Approach for Minimal-siphon Computation in S^4PR

Dan You, Shouguang Wang, Wenzhan Dai, Wenhui Wu
School of Information & Electronic Engineering
Zhejiang Gongshang University
Hangzhou, China
youdan000@hotmail.com; wsg5000@hotmail.com

Abstract—The efficient siphon computation is the key to the development of siphon-based deadlock control strategies with good performance. This work studies the computation of minimal siphons in a class of Petri nets called S^4PR. Firstly, we propose a function with polynomial complexity to determine whether a resource subset can generate a minimal siphon. Next, using the technique of problem partitioning, a new approach is developed to compute all minimal siphons in S^4PR. Finally, an example is given to illustrate the proposed approach.

Keywords—Petri nets; S^4PR nets; minimal-siphon computation; problem partitioning

I. INTRODUCTION

A siphon is a structural object of Petri nets that is strongly related to the properties like deadlock-freedom and liveness. The number of siphons in a Petri net grows in the worst case exponentially with respect to the net size. Consequently, the computational efficiency of siphon-based deadlock control strategies [3], [6], [7], [9], [10], [19] largely depends on that of siphon computation.

Siphon computation can be classified into two categories, i.e., ones applicable to general nets [1], [2], [5], [9], [11], [13], [15] and the others applicable to specific nets only [4], [14], [17], [18]. In this work, we study the computation of minimal siphons in a class of Petri nets named Systems of Sequential Systems with Shared Resources (S^4PR) [12].

The key to minimal-siphon computation in S^4PR is to answer the question: How to determine whether a resource subset can yield a minimal siphon. For a subclass of S^4PR named “Systems of Simple Sequential Processes with Resources (S^4PR)”; our previous work [14] proposes a necessary and sufficient condition to decide if a resource subset can yield a minimal siphon. Based on this condition, a minimal-siphon computation approach with high efficiency is developed [14]. Barkaoui et al. [2] propose a characterization of minimal siphons for general nets using graph theory. However, it is not efficient when applied to S^4PR. Cano et al. [4], using pruning graphs, present a condition for a resource subset to generate a minimal siphon in S^4PR. We observe that such a determination still relies on the definition of minimal siphons. Motivated by their work, we construct characteristic implicit resource-transition (CIRT) nets in our previous work [16] and then propose a sufficient and necessary condition to determine if a resource subset can yield a minimal siphon. However, how to compute all minimal siphons in S^4PR is not presented in [16]. In this work, we develop a new method to decide if a resource subset can yield a minimal siphon in S^4PR. Moreover, based on the determination method and problem partitioning, we develop an approach to compute all minimal siphons in S^4PR.

II. PRELIMINARIES

A. Petri Nets [8]

A generalized Petri net is a 4-tuple N= (P, T, F, W) where P and T are finite, nonempty, and disjoint sets. P is the set of places, and T is the set of transitions. The set F ⊆ (P × T) ∪ (T × P) is the flow relation represented by directed arcs from places to transitions or from transitions to places. W is a mapping that assigns a weight to an arc: W(x, y)>0 if (x, y)∈F, and W(x, y)= 0 otherwise, where x, y ∈ P∪T. Given a node x∈P∪T, *x = {y ∈ P∪T | (y, x) ∈ F} is the preset of x, while x* = {y ∈ P∪T | (x, y) ∈ F} is the post-set of x. ∀X⊆ P∪T, X = X 0 ⋃ xi ∧ X* = X 0 ⋃ xi . N=(P, T, F, W) is called ordinary, denoted as N= {P, T, F}, if ∀(x, y)∈ F, W(x, y)=1. A state machine is an ordinary net such that ∀t∈ T, [t]=1.

The incidence matrix of N is [N]: P×T→Z indexed by P and T such that [N] (p, t) =W(t, p)−W(p, t). A P-vector is a column vector I: P→Z indexed by P, where Z is the set of integers. I is a P-invariant if I≠0 and I∗ [N] =0' hold. A P-invariant I is called a P-semiflow if every element of I is nonnegative. ||I||= {p∈P | |p|≠0} is called the support of I. A P-semiflow I is minimal if the greatest common divisor of its non-zero components is one and ||I|| is not a superset of the support of any other P-semiflow.

A nonempty set S ⊆P is a siphon if *S ⊆ S. A siphon is called minimal if it does not contain any other siphon.

