



Article

Dealing with Deadlocks in Industrial Multi Agent Systems

František Čapkovič

Institute of Informatics, Slovak Academy of Sciences, 845 07 Bratislava, Slovakia; frantisek.capkovic@savba.sk

Abstract: Automated Manufacturing Systems (AMS) consisting of many cooperating devices incorporated into multiple cooperating production lines, sharing common resources, represent industrial Multi-Agent Systems (MAS). Deadlocks may occur during operation of such MAS. It is necessary to deal with deadlocks (more precisely said, to prevent them) to ensure the correct behavior of AMS. For this purpose, among other methods, methods based on Petri nets (PN) are used too. Because AMS are very often described by PN models, two PN-based methods will be presented here, namely based on (i) PN place invariants (P-invariants); and (ii) PN siphons and traps. Intended final results of usage these methods is finding a supervisor allowing a deadlock-free activity of the global MAS. While the former method yields results in analytical terms, latter one need computation of siphons and traps.

Keywords: automated manufacturing systems; multi-agent systems; Petri nets; P-invariants; siphons and traps; supervisor

1. Introduction

Recent decades represent a huge evolution of Automated Manufacturing Systems (AMS), previously named Flexible Manufacturing Systems (FMS). We can even talk about the new industrial revolution. According to [1], this evolution is realized mainly by the development in four axes, namely products, technology, business strategies, and production paradigms. The edited book [2] offers a wide perspective on modern design and operation of production networks. The evolution of the manufacturing system in the future is presented in [3].

The operation of AMS often has a similar character to Discrete-Event Systems (DES), where the next state depends only on both the current state and the occurrence of discrete events. Petri nets (PN) [4,5] are frequently used for modeling and control of DES. The family of AMS is a typical representative of DES where many devices cooperate together—robots, machine tools, transport belts, automatically guided vehicles (AGV), etc. They are frequently called to be resources. They are shared by multiple production lines, robotized working cells, etc. These devices and their aggregates, in the form of lines and cells, can also be considered industrial agents and the whole AMS as a multi-agent system. The adequate resource allocation is very important in AMS in order to avoid deadlocks. A deadlock in general is a state in which two or more processes are each waiting for the other one to execute, but neither can go on. Hence, deadlock is undesirable and bad phenomenon. Due to deadlocks, either the entire plant or some of its parts remain stagnate. Thus, the primal intention of the production cannot be achieved. For a design of deadlock-free AMS discrete mathematics is necessary. PN-based discrete mathematical models of AMS yield a suitable background for this.

There exists a large number of publications interested in this topic. From the older ones, the following works should be mentioned in particular [6–10], as well as the UML (Unified Modeling Language)-based work [11] applied in software engineering. While in [6,9] foundations of the *P*-invariant-based method were laid, in [7,8] the method based on siphons and traps was elaborated. A generalized view on deadlocks in multi-agent systems is presented in [10]. Among the more recent works should be mentioned [12–15], as well as the author's survey article [16]. While in [12] a comprehensive survey of Petri



Citation: Čapkovič, F. Dealing with Deadlocks in Industrial Multi Agent Systems. *Future Internet* **2023**, *15*, 107. <https://doi.org/10.3390/fi15030107>

Academic Editors: Agostino Poggi, Martin Kenyeres, Ivana Budinská and Ladislav Hluchy

Received: 14 February 2023

Revised: 2 March 2023

Accepted: 3 March 2023

Published: 9 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

net siphons were presented, in [13,15] unreliable resources are investigated and the robust control for AMS containing such resources is designed. In [14], deadlocks in mobile agent systems are detected and resolved.

In relation to the elimination of deadlocks, in [16–21] three categories of strategies are distinguished: (i) deadlock detection and recovery—which is used in cases where deadlocks are infrequent and their consequences are not too serious; (ii) deadlock prevention—it imposes restrictions on the interactions among resources and their users to prevent resource requests that may lead to deadlocks; and (iii) deadlock avoidance—it grants a resource to a user only if the resulting state is not a deadlock.

Consequently, it is impossible to address to such a broad issue in one paper. In this paper, only two methods of deadlock prevention in AMS by means of PN-based models are introduced and compared. Both methods add monitors (additional places) into the original PN model of AMS in order to remove deadlocks. One method enumerates monitors and their interconnections with the original PN model using P-invariants and the other using siphons and traps.

1.1. Formal Methods in AMS

Formal methods for modeling, simulation, supervisory control, performance evaluation and fault diagnosis of AMS are very important part of global understanding of AMS. They yield an efficient help and knowledge at constructing and real implementation of AMS. A suitable review of such methods is presented in the e-book [22]. As to modeling AMS there are presented also models based on place/transition Petri nets (P/T PN), timed Petri nets (TPN), and hybrid Petri nets (HPN).

The supervisory control of AMS determined for the deadlock prevention and avoidance are presented by means of: (i) finite state automata (FSA); (ii) the Petri net view on AMS which are perceived as resource allocation systems (RAS); (iii) HPN-based inventory control systems; (iv) stochastic flow models (SFM); (v) the infinitesimal perturbation analysis (IPA); and (vi) the max-plus algebra.

The performance evaluation is watched by colored TPN, by continuous PN (CPN), by the timed process algebra, and by the Petri net-based complex system scheduling.

The fault diagnosis of AMS is analyzed by means of FSA, by the fault diagnosis of PN and by the control reconfiguration of discrete-event manufacturing systems modeled by non-deterministic input/output (I/O) automata.

Along with the development of information technologies, such as cloud computing, mobile Internet, information acquisition technology, and big data technology, traditional engineering knowledge and formal methods for knowledge-based software engineering undergo fundamental changes. Hence, networks also play an increasingly important role.

Within this context, it is necessary to develop new methodologies as well as technical tools for network-based approaches. The term “network” may have different meanings in different contexts. To resolve bottleneck problems in AMS, deadlock prevention and avoidance by means of Petri nets is crucial.

Although introduced topics cover a large set of formal methods in AMS it cannot be said that this set is complete. New methods are under development.

1.2. Agents Cooperation, Negotiation and Reentering in AMS

Important strategies in MAS (multi-agent systems) are cooperation and negotiation. A structural and Petri net-based approach to modeling of these strategies was elaborated in author’s older work [23] and in the chapter [24] where also the perspectives of learning in this area were analyzed.

In [25], a connection of the structural model of reenterable AMS on a computer-based supervisory controller for monitoring the status of jobs and regulating the part routing as well as the machine job selection by means of siphons of Petri nets as to resolving deadlocks was explored. Namely, there are many approaches to modeling and analysis of manufacturing systems. In addition to those ones mentioned above in the previous Subsection (i.e., automata,

Petri nets, perturbation methods) there exist methods based on digraphs, alphabet-based approaches, control theoretic techniques, expert systems design, etc.

The so-called *complex networks* are an interdisciplinary area. Numerous natural and artificial systems are composed of a large number of individuals that interact with each other in various ways, and then perform surprising useful functions. Examples of such networks include, e.g., neural networks, social networks, and many others, but also MAS in general, including industrial MAS where AMS and RAS belong, and doubtlessly also PN that successfully model and analyze them.

1.3. The Paper Organization

In the following Section 2, preliminaries concerning theory of PN and the PN-based modeling of AMS will be concisely introduced. Then, Section 3 will be devoted to the presentation and illustration of the method of deadlock prevention, based on P-invariants. In Section 4, the method of deadlock prevention, based on PN siphons and traps, will be presented and illustrated. Section 5 will discuss and illustrate three PN models of AMS being more complicated from the computation point of view, namely problematic because of very long-lasting even impossible computations. In Section 6, the comparison of both methods will be introduced. Then, Section 7 contains a concise conclusions of this paper.

2. Preliminaries

Firstly, it is necessary to introduce several basic definitions. First, one is the author own definition extracted from knowledge published in [7,16], next three are the standard elementary definitions of P-invariants, siphons, and traps from the classical literature about Petri nets [4,5]. Last two definitions are taken over from [7].

Definition 1. A Petri net is a bipartite directed graph (BDG). It may be formally described by the quadruplet $N = (P, T, F, W)$. Here, P and T are finite nonempty sets, namely P is a set of places ($|P| = n$), representing the first kind of BDG nodes, and T is a set of transitions ($|T| = m$), representing the second kind of BDG nodes. It is valid that $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$. The set $F = (P \times T) \cup (T \times P)$ is the flow relation of the net N . It is expressed by directed arcs from places to transitions and vice versa. The mapping $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ assigns weights to directed arcs. The weight of an arc $f : W(f) > 0$ if $f \in F$ and $W(f) = 0$ otherwise. Here $\mathbb{N} = \{0, 1, 2, \dots\}$, i.e., it contains natural numbers plus zero. The Petri net $N = (P, T, F, W)$ is named ordinary and denoted as $N = (P, T, F)$ if $\forall f \in F, W(f) = 1$. N is called generalized if $\exists f \in F, W(f) > 1$. Petri net is called a state machine if, and only if, (iff) $\forall t \in T, |\bullet t| = |t\bullet| = 1$, i.e., when each transition has only one input place and only one output place.

A mapping from P to \mathbb{N} is a marking M of N . The marking $M(p)$ means the number of tokens located in the place p . Thus, we may say that the place p is marked by a marking M iff $M(p) > 0$. In case of a subset $S \subseteq P$ we say that it is marked by M iff at least one place in S is marked by M . The sum of tokens located in all places of S is $M(S) = \sum_{p \in S} M(p)$. S is empty at M iff $M(S) = 0$. The pair (N, M_0) is called the marked net or a net system. Here, M_0 is an initial marking of N .

PN marking is evolved as follows

$$\mathbf{M}_{k+1} = \mathbf{M}_k + \mathbf{N} \cdot \vec{\sigma}_k, \quad k \in \mathbb{N}, \tag{1}$$

with $\mathbf{M}_0 = (M_0(p_1), M_0(p_1), \dots, M_0(p_n))^T$ being the $(n \times 1)$ initial marking. Here, \mathbf{N} is the $(n \times m)$ -dimensional incidence matrix based on the set F and $\vec{\sigma}_k$ is a $(m \times 1)$ -dimensional firing vector (the binary vector, where its entry with the value 1 denotes the corresponding transition $t \in T$ able to be fired). $M(p)$ denotes (as it was already mentioned above) the number of tokens in place p . $(n \times 1)$ marking vectors \mathbf{M} are sometimes named also as *state vectors*. $\mathbf{N} = \mathbf{N}^+ - \mathbf{N}^-$, where $\mathbf{N}^+ = \mathbf{Post}$ and $\mathbf{N}^- = \mathbf{Pre}$ are, respectively, the output flow matrix and the input flow matrix. In other words, the matrix representing the output of transitions to places and the matrix representing the input of places to transitions.

For economy of space, in general, markings and vectors can be described by the formal sum notation. Thus, $\sum_{p \in P} M(p)p$ denotes the vector \mathbf{M} . For example, in a net with $P = \{p_1, \dots, p_5\}$ a marking where in p_2 are 3 tokens and in p_3 are 2 tokens may be written as $3p_2 + 2p_3$ instead of the vector notation $(0, 3, 2, 0, 0)^T$.

Definition 2. A vector \mathbf{I} is a place invariant P (shortly, P -invariant) of PN iff $\mathbf{I} \neq \mathbf{0}$ and $\mathbf{I}^T \cdot \mathbf{N} = \mathbf{0}^T$. Here, \mathbf{N} is the incidence matrix of N .

Definition 3. Any set $S \subseteq P$, $S \neq \emptyset$ with $\bullet S \subseteq S^\bullet$ is said to be a siphon. Verbally said, if every transition having an output place in S has an input place in S . If a siphon does not contain another siphon, being its proper subset, it is said to be a minimal siphon. A proper subset is any subset of the set except itself. A minimal siphon S is called a strict minimal siphon (SMS) if $\bullet S \subsetneq S^\bullet$. In other words, if there is no siphon contained in it as a proper subset. A SMS does not contain a trap (the trap is defined below by Definition 4). When at the initial marking \mathbf{M}_0 $M_0(S) = \sum_{p \in S} M_0(p) = 0$, S is called an empty siphon.

