Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Enumeration Algorithms for Maximal Perfect-resource-transition Circuits and Strict Minimal Siphons in S^3PR

Keyi Xing*, Mengchu Zhou, Kefu Shi*, Lei Ren***

** The State Key Laboratory for Manufacturing Systems Engineering, and Systems Engineering Institute, Xian Jiaotong University, Xian, 710049, P. R. China E-mail: {kyxing, kfshi, lren}@sei.xjtu.edu.cn*

***The Department of Electrical and Computer Engineering, New Jersey Institute of Technology, NJ 07102, USA*

E-mail: zhou@njit.edu

Abstract: Resource-transition circuits (RTCs) and siphons are related to the deadlock problem and liveness control problem in Petri net models of automated manufacturing systems. This paper will concentrate on a particular type of Petri nets called systems of sequential processes with resources (S^3PRs) and solves the RTC and siphon enumeration problems. A graph-based technique is first used to find all elementary RTC structures. Any RTC can be expressed as a union of some elementary RTCs. Then, an iterative method is developed to recursively construct all maximal perfect-resource-transition circuits (MPCs), which can lead the system to deadlock, from the elementary RTCs. Finally, by the one-to-one correspondence between strict minimal siphons and MPCs, a new algorithm is obtained to compute strict minimal siphons in S^3PRs.

1. INTRODUCTION

In the Petri net models of automated manufacturing systems (AMS), there are two kinds of structural objects, siphons and resource transition circuits (RTCs), which are related to the liveness properties of Petri net models (Ezpeleta *et al*., 1995, Chu and Xie 1997, Fanti and Zhou, 2004, Huang *et al*., 2001, Li and Zhou, 2006, Park *et al*., 2001, Wu and Zhou, 2007, Xing *et al*., 1996, 2005, 2007b). They can be used to characterize and prevent/avoid deadlocks.

A siphon is a place set whose input transition set is included in its output transition set. A siphon is said to be minimal if it does not contain other siphons. A Petri net is deadlock-free if no strict minimal siphon (SMS) eventually becomes empty. The deadlock avoidance policy determines the set of minimal siphons that can be emptied and introduces additional places that constrain the behaviour of the systems.

A RTC is a circuit in Petri net models of AMS, which contains only resource places and transitions. Deadlocks are linked to particular RTCs called maximal perfect-resource-transition circuits (MPCs). The system liveness is characterized as no MPCs can reach its saturated states.

Most Petri net-based methods for avoiding deadlocks in AMS's are to add some control places and related arcs to strict minimal siphons or MPCs such that no siphons can be emptied (Ezpeleta *et al*., 1995, Chu and Xie 1997, Fanti and Zhou, 2004, Huang *et al*., 2001, Li and Zhou, 2006, Park *et al*., 2001, Wu and Zhou, 2007) or no MPCs can reach saturated states (Xing *et al*., 1996, 2005, 2007b).

Although SMSs and MPCs are different structural objects, it has been proved that there exists a one-to-one correspondence between them in S^3PRs (Xing *et al*., 2007a). Deadlock prevention methods based on a siphon or RTC rely on the computation and enumeration of SMSs or RTCs. But since the numbers of siphons and RTCs are exponential with the number of the size of the system (the numbers of resources, processed parts, and operations of parts), the computation of these structural components can be very time consuming. Many different methods have been developed for the computation of families of siphons (Boer and Murata, 1994, Ezpeleta, 1991, Jeng and Peng, 1996, Lautenbach, 1987).

This paper will concentrate on a particular type of AMS, which can be modelled by means of S^3PR. A graph-based technique is developed to solve the MPC enumeration problem. A well-known method for elementary circuit computation is adapted to get all elementary RTC structures in S^3PRs. An MPC can be expressed as the union of some elementary RTCs. Hence, an iterative method is developed to recursively construct all MPCs from the already-found elementary RTCs. By the one-to-one correspondence between SMS and MPC in S^3PR, an algorithm to enumerate SMS in S^3PRs is obtained.

The rest of this paper is organized as follows. Section 2 reviews basic definitions of Petri nets and S^3PRs used throughout the paper. In Section 3, structures and properties of MPCs and some useful results on siphons and MPCs in S^3PRs are introduced. The MPC and siphon enumeration algorithms are introduced in Section 4. An illustrative example is provided to illustrate the presented method and algorithm in Section 5.

2. PETRI NET PRELIMINARIES AND S^3PR CLASS

This section is a brief presentation of Petri nets and S^3PRs. For a complete study of this subject, the reader is referred to Murata 1989 and Hruz and Zhou 2007.

2.1 *Basic Definition of Petri Nets*

A Petri net is a 3-tuple $N = (P, T, F)$, where P and T are finite and disjoint sets. P is a set of places and T is a set of transitions. $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs.

