Information Sciences 363 (2016) 198–220

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A survey of siphons in Petri nets

GaiYun Liu[a,b,*], Kamel Barkaoui[b]

[a] School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, PR China
[b] Cedric Lab and Computer Science Department, Conservatoire National des Arts et Métiers, Paris 75141, France

ARTICLE INFO

Article history:
Received 31 December 2014
Revised 28 June 2015
Accepted 21 August 2015
Available online 29 August 2015

Keywords:
Concurrent system
Petri net
Deadlock
Siphon
Supervisor

ABSTRACT

Petri nets have gained increasing usage and acceptance as a basic model of asynchronous concurrent systems since 1962. As a class of structural objects of Petri nets, siphons play a critical role in the analysis and control of systems modeled with Petri nets. This paper surveys the state-of-the-art siphon theory of Petri nets including basic concepts, computation of siphons, controllability conditions, and deadlock control policies based on siphons. Some open problems on siphons are discussed, such as the maximally permissive supervisor design problems based on siphons and the application of siphons to robust supervisory control. This survey is expected to serve as a reference source for the growing number of Petri net researchers and practitioners.

1. Introduction

In 1962, Petri invented a net-theoretic approach to model and analyze communication systems in his thesis [146]. This model was based on the concepts of asynchronous and concurrent operation by the parts of a system and the realization that relationships between the parts could be represented by a net. A large amount of research has been done on both the nature [153] and the application of Petri nets, and their application seems to be expanding [43,51,147]. Petri nets have been proven to be very useful in the modeling, analysis [202,203], simulation, and control of concurrent systems [34,35, 66–81,92,94,98,108,111,113,123,124,133,137,150,151,165,183,184,189,196–198,205–213].

The concurrent flow of multiple jobs in a resource allocation system (RAS), which all compete for a finite set of resources, can lead to a deadlock. A deadlock occurs when a set of jobs are in a "circular wait" state [122], where each job in the set is waiting for a resource being held by another job in the set while occupying a resource that is, in turn, needed by one of the other jobs [3]. The notion of partial or total deadlock is frequent and validation before implementation is preferable to reduce the risks.

Due to the easy and concise description of the concurrent execution of processes and the resource sharing by Petri nets, many methods to verify deadlock-freeness and to synthesize controllers using structural theory or reachability graph analysis have been proposed over the past two decades. Traditionally, a deadlock control policy can be evaluated by three performance criteria: structural complexity, behavioral permissiveness, and computational complexity [25–27]. A maximally permissive supervisor has great potential to lead to high utilization of system resources. A supervisor with a simple structure can decrease the hardware and software costs. A policy with low computational complexity means that it can be applied to real-world large-sized systems. Many researchers have developed deadlock control algorithms that can obtain liveness-enforcing supervisors with maximal permissibility, a simple supervisory structure, and low computational complexity. Generally, deadlock control policies based on the state space analysis can approach the maximal permissive behavior, but suffer from the state explosion problem [28–31,140,141].

* Corresponding author. At: School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, PR China. Tel.: +8613572429091.
E-mail addresses: lg_2005@163.com, gaiyunliu@gmail.com (G.Y. Liu), kamel.barkaoui@cnam.fr (K. Barkaoui).

http://dx.doi.org/10.1016/j.ins.2015.08.037
0020-0255/

On the contrary deadlock control policies based on the structural analysis of Petri nets avoid in general the state explosion problem successfully [6,7,11]. Resource circuits and siphons are two typical structural objects which have been widely studied in deadlock control area. This paper will focus on siphons and their applications to the development of deadlock control policies.

The idea behind siphon was first introduced by Commoner and Holt in 1971 for a restricted class of Petri nets called Free-Choice Petri nets [37]. Then Hack in his thesis showed important results for the class of Free-Choice Petri nets and solves the deadlocks (i.e., siphons) and unpredictability problem for a restricted class of systems called Production Schemata [56]. Most of the studies on Petri nets around this time were in the form of theses, dissertations, and reports that are neither readily available nor in wide circulation. From the internet we can find the first paper which gave the name "siphon" to this special structural objects in 1975 in [14]. A nonempty subset of places S in a net N is called a siphon (also known as a deadlock or co-trap) if ${}^{\bullet}S \subseteq S^{\bullet}$, i.e., every transition having an output place in S has an input place in S. We use a siphon instead of a deadlock since the latter may lead to terminological misunderstandings and it is used for a circular waiting condition or behavior in computer science [38,139].

To acquaint the reader with Petri nets, in Section 2, we first present their formal definition and the related concepts, including structural and behavioral properties. The concepts of elementary and dependent siphons are given in this section. A number of important subclasses of Petri nets and their relationships are discussed. In Section 3, different methods of siphon computation are recalled. Section 4 presents controllability conditions of siphons. Section 5 reviews deadlock control policies based on siphons, including the enumeration of strict minimal siphons, mathematical programming techniques, elementary siphons, deadlock control combined with reachability graph analysis and that based on siphons in Gadara nets, and robust deadlock control policies. Their advantages and limitations are discussed. Section 6 shows some open problems. Section 7 concludes this survey.

Because of the breadth of the applications and depth of the investigations into Petri nets, we touch on the results available in the literature. We refer the interested reader to the original works cited in the references for the proofs and details of the related research.

2. Preliminary

2.1. Basics of Petri nets

A generalized Petri net [109,139] is a four-tuple $N = (P, T, F, W)$ where P and T are finite and non-empty. P is the set of places and T is the set of transitions with $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation of the net, represented by arcs with arrows from places to transitions or from transitions to places. $W : (P \times T) \cup (T \times P) \to \mathbb{N}$ is a mapping that assigns a weight to an arc: $W(f) > 0$ if $f \in F$ and $W(f) = 0$ otherwise, where $\mathbb{N} = \{0, 1, 2, \ldots\}$. If $\forall f \in F$, $W(f) = 1$, then $N = (P, T, F, W)$ is called an ordinary net and denoted as $N = (P, T, F)$. A Petri net $N = (P, T, F, W)$ is pure (self-loop free) if $\forall x, y \in P \cup T$, $W(x, y) > 0$ implies $W(y, x) = 0$. A pure Petri net $N = (P, T, F, W)$ can be represented by its incidence matrix $[N]$, where $[N]$ is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$.

A marking M of a Petri net N assigns to each place a nonnegative integer. To facilitate linear algebraic analysis, a marking M is usually treated as a $|P|$-vector. $M(p)$ denotes the number of tokens in place p. For economy of space, $\Sigma_{p \in P} M(p)p$ is used to denote vector M. Place p is marked at M if $M(p) > 0$. Given a subset $S \subseteq P$, the sum of tokens in all the places in S is denoted by $M(S)$ with $M(S) = \sum_{p \in S} M(p)$. S is marked (unmarked) at M if $M(S) > 0$ ($M(S) = 0$). (N, M_0) is called a net system and M_0 is called an initial marking of N.

A transition t is enabled (disabled) at M if $\forall p \in {}^{\bullet}t$ ($\exists p \in {}^{\bullet}t$), arc (p, t) is enabled (disabled). This fact can be denoted by $M[t\rangle$. Firing an enabled transition t reaches a new marking M' such that for each place p, $M'(p) = M(p) - W(p, t) + W(t, p)$, denoted by $M[t\rangle M'$. Marking M' is said reachable from M if there exists a finite sequence of transitions $\sigma = t_0 t_1 \ldots t_n$ and markings M_1, $M_2, \ldots,$ and M_n such that $M[t_0\rangle M_1[t_1\rangle \ldots M_n[t_n\rangle M'$ holds. Specially, when σ is an empty sequence, we have $M[\sigma\rangle M$. The set of markings reachable from M_0 in N is called the reachability set of the marked Petri net (N, M_0) and denoted as $R(N, M_0)$. Marking $M \in R(N, M_0)$ is legal if $M_0 \in R(N, M)$.

Given a marked Petri net (N, M_0), a transition $t \in T$ is live at M_0 if $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M)$, $M'[t\rangle$ holds. A transition $t \in T$ is said to be dead at marking $M \in R(N, M_0)$, if $\nexists M' \in R(N, M)$, $M'[t\rangle$. (N, M_0) is live at M_0 if $\forall t \in T$, t is live at M_0. Otherwise, (N, M_0) is non-live. (N, M_0) is deadlocked at M if $\nexists t \in T$, $M[t\rangle$, where $M \in R(N, M_0)$ and M is called a dead marking. (N, M_0) is deadlock-free if $\forall M \in R(N, M_0)$, $\exists t \in T$, $M[t\rangle$.

A marked net is bounded if $\exists k \in \mathbb{N}$, $\forall M \in R(N, M_0)$, $\forall p \in P$, $M(p) \leq k$. A net N is structurally bounded if it is bounded for any initial marking.

A P(T)-vector is a column vector $I(J) : P(T) \to \mathbb{Z}$ indexed by P(T), where $\mathbb{Z}$ is the set of integers. P-vector I is called a P-invariant (place invariant [193]) if $I \neq \mathbf{0}$ and $I^T[N] = \mathbf{0}^T$. Here $\mathbf{0}$ is a zero vector. A P-vector I is denoted by $\Sigma_{p \in P} I(p)p$ for economy of space. For example, $I = (1, 0, 0, 0, 2)^T$ is written as $I = p_1 + 2p_5$. P-invariant I is a P-semiflow if every element of I is non-negative. $||I|| = \{p \in P | I(p) \neq 0\}$ is called the support of I. $||I||^+ = \{p \in P | I(p) > 0\}$ denotes the positive support of P-invariant I and $||I||^- = \{p \in P | I(p) < 0\}$ denotes the negative one. I is called a minimal P-invariant if $||I||$ is not a proper superset of the support of any other and the greatest common divisor of its elements is one. If I is a P-invariant of (N, M_0), then $\forall M \in R(N, M_0)$, $I^T M = I^T M_0$.

Let $x \in P \cup T$ be a node of net N. ${}^{\bullet}x = \{y \in P \cup T \mid (y, x) \in F\}$ is called the preset of x, while $x^{\bullet} = \{y \in P \cup T \mid (x, y) \in F\}$ is called its postset. This notation can be extended to a set of nodes as follows: given $S \subseteq P \cup T$, ${}^{\bullet}S = \cup_{x \in S} {}^{\bullet}x$ and $S^{\bullet} = \cup_{x \in S} x^{\bullet}$. Let S be a non-empty subset of places. $S \subseteq P$ is a siphon (trap) if ${}^{\bullet}S \subseteq S^{\bullet}$ ($S^{\bullet} \subseteq {}^{\bullet}S$) holds.

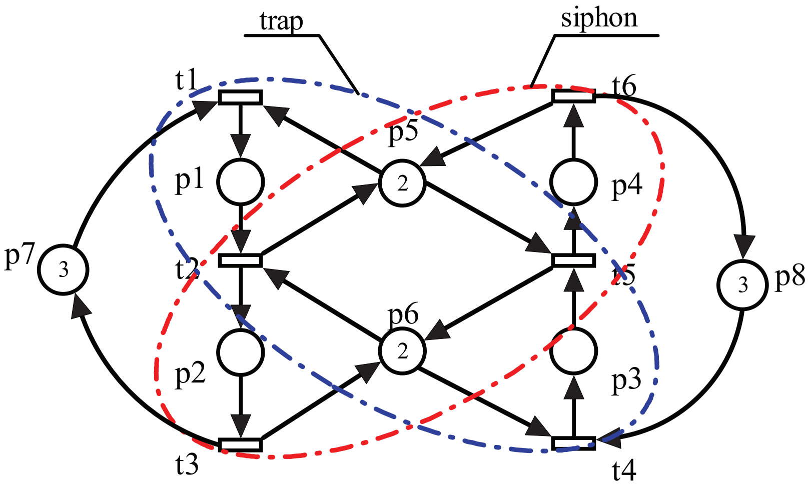

Fig. 1. A marked Petri net (N, M_0).

A siphon that does not contain the support of any P-semiflow is called a strict siphon. A strict siphon is called a strict minimal siphon (SMS) if there is no siphon contained in it as a proper subset. Take Fig. 1 as an example. $S_1 = \{p_2, p_4, p_5, p_6\}$ is an SMS and $S_2 = \{p_1, p_3, p_5, p_6\}$ is a trap.

Let $S \subseteq P$ be a subset of places of Petri net $N = (P, T, F, W)$. P-vector λ_S is called the characteristic P-vector of S if $\forall p \in S$, $\lambda_S(p)$=1; otherwise $\lambda_S(p) = 0$. $\eta_S = [N]^T \lambda_S$ is called the characteristic T-vector of S, where $[N]^T$ is the transpose of incidence matrix $[N]$.

The physical implication of the T-vector of a subset of places is clear: $\eta_S(t) > 0$ means that $\eta_S(t)$ tokens are removed to S after t fires; $\eta_S(t) = 0$ indicates that the number of tokens in S does not change after t fires; and $\eta_S(t) < 0$ implies that $|\eta_S(t)|$ tokens are removed from S after t fires.

Take the Petri net shown in Fig. 1 as an example. Let $S_1 = \{p_2, p_4, p_5, p_6\}$ be a subset of places of the Petri net. We have $\lambda_{S_1} = p_2 + p_4 + p_5 + p_6$, i.e., the space-saving version of vector $(0\ 1\ 0\ 1\ 1\ 1\ 0\ 0)^T$. It is easy to verify that $\eta_{S_1} = -t_1 + t_2 - t_4 + t_5$. Firing t_1 and t_4 removes a token from S_1, respectively, while firing t_2 and t_5 adds a token to S_1, respectively.

Property 1 ([109,153]). *Let S_1 and S_2 are two siphons (traps). Then, $S_1 \cup S_2$ is a siphon (trap).*

Corollary 1 ([109,153]). *If I is a P-semiflow, then $||I||$ is both a siphon and trap.*

Note that the converse of Corollary 1 is not true since a P-invariant depends on not only the topological structure of a net but also the weights attached to the arcs. However, a siphon or trap depends on the topological structure only.

If a siphon contains the support of a P-semiflow and the support is initially marked, then it can never be emptied. In addition, traps and siphons have the following marking invariant laws.

Property 2 ([109,153]). *Let $M \in R(N, M_0)$ be a marking of net (N, M_0) and S a trap. If $M(S) > 0$, then $\forall M' \in R(N, M)$, $M'(S) > 0$.*

This property implies that once a trap is marked under a marking, it is always marked at the subsequent markings that are reachable from the current one.

Property 3 ([109,153]). *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 indicates that once a siphon loses all its tokens, it remains to be unmarked at any subsequent markings that are reachable from the current marking. An empty siphon S causes that no transition in $S^\bullet$ is enabled. Due to the definition of siphons, all transitions connected to S can never be enabled once it is emptied. The transitions are therefore dead, leading to the fact that the net containing these transitions is not live. As a result, deadlock-freedom and liveness of a Petri net are closely related to its siphons, which is shown by the following sections.

2.2. Elementary and dependent siphons

Definition 1 ([95]). Let $N = (P, T, F, W)$ be a net with $|P| = m$, $|T| = n$ and $\Pi = \{S_1, S_2, \ldots, S_k\}$ be a set of siphons of N ($m, n, k \in \mathbb{N}^+$). Let $\lambda_{S_i}(\eta_{S_i})$ be the characteristic $P(T)$-vector of siphon S_i, $i \in \mathbb{N}_k$. $[\lambda]_{k\times m} = [\lambda_{S_1}|\lambda_{S_2}|\cdots|\lambda_{S_k}]^T$ and $[\eta]_{k\times n} = [\lambda]_{k\times m} \times [N]_{m\times n} = [\eta_{S_1}|\eta_{S_2}|\cdots|\eta_{S_k}]^T$ are called the characteristic P- and T-vector matrices of the siphons in N, respectively.

Definition 2 ([96]). Let $\eta_{S_\alpha}, \eta_{S_\beta}, \ldots,$ and η_{S_γ} ($\{\alpha, \beta, \ldots, \gamma\} \subseteq \mathbb{N}_k$) be a linearly independent maximal set of matrix $[\eta]$. Then $\Pi_E = \{S_\alpha, S_\beta, \ldots, S_\gamma\}$ is called a set of elementary siphons in N.

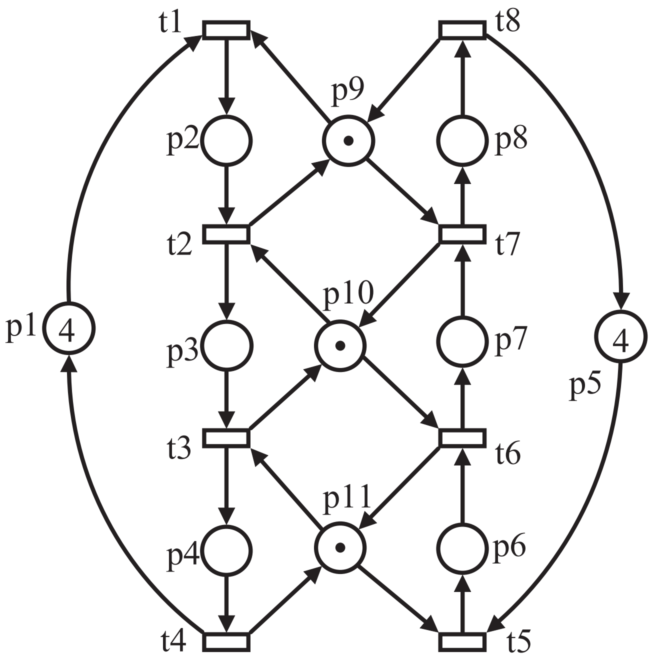

Fig. 2. Example of elementary and dependent siphons in a Petri net.

