Proceedings of the 2015 International Conference on Industrial Engineering and Operations Management
Dubai, UAE, March 3 – 5, 2015

Evaluation of deadlock control designs in automated manufacturing systems

Abdulaziz M. El-Tamimi[1,a], Emad Abouel Nasr[2,b] , Abdulrahman Al-Ahmari[1,c], Zhiwu Li[3,d], and Husam Kaid[1,e]

[1]College of Engineering, Industrial Engineering Department, King Saud University, Riyadh 11421, Saudi Arabia
[2]College of Engineering, Industrial Engineering Department, King Saud University, Riyadh 11421, Saudi Arabia.
And Mechanical Engineering Department, Helwan University, Cairo, Egypt
[3]School of Electro-Mechanical Engineering, Xidian University, No. 2 South Taibai Road, Xi'an 710071, China

[a]atamimi@ksu.edu.sa, [b]eabdelghany@ksu.edu.sa, [c]alahmari@ksu.edu.sa, [d]systemscontrol@gmail.com,
[e]yemenhussam@yahoo.com

***Abstract*—Petri nets are an effective way to model, analyze, and control deadlocks in automated manufacturing systems (AMS). There are three important criteria in designing and evaluating a liveness-enforcing supervisor for a system to be controlled: behavioral permissiveness, structural complexity, and computational complexity. A maximally permissive supervisor can lead to high utilization of system resources. A supervisor with a simple structure can decrease the hardware and software costs. As for the computational complexity, means that a deadlock control policy can be applied to large systems. The objective of this paper is to design liveness-enforcing supervisors for different flexible manufacturing systems, simulate the controlled systems, and estimate the utilization of resources and throughput of the system. The siphon control methods (Strict Minimal Siphons and Elementary Siphons) are used to solve the deadlock control problems for a number of AMSs with different sizes. Moreover, the paper aims to evaluate the performance of selected methods such as utilization of resources, throughput, and the number of monitors, arcs, and states. Finally, the computational results indicate that the elementary siphons based policy provides better criteria for designing and evaluation of the systems than the strict minimal siphons based policy.**

Keywords*— *Deadlock prevention, automated manufacturing system, Petri net, Siphon, Simulation, Matlab software.

I. INTRODUCTION

Manufacturers must adapt to changes in the production environment as well as in the market in order to achieve and maintain competitiveness. Effectively designing and operating an automated manufacturing system (AMS) can be of assistance to manufacturers in reaching this goal. In an AMS, different types of parts enter the system at discrete points of time and are processed concurrently; these parts have to share some common resources, such as machine tools, robots, buffers, automated guided vehicles (AGVs), and fixtures. This competition for shared resources can cause deadlocks in an AMS during its operation, which are highly undesirable phenomena that often cause low use of some critical and expensive resources, long downtime, and may lead to catastrophic results in highly automated manufacturing systems. Therefore, it is required to develop an effective AMS control policy to make sure that deadlocks will never occur in AMS.

Petri net is a major mathematical tool to model, analyze, and control deadlocks in resource allocation systems including AMS [1]. Moreover, the Petri net can be used to describe the behavior of AFMS and develop appropriate deadlock resolution methods. To overcome the deadlocks in AMS, there are three major applied strategies which are deadlock detection and recovery, deadlock avoidance, and deadlock prevention [2]. In AMS environment, usually achieving deadlock prevention is either by an off-line mechanism or by using effective system design to control the requests for resources to ensure that deadlocks never occur [3].

There are three important criteria in designing and evaluating a liveness-enforcing supervisor for a system to be controlled: behavioral permissiveness, structural complexity, and computational complexity. A maximally permissive supervisor can lead to high utilization of system resources. A supervisor with a simple structure can decrease the hardware and software costs. As for the computational complexity, means that a deadlock control policy can be applied to large systems. Thus, many researchers try their best to develop deadlock prevention algorithms that can obtain liveness-enforcing supervisors with maximal permissiveness, a simple supervisory structure, and low computational complexity [4].

In general, there are mainly two Petri net analysis techniques for deadlock prevention in AMSs: structural analysis [5],[6] and reachability graph analysis [7],[8],[9]. Structural analysis often applied in terms of structural objects of Petri nets, such as siphons and resource transition circuits. In this case, control laws are often simple, each possible empty minimal siphon requires a monitor to be added to prevent itself from being emptied, but the resulting controlled system is suboptimal in terms of behavioral permissiveness in general and the number of control places is linearly dependent on the size of a net [10]. Reachability graph analysis requires an enumeration of all or a part of reachable markings which always suffers from a state explosion problem. The reachability graph of a net is classified into two parts: live zone (LZ) and deadlock zone (DZ). First met bad markings (FBMs) are first derived from the reachability graph. An FBM is a marking in DZ, representing the very first entry from LZ to DZ. A control place is designed to prevent the FBM from being reached and an FBM should be singled out. These requires iterations to forbid all FBMs. This process cannot terminate until the resulting net is live [11]. There are several methods and algorithms used for deadlocks prevention; some of them are siphon control method, theory of region and iterative methods [8], [11], [12], [13],[14]. Most of researchers use and develop these methods to design liveness-enforcing supervisors with maximal permissiveness, a simple supervisory structure, and low computational complexity. Unfortunately, for maximal permissiveness, they do not use simulation to test the utilization of resources and throughput of plant model.

The objective of this paper is to explore whether a permissive liveness-enforcing Petri net supervisor can definitely lead to better time performance using a simulation based on a large number and a large variety of Petri net models and exiting deadlock prevention policies. The simulation work will be conducted in the following steps. (1) Assign the time to the Petri net models in a variety of ways, which leads to time Petri nets, timed Petri nets, and stochastic Petri nets. (2) Built the Petri net model using Matlab software (Petri Net Toolbox (PN Toolbox)). (3) Run and simulate the model, the time performance of the system can be obtained. (4) Finally, the simulation results will be analyzed to determine which deadlock prevention policies are suitable for different classes of production systems. The siphon control methods (Strict Minimal Siphons and Elementary Siphons) will be used for deadlocks prevention. Moreover, this paper aims to evaluate the performance of the selected methods such as utilization of resources, plant throughput, and number of monitors, number of arcs, and number of states. Including this introductory section, the paper is organized as follows. Section II provide literature review for previous research work related to deadlock prevention. The basics of Petri nets are discussed in Section III. The deadlock prevention methods and policies are presented in Section IV. Section V shows the analysis of the selected methods for two different AFM systems. Finally, Section VI presents the conclusion.

II. Literature Review

In this section, the researches in siphons are summarized, although, the methods used to solve siphons problems. Ezpeleta et al. [15] used special class of Petri Nets that called System of Simple Sequential Processes with Resources (S^3PR) and proposed a policy for resource allocation in FMS based on the addition of new places to the net imposing restrictions that prevent the presence of unmarked siphons. The work of Huang et al. [16] presents a new deadlock prevention algorithm for the class Petri nets, where deadlocks are related to unmarked siphons. Two kinds of control places are added to the original model for FMS called ordinary control place and weighted control place to prevent siphons from being unmarked. An FMS example is investigated by Li and Zhou [3] with its Petri net model having 26 places, 20 transitions, and 18 strict minimal siphons. By using the concept of elementary siphons, a liveness-enforcing Petri net supervisor is computed by explicitly adding only six monitors to control the six elementary siphons among the 18 strict minimal siphons. However, the approach described by Ezpeleta et al. [15] requires the design of monitors for all 18 strict minimal siphons. Huang and Jeng [17] proposed siphon-based algorithm for deadlock prevention of a class of Petri nets. Two kinds of control places are added to the original model for FMS called ordinary control place and weighted control place to prevent siphons from being unmarked. The work of Li and Zhou [18] develops a two-phase deadlock prevention policy. The first phase adds a monitor for each elementary siphon derived from the mixed integer programming (MIP) -based deadlock detection method. The output arcs of a monitor lead to the source transitions of a plant net model, which represents the entry of raw parts into the system. The second phase rearranges the output arcs of the monitors such that they, as much as possible, lead to non-source transitions if this rearrangement does not result in dead transitions. Such an improvement increases the behavioral permissiveness of the supervisor.

