IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 31, 2017, accepted September 25, 2017, date of publication October 17, 2017,
date of current version February 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2763783

An Approach for Enumerating Minimal Siphons in a Subclass of Petri Nets

DAN YOU[1], SHOUGUANG WANG[1], WENZHAN DAI[1], WENHUI WU[1], AND YONGSHU JIA[2]
[1]School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China
[2]Zhejiang Industry and Trade Vocational College, Wenzou 325003, China

Corresponding author: Shouguang Wang (wsg5000@hotmail.com)

This work was supported in part by the Zhejiang Natural Science Foundation under Grant LY15F030003, in part by the National Natural Science Foundation of China under Grant 61472361 and Grant 61374022, and in part by the Zhejiang Science and Technology Project under Grant 2015C31064.

ABSTRACT Siphons, as a structural object of Petri nets (PNs), are closely related to deadlock-freedom in PNs. Efficient siphon computation is of great importance in developing siphon-based deadlock control strategies with good performance. This paper is concerned with the enumeration of minimal siphons in a subclass of PNs called systems of sequential systems with shared resources (S^4PR). First, a method with polynomial complexity is proposed to decide whether a subset of resource places can generate a minimal siphon. Next, by utilizing the technique of problem partitioning, we develop an approach to compute all minimal siphons in S^4PR. The proposed approach is illustrated by an example and its advantage is finally demonstrated via a comparison with other approaches.

INDEX TERMS Petri nets, S^4PR nets, minimal siphon enumeration, problem partitioning.

I. INTRODUCTION

PETRI nets (PNs) are a popular modelling tool of discrete event systems (DESs) [4], [10]–[20], [34], [36], [43], [46], [49], [51] to tackle problems like deadlock problems [3], [13], [16]–[19], [22], [26], [28], [30], [40], [41], [44], fault diagnosis [5], [6], [27] and process scheduling [24], [39], and they enjoy applications in various real-world DES such as flexible manufacturing systems [18], workflow systems [20], microgrid systems [23], railway systems [11], and business systems [47], [48]. A siphon is a structural object of PNs and strongly related to the properties of deadlock-freedom and liveness. The number of siphons in a PN grows exponentially in the worst case with respect to the net size [21]. Consequently, the computational efficiency of siphon-based deadlock control strategies [3], [12], [16], [22], [26], [28], [41], [44] largely depends on that of siphon computation.

The computation of siphons can be classified into two categories, i.e., one applicable to general nets [1], [2], [9], [26], [29], [31], [35] and the other applicable to specific nets only [7], [8], [17], [33], [40]–[42], [45]. In specific nets, siphons possess their particular properties that are conducive to proposing computation approaches with low complexity. Hence, the approaches of siphon computation for specific nets usually have higher efficiency than those for general nets. In this work, we study the enumeration of minimal siphons in a subclass of PNs named Systems of Sequential Systems with Shared Resources (S^4PR) [30].

Places in S^4PR are divided into resource, activity and idle places and minimal siphons are divided into ones with and without resources. The number of minimal siphons without resources is exactly the same as that of idle places in an S^4PR and they are easy to be computed. Thus, the difficulty of minimal siphon computation in S^4PR lies in the computation of minimal siphons with resources. Due to the fact that each resource-place subset in an S^4PR may yield a minimal siphon and at most one minimal siphon [8], the key to minimal siphon computation in S^4PR is to answer the question: How to decide whether a resource-place subset can yield a minimal siphon.

For a subclass of S^4PR named "Systems of Simple Sequential Processes with Resources (S^3PR)", our previous work [33] proposes a sufficient and necessary condition to decide if a subset of resource places corresponds to a minimal siphon. Based on this condition, an approach with high computational efficiency is developed [33] to compute all the minimal siphons in S^3PR. Barkaoui and Lemaire [2] propose a characterization of minimal siphons for general nets using graph theory. However, it is not efficient when applied to S^4PR. For another subclass of S^4PR named "Extended Systems of Linear Simple Sequential Process with

Resources", a necessary but not sufficient condition for a subset of resource places to generate a strict minimal siphon is established by Wang *et al.* [32] based on resource digraphs. Cano *et al.* [8] present the concept of pruning graphs and then propose a determination condition for a subset of resource places to generate a minimal siphon in S^4PR. We observe that such a determination still turns to the definition of minimal siphons. Motivated by their work, we construct characteristic implicit resource-transition (CIRT) nets in our previous work [37] and then propose a sufficient and necessary condition to decide if a resource-place subset can yield a minimal siphon. However, how to enumerate all minimal siphons in S^4PR is not presented in [37]. In this work, it is answered. Firstly, we develop a new method that is shown to be polynomial complexity to decide whether a resource-place subset can yield a minimal siphon in S^4PR. Next, based on the determination method and adopting the technique of problem partitioning, we propose an approach that enumerates all minimal siphons in S^4PR.

The remainder of this paper is organized as follows. Section II recalls necessary concepts and results in [37] and develops a new method to decide whether a resource-place subset can yield a minimal siphon in S^4PR. The new approach to enumerate minimal siphons in S^4PR is presented in Section III and Section IV shows the comparison between the proposed approach and an existing one via an example. Section V concludes this paper.

II. CONDITION FOR RESOURCE-PLACE SUBSET TO GENERATE MINIMAL SIPHON

The basic concepts and notations related to PNs, siphons and S^4PR are reviewed in detail in *Section II Preliminaries* of our prior work [37], and we thus do not repeat them in this paper. More knowledge of PNs can be found in [25], [38], and [50]. According to the definition of S^4PR, any transition in S^4PR has one output activity place at most and one input activity place at most. We thus use t^a and ^at to denote the unique output and input activity places of a transition t, respectively.

The net N in Fig. 1 is an S^4PR. By the definition of S^4PR, the sets of idle places, resource places and activity places are $P_0 = \{p_1, p_6\}$, $P_R = \{r_1, r_2, r_3\}$ and $P_A = \{p_2 - p_5, p_7 - p_{10}\}$, respectively. Moreover, the sets of holders of r_1, r_2, r_3 are $H(r_1) = \{p_5, p_9\}$, $H(r_2) = \{p_2 - p_4, p_7, p_{10}\}$ and $H(r_3) = \{p_8 - p_{10}\}$, respectively. Besides, consider the transition t_5. We have $^at_5 = p_3$ and $t_5^a = p_5$.

According to the work [8], minimal siphons in an S^4PR are divided into two types: the ones with and without resources. The latter can be easily computed. Indeed, each minimal siphon in an S^4PR without resources consists of all places in a subnet N_i [37, Definition 1.3], i.e., $P_{Ai} \cup \{p_0^i\}$. In this section, a method is proposed to decide whether a subset of resource places can yield a minimal siphon in an S^4PR. We recall the following concepts and results from the work [37] before proposing the method.

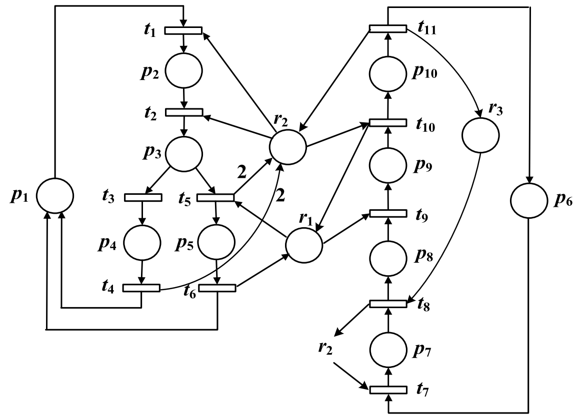


FIGURE 1. An S^4PRN.

Definition 1 [37]: Given an elementary path $\pi = x_1x_2\ldots x_n$ and a resource-place subset Ω in an S^4PR, π is said to be a *pure activity path* with respect to Ω if

1) $\forall p \in ||\pi|| \cap P, p \in P_A$; and
2) $\forall t \in (||\pi||\backslash\{x_1, x_n\}) \cap T, (t^\bullet \cup {}^\bullet t) \cap \Omega = \emptyset$,

where $||\pi||$ represents the set of all nodes in π.

Consider the resource-place subset $\Omega = \{r_1, r_2, r_3\}$ in Fig. 1. We can see that the elementary path $\pi_1 = t_2p_3t_3p_4t_4$ is a pure activity path with respect to Ω, while the elementary path $\pi_2 = t_1p_2t_2p_3t_3p_4t_4$ is not.

Definition 2 [37]: Given a resource-place subset Ω, a transition $t \in T$ and an activity place $p \in P_A$ in an S^4PR, p is said to be a *restoring place* of t with respect to Ω if t is accessible from p via a pure activity path with respect to Ω. $P^+(t, \Omega)$ denotes the set of all *restoring places* of t with respect to Ω.