A string π=x1x2...xn is called a path of N if ∀i∈{1, 2, ..., n-1}, xi∈x i+1 , where x1∈ P∪T. A path π=x1x2...xn is called a circuit if x1=xn. An elementary path from x_i to x_a is a path whose nodes are all different (except, perhaps, x_i and x_a). Given an elementary path π, ||π|| denotes the set of nodes in it. If there exists a path from x_i to x_a, we say that π is accessible from x_i. By default, every node is accessible from itself.
B. $S^{4}$PR

Definition 1[12]: A system of sequential systems with shared resources ($S^{4}$PR) is a generalized connected self-loop free Petri net $N=(P, T, F, W)$, where:

1. $P=P_{A} \cup P_{b} \cup P_{R}$ is a partition such that
   (a) $P_{A}=\{p_{j}, p_{r}, p_{s}, p_{t}\}$ is called the set of activity places, where $\forall i, j \in \{1, 2, \ldots, n\}, \alpha, \beta \in P_{A}$ such that $P_{A} \neq \emptyset$ and $P_{A} \cap P_{t}=\emptyset$;
   (b) $P_{b}=\{p_{j} \in P_{b} \mid \forall m \in \mathbb{N}\}$ is called the set of idle places;
   (c) $P_{R}=\{p_{r}, p_{2}, \ldots, p_{r}\}$ is the set of resource places, where $\mathbb{N}$ is the set of positive integers.

2. $T=\{t_{i}, t_{j}, t_{r} \mid \forall i, j \in \{1, 2, \ldots, n\}, \alpha, \beta \in P_{A}, T_{\alpha} \neq \emptyset, T_{j} \neq \emptyset$ and $T_{\alpha} \cap T_{j}=\emptyset\}$.

3. $\forall i \in \{1, 2, \ldots, n\}$, the subnet $N_{i}$ generated by $P_{A} \cup \{p_{i}\} \cup T_{i}$ is a strongly connected state machine such that every circuit of the state machine contains idle place $p_{i}$.

4. $\forall r \in P_{R}$, there exists a unique minimal $P$-semiflow, $I_{r} \subseteq P_{R}$ such that $[r]=||I_{r}|| \cap \mathbb{R}, P_{R} \cap ||I_{r}|| \neq \emptyset$, and $I_{r}(r)=1$. $H(r)=||I_{r}|| \cap \{r\}$ is the set of activity places that use $r$ and are called the holders of $r$. The holders of a subset of resources $H(r)$ is defined as $H(r)=\bigcup_{r \in H(r)} H(r)$.

5. $P_{t}=\bigcup_{r \in T_{t}} (||I_{r}|| \cap \{r\})$.

By Definition 1, any transition in $S^{4}$PR has at most one input activity place and at most one output activity place. The unique input and output activity places of a transition $t \in T$ are denoted by "$t$" and "$t'$", respectively.

III. CONDITION FOR RESOURCE SUBSET TO GENERATE MINIMAL SIPHON

According to the work [4], minimal siphons in an $S^{4}$PR are divided into two types: the ones with and without resources. The latter can be easily computed. Indeed, each minimal siphon in an $S^{4}$PR without resources consists of all places in a subnet $N_{i}$ (Definition 1.3), i.e., $P_{A} \cup \{p_{i}\}$. In this section, we present a method to determine whether a resource subset can generate a minimal siphon in an $S^{4}$PR. We recall the following concepts and results from our prior work [16] before proposing the method.

Definition 2[16]: Given an elementary path $\pi=x_{1}x_{2} \ldots x_{n}$ and a resource subset $\Omega$ in an $S^{4}$PR $N$, $\pi$ is said to be a pure activity path with respect to $\Omega$ if

1. $\forall p \in ||\pi|| \cap P, p \in P_{A}$; and
2. $\forall r \in (||\pi|| \cap \{x_{1}, x_{2}\}) \cap T_{t} \cap (t' \cup t) \cap \Omega=\emptyset$.

Definition 3[16]: Given a resource subset $\Omega$, $t \in T$ and $p \in P_{t}$ in an $S^{4}$PR $N$, we call $p$ is a restoring place of $t$ with respect to $\Omega$ if $t$ is accessible from $p$ via a pure activity path with respect to $\Omega$. The set of all restoring places of $t$ with respect to $\Omega$ is denoted as $P^{t}(t, \Omega)$.

Definition 4[16]: Given a resource subset $\Omega$ of an $S^{4}$PR, $S_{\Omega}$ is defined as a set of places such that $S_{\Omega}=\Omega \cup \{P_{b} \mid \forall (t, r) \in P_{b} \cap \Omega \}$.