Definition 4. Any set $Q \subseteq P$, $Q \neq \emptyset$ with $Q^\bullet \subseteq \bullet Q$ is called a trap. In other words, if every transition having an input place in Q has an output place in Q .

Definition 5. A siphon S is said to be controlled in a net (N, M_0) iff $\forall M \in R(N, M_0), M(S) > 0$. Here, $R(N, M_0)$ expresses the reachability of markings from the initial marking. Consequently, any siphon containing a marked trap is controlled, since the marked trap can never be emptied. Thus, in ordinary PN a controlled siphon does not cause any deadlock.

Definition 6. A siphon S in an ordinary PN (N, M_0) is [7] invariant controlled by P -invariant \mathbf{I} under the initial marking \mathbf{M}_0 iff $\mathbf{I}^T \cdot \mathbf{M}_0 > 0$ and $\forall p \in P \setminus S, I(p) \leq 0$, or equivalently, $\mathbf{I}^T \cdot \mathbf{M}_0 > 0$ and $\|\mathbf{I}\|^+ \subseteq S$. Here, $\|\mathbf{I}\|^+ = p \in P | I(p) > 0$ is the positive support of \mathbf{I} .

Here, it is necessary to notify and emphasize that both methods examined in this paper require a rather wide mathematical background that cannot be either simplified nor abbreviated. Otherwise, the integrity of the interpretation of the methods would be violated. The mathematics used in examples should contribute to an accurate understanding of the application of both methods to particular cases of AMS.

2.1. Petri Nets and Resource Allocation Systems

Resource allocation systems (RAS) are a special class of concurrent systems, especially AMS. There, the attention is focused above all on resources. The set of Petri net-based models of RAS represent, in general, a subset of complete set of PN. Finite set of resources is shared in a competitive way by a finite set of processes. Such a competition induces (or may induce) existence of deadlocks. PN-based models of RAS are useful at synthesizing policies of deadlock prevention and/or deadlock avoiding. They make possible to design beforehand deadlock-free AMS, what is very useful especially preliminary to the creation of the final design of the structure of real AMS determined for application in production.

There are several standard paradigms of RAS [26–29]. Specific nomenclatures have been established for them—see, e.g., a survey made in [16], where also their mutual relations and their relations to PN as a whole, are displayed. Most frequently used paradigms of them are the following two: (i) S^3PR (Systems of Simple Sequential Processes with Resources) suitable for AMS with flexible routing and the acquisition of the single-unit of the resource and (ii) S^4PR (System of Sequential Systems with Shared Process Resources) which enables the modeling AMS with flexible routing and the acquisition of more units of the resource. The relation $S^3PR \prec S^4PR$ in this case expresses that systems S^4PR model more complicated AMS than systems S^3PR are able to model. In future, there may be S^*PR ($S^3PR \prec S^4PR \prec S^*PR$) paradigms suitable to model of yet more complicated AMS, where yet more copies of resources will be allowed. However, the asterisk needs not always be just a greater integer than it was in previous two paradigms.

Some of the amount of paradigms, especially S^3PR and S^4PR , can be modeled and simulated in Matlab by means of the tool [30].

2.2. Methods for the Deadlock Prevention and Avoiding

As already mentioned, two main streams of methods are represented by methods based on: (i) reachability trees and P -invariants of PN and (ii) PN siphons and traps.

In the former case, it is necessary to mention the most principal contributions [6,8,9,31–33] important for the evolution of the P -invariants-based methods. The origin of this method consists in the effort to deal with the problem of the forbidden state in DES. Namely, it was shown in the literature that if a set \mathcal{L} of legal PN markings is expressed by a set of s linear inequality constraints (so called General Mutual Exclusion Constraints (GMEC)) and if \mathcal{L} is controllable, then the PN-based solution exists and it is maximally permissive. The controllability of \mathcal{L} means that from any marking $M \in \mathcal{L}$ no forbidden marking is reachable by firing a sequence containing only uncontrollable transitions. In the opposite case, a general forbidden marking constraint may be enforced by PN-based controller only if the Petri net model of the system is safe. Hence, PN P -invariants are used in order to compute the feedback controller that enforces GMEC. They perform transformations on the system's specifications to obtain constraints in the desired form. The controller is a set of control Petri net places. When the controller is added to the PN model of the plant, control places ensure a live and maximally permissive behavior of the closed loop system with respect to the forbidden states or markings. The controller design has a linear complexity in the number of system states. However, there are cases where such PN controller does not exist. Especially, when the set of legal markings \mathcal{L} is not convex, there does not exist any PN place that can forbid the reachability of bad markings when allowing all legal ones. The concept of PN P -invariants was extended also to PN with uncontrollable and unobservable transitions. However, in this case maximally permissiveness cannot be guaranteed.

In the latter case, there are lots more such basal publications—[6,7,10,28,34–57]. Siphons are tied to dead transitions whose existence leads to the loss of liveness of PN-based models of AMS. Siphon control is an effective way to prevent the occurrence of deadlocks. In a general case, particularly for models of AMS based on generalized PN, a siphon-based deadlock prevention policy cannot find an optimal (i.e., maximally permissive) supervisor. However, for PN models of AMS based on ordinary PN it is possible. Siphons in the PN model of a plant are divided into elementary and dependent ones. A monitor is added to the plant model for each elementary siphon, in such a way that the siphon will be *invariant-controlled*. The method guarantees that no further emptiable control-induced siphon is generated due to the addition of the monitors. 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.

The explanation of the term invariant-controlled siphon, as well as the term control depth variable, will be explained and illustrated below in the Section 4.2. However, this is applicable only for S^3PR paradigm of RAS. In case of S^4PR paradigm (modeled by generalized PN), which uses multiple resource requirements and information about the weight of arcs, the concept of extended elementary siphons have to be used. Namely, due to the complicated utilization of resources, it is insufficient to represent the relationship between elementary and dependent siphons in generalized PN in such way as it is in case of ordinary PN. Extended elementary siphons are found by an iterative process—see, e.g., [58].

A methodology for modeling and analyzing fault-tolerant manufacturing systems that not only optimizes normal productive processes, but also performs detection and treatment of faults is presented in [59]. The coordination of cooperative multirobot system by means of PN is elucidated in [60].

The methodological framework yielded by the S^4PR paradigm of RAS has raised considerable interest [61] on the grounds of a well-balanced compromise between modeling flexibility and the provision of sound and effective correction techniques.

In [62], a very general class of Petri nets are defined. This class has been called Petri nets of resource allocation (PNRA) to model as many kinds of RAS as possible. It not only focuses on the resources shared by processes, but also pays attention to the interaction/collaboration among processes.

At the solving of deadlock problems in AMS modeled with PN in [63], the attention is focused on adding a set of recovery transitions.

Here, in this paper, devoted to deadlock prevention and avoiding, it is not a sufficient space to analyze all introduced literature sources in details item by item. Namely, this area has many isolated particulars which are analyzed in detail in the above literature [6–10,28,31–57].

Methods based on P -invariants are divided into many subclasses interested in deterministic and/or unreliable resources, methods suitable for ordinary and/or generalized PN, methods using linear and/or non-linear restrictions, methods applying optimization based on the linear programming (usually mixed integer linear programming (MILP)), etc.

Methods based on siphons and traps have still lots more such subclasses. The very good literature review of methods concerning the deadlock prevention is made in [26]. Likewise, in [53] a crucial literature review concerning the deadlock control is presented. A variety of deadlock-control policies is mentioned there. The optimization based on MILP is one of them. Such optimization can eliminate the need to compute completely all elementary siphons, which shorten the computational time for finding indispensably needed siphons.

In the following two sections, principles of both methods of the prevention of deadlocks will be introduced and mathematically described. Illustrative examples explaining the application of both methods in details will be introduced there as well.

3. P -Invariant-Based Method of Prevention Deadlocks

P -invariants are vectors \mathbf{I} with the important property. The multiplication of such vectors with any marking (state) vector \mathbf{M} reachable from a given initial marking \mathbf{M}_0 yields the same result. It is the relation of the state conservation, as follows

$$\mathbf{I}^T \cdot \mathbf{M} = \mathbf{I}^T \cdot \mathbf{M}_0 \tag{2}$$

Taking into account the consecutive states obtained by firing of only one transition, we have for each transition t

$$\mathbf{I}^T \cdot \vec{col}_t(\mathbf{N}) = 0, \tag{3}$$

where, $\vec{col}_t(\mathbf{N})$ is the column of the incidence matrix \mathbf{N} corresponding to the transition t . These vectors are solutions of the following equation

$$\mathbf{I}^T \cdot \mathbf{N} = \mathbf{0}^T \tag{4}$$

what corresponds with the original Definition 2 of P -invariants introduced in the Section 2.

Consider the inequality representing linear combinations restricting behavior of marking \mathbf{M}

$$\mathbf{L} \cdot \mathbf{M} \leq \mathbf{b} \tag{5}$$

Here, \mathbf{L} is a matrix of integers, \mathbf{b} is an integer vector. An efficient help at creating \mathbf{L} yields the reachability tree (RT) of the net N . After removing inequality by adding slack variables, we have

$$\mathbf{L} \cdot \mathbf{M} + \mathbf{s} = \mathbf{b}, \tag{6}$$

where entries of the $(n_s \times 1)$ vector \mathbf{s} represents slack variables.

Just the Equation (4) represents the base of the method for the supervisor synthesis. Namely, when we force there $[\mathbf{L}|\mathbf{I}_s]$ instead of \mathbf{I}^T , we obtain (7) and some additional PN places (slacks) can be added to N . Thus, the extended (augmented) PN $N_1 = N \circ N_s$ (where the symbol \circ expresses a composition) will arise. N_s is the PN which will (after finding

its interconnections with N) ultimately create the supervisor. Then, slacks will create the places of the supervisor, frequently called monitors. In N_1 we have the following structure of (4) in the form

$$[\mathbf{L} | \mathbf{I}_s] \cdot \begin{bmatrix} \mathbf{N} \\ \mathbf{N}_s \end{bmatrix} = \mathbf{0}, \tag{7}$$

where \mathbf{I}_s is $(n_s \times n_s)$ -dimensional *identity matrix*; n_s is the number of slacks; \mathbf{L} is a $(n_s \times n)$ -dimensional matrix which in a suitable form expresses the conditions imposed on marking (state) vectors of the original PN N , and \mathbf{N}_s is the $(n_s \times m)$ -dimensional matrix. After finding \mathbf{N}_s from (7) by computing, we have the structure of the PN-based model of the supervisor—i.e., interconnecting arcs between N and N_s . Consequently, in general,

$$\mathbf{L} \cdot \mathbf{N} + \mathbf{N}_s = \mathbf{0} \tag{8}$$

$$\mathbf{N}_s = -\mathbf{L} \cdot \mathbf{N} \tag{9}$$

$$\mathbf{N}_s = \mathbf{N}_s^+ - \mathbf{N}_s^-, \tag{10}$$

where \mathbf{N}_s^+ is \mathbf{Post}_s and \mathbf{N}_s^- is \mathbf{Pre}_s of \mathbf{N}_s .