Let $N = (P, T, F)$ be a Petri net. Given a vertex $x \in P \cup T$, the preset of x is defined as ${}^\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$, and the post set of x is defined as $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$. The notation can be extended to a set, for example, let $X \subseteq P \cup T$, then ${}^\bullet X = \cup_{x \in X} {}^\bullet x$ and $X^\bullet = \cup_{x \in X} x^\bullet$. N is pure if $\forall (x, y) \in F$, then $(y, x) \notin F$. A state machine is a Petri net in which each transition has exactly one input place and exactly one output place.

A marking or state of N is a mapping M: $P \to \mathbb{Z}$, where $\mathbb{Z} = \{0, 1, 2, \ldots\}$. Given a place $p \in P$ and a marking M, $M(p)$ denotes the number of tokens in p at M. Let $S \subseteq P$ be a set of places. The sum of tokens in all places of S at M is denoted by $M(S)$, i.e., $M(S) = \Sigma_{p \in S} M(p)$. A Petri net N with an initial marking M_0 is called a marked Petri net or simply, a Petri net, denoted as (N, M_0).

A transition $t \in T$ is enabled at a marking M, denoted by $M[t>$, iff $\forall p \in {}^\bullet t$, $M(p)>0$. An enabled transition t at M can be fired, resulting in a new marking M', denoted by $M[t>M'$, where $M'(p) = M(p) - 1$, $\forall p \in {}^\bullet t \backslash t^\bullet$; $M'(p) = M(p) + 1$, $\forall p \in t^\bullet \backslash {}^\bullet t$; and otherwise $M'(p) = M(p)$, for all $p \in P$. A sequence of transitions $\alpha = t_1 t_2 \ldots t_k$, $t_i \in T$, $i = 1, 2, \ldots, k$, is feasible from a marking M, if $M_i[t_i>M_{i+1}$, $i = 1, 2, \ldots, k$, where $M_1 = M$ and M_i's are called reachable markings from M. Let $\mathbb{R}(N, M_0)$ denote the set of all reachable markings of N from the initial marking M_0. (N, M_0) is bounded iff $\exists k \in \mathbb{Z} \backslash \{0\}$, $\forall M \in \mathbb{R}(N, M_0)$, $\forall p \in P$: $M(p) \leq k$ holds. We assume that in this paper all Petri nets are bounded and pure.

The incidence matrix of N is a matrix $[N]$: $P \times T \to \{-1, 0, 1\}$ such that $[N](p, t) = -1$, $p \in {}^\bullet t \setminus t^\bullet$; $[N](p, t) = 1$, $p \in t^\bullet \setminus {}^\bullet t$; and otherwise $[N](p, t) = 0$ for all $p \in P$ and $t \in T$. A nonzero $|P|$-vector I: $P \to Z$ is a *P-invariant* if $I \geq 0$ and $I^T * [N] = 0^T$, where Z is the set of integers. *The support of a P-invariant I* is the set of places: $||I|| = \{p \in P \mid I(p) \neq 0\}$. A P-invariant I is minimal if there does not exist a P- invariant I' such that $||I'|| \subset ||I||$.

A nonempty subset of places $S \subseteq P$ is a *siphon* if and only if ${}^\bullet S \subseteq S^\bullet$, i.e., an input transition is also an output transition of S. A siphon is *minimal* if and only if there does not exist a siphon contained in S as a proper subset. A minimal siphon is *strict* if it does not contain the support of any P-invariant in N. For short, strict minimal siphon is written as *SMS*.

Let $X \subseteq P \cup T$. The subnet generated by X is a Petri net $N[X] = (P_X, T_X, F_X)$, where $P_X = P \cap X$, $T_X = T \cap X$, $F_X = F \cap (X \times X)$.

A Petri net N is a bigraph in which the vertex set consists of the set of places, P, and the set of transitions, T. A *path* in N is a sequence of vertices and arcs $\alpha_{uv} = (u = x_1, x_2, \ldots, x_{k+1} = v)$, where $x_i \in P \cup T$ and $(x_i, x_{i+1}) \in F$, $i = 1, \ldots, k$, and k is the length of α. A *circuit* is a path in which the first and last vertices are identical. A vertex may appear more than once in a circuit. Two circuits are *same* if the sets of their vertices and arcs are the same, respectively. A circuit α can determine a unique subnet whose vertices and arcs are in the circuit α. We call this subnet a circuit too, for simplicity. Hence a circuit is a strongly connected subnet. A path is *elementary* if no vertex appears twice. A circuit is *elementary* if no vertex but the first and last appears twice in it. Any a circuit is the union of some elementary ones.

The composition of two Petri nets, $N_i = (P_i, T_i, F_i)$, $i \in \{1, 2\}$, via the same elements, denoted as $N_1 \otimes N_2$, is a Petri net $N_1 \otimes N_2 = (P, T, F)$, where $P = P_1 \cup P_2$, $T = T_1 \cup T_2$, and $F = F_1 \cup F_2$. And two marked Petri nets, $(N_i, M_{i0}) = (P_i, T_i, F_i, M_{i0})$, $i \in \{1, 2\}$, are compatible if $\forall p \in P_1 \cap P_2$, $M_{10}(p) = M_{20}(p)$. The composition of two compatible marked Petri nets (N_1, M_{10}) and (N_2, M_{20}) is a marked Petri net $(N_1, M_{10}) \otimes (N_2, M_{20}) = (P, T, F, M_0)$ where $N_1 \otimes N_2 = (P, T, F)$, and $\forall p \in P_1$, $M_0(p) = M_{10}(p)$, and $\forall p \in P_2$, $M_0(p) = M_{20}(p)$.