Theorem 1[16]: Given a minimal siphon $S$ with $S \cap P_{b}=\Omega \neq \emptyset$ in an $S^{4}$PR, $S=S_{\Omega}=\Omega \cup \{P_{b} \mid \forall (t, r) \in P_{b} \}$.

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

Property 1[16]: $S_{\Omega}$ is a minimal siphon if $|\Omega|=1$.

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

Definition 5[16]: Given a resource subset $\Omega$ of an $S^{4}$PR $N=(P_{A} \cup P_{b} \cup P_{R}, T, F)$, $N_{t_{\Omega}}=(P_{t_{\Omega}}, T_{t_{\Omega}}, F_{t_{\Omega}})$ is called an $\Omega$-induced implicit resource-transition (IRT) net if

1. $P_{t_{\Omega}}=\Omega$;
2. $T_{t_{\Omega}}=T \cap H(\Omega)$; and
3. $F_{t_{\Omega}}=F_{t_{\Omega}} \cap F_{\text{out}}$, where $F_{t_{\Omega}}=(P_{t_{\Omega}} \times T_{t_{\Omega}}) \cap F$.

Definition 6[16]: Given a resource subset $\Omega$ of an $S^{4}$PR $N$, the $\Omega$-induced implicit resource-transition (IRT) net $N_{t_{\Omega}}=(P_{t_{\Omega}}, T_{t_{\Omega}}, F_{t_{\Omega}})$ is called an $\Omega$-induced implicit resource-transition (IRT) net.

Definition 7[16]: Given a resource subset $\Omega$ of an $S^{4}$PR $N$, a transition $t \in T_{t_{\Omega}}$ is called an $\alpha$-transition related to $\Omega$ if $\exists \tau \in \Omega \times \Omega$ such that $(t, r) \in F(t, \Omega)$. The set of all $\alpha$-transitions related to $\Omega$ is denoted by $T_{\alpha}(\Omega)$.

Definition 8[16]: Given a resource subset $\Omega$ of an $S^{4}$PR $N$, $N_{t_{\Omega}}=(P_{t_{\Omega}}, T_{t_{\Omega}}, F_{t_{\Omega}})$ is called an $\Omega$-induced reduced implicit resource-transition (RIRT) net if

1. $P_{t_{\Omega}}=\Omega$;
2. $T_{t_{\Omega}}=T \cap H(\Omega)$; and
3. $F_{t_{\Omega}}=F_{t_{\Omega}} \cap ((P_{t_{\Omega}} \times T_{t_{\Omega}}) \cap (T_{\alpha}(\Omega) \times P_{t_{\Omega}}))$.

According to the above definition, an $\Omega$-induced RIRT net is the net derived by deleting all $\alpha$-transitions and their related arcs in an IRT net.

Definition 9[16]: Let $\Omega$ be a resource subset of an $S^{4}$PR $N$ and $N_{t_{\Omega}}=(P_{t_{\Omega}}, T_{t_{\Omega}}, F_{t_{\Omega}})$ be the $\Omega$-induced RIRT net. An arc $(t, r) \in F_{t_{\Omega}}$ is called a $\beta$-arc related to $\Omega$ if $\exists (t, r) \in F_{t_{\Omega}}$ such that $\Gamma((t, r)) \cap \Gamma((t', r))$. The set of all $\beta$-arcs related to $\Omega$ is denoted by $\beta(\Omega)$. A transition $t \in T_{\Omega}$ is called a $\beta$-transition related to $\Omega$ if $\forall (t, r) \in F_{t_{\Omega}}$, $(t, r) \in F_{t_{\Omega}}$. The set of all $\beta$-transitions related to $\Omega$ is denoted by $\beta(\Omega)$.

Definition 10[16]: Given a resource subset $\Omega$ of an $S^{4}$PR $N$, $N_{t_{\Omega}}=(P_{t_{\Omega}}, T_{t_{\Omega}}, F_{t_{\Omega}})$ is called an $\Omega$-induced characteristic implicit resource-transition (CIRT) net if

1. $P_{t_{\Omega}}=\Omega$;
2. $T_{t_{\Omega}}=T \cap H(\Omega)$; and
3. $F_{t_{\Omega}}=F_{t_{\Omega}} \cap ((P_{t_{\Omega}} \times T_{t_{\Omega}}) \cap (T_{\alpha}(\Omega) \times P_{t_{\Omega}})) \cap F_{\beta}(\Omega)$.