Huang [19] proposed a methodology to synthesize supervisors for a class of sequential resource allocation system for flexible manufacturing systems. The type of Petri nets are called S3PR, where deadlocks are related to emptied siphons. Since all minimal siphons should be controlled. In this study, a concept of the elementary siphon is used to reduce the number of control places. Based on P-invariants and elementary siphons of Petri nets, a deadlock prevention policy is developed by Li and Wei [20] for a special class of Petri nets that can well model many FMS. Siphons in a plant net model are divided into elementary and dependent ones. For each elementary siphon, a monitor is added to the plant model such that the siphon is invariant-controlled. When all elementary siphons are controlled, the controllability of a dependent siphon is ensured by properly setting the control depth variables of its related elementary siphons. In the work of Li et al. [21], a mixed integer programming (MlP)-based deadlock detection technique is used to find some, in general not all, minimal siphons in a plant model without complete siphon enumeration. For each siphon found, depending on its non-controllability, a monitor is added such that it is invariant-controlled. Their siphon control method guarantees that no emptiable control-induced siphon is generated due to the addition of the monitors. The siphon control process proceeds iteratively until there is no unmarked siphon in the supervisor of a plant model. They found that compared with the work of Ezpeleta et al. [15], the novel deadlock prevention policy can usually lead to a structurally simple liveness-enforcing supervisor by adding only a small number of monitors and arcs. However, this work does not improve the behavioral permissiveness. Li et al. [22]presented a novel deadlock prevention policy for Petri nets that model flexible manufacturing systems using siphon extraction. At each iteration, a siphon extraction algorithm finds a maximal deadly marked siphon, classifies the places in it, and decides a necessary siphon from the classified places. Accordingly, the deadlock prevention policy adds a proper control place (CP) to make each necessary siphon marked or max-controlled until the controlled system is live.

A variety of deadlock control policies based on Petri nets have been proposed for automated manufacturing systems (AMSs) with unreliable resources by Liu et al. [23], recovery subnets and monitors are designed for unreliable resources and strict minimal siphons that may be emptied, respectively, normal and inhibitor arcs are used to connect monitors with recovery subnets in case of necessity. Hou et al. [24] investigated the structure of a class of generalized Petri nets, WS3PR, to compute all strict minimal siphons (SMS) and a compact set of elementary siphons. In order to reduce structural complexity of a supervisor, a set of elementary siphons derived from all strict minimal siphons (SMS) is explicitly controlled. The study of Wang et al. [25] deals with the problems of computational and structural complexity in designing maximally permissive liveness-enforcing supervisors for a class of Petri nets called (S3PR) without ξ-resources. They used two steps to obtain supervisor for the model, the first step, proposed an algorithm to extract a desired emptied strict minimal siphon (SMS) from a given emptied siphon based on loop resource subsets. The second step, proposed a siphon-based deadlock prevention policy, which can obtain a maximally permissive liveness-enforcing supervisor with reduced structural complexity and no weighted monitors.

III. Basics of Petri nets

Petri nets are a mathematical and graphical modeling tool applicable to many systems. It is a major tool for studying and describing information processing systems that are characterized as being asynchronous, concurrent, parallel, distributed, nondeterministic, and/or stochastic. A Petri net or place/transition net N is a four-tuple (P, T, F,W) where P and T are finite, non-empty sets, and disjoint sets. P is a set of places, and T is a set of transitions, elements belonging to $P \cup T$ are called nodes $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation of the net or the set of directed arcs, represented by arcs with arrows from places to transitions or from transitions to places. W: $(P \times T) \cup (T \times P) \rightarrow \mathbf{N}$ 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 is the set of non-negative integers. N = (P, T, F,W) is called an ordinary net, denoted as N = (P, T,F), if $\forall f \in F$, $W(f) = 1$. Given a node $x \in P \cup T$, $\bullet x = \{y \in P \cup T | (y, x) \in F\}$ is called the preset of x, while $x\bullet = \{y \in P \cup T | (x, y) \in F\}$ is called the post set of x. A marking is a mapping $M:P \rightarrow N$. M(p) denotes the number of tokens in place p. We usually describe markings and vectors using a multiset or formal sum for economy of space. As a result, $\sum_{p \in P} M(p)p$ is used to denote vector M. For instance, a marking that puts two tokens in place p_2, fife tokens in place p_4, and no tokens in other places in a net is denoted as $2p_2 + 5p_4$. The pair (N,M_0) is called a marked Petri net or a net system. Sometimes we use $N = (P, T, F,W, M_0)$ to denote a net system. A net is pure (self loop free) if $\forall x, y \in P \cup T$, $W(x, y) > 0$ implies $W(y, x) = 0$. Incidence matrix [N] of net N is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$. A transition $t \in T$ is enabled at marking M if $\forall p \in {}^{\bullet}t$, $M(p) \geq W(p, t)$. This fact is denoted as $M[t\rangle$. Once a transition t fires, it yields a new marking M′, which is obtained by removing W(p, t) tokens from each place $p \in {}^{\bullet}t$, and placing W(t, p) tokens in each place $p \in t^{\bullet}$, denoted as $M[t\rangle M'$, where $M'(p) = M(p) - W(p, t) + W(t, p)$. M′ is reachable from M if there exists a firable transition sequence $\delta = t_1, t_2, \ldots t_n$ and markings $M_1, M_2, \ldots$, and M_{n-1} such that $M[t_1\rangle M1[t_2\rangle M_2 \ldots M_{n-1}[t_n\rangle M'$ holds, which is denoted as $M[\delta\rangle M'$ and satisfies the state equation $M' = M + [N]\vec{\delta}$, where $\vec{\delta} : T \rightarrow N$ is a vector of non-negative integers, called a counting vector, and $\vec{\delta}(t)$ indicates the algebraic sum of all occurrences of t in δ. $M[\rangle$is the set of all markings reachable from M by firing any possible sequence of transitions. $M_0[\rangle$ is called the set of reachable markings of a Petri net N from initial marking M_0, denoted by $R(N,M_0)$. The reachable markings $R(N,M_0)$ can be graphically expressed by a reachability graph. The reachability graph of a net (N,M_0), denoted as $G(N,M_0)$, is a directed graph whose nodes are markings in the reachable markings $R(N,M_0)$, and arcs are labeled by the transitions of N. An arc from M_2 to M_3 is labeled by t_1 if $M_2[t_1\rangle M_3$. A marked Petri nets N with initial marking M_0 is said to be k-bounded if $\forall M \in R(N,M_0)$: $M(p) \leq k$ $(k \in N \setminus \{0\})$. A net is k-bounded if every place is k-bounded. A net is bounded if it is k-bounded for some k. A net is safe if it is 1-bounded. Let (N,M_0) be a net system with N = (P, T, F,W). A transition $t \in T$ is live at M_0 if $\forall M \in R(N,M_0)$, $\exists M \in R(N,M)$, $M'[t\rangle$. (N,M_0) is live if $\forall t \in T$, t is live at M_0. It is dead at M_0 if $\nexists\, t \in T$, $M_0[t\rangle$. A P-vector is a column vector $I : P \rightarrow Z$ indexed by P, and a T-vector is a column vector $J : T \rightarrow Z$ indexed by T, where Z is the set of integers. P-vector I is called a P-invariant (place invariant, PI for short) if $I \neq 0$ and $I^T.[N] = 0^T$. T-vector J is called a T-invariant (transition invariant) if $J \neq 0$ and $[N].J = 0$. P-vector I is a P-semiflow if every element of I is non-negative. $||I|| = \{p \,|I(p) \neq 0\}$ is called the support of I. I is called a minimal P-invariant if $||I||$ is not a superset of the support of any other one and its components are mutually prime. l_i's are called the coefficients of I if $\forall p_i \in P$, $l_i = I(p_i)$. Let I be a P-invariant of (N,M_0) and M be a reachable marking from M_0. Then, $I^T M = I^T M_0$.