2.2 *S^3PR Class*

Researchers use Petri nets as a formalism to describe AMS and to develop appropriate deadlock resolution methods (Ezpeleta *et al.*, 1995, Chu and Xie 1997, Fanti and Zhou, 2004, Huang *et al.*, 2001, Li and Zhou, 2006, Park and Reveliotis, 2001, Wu & Zhou, 2007, Xing, *et al.*, 1996, 2007b). This subsection reviews the basic concepts and properties of the system of simple sequential processes with resources (S^3PRs), originally developed by Ezpeleta *et al.* (1995) for AMS with flexible routings. A formal definition of S^3PR is as follows.

Definition 1*:* Let $K = \{1, 2, ..., m\}$ be a finite set of indices. A net of the S^3PR class is a connected self-loop free Petri net $N = (P, T, F)$, where we have the following.

1) $P = P_S \cup P_0 \cup P_R$ is a partition such that

$P_S = \cup_{i \in K} P_{Si}$, where for each $i \in K$, $P_{Si} \neq \varnothing$, and for each $i, j \in K$, $i \neq j$, $P_{Si} \cap P_{Sj} = \varnothing$. $p \in P_{Si}$ is called an operation place.

$P_0 = \cup_{i \in K} \{p_{0i}\}$. For each $i \neq j$, $p_{0i} \neq p_{0j}$. p_{0i} is called an idle-state place.

$P_R = \{r_1, r_2, \ldots, r_n\}$, $n > 0$. r_i is called a resource place.

2) $T = \cup_{i\in K} T_i$, where for each $i\in K$, $T_i \neq \emptyset$, and for each $i, j \in K$, $i\neq j$, $T_i \cap T_j = \emptyset$.

3) $\forall i\in K$, the subnet N_i, generated by $P_{Si}\cup\{p_{0i}\}$ and the transition subset T_i connected to these places, is a strongly connected state machine, such that every directed cycle contains place p_{0i}.

4) $\forall r_l\in P_R$, there are a unique minimal p-semiflow $Y_{rl}\in\{0, 1\}^{|P|}$ such that $\{r_l\} = \|Y_{rl}\| \cap P_R$, $P_0\cap\|Y_{rl}\| = \emptyset$, $H(r_l) \equiv P_S \cap \|Y_{rl}\| \neq \emptyset$, and $Y_{rl}(r_l) = 1$. $\forall p\in H(r_l)$, $p^{\bullet\bullet}\cap P_R = {}^{\bullet\bullet}p\cap P_R = \{r_l\}$.

5) $\forall r_i, r_j\in P_R$, $\|Y_{ri}\|\cap\|Y_{rj}\| = \emptyset$, and $P_S = \bigcup_{r_l\in P_R}(\|Y_{rl}\|\setminus\{r_l\})$.

Let $N = (P_S\cup P_0\cup P_R, T, F)$ be a S^{3}PR. An initial marking M_0 of N is called an acceptable initial marking for N iff 1) $\forall p\in P_0\cup P_R$, $M_0(p) \geq 1$; 2) $\forall p\in P$, $M_0(p) = 0$.

Let $(N, M_0) = (P_S\cup P_0\cup P_R, T, F, M_0)$ be a marked S^{3}PR and a transition $t\in T$, let ${}^{(p)}t$ and $t^{(p)}$ denote the input and output operation place of t, respectively, and let ${}^{(r)}t$ and $t^{(r)}$ denote the input and the output resource place of t, respectively. Then ${}^{\bullet}t = {}^{(p)}t\cup{}^{(r)}t$ and $t^{\bullet} = t^{(p)}\cup t^{(r)}$ in N. For a given marking $M\in R(N, M_0)$, t is process-enabled at M if $M({}^{(p)}t)>0$, and t is resource-enabled at M if $M({}^{(r)}t)>0$. In S^{3}PR, only transitions, which are resource and process-enabled at the same time, can be fired.

In S^{3}PR, $H(r)$ is actually the set of all operation places that uses the resource r. For a given subset of resource places R, let $H(R) = \cup_{r\in R} H(r)$.

2.3 *MPC in S^{3}PRs*

Definitions 2–4 and Lemma 1-2 are from (Xing, *et al.*, 2007a, b).

Definition 2: A directed circuit θ in S^{3}PR N is called a resource-transition circuit (RTC) if it contains only resource places and transitions.

