Advance Access published on May 12, 2006
doi:10.1093/comjnl/bxl019

Computation of Elementary Siphons in Petri Nets For Deadlock Control

Daniel Yuh Chao[1]

Department of Management and Information Science, National Cheng Chi University, Taipei, Taiwan, ROC
Email: yaw@mis.nccu.edu.tw

When designing liveness-enforcing Petri net supervisors, unlike other techniques, Li *et al*. added control places and arcs to a plant net model for its elementary siphons only, greatly reducing the structural complexity of the controlled system. Their method, however, suffers from the expensive computation of siphons. We propose a new *T*-characteristic vector ζ to compute *strict minimal siphons* (*SMS*) for *S^3PR* (*systems of simple sequential processes with resources*) in an algebraic fashion. For a special subclass of S^3PR, called *S^4PR* (*simple S^3PR*), we discover that elementary siphons can be constructed from elementary circuits where all places are resources. Thus, the set of elementary siphons can be computed without the knowledge of all SMS. We also propose to construct characteristic *T*-vectors η by building a graph to find dependent siphons without their computations.

Keywords: flexible manufacturing systems, deadlock prevention, Petri nets, siphons, S^3PR.

Received 20 June 2005; revised 2 April 2006

1. INTRODUCTION

Flexible manufacturing systems (*FMS*) often share resources such as machines, robots, fixtures, buffers and palletized and programmable material handling systems. An FMS offers a very promising approach to productivity increase through state-of-the-art manufacturing technology. Deadlock interrupts normal operation schedules significantly degrading the performance.

There are three approaches to deadlock controls: recovery, avoidance and prevention [1–15]. Recovery restores the system to a normal state so as to be able to finish the production. Avoidance determines possible system evolutions at each system state and chooses the correct ones to proceed. Prevention synthesizes or builds net models with desired properties such as deadlock-freeness or liveness using special rules [14, 15], or it establishes the control policy in a static way [7–11] by building freely a *Petri net* (*PN*) model and adding necessary control places and related arcs to make it deadlock-free [8–11, 16–19].

Prevention is preferred to avoidance by carrying out the computation once and off-line. Without the need to do on-line analysis upon system changing states, it is essential when real time response time is critical since it runs much faster.

Ezpeleta *et al*. [10] proposed a class of PN called systems of simple sequential processes with resources (S^3PR). They compute all minimal siphons with no traps (called *strict minimal siphons*) of the given model and find the maximum number of tokens at each idle state followed by a control policy of adding arcs and nodes with tokens. Most recent deadlock control approaches [8, 9, 11, 14] extend Ezpeleta's work.

Liveness can be enforced by adding a control place to each strict minimal siphon (SMS). The method is simple and guarantees a success, but suffers from adding too many control places and arcs, leading to a much more complex PN than the uncontrolled one. In fact, the same amount of places as that of SMS are added in the target net and further, much more arcs are generally added, particularly for large-scale PN.

Iterative control methods in [16] find all SMS and add control places in each iteration. Repeat it until there are no more new emptiable SMS. With so many SMS, it becomes difficult and complex even for a moderate-size model.

To control fewer SMS, Huang *et al*. [8, 11] employed the mixed integer programming (MIP) technique to find a maximal siphon under a given marking. From it, all possible emptiable SMS can be extracted [8] without finding all SMS. However, Li and zhou [18] proposed simpler Petri net controllers by adding control places to elementary siphons

[1]The former name of the author is Yuh Yaw, which appeared in some of his earlier papers.

only (generally a much smaller set than that of SMS in large Petri nets). SMS can be divided into two groups: elementary and dependent; characteristic T-vectors of the latter are linear combinations of that of the former.

They added a control place for each elementary siphon without generating new SMS by the method developed in [10], while controlling all other SMS (i.e. always marked) too. This leads to much fewer control places to be suitable for large-scale Petri nets.

However, their method has to compute all SMS; the number of them grows exponentially with that of places [19, 20]. Hence their algorithm takes exponential time in addition to the extra efforts (also exponential time) to extract elementary siphons from all SMS. We propose an algorithm to compute elementary siphons based on a new characteristic T-vector (Section 4) without the knowledge of all SMS.

Section 2 presents the preliminaries followed by Section 3 where we show that a siphon can be synthesized by constructing handles upon a circuit. Section 4 defines elementary, dependent siphons and characteristic T-vectors respectively. Sections 5 and 6 compute all elementary and dependent siphons respectively. Section 7 concludes the paper.

2. PRELIMINARIES

In this paper, we consider strongly connected nets only.

DEFINITION 1. *A Petri net is a 4-tuple* $PN = (P, T, F, M_0)$ *where* $P = \{p_1, p_2, \ldots, p_a\}$ *is a set of places,* $T = \{t_1, t_2, \ldots, t_b\}$ *a set of transitions, with* $P \cup T \neq \emptyset$ *and* $P \cap T = \emptyset$, F: $(P \times T) \cup (T \times P) \to \{0, 1, 2, \ldots\}$ *is the flow relation and a marking of* N *is a mapping* M: $P \to \boldsymbol{IN}$, *where* $\boldsymbol{IN} = \{0, 1, 2, \ldots\}$. *The i-th component of* M, $M(p_i)$, *represents the number of tokens in place* p_i *under* M. *A node* x *in* $N = (P, T, F)$ *is either a* $p \in P$ *or a* $t \in T$. *The postset of node* x *is* $x\bullet = \{y \in P \cup T \mid F(x, y) > 0\}$, *and its preset* $\bullet x = \{y \in P \cup T \mid F(y, x) > 0\}$. *The preset (postset) of a set is defined as the union of the presets (postsets) of its elements. A directed path* $\Gamma = [n_1\ n_2 \cdots n_k]$, $k \geq 1$, *is a graphical object containing a sequence of nodes and the single arc between each two successive nodes in the sequence.* $N' = (P', T', F')$ *is called a subnet of* N *where* $P' \subseteq P$, $T' \subseteq T$ *and* $F' = F \cap ((P' \times T') \cup (T' \times P'))$.