The marking \mathbf{M}_a of the supervised system N_1 consists of the state vector \mathbf{M} of the N augmented with the state vector \mathbf{M}_s of the supervisor N_s . Analogically, the incidence matrix \mathbf{N}_a of the net N_1 consists of the incidence matrix \mathbf{N} of N augmented with the incidence matrix \mathbf{N}_s of the net N_s . The marking and the incidence matrix of the net N_1 are as follows

$$\mathbf{M}_a = \begin{pmatrix} \mathbf{M} \\ \mathbf{M}_s \end{pmatrix} \tag{11}$$

$$\mathbf{N}_a^- = \begin{pmatrix} \mathbf{N}^- \\ \mathbf{N}_s^- \end{pmatrix} \tag{12}$$

$$\mathbf{N}_a^+ = \begin{pmatrix} \mathbf{N}^+ \\ \mathbf{N}_s^+ \end{pmatrix}, \tag{13}$$

Here, \mathbf{N}_a , \mathbf{N} and \mathbf{N}_s are expressed by sub-matrices \mathbf{N}_a^+ and \mathbf{N}_a^- , \mathbf{N}^+ and \mathbf{N}^- , and \mathbf{N}_s^+ and \mathbf{N}_s^- .

Because of the prescribed conditions we have

$$[\mathbf{L} | \mathbf{I}_s] \cdot \begin{bmatrix} \mathbf{M}_0 \\ {}^s\mathbf{M}_0 \end{bmatrix} = \mathbf{b}, \tag{14}$$

where \mathbf{b} is the $(n_s \times 1)$ -dimensional vector (see (5) and (6)). Its integer entries represent the limits for number of tokens in (5). Sometimes $\mathbf{b} = \mathbf{1}$ where $\mathbf{1}$ is the vector with all its entries equal to 1. Because

$$\mathbf{L} \cdot \mathbf{M}_0 + {}^s\mathbf{M}_0 = \mathbf{b} \tag{15}$$

the initial state vector of the supervisor can be simply computed as follows

$${}^s\mathbf{M}_0 = \mathbf{b} - \mathbf{L} \cdot \mathbf{M}_0 \tag{16}$$

Example 1

Consider the simple example of N modeling an AMS with two production lines $\{p_7, t_1, p_1, t_2, p_2, t_3\}$ and $\{p_8, t_4, p_3, t_5, p_4, t_6\}$ with two common resources p_5 and p_6 , introduced in Figure 1. Places p_7 and p_8 are idle process places, p_1, p_2, p_3 , and p_4 are active process places, and p_5 and p_6 are resource places. A token in an active process place models, e.g., a part being processed. In Figure 1, active process places are empty till firing t_1 and/or t_4 . Tokens located in a resource place model, e.g., the available capacity of buffers. Tokens located in an idle place indicate the maximal number of concurrent activities that can occur in a process represented by the corresponding production line.

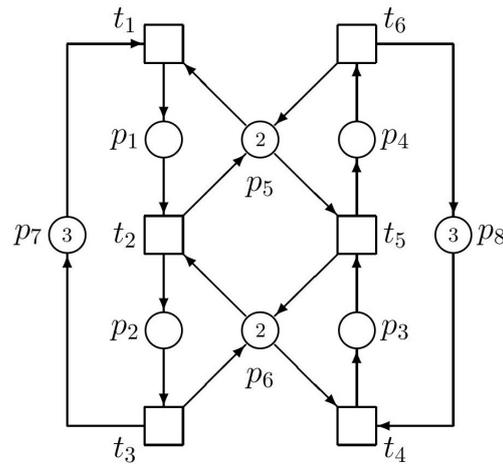


Figure 1. The PN model of an AMS.

Parameters of N are as follows

$$\mathbf{N}^- = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad \mathbf{N}^+ = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{17}$$

This N has, at the given initial state $M_0 = (0, 0, 0, 0, 2, 2, 3, 3)^T$, the fairly branched (patulous) reachability tree (RT) with 33 nodes (including M_0). All RT nodes, being the reachable markings M_0, \dots, M_{32} are the columns of the following matrix (in the ascending order):

$$\mathbf{X}_r = \begin{pmatrix} 0 & 1 & 0 & 2 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 1 & 1 & 0 & 2 & 0 & 1 & | & 2 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 2 & 1 & | & 0 & 1 & 0 & 0 & 0 & 2 & 1 & 1 & 1 & 0 & 0 & 2 & 1 & 1 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 1 & 1 & 2 & 0 & 1 & 0 & 0 & 1 & | & 2 & 0 & 1 & 2 & 0 & 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & | & 0 & 1 & 1 & 2 & 0 & 0 & 1 & 1 & 1 & 2 & 1 & 1 & 2 & 1 & 2 \\ 2 & 1 & 2 & 0 & 2 & 1 & 2 & 1 & 1 & 0 & 2 & 1 & 0 & 1 & 0 & 2 & 1 & | & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 1 & 2 & 1 & 1 & 0 & 2 & 1 & 1 & 0 & 0 & 2 & 1 & 1 & 0 & 0 & | & 0 & 1 & 1 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 3 & 2 & 3 & 1 & 2 & 2 & 3 & 3 & 1 & 1 & 2 & 2 & 2 & 3 & 0 & 1 & 1 & | & 1 & 2 & 2 & 3 & 3 & 0 & 0 & 1 & 2 & 2 & 3 & 1 & 1 & 2 & 0 & 2 \\ 3 & 3 & 2 & 3 & 3 & 2 & 1 & 2 & 3 & 2 & 2 & 1 & 2 & 1 & 3 & 3 & 2 & | & 1 & 2 & 1 & 0 & 1 & 3 & 2 & 2 & 1 & 0 & 0 & 2 & 1 & 1 & 2 & 0 \end{pmatrix} \tag{18}$$

Because RT is too big, it cannot be introduced here full. In Figure 2, at least its principal fragment is introduced.

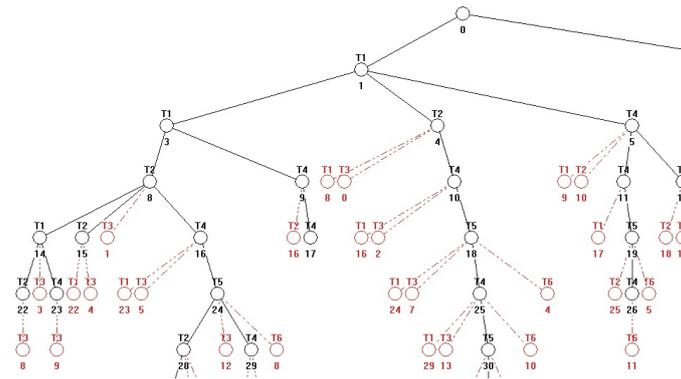


Figure 2. The principal fragment of the RT.

From the RT fragment we can find the solitary deadlock $M_{17} = (2, 0, 2, 0, 0, 0, 1, 1)^T$ —see (18), where it is separated. This deadlock can be reached by following paths:

$$M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_1} M_3 \xrightarrow{t_4} M_9 \xrightarrow{t_4} M_{17} \tag{19}$$

$$M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_4} M_5 \xrightarrow{t_1} M_9 \xrightarrow{t_4} M_{17} \tag{20}$$

$$M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_1} M_3 \xrightarrow{t_2} M_8 \xrightarrow{t_1} M_{14} \xrightarrow{t_4} M_{23} \xrightarrow{t_3} M_9 \xrightarrow{t_4} M_{17} \tag{21}$$

$$M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_1} M_3 \xrightarrow{t_2} M_8 \xrightarrow{t_4} M_{16} \xrightarrow{t_1} M_{23} \xrightarrow{t_3} M_9 \xrightarrow{t_4} M_{17} \tag{22}$$

Hence, we can see that in RT the node M_9 immediately precedes the node M_{17} representing the deadlock. These nodes are marking vectors $M_9 = (2, 0, 1, 0, 0, 1, 1, 2)^T$ (the 10th column of X_r because the first column is M_0) and $M_{17} = (2, 0, 2, 0, 0, 0, 1, 1)^T$ (the 18th column of X_r). After comparing entries of these vectors, from (19) follows the restriction $M(p_1) + M(p_3) < 2$ (i.e., only one of them can be marked), i.e., $L = (1, 0, 1, 0, 0, 0, 0, 0)^T$ and $b = (2)$. Consequently, $N_s = (-1, 1, 0, -1, 1, 0)$ and ${}^sM_0 = (2)$ because in (16) $L.M_0 = 0$. Hence, with respect to (11)–(13) we have the augmented PN N_a (i.e., the original uncontrolled net N together with the net N_s representing the supervisor) and the augmented vector aM_0 of the initial marking as follows

$$N_a = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 1 & 0 & 0 & -1 & 1 \\ 0 & -1 & 1 & -1 & 1 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ - & - & - & - & - & - \\ -1 & 1 & 0 & -1 & 1 & 0 \end{pmatrix} \quad {}^aM_0 = (M_0^T | {}^sM_0^T)^T = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 2 \\ 3 \\ 3 \\ - \\ 2 \end{pmatrix} \tag{23}$$

Thus, we have the supervisor represented by the single PN place p_9 (monitor) interconnected with the original PN model as it can be seen in Figure 3.

After computing RT of the augmented net, we can find that it contains 28 nodes and no deadlock exists there. It means that the augmented net works correctly.

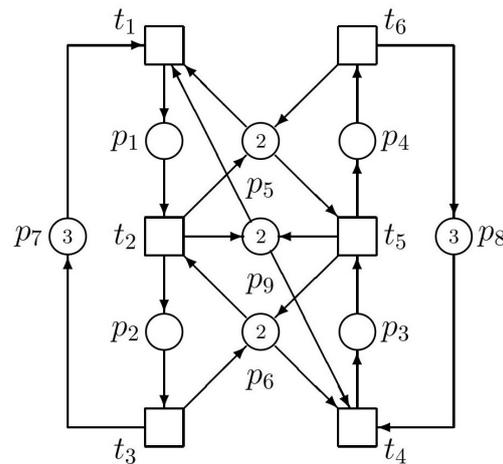


Figure 3. The controlled PN model of the AMS.

4. Siphon and Trap-Based Method of Prevention Deadlocks

Such approaches are much more widespread than approaches based on P-invariants. There are many sources in literature. Let us mention at least some of them [7,8,12,13,16,29].

Invariants, siphons, and traps are structural entities of PN. All of them can be computed, e.g., by the Matlab-based tool presented in [30]. The problem of deadlock prevention in a concurrent system represented by N is equivalent to the problem of avoidance of empty siphons in the ordinary PN model of N . When at least one empty siphon occurs in ordinary PN N , the net is totally deadlocked [5].

At the supervisor synthesis it is sufficient to consider only minimal siphons. It is necessary to ensure that the sum of the number of tokens in each minimal siphon S is never less than one, namely in any reachable marking M . In such a case, the general condition $S_j \cdot M \geq b$ for i th siphon can be replaced by the condition in the form $S_j \cdot M \geq 1$. This can be derived as follows.

Consider the following formal specification in N

$$\mathbf{l}^T \cdot \mathbf{M} \geq b \tag{24}$$

where \mathbf{l} is a $(n \times 1)$ -dimensional vector, \mathbf{M} is a marking and b is a scalar; b and the entries of \mathbf{l} are integers. The relation (24) says that the weighted sum of the number of tokens in each place should be greater than or equal to a constant.

In [6], it was proved that if a Petri net $N = (P, T, F, W)$ with incidence matrix \mathbf{N} and initial state \mathbf{M}_0 satisfies the following relation

$$b - \mathbf{l}^T \cdot \mathbf{M}_0 \leq 0 \tag{25}$$

then a control place p_c can be added, that enforces (24). Let $\mathbf{N}_c : T \rightarrow \mathbb{Z}$ (\mathbb{Z} is the set of integers) denotes the weight vector of arcs connecting p_c with the transitions in the net N ; \mathbf{N}_c is obtained by $\mathbf{N}_c = \mathbf{l}^T \cdot \mathbf{N}$. In general, when there are more p_c than one, $\mathbf{N}_c = \mathbf{L}^T \cdot \mathbf{N}$ with \mathbf{L} being the matrix where rows represent particular \mathbf{l}^T appertaining to particular p_c . The initial number of tokens in p_c is $M_0(p_c) = \mathbf{l}^T \cdot \mathbf{M}_0 - b \geq 0$ or, in general, $\mathbf{M}_c^0 = \mathbf{L} \cdot \mathbf{M}_0 - \mathbf{b} \geq \mathbf{0}$.