Let $\Im[\theta]$ and $\Re[\theta]$ denote the sets of all transitions and all resource places on θ, respectively. Let $R_1 = \Re[\theta]$. Then θ is said to be with resource set R_1. Let $\Gamma(R_1)$ denote the set of all RTCs with resource set R_1.

A RTC is determined uniquely by its transition set and resource place set. Hence, RTC θ can be denoted as $\theta = <\Re[\theta], \Im[\theta]>$.

Definition 3: Let R_1 be a set of resource places, and $\theta_1, \theta_2\in\Gamma(R_1)$. If θ_2 contains θ_1, that is, θ_1 is a subcircuit of θ_2, denoted by $\theta_1\subseteq\theta_2$, then the inclusion relation $\subseteq$ is a partial ordering relation defined on $\Gamma(R_1)$. The union of any two RTCs with resource set R_1 is also an RTC with the same resource set $R1$. Therefore, $\Gamma(R_1)$ is closed under the operator "union" and has a unique *maximal* RTC with resource set R_1.

Definition 4: A RTC θ is *perfect* if $({}^{(p)}\Im[\theta])^{\bullet} = \Im[\theta]$. Let $t\in\Im[\theta]$, θ is *perfect on t* if $\forall t_j\in({}^{(p)}t)^{\bullet}$, $t_j\in\Im[\theta]$. Let $\Pi(R_1)$ denote the set of all perfect RTCs (PRTCs) with resource set R_1.

Let $\theta_1, \theta_2\in\Pi(R_1)$. Then the union of θ_1 and θ_2 is also a PRTC with the resource set R_1, that is, $\theta_1\cup\theta_2\in\Pi(R_1)$. Therefore, $\Pi(R_1)$ contains a unique *maximal* PRTC (MPC), denoted as $\xi(R_1)$. Then $\forall\theta\in\Pi(R_1)$, $\theta\subseteq\xi(R_1)$. Let Θ denote the set of all MPCs in N.

A MPC θ is called to be *saturated* under a reachable marking M of (N, M_0) iff $M({}^{(p)}\Im[\theta]) = M_0(\Re[\theta])$.

With the above concepts, liveness characterization of S^{3}PRs is established and stated as follows (Xing *et al.*, 2007b).

Lemma 1: A marked S^{3}PR (N, M_0) is live if and only if no MPC of N is saturated at any reachable marking of (N, M_0).

Let $\aleph$ denote the set of all SMSs in N. In (Xing *et al.*, 2007a), a one-to-one mapping between Θ and $\aleph$ in S^{3}PR is established as follows.

Lemma 2: Let $N = (P_S\cup P_0\cup P_R, T, F)$ be a S^{3}PR. Define the map $\chi: \Theta\rightarrow\aleph$ as follows

$$f(\theta) = \Re[\theta]\cup H(\Re[\theta]\setminus{}^{\bullet}\Im[\theta]),\ \theta\in\Theta,$$

then f is a one-to-one mapping from Θ to $\aleph$.

3. STRUCTURES AND PROPERTIES OF MPC IN S^{3}PRS

The perfectness of RTC θ implies that if an output transition of an operation place is in θ, then its all output transitions are in θ. Let $P_\Pi = \{p\in P_S \mid |p^{\bullet}| > 1\}$. $p\in P_\Pi$ is called as *a split operation place*. A split operation place is the first node of different processing subroutes. That is, from p, the parts processed in the systems can choose different processing routes.

Let θ be an RTC in N and $t\in\Im[\theta]$. If $|({}^{(p)}t)^{\bullet}| = 1$, then $({}^{(p)}t)^{\bullet} = t\in\Im[\theta]$. Hence, to check the perfectness of RTC θ, we need only to check if $p^{\bullet}\subseteq\Im[\theta]$ for all $p\in P_\Pi\cap{}^{(p)}\Im[\theta]$. As a conclusion, we have the following results.

Lemma 3. Let θ be a RTC in N. If $\forall p\in P_\Pi$ and $p^{\bullet}\cap T[\theta]\neq\emptyset$ implying $p^{\bullet}\subseteq T[\theta]$, then θ is perfect.

By Lemma 3, if an RTC contains one output transition of a split operation place, then it contains all output transitions of the place, such RTC is perfect.

Let θ be an RTC in N. Then $<\Re[\theta], \Re[\theta]^{\bullet}\cap{}^{\bullet}\Re[\theta]>$ is connected and a maximal RTC with the resource set $\Re[\theta]$. For simplicity, let $\gamma(\theta)$ denote $<\Re[\theta], \Re[\theta]^{\bullet}\cap{}^{\bullet}\Re[\theta]>$, that is, $\gamma(\theta)\equiv<\Re[\theta], \Re[\theta]^{\bullet}\cap{}^{\bullet}\Re[\theta]$. Furthermore, if $\gamma(\theta)$ is perfect, then $\gamma(\theta)$ is MPC, that is, $\xi(\Re[\theta]) = \gamma(\theta)$.