Definition 3 ([96]). $S \notin \Pi_E$ is called a strongly dependent siphon if $\eta_S = \sum_{S_i \in \Pi_E} a_i \eta_{S_i}$, where $a_i \geq 0$.

Definition 4 ([96]). $S \notin \Pi_E$ is called a weakly dependent siphon if $\exists A$, $B \subset \Pi_E$, such that $A \neq \emptyset$, $B \neq \emptyset$, $A \cap B = \emptyset$, and $\eta_S = \sum_{S_i \in A} a_i \eta_{S_i} - \sum_{S_i \in B} a_i \eta_{S_i}$, where $a_i > 0$.

Let $\Gamma^+(S) = \sum_{S_i \in A} a_i \eta_{S_i}$ and $\Gamma^-(S) = \sum_{S_i \in B} a_i \eta_{S_i}$ for a weakly dependent siphon S. We have $\eta_S = \Gamma^+(S) - \Gamma^-(S)$. If S is strongly dependent, we define $\Gamma^-(S) = 0$.

Take the net in Fig. 2 as an example. There are three SMS: $S_1 = \{p_3, p_8, p_9, p_{10}\}$, $S_2 = \{p_4, p_7, p_{10}, p_{11}\}$, and $S_3 = \{p_4, p_8, p_9, p_{10}, p_{11}\}$. $\Pi_E = \{S_1, S_2\}$ is a set of elementary siphons in the net and S_3 is a corresponding strongly dependent siphon, thanks to the truth of $\eta_{S_3} = \eta_{S_1} + \eta_{S_2}$.

Lemma 1 ([109]). *The number of elements in any set of elementary siphons in net N equals the rank of* $[\eta]$.

Let Π_E denote a set of the elementary siphons in a Petri net. Since the rank of $[\eta]$ is at most the smaller of $|P|$ and $|T|$, Lemma 1 leads to the following important conclusion.

Theorem 1 ([109]). $|\Pi_E| \leq min\{|P|, |T|\}$.

This result indicates that the number of elementary siphons in a Petri net is bounded by the smaller of place and transition counts.

Let S be a (strongly or weakly) dependent siphon. In sequel, if η_S can be linearly represented by elementary siphons' characteristic T-vectors η_{S_1}, η_{S_2}, ..., and η_{S_n} with non-zero coefficients, we say that $S_1, S_2, \ldots$, and S_n are the elementary siphons of S. Let Π be the set of siphons in which we are interested given a net, and Π_D be the set of dependent ones within the scope of Π. Obviously, we have $\Pi = \Pi_E \cup \Pi_D$.

2.3. Subclasses of Petri nets

Table 1 lists a variety of Petri net subclasses. Note that in Table 1, by their definitions, S^{4}R, S^{4}PR, and S^{3}PGR^2 are equivalent, which are named by different researchers. Similarly, ES^{3}PR (defined in [82]) and S^{3}PMR [195] are equivalent, and also WS^{3}PR and GS^{3}PR are equivalent.

A seminal work on deadlock prevention is made in [47], where a paramount class of Petri nets, S^{3}PR, is introduced. A processing stage in a process in an S^{3}PR can use at most a single resource unit at a given state. A subclass of S^{3}PR, called Linear S^{3}PR (LS^{3}PR), is presented in [48], which features some useful properties. The mentioned restriction over resources usage is eliminated by the (more general) S^{4}R class [125]. This allows processes to simultaneously reserve several resources belonging to distinct types.

S^{3}PR and S^{4}R are typical classes of ordinary and generalized Petri nets, respectively. Both S^{3}PR and S^{4}R are composed of a set of state machines and a set of resource places. In S^{3}PR, only one shared resource is allowed to be used at each stage

Table 1
Subclasses of Petri nets in the literature.

Subclass	Full Name	Reference	Type
S^3PR	Systems of Simple Sequential Processes with Resources	[47]	ordinary
LS^3PR	Linear System of Simple Sequential Processes with Resources	[48]	ordinary
ES^3PR	Extended S^3PR	[157]	generalized
ES^3PR	Extended S^3PR	[82]	ordinary
S^3PMR	System of Simple Sequential Processes with Multiple Resources	[84]	ordinary
WS^3PSR	Weighted System of Simple Sequential Processes with Several Resources	[156]	generalized
WS^3PR	Weighted System of Simple Sequential Processes with Resources	[201]	generalized
GS^3PR	Generalized Systems of Simple Sequential Processes with Resources	[130]	generalized
β-Net		[128]	ordinary
S^4PR		[158]	generalized
S^4PR	Simple S^3PR	[17]	ordinary
S^3PGR^2	System of Simple Sequential Processes with General Resource Requirements	[145]	generalized
S^4R	System of Sequential Systems with Shared Resources	[125]	generalized
S^5PR		[135]	generalized
S^*PR		[49]	generalized
WFR-Net	Workflow Nets with Shared Resources	[10]	generalized
G-tasks		[9]	generalized
G-systems		[9]	generalized
PPN	Production Petri Nets	[3]	ordinary
PNR	Process Nets with Resources	[88]	ordinary
RCN-Merged Nets	Resource Control Nets-Merged Nets	[87]	ordinary
ERCN-Merged nets	Extended Resource Control Nets-Merged Nets	[190]	ordinary
ERCN$*$-Merged nets	Well-behaved Extended Resource Control Nets-Merged Nets	[89]	ordinary
AMG	Augmented Marked Graphs	[36]	ordinary
SPQR	System of Processes Quarrelling over Resources	[135]	generalized
PC^2R	Processes Competing for Conservative Resources	[136]	generalized
Gadara Nets		[119]	generalized

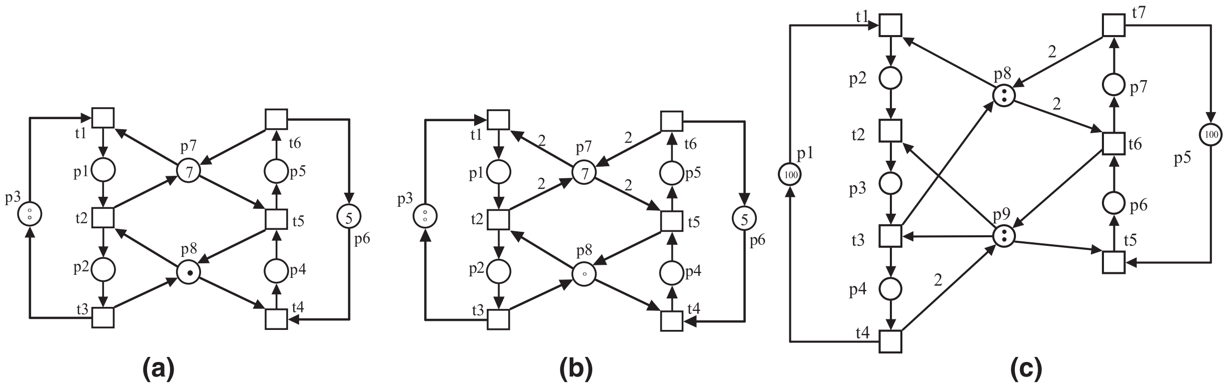

Fig. 3. (a) An S^3PR, (b) a GS^3PR, and (c) an S^4R.

in a job. Compared with the resource occupation in S^3PR, the usage of resources in S^4R is almost arbitrary and requires only conservativeness.

A GS^3PR is a subclass of S^4R and a generalized version of an S^3PR. A GS^3PR is equivalent to a WS^3PR in [201]. Since any Petri net has a weight function, it is sound and rational to rename a *weighted* S^3PR in [201] to be a *generalized* S^3PR. A GS^3PR becomes an S^3PR if the weight of each arc is changed to be one. Fig. 3 depicts intuitive examples of the Petri net subclasses mentioned-above. We can see that in an S^3PR, an activity place (representing a processing stage) of a job needs a single unit of a single resource type. For example, the activity modeled with place p_1 in Fig. 3(a) needs one unit of only one resource type, i.e., p_7. In a GS^3PR, an activity place of a job may need multiple units of a single resource type. For instance, the activity place modeled with p_1 in Fig. 3(b) needs two units of only one resource type, i.e., p_7. While, in an S^4R, an activity place may be supported by any resource requirements, specifically, multiple units of multiple resource types. In Fig. 3(c), the activity place modeled with p_3 needs one unit of resource p_8 and one unit of resource p_9 [130].

A GS^3PR is a subclass of S^4R and a generalized version of an S^3PR. It is easy to understand that the decision conditions for S^4R still hold for GS^3PR [130].

Nowadays, the most general class of the S^nPR family is the S^*PR class [136], in which processes are ordinary state machines with internal cycles. In order to model general systems, a variety of manufacturing-oriented Petri net subclasses are proposed by different researchers [136]. Except Gadara nets, all the aforementioned Petri net classes in Table 1 are frequently presented in the

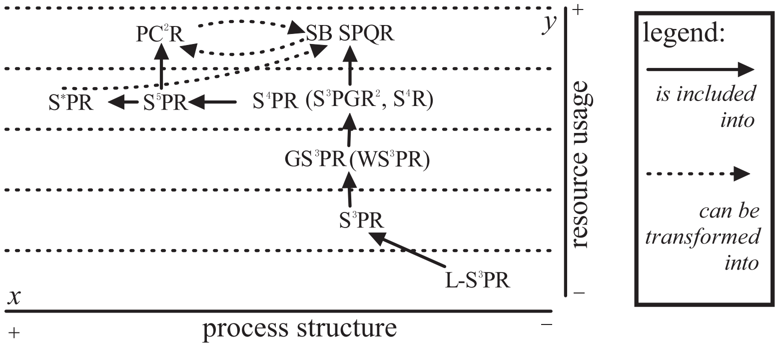

Fig. 4. Inclusion relations among Petri net classes for sequential RAS in [136].

context of manufacturing-oriented modeling, and make sense as artifacts conceived for properly modeling significant physical aspects of this kind of systems. For all of these, a siphon-based liveness characterization is known. Due to its structural nature, it opens a door to an efficient detection and correction of deadlocks, by implementing controllers (usually by the addition of places) that restrain the behavior of the net and avoid the bad markings to be reached.

Gadara nets [119], a special class of Petri nets that arises in the modeling of the execution of multithreaded computer programs for the purpose of deadlock avoidance. In [136], the class of Processes Competing for Conservative Resources (PC^{2}R) is introduced. This class aims to overcome the deficiencies identified in finding models which properly capture the RAS view of multithreaded software systems. Furthermore, it generalizes other subclasses of the S^{n}PR family while respecting the design philosophy on these.

Finally, there exists another class for sequential RAS, called System of Processes Quarrelling over Resources (SPQR), which does not strictly contain or is contained by the PC^{2}R class. Yet, there exist transformation rules to travel between PC^{2}R and structurally bounded SPQR. Note that, by construction, PC^{2}R nets are conservative, and hence structurally bounded, but this is not true for general SPQR [136]. Fig. 4 summarizes the inclusion relations between the reviewed Petri net classes for sequential RAS. The left on the x-axis, the more complex the process structure can be (i.e., linear state machines are on the right while general state machines are on the left). The upper on the y-axis, the higher degree of freedom in the way the processes use the resources (resource lending is on top). The figure also illustrates the fact that every model of the S^{n}PR family can be transformed into a PC^{2}R net [136].

Traditionally, empty or insufficiently marked siphons have been a fruitful structural element for characterizing non-live RAS. The more general the net class, however, the more complex the siphon-based characterization is.

3. Computation of siphons

Siphons are related to liveness properties of Petri net models. However, the computation of these structural components can be very time-consuming or, even, impossible. This section reviews and discusses existing siphon generating algorithms in the literature.

An algorithm with polynomial complexity to decide whether a set of places is a minimal siphon can be found in [4]. From this characterization of minimal siphons, Barkaoui and Minoux proved first that liveness is decidable in polynomial time for large classes of Petri nets [12]. Classic and typical siphon computation methods are presented in [46,90,167], and [194]. A siphon computation method that is claimed to be rather efficient is developed in [41], which can find 2×10^7 siphons within one hour. For a class of Petri nets, a siphon solution approach is given in [1] that is also efficient through experimental studies. A parallel solution to compute siphons is established by Tricas and Ezpeleta [161]. An efficient minimal siphon computation approach by using binary decision diagrams (BDD) is provided in [31]. Resource-transition circuits can be used to characterize deadlocks [191]. The relationship between maximal perfect resource-transition circuits and siphons is explored in [192].

In [15], a sign incidence matrix is introduced which can be used as a new approach to structural analysis of Petri nets. For example, the presented algorithm can find basis siphons (traps). A new concept of place-minimal siphons which play a central role in the algorithm is introduced. The most important feature of the algorithm is that it only generates basis siphons (traps). This is crucial since in general, the number of siphons (traps) grows quickly, easily beyond practical limits. In addition, the algorithm is highly amenable to parallel computations. Finally, an upper limit on the maximum number of basis siphons in a class of Petri nets is postulated based on numerical experiments.

The work in [39] addresses the problem of computing siphons and traps in a standard Petri net. In particular, starting from a clear formulation in terms of predicate logic, it is shown how binary programming techniques can be adopted to formulate and solve the problem of finding minimal and basis siphons. An iterative algorithm is proposed which operates with a general purpose mixed integer programming (MIP) solver. Cordone et al. state and prove some new theoretical results which aim at reducing the original problem to a set of smaller sub-problems in [40]. Accordingly, a conceptual algorithm is proposed which efficiently computes a set of siphons containing the minimal siphons of a net. The algorithm is based on a strategy which divides the original problem into several but smaller sub-problems that are definitely easier to solve. The subproblems, in fact, coincide with the general problem of finding a generic siphon in a given net, with additional constraints including or excluding given places.

Tricas and Ezpeleta show how the special syntactical constraints of S^4PR can help in developing specific implementations to compute siphons in a very efficient way [159,161]. Some empirical experiments and research about the available methods for siphon computation have shown that the method proposed in [15] is adequate for these needs. Taking the advantage of both the structure of S^4PR nets and the way by which the siphons are computed in the selected method, it has been improved by means of a parallel solution. In order to measure the improvements some experiments have been carried out, comparing the time required for the computation of siphons in families of Petri nets. As a result of such experiments they conclude that the proposed parallel algorithm is of great interest since the speedup charts show that the use of a set of processors (even in the case that they are connected using a network of computers instead of a multi-processor) gives very good results. Another conclusion is that, as it is predictable, the Petri net structure is very important when an adequate load balance is needed. However, the authors can provide little insight of how resources should be distributed among processors in order to obtain an adequate load balance.

In [155], the authors propose an algorithm Generating-Disjoint-Minimal-Siphon-Traps (GMST) with complexity $O(|P|^2(|P|+|T|+|E|))$ and another one $GMST_i$ that repeats GMST i times. By incorporating $GMST_i$ into the Fourier–Motzkin method as preprocessing, they propose an algorithm $STFM\_G_i$ for computing minimal-support nonnegative integer invariants.

In [31], using binary decision diagrams (BDD), a symbolic approach to the computation of minimal siphons in Petri nets is presented. The siphons of a Petri net can be found via a set of logic conditions. The logic conditions are symbolically modeled by using Boolean algebras. The operations of Boolean algebras are implemented by BDD that are capable of representing large sets of siphons with small shared data structures. The proposed method first uses BDD to compute all siphons of a Petri net and then a binary relation is designed to extract all minimal siphons. However, the computation for identifying minimal siphons from all siphons requires much more time. It is necessary to speed BDD operations for identifying minimal siphons.

The work in [16] also tackles the siphon computation problem in S^4PR. A new method to compute the minimal siphons based on the use of higher level objects is proposed. The computation of a minimal siphon containing the set of resources D_R requires the union of the minimal siphons corresponding to each resource of D_R. Each of these minimal siphons, used as operands, must be diminished by a pruning set of process places that can be computed from a pruning relation on the minimal siphons with one resource. This pruning relation is represented by a graph that it is used to compute the minimal siphons by means of maximal strongly connected subgraphs. The use of these high level objects for the computation of the minimal siphons improves remarkably the time and space needed for this purpose, with respect to the previously known algorithms. The algorithm is very economic in memory in all intermediate steps with respect to the classical algorithms. Nevertheless, much more work must be done in order to find an optimal implementation with a good selection of data structures and heuristics to reduce the number of comparisons.