IV. Deadlock Prevention Methods and Policies

A. *Strict Minimal Siphons (SMS) Control Method*

A siphon in a Petri net is a set of places $S = \{p_1, \ldots, p_k\}$. Let $S_P = 2^P$ be the set of all subsets of the places representing siphons of a Petri net with |P| places.

Definition 1. A non-empty set $S \subseteq P$ is a siphon if ${}^{\bullet}S \subseteq S^{\bullet}$. $S \subseteq P$ is a trap if $S^{\bullet} \subseteq {}^{\bullet}S$. A siphon (trap) is minimal if there is no siphon (trap) contained in it as a proper subset. A minimal siphon S is said to be strict (SMS) if ${}^{\bullet}S \subsetneq S^{\bullet}$.

Property 1. Let S_1 and S_2 be two siphons (traps). Then, $S_1 \cup S_2$ is a siphon (trap).

Property 2. Let $M \in R(N,M_0)$ be a marking of net (N,M_0) and S a siphon. If $M(S) = 0$, then $\forall M' \in R(N,M)$, $M'(S) = 0$.

Property 3. Let $M \in R(N,M_0)$ be a marking of net (N,M_0) and S a trap. If $M(S) > 0$, then $\forall M' \in R(N,M)$, $M'(S) > 0$.

Property 2 indicates that once a siphon loses all its tokens, it remains unmarked at any subsequent markings that are reachable from the current marking. Property 3 implies that once a trap is marked at a marking, it is always marked at the subsequent markings that are reachable from the current one. A siphon provides a further negative reachability test. If $M \in R(N,M_0)$ marks a siphon that is unmarked at M_0, then M is unreachable. In summary, once the invariant laws associated with siphons and traps become true, they remain true at any subsequently reachable markings.

Siphons and traps play an important role in the liveness analysis of a net, particularly in ordinary ones. If a siphon S in a net has no tokens, then no transition in $S^\bullet$ is enabled and all the transitions connected to S can never be enabled, i.e., the transitions are therefore dead, leading to the loss of liveness of the net. The deadlock-freedom and liveness of a Petri net are closely related to its siphons, which is shown by Desel and Esparza [26] results.

Theorem 1. Let (N,M_0) be an ordinary net and Π the set of its siphons. The net is deadlock-free if $\forall S \in \Pi$, $\forall M \in R(N,M_0)$, $M(S) > 0$.

Theorem 2. Let (N,M) be an ordinary net that is in a deadlock state. Then, $\{p \in P | M(p) = 0\}$ is a siphon.

Theorem 1 states that an ordinary Petri net is deadlock-free if no (minimal) siphon eventually becomes empty. Theorem 2 means that if an ordinary net is dead, i.e., no transition is enabled, then all the unmarked places form a siphon.

Definition 2. A siphon S is said to be controlled in an ordinary net system (N,M_0) if $\forall M \in R(N,M_0)$, $M(S)> 0$.

A siphon in an ordinary net that contains a marked trap is controlled, which does not imply a deadlock. In a generalized Petri net, however, a siphon that is always marked may lead to dead transitions.

Definition 3. Siphon S in an ordinary net system (N,M_0) is invariant controlled by P-invariant I under M_0 if $I^T M_0 > 0$ and $\forall p \in P\backslash S$, $I(p) \le 0$, or equivalently, $I^T M_0 > 0$ and $\|I\|^+ \subseteq S$.

If S is controlled by P-invariant I at M_0, S can be always marked, i.e., $\forall M \in R(N,M_0)$, S cannot be emptied at M.

Remark 1. The number of siphons (minimal siphons) grows fast with respect to the size of a Petri net and in the worst case grows exponentially with a net size.

A control place is added to every SMS such that liveness can be enforced. The method is simple and guarantees a success. However, too many control places and arcs have to be added, leading to a much more complex Petri net than the originally built one. In fact, the number of places added is equal to the number of SMS in the target net and the number of arcs added is generally much larger than that of places added, particularly for large-scale Petri nets. The SMS control based on complementary sets of siphons [S] is used to design the control places, it is the 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 P -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 emptied when all tokens marked in S flow into [S]. The following notation will be used in the establishment of the control policy.

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

Based on the concept of elementary siphons, The proposed deadlock prevention algorithm developed by Ezpeleta et al. [15] as follow:

SMS based policy (Algorithm 1)
Input: Petri net model (N,M_0) of an FMS with $N = (P^0 \cup P_A \cup P_R, T, F,W)$. Output: A controlled Petri net system (N_1,M_1). Step 1: Find all strict minimal siphons (SMS) of a given Petri net (N,M_0). Step 2: For each siphon S, add a control place Vs such that: • The output arcs (weights are all ones) of Vs are connected to the source transitions that have paths leading to the sink transitions of S. • The input arcs (weights are all ones) of Vs are connected to the transitions related to the stealing places of S. • $M_{0A}(Vs)=M_0(S)-1$, $M_0(Vs)$ is an initial marking of control place. Step 3: Repeat Step 2 until all such SMSs are considered.

For example, the Petri net model shown in Fig.1(b). It consists of one robots R1, which can hold one part at a time, two machines M1-2, each of which can process one part at a time, two loading and unloading buffers, Two parts types are considered in the system: PA and PB. PA moves to the M1 and PB moves to the M2. The robot R1 reaches the machine that finishes its operation first, grips and loads the part to the next machine 1 or 2. There are 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 $P_A = \{P_2, \ldots, P_7\}$,where P^0 is the input places, P_R is the resources 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 SMS 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 I shows the required monitors using algorithm 1 for Fig.1(b). Fig.2 illustrates the controlled system of the Petri net in Fig. 1(b) after adding control places by algorithm 1.

B. Elementary Siphons Control Method

The strict minimal siphons (SMS) in a Petri net are classified into elementary and dependent ones. In the sequel, $\prod$ is used to denote the set of strict minimal siphons, while $\prod_E$ and $\prod_D$ the sets of elementary and dependent (redundant) ones, respectively. Unless otherwise stated, we refer to a strict minimal one when mentioning a siphon.