The controlled Petri net is maximally permissive. The control process enforces just enough control to avoid all illegal markings. This is the basis for deadlock prevention, i.e., for the synthesis of the control algorithm based on siphons. Thus, the place invariant guarantees that for any marking \mathbf{M} in the set of reachable markings of N

$$\mathbf{l}^T \cdot \mathbf{M} - M(p_c) = b \tag{26}$$

Here, as it was defined above, $M(p_c)$ is the number of tokens in the control place p_c . Due to the fact that $M(p_c)$ is non-negative, the inequality in (24) is satisfied. Equation (25)

expresses the demand that (24) must be satisfied for \mathbf{M}_0 , otherwise no solution exists there. \mathbf{b}_c represents the row which extends the incidence matrix \mathbf{N} of the uncontrolled net N with respect to the control place p_c . In general, for more additive places p_1, \dots, p_s , it can be written

$$\mathbf{L.M} \geq \mathbf{b}, \quad \text{or} \quad \mathbf{L.M} - \mathbf{M}_c = \mathbf{b} \tag{27}$$

where \mathbf{L} is $(n_s \times n)$ matrix of positive integers greater than or equal to 0. \mathbf{L} has the weighted vectors $\mathbf{l}_i^T, i = 1, \dots, s$, as its rows. The vector \mathbf{b} is $(n_s \times 1)$ vector of restrictions with entries being positive integers greater than or equal to 0; \mathbf{M}_c is $(n_s \times 1)$ vector of slacks with entries being positive integers greater than or equal to 0. Thus, we obtain the controlled PN model (with the incidence matrix \mathbf{N}_{ex}) consisting of original uncontrolled model (with the incidence matrix \mathbf{N}) and the siphon-based controller (with the incidence matrix \mathbf{N}_c) as follows

$$\mathbf{N}_c = \mathbf{L.N} \tag{28}$$

$$\mathbf{M}_c^0 = \mathbf{L.M}_0 - \mathbf{b} \geq \mathbf{0} \tag{29}$$

$$\mathbf{N}_{ex} = \begin{pmatrix} \mathbf{N} \\ \mathbf{N}_c \end{pmatrix}; \quad \mathbf{M}_{ex}^0 = \begin{pmatrix} \mathbf{M}_0 \\ \mathbf{M}_c^0 \end{pmatrix} \tag{30}$$

The controller consists of monitors and their interconnections with the original PN model.

Replacing \mathbf{l}_i^T by the siphon \mathbf{S}_i , or, in general, replacing \mathbf{L} by the matrix of siphons \mathbf{S}_m (where particular siphons are its rows) we obtain

$$\mathbf{N}_c = \mathbf{S}_m.\mathbf{N} \tag{31}$$

4.1. Example 2

Minimal siphons and traps of the PN model in Figure 1 are the following

$$\begin{aligned} \mathbf{S}_1 &= \{p_1, p_2, p_7\} & \mathit{Tr}_1 &= \{p_1, p_2, p_7\} \\ \mathbf{S}_2 &= \{p_2, p_3, p_6\} & \mathit{Tr}_2 &= \{p_2, p_3, p_6\} \\ \mathbf{S}_3 &= \{p_1, p_4, p_5\} & \mathit{Tr}_3 &= \{p_1, p_4, p_5\} \\ \mathbf{S}_4 &= \{p_3, p_4, p_8\} & \mathit{Tr}_4 &= \{p_3, p_4, p_8\} \\ \mathbf{S}_5 &= \{p_2, p_4, p_5, p_6\} & \mathit{Tr}_5 &= \{p_1, p_3, p_5, p_6\} \end{aligned}$$

Denote $S = S_5$ and $T = Tr_5$. In this simple example, the siphon and trap can be illustrated by Figure 4. Above introduced crossed siphons are equaled to traps. Such a siphon cannot be emptied once it is initially marked (this is ensured by the corresponding marked trap). Residual siphons are strict minimal siphons (SMS). Consequently, there is only one SMS, namely S_5 . In a vector form it is denoted as \mathbf{S} as follows

$$\mathbf{S} = (0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0)^T \tag{32}$$

The siphon corresponds with the Figure 4 as well as the trap Tr_5 . With respect to (31)

$$\mathbf{N}_c = \mathbf{S.N} = (-1 \ 1 \ 0 \ -1 \ 1 \ 0 \ 0 \ 0) \tag{33}$$

what is the same structure of the supervisor like in the previous approach based on P -invariants. Thus, the PN model of the controlled (supervised) AMS is in Figure 3. However, it must be said that such a coincidence of the results achieved by both methods occurs only rarely, not in general.

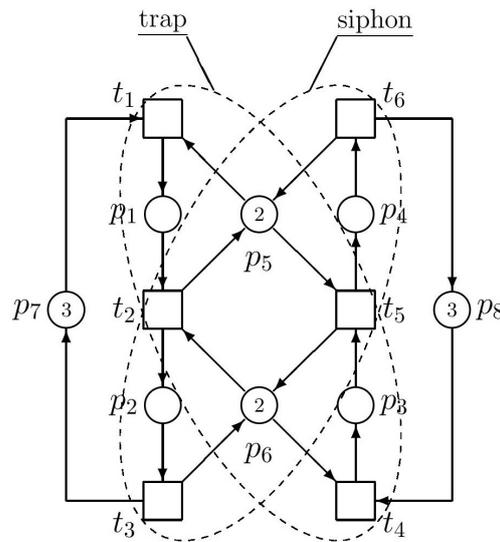


Figure 4. The siphon S and trap T in the PN model.

4.2. Invariant-Controlled Siphons and Setting the Marking of Monitors in the Siphon-Based Approach

The control places p_c creating the supervisor are frequently named as monitors. Denote them as V_{S_i} , $i = 1, \dots, n_m$ (n_m is a number of monitors). Now it is important to find a suitable marking of the monitors V_{S_i} . Namely, an inadequate setting of marking of monitors may cause other deadlocks in the controlled plant. In general, for setting the marking of monitors V_{S_i} , $i = 1, n_m$ are valid the following general rules.

Let $S = \{p_i, p_j, \dots, p_k\}$ be a strict minimal siphon (SMS) of an original (uncontrolled) ordinary net system (N_0, M_0) , where $N_0 = (P_0, T_0, F_0)$. Add a monitor V_S to N_0 to make the vector $\mathbf{I} = (0, \dots, 1_i, \dots, 1_j, \dots, 1_k, \dots, 0, -1)^T$ be a P -invariant of a new (controlled) net system (N_1, M_1) . Here, $\forall p \in P_0 \setminus S, I(p) = 0, I(V_S) = -1, \forall p \in P_0, M_1(p) = M_0(p)$, and $\mathbf{N}_1 = [\mathbf{N}_0^T | \mathbf{L}_{V_S}^T]^T$, where \mathbf{L}_{V_S} is a row vector corresponding to adding the place V_S . Let $M_1(V_S) = M_0(S) - \zeta_S$, where $1 \leq \zeta_S \leq M_0(S)$. Then S is an invariant-controlled SMS. Hence, it is always marked at any reachable marking of the net system (N_1, M_1) . Namely, \mathbf{I} is a P -invariant and $\forall p \in (P_0 \cup \{V_S\}) \setminus S, I(p) < 0$. Note that $\mathbf{I}^T \cdot \mathbf{M}_1 = \mathbf{I}^T \cdot \mathbf{M}_0 = M_0(S) - M_1(V_S) = \zeta_S > 0$. Thus, S is an invariant-controlled siphon.

A siphon S is controlled in a net (N, M_0) iff $\forall M \in R(N, M_0), M(S) > 0$. Thus, any siphon that contains a marked trap is controlled. Namely, the marked trap can never be emptied. In ordinary PN a controlled siphon does not cause any deadlock.

A siphon S in an ordinary PN (N, M_0) is [7] invariant-controlled by P -invariant I under M_0 iff $\mathbf{I}^T \cdot \mathbf{M}_0 > 0$ and $\forall p \in P \setminus S, I(p) \leq 0$, or equivalently $\mathbf{I}^T \cdot \mathbf{M}_0 > 0$ and $\|I\|^+ = p \in P | I(p)$ is the positive support of I .

More succinctly said, a siphon S is controlled if it can never be emptied, and invariant-controlled by P -invariant I if $\mathbf{I}^T \cdot \mathbf{M}_0 > 0$ and $\|I\|^+ \subseteq S$.

So that, to guarantee that a siphon S is always marked in a net system, it is necessary to keep at least one token being present at S at any reachable marking of the net system. Suppose that it was found such a control of a siphon S that S will never be emptied. Let ζ_S is the least number of tokens being present at S . Such ζ_S is called the siphon control depth variable. It is clear that the larger ζ_S is, the more behavior of the modeled system will be restricted. However, it may means that more reachable states will be forbidden than it is necessary. Therefore, it is suitable when the siphon control depth variable is as small as possible, i.e., 1, if possible.

Setting the Marking of Monitors in Example 2

It is easy to check from Figure 3 with taking into account Figure 4 that $I = \{p_2, p_4, p_5, p_6\} - V_S$ is P -invariant of N_1 . Clearly, $S = \{p_2, p_4, p_5, p_6\}$ is invariant-controlled by I , since $\|I\|^+ = S$ and $M_0(S) - M_1(V_S) = 4 - 2 > 0$. Here, $\|I\|^+$, is the positive support of I .

5. More Complicated Examples

5.1. Example 3

Consider a PN model in the form presented in Figure 5.

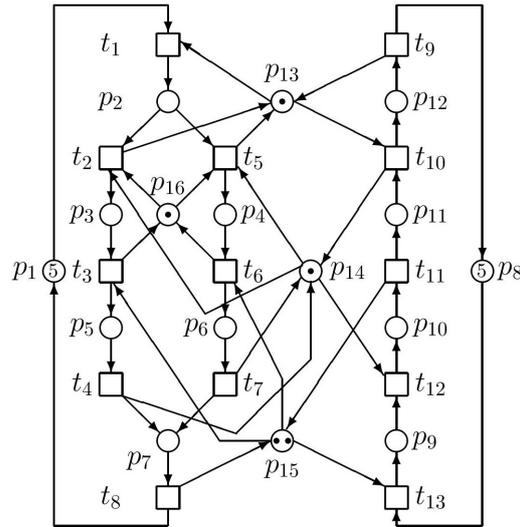


Figure 5. The bigger PN model.