In [169], a more effective approach to compute SMS in S^3PR is reported. First, the concept of loop resource subsets and their characteristic resource subnets are proposed. Next, sufficient and necessary conditions for loop resource subsets to generate SMS are established. Finally, an algorithm is developed to find all the SMS based on loop resource subsets. Since the number of loop resource subsets is much less than that of resource circuits, the computational efficiency of the SMS enumeration task is significantly improved by the proposed method.

In [134], the authors point out that the method in [169] needs to generate all the characteristic resource subnets and then to decide the strong connectivity of each characteristic resource subnet, which is a tedious process. In [134], the concept of a critical resource place is proposed, which is important in deciding whether a simple loop resource subset can derive an SMS. Then, sufficient conditions for the simple loop resource subsets to derive SMS are established, which facilitates deciding whether a composed loop resource subset can generate an SMS. Finally, based on the proposed method, all SMS can be obtained in an S^3PR net.

Recently, a new algorithm based on depth-first search of a problem decomposition process is proposed to compute all minimal siphons in an ordinary Petri net [142]. The algorithm can reduce the number of problems in the problem list. Compared with Cordone's algorithm, the proposed algorithm can solve the problem of high requirement for computer memory in computing all minimal siphons and decrease the memory consumption since the computer memory size is closely related to the number of problems in the problem list.

When designing liveness-enforcing Petri net supervisors, unlike other techniques, Li and Zhou add control places and arcs to a plant net model for its elementary siphons only [95], greatly reducing the structural complexity of the controlled system. It is shown that elementary siphons can be used to derive structurally simple liveness-enforcing supervisors and thus their computation is important.

By using the concept of handles and bridges [45], much work is done by Chao on the computation of minimal and elementary siphons in an RAS [19], [20]. In [17], Chao proposes a new T-characteristic vector ζ to compute SMS for S^3PR in an algebraic fashion. Chao constructs an SMS based on the concept of handles. Roughly speaking, a 'handle' is an alternate disjoint path between two nodes. For S^4PR (simple S^3PR), the work in [17] discovers that elementary siphons can be constructed from elementary circuits where all places are resources. Thus, the set of elementary siphons can be computed without the knowledge of all SMS. Since only one resource is used in each processing stage in a job and the processes are modeled using state machines in an S^3PR, its modeling power is limited. It cannot model iteration statements (loop) in each sequential process and the relationships between synchronization and communication among sequential processes. At any operation place of a process, it cannot use multisets of resources.

Li et al. develop a novel methodology to find interesting siphons for deadlock control purposes in S^{3}PR [107]. Resource circuits in an S^{3}PR are first detected, from which, in general, a small portion of emptiable minimal siphons can be derived. The remaining emptiable ones can be found by their composition. The significance of this work is the development of a polynomial-time algorithm to find a set of elementary siphons without the complete siphon enumeration whose computation is NP-hard.

Computation of elementary siphons proposed by Li and Zhou in [107] is essential for deadlock control. They assumed that the siphon constructed from each resource circuit is an elementary one and proposed a polynomial algorithm to compute elementary siphons. However, the study in [21] demonstrates a counter example where there may exist an exponential number of resource circuits. Hence, constructing elementary siphons from resource circuits will result in an exponential number of elementary siphons, which contradicts the known results. Chao then develops a polynomial algorithm to find elementary siphons, which also constructs all SMS on the way. This is because, in the method proposed by Li and Zhou, a linear algebraic expression must be established for each dependent siphon, which implies that, all SMS must be located.

Based on graph theory, the work in [166] proposes an effective algorithm with polynomial complexity to find a set of elementary siphons for an LS^{3}PR. The algorithm is established through the use of a resource directed graph and complementary sets of SMS of the net. The upper bound of the number of SMS in such a net is figured out. Another work on computation of elementary siphons based on graph theory can be found in [61].

4. Controllability conditions of siphons

4.1. Controllability condition of siphons in an ordinary Petri net

Theorem 2 ([8,44,109])**.** *Let (N, M_0) be an ordinary net and Π the set of its siphons. The net is deadlock-free if $\forall S \in \Pi$, $\forall M \in R(N, M_0)$, $M(S) > 0$.*

This theorem states that an ordinary Petri net is deadlock-free if no (minimal) siphon eventually becomes empty.

Theorem 3 ([8,44,109])**.** *Let (N, M) be an ordinary net that is in a deadlock state. Then, $\{p \in P | M(p) = 0\}$ is a siphon.*

This result means that if an ordinary net is dead at some reachable marking, i.e., no transition is enabled, then the unmarked places form a siphon.

Corollary 2 ([8,109])**.** *A deadlocked ordinary Petri net contains at least one empty siphon.*

Corollary 3 ([8,109])**.** *Let $N = (P, T, F, W)$ be a deadlocked net at marking M. Then, it has at least one siphon S such that $\forall p \in S$, $\exists t \in p^\bullet$ such that $W(p, t) > M(p)$.*

Definition 5 ([8,109])**.** A siphon S is said to be controlled in an ordinary net system (N, M_0) if $\forall M \in R(N, M_0)$, $M(S) > 0$.

Clearly, any siphon that contains a marked trap is controlled since it can never be emptied. In an ordinary Petri net, a siphon that is controlled does not imply a deadlock. However, this is not the case in a generalized Petri net. For a siphon that can be emptied in a net, some external control mechanism can be exerted on the net such that it becomes controlled. A siphon can be controlled by adding a monitor. When we talk about siphon control, we are usually concerned with minimal siphons since the controllability of a minimal siphon implies that of those containing it. A natural problem is to decide whether a set of places S in a Petri net is a minimal siphon. It is shown in [4] that the decision can be done in polynomial time with complexity $O(m^2 + mn^2)$, where $m = |S^\bullet|$ and $n = |S|$.

Definition 6 ([8,36,109])**.** Siphon S in an ordinary net system (N, M_0) is invariant-controlled by P-invariant I at 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 cannot be emptied, i.e., $\forall M \in R(N, M_0)$, S is marked under M.

In essence, the controllability of siphon S by adding a monitor is ensured by the fact that the number of tokens leaving S is limited by a marking invariant law imposed on the Petri net, which is implemented by a P-invariant whose support contains the monitor.

In order to test whether a siphon S is controlled by a P-invariant I, it is sufficient to solve the following system of linear homogeneous inequalities and equations:

$$\begin{aligned} &I^T[N] = \mathbf{0}^T \\ &I^T M_0 > 0 \\ &I(p) \le 0, \quad \forall p \in P \backslash S \end{aligned}$$

For the above system, the existence of a solution can be proved through *Phase I* of the simplex algorithm applied to the following linear programming problem (LPP):

$$\begin{aligned} &\text{maximize } \mathbf{0}^T I \\ &\text{subject to} \\ &I^T[N] = \mathbf{0}^T \end{aligned}$$

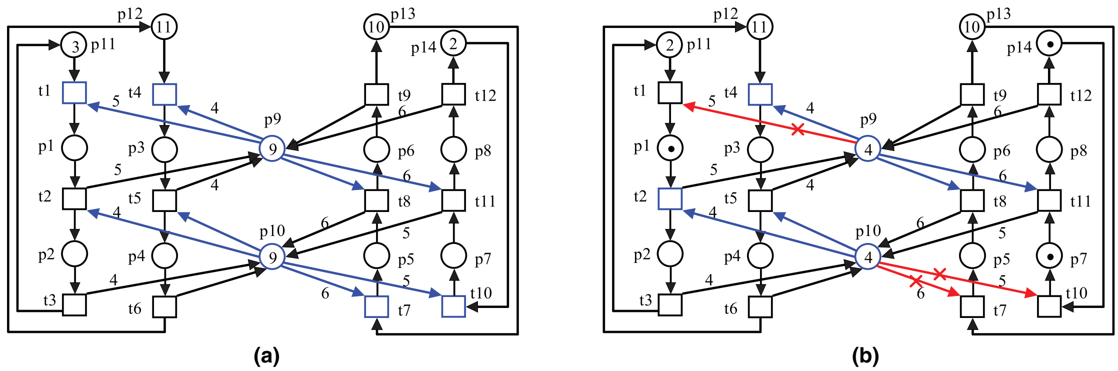

Fig. 5. (a) A GS^{3}PR net (N, M_0), and (b) a live GS^{3}PR with a non-max-controlled siphon.

$$I^T M_0 > 0$$
$$I(p) \le 0, \quad \forall p \in P \backslash S$$

Phase I of the simplex algorithm computes a *basic feasible solution* of the set of constraints of the LPP if it exists.

An empty or insufficiently marked siphon in a Petri net can cause some transitions not to be enabled. A siphon in an ordinary Petri net can be made invariant-controlled as defined above. The case in a generalized Petri net is much more complicated and is treated as follows.

4.2. Controllability condition of siphons in a generalized Petri net

In a generalized Petri net, owing to the weighted arcs, the non-emptiness of a siphon is not sufficient for the absence of dead transitions. The existence of an SMS is no longer necessary for the occurrence of deadlocks. As a whole, the controllability concept is concerned with the enabling and firing of transitions.

4.2.1. Max-controlled siphons

The following properties are from Barkaoui and Pradat-Peyre's work [8]. Given a place p, we denote $max_{t\in p^\bullet}\{W(p,t)\}$ by $max_{p^\bullet}$.

Definition 1 ([8])**.** Let S be a siphon of a net system (N, M_0). S is said to be max-marked at $M \in R(N, M_0)$ if $\exists p \in S$ such that $M(p) \ge max_{p^\bullet}$.

Definition 2 ([8])**.** Let S be a siphon in a well-marked net (N, M_0). S is said to be max-controlled if S is max-marked at any reachable marking $M \in R(N, M_0)$.

Definition 3 ([8])**.** (N, M_0) satisfies the max cs-property (controlled siphon-property) if each minimal siphon of N is max-controlled.

Theorem 4 ([8])**.** *A generalized Petri net is deadlock-free if it satisfies the max cs-property.*

As shown in Fig. 5(a), (N, M_0) is a well-marked net. $S = \{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ is its unique SMS, where $S_R = \{p_9, p_{10}\}$ and $S_A = \{p_2, p_4, p_6, p_8\}$. In this net, $M_0(p_9) = 9$ and $max_{p_9^\bullet} = max\{W(p_9, t_1), W(p_9, t_4), W(p_9, t_8), W(p_9, t_{11})\} = max\{5, 4, 1, 6\} = 6$. We have $M_0(p_9) > max_{p_9^\bullet}$. Hence, S is max-marked at M_0. Suppose that both t_1 and t_{10} fire once. Then, the net system reaches marking $M = p_1 + p_7 + 4p_9 + 4p_{10} + 2p_{11} + 11p_{12} + 10p_{13} + p_{14}$ as shown in Fig. 5(b). We have $M(p_9) = 4 < max_{p_9^\bullet} = 6$ and $M(p_{10}) = 4 < max_{p_{10}^\bullet} = 6$. Based on the definition of the max-marked siphons, S is not max-marked at this marking. Therefore, (N, M_0) does not satisfy the max cs-property. However, the net is live by, as can be obtained, reachability graph analysis. The max-controllability condition for siphons is restrictive in the sense of deadlock control. A deep study on relation between liveness, deadlock-freeness and cs-property is given in [11].

4.2.2. Max′-controlled siphons

In [18], Chao first proposes a concept namely max′-controlled siphons to relax the max-controlled condition. Zhong and Li [201] refine the concept and propose the formal definition of max′-controlled siphons.

Definition 4 ([201])**.** Let S be a siphon in a well-marked S^{4}R (N, M_0). S is said to be max′-marked at $M \in R(N, M_0)$ if $\exists p \in S_A$ such that $M(p) \ge 1$ or $\exists p \in S_R$ such that $M(p) \ge max_{t\in p^\bullet \cap [S]^\bullet}\{W(p,t)\}$.

Definition 5 ([201])**.** Let S be a siphon in a well-marked S^{4}R (N, M_0). S is said to be max′-controlled if S is max′-marked at M, $\forall M \in R(N, M_0)$.

Theorem 5 ([201])**.** *Let (N, M_0) be a well-marked S^{4}R. If every siphon in the net is max′-controlled, it is live.*

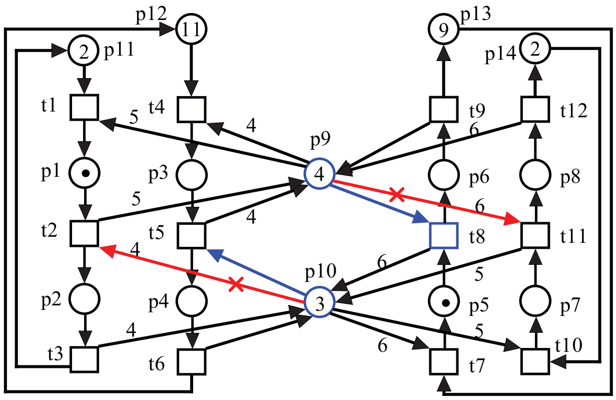

Fig. 6. A live GS^{3}PR with a non-max′-controlled siphon.

The net at $M' = p_1 + p_5 + 4p_9 + 3p_{10} + 2p_{11} + 10p_{12} + 9p_{13} + 2p_{14}$ shown in Fig. 6 is obtained by firing t_1 and t_7 once in Fig. 5(a). We have $M'(p_9) = 4$, $M'(p_{10}) = 3$, $max_{p_9^\bullet \cap [S]^\bullet} = max\{W(p_9, t_8), W(p_9, t_{11})\} = 6$, and $max_{p_{10}^\bullet \cap [S]^\bullet} = max\{W(p_{10}, t_2), W(p_{10}, t_5)\} = 4$. Thus, $M'(p_9) < max_{p_9^\bullet \cap [S]^\bullet}$ and $M'(p_{10}) < max_{p_{10}^\bullet \cap [S]^\bullet}$. Based on the definition of the max′-marked siphons, S is not max′-marked at this marking. However, the net is live. The max′-controllability condition for siphons is still restrictive in the sense of deadlock control.

4.2.3. Max″-controlled siphons

In [125], Liu et al. present max″-controllability condition of siphons to relax the max′-controllability condition.

Definition 6 ([125])**.** Let S be a siphon in a well-marked S^{4}R (N, M_0). S is said to be max″-marked at $M \in R(N, M_0)$ if at least one of the following conditions holds:

(i) M is an initial marking;
(ii) $\exists p \in S_A$ such that $M(p) \geq 1$;
(iii) $\exists r \in S_R$, $min \sum_{t\in T'} \alpha_t \cdot W(t, r) + M(r) \geq max_{t' \in r^\bullet \cap [S]^\bullet}\{W(r, t')\}$, where $T' = \{t | t \in {}^\bullet r \cap [S]^\bullet, \forall r' \in {}^\bullet t \cap P_R, M(r') \geq W(r', t), M(P_A \cap {}^\bullet t) \geq 1\}$, α_t denotes the times that t is fired from marking M, and $min \sum_{t\in T'} \alpha_t \cdot W(t, r)$ can be solved by the following mixed integer program (MIP):

$$
\begin{aligned}
& min \sum_{t\in T'} \alpha_t \cdot W(t, r) \\
p \in^\bullet t \cap P_A, \quad & M(p) \geq 1, t_x \in p^\bullet \cap T' \\
& \sum \alpha_{t_x} \leq M(p) \\
r' \in^\bullet t \cap P_R, \quad & t_y \in r'^\bullet \cap T' \\
& \sum \alpha_{t_y} \cdot W(r', t_y) \leq M(r') \\
t \in^\bullet r \cap [S]^\bullet & \\
& min \left\{ \frac{M(r') - \sum \alpha_{t_y} \cdot W(r', t_y)}{W(r', t)}, M(p) - \sum \alpha_{t_x} \right\} < 1 \\
& \alpha_t \in \mathbb{N}
\end{aligned}
$$

Definition 7 ([125])**.** Let S be a siphon in a well-marked S^{4}R (N, M_0). S is said to be max″-controlled if $\forall M \in R(N, M_0)$, S is max″-marked at M.

Theorem 6 ([125])**.** *Let (N, M_0) be a well-marked S^{4}R. The net is live if all its siphons are max″-controlled.*

The net at marking $M'' = p_1 + p_3 + p_5 + 3p_{10} + 2p_{11} + 10p_{12} + 9p_{13} + 2p_{14}$ shown in Fig. 7 is obtained by firing t_1, t_4, and t_7 once in Fig. 5(a). We have $M''(p_9) = 0$, $M''(p_{10}) = 3$, $M''(p_3) = 1$, and $T' = \{t_5\}$. Thus, $min\alpha_{t_5} \cdot W(t_5, p_9) + M''(p_9) = 4\alpha_{t_5} = 4 < max_{p_9^\bullet \cap [S]^\bullet}$, where α_{t_5} can be obtained by solving the MIP in Definition 6. Based on the definition of the max″-marked siphons, this S is not max″-marked at this marking. However, the net is live. The max″-controllability condition for siphons is still restrictive.