The incidence matrix of N *is a matrix* A: $P \times T \to Z$ *indexed by* P *and* T *such that* $A(p, t) = -1$, *if* $p \in \bullet t \backslash t\bullet$; $A(p, t) = 1$, *if* $p \in t\bullet \backslash \bullet t$; *otherwise* $A(p, t) = 0$ *for all* $p \in P$ *and* $t \in T$, *where* Z *is the set of integers.*

Ordinary Petri nets (OPN) are those for which F: $(P \times T) \cup (T \times P) \to \{0, 1\}$. *An OPN is called a state machine (SM) if* $\forall t \in T$, $|t\bullet| = |\bullet t| = 1$. *For an OPN,* t_i *is firable or enabled under* M, *in symbols* $M[t>$, *iff* $\forall p_j \in \bullet t$, $M(p_j) > 0$. *Firing* t_i *under* M_0 *removes a token from* p_j *and deposits a token into each place* p_k *in* $t\bullet$, *moving the system state from* M_0 *to* M_1. *Repeating this process, it reaches* M' *by firing a sequence* σ *of transitions.* M' *is said to be reachable from* M_0, *i.e.* $M_0[\sigma > M'$.

All nets referred to in this paper will be OPN.

DEFINITION 2. *Let* $N = (P, T, F)$ *be a net,* (N, M_0) *a marked net and* $R(N, M_0)$ *the set of markings reachable from* M_0. *A transition* $t \in T$ *is live under* M_0 *iff* $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M_0)$: $M'[t>$ *holds. A transition* $t \in T$ *is dead under* M_0 *iff* $\nexists M \in R(N, M_0)$: $M[t>$ *holds. A PN* (N, M_0) *is live under* M_0 *iff* $\forall t \in T$, t *is live under* M_0. *It is bounded if* $\forall M \in R(N, M_0)$, $\forall p \in P$, $M(p) \leq k$, *where* k *is a positive integer.*

DEFINITION 3. $\Gamma = [n_1\ n_2 \cdots n_k]$, $k \geq 1$, *is an elementary directed path in* N *if* $\forall (i, j)$, $1 \leq i < j \leq k$, $n_i \neq n_j$. Γ *is (non) virtual if it contains only (more than) two nodes.* Γ *is an elementary circuit* c *in* N *if* $\forall (i, j)$, $1 \leq i \leq j \leq k$, $n_i = n_j$ *implies that* $i = 1$ *and* $j = k$.

DEFINITION 4. *A siphon (trap)* D (τ) *is a nonempty subset of places if* $\bullet D \subseteq D\bullet$ $(\tau\bullet \subseteq \bullet\tau)$, *i.e. every transition having an output (input) place in* $D(\tau)$ *has an input (output) place in* $D(\tau)$. *A minimal siphon does not contain a siphon as a proper subset. It is called a strict minimal siphon (SMS), denoted by* S, *if it does not contain a trap.*

DEFINITION 5. *A subnet* $N_i = (P_i, T_i, F_i)$ *of* N *is generated by* $X = P_i \cup T_i$, *if* $F_i = F \cap (X \times X)$. *It is an I-subnet of* N *if* $T_i = \bullet P_i$. I_S *is the I-subnet (derived from* $(S, \bullet S)$*) of an SMS* S. *Note that* $S = P(I_S)$; S *is the set of places in* I_S.

We add ***bold texts*** *for new terms to the following definitions* [10]. *Refer to* [10] *for more details of the* S^3PR *model.*

DEFINITION 6. [10]. *A simple sequential process* (S^2P) *is a net* $N = (P \cup \{p^0\}, T, F)$ *where (1)* $P \neq \emptyset$, $p^0 \notin P$ *(*p^0 *is called the process idle or initial or final state); (2)* N *is a strongly connected state machine (SM) and (3) every circuit of* N *contains place* p^0. ***Any elementary circuit containing*** p^0 ***is called a subprocess***.

DEFINITION 7. [10]. *A simple sequential process with resources* (S^2PR), *also called a working processes (WP), is a net* $N = (P \cup \{p^0\} \cup R, T, F)$ *so that (1) the subnet generated by* $X = P \cup \{p^0\} \cup T$ *is an* S^2P; *(2)* $R \neq \emptyset$ *and* $P \cup \{p^0\} \cap R = \emptyset$; *(3)* $\forall p \in P$, $\forall t \in \bullet p$, $\forall t' \in p\bullet$, $\exists r_p \in R$, $\bullet t \cap R = t'\bullet \cap R = \{r_p\}$; *(4) the two following statements are verified:* $\forall r \in R$, *(4.a)* $\bullet\bullet r \cap P = r\bullet\bullet \cap P \neq \emptyset$, *(4.b)* $\bullet r \cap r\bullet = \emptyset$; *(5)* $\bullet\bullet(p^0) \cap R = (p^0)\bullet\bullet \cap R = \emptyset$. $\forall p \in P \cup \{p^0\}$, p *is called a state place.* $\forall r \in R$, r *is called a resource place.* $H(r) = \bullet\bullet r \cap P$ *denotes the set of holders of* r *(states that use* r*).* $\rho(\boldsymbol{r}) = \{\boldsymbol{r}\} \cup \boldsymbol{H}(\boldsymbol{r})$ *denotes the union of* $H(r)$ *and* $\{r\}$.

The above models the constraints as follows: Definition 7.3 allows only one shared resource to be used at each state; Definition 7.4.a dictates that the resource used in a state be released when moving to the next state; Definition 7.4.b shows that two adjacent states cannot use the same resource and Definition 7.5 ensures that the initial and final state do