This PN model has RT with 77 nodes (including M_0). There are three deadlocks in the PN, namely

$$M_{33} = (3, 1, 1, 0, 0, 0, 0, 3, 2, 0, 0, 0, 0, 0, 0, 0)^T \tag{34}$$

$$M_{36} = (3, 1, 0, 1, 0, 0, 0, 3, 2, 0, 0, 0, 0, 0, 0, 0)^T \tag{35}$$

$$M_{44} = (4, 1, 0, 0, 0, 0, 0, 2, 2, 0, 1, 0, 0, 0, 0, 1)^T \tag{36}$$

Because of the large RT it is difficult or practically impossible (by commonly available means) to analyze RT in order to find the matrix L and the vector b in (5). It remains to use the approach based on siphon and traps. Using the Matlab-based tool in [30] we obtain the following minimal siphons and traps

$S_1 = \{p_3, p_4, p_{16}\}$	$Tr_1 = \{p_3, p_4, p_{16}\}$
$S_2 = \{p_2, p_{12}, p_{13}\}$	$Tr_2 = \{p_2, p_{12}, p_{13}\}$
$S_3 = \{p_8, p_9, p_{10}, p_{11}, p_{12}\}$	$Tr_3 = \{p_2, p_{10}, p_{11}, p_{13}, p_{14}\}$
$S_4 = \{p_5, p_6, p_7, p_9, p_{10}, p_{15}\}$	$Tr_4 = \{p_8, p_9, p_{10}, p_{11}, p_{12}\}$
$S_5 = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$	$Tr_5 = \{p_5, p_6, p_7, p_9, p_{10}, p_{15}\}$
$S_6 = \{p_3, p_4, p_5, p_6, p_{12}, p_{13}, p_{14}\}$	$Tr_6 = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$
$S_7 = \{p_3, p_4, p_5, p_6, p_{10}, p_{11}, p_{14}\}$	$Tr_7 = \{p_3, p_4, p_5, p_6, p_{10}, p_{11}, p_{14}\}$
$S_8 = \{p_5, p_6, p_7, p_{10}, p_{11}, p_{14}, p_{15}\}$	$Tr_8 = \{p_2, p_9, p_{10}, p_{13}, p_{14}, p_{15}, p_{16}\}$
$S_9 = \{p_5, p_6, p_7, p_{10}, p_{12}, p_{13}, p_{14}, p_{15}\}$	$Tr_9 = \{p_2, p_5, p_6, p_9, p_{10}, p_{13}, p_{14}, p_{15}\}$
	$Tr_{10} = \{p_3, p_4, p_5, p_6, p_9, p_{10}, p_{14}, p_{15}\}$

Because $S_1 = Tr_1, S_2 = Tr_2, S_3 = Tr_4, S_4 = Tr_5, S_5 = Tr_6$, and $S_7 = Tr_7$, there are three minimal siphons where emptying must be prevented, namely S_6, S_8 , and S_9 . Rename S_6 to be S_1, S_8 to be S_2 and S_9 to be S_3 . Consequently, we have

$$S = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{37}$$

$$N_c = S.N = \begin{pmatrix} -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \end{pmatrix} \tag{38}$$

where the incidence matrix of the PN model is the following

$$N = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{39}$$

The uncontrolled PN model together with its supervisor is displayed in Figure 6. This form of the mutual interaction between the supervisor and the PN model is used in order to avoid intricate interconnections between the uncontrolled model and the supervisor in the same figure. Marking of monitors $V_{S_1} - V_{S_3}$ were established by the relation $M_0(S_i) - M_1(V_{S_i}) > 0$ for $i = 1, 2, 3$. Because $M_0(S_1) = 2$ it is sufficient to put $M_1(V_{S_1}) = 1$. Similarly, because $M_0(S_2) = 3$ it is sufficient to put $M_1(V_{S_2}) = 2$ and because $M_0(S_3) = 4$ it is sufficient to put $M_1(V_{S_3}) = 2$. In such a way controlled PN model (30) has RT with 56 nodes (it creates the state space of reachable markings) and no deadlocks. In case of putting $M_1(V_{S_3}) = 3$ a new deadlock would occur in controlled system, namely $\mathbf{M}_{14} = (4, 1, 0, 0, 0, 0, 0, 0, 3, 2, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0)^T$.

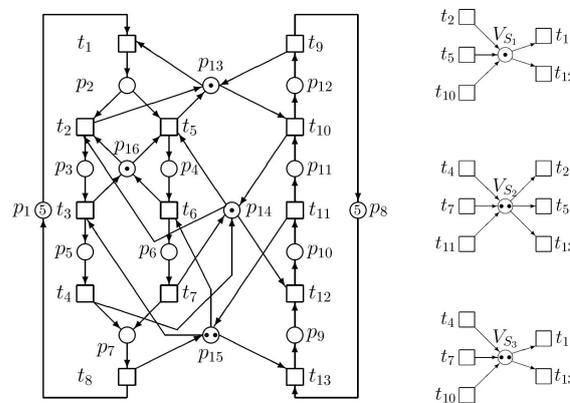


Figure 6. The PN model (left) with its supervisor (right) created by three monitors and their interconnections with the uncontrolled model.

5.2. Example 4

Consider the PN model given in Figure 7. There are 16 places and 12 transitions.

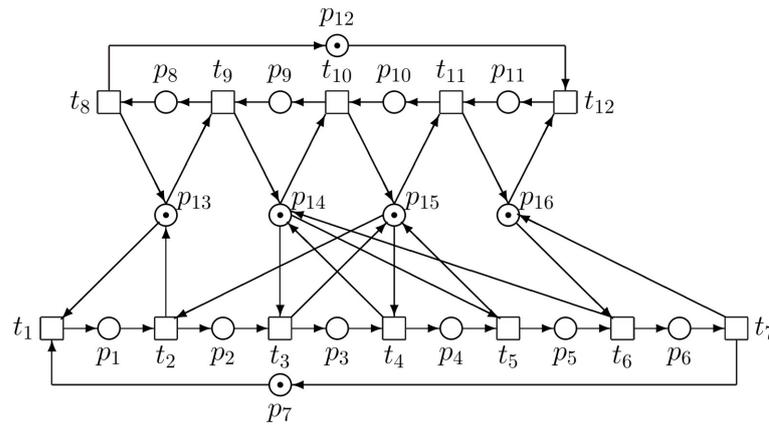


Figure 7. The PN model.

The transpose of the incidence matrix is the following

$$N^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 \end{pmatrix}$$

The siphons and traps are the following.

$$\begin{aligned} S_1 &= \{p_{11}, p_{16}, p_6\} & Tr_1 &= \{p_{11}, p_{16}, p_6\} \\ S_2 &= \{p_1, p_{13}, p_8\} & Tr_2 &= \{p_1, p_{13}, p_8\} \\ S_3 &= \{p_{10}, p_{15}, p_2, p_4\} & Tr_3 &= \{p_{10}, p_{14}, p_{15}, p_2\} \\ S_4 &= \{p_{14}, p_3, p_5, p_9\} & Tr_4 &= \{p_{10}, p_{15}, p_2, p_4\} \\ S_5 &= \{p_{14}, p_{15}, p_5, p_9\} & Tr_5 &= \{p_{14}, p_3, p_5, p_9\} \\ S_6 &= \{p_{14}, p_{15}, p_{16}, p_6, p_9\} & Tr_6 &= \{p_1, p_{10}, p_{13}, p_{14}, p_{15}\} \\ S_7 &= \{p_{10}, p_{11}, p_{12}, p_8, p_9\} & Tr_7 &= \{p_{11}, p_{14}, p_{15}, p_{16}, p_2\} \\ S_8 &= \{p_{13}, p_{14}, p_{15}, p_5, p_8\} & Tr_8 &= \{p_{10}, p_{11}, p_{12}, p_8, p_9\} \\ S_9 &= \{p_{13}, p_{14}, p_{15}, p_{16}, p_6, p_8\} & Tr_9 &= \{p_1, p_{11}, p_{13}, p_{14}, p_{15}, p_{16}\} \\ S_{10} &= \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\} & Tr_{10} &= \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\} \end{aligned}$$

Here, $S_1 = Tr_1, S_2 = Tr_2, S_4 = Tr_5, S_7 = Tr_8, S_{10} = Tr_{10}$. Consequently, $S_1, S_2, S_4, S_7, S_{10}$ cannot be emptied. It is sufficient to prevent S_3, S_5, S_6, S_8, S_9 before emptying. Rename them to be, respectively, S_1, S_2, S_3, S_4, S_5 . Hence,

$$S = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{40}$$

$$N_c = S.N = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 \end{pmatrix} \quad (41)$$

Thus, we obtain five monitors $V_{S_1} - V_{S_5}$ (creating the supervisor) and their interconnections with the uncontrolled PN model. In order to establish marking of these monitors the following settings are necessary: $M_0(S_i) - M_1(V_{S_i}) > 0, i = 1, \dots, 5$. Because $M_0(S_1) = 1, M_1(V_{S_1}) = 0$. Analogically, $M_0(S_2) = 2, M_1(V_{S_2}) = 1; M_0(S_3) = 3, M_1(V_{S_3}) = 2; M_0(S_4) = 3, M_1(V_{S_4}) = 2; M_0(S_5) = 4, M_1(V_{S_5}) = 3$. With markings of monitors set in such a way, we obtain the controlled PN model without any deadlocks. RT of the controlled PN model has 54 nodes including the initial marking $M_0 = (0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1 | 0, 1, 2, 2, 3)^T$. These nodes represent the space of reachable markings of the controlled PN model. The interconnection between original uncontrolled PN model and the supervisor represented by the set of monitors is realized by means of the incidence matrix N_c , namely by N_c^+ and N_c^- . The supervisor is displayed in Figure 8 (right) together with the uncontrolled system (left).

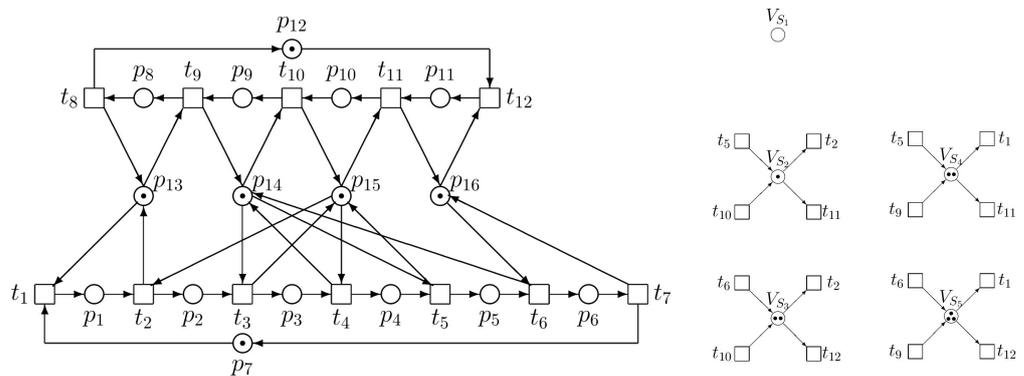


Figure 8. The PN model (left) with its supervisor (right) created by five monitors and their interconnections with the uncontrolled model. The monitor V_{S_1} is not connected as it is clear from the incidence matrix N_c in (41).

5.3. Example 5

Consider the PN model displayed in Figure 9.

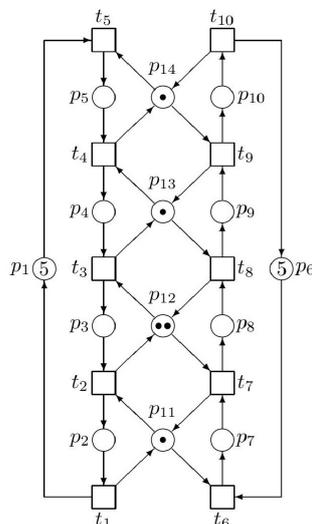


Figure 9. The PN model.