Example 1: Consider S^3PR shown in Fig. 1, $P_\Pi = \{p_1\}$, and $p_1^\bullet = \{t_2, t_5, t_9\}$. From p_1, there are 3 different processing routes for a part. By Lemma 2, a RTC is perfect if it contains one transition in $p_1^\bullet$, then it must contain all transitions in $p_1^\bullet$. $\theta_1 = r_4t_{11}r_1t_{10}r_5t_7r_3t_6r_4$, $\theta_2 = r_4t_5r_1t_{10}r_5t_7r_3t_6r_4$, $\theta_3 = r_4t_5r_1t_{10}r_5t_7r_3t_6r_4t_{11}r_1t_{10}r_5t_9r_1t_{10}r_5t_7r_3t_6r_4$, $\theta_4 = r_2t_2r_1t_{10}r_5t_7r_3t_3r_2$, and $\theta_5 = r_2t_2r_1t_{10}r_5t_7r_3t_6r_4t_5r_1t_{10}r_5t_9r_1t_{10}r_5t_7r_3t_3r_2$ are RTCs. The S^3PR net contains only one split operation place p_1. By Lemma 3, if a perfect RTC contains one output transition of p_1, then it must contain its all output transitions. Since $P_\Pi^\bullet \cap \Im[\theta_1] = \varnothing$, and $p_1^\bullet \cap \Im[\theta_5] = p_1^\bullet$, θ_1 and θ_5 are perfect. θ_2, θ_3 and θ_4 contain one or more transitions in $p_1^\bullet$, but do not contain all of them. Hence they are not perfect.

θ_1, θ_2, and θ_4 are elementary RTCs, while θ_3 and θ_5 are not. $\Re[\theta_1] = \Re[\theta_2] = \Re[\theta_3] = \{r_1, r_3, r_4, r_5\}$, $\Re[\theta_1]^\bullet \cap {}^\bullet\Re[\theta_1] = \{t_5, t_6, t_7, t_9, t_{10}, t_{11}\}$. $\gamma(\theta_1) = \gamma(\theta_2) = \theta_3$ is not perfect. The set of all perfect RTCs with resource set $\Re[\theta_1]$, $\Pi(\Re[\theta_1])$, contains only θ_1. Thus θ_1 is an MPC. That is, $\xi(\Re[\theta_1]) = \theta_1$. $\Re[\theta_5] = \{r_1, r_2, r_3, r_4, r_5\}$, $\theta_5 \subset \gamma(\theta_5) = \xi(\theta_5)$, and $\gamma(\theta_5)$ is an MPC.

Any RTC can be expressed as a union of some elementary circuits. Let θ be an MPC with resource set R. Then there exists a set of k elementary RTCs, $\Omega = \{\theta_i, i = 1, 2, \ldots, k\}$, such that $\theta = \cup_{i\in K}\theta_i$, where $K = \{1, 2, \ldots, k\}$. If $\gamma(\theta_i)$ is perfect, then $\gamma(\theta_i) \subseteq \theta$ by the perfectness and maximality of θ. If all $\gamma(\theta_i)$, $\theta_i \in \Omega$, are MPCs, then $\theta = \cup_{i\in K}\gamma(\theta_i)$.

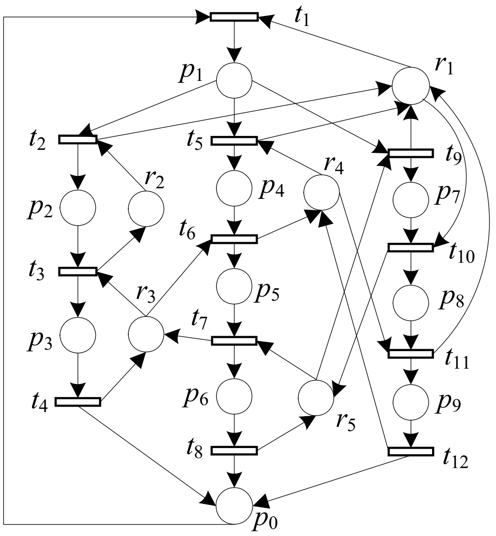

Fig. 1 A simple S^3PR

Because $\theta_i \in \Omega$ is an elementary circuit, if $\Im[\theta_i] \cap P_\Pi^\bullet \neq \varnothing$, then θ_i is not perfect. Let $t \in \Im[\theta_i] \cap P_\Pi^\bullet$. Then for each transition $t_j \in ({}^{(p)}t)^\bullet$, there exists an elementary circuit $\theta_j \in \Omega$ such that $t_j \in \Im[\theta_j]$, and $\theta = \cup\theta_j$ is perfect on t, that is, $({}^{(p)}t)^\bullet \subseteq \Im[\theta]$. Hence, for each elementary circuit $\theta_i \in \Omega$, there exists a subset Ω_i of Ω such that the union of its all elements is a perfect RTC.

Hence, for each nonperfect RTC θ contained in a MPC θ, we can construct a perfect subcircuit α of θ, which contains θ.