By Definitions 1, 4, and 6, we check siphon $\{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ at markings shown in Figs. 5(a), 5(b), 6, and 7. Test results are shown in Table 2. We can see that the above three controllability conditions of siphons are all sufficient but not necessary for the liveness of GS^{3}PR.

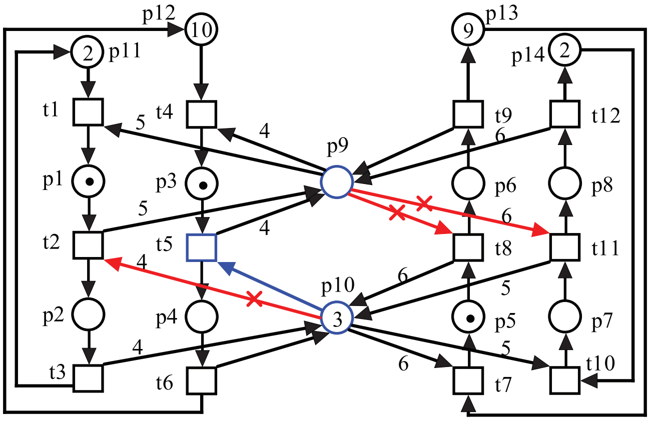

Fig. 7. A live GS^{3}PR with a non-max$''$-controlled siphon.

Table 2
Controllability of siphon $\{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ at different markings.

The marking in	Max-marked	Max$'$-marked	Max$''$-marked
Fig. 5(a)	yes	yes	yes
Fig. 5(b)	no	yes	yes
Fig. 6	no	no	yes
Fig. 7	no	no	no

4.2.4. Max*-controlled siphons

Recently, in [130], Liu and Barkaoui present max*-controllability condition of siphons to relax the existing controllability conditions.

Definition 8 ([130]). Let S be a strict siphon in a well-marked GS^{3}PR net (N, M_0). S is said to be max*-marked (non-max*-marked) at $M \in R(N, M_0)$ if at least one (none) of the following conditions holds:

(i) $\exists p \in S_A, M(p) \geq 1$;
(ii) $\exists r \in S_R, M(r) \geq max_{t \in T_r^c} W(r, t)$;
(iii) $\exists t \in T_S^c$, $e_{pt} = 1$ and $e_{rt} = 1$ (t is enabled at M).

Theorem 7 ([130]). *Let (N, M_0) be a well-marked GS^{3}PR net and $\Pi \neq \emptyset$ be the set of SMS. It is live iff $\forall S \in \Pi$, S is max*-controlled.*

4.2.5. Discussion

From the definition of max-controlled siphons [8], the number of tokens in each place of a siphon is restricted by the maximal weights of its output arcs, i.e., $M(p) \geq max_{p^\bullet}$. The results in [18] points out that this condition is too restrictive and proposes the concept of max$'$-controlled siphons. As for the marking of resource places considered in a siphon, condition $M(p) \geq max_{p^\bullet}$ is relaxed to be $M(p) \geq max_{t \in p^\bullet \cap [S]^\bullet}\{W(p, t)\}$. Note that the max$'$-controlled condition of a siphon is still a sufficient but not necessary.

In [125], the authors show that the max$'$-controllability condition can be further relaxed by a new concept called max$''$-controlled siphons. As shown in Definition 6, at the marking of the considered resource places of a siphon, condition $M(p) \geq max_{t \in p^\bullet \cap [S]^\bullet}\{W(p, t)\}$ is relaxed to be $min \sum_{t \in T'} \alpha_t \cdot W(t, r) + M(r) \geq max_{t' \in r^\bullet \cap [S]^\bullet}\{W(r, t')\}$. This constraint guarantees that each transition in $r^\bullet \cap [S]^\bullet$ is potentially enabled. However, we need to solve an MIP to decide whether a siphon is max$''$-marked at a marking $M \in R(N, M_0)$. What is more, the max$''$-controllability condition of the siphons in S^{4}R is a sufficient condition only.

Recently, Liu and Barkaoui improve the work in [125] by splitting Condition (iii) of Definition 6 into two parts: Conditions (ii) and (iii) of Definition 8. This method avoids solving MIP problems and loosens the constraint $min \sum_{t \in T'} \alpha_t \cdot W(t, r) + M(r) \geq max_{t' \in r^\bullet \cap [S]^\bullet}\{W(r, t')\}$. It is easy to see that the former two conditions of the definition of max*-marked siphons are completely the same with the definition of max$'$-marked siphons. The new definition allows a new condition. Understandably, it is more general than max$'$-marked siphons. Condition (iii) in Definition 8 means that a critical transition t of S is enabled at M. At M, a max*-marked siphon can guarantee that at least one transition in its preset can fire once.

The net shown in Fig. 5(a) is live with 654 legal markings. S is non-max-marked at 8 markings, non-max$'$-marked at 4 markings, and non-max$''$-marked at one marking as shown in Table 3, where the numbers of tokens in p_{11}, p_{12}, p_{13}, and p_{14} are not shown. However, S is max*-marked at all 654 markings. In a word, from a max-controlled siphon to max*-controlled one, constraints for siphon control become more and more weaker.

Based on the max*-controllability condition, we can design supervisors with more permissive behavior for GS^{3}PR nets in theory. Hopefully, this condition can be used in an appropriate policy that leads a supervisor of a Petri net system to be optimal.

Table 3
Controllability conditions of siphon $\{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ in Fig. 5(a).

Markings of $p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}$	Max	Max′	Max″	Max*
1,0,1,0,1,0,0,0,0,3	no	no	no	yes
1,0,0,0,1,0,0,0,4,3	no	no	yes	yes
0,0,2,0,1,0,0,0,1,3	no	no	yes	yes
0,0,1,0,1,0,0,0,5,3	no	no	yes	yes
1,0,1,0,0,0,1,0,0,4	no	yes	yes	yes
0,0,1,0,0,0,1,0,5,4	no	yes	yes	yes
0,0,2,0,0,0,1,0,1,4	no	yes	yes	yes
1,0,0,0,0,0,1,0,4,4	no	yes	yes	yes

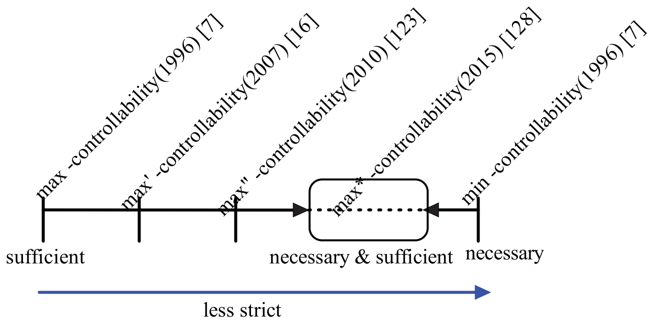

Fig. 8. Controllability conditions of siphons.

Then the corresponding manufacturing system is of more flexibility. In this sense, the max*-controllability condition in [130] can promote the development of optimal or suboptimal deadlock control. In other words, this necessary and sufficient liveness condition for GS^{3}PR will be considered as an important progress in deadlock control of generalized Petri nets. Due to the complex structures of S^{4}R, finding a necessary and sufficient siphon control condition for S^{4}R is a challenging problem and remains open, which, in our own opinion, still needs many efforts.

Here we use Fig. 8 to show the truth that from a max-controlled siphon to max*-controlled one, constraints for siphon control become more and more weaker. The controlled siphon properties max and min-controllability proposed by Barkaoui and Peyre [8] look like bounds. Over the past decades, researchers have been looking for less strict conditions. The necessary and sufficient liveness conditions of Petri nets are here in the window of Fig. 8. Up to now, only max*-controllability proposed in recent work [130] is in this window.

5. Deadlock control policies based on siphons

5.1. Enumeration of SMS

In 1995, seminal work was conducted by Ezpeleta et al. [47] who developed a design method of monitor-based liveness-enforcing Petri net supervisors. It is usually considered to be a classical contribution that utilizes structural-analysis techniques of Petri nets to prevent deadlocks in automated manufacturing systems (AMS). A S^{3}PR is proposed and the relationship between SMS and its liveness is established. It is shown that an S^{3}PR is live iff no siphon can be emptied. For each SMS that can be empty at a reachable marking, a monitor is added such that it is controlled, i.e., cannot be unmarked at any reachable marking. After all siphons are controlled, the resulting net that is called a controlled net system is live. The significance of this approach lies in the fact that it successfully separates a plant net model and its supervisor. The liveness-enforcing supervisor in [47] is a Petri net that consists of the monitors and the transitions of the plant model. Unfortunately, the approach in [47] suffers from the following problems: behavioral permissiveness, computational complexity, and structural complexity. The behavioral permissiveness problem is referred to as the fact that the permissive behavior of a plant net model is overly restricted by the deadlock prevention policy, i.e., the supervisor excludes some safe (admissible) states. This is so since the output arcs of a monitor are led to the source transitions of the net model, which limits the number of workpieces to be released into and processed by the system. A source transition is the output transition of an idle place, which models the entry of raw parts into the system. Computational complexity results from the complete siphon enumeration that is necessary to compute a supervisor [47]. As known, the number of siphons grows fast and in the worst case grows exponentially with respect to the size of a net model. Structural complexity is referred to as the number of monitors in a liveness-enforcing supervisor, which is in theory exponential with respect to the size of a net model since every strict minimal siphon that can be unmarked at a reachable marking needs a monitor to prevent from being emptied. The structural complexity of a supervisor means extra cost in system verification, validation, and implementation. Since 1995, much work has focused on solving the aforementioned problems.

The resource-transition circuits (RTC) proposed by Xing et al. [192] and siphons are two different structural objects of Petri nets and used to develop deadlock control policies for AMS. They are related to the liveness property of Petri net models and thus used to characterize and avoid deadlocks. Based on them, there are two kinds of methods for developing deadlock controllers. Such methods rely on the computation of all maximal perfect RTC and SMS, respectively. The work in [192] concentrates on S^3PR, establishes the relation between two kinds of control methods, and identifies maximal perfect RTC and SMS. A graph-based technique is used to find all elementary RTC structures. They are then used to derive all RTC. Next, an iterative method is developed to recursively construct all maximal perfect RTC from elementary ones. Finally, a one-to-one correspondence between SMS and maximal perfect RTCs and hence an equivalence between two deadlock control methods are established [192].

In [132], Liu et al. define the concept of a controllable siphon basis, and propose a new criterion for selecting a proper subset of siphons to control. A controllable siphon basis is a subset of SMS whose complementary sets can cover complementary sets of all other SMS. By adding a control place to each SMS in a controllable siphon basis, Liu et al. present a new way to design a live Petri net controller for AMS and the number of its control places is no more than that of activity places in the original Petri net. Based on a controllable siphon basis, a novel deadlock prevention policy is established. It is shown that a controllable siphon basis and a set of elementary siphons are different subsets of SMS in an S^3PR. One can obtain a live Petri net controller based on a controllable siphon basis, while it may not be based on elementary siphons.

5.2. Mathematical programming

Because of the inherent complexity of Petri nets, any deadlock-prevention policy that depends on a complete siphon enumeration suffers definitely from an exponential complexity problem with respect to the size of its plant net model [101]. Given a Petri net, a maximal unmarked siphon can be obtained by the following traditional siphon solution. First, remove all the unmarked places. Then, remove the transitions without input places as well as their output places. Repeat the two steps until no places and transitions can be removed.

In [36], Chu and Xie first use mixed integer programming (MIP) [175] to detect whether a structurally bounded Petri net is deadlock-free. This method avoids the explicit enumeration of all SMS and opens a new research avenue. A feasible solution corresponds to a maximal unmarked siphon when there exists a siphon that can be emptied at a marking that is reachable from the initial marking. Otherwise, its optimal solution is equal to the number of all the places in the Petri net. Although an MIP problem is NP-hard in theory [175], extensive numerical studies show that its computational efficiency is relatively insensitive to the initial marking and is more efficient than those that depend on a complete state or siphon enumeration.

Motivated by the fact that deadlock control is usually concerned with minimal siphons in a Petri net, minimal siphon extraction methods from a maximal unmarked siphon are investigated [83,100]. Huang et al. propose an iterative two-stage deadlock prevention policy based on the work in [83]. At each iteration, an unmarked maximal siphon is detected by solving an MIP problem. If such a siphon exists, then an algorithm extracts an SMS from the maximal one. A monitor is added such that the minimal siphon is controlled by the enforcement that the set consisting of the monitor and the complementary set of the derived minimal siphon is the support of a P-semiflow. Repeat the aforementioned steps until all siphons in the original plant model are controlled. The second phase becomes necessary if the resulting net after the first phase contains deadlocks. At the second phase, a minimal siphon that contains at least a monitor is derived by solving an MIP problem. Then, a monitor is added to make the siphon controlled by leading the output arcs of the monitor to the source transitions of the plant model. This step is repeated until the MIP problem shows that there is no unmarked siphon at a reachable marking, implying that liveness is enforced. Experimental study shows that the two-phase policy is more permissive than those in [47] and [95], although no formal proof is provided. However, there is some uncertainty in the number of reachable states of the controlled system. This is not surprising since in the second phase the selection of different siphons to control can lead to liveness-enforcing controlled systems with different permissive behavior. Later, the idea is applied to ES^3PR [82] which is a more general class of Petri nets than S^3PR.

The work in [22] proposes the concept of basic and compound siphons. First, monitors are added to each basic siphon. Then, it finds conditions for a compound siphon to be implicitly controlled. This research avoids the problem of a full siphon enumeration and reduces the number of subsequent time-consuming MIP iterations.

It is not surprising that there exist redundant monitors in a liveness-enforcing Petri net supervisor, particularly when it is derived from an iterative siphon control approach. In [164], a redundant monitor is identified and eliminated by computing the reachability graph of a controlled system. If the removal of a monitor does not lead to the loss of liveness of the controlled system, it can be removed from the supervisor. The major drawback is the computational-complexity problem, since the reachability graph and liveness check are necessary. In order to avoid a complete state enumeration, Li and Hu [112] propose two methods to remove monitors from a Petri net supervisor. The first is based on the concept of implicit places [52]. It is shown that the implicity of a monitor is decided by solving an LPP that can be done in polynomial time. The second is derived from the MIP-based deadlock-detection method. If the removal of a monitor does not change the optimal solution of an MIP problem that is derived from the controlled system, then it is implicit or its removal may lead to more permissive behavior while liveness is preserved. In [13,54], and [22], the MIP technique is also used. Their methods can find a minimal unmarked siphon directly.

The work in [91] presents an iterative deadlock prevention policy for Petri nets using siphon extraction. At each iteration step, a siphon extraction algorithm finds a maximal deadly marked siphon (DMS), 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. By adopting the classification of places, deciding necessary siphons, and adding the proper CP, the deadlock prevention policy in [91] avoids a complete siphon enumeration, adds a small number of CP, and leads to a liveness-enforcing supervisor with a simple structure compared with closely related approaches in the literature.

In [199], based on DMS [145] in well-marked S^4R, Zhao et al. modify the MIP test in [36] to detect DMS for S^4R. However, an S^4R may have livelocks [126] even though it is deadlock-free. In this case, the siphons causing livelocks cannot be detected by the modified MIP and the net cannot be further controlled. Furthermore, the techniques in both [36] and [199] cannot obtain a minimal problematic siphon directly.

In [200], Zhong and Li propose an MIP model to detect a minimal non-max-marked siphon. However, their method cannot detect the siphons that cause livelocks. Furthermore, it outputs an SMS when a Petri net is live with non-max-marked siphons, impressing falsely that the net is non-live and thus needs a control place to control it.

In [126], the existing MIP-based methods are improved in the literature in terms of the max$''$-controllability condition of siphons. The authors define extended DMS (EDMS) and then develop a more general MIP model that can detect deadlocks and livelocks caused by siphons in an S^4R. They conclude that a net is live if there is no feasible solution for the MIP model. This programming is more powerful than the MIP in [199] and [200] but still restrictive since it outputs an SMS when a Petri net is live with non-max$''$-marked siphons.

Based on the concept of max*-controllability condition of siphons, in recent work [129], the authors formulate an integer programming (IP) model to detect the minimal non-max$*$-marked siphon that causes deadlocks or livelocks in GS^3PR. However, the use of the max*-controllability condition and the proposed IP model to perfectly control a generalized Petri net remains open, requiring a further study.

5.3. Elementary siphons

Even in a Petri net with a rather simple structure such as linear S^3PR, the number of its SMS in the worst case is proved to be exponential with respect to its size [166]. If all the SMS are explicitly controlled without considering any difference or relationship among them, the resulting supervisor is structurally complex in theory, as shown in [47].