Its RT has 119 nodes (reachable markings) including the initial marking $M_0 = (5, 0, 0, 0, 0, 5, 0, 0, 0, 0, 1, 2, 1, 1)^T$. There are five deadlocks in the model, namely $M_{76}, M_{85}, M_{91}, M_{96}, M_{100}$ as follows,

$$M_{76} = (3, 0, 0, 1, 1, 2, 1, 2, 0, 0, 0, 0, 0, 0)^T \tag{42}$$

$$M_{85} = (2, 0, 1, 1, 1, 3, 1, 1, 0, 0, 0, 0, 0, 0)^T \tag{43}$$

$$M_{91} = (4, 0, 0, 0, 1, 1, 1, 2, 1, 0, 0, 0, 0, 0)^T \tag{44}$$

$$M_{96} = (1, 0, 2, 1, 1, 4, 1, 0, 0, 0, 0, 0, 0, 0)^T \tag{45}$$

$$M_{100} = (3, 0, 1, 0, 1, 2, 1, 1, 1, 0, 0, 0, 0, 0)^T \tag{46}$$

and the following siphon and traps

$$\begin{aligned} S_1 &= \{p_2, p_7, p_{11}\} & Tr_1 &= \{p_2, p_7, p_{11}\} \\ S_2 &= \{p_3, p_8, p_{12}\} & Tr_2 &= \{p_3, p_8, p_{12}\} \\ S_3 &= \{p_4, p_9, p_{13}\} & Tr_3 &= \{p_4, p_9, p_{13}\} \\ S_4 &= \{p_5, p_{10}, p_{14}\} & Tr_4 &= \{p_5, p_{10}, p_{14}\} \\ S_5 &= \{p_2, p_8, p_{11}, p_{12}\} & Tr_5 &= \{p_3, p_7, p_{11}, p_{12}\} \\ S_6 &= \{p_3, p_9, p_{12}, p_{13}\} & Tr_6 &= \{p_4, p_8, p_{12}, p_{13}\} \\ S_7 &= \{p_4, p_{10}, p_{13}, p_{14}\} & Tr_7 &= \{p_5, p_9, p_{13}, p_{14}\} \\ S_8 &= \{p_1, p_2, p_3, p_4, p_5\} & Tr_8 &= \{p_1, p_2, p_3, p_4, p_5\} \\ S_9 &= \{p_2, p_9, p_{11}, p_{12}, p_{13}\} & Tr_9 &= \{p_4, p_7, p_{11}, p_{12}, p_{13}\} \\ S_{10} &= \{p_3, p_{10}, p_{12}, p_{13}, p_{14}\} & Tr_{10} &= \{p_5, p_8, p_{12}, p_{13}, p_{14}\} \\ S_{11} &= \{p_6, p_7, p_8, p_9, p_{10}\} & Tr_{11} &= \{p_6, p_7, p_8, p_9, p_{10}\} \\ S_{12} &= \{p_2, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\} & Tr_{12} &= \{p_5, p_7, p_{11}, p_{12}, p_{13}, p_{14}\} \end{aligned}$$

As we can see, $S_1 = Tr_1, S_2 = Tr_2, S_3 = Tr_3, S_4 = Tr_4, S_8 = Tr_8, S_{11} = Tr_{11}$. As we already know, if a siphon contains a marked trap, it will never become empty. Therefore, we do not have to take the listed siphons into account. When synthesizing the supervisor, it is sufficient to use the remaining six siphons, i.e., $S_5, S_6, S_7, S_9, S_{10}, S_{12}$. After elimination of siphons being consensual to traps, rename the residual siphons in ascending order starting from S_1 , i.e., $S_1 = S_5, S_2 = S_6, S_3 = S_7, S_4 = S_9, S_5 = S_{10},$ and $S_6 = S_{12}$. Thus, the matrix of siphons will have the following shape

$$S_m = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{47}$$

Since the incidence matrix N of the uncontrolled PN model directly follows from the model structure introduced in Figure 9, the incidence matrix of the supervisor will be as follows

$$N_c = S_m \cdot N = \begin{pmatrix} 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{48}$$

To set marking of particular monitors V_{S_1}, \dots, V_{S_6} , we will use the relation $M_0(S_i) - M_1(V_{S_i}) > 0$ for $i = 1, \dots, 6$. Because $M_0(S_1) = M_0(S_2) = 3, M_0(S_3) = 2, M_0(S_4) = M_0(S_5) = 4,$ and $M_0(S_6) = 5$. Hence, for $M_1(V_{S_1}) = M_1(V_{S_2}) = 2, M_1(V_{S_3}) = 1, M_1(V_{S_4}) = M_1(V_{S_5}) = 2,$ and $M_1(V_{S_6}) = 2$ there are no deadlocks in the supervised PN model. The supervisor (the set of monitors) is displayed in Figure 10 right, in the neighborhood of the unsupervised PN model.

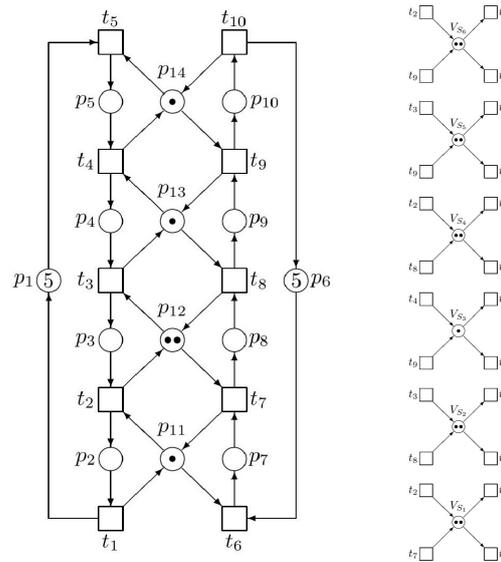


Figure 10. The unsupervised PN model together with its supervisor.

5.4. Example 6

In order to show that there may occur computational problems in case of PN models which need not be neither large nor structurally complicated consider the PN model in Figure 11.

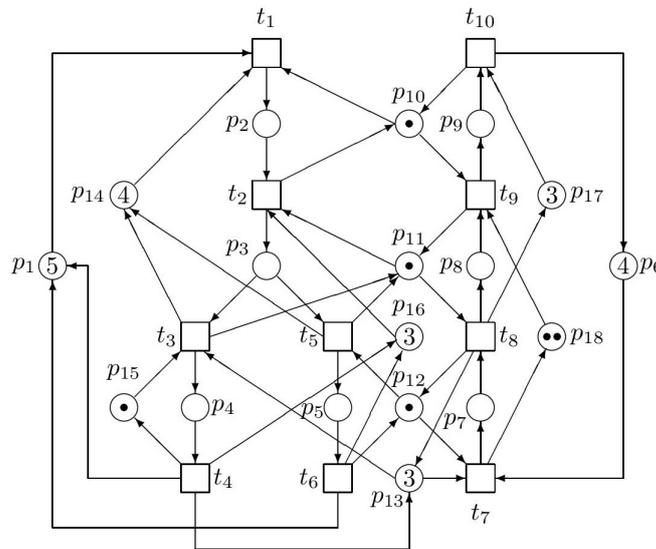


Figure 11. The PN model.

In this case, it was not possible to calculate siphons and traps by the tool offered in [30] on PC (with Intel(R) Core(TM), i7-10700 CPU @ 2.90 GHz 2.90 GHz with 16 GB RAM and 64-bit operating system Windows 11) even after tens of hours (more than 24) of computation. So let us try to go back to the approach based on *P*-invariants RT of this uncontrolled PN model has 40 nodes (including the initial marking). There is only one deadlock $\mathbf{M}_{17} = (4, 1, 0, 0, 0, 2, 1, 1, 0, 0, 0, 0, 2, 3, 1, 3, 4, 4)^T$, as we can see from the segment of RT displayed in Figure 12.

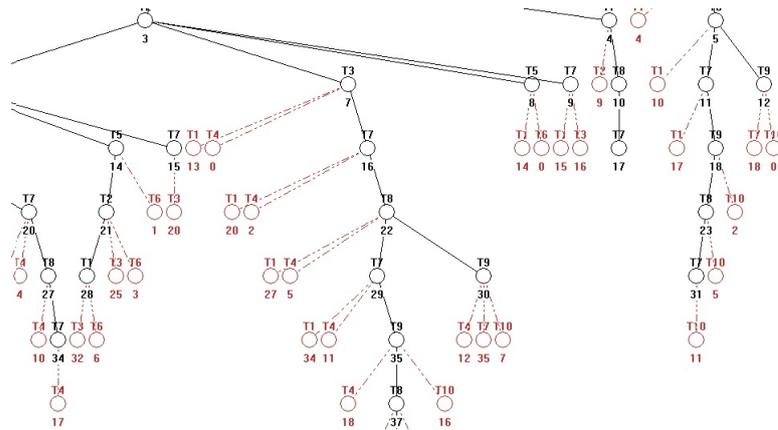


Figure 12. The segment of the PN model RT containing the deadlock and access paths to it.

$$M_{10} \xrightarrow{t_7} M_{17} \tag{49}$$

$$M_4 \xrightarrow{t_8} M_{10} \xrightarrow{t_7} M_{17} \tag{50}$$

$$M_5 \xrightarrow{t_1} M_{10} \xrightarrow{t_7} M_{17} \tag{51}$$

$$M_5 \xrightarrow{t_7} M_{11} \xrightarrow{t_1} M_{17} \tag{52}$$

$$M_{27} \xrightarrow{t_7} M_{34} \xrightarrow{t_4} M_{17} \tag{53}$$

Here,

$$M_4 = (4, 1, 0, 0, 0, 3, 1, 0, 0, 0, 1, 0, 2, 3, 1, 3, 3, 3)^T \tag{54}$$

$$M_5 = (5, 0, 0, 0, 0, 3, 0, 1, 0, 1, 0, 1, 3, 4, 1, 3, 4, 3)^T \tag{55}$$

$$M_{10} = (4, 1, 0, 0, 0, 3, 0, 1, 0, 0, 0, 1, 3, 3, 1, 3, 4, 3)^T \tag{56}$$

$$M_{11} = (5, 0, 0, 0, 0, 2, 1, 1, 0, 1, 0, 0, 2, 4, 1, 3, 4, 4)^T \tag{57}$$

$$M_{27} = (3, 1, 0, 1, 0, 3, 0, 1, 0, 0, 0, 1, 2, 3, 0, 2, 4, 3)^T \tag{58}$$

$$M_{34} = (3, 1, 0, 1, 0, 2, 1, 1, 0, 0, 0, 0, 1, 3, 0, 2, 4, 4)^T \tag{59}$$

After a short analysis of RT, we can set the constraints in the form:

$$p_7 + p_{11} \leq 2 \tag{60}$$

$$p_7 + p_{18} \leq 3 \tag{61}$$

$$p_2 + p_7 \leq 1 \tag{62}$$

It means that

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}; \mathbf{b} = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} \tag{63}$$

$$N_s = -L.N = \begin{pmatrix} 0 & 1 & -1 & 0 & -1 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 1 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \end{pmatrix} \tag{64}$$

$${}^sM_0 = \mathbf{b} - L.M_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{65}$$

Such supervisor indeed prevents the deadlock but RT of the supervised system has only 16 nodes. It means that its state space is (in comparison with the state space of uncontrolled system having RT with 40 nodes) considerably limited. Unfortunately, the right section (sub-net) in Figure 11 consisting of $p_6, p_7, p_8, p_9, p_{17}, p_{18}$ does not operate because it is eliminated by such supervisor.

Therefore, we have to find another supervisor which will be able to connect also this section in the activity. Consider the following constraints

$$p_2 + p_7 + p_8 + p_9 \leq 1 \tag{66}$$

$$p_2 + p_3 + p_7 + p_8 \leq 1 \tag{67}$$

$$p_2 + p_3 + p_5 + p_7 \leq 1 \tag{68}$$

$$p_2 + p_3 + p_4 + p_7 \leq 3 \tag{69}$$

$$p_2 + p_3 \leq 4 \tag{70}$$

$$p_2 + p_3 + p_4 \leq 1 \tag{71}$$

$$p_2 + p_3 + p_4 + p_5 \leq 3 \tag{72}$$

$$p_7 + p_8 + p_9 \leq 3 \tag{73}$$

$$p_7 + p_8 \leq 2 \tag{74}$$

or in the matrix form

$$\mathbf{L} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}; \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 3 \\ 4 \\ 1 \\ 3 \\ 3 \\ 3 \\ 2 \end{pmatrix} \tag{75}$$

$$\mathbf{N}_s = -\mathbf{L.N} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & 0 & 1 & -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 & -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \end{pmatrix}; {}^s\mathbf{M}_0 = \mathbf{b} - \mathbf{L.M}_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 3 \\ 4 \\ 1 \\ 3 \\ 3 \\ 3 \\ 2 \end{pmatrix} \tag{76}$$

Applying the supervisor to the uncontrolled PN we obtain the controlled model without any deadlocks. Its RT, expressing the space of reachable states, has 11 nodes as it can be seen in Figure 13. In the same Figure right is displayed also the supervisor consisting of 9 monitors. As we can see, both of the sections (sub-nets) are active in this case.