Consider the resource-place subset $\Omega = \{r_1, r_2, r_3\}$ in Fig. 1 and the transition t_4. We can see t_4 is accessible from the place p_4 via the pure activity path $\pi = p_4t_4$ and from the place p_3 via the pure activity path $\pi' = p_3t_3p_4t_4$. Thus, both p_4 and p_3 are restoring places of t_4 with respect to Ω and we have $P^+(t_4, \Omega) = \{p_3, p_4\}$.

Definition 3 [37]: Given a resource-place subset Ω of an S^4PR, S_Ω is defined as a place set such that

$S_\Omega = \Omega \cup (\cup_{t \in {}^\bullet\Omega\backslash\Omega^\bullet} P^+(t, \Omega))$.

Theorem 1 [37]: Given a minimal siphon S with $S \cap P_R = \Omega \neq \emptyset$ in an S^4PR, $S = S_\Omega = \Omega \cup (\cup_{t \in {}^\bullet\Omega\backslash\Omega^\bullet} P^+(t, \Omega))$.

Theorem 1 indicates that each minimal siphon containing resources in an S^4PR is in the form of $S_\Omega = \Omega \cup (\cup_{t \in {}^\bullet\Omega\backslash\Omega^\bullet} P^+(t, \Omega))$. However, given a resource-place subset Ω of an S^4PR, S_Ω is not necessarily a minimal siphon.

Property 1 [37]: S_Ω is a minimal siphon if $|\Omega| = 1$.

Consider the S^4PR in Fig. 1 again. Due to Definition 3, $S_{\{r1\}} = \{r_1\} \cup (\cup_{t \in \{t_6, t_{10}\}} P^+(t, \{r_1\})) = \{r_1, p_5, p_9\}$, $S_{\{r2\}} = \{r_2\} \cup (\cup_{t \in \{t_4, t_5, t_8, t_{11}\}} P^+(t, \{r_2\})) = \{r_2, p_3, p_4, p_7, p_{10}\}$, and $S_{\{r3\}} = \{r_3\} \cup P^+(t_{11}, \{r_3\}) = \{r_3, p_8, p_9, p_{10}\}$. We can see$S_{\{r1\}}$, $S_{\{r2\}}$ and $S_{\{r3\}}$ are all minimal siphons by Property 1.

In the following, we show how to determine whether S_Ω is a minimal siphon in the case that $|\Omega| \geq 2$. To achieve this aim, some concepts are introduced first.

Definition 4 [37]: Given a resource-place subset Ω of an S^4PR $N = (P_0 \cup P_A \cup P_R, T, F)$, $N_\Omega = (P_\Omega, T_\Omega, F_\Omega)$ is said to be an Ω-induced *implicit resource-transition (IRT) net* if

1) $P_\Omega = \Omega$;
2) $T_\Omega = \Omega^\bullet \cap H(\Omega)^\bullet$; and
3) $F_\Omega = F_{\Omega in} \cup F_{\Omega out}$, where $F_{\Omega in} = (P_\Omega \times T_\Omega) \cap F$ and $F_{\Omega out} = \{(t, r) \in T_\Omega \times P_\Omega | {}^a t \in H(r)\}$.

Definition 5 [37]: Given a resource-place subset Ω of an S^4PR N and the Ω-induced IRT net $N_\Omega = (P_\Omega, T_\Omega, F_{\Omega in} \cup F_{\Omega out})$, an *Arc labelling function* is defined as Γ: $F_{\Omega out} \rightarrow 2^{P_A}$ such that $\forall (t, r) \in F_{\Omega out}$, $\Gamma((t, r)) = P^+(t, \{r\})$.

Consider the resource-place subset $\Omega = \{r_1, r_2, r_3\}$ of the S^4PR in Fig. 1. According to Definition 4, $P_\Omega = \{r_1, r_2, r_3\}$. Moreover, considering that $t_5 \in r_1^\bullet \cap H(r_2)^\bullet$, we have $(r_1, t_5) \in F_{\Omega in}$ and $(t_5, r_2) \in F_{\Omega out}$. By Definition 5, $\Gamma((t_5, r_2)) = P^+(t_5, \{r_2\}) = \{p_3\}$. Considering that $t_9 \in r_1^\bullet \cap H(r_3)^\bullet$, we have $(r_1, t_9) \in F_{\Omega in}, (t_9, r_3) \in F_{\Omega out}$, and $\Gamma((t_9, r_3)) = P^+(t_9, \{r_3\}) = \{p_8\}$. After other arcs are determined in the similar way, the Ω-induced IRT net N_Ω is obtained with arcs being labelled, as shown in Fig. 2(a).

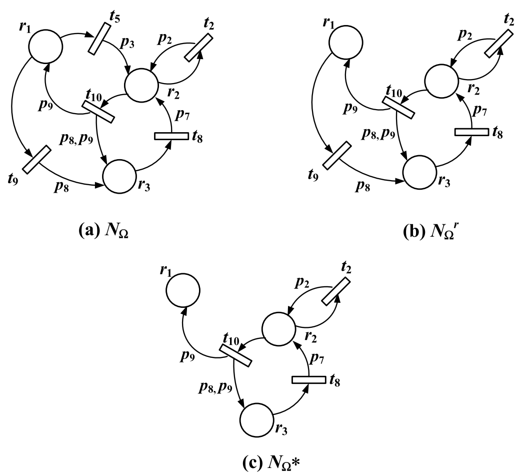


FIGURE 2. **(a) The IRT net (b) the RIRT net and (c) the CIRT net of *N* in Fig. 1 induced by $\Omega = \{r_1, r_2, r_3\}$.**

Definition 6 [37]: Given a resource-place subset Ω of an S^4PR N, we call $t \in T_\Omega$ an *α-transition* related to Ω if $\exists t' \in {}^\bullet\Omega \backslash \Omega^\bullet$ such that ${}^a t \in P^+(t', \Omega)$ in N. $T_\alpha(\Omega)$ denotes the set of all α-transitions related to Ω.

For example, t_5 is an α-transition related to the resource subnet $\Omega = \{r_1, r_2, r_3\}$ of the S^4PR in Fig. 1. This is because we can find $t_4 \in {}^\bullet\Omega \backslash \Omega^\bullet$ such that ${}^a t_5 \in P^+(t_4, \Omega)$. Furthermore, we have $T_\alpha(\Omega) = \{t_5\}$.

Definition 7 [37]: Given a resource-place subset Ω of an S^4PR N, $N^r_\Omega = (P^r_\Omega, T^r_\Omega, F^r_\Omega)$ is said to be an Ω-induced *reduced implicit resource-transition (RIRT) net* if

1) $P^r_\Omega = \Omega$;
2) $T^r_\Omega = T_\Omega \backslash T_\alpha(\Omega)$; and
3) $F^r_\Omega = F_\Omega \cap ((P^r_\Omega \times T^r_\Omega) \cup (T^r_\Omega \times P^r_\Omega))$.

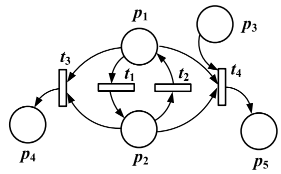


FIGURE 3. **A PN.**

According to the above definition, an Ω-induced RIRT net is derived from an IRT net via deleting all α-transitions and their related arcs.

The RIRT net N^r_Ω of the S^4PR in Fig. 1 induced by the resource-place subset $\Omega = \{r_1, r_2, r_3\}$ is presented in Fig. 2(b). It is derived from the IRT net in Fig. 2(a) by deleting α-transition t_5 and its related arcs.

Definition 8 [37]: Let Ω be a resource-place subset of an S^4PR N and $N^r_\Omega = (P^r_\Omega, T^r_\Omega, F^r_\Omega)$ be the Ω-induced RIRT net. An arc $(t, r) \in F^r_\Omega$ is said to be a *β-arc* related to Ω if $\exists (t', r) \in F^r_\Omega$, such that $\Gamma((t, r)) \subset \Gamma((t', r))$. $F_\beta(\Omega)$ denotes the set of all β-arcs related to Ω. A transition $t \in T^r_\Omega$ is said to be a *β-transition* related to Ω if $\forall (t, r) \in F^r_\Omega, (t, r) \in F_\beta(\Omega)$. $T_\beta(\Omega)$ denotes the set of all β-transitions related to Ω.

Observe the RIRT net N^r_Ω in Fig. 2(b). We can see the arc (t_9, r_3) is a β-arc related to $\Omega = \{r_1, r_2, r_3\}$ since there is the arc (t_{10}, r_3) such that $\Gamma((t_9, r_3)) \subset \Gamma((t_{10}, r_3))$. Clearly, t_9 is a β-transition related to Ω. Furthermore, we have $F_\beta(\Omega) = \{(t_9, r_3)\}$ and $T_\beta(\Omega) = \{t_9\}$.