To alleviate such a problem, Li and Zhou propose the concept of elementary siphons [95,96,104,105,109]. For a deadlock control purpose, problematic siphons (that can cause dead transitions) in a Petri net are divided into elementary and dependent ones. The latter is originally named as redundant siphons [95]. By the incidence matrix of a Petri net, Li and Zhou define the characteristic T-vector of a siphon, which is the sum of the rows corresponding to the places in the siphon, and indicates the change of the number of tokens in it when a transition fires. The characteristic T-vectors of all problematic siphons constitute a matrix called their characteristic T-vector matrix. From the matrix, a maximal linearly independent set of vectors can be found. The siphons corresponding to the vectors in this set are said to be elementary. The others are called dependent siphons that are further distinguished into weakly and strongly dependent ones by deciding whether the linear combination coefficients are all positive or not.

Two key contributions that underlie the concept of elementary siphons are as follows: (1) the number of elementary siphons in a Petri net is bounded by the smaller of the place and transition counts; and (2) a dependent siphon can be implicitly controlled by controlling its elementary ones. For deadlock prevention that is achieved by monitors, it is of significance that a dependent siphon can be controlled via explicit control of its elementary siphons by designing monitors and properly setting their control depth variables. An AMS example is investigated in [95] with its Petri net model having 26 places, 20 transitions, and 18 SMS. 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 18 strict minimal siphons. However, the work in [47] needs to design monitors for all 18 SMS. In [95], the controllability of a dependent siphon is explored with respect to elementary siphons that are invariant controlled. A more general controllability condition than that in [95] is developed in [104] for dependent siphons in ordinary Petri nets. In [105], Li and Zhao extend such results to generalized nets, which is based on the max-controlled siphons [8]. However, the computational complexity of supervisor design in [104] remains to be exponential with respect to the net size, since the computation of a set of elementary siphons depends on a complete siphon enumeration.

The computational efforts for a supervisor based on elementary siphons are reduced by introducing a siphon solution technique using MIP-based deadlock detection. The study in [102] proposes an iterative deadlock-prevention policy by using the concept of elementary siphons. At each iteration step, a maximal unmarked siphon is found by solving an MIP problem. Then, an SMS is extracted from the maximal unmarked one. If the siphon extracted is elementary with respect to the computed ones, it is explicitly controlled by a monitor. If it is dependent, its controllability is decided by checking whether it needs to be explicitly controlled. The iteration process terminates when no unmarked siphon is found in the controlled Petri net with monitors.

The work in [102] to a large extent reduces the computational cost to design a supervisor and the resulting supervisor's structural complexity compared with [47]. However, it does not improve the behavioral permissiveness. The work in [97] develops a two-phase deadlock-prevention policy. The first phase adds a monitor for each elementary siphon that is derived from the MIP-based deadlock-detection method. The output arcs of a monitor are led to the source transitions of a plant net model. The source transitions represent the entry of raw parts into a system. The second phase rearranges the output arcs of the monitors such that the transitions with which they are associated are away from the source transitions as far as possible if this rearrangement does not result in dead transitions. Such an improvement increases the behavioral permissiveness of a supervisor. The policies that underlie the idea of elementary siphons can also be found in [23,24,57–60,65,93,99,106,109,110,131,152,170].

As known, dependent siphons can be further divided into strongly and weakly dependent ones. An interesting work is done by Chao and Li in [23], which explores the structural condition in a class of Petri nets under which there exists a set of elementary siphons such that all the others are strongly dependent.

For S^3PR, the studies in [57] and [58] explore sufficient conditions with respect to an initial marking at which there exists a maximally permissive liveness-enforcing Petri net supervisor. Based on elementary siphons, algorithms with polynomial complexity are proposed to decide the existence of a maximally permissive supervisor for S^3PR. Their developments are based on the computation of a set of elementary siphons and siphons composition operations, which has been shown to be of polynomial complexity with respect to the size of a plant net model.

By utilizing the structural information in a Petri net, the concept of elementary siphons paves an effective and efficient way to design a structurally simple supervisor based on siphon control methods. In [59], based on the structural analysis technology, elementary and dependent siphons are redefined for classes of generalized Petri nets, for example S^4PR, by introducing augmented siphons. The redefined elementary siphons are more compact and well suitable for generalized Petri nets.

The concept of elementary siphons aims to simplify the structure of a liveness-enforcing Petri net supervisor, which can also be used in deadlock prevention for timed Petri nets [55]. The number of transitions in a supervisor is not greater than that of a plant model. As a result, its structural complexity is usually evaluated by the number of its monitors. However, in a general case, a deadlock-prevention policy based on elementary siphons cannot find a minimal supervisory structure.

Elementary siphons have been extensively studied to solve deadlock problems in Petri nets. However, most of the existing approaches based on elementary siphons can deal with the Petri nets in which all the transitions are supposed to be controllable and observable. In [152], Qin et al. develop an elementary-siphon-based deadlock control policy for AMS with uncontrollable and unobservable transitions. For elementary siphons, their complementary sets are successively expanded by considering unobservability and uncontrollability of transitions. Monitors are designed for the expanded complementary sets. The proposed method in [152] permits that there are arcs from monitors to a set of special uncontrollable transitions. Moreover, the dependent siphons are always marked in the controlled system that is derived from the proposed method when there exist uncontrollable and unobservable transitions.

5.4. Siphon control combined with reachability graphs

Recently, several deadlock control policies based on combination the state space and structural analysis have been proposed. The work in [176] can be considered as an improvement of the theory of regions [2,53,162,163]. It designs a supervisor for a plant net model with maximally permissive behavior by using the theory of regions. Then, the SMS in the maximally permissive controlled system are computed and divided into elementary and dependent ones. To prevent them from being emptied, algebraic expressions about the markings of the additional monitors in the supervisor and the resource places in the plant net model are derived, under which the supervisor is live. The expressions are used to derive the live initial markings for the supervisor without changing its structure when the initial marking of the plant changes. A case study shows that the combined method is computationally efficient compared to existing ones in which the theory of regions is used alone, and the permissive behavior of the supervisor is near-optimal.

Piroddi et al. believe that it is important to integrate the structural information related to SMS with reachability graph analysis to avoid unnecessary control places. The work in [148] develops a selective siphon control policy in which the concepts of essential, dominated, and dominating siphons and critical, dominating and dominated markings play an important role. By solving set covering problems, dominating siphons are found to ensure that dominated siphons are controlled. The resulting supervisor is highly permissive. The major technical problem in [148] is its computational complexity. At each iteration, it needs to compute all minimal siphons and all dominating markings and to solve a set covering problem, each of which is NP-hard in theory with respect to the net size. Later, in [149], Piroddi et al. improve the method by using the MIP-based deadlock detection approach such that the complete minimal siphon enumeration is avoided. From the case study, the improved version of the combined siphon and marking policy is computationally competitive. In [168], Wang et al. explain the iterative processes in [149] clearly.

To reduce the number of marking/transition separation instances (MTSI) when using the theory of regions, Li et al. develop a two-phase deadlock-prevention policy by siphon control and the theory of regions [103]. First, SMS are identified through resource circuits only and controlled by monitors, whose quantity is bounded by the smaller of place and transition counts. This leads to the fact that siphon identification and control is of polynomial complexity. Then, the theory of regions is applied to the augmented Petri net with monitors to find a supervisor. Since the siphon control in the first phase is optimal from deadlock-prevention point of view, the final supervisor is optimal if such a supervisor exists.

The most attractive advantage of the deadlock-prevention policy in [103] is that the number of separation instances is significantly reduced after some elementary siphons of a system are controlled. However, it fails to determine all sets of MTSI, and its application seems limited to some special nets only. In [85], Huang et al. propose crucial MTSI that allow designers to employ much fewer MTSI to deal with deadlocks. The advantage of the proposed policy is that an optimal deadlock controller can be obtained with drastically reduced computation. Huang et al. explain that once a dead marking is removed, i.e., a crucial MTSI is controlled, its corresponding quasi-dead markings are removed as well based on siphon theory. Once a minimal siphon S loses all tokens, it cannot regain a token. Thus, all the transitions in $S^\bullet$ are permanently disabled and a partial or total deadlock results. Other transitions may still be enabled. However, their firings eventually lead to a dead marking. It is known that a Petri net is deadlock-free if no siphon becomes empty. It indicates that no dead markings exist if no siphon eventually becomes empty. Hence, quasi-dead markings will vanish if no dead markings exist [85].

In [143], Pan et al. try to enhance the computational efficiency of the crucial MTSI method in [85]. The combined selective siphons and critical markings method in [148] is merged in [143]. The advantage of the proposed method is that the number of two types of crucial MTSI can then be simplified. The proposed policy can be applied to S^3PMR [84] instead of S^3PR nets only. Later, in [144], Pan et al. improve the work in [143] by reducing the number of LPP.

Motivated by the tight connections between directed graphs (digraphs) and Petri nets in deadlock control for AMS [50], Maione and DiCesare [138] propose a hybrid deadlock prevention policy by using directed graphs and Petri nets, taking the advantages of both techniques. In order to avoid searching siphons in the Petri net model of a system, deadlock detection is carried out by analyzing the structures of the digraph that models the system. Then, the digraph-based information is translated into the deadlock marking of the corresponding Petri net, which is used to design monitors to prevent empty siphons. Finally, a number of new control policies are developed, which are less restrictive than other efficient policies [138].

The work in [114] proposes a novel design method of deadlock prevention supervisors based on Petri nets, which does not guarantee optimality but empirical results show its superiority over other approaches based on siphon control. Given the Petri net model, one first designs an optimal liveness-enforcing controlled system for the model at a minimal initial marking by utilizing the theory of regions. Then, we calculate all SMS in the controlled system. Such a siphon does not contain a trap. For each SMS, an algebraic inequality with respect to the markings of monitors and resource places in the controlled system, also called a liveness constraint, is established in terms of the concept of max-controlled or invariant-controlled siphons. Its satisfaction implies the absence of dead transitions in the postset of the corresponding siphon. Consequently, given initial markings that satisfy all the liveness inequality constraints, all siphons can be max-controlled, and the resulting controlled system is live. After a controlled system structure is found, one can reallocate the initial markings according to the inequality constraints. No matter how large the initial markings and the number of states are, the liveness constraints remain unchanged. Their satisfaction ensures the absence of uncontrolled siphons. This implies that, for a plant model with a fixed net structure, we only need to compute its reachability graph at a minimal initial marking and the siphons of the controlled system once. Whenever the number of process instances and the capacity of manufacturing resources change, a Petri net supervisor can be determined easily via these algebraic inequality constraints.

5.5. Deadlock control policies based on siphons in Gadara nets

In the past decade, computer hardware has undergone a true revolution, moving from uniprocessor architectures to multiprocessor ones. In order to exploit the full potential of multicore hardware, there is an unprecedented interest in parallelizing computing tasks that were previously conducted in series. This trend forces parallel programming upon the average programmer. Parallel programming is fundamentally more challenging than serial programming because of the complexity of reasoning about concurrency. Lock primitives, such as mutual exclusion locks (mutexes), are often employed to protect shared data and prevent data races. Inappropriate use of mutexes can lead to circular wait deadlocks in the program, where a set of threads are waiting indefinitely for one another and none of them can proceed. Significant effort has to be spent to detect and fix deadlock bugs. Development of highly reliable and robust software is a very active research area in the software and operating systems communities. There is an emerging need for systematic methodologies that will enable programmers to characterize, analyze, and resolve software failures, such as deadlocks [121].

Decades of studies have yielded numerous approaches to program deadlock resolution, but none is a panacea. Static deadlock prevention via strict global lock-acquisition ordering is straightforward in principle but can be remarkably difficult to apply in practice. Static deadlock detection via program analysis has made impressive strides in recent years, but spurious warnings can be numerous and the cost of manually repairing genuine deadlock bugs remains high. Dynamic deadlock detection may identify the problem too late, when recovery is awkward or impossible [121]. The Gadara project is a multidisciplinary effort to develop a software tool that takes as input a deadlock-prone multithreaded C program and outputs a modified version of the program that is guaranteed to run deadlock-free without affecting any of the functionalities of the program [116–118,121,171]. They build a formal model of the program, analyze its properties, and synthesize control logic to enforce deadlock-freeness.

Various special classes of Petri nets have been proposed to analyze manufacturing systems [109]. The existing special classes of Petri nets in the literature do not exactly match the specific features of Petri nets that arise when modeling the locking behavior of multithreaded programs. Therefore, Wang et al. propose a new class of Petri nets, called Gadara nets [121,135,171,173,174] that explicitly model multithreaded programs with lock acquisition and release operations. With the class of Gadara nets formally defined, we can efficiently analyze program deadlocks via formal models, and synthesize deadlock avoidance policies that can in turn be instrumented in the underlying programs. By establishing a set of important properties of Gadara nets (e.g., liveness and reversibility), the deadlock-freeness of the program can be analyzed via the program's corresponding Gadara net model by exploiting the structural properties of the net. This correspondence is crucial to the effectiveness and efficiency of control synthesis of Gadara nets for the purpose of deadlock avoidance in the programs.

The work [171] is the first paper which addresses Gadara nets. Wang et al. apply the following permissiveness-preserving heuristics for offline control synthesis algorithm. First, a siphon containing fewer than two mutex places is controlled. This is a well-known result in [47]. Second, minimal siphons are controlled only; the minimal siphon induced by a subset of mutexes is unique. The first recipe is well-known. The uniqueness of the siphon stems from the fact that the Petri net model of each function itself (without any mutex place) is a state machine. Third, siphons among circular waiting mutexes are calculated only. A siphon induced by a subset of mutexes without circular waiting is a controlled siphon. Wang et al. first traverse through the net to find lock dependencies. For every subset of locks with a cycle, the minimal siphon is calculated. Fourth, among equivalent siphons, control only one. Among a set of equivalent siphons [95], controlling the one with minimum number of initial tokens, called the token poor siphon [42,106], guarantees that the others are never empty. Fifth, control places with redundant logic checking are removed. For simplicity, the authors compare the control logic between newly-generated control places and existing ones. If the control logic of one control place is more permissive than that of another, i.e., its token is always available when requested, it is removed.

The work in [172] formally defines Gadara nets. The contributions of this paper include: (i) formal definition of Gadara nets and of controlled Gadara nets; (ii) behavioral analysis of Gadara nets for liveness and reversibility using siphons; and (iii) identification of a convexity-type property for the set of live markings.

In [119], the authors thoroughly exploit the structural properties of Gadara nets for the efficient synthesis of maximally-permissive liveness-enforcing (MPLE) control policies. In general, the proposed MPLE control synthesis is an iterative process [172], since the synthesized control logic may introduce new potential deadlocks. That is, the added net structure, when coupled with the original net structure, may cause new potential deadlocks in the controlled net. This necessitates iterations on the controlled nets until no further deadlock is found. Few works address such an iterative process and its implications for MPLE control synthesis. A siphon-based iterative control synthesis method is proposed in [160] for Gadara nets. However, this method is sub-optimal in general, i.e., it does not guarantee maximal permissiveness. In [86], the role of iterations in liveness-enforcing control synthesis is discussed and a net transformation technique is employed to transform non-ordinary nets into PT-ordinary nets during the iterations. This approach, however, may not guarantee convergence within a finite number of iterations. In fact, as pointed out in [115], it is not easy to establish a formal and satisfactory proof of finite convergence for this type of problems; moreover, achieving optimal control logic is very difficult. The key reason is that the Petri net modeling framework might not be able to express the MPLE property for general process-resource net. As a result, the problem of MPLE control synthesis based on siphon analysis in non-ordinary nets has not been well-resolved yet.

Liao et al. present a new control synthesis algorithm for liveness enforcement of Gadara nets that need not be ordinary [119]. The algorithm employs structural analysis of the net and synthesizes monitor places to prevent the formation of a special class of siphons, termed resource-induced deadly-marked siphons. The algorithm also accounts for uncontrollable transitions in the net in a minimally restrictive manner. The algorithm is generally an iterative process and converges in a finite number of iterations. It exploits a covering of the unsafe states that are updated at each iteration. The proposed algorithm is shown to be correct and maximally permissive with respect to the goal of liveness enforcement.

The main contributions in [119] can be summarized as follows: (i) a new iterative control synthesis scheme (called ICOG) for Gadara nets is reported; this scheme is based on structural analysis and converges in finite iterations. (ii) a new algorithm (called UCCOR) for controlling siphons in Gadara nets is developed; it uses the notion of covering of unsafe states (markings) in order to achieve greater computational efficiency. (iii) The UCCOR Algorithm accounts for uncontrollable transitions in the net in a minimally restrictive manner by the technique of constraint transformation. (iv) It is shown that the proposed ICOG methodology and the associated UCCOR algorithm synthesize a control policy that is correct and maximally permissive with respect to the goal of liveness enforcement.

Later, the work in [120] proposes an efficient optimal control synthesis methodology for ordinary Gadara nets. It exploits the structural properties of Gadara nets via siphon analysis. Optimality in this context refers to the elimination of deadlocks in the program with minimally restrictive control logic. The authors formally establish a set of important properties of the proposed control synthesis methodology, and show that their algorithms never synthesize redundant control logic. The study in [120] conducts experiments to evaluate the efficiency and scalability of the proposed methodology, and discusses the application of their results to real-world concurrent software.