It can be seen that an $\Omega$-induced CIRT net is the net derived from an IRT net by deleting all $\alpha$-transitions as well as their related arcs, then $\beta$-transitions as well as their related arcs, and finally $\beta$-arcs.

Property 2[16]: Given an $S^{4}$PR $N$, a resource subset $\Omega$ such that $|\Omega| \geq 2$ and the $\Omega$-induced CIRT net $N_{t_{\Omega}}$, $S_{\Omega}$ is not a
Definition 11: Let $N=(P, T, F, W)$ be a Petri net and $P' \subseteq P$. A transition $t \in P' \cap \tau(P,P')$ such that $t \not\subseteq \tau$ is said to be a particular output transition of $P'$.

Now, we develop a function next, by which it can be determined whether a resource subset $\Omega$ of an S4PR such that $|\Omega| \geq 2$ can generate a minimal siphon.

Function $Flag = Check(N_{\Omega}^*)$

Input: An $\Omega$-induced CIRT net $N_{\Omega}^*=(\Omega, T_\Omega^*, F_\Omega^*, W)$

Output: $Flag$. */ Flag=True implies $S_{\Omega}$ is a minimal siphon and not otherwise. */

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

Theorem 2: Given an S4PR $N$, a resource subset $\Omega$ such that $|\Omega| \geq 2$ and the $\Omega$-induced CIRT net $N_{\Omega}^*$, $S_{\Omega}^*=\Omega \cap (\bigcup_{x \in \Omega} X_{\Omega}^*)$. $P'(t, \Omega)$ is a minimal siphon iff $Check(N_{\Omega}^*)$=

We illustrate Function $Check$ by the following example.

Consider an S4PR $N$ in Fig. 1 with $P_0^*=\{p_{10}, p_{20}, p_{30}\}$, $P_r^*=\{r_1-r_5\}$, and $P_c^*=\{p_{11}-p_{15}, p_{21}-p_{25}, p_{31}-p_{34}\}$. Consider the resource subset $\Omega^*=\{r_1-r_5\}$ of $N$. The $\Omega$-induced CIRT net $N_{\Omega}^*$ is obtained, as shown in Fig. 2. The execution of $Check(N_{\Omega}^*)$ is as follows: First, $r_1$ is selected and $C=\{r_1\}$ is pushed onto $\Delta$. We can see that $C$ has its particular output transition $t_{14}$ and $r_2 \not\subseteq \tau C$ is not in $\Delta$. Hence, $C$ is updated as $C=\{r_2\}$ and it is also pushed onto $\Delta$. Similarly, following $\{r_2\}$'s particular output transition $t_{13}$, $r_3$ is found and we push $\{r_3\}$ onto $\Delta$. Following $\{r_3\}$'s particular output transition $t_{12}$, $r_2$ is found. Note that $r_2$ is in $\Delta$. According to Steps 11-13, we pop $\{r_2\}$ and $\{r_3\}$ out of $\Delta$ and then push $\{r_2, r_3\}$ onto $\Delta$. $C$ is updated as $C=\{r_2, r_3\}$. We observe that $\{r_2, r_3\}$ has no particular output transitions and it is not equal to $\Omega$. Hence, $Flag=False$ is outputted, implying $S_\Omega$ is not a minimal siphon, i.e., $\Omega$ cannot generate a minimal siphon.

Fig. 1 An S4PR $N$
IV. COMPUTATION OF MINIMAL SIPHONS IN S^4PR

In this section, we study the computation of all minimal siphons in S^4PR.

A. Computation of Strongly Connected RIRT components

Definition 12: Let N' be an Ω-induced RIRT net and Φ={Φ', Φ''} be the set of all strongly connected RIRT nets in N', N' is said to be a strongly connected RIRT component of S' if there exists a strongly connected component in N' such that Ω, where i, j ∈ {1, 2, ..., k} and i ≠ j.

In what follows, we present a way to compute all strongly connected RIRT components including a resource set R_m in a given RRT net using Function FindSCRC, where Function Do is called.