Definition 9 [37]: Given a resource-place subset Ω of an S^4PR N, $N^*_\Omega = (P^*_\Omega, T^*_\Omega, F^*_\Omega)$ is said to be an Ω-induced *characteristic implicit resource-transition (CIRT) net* if

1) $P^*_\Omega = \Omega$;
2) $T^*_\Omega = T^r_\Omega \backslash T_\beta(\Omega)$; and
3) $F^*_\Omega = (F^r_\Omega \cap ((P^*_\Omega \times T^*_\Omega) \cup (T^*_\Omega \times P^*_\Omega))) \backslash F_\beta(\Omega)$.

It can be seen that we can derive an Ω-induced CIRT net from an IRT net via removing all α-transitions as well as their related arcs, then β-transitions as well as their related arcs, and finally β-arcs.

We can see that, by deleting the β-transition t_9 as well as its related arcs from the RIRT net in Fig. 2(b), the CIRT net $N_\Omega{}^*$ of the S^4PR in Fig. 1 induced by the resource-place subset $\Omega = \{r_1, r_2, r_3\}$ is obtained, as shown in Fig. 2(c).

Property 2 [37]: Given an S^4PR N, a resource-place subset Ω such that $|\Omega| \geq 2$ and the Ω-induced CIRT net N^*_Ω, S_Ω is not a minimal siphon if N^*_Ω is not strongly connected.

Consider the resource-place subset $\Omega = \{r_1, r_2, r_3\}$ of the S^4PR in Fig. 1. S_Ω is not a minimal siphon since the CIRT net N^*_Ω shown in Fig. 2(c) is not strongly connected.

Definition 10: Let $N = (P, T, F, W)$ be a PN and $P' \subseteq P$. A transition $t \in P'^\bullet \cap^\bullet (P \backslash P')$ such that ${}^\bullet t \subseteq P'$ is called a *particular output transition* of P'.

Consider the PN in Fig. 3 and a place set $P' = \{p_1, p_2\}$. We can see that $t_3, t_4 \in P'^\bullet \cap^\bullet (P \backslash P')$. By Definition 10, t_3 is a particular output transition of P', while t_4 is not.

Now, we develop a function next, through which it can be determined whether a resource-place subset Ω of an S^4PR such that $|\Omega| \geq 2$ can generate a minimal siphon.

Function Flag = $Check(N^*_\Omega)$

Input: An Ω-induced CIRT net $N^*_\Omega = (\Omega, T^*_\Omega, F^*_\Omega)$;
Output: Flag. /*Flag=True implies S_Ω is a minimal siphon and not otherwise. */

1) Flag:=True;
2) Select a resource r in N^*_Ω and let C:=$\{r\}$;
3) Create an empty stack Δ; /*Δ is used to store sets of resources.*/
4) *PushStack*(Δ, C);
5) **While** $\exists t \in C^\bullet \cap^\bullet (\Omega \backslash C)$ such that $^\bullet t \subseteq C$ **do** /*C has its particular output transitions.*/
6) **if** $\exists r' \in t^\bullet \backslash C$ such that r' is not in any resource set in Δ **then**
7) *PushStack*(Δ, $\{r'\}$); /**PushStack*(Δ, $\{r'\}$) pushes $\{r'\}$ onto the top of stack Δ*/
8) C:=$\{r'\}$;
9) **else**
10) Let r' be a resource in a resource set in Δ such that $r' \in t^\bullet \backslash C$;
11) X:=*PopStack*(Δ, C'), where C' is the set in Δ that r' belongs to. /*Function *PopStack* pops resource sets from C' to the one at the top of Δ out of Δ and X stores the popped resources sets. */
12) C'':= $\bigcup_{C \in X} X$;
13) *PushStack*(Δ, C'');
14) C:=C'';
15) **end if**
16) **end while**
17) **if** $C \neq \Omega$ **then**
18) Flag:=False;
19) **end if**
20) **Output:** Flag.

Theorem 2: Given an S^4PR N, a resource-place subset Ω such that $|\Omega| \geq 2$ and the Ω-induced CIRT net N^*_Ω, $S_\Omega = \Omega \cup (\cup_{t \in {}^\bullet\Omega \backslash \Omega^\bullet} P^+(t, \Omega))$ is a minimal siphon *iff* $Check(N^*_\Omega)$ =True.

Proof: (=>) By contradiction, suppose that S_Ω is not minimal. There exists a siphon $S^* \subset S_\Omega$ such that $S^*_R \subset \Omega$, where $S^*_R = S^* \cap P_R$. Let $r \in \Omega \backslash S^*_R$. Since $r \in \Omega$ and $Check(N^*_\Omega)$ =True, r has its particular output transition t and $r' \in t^\bullet$ in N^*_Ω. Due to [37, Property 3], $r' \notin S^*_R$ since otherwise $S^* \not\subset S_\Omega$. Similarly, $\exists r'' \in r'^{\bullet\bullet}$ in N^*_Ω such that $r'' \notin S^*_R$. Since $Check(N^*_\Omega)$ =True, all resources in Ω can be searched via a particular output transition of a resource or a resource set. It implies $\forall r \in \Omega, r \notin S^*_R$. Hence, no siphon in S_Ω can be found with resource set being a proper subset of Ω. Thus, S_Ω is a minimal siphon.

(<=) By contradiction, suppose that $Check(N^*_\Omega)$ =False. We thus have three cases. 1) N^*_Ω is not strongly connected; 2) a resource r in N^*_Ω has no particular output transitions;

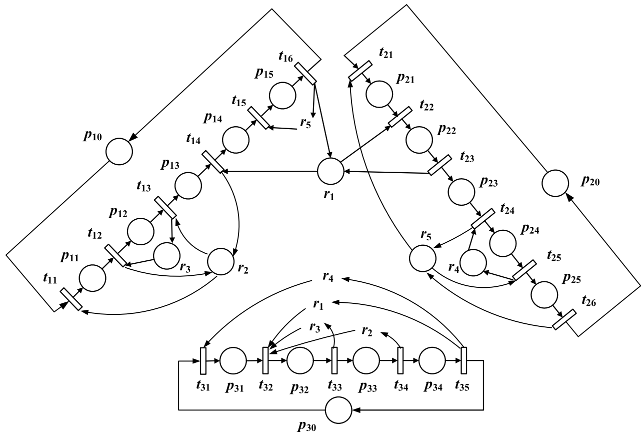


FIGURE 4. An S^4PR N.

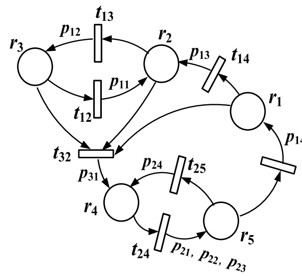


FIGURE 5. The CIRT net N^*_Ω w.r.t. the S^4PR in Fig. 4 with $\Omega = \{r_1 - r_5\}$.

3) there exists a strongly connected subnet of N^*_Ω whose resource set C has no particular output transitions. For Case 1, S_Ω is not a minimal siphon by Property 2. For Cases 2 and 3, there exists a siphon in S_Ω without r or C, i.e., S_Ω is not a minimal siphon. As a result, $Check(N^*_\Omega)$ =True.

We illustrate Function *Check* by the following example.

Consider an S^4PR N in Fig. 4 with $P_0 = \{p_{10}, p_{20}, p_{30}\}$, $P_R = \{r_1 - r_5\}$ and $P_A = \{p_{11} - p_{15}, p_{21} - p_{25}, p_{31} - p_{34}\}$. Consider the resource-place subset $\Omega = \{r_1 - r_5\}$ of N. The Ω-induced CIRT net N^*_Ω is obtained, as shown in Fig. 5. The execution of $Check(N^*_\Omega)$ is as follows: First, r_1 is selected and $C = \{r_1\}$ is pushed onto Δ. We can see that C has its particular output transition t_{14} and $r_2 \in t^\bullet_{14} \backslash C$ is not in Δ. Hence, C is updated as $C = \{r_2\}$ and it is also pushed onto Δ. Similarly, following $\{r_2\}'$s particular output transition t_{13}, r_3 is found and we push $\{r_3\}$ onto Δ. Following $\{r_3\}$'s particular output transition t_{12}, r_2 is found. Note that r_2 is in Δ. According to Steps 11-13, we pop $\{r_2\}$ and $\{r_3\}$ out of Δ and then push $\{r_2, r_3\}$ onto Δ. C is updated as $C = \{r_2, r_3\}$. We can see that $\{r_2, r_3\}$ has no particular output transitions and it is not equal to Ω. Hence, Flag=False is outputted, implying S_Ω is not a minimal siphon, i.e., Ω cannot generate a minimal siphon.