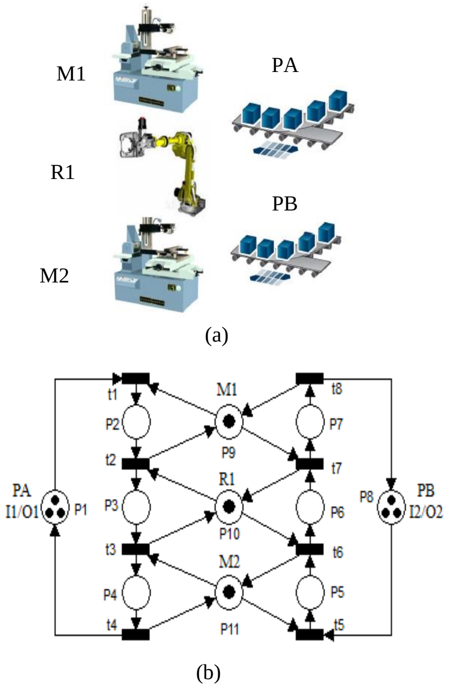


Fig. 1. (a) An AMS example (b) Petri net model of an AMS.

Definition 4. Let S⊆P be a siphon of N. P-vector λ_S is called the characteristic P-vector of S if ∀p∈S, $\lambda_S(p)=1$, otherwise $\lambda_S(p)=0$.

Definition 5. Let N= (P, T, F) be a net with $|P|$ = m, $|T|$=n and we assume N has k SMS, $S_1, S_2, \ldots$, and S_K, m, n, k ∈ℕ.

Let $\lambda_{Si}(\eta_{Si})$ be the characteristic *P(T)* - vector of siphon S_i, i ∈ {1, 2, …n}. we define $[\lambda]_{kxm} = [\lambda_{S1}, \lambda_{S2}, \ldots, \lambda_{Sk}]^T$ and $[\eta]_{kxn}$=$[\lambda]_{kxm}.[N]_{mxn}$ is called the characteristic *P(T)*- vector matrix of the siphons of N .

Proposition 1: Let be N a net and $[\eta]$ be the characteristic T - vector matrix of the siphons of it. The number of elementary siphons in N is the rank of $[\eta]$. Due to Proposition 1, it is easy to find elementary siphons in a Petri net system (N, M_0) given all siphons. First, we construct matrix $[\lambda]$ and then $[\eta]$. Then linearly independent vectors can be found in $[\eta]$. Finally, the siphons that correspond to these linearly independent vectors are the elementary siphons in the net system (N, M_0).

Definition 6. Let S_1-S_n be the siphons in net N. T-vector η_1, η_2,…, and η_n form a vector space [η]. The base of the vector space is denoted by η_B={$\eta_{B1}, \eta_{B2}, \ldots, \eta_{BK}$}, where k is the rank of the vector space [η]. Then {$S_{B1}, S_{B2}, \ldots, S_{BK}$}is called a set of elementary siphons in net N.

Definition 7. Let S∈∏\∏$_E$ be a siphon in a net N and S_1, S_2 , …,S_n ∈ ∏$_E$. S is strongly dependent on S_1 –S_n if $\eta_S = \sum_{i=1}^{n} a_i.\eta_i$ holds, where $a_i > 0$.

Definition 8. Let S∈∏\∏$_E$ be a siphon of net N and S_1, S_2, … ,S_n, Sn_{+1}, S_{n+2}, …, and S_{n+m} ∈ ∏$_E$ (n ≥ 1, m ≥ 1). S is weakly dependent on elementary siphons S_1-S_{n+m} if $\eta_i = \sum_{i=1}^{n} a_i.\eta_{S_i} - \sum_{j=n+1}^{n+m} a_j.\eta_{S_j}$ holds, where ∀i, j∈{1, 2, …, n+m}, $a_i > 0$, $a_j > 0$.

If S is dependent on S_1-S_k, we say that S_1-S_k are the elementary siphons of S. The following results and Theorems are from Li and Zhou [3].

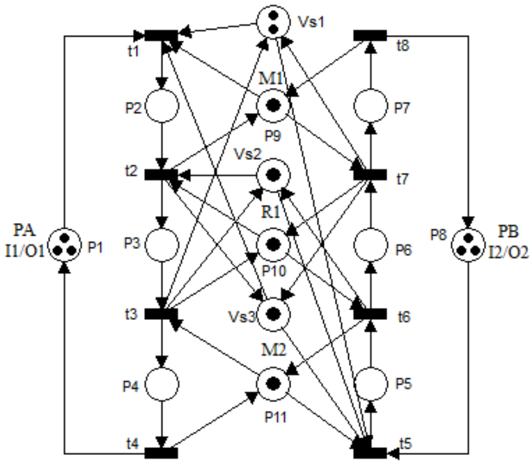


Fig. 2. Controlled system of the Petri net in Fig. 1(b) by Algorithm 1.

Theorem 3. Let N_0 be a marked net and S= {p_{i1}, p_{i2}, …, p_{in}} be a siphon in N_0 . Control place V_S is added to N_0 , the new net system is denoted as (N_1, M_1), such that: 1) I_1=(…, 1p_{i1}, 1p_{i2}, …, 1p_{in}, …, -1vs, …) ^{T} is a P-invariant of N_1, 2) M_1 (V_S)=M_0(S)-ξ_S, where ξ_S is called the control depth variable of siphon S, which indicates the least number of tokens that siphon can hold. Obviously is equal to or greater than 1 to achieve purpose of control 1≤ ξ_S ≤ M_0(S)-1, and 3) ∀p∈P_0, M_1(p)=M_0(p), where P_0 is the set of places of N_0. Then S is invariant-controlled.

Theorem 4. Let (N, M_0) be a net system and S_0 be a strictly dependent siphon with respect to elementary siphons S_1 , S_2 , …, and S_n . If S_1 , S_2 , …, and S_n are invariant controlled by adding control places V_{s1}, V_{s2}, … and V_{sn}, and $M_0(S_0) > \sum_{i=1}^{n} a_i.M_0(S_i) - \sum_{i=1}^{n} a_i.\xi_{Si}$ holds, then S_0 is invariant controlled.

Theorem 5. Let (N, M_0) be a well-initially-marked net system and S_0 be a generally dependent siphon with respect to elementary siphons S_1, S_2, …, S_n, S_{n+1}, S_{n+2}, …, and S_{n+m} which means $\eta_{S_0} = \sum_{i=1}^{n} a_i.\eta_{S_i} - \sum_{j=n+1}^{n+m} a_j.\eta_{S_j}$. If S_1 , S_2, …, S_n, S_{n+1}, S_{n+2}, …, and S_{n+m} are invariant-controlled by adding control places V_{s1}, V_{s2}, …, V_{sn+m},…, and $M_0(S_0) > \sum_{i=1}^{n} a_i.M_0(S_i) - \sum_{i=1}^{n} a_i.\xi_{Si}$, where 1≤ ξ_{Si} ≤ $M_0(S_i)$-1, S_0 is controlled.

Theorem 6. Let N= (P, T, F) be a net and ω the number of elementary siphons of N. We have ω ≤ min (|P|, |T|). This indicates that the number of elementary siphons is bounded by the smaller of place count and transition count in a Petri net. From Theorems 3 and 4, a dependent siphon can be implicitly controlled under some conditions related to its elementary siphons.