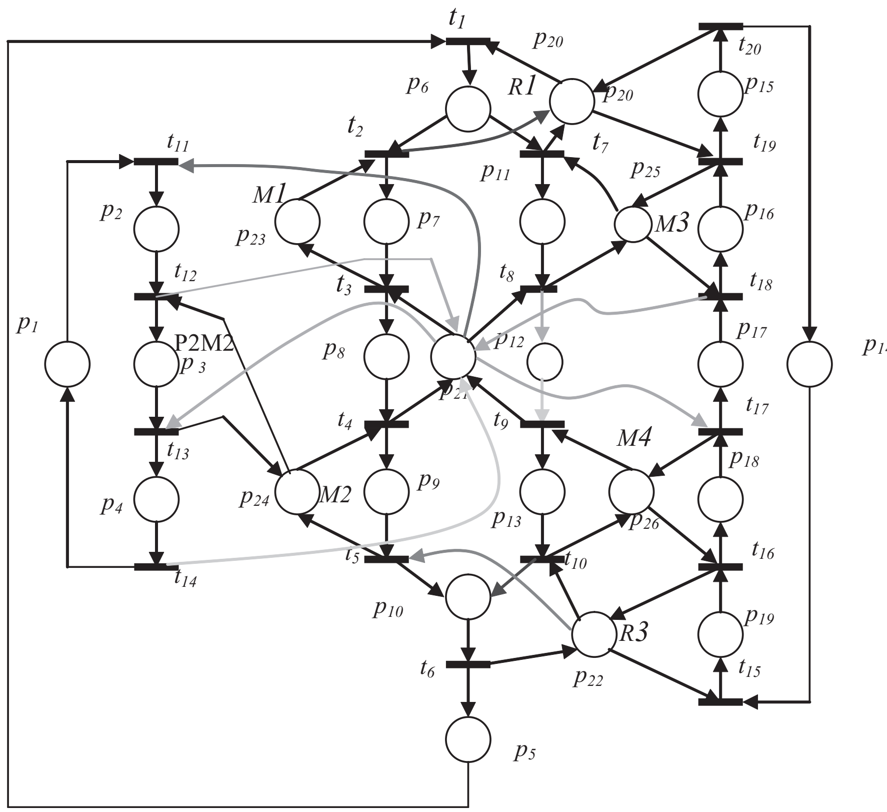

FIGURE 1. An example of systems of simple sequential processes with resources (S^3PR) [18]. A colour version of this figure is on *The Computer Journal* website.

not use resources. S^2PR is an S^2P which uses a unique resource at every nonidle state.

DEFINITION 8. [10]. *A system of S^2PR (S^3PR) is defined recursively as follows*: (1) *An S^2PR is defined as an S^3PR*; (2) *Let $N_i = (P_i \cup P_i^0 \cup R_i, T_i, F_i)$, $i \in \{1, 2\}$ be two S^3PR so that $(P_1 \cup P_1^0) \cap (P_2 \cup P_2^0) = \emptyset$. $R_1 \cap R_2 = P_C(\neq\emptyset)$ and $T_1 \cap T_2 = \emptyset$. The net $N = (P \cup P^0 \cup R, T, F)$ resulting from the composition of N_1 and N_2 via P_C defined as follows*: (1) $P = P_1 \cup P_2$; (2) $P^0 = P_1^0 \cup P_2^0$; (3) $R = R_1 \cup R_2$; (4) $T = T_1 \cup T_2$ *and* (5) $F = F_1 \cup F_2$ *is also an S^3PR.* ***A directed path (circuit, subnet) Γ in N is called a resource path (circuit, subnet) if $\forall p \in \Gamma$, $p \in R$. An elementary resource circuit is both a resource and an elementary circuit***.

The S^3PR example in Figure 1 [18] consists of three robots ($R1$, $R2$, $R3$) and four machines ($M1$–$M4$).

3. SYNTHESIS OF SMS

We construct an SMS based on the concept of handles. Roughly speaking, a 'handle' is an alternate disjoint path between two nodes. A *PT-handle* starts with a *place* and ends with a *transition* while a *TP-handle* starts with a *transition* and ends with a *place*.

DEFINITION 9. [21]. *Let $N = (P, T, F)$. $H_1 = [n_s n_1 n_2 \cdots n_k n_e]$ and $H_2 = [n_s n'_1 n'_2 \cdots n'_h n_e]$ are elementary directed paths, n_i, $n'_j \in P \cup T$, $i = 1, 2, \ldots, k$, $j = 1, 2, \ldots, h$. Each is called a handle in N if $n_i \neq n'_j$, $\forall i, j$. H_1 is an XY-handle where X and Y can be T or P. X is T (P) if $n_s \in T$ ($n_s \in P$). Y is T (P) if $n_e \in T$ ($n_e \in P$). H_1 is a resource handle if all places in H_1 are resource places. The handle H to a subnet N' (similar to the handle of a tea pot) is an elementary directed path from n_s in N' to another node n_e in N'; any other node in H is not in N'. H is said to be a handle in $N' \cup H$ which denotes the union of graphical objects N' and H.*

In Figure 1, $[t_{18}\ p_{16}\ t_{19}\ p_{25}]$ is a TP-handle with $n_s = t_{18}$ and $n_e = p_{25}$, and $[p_{21}\ t_{11}\ p_2\ t_{12}\ p_{21}]$ is a PP-handle. H = $[p_{21}\, t_{17}]$ is a virtual PT-handle since it contains only two nodes.

LEMMA 1. (1) *I_S is strongly connected* [20, 21]. (2) *A subnet N' is an I-subnet* (*see Definition* 5) *of a minimal siphon iff N' is maximal in the sense that each handle H in N' is a PP- or TP- or virtual PT-handle and there are none of PP-, TP- and virtual PT-handles to N'.* (3) *$P(N')$ is an SMS iff there is a nonvirtual PT-handle to N'', which is a subnet of N' without any TP-handles.*

Proof. (2) ($\rightarrow$) Assume contrarily that H may be a nonvirtual PT- or TT-handle. All places in H can be deleted from the siphon while other places still form a siphon violating the fact that it is minimal. (←) There is a circuit c in N' since it is strongly connected by (1). $P(N')$ is a siphon because $\forall p \in P(N')$, $\bullet p \in T(N')$ since $\bullet p$ is in c or in a PP- or TP-handle of N′. For the same reason, $\forall t \in T(N')$, $t \in P(N')\bullet$, i.e. $T(N') \subseteq P(N')\bullet$. (Note that $p\bullet$ may not be in $T(N')$ if $p\bullet$ is in a PT-handle to N'.) Thus $\bullet P(N') \subseteq P(N')\bullet$. It is minimal since if we remove a p in any TP- or PP-handle, then it is not a siphon. (3) (→) Assume contrarily that there are no nonvirtual PT-handles to N'. $\forall p \in P(N')$, $\forall t \in P(N')\bullet$, it is in I_S and $\exists p' \in P(N')$, $t \in \bullet p'$. We have $\bullet P(N') = P(N')\bullet$, and $P(N')$ is a trap—contradiction. (←) Assume contrarily that $P(N')$ contains a trap and is not an SMS whose I-subnet must include N''. Let t be the output transition of n_s of the nonvirtual PT-handle. We have $t \in P(N'')\bullet$, $t \notin \bullet P(N'')$; hence, $P(N'')$ cannot be a trap and $P(N')$ does not contain a trap—contradiction. □

DEFINITION 10. *An elementary resource circuit is called a basic circuit, denoted by c_b. The procedure to add handles to form I_S based on Lemma 1 is called handle-construction. The corresponding $S = P(I_S)$ is called a basic siphon. The set of resource places in c_b is denoted by R_b. An expanded c_b is the union of c_b and the set of all PP-handles (called PP′-handles) of the form $[r_1\ t'\ r_2]$, where $r_1 \in c_b$ and $r_2 \in c_b$.*