Let us observe Function *Check*. When $Check(N^*_\Omega)$ is performed, we can see that the loop from Steps 5 to 16 is executed at most $2(a - 1)$ times, where a is the number of places in N^*_Ω. Trivially, the computational complexity of

Function *Check* is not higher than $O(a)$. Furthermore, we can conclude that Function *Check* is of polynomial complexity with respect to the size of inputted CIRT net.

III. ENUMERATION OF MINIMAL SIPHONS IN S^{4}PR

In this section, the enumeration of all minimal siphons in S^{4}PR is studied. Π_0 and Π_1 are used to represent the sets of all minimal siphons containing no resources and only one resource in an S^{4}PR, respectively. Clearly, it is easy to compute Π_0 and Π_1. Given an S^{4}PR$N = (P_0 \cup P_A \cup P_R, T, F, W)$, we have $\Pi_0 = \bigcup_{i\in\{1,2,\ldots,|P_0|\}} \{\{p_0^i\} \cup P_{Ai}\}$ and $\Pi_1 = \bigcup_{r\in P_R}\{S_{\{r\}}\}$ due to Property 1. As for computing minimal siphons with more than one resource, we need to find out resource-place subsets that can yield minimal siphons. Due to Property 2, we intend to search all strongly connected CIRT nets. In more detail, we hope to search them in the P_R-induced IRT net from large to small size by gradually deleting transitions and places. However, we notice that given two IRT nets $N_{\Omega 1}$ and $N_{\Omega 2}$ such that $\Omega_1 \subset \Omega_2$, it can happen that an arc in $N_{\Omega 2}$ is a β-arc but it becomes a non-β-arc in $N_{\Omega 1}$. Considering this fact, we find out all strongly connected RIRT nets from the P_R-induced IRT net as "candidates" instead of strongly connected CIRT nets. **Note that IRT nets, RIRT nets and CIRT nets in this section all refer to those with more than one resource by default.**

A. COMPUTATION OF STRONGLY CONNECTED RIRT COMPONENTS

Definition 11: Let N' be an Ω-induced IRT net and $\Phi = \{N^r_{\Omega 1}, N^r_{\Omega 2}, \ldots, N^r_{\Omega k}\}$ be the set of all strongly connected RIRT nets in N'. $N^r_{\Omega i}$ is said to be a *strongly connected RIRT component* of N' if $\nexists N^r_{\Omega j} \in \Phi$ such that $\Omega_j \supseteq \Omega_i$, where $i, j \in \{1, 2, \ldots, k\}$ and $i \neq j$.

Suppose that N' is an Ω-induced IRT net with the set of all strongly connected RIRT nets in N' being $\Phi = \{N^r_{\Omega 1}, N^r_{\Omega 2}, N^r_{\Omega 3}, N^r_{\Omega 4}\}$, where $\Omega_1 = \{r_1, r_2\}$, $\Omega_2 = \{r_2, r_3\}$, $\Omega_3 = \{r_4, r_5\}$ and $\Omega_4 = \{r_1, r_2, r_3\}$. According to Definition 11, we can see $N^r_{\Omega 3}$ and $N^r_{\Omega 4}$ are strongly connected RIRT components of N'.

An IRT net may contain more than one strongly connected RIRT component. In what follows, we present a way to compute all strongly connected RIRT components including a resource set R_{in} in a given IRT net using Function *FindSCRC*, where Function *Do* is called.

Function Φ = *FindSCRC*(*N'*, R_{in})

Input: An IRT net N' and a resource set R_{in}.
Output: The set of all strongly connected RIRT components including R_{in} in N', denoted by Φ.

1) $\Phi := \emptyset$; /*Φ is a global variable that can be updated in Function *Do*. */
2) *Do* (*N'*, R_{in});
3) **Output:** Φ;

Function *Do* (*N'*, R_{in})

Input: A netN' and a resource set R_{in}.

1) Ψ:=*Tarjan* (N'); /* Function *Tarjan* here returns the set of all strongly connected components with more than one place of a net. */
2) **for** $N'' = (P'', T'', F'') \in \Psi$ such that $P'' \supseteq R_{in}$**do**
3) **if** N'' contains no α-transitions **then** /*N'' is a strongly connected RIRT component. */
4) $\Phi : \Phi \cup \{N''\}$;
5) **else**
6) N'' := *DeleteAlpha*(N''); /* Function *DeleteAlpha* returns a net by deleting α-transitions as well as their related arcs from a net. */
7) *Do* (*N"*, R_{in});
8) **end if**
9) **end for**

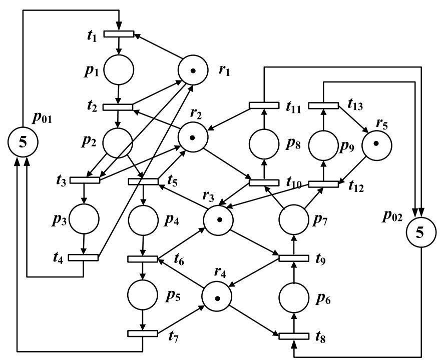


FIGURE 6. An S^{4}PR N.

When α-transitions are deleted from a net as shown in Step 6 of Function *Do*, the obtained net may be not strongly connected. Besides, after Function *Tarjan* is applied to the net, resulting in nets with smaller sizes, it is possible that α-transitions emerge again in these obtained nets. Hence, Function *Do* has to be recursively called when computing strongly connected RIRT components. It is trivial to derive the following proposition.

Proposition 1: Let N' be an Ω-induced IRT net and R_{in} be a set of resources.Φ = *FindSCRC*(*N'*, R_{in}) is the set of all strongly connected RIRT components including R_{in} in N'.

It can be seen that Φ*FindSCRC*(*N'*, R_{in}) is the set of all strongly connected RIRT components of N' in the case that $R_{in} = \emptyset$ and it holds that $|\Phi \leq 1$ when $R_{in} \neq \emptyset$.

Consider the S^{4}PR N in Fig. 6. Its IRT net N_Ω induced by $\Omega = \{r_1 - r_5\}$ is shown in Fig. 7(a). We compute the set of all strongly connected RIRT components in N_Ω by calling *FindSCRC*(N_Ω, $\emptyset$). It executes as follows: First, *Tarjan* (N_Ω) is called, outputting a strongly connected component, i.e., the net N_1 in Fig. 7(b). Due to the emergence of α-transition t_{10}, *DeleteAlpha*(N_1) is then called, outputting the net N_2 in Fig. 7(c). Next, Function *Tarjan* is called again to deal with N_2, resulting in two strongly connected components, i.e., N_3 and N_4 in Fig. 7(d) and (e). We can see that α-transition t_3 emerges in N_3. Thus, *DeleteAlpha*(N_3)

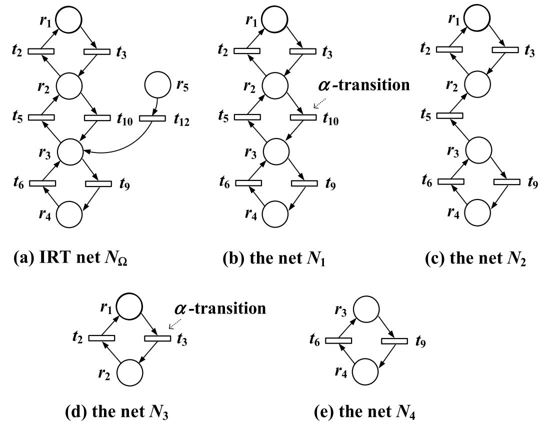


FIGURE 7. **(a) The IRT net induced by $\Omega = \{r_1 - r_5\}$ of N in Fig. 6; and (b)-(e) Four nets generated during the execution of *FindSCRC*$(N_\Omega, \emptyset)$.**

is called and the outputted net is then handled by Function *Tarjan*, resulting in no nets. As for N_4, it contains no α-transitions. Therefore, we finally have $\Phi = FindSCRC(N_\Omega, \emptyset) = \{N_4\}$. In other words, the only strongly connected RIRT component in N_Ω is found out, that is N_4.

B. COMPUTATION OF ALL MINIMAL SIPHONS

In this subsection, we propose an approach to enumerate all minimal siphons in S^4PR. Π_0 and Π_1 are easily computed. We thus focus on computing minimal siphons with more than one resource. Such a computation consists of two stages:

Stage 1: We compute all strongly connected RIRT nets.

Stage 2: For each obtained strongly connected RIRT net, β-arcs and β-transitions as well as their related arcs are deleted and then Function *Check* is used to determine if the resource set of the obtained net can generate a minimal siphon.

The following Function *ComputeMiniSiphon* computes all minimal siphons in S^4PR.

Function Π =*ComputeMiniSiphon* (N)

Input: An S^4PR $N = (P_0 \cup P_A \cup P_R, T, F, W)$.
Output: The set of all minimal-siphons Π.