4. ALGORITHMS FOR FINDING MPCS AND SMSS IN S^3PRS

RTCs are only related to the transitions and resource places of N. A transition without input or output resource place cannot be in any RTCs. Actually, a transition in an RTC is in $P_R^\bullet \cap {}^\bullet P_R$. Hence, to establish an algorithm for finding all MPCs, we consider the subnet of N, defined as follows.

Definition 5: Let $N = (P_S \cup P_0 \cup P_R, T, F)$ be a S^3PR. *The resource-transition net of* N, denoted as N_R, is a subnet of N, which is generated by R and $P_R^\bullet \cap {}^\bullet P_R$, that is, $N_R = N[P_R \cup (P_R^\bullet \cap {}^\bullet P_R)]$.

In order to compute MPCs in N, all elementary RTCs in N must be found in our implementation of the procedure. The algorithm proposed by Johnson *et al.* (1975) is used to find all elementary circuits in N_R. Johnson's algorithm is extremely efficient and can find all elementary circuits in a directed graph in time bounded by $O((v + e)(c + 1))$ if the graph has v vertices, e edges, and c elementary circuits.

Furthermore, in N_R, vertices are transitions and resource places. The number of transitions is about equal to the number of all operations, $|P_S|$, and the edge number of N_R is twice the number of transitions. Therefore, there are about $(|P_S| + |P_R|)$ vertices and $2\times|P_S|$ edges.

An MPC can be expressed as the union of some elementary RTCs. Hence it can be obtained by an iterative method from the already-found elementary RTCs. The general idea to find all MPCs can be summarized as follows.

MPC Enumeration Algorithm:

Step 1 Find all elementary circuits in N_R (that is, all elementary RTCs in N) with Johnson's Algorithm. Let Ξ_{EC} be the set of all elementary circuits in N_R.

Step 2 For each $\theta = <\Re[\theta], \Im[\theta]> \in \Xi_{EC}$, if $\gamma(\theta)$ is perfect, then add $\gamma(\theta)$ into Ξ_{EC}, and delete all other $\theta^* \in \Xi_{EC}$, where $\Re[\theta^*] = \Re[\theta]$.

Step 3 Construct recursively the set Ω_C of RTCs from Ξ_{EC} as follows:

At the beginning, $\Omega_C = \varnothing$.

Add all $\theta \in \Xi_{EC}$ into Ω_C.

For each $\theta_1 \in \Xi_{EC}$, and for every RTC circuit $\theta_2 \in \Omega_C$, if $\Re[\theta_1] \cap \Re[\theta_2] \neq \varnothing$, and $\theta_1 \cup \theta_2$ is not in Ω_C, then add $\theta_1 \cup \theta_2$ into Ω_C. If $\gamma(\theta_1 \cup \theta_2)$ is perfect, then add $\gamma(\theta_1 \cup \theta_2)$ into Ω_C, and delete all other $\theta \in \Omega_C$, where $\Re[\theta] = \Re[\theta_1 \cup \theta_2]$.

Step 4 Delete all RTCs in Ω_C which are not perfect; then, for a RTC $\theta \in \Omega_C$, if $\gamma(\theta) = \theta$, move it into Θ; finally, for $\theta \in \Omega_C$, add $\cup_{\Re(\theta)=\Re(\theta)}\theta$ into Θ, and delete all RTCs with resource set $\Re(\theta)$ from Ω_C.

MPC Enumeration Algorithm can enumerate all MPCs in N.

Combining the one to one mapping from MPC to SMS in Lemma 2 with MPC Enumeration Algorithm, an algorithm to find all SMSs is obtained.

5. AN ILLUSTRATIVE EXAMPLE

Let us consider S^3PR N shown in Fig. 2, which is used to model an AMS in (Ezpeleta *et al.*, 1995). MPCs and SMSs in S^3PR are only related with the structure of Petri net models, thus, in Fig. 2, the initial marking is omitted. $P_R = \{p_{20}, p_{21}, p_{22}, p_{23}, p_{24}, p_{25}, p_{26}\}$. N contains only one split operation place p_7, $P_\Pi = \{p_7\}$, and $p_7^\bullet = \{t_6, t_{11}\}$. Then N_R is shown as in Fig. 3, where for programming simplicity, t_i or p_i is renumbered as k.

Step 1: All elementary circuits in N_R, found By Johnson's Algorithm, are listed in Table 1.

Step 2: Test the perfectness of $\gamma(\theta)$ for each $\theta \in \Xi_{EC}$. Only $\gamma(\theta_6)$ is not perfect. $\Re[\theta_2] = \Re[\theta_4]$, $\Re[\theta_3] = \Re[\theta_5]$. So delete θ_i, i=1, 2, 3, 4, 5, 7, 8, 9, from Ξ_{EC}, and then add $\gamma(\theta_i)$, i=1, 2, 3, 7, 8, 9 into Ξ_{EC}. Finally, Ξ_{EC} have 7 elements and $\Xi_{EC} = \{\gamma(\theta_1), \gamma(\theta_2), \gamma(\theta_3), \theta_6, \gamma(\theta_7), \gamma(\theta_8), \gamma(\theta_9)\}$.