EXAMPLE. In Figure 1, first find a circuit $c_b = [p_{21}\ t_{17}\ p_{26}\ t_{16}\ p_{22}\ t_5\ p_{24}\ t_4\ p_{21}]$. Second add PP′-handles $[p_{24}\ t_{12}\ p_{21}]$, $[p_{21}\ t_{13}\ p_{24}]$, $[p_{26}\ t_9\ p_{21}]$ and $[p_{22}\ t_{10}\ p_{26}]$ to form the expanded c_b. Next, add TP-handles $[t_{10}\ p_{10}\ t_6\ p_{22}]$, $[t_{13}\ p_4\ t_{14}\ p_{21}]$ and $[t_{17}\ p_{17}\ t_{18}\ p_{21}]$ to get $S_4 = \{p_4, p_{10}, p_{17}, p_{21}, p_{22}, p_{24}, p_{26}\}$ with nonvirtual PT-handles $[p_{21}\ t_{11}\ p_2\ t_{12}]$, $[p_{21}\ t_3\ p_8\ t_4]$, $[p_{21}\ t_8\ p_{12}\ t_9]$ and $[p_{22}\ t_{15}\ p_{19}\ t_{16}]$ to the c_b.

Note that in the expanded c_b, we can find other basic circuits, e.g. $[p_{22}\ t_{10}\ p_{26}\ t_{16}\ p_{22}]$. Thus, fewer basic circuits need to be searched.

Note that if there were PP-, TP- and virtual PT-handles to N', i.e. N' was not maximal, then S would not be a siphon. If I_S contains only one resource place r, it cannot be strict since $\rho(r)$ (e.g. $\rho(p_{26}) = \{p_{26}, p_{18}, p_{13}\}$) is both a trap and a siphon [10].

Also the synthesized siphon may not be minimal since part of the c_b or handles may become a TT-handle. In Figure 1, the siphon constructed based on $c_b = [p_{20}\ t_{19}\ p_{25}\ t_7\ p_{20}]$ is not minimal since $[t_{19}\ p_{25}\ t_7]$ is a TT-handle to $I(\rho(p_{20}))$. Note the presence of virtual PT-handle $[p_6\ t_7]$.

LEMMA 2. *All places in a c_b must be resource places.*

Proof. Suppose there are state places in the c_b, then there is an edge $[t_1\ p_1\ t_2]$ (on the c_b) in a certain WP_j. There are two possible outgoing edges from t_2: (a) $[t_2\ r_1]$, $p_1 \in H(r_1)$ and (b) $[t_2\ p_2]$, $p_2 \in WP_j$. For Case (b), repeat the arguments and continue. For Case (a), the next edge from r_1, $[r_1\ t_3]$ is part of a circuit of r_1 as edges $[t_1\ p_1\ t_2]$ and $[t_2\ r_1]$ are. t_3 belongs to WP_j or a different WP_k. In both cases, $[r_1\ t_1]$ is a virtual PT-handle and $[t_3\ p_3\ t_4\ r_1]$ a TP-handle to or part of c_b respectively. Thus both circuits $[r_1\ t_1\ p_1\ t_2\ r_1]$ and $[r_1\ t_3\ p_3\ t_4\ r_1]$ belong to the I_D constructed from c_b. Other circuits of r_1 are in I_D. Hence $I(\rho(r_1)) \subset I_D$ and the siphon D constructed is not minimal since $I(\rho(r_1))$ is the I-subnet of a minimal siphon containing r_1 and $H(r_1)$ and D contains another minimal siphon as a proper subset. □

For instance, in Figure 1, the nonminimal siphon constructed on $c = [p_{25}\ t_7\ p_{11}\ t_8\ p_{12}\ t_9\ p_{13}\ t_{10}\ p_{26}\ t_{16}\ p_{18}\ t_{17}\ p_{17}\ t_{18}\ p_{16}\ t_{19}\ p_{25}]$ contains state places p_{11}, p_{12} etc. and $\rho(p_{25})$. Lemma 2 helps to locate a c_b.

The following table shows all possible basic siphons and circuits in Figure 1.

Basic siphons	Places	c_b
S_1	$p_{10}, p_{18}, p_{22}, p_{26}$	$[p_{22}\ t_{10}\ p_{26}\ t_{16}\ p_{22}]$
S_4	$p_4, p_{10}, p_{17}, p_{21}, p_{22}, p_{24}, p_{26}$	$[p_{21}\ t_{17}\ p_{26}\ t_{16}\ p_{22}\ t_5\ p_{24}\ t_4\ p_{21}]$
S_{10}	$p_4, p_9, p_{12}, p_{17}, p_{21}, p_{24}$	$[p_{21}\ t_{13}\ p_{24}\ t_4\ p_{21}]$
S_{16}	$p_2, p_4, p_8, p_{13}, p_{17}, p_{21}, p_{26}$	$[p_{21}\ t_{17}\ p_{26}\ t_9\ p_{21}]$
S_{17}	$p_2, p_4, p_8, p_{12}, p_{15}, p_{20}, p_{21}, p_{23}, p_{25}$	$[p_{21}\ t_3\ p_{23}\ t_2\ p_{20}\ t_{19}\ p_{25}\ t_{18}\ p_{21}]$
S_{18}	$p_2, p_4, p_8, p_{12}, p_{16}, p_{21}, p_{25}$	$[p_{21}\ t_8\ p_{25}\ t_{18}\ p_{21}]$

There may be PP-handles to a c_b which are also resource paths resulting in a new c_b. We do not have to consider TP-handles of resource paths for a different reason—they simply do not exist.

LEMMA 3. *Let H be a TP-handle or a nonvirtual PT-handle to a c_b. H is not a resource path.*

Proof. Assume contrarily that the n_s (n_e) of H (recall Definition 9) has two output (input) resource places against the fact that any state place can use and release only one resource. □

COROLLARY 1. *The resource subnet (Definition 8) of any I_S in an S^3PR is a state machine (SM).*

Proof. Lemma 3 indicates that all resource handles to a c_b must be PP-handles. The c_b plus these PP-handles is an SM. □

Upon a strongly connected resource subnet N', we can add nonresource TP- and PP-handles to form an I_S. Since $\rho(r)$ is a siphon for each $r \in N'$, so is the union δ of all such $\rho(r)$. Deleting nonvirtual PT- and TT-handles from δ forms the I_S. The rest are PP- or TP-handles which are parts of circuits of some $I(\rho(r))$. Note that each circuit of an $I(\rho(r))$ contains exactly one state place.

Let R be the set of their resource places in the net N, $T_u = \bullet R \cap R\bullet$ and N_u the net generated by R and T_u. To find all SMS, we have to locate all possible N' as in algorithm 1.