1) Π:=$\Pi_0 \cup \Pi_1$; /*Π is initialized as the set of all minima-siphons containing at most one resource in N.*/
/* Stage 1*/
2) $\Theta := \emptyset$; /*Θ denotes the set of all strongly connected RIRT nets induced by resource-place subsets of N, which is a global variable and can be updated in function*SonofNode* */
3) $R_{in} := \emptyset$;
4) Compute the IRT net N_Ω, where $\Omega = P_R$;
5) Let N_Ω be the root node of a tree;
6) $\Phi := FindSCRC(N_\Omega, R_{in})$;
7) **if**$\Phi \neq \emptyset$ **then**
8) Create a node Φ, R_{in});
9) Add an arc from N_Ω to the node (Φ, R_{in});
10) $\Theta := \Theta \cup \Phi$;
11) *SonofNode* (Φ, R_{in});
12) **end if**
/* Stage 2*/
13) **for** each $N' \in \Theta$ **do**
14) $N' := DeleteBeta(N')$; /* Function *DeleteBeta* returns a net by deleting β-arcs and β-transitions as well as their related arcs in a net.*/
15) **if** *Check*(N') =True**then**
16) $\Pi := \Pi \cup \{S_\Omega\}$, where Ω is the set of all resources of N';
17) **end if**
18) **end for**
19) **Output:** Π;
20) **End.**

The computation in Stage 1, i.e., the computation of all strongly connected RIRT nets, is performed based on problem partitioning [9], [35]. It is executed as follows:

Firstly, an IRT net induced by all resources of N is obtained and the set of all strongly connected RIRT components in it is computed using Function *FindSCRC*.

Secondly, problem partitioning is applied to each obtained strongly connected RIRT components to compute its inner strongly connected RIRT components, which is realized by Function *SonofNode*. More specifically, let $N' = (P', T', F')$ be a strongly connected RIRT component and R_{in} be a resource set. Suppose that $P' \backslash R_{in} = \{p_1, p_2, \ldots, p_k\}$. In this case, the problem of computing all the inner strongly connected RIRT components including R_{in} (except N') in N' is partitioned into k sub-problems, i.e.,

1) Computing all strongly connected RIRT components in N' excluding p_1 but including R_{in};

2) Computing all strongly connected RIRT components in N' excluding p_2 but including$R_{in} \cup \{p_1\}$;

...

k) Computing all strongly connected RIRT components in N' excluding p_k but including $R_{in} \cup \{p_1, p_2, \ldots, p_{k-1}\}$.

The computation in the above sub-problems can be performed by Function *FindSCRC*.

We can see that after problem partitioning is applied to each newly obtained strongly connected RIRT component, all the strongly connected RIRT nets of S^4PR can be derived. Note that Function *SonofNode* adopts depth-first search and a tree is generated to show the procedure of problem partitioning.

Theorem 3: Let N be an S^4PR. Π =*ComputeMiniSiphon*(N) is the set of all minimal siphons of N.

Proof: Based on the above analysis, all strongly connected RIRT nets of the S^4PR are derived after Stage 1 of *ComputeMiniSiphon*(N) is finished. In Stage 2, for each strongly connected RIRT nets, β-arcs and β-transitions as well as their related arcs are removed. Clearly, the obtained nets are CIRT nets. Then, for each obtained CIRT net, Function *Check* is applied. According to Theorem 2, the outputted Π consists of minimal siphons. Now, consider those RIRT nets induced by resource-place subsets that are not strongly

Function *SonofNode* (Φ, R_{in})

Input: A set of strongly connected RIRT components Φ and a resource set R_{in}.

for $N' = (P', T', F') \in \Phi$ **do**
  $R'_{in} := R_{in}$;
  for $p \in P' \backslash R_{in}$ **do**
    $N' := DeletePlace(N', p)$; /* Function *DeletePlace* returns a net by deleting a place and its related arcs in a net. */
    $\Phi' := FindSCRC(N', R'_{in})$;
    if $\Phi \neq \emptyset$ **then**
      Create a node (Φ, R_{in}');
      Add an arc labeled by "p" from N' to the node (Φ, R_{in}');
      $\Theta := \Theta \cup \Phi$;
      SonofNode (Φ, R_{in}');
    end if
    $R'_{in} := R'_{in} \cup \{p\}$;
  end for
end for

connected. The CIRT nets induced by these resource-place subsets are obviously not strongly connected. Observing Function Check, we can see it outputs False when dealing with these CIRT nets. In other words, any RIRT net that is not strongly connected cannot correspond to a minimal siphon. Consequently, it can be concluded that Π consists only and all minimal siphons of N.

C. ILLUSTRATIVE EXAMPLE

The example below is presented to illustrate the proposed approach. Consider the S[4]PR net in Fig. 4 again. We apply Function *ComputeMiniSiphon* to the net to compute all minimal siphons in it. First, we have $\Pi_0 = \{S_1, S_2, S_3\}$ and $\Pi_1 = \{S_4 - S_8\}$, as shown in Table 1.

TABLE 1. Minimal siphons with no resources and one resource.

Π_0	minimal siphons	Π_1	minimal siphons
S_1	$\{p_{10}$-$p_{15}\}$	S_4	$\{r_1, p_{14}, p_{15}, p_{22}, p_{32}$-$p_{34}\}$
S_2	$\{p_{20}$-$p_{25}\}$	S_5	$\{r_2, p_{11}, p_{13}, p_{32}, p_{33}\}$
S_3	$\{p_{30}$-$p_{34}\}$	S_6	$\{r_3, p_{12}, p_{32}\}$
		S_7	$\{r_4, p_{24}, p_{31}$-$p_{34}\}$
		S_8	$\{r_5, p_{15}, p_{21}$-$p_{23}, p_{25}\}$

Next, we compute minimal siphons with two or more resources. The procedure is as follows:

Stage 1: We compute all strongly connected RIRT nets and generate a tree to show the procedure of problem partitioning.

1) We generate the IRT net induced by all resources of the S[4]PR N, denoted as N_Ω, as shown in Fig. 8. Let N_Ω be the root node of the tree.

2) Function *FindSCRC* is applied to N_Ω with $R_{in} = \emptyset$. Since N_Ω is strongly connected and has no α-transitions, we have $\Phi_1 = FindSCRC(N_\Omega, \emptyset) = \{N_1\}$, where N_1 is exactly the same as N_Ω. Accordingly, we create a node $(\Phi_1, \emptyset)$ and add an arc from the root node to $(\Phi_1, \emptyset)$.

TABLE 2. Minimal siphons with two or more resources.

$\Pi_{x\geq 2}$	minimal siphons
$S_{\Omega 21}$	$\{r_2, r_3, p_{13}, p_{32}, p_{33}\}$
$S_{\Omega 22}$	$\{r_4, r_5, p_{15}, p_{25}, p_{31}$-$p_{34}\}$
$S_{\Omega 3}$	$\{r_1, r_4, r_5, p_{15}, p_{22}, p_{25}, p_{32}$-$p_{34}\}$
$S_{\Omega 4}$	$\{r_1, r_5, p_{15}, p_{22}, p_{23}, p_{25}, p_{32}$-$p_{34}\}$

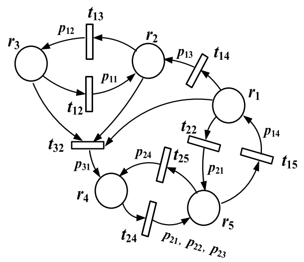


FIGURE 8. The IRT net N_Ω (N_1) w.r.t. the S[4]PR in Fig. 4 with $\Omega = \{r_1 - r_5\}$.

3) Function *SonofNode* is applied to Φ_1 with $R_{in} = \emptyset$.

Firstly, we delete r_1 and its related arcs from N_1. Function *FindSCRC* is applied to the obtained net with $R_{in} = \emptyset$, resulting in $\Phi_2 = \{N_{21}, N_{22}\}$, where N_{21} and N_{22} are shown in Fig. 10. Accordingly, we create a node $(\Phi_2, \emptyset)$ and add an arc labeled "r_1" from N_1 to $(\Phi_2, \emptyset)$. Then, Function *SonofNode* is applied to $(\Phi_2, \emptyset)$ to perform problem partitioning. Since no inner strongly connected RIRT components of N_{21} and N_{22} can be found, no son-nodes of node $(\Phi_2, \emptyset)$ are created. Next, we delete r_2 and its related arcs from N_1 and R_{in} is expanded as $R_{in} = \{r_1\}$. By the similar way, the node $(\Phi_3, \{r_1\})$ is created with an arc labeled "r_2" from N_1 to $(\Phi_3, \{r_1\})$. Then, $(\Phi_4, \{r_1\})$ and $(\Phi_5, \{r_1, r_2\})$ are similarly created one after another according to depth-first search. Finally, a tree in Fig. 9 is generated and we obtain the set of all strongly connected RIRT nets, i.e., $\Theta = \{N_1, N_{21}, N_{22}, N_3 - N_5\}$ shown in Fig. 8 and Fig. 10.