Table I: Control places computations by algorithm 1.

SMS	$\|I\|^+$	$H(r_R)$	$[S]$	$^{\bullet}V_{si}$	$V_{si}^{\bullet}$	$M_{0A}(V_{si})$
$S_1=\{P_4,P_7,P_9,P_{10},P_{11}\}$, $S_A=\{P_4,P_7\}$, $S_R=\{P_9,P_{10},P_{11}\}$.	$\|I_{p9}\|^+=\{p_2,p_7,p_9\}$, $\|I_{p10}\|^+=\{p_3,p_6,p_{10}\}$, $\|I_{p11}\|^+=\{p_4,p_5,p_{11}\}$.	$H(r_9)=\{p_2,p_7\}$, $H(r_{10})=\{p_3,p_6\}$, $H(r_{11})=\{p_4,p_5\}$.	$[S_1]=\{p_2,p_3,p_5,p_6\}$.	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

Based on the concept of elementary siphons, The proposed deadlock prevention algorithm developed by Li and Zhou [14] as follow:

Elementary siphon based policy (Algorithm 2)

Input: Petri net model (N,M_0)

Output: A controlled Petri net system (N_1,M_1).

Step 1: Find all strict minimal siphons of Petri net model N.

Step 2: T-vector matrix of the SMS $[\eta]$.

Step 3: Find the elementary siphons of N. The others are the dependent siphons.

Step 4: For each elementary siphon S, add a control place Vs such that:

- The output arcs (weights are all ones) of Vs are connected to the source transitions that have paths leading to the sink transitions of S.
- The input arcs (weights are all ones) of Vs are connected from the stealing places of S.
- Compute the initial token of place control V_S $1\leq \xi_S \leq M_0(S)-1$, $M(V_S)=M_0(S)-\xi_S$.

Step 5: Repeat Step 4 until all elementary siphons are considered.

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

For example, there are three SMS in the net shown in Fig. 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 $\lambda_{S1}=(0,0,0,1,0,0,1,0,1,1,1)^T$, $\lambda_{S2}=(0,0,0,1,0,1,0,0,0,1,1)^T$, and $\lambda_{S3}=(0,0,1,0,0,0,1,0,1,1,0)^T$. Hence, $\eta_{S1}=(-1,0,1,0,-1,0,1,0)^T$, $\eta_{S2}=(0,-1,1,0,-1,1,0,0)^T$, and $\eta_{S3}=(-1,1,0,0,0,-1,1,0)^T$. The $[\eta]$ for the Petri net in Fig. 1 is as follows. Evidently, the rank of $[\eta]$ is 2 since the first row η_{S1} can be linearly represented by the second and third rows, we can see that $\eta_{S1}=\eta_{S2}+\eta_{S3}$. Therefore, the siphons which correspond to the second and third rows S_2,S_3 are elementary siphons. Thus S_1 is a dependent siphon.

$$[\eta]=\begin{pmatrix}-1 & 0 & 1 & 0 & -1 & 0 & 1 & 0\\ 0 & -1 & 1 & 0 & -1 & 1 & 0 & 0\\ -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0\end{pmatrix}$$

By the meaning of the characteristic T-vector of a siphon, we can see that the change of token count in a dependent siphon is strongly related to the token flow in its elementary siphons. For example in Fig.1(b), S_1 is a dependent siphon and S_2,S_3 are elementary siphons. It has eight transitions, it is clear that firing t_1 decreases the number of tokens in $S_3$and S_1 and firing t_2 decreases the number of tokens in S_2, increases the number of tokens in S_3, and no change the number of tokens in S_1, also firing t_3 increases the number of tokens in S_2 and S_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 5 are shown in Table II. Table III shows the required monitors using algorithm 2 for Fig.1(b). Fig.3 illustrates the controlled system of the Petri net in Fig. 1(b) after adding control places by Algorithm 2.

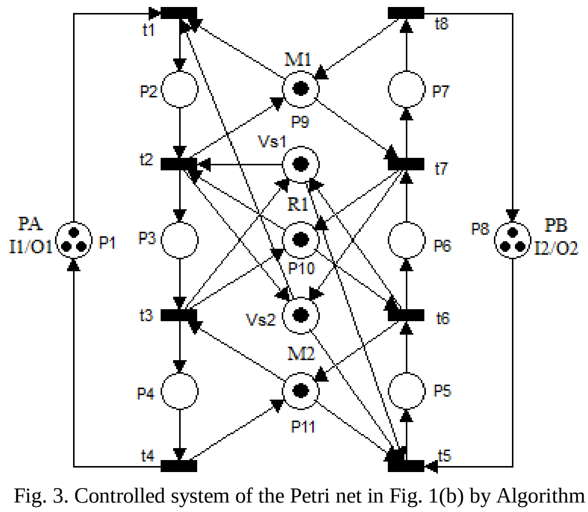


Fig. 3. Controlled system of the Petri net in Fig. 1(b) by Algorithm 2.

V. Computational Results

In this section, the computational results of the applied methods are conducted for two AMS case studies. Both cases have modeled using Matlab Software. The simulation is done for 24 time units.

Table II: Marking relationships between dependent and elementary siphons

Dependent	η relationship	Initial Marking Relationships, $M_0(S_0) > \sum_{i=1}^{n} a_i .M_0(S_i) - \sum_{i=1}^{n} a_i .\xi_{Si}$	Controlled
S_1	$\eta_1 = \eta_2 + \eta_3$	$M_0(S_1) > (M_0(S_2) + M_0(S_3)) - (\xi_{S2} + \xi_{S3})$ $3>(2+2)-(1+1)$, $3>2$	Yes

Table III: Required Monitors using algorithm 2.

Siphon	${}^{\bullet}V_{si}$	$V_{si}{}^{\bullet}$	$M_{oA}(V_{si})$
S_2	t_3,t_6	t_2,t_5	1
S_3	t_2,t_7	t_1,t_5	1

A. Case study 1

The Petri net model of the first AMS case study is shown in Fig. 4(b). It consists of two robots R1-2, each of which can hold one part at a time, fife machines M1-5, each of which can process one part at a time, 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 production sequences are shown in Fig. 4(a). There are 26 places and 20 transitions. The places have the following set partition: $P^0 = \{p_1, p_{25,} p_{26}\}$, $P_R = \{p_{18} - p_{24}\}$, and $P_A = \{p_2, \ldots, p_{17}\}$. It has 11 SMS, 6 and 5 of which are dependent siphons and elementary siphons, respectively. Moreover, it has 1492 reachable markings. The applications of applied algorithms are shown in Table IV.