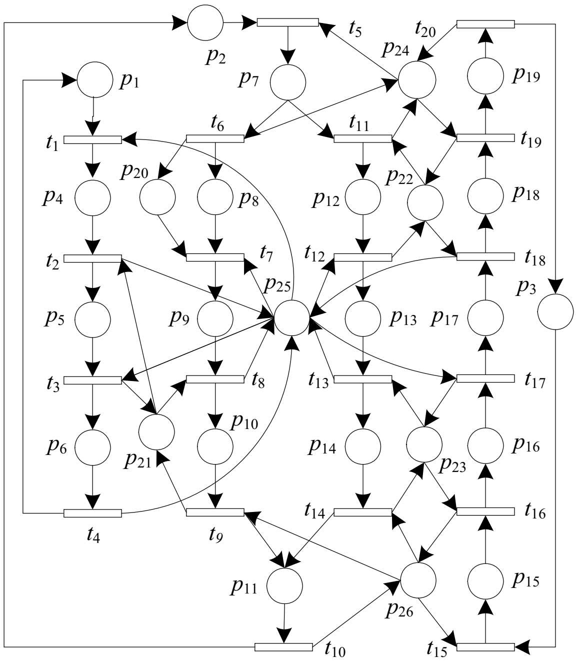

Fig. 2 A S^3PR Model N

Step 3: From Ξ_{EC}, construct the set of different RTCs, Ω_C, which contains 25 RTCs. They are listed in Table 2, where only their resource sets are listed because each of them is determined uniquely by its resource set.

Step 4: Check if a RTC $\theta \in \Omega_C$ is perfect. Since N contains only one split operation $p_{24}(5)$. If θ contains θ_6 (or α_4 in Table 2), that is, $\theta_6 \leq \theta$, but $1 \notin \Re[\theta]$, then θ is not perfect about p_{24} (5). Finally, there are 18 perfect RTCs and they are all MPCs in N.

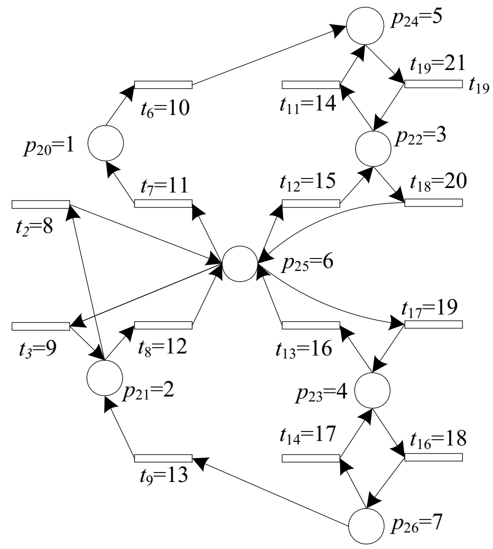

Fig. 3. The resource-transition net N_R of N shown in Fig. 2.

Table 1. All nine elementary circuits in N_R shown in Fig. 3

Ξ_{EC}	Elementary RTCs	$\Re[\theta]$	$\gamma(\theta)$ perfect?
θ_1	1 10 5 21 3 20 6 11 1	1 3 5 6	yes
θ_2	2 8 6 9 2	2 6	yes
θ_3	2 8 6 19 4 18 7 13 2	2 4 6 7	yes
θ_4	2 12 9 6 2	2 6	yes
θ_5	2 12 6 19 4 18 7 13 2	2 4 6 7	yes
θ_6	3 14 5 21 3	3 5	no
θ_7	3 20 6 15 3	3 6	yes
θ_8	4 16 6 19 4	4 6	yes
θ_9	4 18 7 17 4	4 7	yes

Table 2. RTCs in Ω_C and MPCs in N_R shown in Fig. 3

Ω_C	$\Re[\alpha]$	α perfect?	Ω_C	$\Re[\alpha]$	α perfect?
α_1	1 5 3 6	Y	α_{14}	7 4 2 3 6	Y
α_2	2 6	Y	α_{15}	5 3 6	N
α_3	2 6 4 7	Y	α_{16}	5 2 3 6	N
α_4	3 5	N	α_{17}	5 7 4 2 3 6	N
α_5	3 6	Y	α_{18}	3 4 6	Y
α_6	4 6	Y	α_{19}	3 2 4 6	Y
α_7	4 7	Y	α_{20}	6 4 7	Y
α_8	3 5 1 2 6	Y	α_{21}	4 5 3 6	N
α_9	3 5 1 2 6 4 7	Y	α_{22}	4 5 2 3 6	N
α_{10}	3 5 1 4 6	Y	α_{23}	7 3 5 1 4 6	Y
α_{11}	2 3 6	Y	α_{24}	7 3 4 6	Y
α_{12}	2 4 6	Y	α_{25}	7 4 5 3 6	N
α_{13}	2 3 5 1 4 6	Y			

5. CONCLUSIONS

In this paper, the algorithms for enumerating all MPC and SMS in S^3PR are proposed. A graph-based technique is used to find all elementary RTC structures of the subnet generated by the sets of transitions and operation places. A MPC can be expressed as the union of some elementary RTCs. Hence, an iterative method is developed to recursively construct all MPCs from the already-found elementary RTCs. By the one-to-one correspondence between SMS and MPC in S^3PR, an algorithm to enumerate SMS in S^3PR is obtained in this work.