Stage 2: For each net in Θ, we delete β-arcs and β-transitions as well as their related arcs and then apply Function *Check* to the obtained net. Consider N_1. t_{22} is a β-transition in N_1 and thus it is removed, resulting in a net N_1' that is exactly the one in Fig. 5. Since *Check*(N_1')=False, $\Omega_1 = \{r_1 - r_5\}$ cannot generate a minimal siphon. Similarly, we can see $\Omega_{21} = \{r_2, r_3\}$, $\Omega_{22} = \{r_4, r_5\}$, $\Omega_3 = \{r_1, r_4, r_5\}$, $\Omega_4 = \{r_1, r_5\}$ can generate a minimal siphon while $\Omega_5 = \{r_1, r_2, r_4, r_5\}$ cannot since r_2 has no particular transitions in N_5 after removing the β-transition.

Finally, all minimal siphons are computed, that is, $\Pi = \Pi_0 \cup \Pi_1 \cup \Pi_{x\geq 2}$, where $\Pi_{x\geq 2}$ denotes the set of minimal siphons with two or more resources, as shown in Table 2.

IV. COMPARISON

Due to the fact that the number of siphons in a PN grows exponentially in the worst case with respect to the net size, all methods of siphon enumeration, including the proposed one, are theoretically of exponential complexity with respect

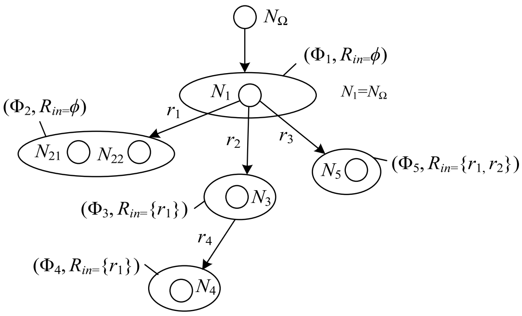


FIGURE 9. A tree generated w.r.t. the S^{4}PR in Fig. 4.

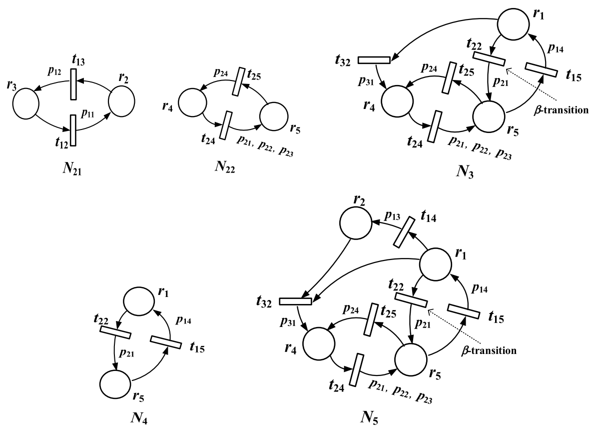


FIGURE 10. Strongly connected RIRT nets N_{21}, N_{22}, N_3, N_4, N_5.

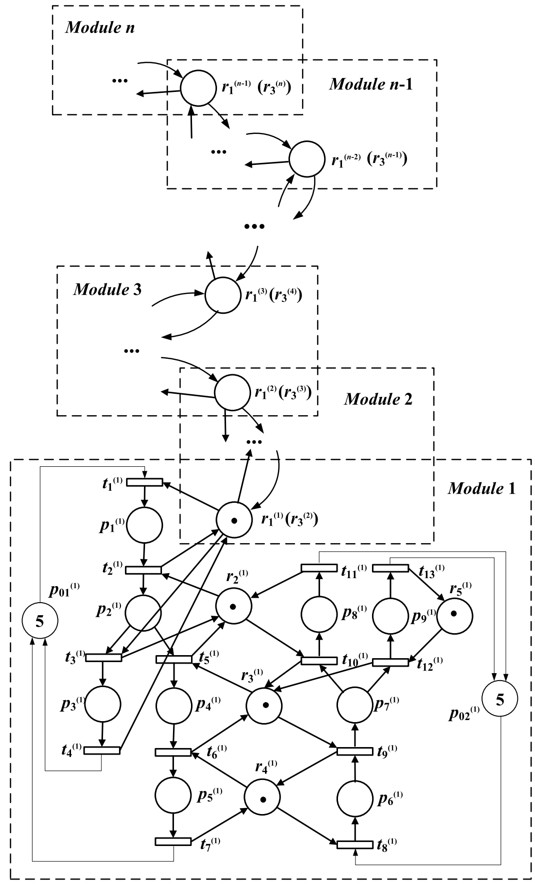


FIGURE 11. An S^{3}PR N.

to the net size. However, different methods differ greatly in computational efficiency. In this section, we present a comparison between the proposed method and the one in [33] in terms of a subclass of S^{4}PR.

S^{3}PR is a well-known subclass of PNs and it is also a subclass of S^{4}PR. An S^{4}PR $N = (P_A \cup P_0 \cup P_R, T, F, W)$ can be called an S^{3}PR if 1) N is ordinary; and 2) $\forall p \in P_A$, ${}^{\bullet\bullet}p \cap P_R = p^{\bullet\bullet} \cap P_R \neq \emptyset$ and $|{}^{\bullet\bullet}p \cap P_R| = |p^{\bullet\bullet} \cap P_R| = 1$. There are a large number of methods proposed to compute siphons in S^{3}PR, among which the one in [33] enjoys relatively good performance. Clearly, the proposed method in this work is applicable to S^{3}PR. We present the following result first.

Proposition 2 [37]: Given an S^{4}PR N such that $|{}^{\bullet}t \cap P_R| \leq 1$, $\forall t \in T$ and a resource-place subset Ω with $|\Omega| \geq 2$, $S_\Omega = \Omega \cup (\cup_{t \in {}^{\bullet}\Omega \setminus \Omega^{\bullet}} P^+(t, \Omega))$ is a minimal siphon *iff* N^*_Ω is strongly connected.

Each transition in S^{3}PR has no more than one input resource. Thus, according to Proposition 2, the key to the computation of all minimal siphons in S^{3}PR with two or more resources is to compute all strongly connected CIRT nets. Note that there are no β-arcs in any S^{3}PR. Hence, all strongly connected CIRT nets are obtained once Stage 1 of the proposed method (Function *ComputeMiniSiphon*) is finished. In other words, there is no need to call Functions *DeleteBeta* and *Check* in Stage 2 when we deal with S^{3}PR.

Consider the S^{3}PR $N = (P_0 \cup P_A \cup P_R, T, F)$ in Fig. 11. This S^{3}PR can be regarded as the combination of n ($n \geq 2$) modules with adjacent modules sharing one resource. Indeed, the net in Module 1 is exactly the one in Fig. 6 that we consider above and nets in Modules 2 to n have the same structure that is shown in Fig. 12. Note that transitions and places in different modules are distinguished by different superscripts and Modules i-1 and i share a resource that is named as $r_1^{(i-1)}$ and $r_3^{(i)}$ respectively in these two modules. We can see that the structure of Module i ($2 \leq i \leq n$) is the duplication of a part of Module 1.

We focus on the computation of minimal siphons with more than one resource. First, the proposed method is applied to the S^{3}PR:

1) The IRT net of N induced by all resources is computed, as shown in Fig. 13, denoted as N_Ω.

2) *FindSCRC* (N_Ω, .$\emptyset$) is called.

First, *Tarjan* (N_Ω) is called, outputting a net obtained from the net in Fig. 13 by deleting places $r_5^{(1)}, r_5^{(2)} \ldots, r_5^{(n)}$ and transitions $t_{12}^{(1)}, t_{12}^{(2)}, \ldots, t_{12}^{(n)}$, as well as their related arcs. Then, α-transitions emerge that are $t_{10}^{(1)}, t_{10}^{(2)}, \ldots, t_{10}^{(n)}$. Thus, after deleting them, Function *Tarjan* is called again, resulting in $n+1$ strongly connected components i.e., circuits $c_0 = r_3^{(1)} t_9^{(1)} r_4^{(1)} t_6^{(1)}$ and $c_i = r_1^{(i)} t_3^{(i)} r_2^{(i)} t_2^{(i)}$, where

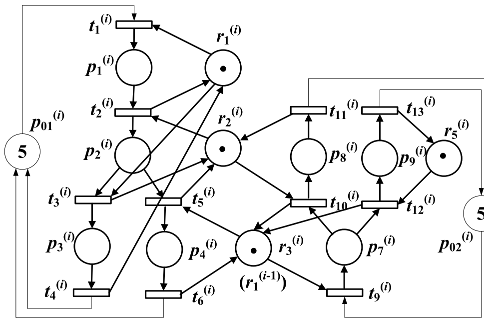


FIGURE 12. The structure of Module i ($2 \leq i \leq n$) in Fig. 11.