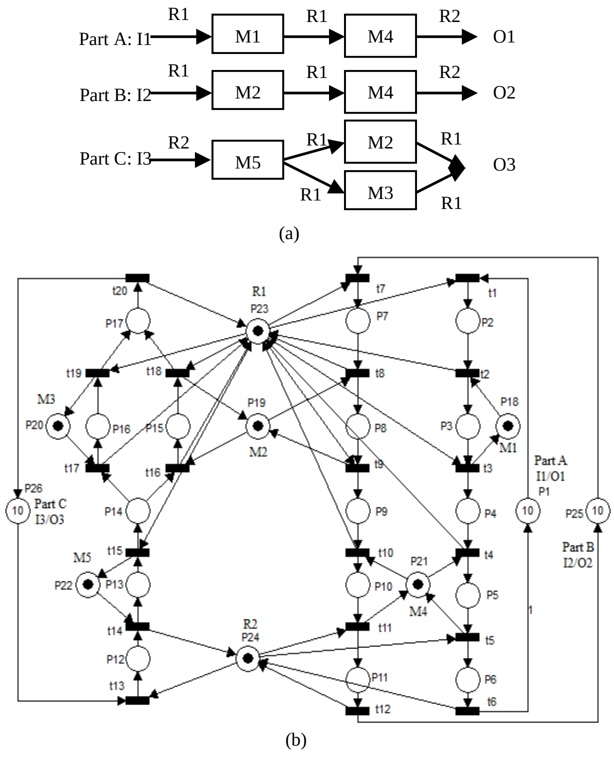
Table IV: Supervisor performance of applied algorithms for Case Study 1.

Parameters	Algorithm 1	Algorithm 2
No. of Monitors	11	5
No. of Arcs	66	23
No. of Reachable States	1072	1072

Table IV shows the results for the case study 1 in terms of the numbers of additional places, additional arcs, and reachable markings of the controlled net for both algorithms. From the table algorithm 2 obtains fewer numbers of monitors and arcs than algorithm 1 and obtain the same number of reachable states with algorithm 1. Fig.5 and Fig.6 show the results of Matlab simulation for utilization resources and throughput of the case study1, respectively. From the figures, it can be found that both applied algorithms obtain the same utilization for M1and M4, and R1, for M3, R2 algorithm 1 obtains utilization better than algorithm 2, moreover, algorithm2 obtains utilization better than algorithm1in M2 and M5. Related to the throughput algorithm 2 obtains greater produced number of part A and part B than algorithm1; algorithm1 obtains type C greater than algorithm2. Tables V and VI show the required monitors using algorithm 1 and 2 for case study 1, respectively.

Table V: Required Monitors using algorithm 1 for Case Study 1.

Siphon	${}^{\bullet}V_{si}$	$V_{si}{}^{\bullet}$	$M_{oA}(V_{si})$
S_1	t_5,t_{11},t_{18},t_{19}	t_1,t_7,t_{13}	6
S_2	t_5,t_{11}, t_{15},t_{18}	t_1,t_7,t_{13},t_{16}	5
S_3	t_5,t_{11},t_{15}	t_1,t_9,t_{13}	4
S_4	t_3,t_9, t_{18},t_{19}	t_1,t_7,t_{15}	3
S_5	t_5,t_{11}, t_{18},t_{19}	t_3,t_7,t_{13}	5
S_6	t_5,t_{11}, t_{15},t_{18}	t_3,t_7, t_{13},t_{16}	4
S_7	t_5,t_{11},t_{15}	t_3,t_9,t_{13}	3
S_8	t_9,t_{18}, t_{19}	t_7,t_{15}	2
S_9	t_3,t_9, t_{18}	t_1,t_7,t_{16}	2
S_{10}	t_3	t_1	1
S_{11}	t_9,t_{18}	t_7, t_{16}	1



(a)

(b)

Fig.4. (a) Production sequence (b) Petri net model of a case study 1.

Table VI: Required Monitors using algorithm 2 for Case Study 1.

Siphon	${}^{\bullet}V_{si}$	$V_{si}^{\bullet}$	$M_{oA}(V_{si})$
S_3	t_5,t_{11},t_{15}	t_1,t_9,t_{13}	4
S_7	t_5,t_{11},t_{15}	t_3,t_9,t_{13}	3
S_8	t_9,t_{18}, t_{19}	t_7,t_{15}	2
S_{10}	t_3	t_1	1
S_{11}	t_9,t_{18}	t_7, t_{16}	1

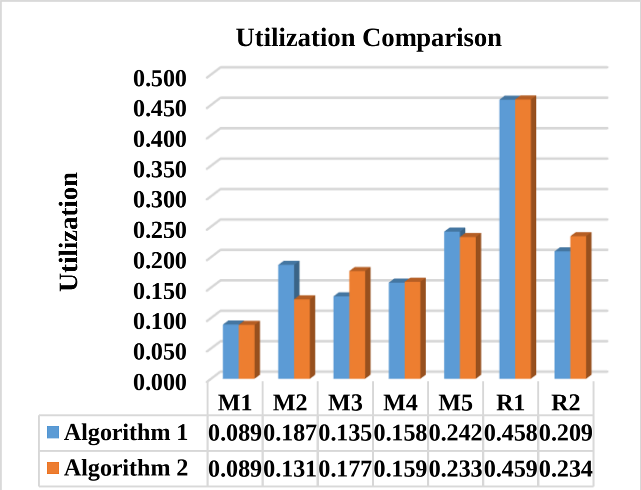


Fig.5. Utilization of resources for the Petri net model in case study 1

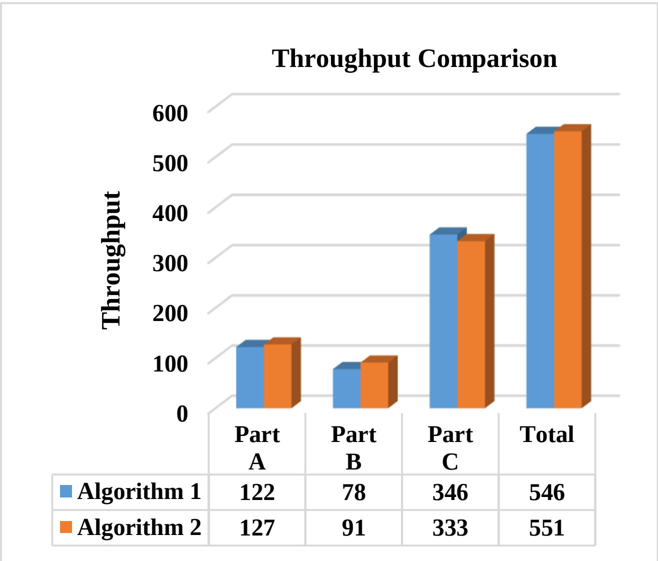


Fig.6. Throughput of the system for the Petri net model in case study 1

B. *Case study 2*

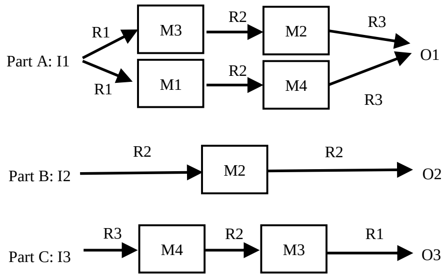
The Petri net model of second AFMS case study is shown in Fig. 6(b). This case developed by Ezpeleta et al. [15]. It consists of three robots (Rl. R2 and R3; each one can hold a product at a time) and four machines (Ml, M2, M3, and M4; each one can process two products at a time). There are three loading buffers (named I1,I2, and I3) and three unloading buffers (named O1,O2, and O3) for loading and unloading the cell. Every raw product arriving to the cell belongs to one of the three following types: PA, PB, and PC.