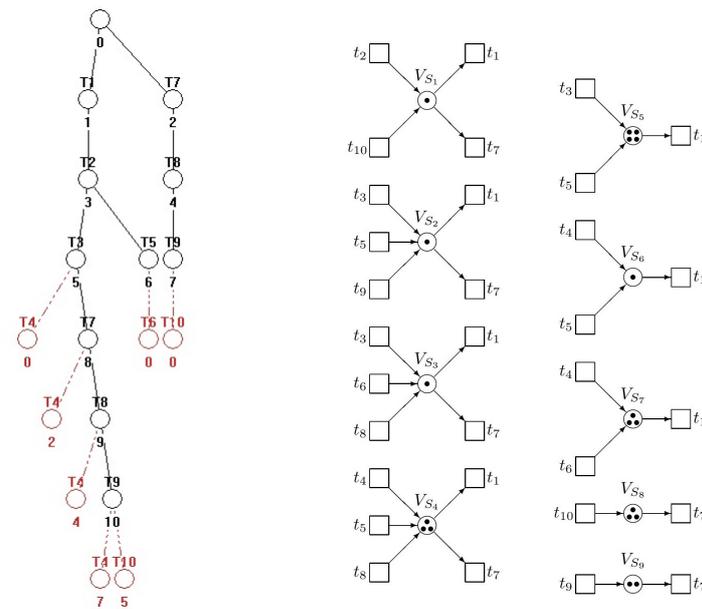


Figure 13. The RT of the controlled PN model (left). No deadlocks are there. Both of sections (sub-nets) are active. The supervisor consisting of nine monitors (right).

6. Discussion

Two methods (approaches) how to avoid deadlocks in industrial multi-agent systems were presented in this paper, namely: (i) the approach based on *P*-invariants, and (ii) the approach based on siphons and traps. Let us compare them now.

Both approaches have their advantages and disadvantages.

The former approach represents the procedure in exact analytical terms. The supervisor synthesis is clear and simple. It is suitable especially for the S^3PR paradigm of AMS. Thorough analysis of RT makes possible to find conditions how to mutually eliminate certain states (markings). In such a way it is possible to eliminate deadlocks. However, in a more complicated structure of the PN model with very branched RT the choice of the restriction inequality (5) (especially the matrix *L* and the vector *b*) may be intricate or even impossible. Another disadvantage is that RT depends on the initial marking M_0 of the uncontrolled PN. For another M_0 RT is different.

The latter approach has not so exact analytically expressed procedure. This is due to the fact that siphons and traps are structural entities of the PN model. Their computation (notwithstanding that it may be realized by different algorithms) may sometimes take a very long time, especially in case of structurally complicated PN models. An advantage of this approach is that calculation of siphons and traps does not depend on the initial state of the PN model. This approach is able to deal with both paradigms of AMS— S^3PR and S^4PR .

As a summary it can be said that the weak point (shortcoming, weakness) in both approaches is computational complexity—at computing RT and handling it as well as at computation of siphons and traps—especially at large-scale and structurally complicated PN models.

Despite what has been said, the approach based on siphons and traps seems to be upon the whole more advantageous than the approach based on *P*-invariants. Apart from the time-consuming calculation of siphons and traps, it is also simpler. However, as it was demonstrated by the Example 6 in the Section 5.4, computational problems may be huge also in relatively simple and not very large PN models.

7. Conclusions

The presented paper is, in a broader sense, an overview paper concerning the deadlock prevention and avoidance in industrial MAS in general. RAS are very important subclass of AMS. Dealing with deadlocks in RAS has been the main topic of this paper. Benefits

following from this are that deadlock free RAS can be designed off-line, preliminary to the construction of real manufacturing systems, still before their actual deployment in practice. Hence, this rapidly decreases a risk of defects in operation of AMS/RAS and prevents shutdowns of them. Thus, it is possible to avoid significant economic losses. Of course, another defects unrelated to deadlocks (e.g., some external disturbances) cannot be prevented in such a way. Attention has been paid to the present most frequently used RAS, in particular to their paradigms S^3PR and S^4PR .

A wide review of literature was introduced. A general view on formal methods used in AMS was presented in Section 1.1. A short comment about the agent cooperation and negotiation, as well as on reentering in AMS was introduced in Section 1.2. The most used Petri net-based paradigms of RAS were adduced in Section 2.1. A literature overview about deadlock prevention and avoiding was referred to in Section 2.2. Setting marking of monitors in case of the siphon-based method was described in details in Section 4.2. Two simple illustration examples introduced in Section 3 (Example 1) and in Section 4 (Example 2) were supplemented by four more complicated illustrative examples in Section 5 (Example 3–Example 6).

Two basic deadlock prevention techniques using Petri nets were presented in mathematical details, namely P -invariant-based method and the method based on siphons and traps. Although in Section 6 a rough comparison of both methods was introduced, in this closing section it is necessary to point out the principled difference between them. In Table 1, the main characteristics of both methods are introduced. The ability to address the deadlock prevention in the S^3PR and S^4PR paradigms of RAS is also included there. After comparing those characteristics, the method based on siphons and traps unambiguously appears more advantageous than the method based on P -invariants. The same result follows also from Table 2 (where disadvantageous of both methods are introduced) because long lasting computations can be accelerated, e.g., by using more powerful computing technique and/or by finding algorithms with a less computational complexity.

Table 1. Characteristics of both methods.

<i>P</i>-Invariant-Based Method	Siphon and Trap-Based Method
simple analytic method	Matlab-based computational method
it needs additional elaboration of RT it is more labored	it does need any additional elaboration —
it has analytical setting of the initial marking of monitors	it needs additional setting of the initial marking of monitors
—	it yields a possibility to control elementary siphons and to deal with dependent siphons by means of setting the siphon control depth variables
it is suitable above all for S^3PR paradigm of AMS	it is suitable for S^3PR as well as for S^4PR paradigms of AMS
in the S^4PR paradigm it cannot eliminate the origin of new deadlocks and requires new iterations of the deadlock prevention	it is able to preclude origin of new deadlocks in the S^4PR paradigm

Table 2. Disadvantageous of both methods.

<i>P</i>-Invariant-Based Method	Siphon and Trap-Based Method
possible computational problems at finding RT necessity to additionally analyze RT	often long lasting computation of siphons —
dependence on the initial state of the PN model	—

However, in spite of all, it must be said that even the method based on P -invariants should not be completely damned, especially in case of S^3PR paradigm of AMS. As it was shown in Example 6 in the Section 5.4 this method has helped us to resolve a case where the siphon-based method was not able to give a result in a reasonable time.

Funding: This research received no external funding.

Data Availability Statement: Not Applicable, the study does not report any data.

Acknowledgments: The author thanks for the partial support of the VEGA Agency (under grant No. 2/0020/21).

Conflicts of Interest: The author declares no conflict of interest.

References

1. Mourtzis, D. Simulation in the Design and Operation of Manufacturing Systems: State of the Art and New Trends. *Int. J. Prod. Res.* **2020**, *58*, 1927–1949. [\[CrossRef\]](#)
2. Mourtzis, D. (Ed.) *Design and Operation of Production Networks for Mass Personalization in the Area of Cloud Technology*; Elsevier Inc.: Amsterdam, The Netherlands, 2022; ISBN 978-0-12-823657-4. [\[CrossRef\]](#)
3. El Maraghy, H.; Monostori, L.; Schuh, G.; El Maraghy, W. Evolution and future of manufacturing systems. *CIRP Ann. Manuf. Technol.* **2021**, *70*, 635–658. [\[CrossRef\]](#)
4. Murata, T. Petri Nets: Properties, Analysis and Applications. *Proc. IEEE* **1989**, *77*, 541–580. [\[CrossRef\]](#)
5. Reisig, W. *Petri Nets*; Springer: Berlin/Heidelberg, Germany, 1985.
6. Iordache, M.V.; Antsaklis, P.J. *Supervisory Control of Concurrent Systems: A Petri Net Structural Approach*; Birkhäuser: Basel, Switzerland, 2006.
7. Li, Z.W.; Zhou, M.C. *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*; Springer: London, UK, 2009.
8. Wang, Y.; Kelly, T.; Kudlur, M.; Mahlke, S.; Lafortune, S. The Application of Supervisory Control to Deadlock Avoidance in Concurrent Software. In Proceedings of the 9th IEEE/IFAC International Workshop on Discrete Event Systems—WODES’08, Goteborg, Sweden, 28–30 May 2008; pp. 287–292.
9. Iordache M.V. Methods for the Supervisory Control of Concurrent Systems Based on Petri Net Abstractions. Ph.D. Thesis, University of Notre Dame, Notre Dame, IN, USA, 2003.
10. Duhaut, D.; Carrillo, E.; Saint-Aimé, S. Avoiding Deadlock in Multi-agent Systems. In Proceedings of the 2007 IEEE International Conference on Systems, Man and Cybernetics, Montreal, QC, Canada, 7–10 October 2007; pp. 1642–1647. [\[CrossRef\]](#)
11. Mani, N.; Garousi, V.; Far, B.H. Monitoring Multi-Agent Systems for deadlock detection based on UML models. In Proceedings of the 2008 Canadian Conference on Electrical and Computer Engineering (CCECE), Niagara Falls, ON, Canada, 4–7 May 2008; pp. 1611–1616. [\[CrossRef\]](#)
12. Liu, G.Y.; Barkaoui, K. A Survey of Siphons in Petri Nets. *Inf. Sci.* **2016**, *363*, 198–220. [\[CrossRef\]](#)
13. Li, X.Y.; Liu, G.Y.; Li, Z.W.; Wu, N.Q.; Barkaoui, K. Elementary Siphon-Based Robust Control for Automated Manufacturing Systems with Multiple Unreliable Resources. *IEEE Access* **2019**, *7*, 21006–21019. [\[CrossRef\]](#)
14. Yang, Y.; Lu, W.; Xing, W.; Wang, L.; Che, X.; Chen, L. Detecting and Resolving Deadlocks in Mobile Agent Systems. *J. Vis. Lang. Comput.* **2017**, *42*, 23–30. [\[CrossRef\]](#)
15. Luo, J.; Liu, Z.; Zhou, M.; Xing, K.; Wang, X.; Li, X.; Liu, H. Robust deadlock control of automated manufacturing systems with multiple unreliable resources. *Inf. Sci.* **2019**, *479*, 401–415. [\[CrossRef\]](#)
16. Čapkovič, F. Modelling and Control of Resource Allocation Systems within Discrete Event Systems by Means of Petri Nets—Part 1: Invariants, Siphons and Traps in Deadlock Avoidance. *Comput. Inform.* **2021**, *40*, 648–689_3_648. [\[CrossRef\]](#)
17. Ezpeleta, J.; Colom, J.M.; Martinez, J. A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems. *IEEE Trans. Robot. Autom.* **1995**, *11*, 173–184. [\[CrossRef\]](#)
18. Fanti, M.P.; Maione, B.; Turchiano, B. Comparing Digraph and Petri Net Approaches to Deadlock Avoidance in FMS. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2000**, *30*, 783–798. [\[CrossRef\]](#)
19. Gebraeel, N.Z.; Lawley, M.A. Deadlock detection, prevention, and avoidance for automated tool sharing systems. *IEEE Trans. Robot. Autom.* **2001**, *17*, 342–356. [\[CrossRef\]](#)
20. Tricas, F. Deadlock Analysis, Prevention and Avoidance in Sequential Resource Allocation Systems. Ph.D. Thesis, Department of Informatics and System Engineering, University of Zaragoza, Zaragoza, Spain, 2003 [\[CrossRef\]](#)
21. Liu, G.Y. Supervisor Synthesis for Automated Manufacturing Systems Based on Structure Theory of Petri Nets. Ph.D. Thesis, École Doctorale Informatique, Télécommunications et Electronique de Paris, Paris, France, 2014
22. Campos, J.; Seatzu, C.; Xie, X. (Eds.) *Formal Methods in Manufacturing*; CRC Press: Boca Raton, FL, USA; London, UK; Taylor & Francis Group: New York, NY, USA, 2017.
23. Čapkovič, F. Cooperation and Negotiation of Agents by Means of Petri Net-based Models. In Proceedings of the 2012 IEEE 17th Conference Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, 27–30 August 2012; pp. 256–261. [\[CrossRef\]](#)