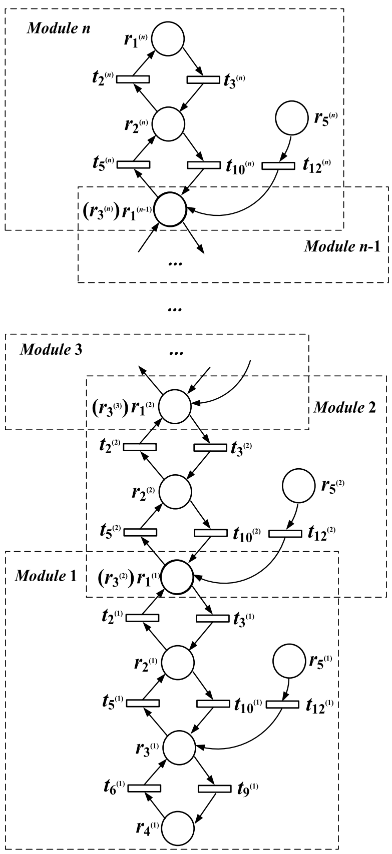


FIGURE 13. The IRT net N_Ω of the S^3PR in Fig. 11 induced by $\Omega = P_R$.

$1 \leq i \leq n$. It can be seen that for each $i \in \{1, 2, \ldots, n\}$, $t_3^{(i)}$ in c_i is an α-transition. After deleting them, we finally obtain a strongly connected RIRT component in N_Ω, that is, $c_0 = r_3^{(1)} t_9^{(1)} r_4^{(1)} t_6^{(1)}$.

It is easy to see that $c_0 = r_3^{(1)} t_9^{(1)} r_4^{(1)} t_6^{(1)}$ is the only strongly connected RIRT net in N_Ω. As analyzed before, $c_0 = r_3^{(1)} t_9^{(1)} r_4^{(1)} t_6^{(1)}$ is clearly the only strongly connected CIRT net and the only minimal siphon with more than one resource is thereby found, that is, $S = \{r_3^{(1)}, r_4^{(1)}, p_5^{(1)}, p_7^{(1)}\}$.

Next, we apply the method in [33] to the S^3PR. Before doing so, we recall its procedure: 1) Compute the resource subnet generated by all resources of the S^3PR; 2) Find out all resource circuits; 3) Derive simple loop resource-place subsets from resource circuits and resultant loop resource-place subsets via composing simple ones; 4) Compute the characteristic resource subnet for each loop resource-place subset; 5) For each characteristic resource subnet, determine whether it is strongly connected. If so, generate a minimal siphon from the corresponding loop resource-place subset. By these five steps, all minimal siphons in S^3PR with more than one resource can be computed.

Consider the S^3PR in Fig. 11. We should point out that the resource subnet generated by a resource-place subset of S^3PR is the same as the IRT net induced by the resource-place subset. It implies that the resource subnet generated by all resources of S^3PR in Fig. 11 is exactly the net N_Ω in Fig. 13. Observing N_Ω, we can see $2n+1$ resource circuits have to be found out and then $2n + 1$ simple loop resource-place subsets are computed. By composing simple loop resource-place subsets, $n+2n^2$ resultant ones are derived. As a result, $n+2n^2$ characteristic resource subnets have to be computed and for each of them, whether it is strongly connected is determined. Finally, one strongly connected characteristic resource subnet is found out, which corresponds to the minimal siphon $S = \{r_3^{(1)}, r_4^{(1)}, p_5^{(1)}, p_7^{(1)}\}$.

Clearly, the S^3PR in Fig. 11 contains only one minimal siphon with more than one resource no matter what n is equal to. It can be seen that the method in [33] requires much computation to find out this minimal siphon, whereas the proposed one offers much higher computational efficiency especially when n is large.

V. CONCLUSION

This work studies the computation of all minimal siphons in S^4PR. First, by checking structural features of CIRT nets, we propose an efficient method to decide whether a resource-place subset can yield a minimal siphon. Next, based on the determination method, an approach involving problem partitioning is developed to enumerate all minimal siphons in S^4PR. Our future work include: 1) Further improve the efficiency of minimal siphon computation in S^4PR; 2) Develop deadlock control strategies based on the proposed approach; and 3) Develop software to implement the proposed approach, making it applicable to practical large systems.

REFERENCES

[1] E. R. Boer and T. Murata, "Generating basis siphons and traps of Petri nets using the sign incidence matrix," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 41, no. 4, pp. 266–271, Apr. 1994.

[2] K. Barkaoui and B. Lemaire, "An effective characterization of minimal deadlocks and traps in Petri nets based on graph theory," in *Proc. 10th Int. Conf. Appl. Theory Petri Nets*, 1989, pp. 1–21.

[3] K. Barkaoui and I. B. Abdallah, "A deadlock prevention method for a class ofFMS," in *Proc. IEEE Int. Conf. Syst., Man*, Vancouver, BC, Canada, Oct. 1995, pp. 4119–4124.

[4] J. Bi, H. Yuan, and M. C. Zhou, "A Petri net method for compatibility enforcement to support service choreography," *IEEE Access*, vol. 4, pp. 8581–8592, 2016.

[5] A. Benveniste, E. Fabre, S. Haar, and C. Jard, "Diagnosis of asynchronous discrete-event systems: A net unfolding approach," *IEEE Trans. Autom. Control*, vol. 48, no. 5, pp. 714–727, May 2003.

[6] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of discrete event systems using labeled Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 144–153, Jan. 2014.

[7] D. Chao, "Searching strict minimal siphons for SNC-based resource allocation systems," *J. Inf. Sci. Eng.*, vol. 23, no. 3, pp. 853–867, 2007.

[8] E. E. Cano, C. A. Rovetto, and J.-M. Colom, "An algorithm to compute the minimal siphons in S^4PR nets," *Discrete Event Dyn. Syst.*, vol. 22, no. 4, pp. 403–428, 2012.

[9] R. Cordone, L. Ferrarini, and L. Piroddi, "Enumeration algorithms for minimal siphons in Petri nets based on place constraints," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 6, pp. 844–854, Nov. 2005.

[10] Y. F. Chen, Z. W. Li, K. Barkaoui, N. Q. Wu, and M. C. Zhou, "Compact supervisory control of discrete event systems by Petri nets with data inhibitor arcs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 364–379, Feb. 2017.

[11] Z. Ding, M. Jiang, and M. C. Zhou, "Generating Petri net-based behavioral models from textual use cases and application in railway networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3330–3343, Dec. 2016.

[12] H. Hu and M. C. Zhou, "A Petri net-based discrete event control of automated manufacturing systems with assembly operations," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 2, pp. 513–524, Feb. 2015.

[13] H. Hu, M. Zhou, Z. Li, and Y. Tang, "Deadlock-free control of automated manufacturing systems with flexible routes and assembly operations using Petri nets," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 109–121, Feb. 2013.

[14] B. Huang, M. C. Zhou, P. Y. Zhang, and J. Yang, "Speedup techniques for multiobjective integer programs in designing optimal and structurally simple supervisors of AMS," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2016.2587671.

[15] B. Huang, M. C. Zhou, and G. X. Zhang, "Synthesis of Petri net supervisors for FMS via redundant constraint elimination," *Automatica*, vol. 61, pp. 156–163, Nov. 2015.

[16] Y.-S. Huang, M. D. Jeng, X. Xie, and D.-H. Chung, "Siphon-based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 6, pp. 1248–1256, Jun. 2006.

[17] Z. W. Li and M. C. Zhou, "On siphon computation for deadlock control ina class of Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 3, pp. 667–679, May 2008.

[18] Z. W. Li, N. Q. Wu, and M. C. Zhou, "Deadlock control of automatedmanufacturing systems based on Petri nets-A literature review," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 437–462, Apr. 2012.

[19] G. Liu, "Complexity of the deadlock problem for Petri nets modeling resource allocation systems," *Inf. Sci.*, vol. 363, pp. 190–197, Oct. 2016.

[20] G. Liu, W. Reisig, C. J. Jiang, and M. C. Zhou, "A branching-process-based method to check soundness of workflow systems," *IEEE Access*, vol. 4, pp. 4104–4118, 2016.

[21] G. Y. Liu and K. Barkaoui, "A survey of siphons in Petri nets," *Inf. Sci.*, vol. 363, pp. 198–220, Oct. 2016.

[22] H. Liu, K. Xing, W. Wu, M. C. Zhou, and H. Zou, "Deadlock prevention for flexible manufacturing systems via controllable siphon basis of Petri nets," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 3, pp. 519–529, Mar. 2015.

[23] X. Lu, M. C. Zhou, A. C. Ammari, and J. Ji, "Hybrid Petri nets for modeling and analysis of microgrid systems," *IEEE/CAA J. Autom. Sinica*, vol. 3, no. 4, pp. 349–356, Oct. 2016.