The production sequences are shown in Fig. 6(a). There are 26 places and 20 transitions. The places have the following set partition: $P^0 = \{p_1, p_{10}, p_{14}\}$, $P_R = \{p_{20}, \ldots, p_{26}\}$, and $P_A = \{p_2, \ldots, p_4, p_6, \ldots, p_{13}, p_{15}, \ldots, p_{19}\}$. It has 18 SMS, 9 and 9 of which are dependent siphons and elementary siphons, respectively. Moreover, it has 26750 reachable markings. The applications of applied algorithms are shown in Table VII.

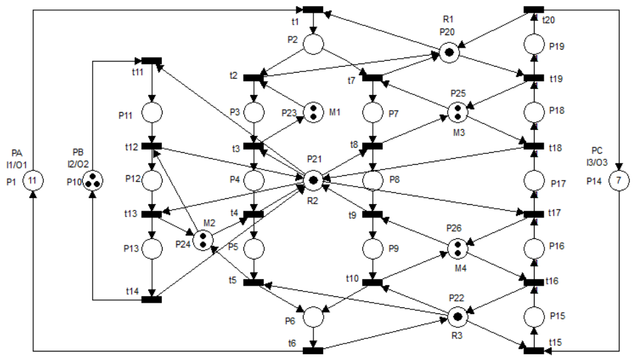
Table VII: Supervisor performance of applied algorithms for Case Study 2.

Parameters	Algorithm 1	Algorithm 2
No. of Monitors	18	9
No. of Arcs	106	50
No. of Reachable States	6287	6287

From the table VII algorithm 2 obtains fewer numbers of monitors, arcs, and reachable states than algorithm 1. Fig.7 and Fig.8 show the results of Matlab simulation for utilization resources and throughput of the case study2, respectively. In Fig.7, it can be found that when applied algorithm 2 M2, M4, R2, and R3 have higher utilization compared with algorithm 1 ,the same situation with algorithm 1 M1, M3, and R1 have higher utilization compared with algorithm 1. In Fig.8, algorithm 2 obtains greater produced number of part A and part B compared with algorithm1, moreover, algorithm1 can produced higher number of part C than algorithm 2. Tables VIII and IX show the required monitors using algorithm 1 and 2 for case study 2, respectively.



(a)



(b)

Fig.6. (a) Production sequence (b) Petri net model of a case study 2.

Table VIII: Required Monitors using algorithm 1 for Case Study 2.

Siphon	${}^{\bullet}V_{si}$	$V_{si}^{\bullet}$	$M_{oA}(V_{si})$
S_1	t_5,t_{10},t_{13},t_{19}	t_1,t_{11},t_{15}	10
S_2	t_5,t_{10},t_{13},t_{18}	t_3,t_7,t_{11},t_{15}	7
S_3	t_5,t_{10},t_{13},t_{17}	t_3,t_8,t_{11},t_{15}	5
S_4	t_2,t_{10}, t_{16}	t_9,t_{15}	2
S_5	t_4,t_9,t_{13},t_{19}	t_1,t_{11},t_{16}	9
S_6	t_4,t_9, t_{13},t_{18}	t_3,t_7, t_{11},t_{16}	6
S_7	t_4,t_9,t_{13},t_{17}	t_3,t_8, t_{11},t_{16}	4
S_8	t_4,t_8, t_{13},t_{19}	t_1,t_{11},t_{17}	7
S_9	t_4,t_8, t_{13}, t_{18}	t_3,t_7,t_{11},t_{17}	4
S_{10}	t_4,t_7, t_{13}	t_3,t_{11}	2
S_{11}	t_3,t_{10}, t_{19}	t_1,t_{15}	8
S_{12}	t_2,t_{10}, t_{18}	t_7,t_{15}	5
S_{13}	t_2,t_{10}, t_{17}	t_8,t_{15}	3
S_{14}	t_3,t_8, t_{19}	t_1,t_{17}	5
S_{15}	t_2,t_8, t_{18}	t_7,t_{17}	2
S_{16}	t_3,t_9, t_{19}	t_1,t_{15}	7
S_{17}	t_2,t_9, t_{18}	t_7,t_{16}	4
S_{18}	t_2,t_9, t_{17}	t_8,t_{16}	2

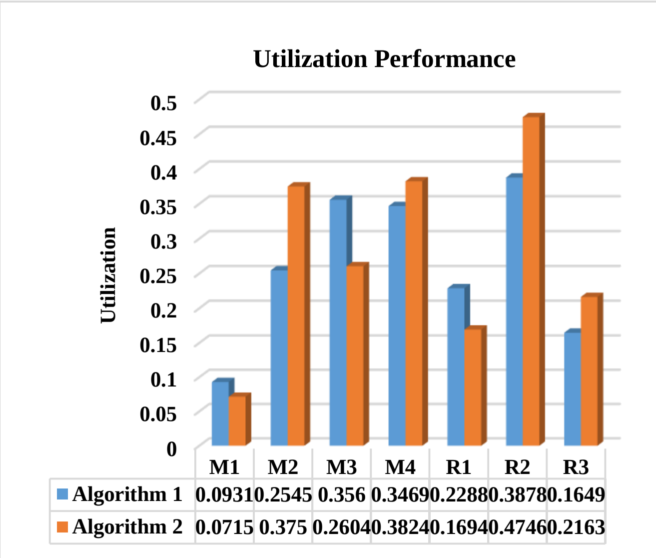


Fig.7. Utilization of resources for the Petri net model in case study 2.

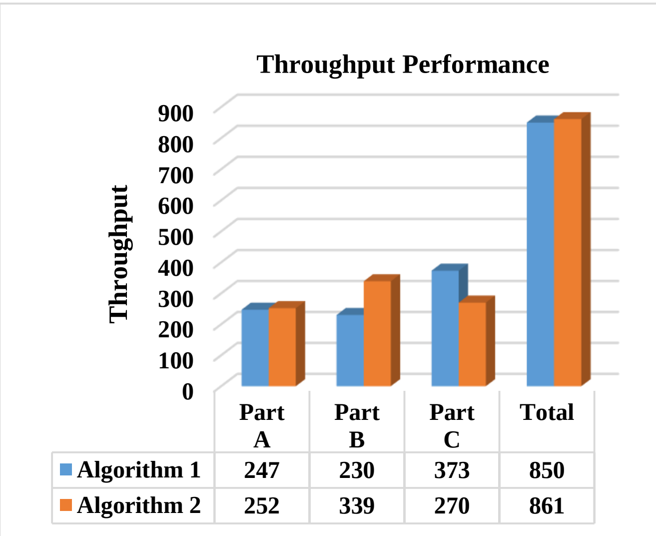


Fig.8. Throughput of the system for the Petri net model in case study 2.

Table IX: Required Monitors using algorithm 2 for Case Study 2.

Siphon	${}^{\bullet}V_{si}$	$V_{si}^{\bullet}$	$M_{oA}(V_{si})$
S_1	t_5,t_{10},t_{13},t_{19}	t_1,t_{11},t_{15}	10
S_3	t_5,t_{10},t_{13},t_{17}	t_3,t_8,t_{11},t_{15}	5
S_4	t_2,t_{10}, t_{16}	t_9,t_{15}	1
S_{10}	t_4,t_7, t_{13}	t_3,t_{11}	2
S_{11}	t_3,t_{10}, t_{19}	t_1,t_{15}	8
S_{14}	t_3,t_8, t_{19}	t_1,t_{17}	5
S_{15}	t_2,t_8, t_{18}	t_7,t_{17}	1
S_{17}	t_2,t_9, t_{18}	t_7,t_{16}	4
S_{18}	t_2,t_9, t_{17}	t_8,t_{16}	1

VI. CONCLUSION

When the two applied deadlock control methods (strict minimal siphons and elementary siphons based polices) are used to design a liveness-enforcing supervisor for two AMS case studies, and the controlled case studies are modeled and simulated using Matlab software (Petri net toolbox) to analysis and evaluate the performance of applied methods such as utilization of resources and throughput, the computational results can be summarized as follows.