Liao et al. propose an iterative control synthesis methodology for ordinary Gadara nets, called ICOG-O, based on structural analysis in terms of siphons. The control logic synthesized by ICOG-O enforces liveness in Gadara nets and provably eliminates all the potential deadlocks in the corresponding multithreaded programs. Compared with the work in [119], the customized ICOG-O presented in [120] focuses on ordinary Gadara nets, and thus enable us to implement control synthesis based on a type of empty siphons. Experimental results show that ICOG-O is very efficient in terms of time and the number of synthesized monitor places. The results also demonstrate the scalability of this approach to large-scale real-world software. From a more general perspective, the results in the Gadara project illustrat that software failure avoidance is a fertile application area for discrete event control, and moreover, special features from this application area are motivating further theoretical developments on the control of discrete-event systems.

5.6. Robust deadlock control policies based on siphons in Petri nets

All the studies reviewed in the previous section assume that resources do not fail. Actually, resource failures are inevitable in most real-world systems, which may also cause a system to be deadlocked. It is a necessary requirement to develop an effective and robust deadlock control policy to ensure that deadlocks cannot occur even if some resources in a system break down. There is a lack of research in Petri nets regarding the impacts of unreliable resources on resource allocation systems under the supervisory control of deadlocks.

Hsieh develops a variety of methods to determine the feasibility of production with a set of resource failures modeled as the extraction of tokens from a Petri net [62–64]. In these works, liveness conditions and robustness analysis of the nets are based on the concepts of token flow paths and minimal resource requirements. His work reports fault tolerant conditions and proposes a structural decomposition method to test the feasibility of production routes. However, all these methods are not intuitive to the Petri net models.

Recently, the work in [127] makes the existing deadlock control policies possess a desirable robust property to cope with resource failures. Specifically, the desirable robustness is a system property to keep a controlled system live as some resources break down. In [127], the authors focus on robust supervision of manufacturing systems. For S^3PR, it bridges the gap between a divide-and-conquer deadlock control strategy [110] and its application to real-world systems with unreliable resources. Recovery subnets and monitors are designed for unreliable resources and SMS that may be emptied, respectively. Normal and inhibitor arcs are used to connect monitors with recovery subnets in case of necessity. Then reanalysis of the original Petri net is avoided and a robust liveness-enforcing supervisor is derived. The supervisors designed for S^3PR by the proposed method has following properties : (1) they can prevent deadlocks for plant models when all resources work normally ; (2) deadlocks are prevented even if some resources fail to work and are removed to repair at any time ; and (3) waiting-for-repair states disappear after the repaired resources are returned.

6. Open problems

So far, many deadlock control policies based on siphon control have been proposed for Petri nets. Despite the points that have been reviewed and discussed above, from many aspects, the study of siphons in different subclasses of Petri nets can be further extended in the future.

The use of the max*-controllability condition to control a generalized Petri net is still an open problem requiring a further study. Also, a sufficient and necessary siphon control condition for G-systems, S^4R, and S^nPR family remains open.

In [204], Zhong et al. point out that a maximally permissive liveness-enforcing pure net supervisor does not always exist for any Petri net. The work in [204] focuses on the non-existence of maximally permissive liveness-enforcing pure net supervisors for a subclass of Petri nets. It shows that if a net contains a siphon that cannot be optimally controlled by a P-invariant, the net does not admit a maximally permissive liveness-enforcing pure net supervisor. Recently, based on the technique of reachability graph analysis and self-loop structures of Petri nets, Chen et al. propose nonpure Petri net supervisors for optimal deadlock control of AMS [32]. Their methods can find maximally permissive liveness-enforcing supervisors for the class of Petri nets mentioned in [204]. It is challenging to design an optimal supervisor for generalized Petri nets based on siphons. Self-loops and interval inhibitor arcs [33] look promising in the design of optimal supervisors for Petri net in the future.

In [128], a maximally permissive control policy for a subclass of S^3PR based on the new ground-breaking theory of token distribution pattern of unmarked siphons is proposed. This has the advantage of avoiding the computation of new siphons derived from monitor places since the unmarked pattern solely determines the controller region (or control arcs) and the initial marking. There is no need to construct the reachability tree and hence the problem is no longer NP-hard. This results in fewer monitors and more reachable states as shown by two well-known S^3PR. Future work should consider how to add minimal number of weighted control arcs to make controlled nets maximally permissive and to extend to other complicated nets.

For manufacturing systems with unreliable resources, no much work is found on robust supervision design based on Petri nets. The work [127] only seeks to simple or even naive solution for S^3PR. There is an obvious drawback in this study. The final supervisor for S^3PR is too complex in structure even though the algorithm is easily implemented in theory. In fact, robust liveness-enforcing supervisor design is much more difficult for complex manufacturing systems. Further, more effective methods with low computational overheads and simple structural complexity can be potentially studied in the future.

As far as we know, most of researchers in Petri net community model AMS by process-oriented Petri nets (POPN). Recently, in deadlock resolution for AMS, another type of Petri nets called resource-oriented Petri nets (ROPN) is developed in [177–188]. These studies present a different way to control deadlock based on ROPN. Especially, in [186], Wu and Zhou show that there is a close relationship between a production process circuit (PPC) in an ROPN and a siphon in an S^3PR. We can draw inspiration from this internal relation. Siphons and PPC can be combined to control deadlocks. Also the analysis and computation of siphons and other structural objects in context of high level [5,154] or timed Petri nets need to extend to be operational.

7. Conclusions

As a structural object, siphons play an important role in the analysis of structural and behavioral properties of Petri nets. This paper surveys the state-of-the-art siphon theory of Petri nets including basic concepts, siphons computation, controllability conditions, and deadlock control policies. Some open problems are discussed, such as the maximally permissive supervisor design problems and their application to robust control area.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China under Grant nos. 61304051 and 61374068 and the Fundamental Research Funds for the Central Universities under Grant no. JB150404.

References

[1] I.B. Abdallah, H.A. ElMaraghy, T. ElMekkawy, A logic programming approach for finding minimal siphons in S^3PR nets applied to manufacturing systems, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, vol. 2, 1997, pp. 1710–1715.

[2] E. Badouel, P. Darondeau, Theory of regions, Lect. Notes in Comput. Sci. 1491 (1998) 529–586.
[3] Z.A. Banaszak, B.H. Krogh, Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows, IEEE Trans. Robot. Autom. 6 (6) (1990) 724–734.
[4] K. Barkaoui, B. Lemaire, An effective characterization of minimal deadlocks and traps in Petri nets based on graph theory, in: Proceedings of the 10th International Conference on Application and Theory of Petri Nets, 1989, pp. 1–21.
[5] K. Barkaoui, C. Dutheillet, S. Haddad, An efficient algorithm for finding structural deadlocks in colored Petri nets, Lecture Notes in Computer Science 691 (1993) 69–88.
[6] K. Barkaoui, I.B. Abdallah, Deadlock avoidance in FMS based on structural theory of Petri nets, in: Proceedings of 1995 INRIA/IEEE Symposium on Emerging Technologies and Factory Automation, vol. 2, 1995a, pp. 499–510.
[7] K. Barkaoui, I.B. Abdallah, A deadlock prevention method for a class of FMS, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, vol. 5, 1995b, pp. 4119–4124.
[8] K. Barkaoui, J.F. Peyre, On liveness and controlled siphons in Petri nets, in: Proceeding of the 17th International Conference on Application and Theory of Petri Nets, Lecture Notes in Computer Science, vol. 1091, 1996, pp. 57–72.
[9] K. Barkaoui, A. Chaoui, B. Zouari, Supervisory control of discrete event systems based on structure theory of Petri nets, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, vol. 4, 1997, pp. 3750–3755.
[10] K. Barkaoui, L. Petrucci, Structural analysis of workflow nets with shared resources, in: Proceedings of the Workshop Workflow Manage: Net-Based Concepts, Models, Technology Tools, 1998, pp. 82–95.
[11] K. Barkaoui, J.M. Couvreur, K. Klai, On the equivalence between liveness and deadlock-freeness in petri nets, in: Proceeding of the 26th International Conference on Application and Theory of Petri Nets, Lecture Notes in Computer Science, vol. 3536, 2005, pp. 90–107.
[12] K. Barkaoui, M. Minoux, A polynomial-time graph algorithm to decide liveness of some basic classes of bounded Petri nets, Lecture Notes in Computer Science 616 (1992) 62–75.
[13] F. Basile, P. Chiacchio, A. Giua, C. Seatzu, Deadlock recovery of Petri net models controlled using observers, in: Proceeding of the 8th IEEE International Conference on Emerging Technologies and Factory Automation, 2001, pp. 441–449.
[14] E. Best, H.A. Schmid, Systems of open paths in Petri nets, Lecture Notes in Computer Science vol. 32 (1975) 186–193.
[15] E.R. Boer, T. Murata, Generating basis siphons and traps of Petri nets using the sign incidence matrix, IEEE Transactions on Circuits and Systems-I 41 (4) (1994) 266–271.
[16] E.E. Cano, C.A. Rovetto, J.M. Colom, On the computation of the minimal siphons of S^4PR nets from a generating family of siphons, in: Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation(ETFA), 2010, pp. 1–8.
[17] D.Y. Chao, Computation of elementary siphons in Petri nets for deadlock control, Comput. J. 49 (4) (2006) 470–479.
[18] D.Y. Chao, Max′-controlled siphons for liveness of S^3PGR^2, IET Control Theory & Appl. 1 (4) (2007a) 933–936.
[19] D.Y. Chao, An incremental approach to extract minimal bad siphons, J. Inf. Sci. Eng. 23 (1) (2007b) 203–214.
[20] D.Y. Chao, A graphic algebraic computation of elementary siphons of BS^3PR, J. Inf. Sci. Eng. 23 (2007c) 1817–1831.
[21] D.Y. Chao, Incremental approach to computation of elementary siphons for arbitrary simple sequential processes with resources, IET Control Theory & Appl. 2 (2) (2008) 168–179.
[22] D.Y. Chao, Technical Note-MIP iteration reductions for deadlock prevention of flexible manufacturing systems, Int. J. Adv. Manuf. Technol. 41 (3) (2009) 343–346.
[23] D.Y. Chao, Z.W. Li, Structural conditions of systems of simple sequential processes with resources nets without weakly dependent siphons, IET Control Theory & Appl. 3 (4) (2009) 391–403.
[24] D.Y. Chao, Conservative control policy for weakly dependent siphons in S^3PR based on elementary siphons, IET Control Theory and Appl. 4 (7) (2010) 1298–1302.
[25] Y.F. Chen, Z.W. Li, Design of a maximally permissive liveness-enforcing supervisor with compressed supervisory structure for flexible manufacturing systems, Automatica 47 (5) (2011) 1028–1034.
[26] Y.F. Chen, Z.W. Li, M. Khalgui, O. Moshabi, Design of maximally permissive liveness-enforcing petri net supervisor for flexible manufacturing systems, IEEE Trans. Autom. Sci. Eng. 8 (2) (2011) 374–393.
[27] Y.F. Chen, Z.W. Li, M.C. Zhou, Behaviorally optimal and structurally simple liveness-enforcing supervisors of flexible manufacturing systems, IEEE Trans. Syst. Man, and Cybern. Part A 42 (3) (2012) 615–629.
[28] Y.F. Chen, Z.W. Li, On structural minimality of optimal supervisors for flexible manufacturing systems, Automatica 48 (10) (2012) 2647–2656.
[29] Y.F. Chen, Z.W. Li, M.C. Zhou, Most permissive petri net supervisors for flexible manufacturing systems, Int. J. Prod. Res. 50 (22) (2012) 6357–6371.
[30] Y.F. Chen, Z.W. Li, Optimal Petri net Supervisors for Flexible manufacturing Systems, CRC Press, 2013.
[31] Y.F. Chen, G.Y. Liu, Computation of minimal siphons in Petri nets by using binary decision diagrams, ACM Trans. Embedded Comput. Syst. 12 (1) (2013) 1–15.
[32] Y.F. Chen, Z.W. Li, A. Al-Ahmari, Nonpure petri net supervisors for optimal deadlock control of flexible manufacturing systems, IEEE Trans. Syst. Man Cybern.Syst. 43 (2) (2013) 252–265.
[33] Y.F. Chen, Z.W. Li, K. Barkaoui, M. Uzam, New Petri net structure and its application to optimal supervisory control: interval inhibitor arcs, IEEE Trans. Syst. Man Cybern.-Syst. 44 (10) (2014a) 1384–1400.
[34] Y.F. Chen, Z.W. Li, M.C. Zhou, Optimal supervisory control of flexible manufacturing systems by Petri nets: a set classification approach, IEEE Trans. Autom. Sci. Eng. 11 (2) (2014b) 549–563.
[35] Y.F. Chen, Z.W. Li, K. Barkaoui, A. Giua, On the enforcement of a class of nonlinear constraints on Petri nets, Automatica 55 (2015) 116–124.
[36] F. Chu, X. Xie, Deadlock analysis of Petri nets using siphons and mathematical programming, IEEE Trans. Robot. Autom. 13 (6) (1997) 793–804.
[37] F. Commoner, A.W. Holt, Marked directed graphs, J. Comput. Syst. Sci. 5 (5) (1971) 511–523.
[38] L.W. Cooprider, 1976. Technical Report.
[39] R. Cordone, L. Ferrarini, L. Piroddi, Characterization of minimal and with predicate logic and binary basis siphons programming, in: Proceedings of the IEEE International Symposium on Computer Aided Control System Design, 2002, pp. 193–198.
[40] R. Cordone, L. Ferrarini, L. Piroddi, Some results on the computation of minimal siphons in Petri nets, in: Proceedings of the 42nd IEEE Conference on Decision and Control, 2003, pp. 3754–3759.
[41] R. Cordone, L. Ferrarini, L. Piroddi, Enumeration algorithms for minimal siphons in Petri nets based on place constraints, IEEE Trans. Syst. Man, and Cybern. Part A 35 (6) (2005) 844–854.
[42] D.C. Craig, W.M. Zuberek, Efficient siphon-based deadlock detection in Petri nets, in: Proceedings of the 3rd International Conference on Computer Science and Information Systems, Athens, Greece, 2007, pp. 23–26.
[43] R. David, H. Alla, Petri nets for modeling of dynamic systems: A survey, Automatica 30 (2) (1994) 175–202.
[44] J. Desel, J. Esparza, Free Choice Petri Nets, Cambridge University Press, London, 1995.
[45] J. Esparza, M. Silva, Circuits, handles, bridges, and nets, Lect. Notes in Comput. Sci. 483 (1990) 210–242.
[46] J. Ezpeleta, J.M. Couvreur, M. Silva, A new technique for finding a generating family of siphons, traps, and st-components: application to colored Petri nets, Lect. Notes in Comput. Sci. 674 (1993) 126–147.
[47] J. Ezpeleta, J.M. Colom, J. Martinez, A Petri net based deadlock prevention policy for flexible manufacturing systems, IEEE Trans. Robot. Autom. 11 (2) (1995) 173–184.
[48] J. Ezpeleta, F.G. Vallés, J.M. Colom, A class of well structured Petri nets for flexible manufacturing systems, Lect. Notes in Comput. Sci. 1420 (1998) 64–83.
[49] J. Ezpeleta, F. Tricas, F.G. Vallés, J.M. Colom, A banker's solution for deadlock avoidance in FMS with flexible routing and multiresource states, IEEE Trans. Robot. Autom. 18 (4) (2002) 621–625.