[24] J. C. Luo, K. Y. Xing, M. C. Zhou, X. L. Li, and X. N. Wang, "Deadlock-free scheduling of automated manufacturing systems using Petri nets and hybrid heuristic search," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 3, pp. 530–541, Mar. 2015.

[25] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.

[26] L. Piroddi, R. Cordone, and I. Fumagalli, "Combined siphon and marking generation for deadlock prevention in Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 3, pp. 650–661, May 2009.

[27] N. Ran, H. Su, and S. G. Wang, "An improved approach to test diagnosability of bounded Petri nets," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 2, pp. 297–303, Feb. 2017.

[28] J. Stanley, H. Liao, and S. Lafortune, "SAT-based control of concurrent software for deadlock avoidance," *IEEE Trans. Autom. Control*, vol. 60, no. 12, pp. 3269–3274, Dec. 2015.

[29] P. H. Starke. (1992). *INA: Integrated Net Analyzer*. [Online]. Available: http://www2.info-rmatik.huberlin.de/ starke/ina.html

[30] F. Tricas, J. M. Colom, and J. Ezpeleta, "A solution to the problem of deadlocks in concurrent systems using Petri nets and integer linear programming," in *Proc. 11th Eur. Simulation Symp.*, Erlangen, Germany, 1999, pp. 542–546.

[31] F. Tricas, J. M. Colom, and J. J. Merelo, "Using the incidence matrix in an evolutionary algorithm for computing minimal siphons in Petri net models," in *Proc. 18th Int. Conf. Syst. Theory, Control Comput.*, Sinaia, Romania, Oct. 2014, pp. 645–651.

[32] A. R. Wang, Z. W. Li, and J. Y. Jia, "Efficient computation of strict minimal siphons for a class of Petri nets models of automated manufacturing systems," *Trans. Inst. Meas. Control*, vol. 33, no. 1, pp. 182–201, 2011.

[33] S. G. Wang, C. Y. Wang, M. C. Zhou, and Z. W. Li, "A method tocompute strict minimal siphons in S3PR based on loop resourcesubsets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 1, pp. 226–237, Jan. 2012.

[34] S. G. Wang, D. You, and C. Seatzu, "A novel approach for constraint transformation in Petri nets with uncontrollable transitions," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2017.2665479.

[35] S. G. Wang, D. You, C. Seatzu, and A. Giua, "Complete enumeration of minimal siphons in ordinary Petri nets based on problem partitioning," in *Proc. IEEE Conf. Decision Control*, Osaka, Japan, Dec. 2015, pp. 356–361.

[36] S. G. Wang, D. You, and C. Y. Wang, "Optimal supervisor synthesis for Petri nets with uncontrollable transitions: A bottom-up algorithm," *Inf. Sci.*, vol. 363, pp. 261–273, Oct. 2016.

[37] S. G. Wang, D. You, and M. C. Zhou, "A necessary and sufficient condition for a resource subset to generate a strict minimal siphon in S^4PR," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4173–4179, Aug. 2017, doi: 10.1109/TAC.2017.2677859.

[38] N. Q. Wu and M. C. Zhou, *System Modeling and Control With Resource-Oriented Petri Nets*. New York, NY, USA: CRC Press, 2010.

[39] N. Q. Wu, M. C. Zhou, and Z. W. Li, "Short-term scheduling of crude-oil operations: Enhancement of crude-oil operations scheduling using a Petri net-based control-theoretic approach," *IEEE Robot. Autom. Mag.*, vol. 22, no. 2, pp. 64–76, Jun. 2015.

[40] K. Xing, M. C. Zhou, F. Wang, H. Liu, and F. Tian, "Resource-transition circuits and siphons for deadlock control of automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 1, pp. 74–84, Jan. 2011.

[41] K. Y. Xing, M. C. Zhou, H. X. Liu, and F. Tian, "Optimal Petri-netbasedpolynomial-complexity deadlock-avoidance policies forautomated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 1, pp. 188–199, Jan. 2009.

[42] S. Xu, L. Dong, D. Zhu, and C. Zhu, "Method to compute minimal siphons in S^4PR nets," *J. Zhejiang Univ.*, vol. 47, no. 3, pp. 431–441, 2013.

[43] J. Ye, Z. Li, and A. Giua, "Decentralized supervision of Petri nets with a coordinator," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 6, pp. 955–966, Jun. 2015.

[44] D. You, S. G. Wang, and M. C. Zhou, "Synthesis of Monitor-based Liveness-enforcing Supervisors for S^3PR with ξ-resources," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 6, pp. 967–975, Jun. 2015.

[45] D. You, S. G. Wang, and M. C. Zhou, "Computation of strict minimal siphons in a class of Petri nets based on problem decomposition," *Inf. Sci.*, vols. 409–410, pp. 87–100, Oct. 2017.

[46] D. You, S. G. Wang, Z. W. Li, and C. Y. Wang, "Computation of an optimal transformed linear constraint in a class of Petri nets with uncontrollable transitions," *IEEE Access*, vol. 5, pp. 6780–6790, 2017.

[47] W. Y. Yu, C. G. Yan, Z. J. Ding, C. J. Jiang, and M. C. Zhou, "Modeling and validating e-commerce business process based on Petri nets," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 3, pp. 327–341, Mar. 2014.

[48] W. Y. Yu, C. G. Yan, Z. J. Ding, C. J. Jiang, and M. C. Zhou, "Modeling and verification of online shopping business processes by considering malicious behavior patterns," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 647–662, Apr. 2016.

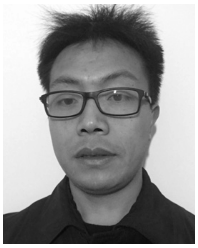
[49] Q. Zhu, N. Wu, Y. Qiao, and M. C. Zhou, "Optimal scheduling of complex multi-cluster tools based on timed resource-oriented Petri nets," *IEEE Access*, vol. 4, pp. 2096–2109, 2016.

[50] M. C. Zhou and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*. Singapore: World Scientific, 1998.

[51] Y. Xia, Y. Liu, J. Liu, and Q. Zhu, "Modeling and performance evaluation of BPEL processes: A stochastic-Petri-net-based approach," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 2, pp. 503–510, Feb. 2012.

DAN YOU received the B.S. and M.S. degrees from the School of Information and Electronic Engineering, Zhejiang Gongshang University, China, in 2014 and 2017, respectively. She is currently a member of the Discrete-Event System Group, School of Information and Electronic Engineering, Zhejiang Gongshang University. Her research interests include supervisory control of discrete event systems, and Petri net theory and application.

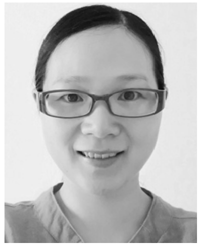
She has authored or co-authored papers published in the IEEE Transactions Automatic Control, the *Information Sciences*, the IEEE Transactions Systems, Man, Cybernetics: Systems, the IEEE Conference on Decision and Control, the IEEE International Conference on Robotics and Automation, and others.



SHOUGUANG WANG received the B.S. degree in computer science from the Changsha University of Science and Technology, Changsha, China, in 2000, and the Ph.D. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2005. He joined Zhejiang Gongshang University in 2005, where he is currently a Professor with the School of Information and Electronic Engineering, the Director of the Discrete-Event Systems Group, and the Dean of System modeling and Control Research Institute, Zhejiang Gongshang University. He is currently an Associate Editor of the IEEE Access and the IEEE/CAA Journal of Automatica Sinica.

He was a Visiting Professor with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, from 2011 to 2012. He is a Visiting Professor with the Electrical and Electronic Engineering Department, University of Cagliari, Cagliari, Italy, from 2014 to 2015. He was the Dean of the Department of Measuring and Control Technology and Instrument from 2011 to 2014.

WENZHAN DAI received the B.S. degree from Zhejiang University in 1981 and the M.S. degree from the East China Institute of Chemical Engineering in 1987, majoring in automatic control. He was promoted as Professor in 1999. He has authored over 200 papers. His research interests cover intelligent control, modeling, and signal process.



WENHUI WU received the B.S. and M.S. degrees from Zhejiang Normal University, China, in 2003 and 2006, respectively. She is currently a Senior Lab Master with the School of Information and Electronic Engineering, Zhejiang Gongshang University. Her main interests include Petri net theory and application.

YONGSHU JIA received the B.S. degree in transportation engineering from the Changsha University of Science and Technology, Changsha, China, in 2000, and the M.S. degree in vehicle utilization engineering from Jiangshu University, Zhenjiang, China, in 2008. He joined Zhejiang Industry and Trade Vocational College in 2000, where he is currently a Professor with the School of Automotive and Mechanical Engineering.

His research interests include vehicle MRF damper, and random vibration control of vehicle.