991.
- [3] D. Y. Chao, "Direct minimal empty siphon computation using MIP," *The International Journal of Advanced Manufacturing Technology*, vol. 45, no. 3–4, pp. 397–405, 2009.
- [4] D. Y. Chao, "Improvement of suboptimal siphon-and FBM-based control model of a well-known," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 2, pp. 404–411, 2011.
- [5] A. Ghaffari, N. Rezg, and X. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 137–141, 2003.
- [6] M. Uzam and M. Zhou, "Iterative synthesis of petri net based deadlock prevention policy for flexible manufacturing systems," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, pp. 4260–4265, October 2004.
- [7] M. Uzam, "The use of the Petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems," *The International Journal of Advanced Manufacturing Technology*, vol. 23, no. 3–4, pp. 204–219, 2004.
- [8] K. Lautenbach, "Linear algebraic calculation of deadlocks and traps," in *Concurrency and Nets*, pp. 315–336, Springer, Berlin, Germany, 1987.
- [9] M. Uzam, "An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions," *The International Journal of Advanced Manufacturing Technology*, vol. 19, no. 3, pp. 192–208, 2002.

- [10] D. Y. Chao, "Fewer monitors and more efficient controllability for deadlock control in S^3PGR^2 (systems of simple sequential processes with general resource requirements)," *The Computer Journal*, vol. 53, no. 10, pp. 1783–1798, 2010.
- [11] Z. Li and M. Zhou, "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 34, no. 1, pp. 38–51, 2004.
- [12] Z. Li, G. Liu, H.-M. Hanisch, and M. Zhou, "Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 42, no. 1, pp. 178–191, 2012.
- [13] 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.
- [14] 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.
- [15] Y. S. Huang, M. Jeng, X. Xie, and D.-H. Chung, "Siphon-based deadlock prevention policy for flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 36, no. 6, pp. 1248–1256, 2006.
- [16] Z. Li and N. Wei, "Deadlock control of flexible manufacturing systems via invariant—controlled elementary siphons of petri nets," *The International Journal of Advanced Manufacturing Technology*, vol. 33, no. 1-2, pp. 24–35, 2007.
- [17] Y. S. Huang, "Design of deadlock prevention supervisors using Petri nets," *The International Journal of Advanced Manufacturing Technology*, vol. 35, no. 3-4, pp. 349–362, 2007.
- [18] G. Y. Liu, Z. W. Li, K. Barkaoui, and A. M. Al-Ahmari, "Robustness of deadlock control for a class of Petri nets with unreliable resources," *Information Sciences*, vol. 235, pp. 259–279, 2013.
- [19] S. Wang, M. Zhou, and W. Wu, "Design of a maximally permissive liveness-enforcing supervisor with reduced complexity for automated manufacturing systems," *Asian Journal of Control*, vol. 17, no. 1, pp. 190–201, 2014.
- [20] N. Viswanadham, Y. Narahari, and T. L. Johnson, "Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models," *IEEE Transactions on Robotics & Automation*, vol. 6, no. 6, pp. 713–723, 1990.
- [21] M. Uzam, Z. Li, and M. Zhou, "Identification and elimination of redundant control places in petri net based liveness enforcing supervisors of FMS," *The International Journal of Advanced Manufacturing Technology*, vol. 35, no. 1-2, pp. 150–168, 2007.
- [22] Z. Li and H. Hu, "On systematic methods to remove redundant monitors from liveness-enforcing net supervisors," *Computers & Industrial Engineering*, vol. 56, no. 1, pp. 53–62, 2009.
- [23] F. Garcia-Valles and J. Colom, "Implicit places in net systems," in *Proceedings of the 8th International Workshop on Petri Nets and Performance Models*, pp. 104–113, Zaragoza, Spain, 1999.
- [24] L. Recalde, E. Teruel, and M. Silva, "Improving the decision power of rank theorems," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation*, pp. 3768–3773, October 1997.
- [25] Y. Chen, Z. Li, and M. Zhou, "Behaviorally optimal and structurally simple liveness-enforcing supervisors of flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 42, no. 3, pp. 615–629, 2012.
- [26] L. Piroddi, R. Cordone, and I. Fumagalli, "Selective siphon control for deadlock prevention in Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 38, no. 6, pp. 1337–1348, 2008.
- [27] 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 Part A: Systems and Humans*, vol. 39, no. 3, pp. 650–661, 2009.