24. Ćapković, F.; Jotsov, V. A System Approach to Agent Negotiation and Learning. In *Intelligent Systems: From Theory to Practice; Studies in Computational Intelligence*; Sgurev, V., Hadjiski, M., Kacprzyk, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 299, pp. 133–160. [CrossRef]
25. Sindičić, I.; Bogdan, S.; Petrović, T. Resource Allocation in Free-Choice Multiple Reentrant Manufacturing Systems Based on Machine-Job Incidence Matrix. *IEEE Trans. Ind. Inform.* **2011**, *7*, 105–114.
26. Guan, X.; Li, Y.; Xu, J.; Wang, C.; Wang, S. A Literature Review of Deadlock Prevention Policy Based on Petri Nets for Automated Manufacturing Systems. *Int. J. Digit. Content Technol. Its Appl. (JDCTA)* **2012**, *6*, 426–433. [CrossRef]
27. Yue, H.; Xing, K.Y.; Hu, H.S.; Wu, W.M.; Su, H.Y. Petri-Net-Based Robust Supervisory Control of Automated Manufacturing Systems. *Control Eng. Pract.* **2016**, *54*, 176–189. [CrossRef]
28. Farooq, A.; Huang, H.; Wang, X.L. Petri Net Modeling and Deadlock Analysis of Parallel Manufacturing Processes with Shared-Resources. *J. Syst. Softw.* **2010**, *83*, 675–688. [CrossRef]
29. Hu, H.; Liu, Y.; Yuan, L. Supervisor Simplification in FMSs: Comparative Studies and New Results Using Petri Nets. *IEEE Trans. Control Syst. Technol.* **2016**, *24*, 81–95. [CrossRef]
30. Davidrajuh, R. *Modeling Discrete-Event Systems with GPenSIM: An Introduction*; SpringerBriefs in Applied Sciences and Technology; Springer: Cham, Switzerland, 2018; 155p, ISBN 978-3-319-73102-5. Available online: <https://link.springer.com/content/pdf/10.1007/978-3-319-73102-5.pdf?pdf=button> (accessed on 3 January 2023). [CrossRef]
31. Iordache, M.V.; Antsaklis, P.J. Supervision Based on Place Invariants: A Survey. *Discret. Event Dyn. Syst.* **2006**, *16*, 4451–4492. [CrossRef]
32. Moody, J.; Antsaklis, P. Petri Net Supervisors for DES with Uncontrollable and Unobservable Transitions. *IEEE Trans. Autom. Control* **2000**, *45*, 462–476. [CrossRef]
33. Wang, Y.; Lafortune, S. The Theory of Deadlock Avoidance via Discrete Control. In Proceedings of the 36th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages—POPL’09, Newsletter ACM SIGPLAN Notices, Savannah, GA, USA, 21–23 January 2009; Volume 44, pp. 252–263. [CrossRef]
34. Reveliotis, S. Implicit Siphon Control and its Role in the Liveness Enforcing Supervision of Sequential Resource Allocation Systems. *IEEE Trans. Syst. Man Cybern. Part A* **2007**, *37*, 319–328.
35. Chao, D.Y. Max'-controlled Siphons for Liveness of S^3PGR2 . *IET Control Theory Appl.* **2007**, *1*, 933–936.:20060275. [CrossRef]
36. Chao, D.Y.; Pan, Y.L. Uniform Formulas for Compound Siphons, Complementary Siphons and Characteristic Vectors in Deadlock Prevention of Flexible Manufacturing Systems. *J. Intell. Manuf.* **2015**, *26*, 13–23. [CrossRef]
37. Hou, Y.F.; Zhao, M.; Liu, D.; Hong, L. An Efficient Siphon-Based Deadlock Prevention Policy for a Class of Generalized Petri Nets. *Discret. Dyn. Nat. Soc.* **2016**, *2016*, 8219424. [CrossRef]
38. Liu, G.; Li, Z.; Zhong, C. New controllability condition for siphons in a class of generalised Petri nets. *IET Control Theory Appl.* **2010**, *4*, 854–864. [CrossRef]
39. Liu, G.Y.; Barkaoui, K. Necessary and Sufficient Liveness Condition of GS3PR Petri Nets. *Int. J. Syst. Sci.* **2015**, *46*, 1147–1160. [CrossRef]
40. Hu, H.S.; Zhou, M.C.; Li, Z.W. Liveness and Ratio-enforcing Supervision of Automated Manufacturing Systems Using Petri Nets. *IEEE Trans. Syst. Man Cybern.—Part A Syst. Hum.* **2012**, *42*, 392–403. [CrossRef]
41. Hu, H.S.; Zhou, M.C.; Li, Z.W. Supervisor Optimization for Deadlock Resolution in Automated Manufacturing Systems with Petri Nets. *IEEE Trans. Autom. Sci. Eng.* **2011**, *8*, 794–804. [CrossRef]
42. Hu, H.S.; Liu, Y. Supervisor Synthesis and Performance Improvement for Automated Manufacturing Systems by Using Petri Nets. *IEEE Trans. Ind. Inform.* **2015**, *11*, 450–458. [CrossRef]
43. Wu, W.H.; Chao, D.Y. Controllability of Weakly Dependent Siphons under Elementary-Siphon Control. *Trans. Inst. Meas. Control* **2016**, *38*, 941–955. [CrossRef]
44. Yan, M.M.; Zhu, R.M.; Li, Z.W.; Wang, A.; Zhou, M.C. A Siphon-based Deadlock Prevention Policy for a Class of Petri Nets— S^3PMR . In Proceedings of the 17th World Congress of the International Federation of Automatic Control (IFAC), Seoul, Republic of Korea, 6–11 July 2008; Volume 6, pp. 3352–3357. Available online: <http://toc.proceedings.com/04672webtoc.pdf> (accessed on 13 February 2023). [CrossRef]
45. Yan, M.M.; Li, Z.W.; Wei, N.; Zhao, M. A Deadlock Prevention Policy for a Class of Petri Nets S^3PMR . *J. Inf. Sci. Eng.* **2009**, *25*, 167–183.
46. Ma, Z.Y.; Li, Z.; Giua, A. Design of Optimal Petri Net Controllers for Disjunctive Generalized Mutual Exclusion Constraints. *IEEE Trans. Autom. Control* **2015**, *60*, 1774–1785. [CrossRef]
47. Ma, Z.Y.; Li, Z.; Giua, A. A Constraint Transformation Technique for Petri Nets with Certain Uncontrollable Structures. In Proceedings of the 12th IFAC/IEEE Workshop on Discrete Event Systems, Cachan, France, 14–16 May 2014; pp. 66–72. [CrossRef]
48. Ma, Z.Y.; Li, Z.; Giua, A. Petri Net Controllers for Disjunctive Generalized Mutual Exclusion Constraints. In Proceedings of the 2013 IEEE 18th Conference on Emerging Technologies and Factory Automation (ETFA), Cagliari, Italy, 10–13 September 2013; pp. 1–8. [CrossRef]
49. Ma, Z.Y.; Li, Z.; Giua, A. Petri net Controllers for Generalized Mutual Exclusion Constraints with Floor Operators. *Automatica* **2016**, *74*, 238–246. [CrossRef]
50. Liu, G.Y.; Barkaoui, K.; Zhou, M.C. On Intrinsically Live Structure of a Class of Generalized Petri Nets Modeling FMS. *IFAC Proc. Vol.* **2012**, *45*, 187–192. [CrossRef]

51. Li, Z.W.; Zhou, M.C. Elementary Siphons of Petri Nets for Efficient Deadlock Control. In *Deadlock Resolution in Computer-Integrated Systems*; Zhou, M.C., Fanti, M.P., Eds.; CRC Press: Boca Raton, FL, USA; London, UK; New York, NY, USA, 2005; pp. 309–348; ISBN: 9781482276534. [[CrossRef](#)]
52. Li, Z.W.; Wei, N. Deadlock Control of flexible Manufacturing Systems via Invariant-controlled Elementary Siphons of Petri Nets. *Int. J. Adv. Manuf. Technol.* **2007**, *33*, 24–35. [[CrossRef](#)]
53. Li, Z.W.; Wu, N.Q.; Zhou, M.C. Deadlock Control of Automated Manufacturing Systems Based on Petri Nets—A Literature Review. *IEEE Trans. Syst. Man Cybern.—Part C* **2012**, *42*, 437–462. [[CrossRef](#)]
54. Liu, G. J.; Jiang, C.J. Incidence Matrix Based Methods for Computing Repetitive Vectors and Siphons of Petri Net. *J. Inf. Sci. Eng.* **2009**, *25*, 121–136. [[CrossRef](#)]
55. Shi, W.; He, Z.; Gu, C.; Run, N.; Ma, Z. Performance Optimization for a Class of Petri Nets. *Sensors* **2023**, *23*, 1447. [[CrossRef](#)]
56. You, D.; Wang, S.; Zhou, M. Supervisory Control of Petri Nets in the Presence of Replacement Attacks. *IEEE Trans. Autom. Sci. Eng.* **2021**, *67*, 1466–1473. [[CrossRef](#)]
57. Seatzu, C. Modeling, Analysis, and Control of Automated Manufacturing Systems Using Petri Nets. In Proceedings of the 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 27–30. [[CrossRef](#)]
58. Hou, Y.F.; Li, Z.W.; Zhao, M.; Liu, D. Extended Elementary Siphon-based Deadlock Prevention Policy for a Class of Generalised Petri Nets. *Int. J. Comput. Integr. Manuf.* **2014**, *27*, 85–102. [[CrossRef](#)]
59. Miyagi, P.E.; Riascos, L.A.M. Modeling and Analysis of Fault-tolerant Systems for Machining Operations Based on Petri Nets. *Control Eng. Pract.* **2006**, *14*, 397–408. [[CrossRef](#)]
60. Stan, A.C.; Oprea, M. Petri Nets Based Coordination Mechanism for Cooperative Multi-Robot System. *J. Electr. Eng. Electron. Control. Comput. Sci.—JEECCS* **2020**, *6*, 7–14. [[CrossRef](#)]
61. Lopez-Grao, J.P.; Colom, J.M. Resource Allocation Systems: Some Complexity Results on the S⁴PR Class. In *Formal Techniques for Networked and Distributed Systems—FORTE*; Lecture Notes on Computer Science; Najm, E., Pradat-Peyre, J.F., Donzeau-Gouge, V.V., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4229, pp. 323–338; ISBN 978-3-540-46219-4. https://doi.org/10.1007/11888116_24.
62. Liu, G.J. Complexity of the Deadlock Problem for Petri Nets Modeling Resource Allocation Systems. *Inf. Sci.* **2016**, *363*, 190–197. [[CrossRef](#)]
63. Dong, Y.Y.; Chen, Y.F.; Li, S.Y.; El-Meligy, M.A.; Sharaf, M. An Efficient Deadlock Recovery Policy for Flexible Manufacturing Systems Modeled with Petri Nets. *IEEE Access* **2019**, *7*, 11785–11795. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.