(1) Find all strongly connected components (SCC) N'' in N_u in linear time using the algorithm by Tarjan [23]. If not found, then exit since no SMS exists.
(2) For each N'' (an SM), find all its strongly connected subcomponents, called sub-SCC υ. For each such sub-SCC, add all nonresource TP- and PP-handles to form an I_S. $P(I_S)$ is an SMS.

ALGORITHM 1. SMS computation.

To the best of our knowledge, no efficient algorithms exist to find all sub-SCC (future work). Since the number of sub-SCC is exponential to the size of the net, the time complexity is at least exponential. Hence we **will consider the simple case where** $c_{b1} \cap c_{b2} = \{r\}$, $r \in R$, whenever a dependent siphon can be constructed from $c_{b1} \cup c_{b2}$ for any pair of c_{b1} and c_{b2}.

Note that if two different υ_1, υ_2 have the same set of resource places, i.e. $R(\upsilon_1) = R(\upsilon_2)$, then they will synthesize the same SMS.

DEFINITION 11. *A simple S^3PR (S^4PR) is defined to be* (1) *an S^3PR and* (2) $\forall c_{b1} \in C_B, c_{b2} \in C_B$, *such that* $c_{b1} \cap c_{b2} \neq \Phi$, *if neither $R_{b1} \subseteq R_{b2}$ nor $R_{b2} \subseteq R_{b1}$ holds, then $c_{b1} \cap c_{b2} = \{r\}$, $r \in R$, where C_B is the set of all c_b in the S^3PR.*

Note that if either $R_{b1} \subseteq R_{b2}$ or $R_{b2} \subseteq R_{b1}$ holds, then $c_{b1} \cup c_{b2} = c_{b2}$ or c_{b1}, and $c_{b1} \cup c_{b2}$ results in no new SMS. The net in Figure 1 is an S^4PR, where $R_{b1} \subseteq R_{b4}, R_{b10} \subseteq R_{b4}, R_{b16} \subseteq R_{b4}$ and $R_{b18} \subseteq R_{b17}$. Also for the rest, if $c_{b1} \cap c_{b2} \neq \Phi$, we have $c_{b1} \cap c_{b2} = \{p_{21}\}$ (e.g. $c_{b4} \cap c_{b17} = c_{b4} \cap c_{b18} = c_{b10} \cap c_{b18} = c_{b16} \cap c_{b18} = c_{b10} \cap c_{b16} = \{p_{21}\}$), except that $c_{b1} \cap c_{b16} = \{p_{26}\}$. We will show in Theorem 3 that for an S^4PR the sets of elementary siphons and basic siphons are identical. Thus, we only need to find the c_b for elementary (also basic) siphons. Since all places in a c_b are resources, we can remove all state places and their incident arcs in N and apply well-known algorithms [24] to search elementary directed circuits. We will present a method to construct a characteristic T-vector followed by an algorithm to construct a basic siphon given a c_b.

In the worst case of densely connected resource subnets, any subset of more than one resource place contributes to a basic siphon. Thus, the total number of basic siphons equals $\omega(k) = C(k, 2) + C(k, 3) + \cdots + C(k, k) = 2^k - k - 1$, where $k = O(|R|)$ and $C(k, h)$ means the number of ways to select h items out of k items with no distinction. For real FMS applications, however, it is loosely connected, and the total number of basic siphons is a polynomial with respect to $|R|$. Since it takes $O(|F|+|T|+|R|)$ time [25] to find a circuit, the total time to find all basic siphons is polynomial.

Further, the constraint for S^4PR limits the number of eligible c_b to be a polynomial with respect to $O(|F|+|T|+|R|)$ if $\forall c_{b1} \in C_B$, $c_{b2} \in C_B$, neither $R_{b1} \subseteq R_{b2}$ nor $R_{b2} \subseteq R_{b1}$ holds. This is because, to be exponential, at least one arc or directed path must be shared by more than one c_b—in contradiction to the definition of S^4PR.

On the other hand, if it is possible that either $R_{b1} \subseteq R_{b2}$ or $R_{b2} \subseteq R_{b1}$ holds, then in the worst case, there are $w(k) - \sum_{i=2}^{k-1} B(k, i)$ basic siphons where $B(k, i)$ is the number of basic siphons that should be deleted from $\omega(k)$ since they share $i > 1$ resource places and are not subset of each other (see Definition 11). Note that $B(k, i)$ is $O(k^2)$ and $\sum_{i=2}^{k-1} B(k, i)$ is $O(k^3)$. Thus, the total number of basic siphons remains to be exponential in the worst case.

For large nets, computing a basic siphon may not be easy. The concept of characteristic T-vector will be presented in the next section to compute basic siphons in an algebraic way.

4. ELEMENTARY SIPHONS AND CHARACTERISTIC *T*-VECTORS

This section defines elementary, dependent siphons and characteristic T-vectors respectively.

DEFINITION 12. [18]. *Let $\Omega \subseteq P$ be a subset of places of N. P-vector λ_Ω is called the characteristic P-vector of Ω iff* $\forall p \in \Omega$, $\lambda_\Omega(p) = 1$; *otherwise* $\lambda_\Omega(p) = 0$.

DEFINITION 13. [18]. *Let $\Omega \subseteq P$ be a subset of places of N, and λ_Ω the characteristic P-vector of Ω. η is called the characteristic T-vector of Ω, if $\eta^T = \lambda_\Omega^T \bullet A$, where A is the incidence matrix.*

Note that for the net in Figure 1, **each η has 20 components, which is too long. Instead, we ignore all zero components, and if $\eta(t_x) = 1$ ($\eta(t_x) = -1$), we replace** 1 (-1) **with $+t_x$ ($-t_x$).**

Physically, the sets where $\eta > 0$, $\eta = 0$ and $\eta < 0$ are the sets of transitions whose firings increase, maintain and decrease the number of tokens in Ω respectively.