Future work is to improve the presented algorithms, reduce the required time, and perform the computational complexity analysis.

ACKNOWLEDGMENT

This work was supported by the National Nature Science Foundation of P. R. China under Grants 60774083 and the State Key Laboratory for Manufacturing System Engineering at Xi'an Jiaotong University, Xian, 710049, P. R. China.

REFERENCES

Boer, R. and T. Murata (1994). 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.

Chu, F. and X. Xie (1997). Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Trans. Robot. Automat.*, **vol. 13, no. 6.** pp. 793–804.

Ezpeleta, J. and J. M. Couvreur (1991). A new technique for finding a generating family of siphons, traps and ST-components. Application to coloured Petri nets. In: *Proc. Int. Conf. Applications and Theory Petri Nets*, Aarhus, Denmark, pp. 126–147.

Ezpeleta, J., J. M. Colom, and J. Martinez (1995). A Petri net-based deadlock prevention policy for flexible manufacturing system. *IEEE Trans. Robot. Automat.,* **vol. 11**, **no. 2.** pp. 173–184.

Fanti M. P. and M. Zhou (2004). Deadlock control methods in automated manufacturing systems, *IEEE Transactions on Systems, Man, & Cybernetics*, Part A, **vol. 34, no. 1**, pp. 5-22.

Hruz, B. and M. C. Zhou (2007). *Modelling and Control of Discrete Event Dynamic Systems*. Springer, London, UK.

Huang, Y. S., M. D. Jeng, X. L. Xie, and S. L. Chung (2001). Deadlock prevention policy based on Petri nets and siphons. *Int. J. Prod. Res.*, **vol. 39, no. 3,** pp. 283–305.

Jeng, M. D. and M. Y. Peng (1996). Generating minimal siphons and traps for Petri nets. In: *Proc. IEEE Int. Conf. Systems*, *Man and Cybernetics*, Beijing, China. **vol. 4**, pp. 2996–2999.

Johnson, D. B (1973). Finding all the Elementary Circuits of a Directed Graph. *SIAM J. Comput.*, **vol. 4, no. 1**. pp. 77-84.

Lautenbach, K. (1987). Linear algebraic calculation of deadlocks and traps. In: *Concurrency and Nets-Advances in Petri Nets*, K. Voss, H. J. Genrich, and G. Rozenberg, Eds. pp. 315-336, New York: Springer-Verlag.

Li, Z. and M. C. Zhou (2006). Two-stage method to design liveness-enforcing Petri net supervisor for FMS. *IEEE Trans. on Industrial Informatics*. **vol. 2, no. 4**, pp. 313-325.

Murata T (1989). Petri nets: Properties, analysis and applications. *Proc. IEEE*, **vol. 77, no. 4**, pp. 541–580.

Park, J. and S. A. Reveliotis (2001). Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. *IEEE Trans. Automat. Contr.*, **vol. 46, no. 10.** pp. 1572–1583.

Reveliotis, S. A (2003). On the siphon-based characterization of liveness in sequential resource allocation systems. In: *Applications and Theory of Perti Nets*, pp. 241-255.

Wu, N. and M. C. Zhou (2007). Deadlock Resolution in Automated Manufacturing Systems with Robots. *IEEE Trans. on Automation Science and Engineering,* **vol. 4, no. 3**, pp. 474-480.

Xing, K. Y., B. S. Hu, and H. X. Chen (1996). Deadlock avoidance policy for Petri net modelling of flexible manufacturing systems with shared resources. *IEEE Trans. Automat. Contr*., **Vol. 41, No. 2**, pp. 289–295.

Xing, K. Y., B. S. Hu (2005). Optimal liveness Petri net controllers with minimal structures for automated manufacturing systems, *IEEE International Conf. on Syst., Man, Cybern.*, Hawaii, pp. 1651-1656.

Xing, K. Y., J. M. Li, F. Tian, and X. J. Yang (2007). Corresponding Relationship of Deadlock Structures in Petri Net Models for Automated Manufacturing Systems. In: *Proceedings of the 4th International Conference on Impulsive and Hybrid Dynamical Systems*, Nianning, China, pp. 2168-2172.

Xing K. Y., F. Tian, and X. J. Yang (2007). Optimal deadlock avoidance Petri net supervisors for automated manufacturing systems. *Journal of Control Theory and Applications*, **vol. 5, no. 2**, pp. 152-158.