Function Φ = FindSCRC(N', R_m)

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

1) Φ := Φ'; /* Φ is a global variable that can be updated in Function Do. */
2) Do (N', R_m);
3) Output: Φ;

Function Do (N', R_m)

Input: A net N' and a resource set R_m.
1) Ψ := Tarjan (N'); /* Function Tarjan returns the set of all strongly connected components of N' */
2) for N'' = (P', T', F') ∈ Ψ such that P'' ∋ R_m do
3) if N'' contains no α-transitions then
   /* N'' is a strongly connected RIRT component. */
   4) Φ := Φ ∪ {N''};
5) else
6)   N'' := DeleteAlpha(N''); /* Function DeleteAlpha returns a net by deleting α-transitions from a net. */
7)   Do (N'', R_m);
8) end if
9) end for

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

B. Computation of All Minimal Siphons

In this subsection, we propose an approach to compute all minimal siphons in S^4PR, which is realized by the following Function ComputeMiniSiphon. Note that we use Π_0 and Π_1 to denote the sets of all minimal siphons containing no resources and only one resource in an S^4PR, respectively. It is easy to see Π_0 = ∪_{i=1}^{n} Ω_i and Π_1 = ∪_{i=1}^{n} Ψ_i due to Property 1.

Function Π = ComputeMiniSiphon (N)

Input: An S^4PR N=(P_0 ∪ P_1 ∪ P_R, T, F, W).
Output: The set of all minimal siphons Π.
1) Π := Π_0 ∪ Π_1; /* Π is initialized as the set of all minimal siphons containing at most one resource in N */
/* Stage 1: Steps 2-12 */
2) Θ := ∅; /* Θ denotes the set of all strongly connected RRT nets induced by resource subsets of N, which is a global variable and can be updated in function SonofNode */
3) R_0 := ∅;
4) Compute the IRT net N_{Ω_0} where Ω = P_R;
5) Let N_{Ω_1} be the root node of a tree;
6) Φ := FindSCRC (N_{Ω_1}, R_m);
7) if Φ ≠ ∅ then
8) Create a node (Φ, R_m);
9) Add an arc from N_{Ω_1} to the node (Φ, R_m);
10) Θ := Θ ∪ Φ;
11) SonofNode (Φ, R_m);
12) end if
/* Stage 2: Steps 13-18 */
13) for each N ∈ Π 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)     Π := Π ∪ {S_{Ω_2}}, where Ω is the set of all resources of N;
17) end if
18) end for
19) Output: Π;
20) End.

Function SonofNode (Φ, R_m)

Input: A set of strongly connected RIRT components Φ and a resource set R_m.
1) for N'' = (P', T', F') ∈ Φ do
2)   R_m'' := R_m;
3) for p ∈ P'' R_m do
4)   N'' := DeletePlace(N, p); /* Function DeletePlace returns a net by deleting a place and its related transitions in a net */
5)   Φ'' := FindSCRC (N'', R_m'');
6)   if Φ'' ≠ ∅ then
7)     Create a node (Φ'', R_m'');
8)     Add an arc labeled by "p" from N'' to the node (Φ'', R_m'');
9) end if
10) R_m := R_m'' ∪ {p};
The focus of Function ComputeMiniSiphon is the computation of minimal siphons with more than one resource. It can be divided into two stages. In Stage 1, all strongly connected RIRT nets are computed based on problem partitioning [5], [15]. In Stage 2, for each computed strongly connected RIRT net, \( \beta \)-arcs and \( \beta \)-transitions as well as their related arcs are deleted and then Function Check determines if the resource set of the obtained net can generate a minimal siphon. Finally, we have the following Theorem.

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

**C. Example**

An example is presented to illustrate the proposed approach. Consider the S\(^4\)PR in Fig. 1 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 \( \Pi_1 \{ \{S_4, S_5\} \} \), as shown in Table 1.

<table>
<thead>
<tr>
<th>( \Pi_0 )</th>
<th>Minimal siphons</th>
<th>( \Pi_1 )</th>
<th>Minimal siphons</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_1 )</td>
<td>( {p_1} )</td>
<td>( S_6 )</td>
<td>( {r_1, p_{14}, p_{15}, p_{22}, p_{23}, p_{24}} )</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>( {p_2} )</td>
<td>( S_7 )</td>
<td>( {r_2, p_{13}, p_{23}, p_{33}} )</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>( {p_{12}} )</td>
<td>( S_8 )</td>
<td>( {r_4, p_{12}, p_{22}} )</td>
</tr>
<tr>
<td>( S_4 )</td>
<td>( {p_{13}} )</td>
<td>( S_5 )</td>
<td>( {r_{12}, p_{12}, p_{22}} )</td>
</tr>
<tr>
<td>( S_5 )</td>
<td>( {p_{14}} )</td>
<td>( S_6 )</td>
<td>( {r_{15}, p_{15}, p_{25}, p_{33}, p_{34}} )</td>
</tr>
</tbody>
</table>

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 \( N_{1\alpha} \), as shown in Fig. 3. Let \( N_{1\beta} \) be the root node of the tree.