DEFINITION 14. [18]. ($S_0 \in \Pi$ (*the set of all S*), *if* $\nexists S_1, S_2, \ldots, S_n \in \Pi$ ($\forall i \in \{1,2,\ldots,n\}$, $S_0 \neq S_i$) *such that* $\eta_0 = a_1{}^* \eta_1 + a_2{}^* \eta_2 + \cdots + a_n{}^* \eta_n$ *holds, where* $a_1, a_2, \ldots, a_n \in$ Я, *the set of all real numbers*, *then S_0 is called an elementary siphon of net N. Π_E denotes the set of all S_0.*

DEFINITION 15. [25]. *Let $S_0 \in \Pi \backslash \Pi_E$ be a siphon in a net N and $S_1, S_2, \ldots, S_n$ be its elementary siphons. S_0 is called a dependent siphon with respect to $S_1, S_2, \ldots, S_n$, and if* $\eta_0 = a_1{}^* \eta_1 + a_2{}^* \eta_2 + \cdots + a_n{}^* \eta_n$ *holds, where* $a_1, a_2, \ldots, a_n \in$ Я, *the set of all real numbers. η_0 is said to touch $\eta_1, \eta_2, \ldots, \eta_n$. S_0 is called a strongly dependent siphon* [25] *if* $\forall i \in \{1, 2, \ldots, n\}$, $a_i > 0$. *S_0 is called a weakly dependent siphon if* $\exists i, j \in \{1, 2, \ldots, n\}$, $a_i > 0$ *and* $a_j < 0$.

Li and zhou [18] proposed to find elementary siphons based on all SMS. First, they construct the characteristic $P(T)$-vector matrix $[\lambda]$ $([\eta])$ **of the siphons in N followed by finding linearly independent vectors in** $[\lambda]$ $([\eta])$**. Note that**

TABLE 1. Elementary siphons and their η for the net in Figure 1.

Elementary SMS	Places	η
S_1	$p_{10}, p_{18}, p_{22}, p_{26}$	$[-t_9 + t_{10} - t_{15} + t_{16}]$
S_4	$p_4, p_{10}, p_{17}, p_{21}, p_{22}, p_{24}, p_{26}$	$[-t_3 + t_5 - t_8 + t_{10} - t_{11} + t_{13} - t_{15} + t_{17}]$
S_{10}	$p_4\ p_9, p_{12}, p_{17}, p_{21}, p_{24}$	$[-t_3 + t_4 - t_{11} + t_{13}]$
S_{16}	$p_2, p_4, p_8, p_{13}, p_{17}, p_{21}, p_{26}$	$[+t_9 - t_8 + t_{17} - t_{16}]$
S_{17}	$p_2, p_4, p_8, p_{12}, p_{15}, p_{20}, p_{21}, p_{23}, p_{25}$	$[-t_1 + t_3 + t_8 - t_{17} + t_{19}]$
S_{18}	$p_2, p_4, p_8, p_{12}, p_{16}, p_{21}, p_{25}$	$[-t_7 + t_8 - t_{17} + t_{18}]$

TABLE 2. Dependent siphons and their η for the net in Figure 1

Dependent siphons	Places	η relationship
S_2	$p_4, p_{10}, p_{15}, p_{20}, p_{21}, p_{22}, p_{23}, p_{24}, p_{25}, p_{26}$	$\eta_2 = \eta_4 + \eta_{17}$
S_3	$p_4, p_{10}, p_{16}, p_{21}, p_{22}, p_{24}, p_{25}, p_{26}$	$\eta_3 = \eta_4 + \eta_{18}$
S_5	$p_4, p_9, p_{13}, p_{15}, p_{20}, p_{21}, p_{23}, p_{24}, p_{25}, p_{26}$	$\eta_5 = \eta_{10} + \eta_{16} + \eta_{17}$
S_6	$p_4, p_9, p_{13}, p_{16}, p_{21}, p_{24}, p_{25}, p_{26}$	$\eta_6 = \eta_{10} + \eta_{16} + \eta_{18}$
S_7	$p_4, p_9, p_{13}, p_{17}, p_{21}, p_{24}, p_{26}$	$\eta_7 = \eta_{10} + \eta_{16}$
S_8	$p_4, p_9, p_{12}, p_{15}, p_{20}, p_{21}, p_{23}, p_{24}, p_{25}$	$\eta_8 = \eta_{10} + \eta_{17}$
S_9	$p_4, p_9, p_{12}, p_{16}, p_{21}, p_{24}, p_{25}$	$\eta_9 = \eta_{10} + \eta_{18}$
S_{11}	$p_2, p_4, p_8, p_{10}, p_{15}, p_{20}, p_{21}, p_{22}, p_{23}, p_{25}, p_{26}$	$\eta_{11} = \eta_1 + \eta_{16} + \eta_{17}$
S_{12}	$p_2, p_4, p_8, p_{13}, p_{15}, p_{20}, p_{21}, p_{23}, p_{25}, p_{26}$	$\eta_{12} = \eta_{16} + \eta_{17}$
S_{13}	$p_2, p_4, p_8, p_{10}, p_{16}, p_{21}, p_{22}, p_{25}, p_{26}$	$\eta_{13} = \eta_1 + \eta_{16} + \eta_{18}$
S_{14}	$p_2, p_4, p_8, p_{13}, p_{16}, p_{21}, p_{25}, p_{26}$	$\eta_{14} = \eta_{16} + \eta_{18}$
S_{15}	$p_2, p_4, p_8, p_{10}, p_{17}, p_{21}, p_{22}, p_{26}$	$\eta_{15} = \eta_1 + \eta_{16}$

$[\lambda] = [\lambda_1|\lambda_2| \cdots |\lambda_k]^T$ **and** $[\eta] = [\eta_1|\eta_2| \cdots |\eta_k]^T$, $k = |\Pi|$. **Finally, the siphons corresponding to these independent vectors are the elementary siphons in the net system. But they have to look for all SMS. We find elementary siphons without the knowledge of all SMS.**

Tables 1 and 2 show all the elementary and dependent siphons and their η for the net in Figure 1 respectively.

Similarly, we propose ζ_R based on the set of resource places in S.

DEFINITION 16. *Let $R \subseteq P$ be a subset of resource places of N and λ_R the characteristic P-vector of R. ζ_R is called the characteristic T-vector for R, if $\zeta_R^T = \lambda_R^T \bullet A$.*

Physically, the sets where $\zeta_R > 0$, $\zeta_R = 0$ and $\zeta_R < 0$ are the sets of transitions whose firings increase, maintain and decrease the number of tokens in R respectively. Note that ζ_R is a linear sum of all ζ_r as in the following lemma.

TABLE 3. Elementary siphons and their ζ for the net in Figure 1.