[50] M.P. Fanti, B. Maione, B. Turchiano, Comparing digraph and Petri net approaches to deadlock avoidance in FMS, IEEE Trans. Syst. Man, and Cybern. Part B 30 (5) (2000) 783–798.
[51] M.P. Fanti, M.C. Zhou, Deadlock control methods in automated manufacturing systems, IEEE Trans. Syst. Man, and Cybern. Part A 34 (1) (2004) 5–22.
[52] F. García-Vallés, J.M. Colom, Implicit places in net systems, in: Proceedings of the 8th International Workshop on Petri Nets and Performance Models, Zaragoza, Spain, 1999, pp. 104–113.
[53] A. Ghaffari, N. Rezg, X.L. Xie, Design of a live and maximally permissive Petri net controller using the theory of regions, IEEE Trans. Robot. Autom. 19 (1) (2003) 137–142.
[54] A. Giua, C. Seatzu, Liveness enforcing supervisors for railway networks using ES^2PR Petri nets, in: Proceedings of the 6th International Workshop on Discrete Event System, 2002, pp. 55–60.
[55] J.W. Guo, Z.W. Li, A deadlock prevention approach for a class of timed Petri nets using elementary siphons, Asian J. Control 12 (3) (2010) 347–363.
[56] M.H.T. Hack, Analysis of production schemata by Petri nets, TR94, Project MAC, MIT, 1972.
[57] Y.F. Hou, Z.W. Li, A.M. Al-Ahmari, Optimal controllability of 2-composed siphons in a class of Petri nets, Electron. Lett. 48 (24) (2012) 1535–1537.
[58] Y.F. Hou, Z.W. Li, L. Hong, A.M. Al-Ahmari, Optimal controllability of 3-composed siphons in a class of Petri nets, Electron. Lett. 49 (11) (2013) 697–699.
[59] Y.F. Hou, Z.W. Li, A.M. Al-Ahmari, A.A.M. El-Tamimi, E.A. Nasr, Extended elementary siphons and their application to liveness-enforcement of generalized Petri nets, Asian J. Control 16 (6) (2014a) 1789–1810.
[60] Y.F. Hou, Z.W. Li, M. Zhao, D. Liu, Extended elementary siphon-based deadlock prevention policy for a class of generalized Petri nets, Int. J. Comput. Integr. Manuf. 27 (1) (2014b) 85–102.
[61] Y.F. Hou, Z.W. Li, Extraction of elementary siphons in a class of generalized Petri nets using graph theory, Eng. Comput. 31 (2) (2014) 331–352.
[62] F.S. Hsieh, Robustness of deadlock avoidance algorithms for sequential processes, Automatica 39 (10) (2003) 1695–1706.
[63] F.S. Hsieh, Robustness analysis of Petri nets for assembly/ disassembly processes with unreliable resources, Automatica 42 (7) (2006) 1159–1166.
[64] F.S. Hsieh, Robustness analysis of holonic assembly/disassembly processes with Petri nets, Automatica 44 (10) (2008) 2538–2548.
[65] H.S. Hu, Z.W. Li, An optimal elementary siphons based iterative deadlock prevention policy for flexible manufacturing systems, Int. J. Adv. Manuf. Technol. 38 (3-4) (2008) 309–320.
[66] H.S. Hu, Z.W. Li, Efficient deadlock prevention policy in automated manufacturing systems using exhausted resources, Int. J. Adv. Manuf. Technol. 40 (5-6) (2009a) 566–571.
[67] H.S. Hu, Z.W. Li, Modeling and scheduling for manufacturing grid workflows using timed Petri nets, Int. J. Adv. Manuf. Technol. 42 (5-6) (2009b) 553–568.
[68] H.S. Hu, Z.W. Li, Local and global deadlock prevention policies for resource allocation systems using partially generated reachability graphs, Comput. Ind. Eng. 57 (4) (2009c) 1168–1181.
[69] H.S. Hu, Z.W. Li, Liveness enforcing supervision in video streaming systems using siphons, J. Inf. Sci. Eng. 25 (6) (2009d) 1863–1884.
[70] H.S. Hu, Z.W. Li, Clarification on the computation of liveness-enforcing supervisor for resource allocation systems with uncontrollable behavior, IEEE Trans. Autom. Sci. Eng. 6 (3) (2009e) 557–559.
[71] H.S. Hu, M.C. Zhou, Z.W. Li, Liveness enforcing supervision of video streaming systems using non-sequential Petri nets, IEEE Trans. Multimedia 11 (8) (2009) 1457–1465.
[72] H.S. Hu, M.C. Zhou, Z.W. Li, Supervisor optimization for deadlock resolution in automated manufacturing systems with Petri nets, IEEE Trans. Automa. Sci. Eng. 8 (4) (2011) 794–804.
[73] H.S. Hu, M.C. Zhou, Z.W. Li, Algebraic synthesis of timed supervisor for automated manufacturing systems using Petri nets, IEEE Trans. Autom. Sci. Eng 7 (3) (2010a) 549–557.
[74] H.S. Hu, M.C. Zhou, Z.W. Li, Low-cost and high-performance supervision in ratio-enforced automated manufacturing systems using timed Petri nets, IEEE Trans. Autom. Sci. Eng. 7 (4) (2010b) 933–944.
[75] H.S. Hu, Z.W. Li, Synthesis of liveness enforcing supervisor for automated manufacturing systems using insufficiently marked siphons, J. Intell. Manuf. 21 (4) (2010) 555–567.
[76] H.S. Hu, Z.W. Li, A. Al-Ahmari, Reversed fuzzy Petri nets and their application for fault diagnosis, Comput. Ind. Eng. 60 (4) (2011a) 505–510.
[77] H.S. Hu, M.C. Zhou, Z.W. Li, Supervisor design to enforce production ratio and absence of deadlock in automated manufacturing systems, IEEE Trans. Syst. Man, and Cybern. Part A 41 (2) (2011b) 201–212.
[78] H.S. Hu, M.C. Zhou, Z.W. Li, Supervisor optimization for deadlock resolution in automated manufacturing systems with Petri nets, IEEE Trans. Autom. Sci. Eng. 8 (4) (2011c) 794–804.
[79] H.S. Hu, M.C. Zhou, Z.W. Li, Liveness and ratio-enforcing supervision of automated manufacturing systems using Petri nets, IEEE Trans. Syst. Man, and Cybern. Part A 42 (2) (2012) 392–403.
[80] H.S. Hu, M.C. Zhou, Z.W. Li, Y. Tang, An optimization approach to improved Petri net controller design for automated manufacturing systems, IEEE Trans. Autom. Sci. Eng. 10 (3) (2013a) 772–782.
[81] H.S. Hu, M.C. Zhou, Z.W. Li, Deadlock-free control of automated manufacturing systems with flexible routes and assembly operations using Petri nets, IEEE Trans. Ind. Inf. 9 (1) (2013b) 109–121.
[82] Y.S. Huang, M.D. Jeng, X.L. Xie, S.L. Chung, A deadlock prevention policy for flexible manufacturing systems using siphons, in: Proceeding of IEEE International Conference on Robotics and Automation, 2001a, pp. 541–546.
[83] Y.S. Huang, M.D. Jeng, X.L. Xie, S.L. Chung, Deadlock prevention policy based on Petri nets and siphons, Int. J. Prod. Res. 39 (2) (2001b) 283–305.
[84] Y.S. Huang, M.D. Jeng, X.L. Xie, D.H. Chung, Siphon-based deadlock prevention policy for flexible manufacturing systems, IEEE Trans. Syst. Man, and Cybern. Part A 36 (6) (2006) 2152–2160.
[85] Y.S. Huang, Y.L. Pan, M.C. Zhou, Computationally improved optimal deadlock control policy for flexible manufacturing systems, IEEE Trans. Syst. Man, and Cybern. Part A 42 (2) (2012) 404–415.
[86] M.V. Iordache, P.J. Antsaklis, Supervisory Control of Concurrent Systems: A Petri Net Structural Approach, Birkhäuser, Boston, MA, 2006.
[87] M.D. Jeng, X.L. Xie, Analysis of modularly composed nets by siphons, IEEE Trans. Syst. Man, and Cybern. Part A 29 (4) (1999) 399–406.
[88] M.D. Jeng, X.L. Xie, M.Y. Peng, Process nets with resources for manufacturing modeling and their analysis, IEEE Trans. Robot. Autom. 18 (6) (2002) 875–889.
[89] M.D. Jeng, X.L. Xie, S.L. Chung, $ERCN^*$ merged nets for modeling degraded behavior and parallel processes in semiconductor manufacturing systems, IEEE Trans. Syst. Man, and Cybern. Part A 34 (1) (2004) 102–112.
[90] K. Lautenbach, Linear algebraic calculation of deadlocks and traps, in: Voss, Genrich, Rozemberg (Eds.), Concurrency and Nets, Springer Verlag, Berlin, Germany, 1987, pp. 315–336.
[91] S.Y. Li, Z.W. Li, H.S. Hu, Siphon extraction for deadlock control in flexible manufacturing systems by using Petri nets, Int. J. Comput. Integr. Manuf. 24 (8) (2011) 710–725.
[92] S.Y. Li, Z.W. Li, Structure reduction of liveness-enforcing Petri nets using mixed integer programming, Asian J. Control 14 (2) (2012) 384–399.
[93] S.Y. Li, Z.W. Li, H.S. Hu, A. Al-Ahmari, An extraction algorithm of a set of elementary siphons based on mixed-integer programming, J. Syst. Sci. Syst. Eng. 21 (1) (2012) 106–125.
[94] S.Y. Li, Z.W. Li, Solving siphons with the minimal cardinality in Petri nets and its applications to deadlock control, Int. J. Prod. Res. 50 (22) (2012) 6203–6218.
[95] Z.W. Li, M.C. Zhou, Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems, IEEE Trans. Syst. Man, and Cybern. Part A 34 (1) (2004) 38–51.
[96] Z.W. Li, M.C. Zhou, Clarifications on the definitions of elementary siphons in Petri nets, IEEE Trans. Syst. Man, and Cybern. Part A 36 (6) (2006a) 1227–1229.
[97] Z.W. Li, M.C. Zhou, Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets, IEEE Trans. Ind. Inf. 2 (4) (2006b) 313–325.
[98] Z.W. Li, J. Zhang, M. Zhao, Liveness-enforcing supervisor design for a class of generalized Petri net models of flexible manufacturing systems, IET Control Theory and Appl. 1 (4) (2007) 955–967.

[99] Z.W. Li, N. Wei, Deadlock control of flexible manufacturing systems via invariant-controlled elementary siphons of Petri nets, Int. J. Adv. Manuf. Technol. 33 (2007) 24–35.
[100] Z.W. Li, D. Liu, A.c. m. s. e. a. f. a. m. u. s. o. a. P. net, International Journal of Production Research 45 (9) (2007a) 2161–2165.
[101] Z.W. Li, M.C. Zhou, M. Uzam, Deadlock control policy for a class of Petri nets without complete siphon enumeration, IET Control Theory & Appl. 1 (6) (2007b) 1594–1605.
[102] Z.W. Li, H.S. Hu, A.R. Wang, Design of liveness-enforcing supervisors for flexible manufacturing systems using Petri nets, IEEE Trans. Syst. Man, and Cybern. Part C 37 (4) (2007c) 517–526.
[103] Z.W. Li, M.C. Zhou, M.D. Jeng, A maximally permissive deadlock prevention policy for FMS based on Petri net siphon control and the theory of regions, IEEE Trans. Autom. Sci. Eng. 5 (1) (2008) 182–188.
[104] Z.W. Li, M.C. Zhou, Control of elementary and dependent siphons in Petri nets and their application, IEEE Trans. Syst. Man, and Cybern. Part A 38 (1) (2008) 133–148.
[105] Z.W. Li, M. Zhao, On controllability of dependent siphons for deadlock prevention in generalized Petri nets, IEEE Trans. Syst. Man, and Cybern. Part A 38 (2) (2008) 369–384.
[106] Z.W. Li, M.C. Zhou, N.Q. Wu, A survey and comparison of Petri net-based deadlock prevention policy for flexible manufacturing systems, IEEE Trans. Syst. Man, and Cybern. 38 (2) (2008) 173–188.
[107] Z.W. Li, M.C. Zhou, On siphon computation for deadlock control in a class of Petri nets, IEEE Trans. Syst. Man, and Cybern. Part A 38 (3) (2008) 667–679.
[108] Z.W. Li, M. Uzam, M.C. Zhou, Deadlock control of concurrent manufacturing processes sharing finite resources, Int. J. Adv. Manuf. Technol. 38 (7-8) (2008) 787–800.
[109] Z.W. Li, M.C. Zhou, Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach, Springer Press, London, 2009.
[110] Z.W. Li, S. Zhu, M.C. Zhou, A divide-and-conquer strategy to deadlock prevention in flexible manufacturing systems, IEEE Trans. Syst. Man, and Cybern. Part C 39 (1) (2009) 156–169.
[111] Z.W. Li, M. Shiptalni, Smart deadlock prevention policy for flexible manufacturing systems using Petri nets, IET Control Theory & Appl. 3 (3) (2009) 362–374.
[112] Z.W. Li, H.S. Hu, On systematic methods to remove redundant monitors from liveness-enforcing net supervisors, Comput. Ind. Eng. 56 (1) (2009) 53–62.
[113] Z.W. Li, M.M. Yan, M.C. Zhou, Synthesis of structurally simple supervisors enforcing generalized mutual exclusion constraints in Petri nets, IEEE Trans. Syst. Man, and Cybern. Part C 40 (3) (2010) 330–340.
[114] Z.W. Li, G.Y. Liu, H.M. Hanisch, M.C. Zhou, Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems, IEEE Trans. Syst. Man, and Cybern. Part A 42 (1) (2012a) 178–191.
[115] Z.W. Li, N.Q. Wu, M.C. Zhou, Deadlock control of automated manufacturing systems based on Petri nets: a literature review, IEEE Trans. Syst. Man, and Cybern. Part C 42 (2012b) 437–462.
[116] H.W. Liao, S. Lafortune, S. Reveliotis, Y. Wang, S. Mahlke, Synthesis of maximally-permissive liveness-enforcing control policies for Gadara Petri nets, in: Proceedings of the 49th IEEE Conference on Decision and Control, Atlanta, USA, December 15–17, 2010, pp. 2797–2804.
[117] H.W. Liao, H. Zhou, S. Lafortune, Similation analysis of multithreaded programs under deadlock avoidance control abstract, in: Proceedings of the 2011 Winter Simulation Conference, 2011a, pp. 703–715.
[118] H.W. Liao, J. Stanley, Y. Wang, S. Lafortune, S. Reveliotis, S. Mahlke, Deadlock-avoidance control of multithreaded software: an efficient siphon-based algorithm for Gadara Petri nets, in: Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, USA, December 12–15, 2011b, pp. 1142–1148.
[119] H.W. Liao, S. Lafortune, S. Reveliotis, Y. Wang, S. Mahlke, Optimal liveness-enforcing control for a class of Petri nets arising in multithreaded software, IEEE Trans. Autom. Control 58 (5) (2013a) 1123–1138.
[120] H.W. Liao, Y. Wang, J. Stanley, S. Lafortune, S. Reveliotis, T. Kelly, S. Mahlke, Eliminating concurrency bugs in multithreaded software: a new approach based on discrete-event control, IEEE Trans. Control Syst. Technol. 21 (6) (2013b) 2067–2082.
[121] H.W. Liao, Y. Wang, H.K. Cho, J. Stanley, T. Kelly, S. Lafortune, S. Mahlke, S. Reveliotis, Concurrency bugs in multithreaded software: modeling and analysis using Petri nets, Discret. Event Dyn. Syst. 23 (2) (2013c) 157–195.
[122] D. Liu, Z.W. Li, M.C. Zhou, Liveness of an extended S^3PR, Automatica 46 (2010) 1008–1018.
[123] D. Liu, Z.W. Li, M.C. Zhou, Hybrid liveness-enforcing policy for generalized Petri net models of flexible manufacturing systems, IEEE Trans. Syst. Man, and Cybern. Syst. 43 (2013a) 85–97.
[124] D. Liu, Z.W. Li, M.C. Zhou, A parameterized liveness and ratio-enforcing supervisor for a class of generalized Petri nets, Automatica 49 (11) (2013b) 3167–3179.
[125] G.Y. Liu, Z.W. Li, C.F. Zhong, New controllability condition for siphons in a class of generalized Petri nets, IET Control Theory & Appl. 4 (5) (2010) 854–864.
[126] G.Y. Liu, Z.W. Li, General mixed integer programming-based liveness test for system of sequential systems with shared resources nets, IET Control Theory & Appl. 4 (12) (2010) 2867–2878.
[127] G.Y. Liu, Z.W. Li, K. Barkaoui, A.M. Al-Ahmari, Robustness of deadlock control for a class of Petri nets with unreliable resources, Inf. Sci. 235 (2013a) 259–279.
[128] G.Y. Liu, D.Y. Chao, F. Yu, Control policy for a subclass of Petri nets without reachability analysis, IET Control Theory & Appl. 7 (8) (2013b) 1131–1141.
[129] G.Y. Liu, Z.W. Li, A.M. Al-Ahmari, Liveness analysis of Petri nets using siphons and mathematical programming, in: Proceedings of the 12th International Workshop on Discrete Event Systems, 12, 2014, pp. 383–387.
[130] G.Y. Liu, K. Barkaoui, Necessary and sufficient liveness condition of GS^3PR Petri nets, Int. J. Syst. Sci. 46 (7) (2015) 1147–1160.
[131] G.J. Liu, C.J. Jiang, M.C. Zhou, Improved sufficient condition for the controllability of dependent siphons in system of simple sequential processes with resources, IET Control Theory & Applications 5 (9) (2011) 1059–1068.
[132] H.X. Liu, K.Y. Xing, W.M. Wu, M.C. Zhou, H.L. Zou, Deadlock prevention for flexible manufacturing systems via controllable siphon sasis of Petri nets, IEEE Trans. Syst. Man, and Cybern. Syst. 45 (3) (2015) 519–529.
[133] M. Liu, S.G. Wang, Z.W. Li, Supervisor reconfiguration for deadlock prevention by resources reallocation, J. Appl. Math. (2013) 315894, doi:10.1155/2013/315894.
[134] M. Liu, S.G. Wang, Z.W. Li, Sufficient conditions for loop resource subsets to derive strict minimal siphons in class of Petri nets, Electron. Lett. 50 (1) (2014) 25–26.
[135] J.P. López-Grao, J.M. Colom, On the deadlock analysis of multithreaded control software, in: Proceedings of the 16th IEEE Conference on Emerging Technologies & Factory Automation, September 5–9, 2011, pp. 1–8.
[136] J.P. López-Grao, J.M. Colom, Structural methods for the control of discrete event dynamic systems C the case of the resource allocation problem, in: C. Seatzu, et al. (Eds.), Control of Discrete-Event Systems, LNCIS, 433, 2013, pp. 257–278. (Chapter 13).
[137] Z.Y. Ma, Z.W. Li, A. Giua, Design of optimal Petri net controllers for disjunctive generalized mutual exclusion constraints, IEEE Transactions on Automatic Control 60 (7) (2015) 1774–1785.
[138] G. Maione, F. DiCesare, Hybrid Petri net and digraph approach for deadlock prevention in automated manufacturing systems, Int. J. Prod. Res. 43 (24) (2005) 5131–5159.
[139] T. Murata, Petri nets: properties, analysis, and applications, Proc. IEEE 77 (4) (1989) 541–580.
[140] A. Nazeem, S.A. Reveliotis, Y. Wang, S. Lafortune, Designing compact and maximally permissive deadlock avoidance policies for complex resource allocation systems through classification theory: The linear case, IEEE Trans. Autom. Control 56 (8) (2011) 1818–1833.
[141] A. Nazeem, S.A. Reveliotis, Designing compact and maximally permissive deadlock avoidance policies for complex resource allocation systems through classification theory: the nonlinear case, IEEE Trans. Autom. Control 57 (7) (2012) 1670–1684.
[142] F. Ning, X.X. Li, S.G. Wang, Q.L. Zhuang, A method based on depth-first search to compute all minimal siphons, J. Softw. 9 (8) (2014) 2148–2153.
[143] Y.L. Pan, Y.S. Huang, M.D. Jeng, S.L. Chung, Enhancement of an efficient control policy for FMSs using the theory of regions and selective siphon method, Int. J. Adv. Manuf. Technol. 66 (9-12) (2013a) 1805–1815.