In terms of behavioral permissiveness, the elementary siphons based policy can obtain behavioral permissiveness better than strict minimal siphons based policy. From the aspect of structural complexity, the elementary siphons based policy can reach smaller number of control places than strict minimal siphons based policy. As for the computational cost, the elementary siphons based policy can lead to computational complexity liveness-enforcing net supervisors than strict minimal siphons based policy.

ACKNOWLEDGMENT

This works is funded and support with a research grants project No. 12-ELE2506-02 by the National Plan for Science & Technology (NPST), King Saud University, Kingdom of Saudi Arabia.

REFERENCES

[1] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, pp. 541-580, 1989.

[2] R. A. Wysk, N.-S. Yang, and S. Joshi, "Detection of deadlocks in flexible manufacturing cells," *Robotics and Automation, IEEE Transactions on*, vol. 7, pp. 853-859, 1991.

[3] Z. Li and M. Zhou, "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 34, pp. 38-51, 2004.

[4] H. Hu, M. Zhou, and Z. Li, "Low-cost and high-performance supervision in ratio-enforced automated manufacturing systems using timed Petri nets," *Automation Science and Engineering, IEEE Transactions on*, vol. 7, pp. 933-944, 2010.

[5] D. Y. Chao, "Direct minimal empty siphon computation using MIP," *The International Journal of Advanced Manufacturing Technology*, vol. 45, pp. 397-405, 2009.

[6] D. Y. Chao, "Improved controllability test for dependent siphons in S3PR based on elementary siphons," *Asian Journal of Control*, vol. 12, pp. 377-391, 2010.

[7] A. Ghaffari, N. Rezg, and X. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *Robotics and Automation, IEEE Transactions on*, vol. 19, pp. 137-141, 2003.

[8] M. Uzam and M. Zhou, "Iterative synthesis of Petri net based deadlock prevention policy for flexible manufacturing systems," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, 2004, pp. 4260-4265.
[9] 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, pp. 204-219, 2004.
[10] K. Lautenbach, "Linear algebraic calculation of deadlocks and traps," in *Concurrency and Nets*, ed: Springer, 1987, pp. 315-336.
[11] 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, pp. 192-208, 2002.
[12] D. Y. Chao, "Fewer monitors and more efficient controllability for deadlock control in S3PGR2 (systems of simple sequential processes with general resource requirements)," *The Computer Journal,* vol. 53, pp. 1783-1798, 2010.
[13] D. Y. Chao, "Improvement of suboptimal siphon-and FBM-based control model of a well-known," *Automation Science and Engineering, IEEE Transactions on,* vol. 8, pp. 404-411, 2011.
[14] Z. Li and M. Zhou, "Comparison of two deadlock prevention methods for different-size flexible manufacturing systems," *International Journal of Intelligent Control and Systems,* vol. 10, pp. 235-243, 2005.
[15] J. Ezpeleta, J. M. Colom, and J. Martinez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *Robotics and Automation, IEEE Transactions on,* vol. 11, pp. 173-184, 1995.
[16] Y. Huang, M. Jeng, X. Xie, and S. Chung, "A deadlock prevention policy for flexible manufacturing systems using siphons," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2001, pp. 541-546.
[17] Y.-S. Huang, M. Jeng, X. Xie, and D.-H. Chung, "Siphon-based deadlock prevention policy for flexible manufacturing systems," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 36, pp. 1248-1256, 2006.
[18] Z.-w. Li and M. Zhou, "Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *Industrial Informatics, IEEE Transactions on*, vol. 2, pp. 313-325, 2006.
[19] Y.-S. Huang, "Design of deadlock prevention supervisors using Petri nets," *The International Journal of Advanced Manufacturing Technology,* vol. 35, pp. 349-362, 2007.
[20] 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, pp. 24-35, 2007.
[21] Z. W. Li, H. S. Hu, and A. R. Wang, "Design of liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 37, pp. 517-526, 2007.
[22] S. Y. Li, Z. W. Li, and H. S. Hu, "Siphon extraction for deadlock control in flexible manufacturing systems by using Petri nets," *International Journal of Computer Integrated Manufacturing,* vol. 24, pp. 710-725, 2011.
[23] G. Liu, Z. 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.
[24] Y. Hou, Z. Li, M. Zhao, and D. Liu, "Extraction of elementary siphons in a class of generalized Petri nets using graph theory," *Engineering Computations,* vol. 31, pp. 331-352, 2014.
[25] 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,* 2014.
[26] J. Desel and J. Esparza, "Free choice Petri nets, volume 40 of Cambridge tracts in theoretical computer science," ed: Cambridge University Press, Cambridge, UK, 1995.

BIOGRAPHIES

Abdulaziz M. El-Tamimi is an emeritus professor of manufacturing systems engineering at the department of industrial engineering, King Saud University. Over forty-three years experience in research, teaching, consultations, and training for the design and analysis of industrial and production systems. This involves various activities and tasks that include Machine design, instrumentation and control, computer control, Computer Aided Design and Manufacturing, knowledgebase design, facility design of production systems, maintenance, quality, and safety applications with strong background in organizing, planning and management of engineering and training programs, engineering projects, technology transfer, and with vast experience in system analysis and design. He holds a Ph.D. from University of Manchester, Institute of science and technology, Manchester, U.K.

Emad Abouel Nasr is an Associate Professor in Industrial Engineering Department, College of Engineering, King Saud University, Saudi Arabia, and Mechanical Engineering Department, Faculty of Engineering, Helwan University, Egypt. He received his PhD in Industrial Engineering from University of Houston, TX, USA, in 2005. His current research focuses on CAD, CAM, rapid prototyping, advanced manufacturing systems, supply chain management, and collaborative engineering.

Abdulrahman Al-Ahmari is the Dean of Advanced Manufacturing Institute, Executive director of CEREM (Center of Excellence for Research in Engineering Materials), Supervisor of Princess Fatimah Alnijris's Research Chair for Advanced Manufacturing Technology, and Supervisor of CMTT (Center of Manufacturing Technology Transfer). He received his Ph.D. (Manufacturing Systems Engineering) in 1998 from University of Sheffield- UK. His research interests are in analysis and design of manufacturing systems; Computer Integrated Manufacturing (CIM); optimization of manufacturing operations; applications of simulation optimization; FMS and cellular manufacturing systems.

ZhiWu Li (M'06–SM'07) received the B.S., M.S., and Ph.D. degrees in mechanical engineering, automatic control, and manufacturing engineering from Xidian University, Xi'an, China, in 1989, 1992, and 1995, respectively. He joined Xidian University, in 1992, where he is currently a Professor of School of Electro- Mechanical Engineering and the director of Systems Control and Automation Group.

Husam Kaid is a Researcher and Master Student in Industrial Engineering Department, College of Engineering, King Saud University, Saudi Arabia. He received the B.S. in Industrial Engineering from University of Taiz, Taiz, Yemen, in 2010.