SMS	Places in c_b, P_b	ζ_R
1. S_1	R3, M4	$[-t_5 + t_6 - t_9 - t_{15} + t_{17}]$
2. S_4	R3, M2, R2, M4	$[-t_3 + t_6 - t_8 - t_{11} + t_{14} - t_{15} + t_{18}]$
3. S_{10}	R2, M2	$[-t_3 + t_5 - t_8 + t_9 - t_{11} + t_{14} - t_{17} + t_{18}]$
4. S_{16}	R2, M4	$[-t_3 + t_4 - t_8 + t_{10} - t_{11} + t_{12} - t_{13} + t_{14} - t_{16} + t_{18}]$
5. S_{17}	R1, R2, M1, M3	$[-t_1 + t_4 + t_9 - t_{11} + t_{12} - t_{13} + t_{14} - t_{17} + t_{20}]$
6. S_{18}	R2, M3	$[-t_3 + t_4 - t_7 + t_9 - t_{11} + t_{12} - t_{13} + t_{14} - t_{17} + t_{19}]$

LEMMA 4. *$\zeta_R^T = \Sigma_{r\in R}\zeta_r^T$ where R is the set of all resource places in a resource subnet of an S^3PR.*

Proof. It holds by Definition 16. $\zeta_R^T = \lambda_R^T \bullet A = \Sigma_{r\in R}(\lambda_r^T \bullet A) = \Sigma_{r\in R}\zeta_r^T$. □

Note that $\forall t \in \bullet r$ ($r\bullet$), $\zeta_r(t) = 1$ (-1). For instance, for S_4 in Figure 1, $\zeta_R = \Sigma_{r\in R}\zeta_r = [-t_3 + t_6 - t_8 - t_{11} + t_{14} - t_{15} + t_{18}]$. The rest are in Table 3.

Thus, it is easy to compute ζ for elementary siphons. But we need to compute SMS from ζ and the computed SMS may not be minimal, the condition of which is unclear. We also need to find dependent siphons from elementary siphons.

We observe that the number of basic siphon circuits is six, the same as that of elementary siphons. Actually we can construct the same set of elementary siphons from these circuits (Theorem 3). The next section will present the algorithm to compute elementary siphons.

5. ALGORITHM OF COMPUTATION OF ELEMENTARY SIPHONS

The algorithm is based on Lemma 4. Just as the η for a dependent SMS is a linear combination of that of some elementary siphons, so is ζ_R of ζ_r of resource places in the circuit. The following lemma helps to understand Lemmas 7 and 8.

LEMMA 5. *Let $\boldsymbol{P_b} = \boldsymbol{P}(\boldsymbol{c_b})$ be the set of places in c_b, ζ the characteristic T-vector of the basic siphon and $\boldsymbol{T_b}' = \bullet\boldsymbol{P_b} \cap \boldsymbol{P_b}\bullet$. (1) $\forall t \in T_b'$, $\zeta(t) = 0$; (2) $t \in P_b\bullet\backslash T_b'$, iff $\zeta(t) = -1$; (3) $t \in \bullet P_b\backslash T_b'$, iff $\zeta(t) = 1$.*

Proof. There is at most one input (output) place, say r (r'), of t in $\boldsymbol{P_b}$ due to the fact that any state place can use and release only one resource. Four possibilities for $t \in T$: (a) exactly one input (output) place, (b) zero input and one output place, (c) one input and zero output place and (d) no input (output) place in $\boldsymbol{P_b}$. $\zeta(t) = 0$ for Cases (a) and (d). $\zeta(t) = 1$ (-1) for Case (b) (c). (1) $\forall t \in T_b'$, Case (a) holds; there is exactly one input (output) place, say r (r'), of t in $\boldsymbol{P_b}$. We have $\zeta_r(t) = 1$ $(\zeta_{r'}(t) = -1)$. Hence $\zeta(t) = \zeta_r(t) + \zeta_{r'}(t) = 0$. (2) $(\rightarrow)$ If $t \in P_b\bullet\backslash T_b'$, Case (c) holds. Hence, $\zeta(t) = -1$. $(\leftarrow)$ If $\zeta(t) = -1$, Case (c) holds and there is exactly one input and zero output place in $\boldsymbol{P_b}$. Hence $t \in P_b\bullet\backslash T_b'$. (3) $(\rightarrow)$ If $t \in \bullet P_b\backslash T_b'$, Case (b) holds; hence, $\zeta(t) = 1$. $(\leftarrow)$ If $\zeta(t) = 1$, Case (b) holds and there is exactly one input and zero output place in $\boldsymbol{P_b}$. Hence $t \in \bullet P_b\backslash T_b'$. □

The following lemma shows the relationship between handles to a c_b and the nonzero components of ζ.

LEMMA 6. *Let H be a handle to c_b and $t \in H$. (1) If $\zeta(t) = -1$, H is a PT- or PP-handle. (2) If $\zeta(t) = 1$, H is a TP- or PP-handle.*

Proof. (1) If $\zeta(t) = -1$, then $t \in P_b\bullet$. $\forall p \in t\bullet$, $p \notin P_b$. Hence p is a state place. Let $p \in \rho(r)$ where $r \in P_b$ and $t' \in \rho(r) \cap p\bullet$. If $\zeta(t') = 1$, then H is a PP-handle. If $\zeta(t') = 0$, then H is a PT-handle. (2) If $\zeta(t) = 1$, then $t \in \bullet P_b$. $\forall p \in \bullet t$, $p \notin P_b$. Hence p is a state place. Let $p \in \rho(r)$, $r \in P_b$ and $t' \in \rho(r) \cap \bullet p$. If $\zeta(t') = -1$, then H is a PP-handle. If $\zeta(t') = 0$, then H is a TP-handle. □

Similarly, we have the relationship between handles to a c_b and the nonzero components of η in the following lemma:

LEMMA 7. *Let H be a handle to c_b and $t \in H$. (1) If $\eta(t) = -1$, H is a PT-handle. (2) If $\eta(t) = 1$, H is a TP-handle.*

Proof. (1) If $\eta(t) = -1$, then $t \in P_b\bullet$. $\forall p \in t\bullet$, $p \notin P_b$. Hence p is a state place. Let $p \in \rho(r)$, $r \in P_b$. If H is a PP-handle, $\eta(t) = 0$. The firing of t removes tokens from S, but they will return to S by firing $t' \in \rho(r) \cap p\bullet$. Hence, H must be a PT-handle; the firing of t decreases the number of tokens in S. (2) If $\eta(t) = 1$, then $t \in \bullet P_b$. $\forall p \in \bullet t$, $p \notin P_b$. Hence p is a state place. If H is a PP-handle, $\eta(t) = 0$. The firing of t increases tokens in S, but they will leave S by firing $t' \in \rho(r) \cap \bullet p$. Hence, H must be a TP-handle; the firing of t increases the number of tokens in S. □

LEMMA 8. *Let $\boldsymbol{P_b} = \boldsymbol{P}(\boldsymbol{c_b})$ and $\boldsymbol{T_b}'$ be defined in Lemma 5. (1) $\forall t \in \bullet P_b\backslash T_b'$, $\eta(t) = 0$. (2) $\forall t \in P_b\bullet\backslash T_b'$, (2.a) if t is on a PP-handle, $\eta(t) = 0$; (2.b) otherwise, $\eta(t) = -1$. (3) $\forall t \in T_b'$, (3.a) if t is on a TT-handle, $\eta(t) = 0$; (3.b) otherwise, it is on a TP-handle, $\eta(t) = 1$.*

Proof. It follows from the proof of Lemmas 5–7. □

Upon the detection of an elementary circuit, we can construct a siphon based on Lemmas 4–6: the computed S may not be minimal if it contains a $\rho(r)$ to be detected by the following lemma.