[144] Y.L. Pan, Y.S. Huang, Y.S. Weng, W.M. Wu, M.D. Jeng, Computationally improved optimal control methodology for linear programming problems of flexible manufacturing systems, J. Appl. Math. 2013 (2013b) 294835.
[145] J. Park, S.A. Reveliotis, Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings, IEEE Trans. Autom. Control 46 (10) (2001) 1572–1583.
[146] C.A. Petri, Kommunikation mit Automaten, Schriften des IIM Nr.2, Bonn, 1962.
[147] J.L. Peterson, Petri nets, Comput. Surv. 9 (3) (1977) 223–252.
[148] L. Piroddi, R. Cordone, I. Fumagalli, Selective siphon control for deadlock prevention in Petri nets, IEEE Trans. Syst. Man, and Cybern. Part A 38 (6) (2008) 1337–1348.
[149] L. Piroddi, R. Cordone, I. Fumagalli, Combined siphon and marking generation for deadlock prevention in Petri nets, IEEE Trans. Syst. Man, and Cybern. Part A 39 (3) (2009) 650–661.
[150] M. Qin, Z.W. Li, M. Khalgui, O. Mosbahi, On applicability of deadlock prevention policies with uncontrollable and unobservable transitions, Int. J. Innov. Comput. Inf. Control 7 (7B) (2011) 4115–4127.
[151] M. Qin, Z.W. Li, M.C. Zhou, M. Khalgui, O. Mosbahi, Deadlock prevention for a class of Petri nets with uncontrollable and unobservable transitions, IEEE Trans. Syst. Man, and Cybern. Part A 42 (3) (2012) 727–738.
[152] M. Qin, Z.W. Li, A.M. Al-Ahmari, Elementary-siphon-based control policy for flexible manufacturing systems with partial observability and controllability of transitions, Asian J. Control 17 (1) (2015) 327–342.
[153] W. Reisig, Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies, Springer-Verlag, Berlin Heidelberg, 2013.
[154] K. Schmidt, Verification of siphons and traps for algebraic Petri nets, Lecture Notes in Computer Science 1248 (1997) 427–446.
[155] A. Tuguchi, S. Tuoku, T. Watunube, An algorithm GMST for extracting minimal siphon-traps and its application to efficient computation of Petri net invariants, in: Proceedings of the International Symposium on Circuits and Systems, vol. 3, 2003, pp. 172–175.
[156] F. Tricas, J. Martinez, An extension of the liveness theory for concurrent sequential processes competing for shared resources, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 1995, pp. 3035–3040.
[157] F. Tricas, F. Garcia-Vallès, J.M. Colom, J. Ezpeleta, A structural approach to the problem of deadlock prevention in processes with shared resources, in: Proceedings of the 4th International Workshop on Discrete Event Systems, 1998, pp. 273–278.
[158] F. Tricas, F. Garcia-Vallès, J.M. Colom, J. Ezpeleta, An iterative method for deadlock prevention in FMSs, in: Proceedings of the 5th Workshop Discrete Event System, 2000, pp. 139–148.
[159] F. Tricas, J. Ezpeleta, Some results on siphon computation for deadlock prevention in resource allocation systems modeled with Petri nets, in: Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, Lisbon, Portugal, 2003, pp. 322–329.
[160] F. Tricas, F. Garcia-Vallès, J.M. Colom, J. Ezpeleta, Using linear programming and the Petri net structure for deadlock prevention in sequential resource allocation systems, XIII Jornadas de Concurrencia y Sistemas Distribuidos (2005) 65–77.
[161] F. Tricas, J. Ezpeleta, Computing minimal siphons in Petri net models of resource allocation systems: a parallel solution, IEEE Trans. Syst. Man, and Cybern. Part A 36 (3) (2006) 532–539.
[162] M. Uzam, An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions, Int. J. Adv. Manuf. Technol. 19 (3) (2002) 192–208.
[163] M. Uzam, M.C. Zhou, An improved iterative synthesis approach for liveness enforcing supervisors of flexible manufacturing systems, Int. J. Prod. Res. 44 (10) (2006) 1987–2030.
[164] M. Uzam, Z.W. Li, M.C. Zhou, Identification and elimination of redundant control places in Petri net based liveness enforcing supervisors of FMS, Int. J. Adv. Manuf. Technol. 35 (2007) 150–168.
[165] M. Uzam, Z.W. Li, On an iterative deadlock prevention approach for automated manufacturing systems, Int. J. Adv. Manuf. Technol. 74 (1-4) (2014) 503–507.
[166] A.R. Wang, Z.W. Li, J.Y. Jia, M.C. Zhou, An effective algorithm to find elementary siphons in a class of Petri nets, IEEE Trans. Syst. Man, and Cybern. Part A 39 (4) (2009) 912–923.
[167] A.R. Wang, Z.W. Li, J.Y. Jia, Efficient computation of strict minimal siphons for a class of Petri nets models of automated manufacturing systems, Trans. Instit. Measurement and Control 33 (1) (2011) 182–201.
[168] A.R. Wang, Z.W. Li, M.C. Zhou, Iterative deadlock control by using Petri nets: a revisit, IEEE Tran. Syst. Man, and Cybern. Part C 42 (6) (2012a) 1204–1218.
[169] S.G. Wang, C.Y. Wang, M.C. Zhou, Z.W. Li, A method to compute strict minimal siphons in a class of Petri nets based on loop resource subsets, IEEE Trans. Syst. Man, Cybern. Part A 42 (1) (2012b) 226–237.
[170] S.G. Wang, W.H. Wu, J. Yang, Deadlock prevention policy for a class of petri nets based on complementary places and elementary siphons, J. Intell. Mauf. 26 (2) (2015) 321–330.
[171] Y. Wang, T. Kelly, M. Kudlur, S. Mahlke, S. Lafortune, The application of supervisory control to deadlock avoidance in concurrent software, in: Proceedings of the 9th International Workshop on Discrete Event Systems, 2008, pp. 287–292.
[172] Y. Wang, H.W. Liao, S. Reveliotis, T. Kelly, S. Mahlke, S. Lafortune, Gadara nets: modeling and analyzing lock allocation for deadlock avoidance in multithreaded software, in: Proceedings of the 48th IEEE Conference on Decision and Control, Shanghai, China, December 16–18, 2009, pp. 4971–4976.
[173] Y. Wang, P. Liu, T. Kelly, S. Lafortune, S. Reveliotis, C. Zhang, On atomicity enforcement in concurrent software via discrete event systems theory, in: Proceedings of the 51st IEEE Conference on Decision and Control, December 10–13, Hawaii, USA, 2012a, pp. 7230–7237.
[174] Y. Wang, J. Stanley, S. Lafortune, Explicit storage and analysis of billions of states using commodity computers, in: Proceedings of the 11th International Workshop on Discrete Event Systems, vol. 11, 2012b, pp. 364–371.
[175] W.L. Winston, M. Venkataramanan, Introduction to Mathematical Programming, Duxbury Resource Center, Belmont CA, 2002.
[176] N. Wei, Z.W. Li, On the suboptimal liveness-enforcing supervisors based on Petri net structural analysis and the theory of regions, Int. J. Adv. Manuf. Technol. 38 (1-2) (2008) 195–204.
[177] N.Q. Wu, Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model, IEEE Trans. Syst. Man, and Cybern. Part C 29 (2) (1999) 192–204.
[178] N.Q. Wu, M.C. Zhou, Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems based on a Petri net model, IEEE Trans. Robot. Autom. 17 (5) (2001) 658–669.
[179] N.Q. Wu, M.C. Zhou, Modeling and deadlock control of automated guided vehicle systems, IEEE/ASME Trans. Mechatron. 9 (1) (2004) 50–57.
[180] N.Q. Wu, M.C. Zhou, Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles, IEEE Trans. Syst. Man, and Cybern. Part B 35 (6) (2005) 1193–1202.
[181] N.Q. Wu, M.C. Zhou, Shortest routing of bi-directional automated guided vehicles avoiding deadlock and blocking, IEEE/ASME Trans. Mechatron. 12 (1) (2007a) 63–72.
[182] N.Q. Wu, M.C. Zhou, Deadlock resolution in automated manufacturing systems with robots, IEEE Trans. Autom. Sci. Eng. 4 (3) (2007b) 474–480.
[183] N.Q. Wu, M.C. Zhou, Deadlock modeling and control of semiconductor track systems using resource-oriented Petri nets, Int. J. Prod. Res. 45 (15) (2007c) 3439–3456.
[184] N.Q. Wu, M.C. Zhou, Z.W. Li, Resource-oriented Petri net for deadlock avoidance in flexible assembly systems, IEEE Trans. Syst. Man, and Cybern. Part A 38 (1) (2008) 56–69.
[185] N.Q. Wu, M.C. Zhou, System Modeling and Control with Resource-Oriented Petri Nets, CRC Press, Taylor & Francis Group, New York, 2009.
[186] N.Q. Wu, M.C. Zhou, Process vs resource-oriented Petri net modeling of automated manufacturing systems, Asian J. Control 12 (3) (2010) 267–280.
[187] N.Q. Wu, M.C. Zhou, Intelligent token Petri nets for modeling and control of reconfigurable automated manufacturing systems with dynamical changes, Trans. Instit. Measurement and Control 33 (1) (2011) 9–29.
[188] N.Q. Wu, M.C. Zhou, G. Hu, One-step look-ahead maximally permissive deadlock control of AMS by using Petri net, ACM Trans. Embedded Comput. Syst. 12 (1) (2013). Article 10 (10:1-10:23).

[189] N.Q. Wu, M.C. Zhou, Z.W. Li, Short-term scheduling of crude-oil operations: Petri net-based control-theoretic approach, IEEE Robotics and Automation Magazine 22 (2) (2015) 64–76.
[190] X.L. Xie, M.D. Jeng, ERCN-merged nets and their analysis using siphons, IEEE Trans. Robot. Autom. 15 (4) (1999) 692–703.
[191] K.Y. Xing, M.C. Zhou, H.X. Liu, F. Tian, Optimal Petri net-based polynomial-complexity deadlock avoidance policies for automated manufacturing systems, IEEE Trans. Syst. Man, and Cybern. Part A 39 (1) (2009) 188–199.
[192] K.Y. Xing, M.C. Zhou, F. Wang, H.X. Liu, F. Tian, Resource-transition circuits and siphons for deadlock control of automated manufacturing systems, IEEE Transactions on System, Man, Cybernetics, Part A 41 (1) (2011) 74–84.
[193] K. Yamalidou, J. Moody, M. Lemmon, P. Antsaklis, Feedback control of Petri nets based on place invariants, Automatica 32 (1) (1996) 15–28.
[194] M. Yamauchi, T. Watanabe, Algorithms for extracting minimal siphons containing specified places in a general Petri net, IEICE Trans. Fund. 82 (11) (1999) 566–575.
[195] M.M. Yan, Z.W. Li, N. Wei, M. Zhao, A deadlock prevention policy for a class of Petri nets S^3PMR, J. Inf. Sci. Eng. 25 (1) (2009) 167–183.
[196] J.H. Ye, Z.W. Li, A. Giua, Decentralized supervision of Petri nets with a coordinator, IEEE Trans. Syst. Man, and Cybern. Syst. 45 (6) (2015) 955–966.
[197] X.Y. Zhang, Z.W. Li, C.F. Zhong, A.M. Al-Ahmari, Reachability analysis of a class of Petri nets using place invariants and siphons, Maejo Int. J. Sci. Technol. 7 (2) (2013) 278–290.
[198] M. Zhao, Z.W. Li, On deadlock control for a class of generalized Petri net models of flexible manufacturing systems, Int. J. Prod. Res. 47 (21) (2009) 6085–6107.
[199] M. Zhao, Z.W. Li, H.S. Hu, Suboptimal liveness-enforcing supervisor design for a class of generalized Petri nets using partial siphon enumeration and mathematical programming, Int. J. Syst. Sci. 41 (9) (2010) 1013–1026.
[200] C.F. Zhong, Z.W. Li, A deadlock prevention approach for flexible manufacturing systems without complete siphon enumeration of their petri net models, Eng. Comput. 25 (2009) 269–278.
[201] C.F. Zhong, Z.W. Li, Self-liveness of a class of Petri net models for flexible manufacturing systems, IET Control Theory & Appl. 4 (3) (2010a) 403–410.
[202] C.F. Zhong, Z.W. Li, Design of liveness-enforcing supervisors via transforming plant Petri net models of FMS, Asian J. Control 12 (3) (2010b) 240–252.
[203] C.F. Zhong, Z.W. Li, K. Barkaoui, Monitor design for siphon control in S^4R nets: from structure analysis points of view, Int. J. Innov. Comput. Inf. Control 7 (12) (2011) 6677–6690.
[204] C.F. Zhong, Z.W. Li, Y.F. Chen, On nonexistence of a maximally permissive liveness-enforcing pure net supervisor, IEEE Trans. Syst. Man, and Cybern. Syst. 43 (1) (2013) 29–37.
[205] M.C. Zhou, F. DiCeasre, A.A. Desrochers, Proceeding of IEEE Robotics Automation Conference, 1989, pp. 534–539.
[206] M.C. Zhou, F. DiCeasre, Adaptive design of Petri net controllers for error recovery in automated manufacturing systems, IEEE Trans. Syst. Man, and Cybern. 19 (5) (1989) 963–973.
[207] M.C. Zhou, F. DiCeasre, Parallel and sequential mutual exclusions for Petri net modeling of manufacturing systems with shared resources, IEEE Trans. Robot. Autom. 7 (4) (1991) 515–527.
[208] M.C. Zhou, F. DiCesare, A.A. Desrochers, A hybrid methodology for synthesis of Petri nets for manufacturing systems, IEEE Trans. Robot. Autom. 8 (3) (1992a) 350–361.
[209] M.C. Zhou, F. DiCesare, D. Rudolph, Design and implementation of a Petri net based supervisor for a flexible manufacturing system, Automatica 28 (6) (1992b) 1199–1208.
[210] M.C. Zhou, F. DiCesare, Petri Net Synthesis for Discrete Event Control of Manufacturing Systems, Kluwer Academic Publishers, London, 1993.
[211] M.C. Zhou, Deadlock avoidance methods for a distributed robotic system: Petri net modeling and analysis, J. Robot. Syst. 12 (3) (1995) 177–187.
[212] M.C. Zhou, K. Venkatesh, Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach, World Scientific, Singapore, 1998.
[213] R. Zurawski, M.C. Zhou, Petri nets and industrial applications: a tutorial, IEEE Trans. Ind. Electron. 41 (6) (1994) 567–583.