2) Function FindSCRC is applied to \( N_{1\alpha} \) with \( R_{\alpha}=\emptyset \). Since \( N_{1\alpha} \) is strongly connected and has no \( \alpha \)-transitions, we have \( \Phi_1=\text{FindSCRC}(N_{1\alpha}, \emptyset)=\{N_1\} \), where \( N_1 \) is exactly \( N_{1\beta} \). Accordingly, we create a node \( \Phi_1, \emptyset \) and add an arc from the root node to \( \Phi_1, \emptyset \).

3) Function SonofNode is applied to \( \Phi_1 \) with \( R_{\alpha}=\emptyset \).

Firstly, we delete \( r_1 \) and its related arcs from \( N_1 \). Function FindSCRC is applied to the obtained net with \( R_{\alpha}=\emptyset \), resulting in \( \Phi_2=\{N_{21}, N_{22}\} \), where \( N_{21} \) and \( N_{22} \) are shown in Fig. 5. 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 the newly created node to perform the problem partitioning for newly obtained strongly connected RIRT components \( N_{21} \) and \( N_{22} \). 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 \( R_{\alpha} \) is expanded as \( R_{\alpha}=\{r_1\} \). By repeating in a similar way, nodes \( \Phi_3, \{r_1\} \), \( \Phi_4, \{r_1\} \) and \( \Phi_5, \{r_1, r_2\} \) are created one after another according to depth-first search. Finally, a tree in Fig. 4 is generated and we obtain the set of all strongly connected RIRT nets, i.e., \( \Theta=\{N_1, N_{21}, N_{22}, N_3, N_5\} \).

Stage 2: For each net in \( \Theta \), we delete \( \beta \)-arcs and \( \beta \)-transitions as well as their related arcs and then Function Check is applied to the obtained net. Consider \( N_1 \), \( t_{22} \) is a \( \beta \)-transition in \( N_1 \) and thus it is removed, resulting in a net \( N' \), that is exactly the one in Fig. 2. As analyzed before, Check \( (N')=\text{False} \). Hence, \( \Omega_1=\{r_1, r_5\} \) cannot generate a minimal siphon. Similarly, we can see \( \Omega_2=\{r_1, r_2\}, \Omega_3=\{r_3, r_4\}, \Omega_4=\{r_1, r_4 \} \), \( \Omega_5=\{r_1, r_2, r_3, r_5\} \) can generate a minimal siphon while \( \Omega_5=\{r_1, r_2, r_3, r_5\} \) cannot since \( r_2 \) has no particular transitions in \( N_5 \) after removing the \( \beta \)-transition.

Finally, all minimal siphons are computed, that is, \( \Pi=\{S_1, S_8, S_2_{21}, S_2_{22}, S_3_{23}, S_3_{34}\} \), where \( S_2_{21}=\{r_2, r_3, p_{13}, p_{32}, p_{33}\} \), \( S_2_{22}=\{r_4, r_5, p_{15}, p_{25}, p_{33}, p_{34}\} \), \( S_3_{23}=\{r_1, r_4, r_5, p_{15}, p_{22}, p_{25}, p_{32}, p_{34}\} \), \( S_3_{34}=\{r_1, r_5, p_{15}, p_{22}, p_{25}, p_{32}, p_{34}\} \).

Fig. 3 The IRT net \( N_0 \) w.r.t. the S\(^4\)PR in Fig. 1 with \( \Omega=\{r_1, r_5\} \)

Fig. 4 A tree generated w.r.t. the S\(^4\)PR in Fig. 1
Fig. 5 Strongly connected RIRT nets N21, N22, N3, N4, N5

This work studies the computation of all minimal siphons in S^PR. First, by checking structural features of CIRT nets, we propose an efficient method to determine whether a resource subset can generate a minimal siphon. Next, based on the determination method, an approach involving problem partitioning is developed to enumerate all minimal siphons in S^PR. Our future work include: 1) Further improve the efficiency of minimal-siphon computation in S^PR; 2) Develop deadlock control strategies based on the proposed approach.

REFERENCES