LEMMA 9. *Let $\zeta_R = \Sigma_{r\in R}\zeta_r$ where $R = P(c_b)$. Let $S = R \cup \{p \mid \zeta_r(t) = 1, p \in \bullet t \cap H(r)\}$. (1) $S = P(I)$ where I is the I-subnet built using the handle-construction procedure. (2) S is minimal iff $\neg(\forall p \in r\bullet\bullet\backslash R$ (p is a state place), $\exists t'$, t'', $\zeta(t') = 1$, $\zeta(t'') = 0$, t', $t'' \in p\bullet$ (i.e. $\exists$ a virtual PT-handle in I)).*

Proof. (1) It follows from Lemma 6. (2) As shown in [10], any minimal siphon in N_i is a $\rho(r)$, $r \in R$. $(\rightarrow)$ By Lemma 6, $[r\ t\ p\ t'\ r]$ is a PP-circuit to N_i and is in I by Lemma 1. I now contains $\rho(r)$ and it is not minimal—contradiction. $(\leftarrow)$ It implies $p \notin S$ (a nonvirtual PT-handle $[r\ t\ p\ t']$ to N_i); hence, S does not contain any $\rho(r)$ and is minimal. □

In Figure 1, let $c = [p_{20}\ t_{19}\ p_{25}\ t_7\ p_{20}]$, $r = p_{20}$, $t_1 \in p_{20}\bullet$. $\zeta(t_1) = -1$. $p_6 \in t_1\bullet$, $p_6 \notin c$. $p_6\bullet = \{t_2, t_7\}$. $\zeta(t' = t_2) = 1$, $\zeta(t_7) = 0$. Hence the computed S $= \{p_{20}, p_{25}, p_6, p_{11}, p_{15}\}$ is not minimal.

We now have

(1) $S = P(c_b)$
(2) $\zeta_R = \Sigma_{r\in R}\zeta_r$ where $R = P(c_b)$
(3) $\forall r \in c_b$, if $\forall p \in r\bullet\bullet\backslash R$ (p is a state place), $\exists t'$, t'', $\zeta(t') = 1$, $\zeta(t'') = 0$, t', $t'' \in p\bullet$, the SMS if computed, will not be minimal; break and exit.
(4) $\forall t''$, $\zeta(t'') = 1$, $p \in \bullet t''$, $p \notin c_b$, $S \cup \{p\} \rightarrow$ S (add p to S).

ALGORITHM 2. Basic-siphon computation based on ζ.

Although Algorithm 2 computes elementary siphons, it can also compute dependent siphons if we replace the c_b in the algorithm by a resource subnet.

Step 1 first sets S to be the set of all resource places in c_b taking $O(|R|)$ time. Then it computes ζ_R in Step 2. Because a holder can be used by only one resource place (constraints in Definitions 7.3 and 7.4.a), Step 2 takes O(|P|+|R|) time. Step 3 tests to see if it is minimal. It takes O(|P|+|R|) time to detect $\rho(r)$. Hence the time complexity for Step 3 is also $O(|P|+|R|)$. The rest of S are in TP- or PP-handles (Step 4) as in Lemma 6. Step 4 takes $O(|R|)$ time owing to the constraints in Definitions 7.3 and 7.4.a.

We will show in the next section that the set of elementary siphons equals that of basic siphons (Theorem 3) and the characteristic T-vectors for dependent siphons can be constructed by building a graph without their computations. Algorithm 2 helps to compute elementary siphons from c_b.

The following theorem is helpful to prove Theorem 3.

THEOREM 1. *For every sub-SCC N^* made of c_{b1} and c_{b2} in an S^4PR (defined earlier) corresponding to an SMS S^*, $\eta^* = \eta_1 + \eta_2$ iff $c_{b1} \cap c_{b2} = \{r\}$, where $r \in R$ and η^* is the η value for S^*.*

Proof. First, we explore the conditions under which $\eta^* = \eta_1 + \eta_2$ holds for all possible cases where $c_{b1} \cap c_{b2} \neq \Phi$. Let ϑ_1 (ϑ_2) be the set of transitions in all nonresource PP-, TP-,

TABLE 4. η_1, η_2, η_* and the testing $\eta_1 + \eta_2 = \eta_*$? for Case T_4.

Case	η_1	η_2	η_*	$\eta_1 + \eta_2 = \eta_*$?	$t \in H$. H unchanged in the union?
1	1_{tp}	1_{tp}	1_{tp}	no	yes
2	-1_{pt}	-1_{pt}	-1_{pt}	no	yes
3	0_{pp}	0_{pp}	0_{pp}	yes	yes
4	0_{tp}	0_{tp}	0_{tp}	yes	yes
5	0_{tt}	0_{tt}	0_{tt}	yes	yes
6	1_{tp}	-1_{pt}	0_{tt}	yes	no
7	0_{pp}	0_{tt}	0_{tt}	yes	no
8	0_{pp}	0_{tp}	0_{tp}	yes	no
9	0_{pp}	1_{tp}	1_{tp}	yes	no
10	0_{pp}	-1_{pt}	-1_{pt}	yes	no
11	1_{tp}	0_{tt}	0_{tt}	no	no
12	-1_{pt}	0_{tt}	0_{tt}	no	no

PT- and TT-handles to c_{b1} (c_{b2}). $T = T_1 + T_2 + T_3 + T_4$ where T_1 is the set of transitions neither in ϑ_1 nor in ϑ_2, T_2 that in $\vartheta_1 \backslash \vartheta_2$, T_3 that in $\vartheta_2 \backslash \vartheta_1$ and T_4 that in $\vartheta_1 \cap \vartheta_2$. $\forall t \in T_1$, $\eta_1(t) = \eta_2(t) = \eta^*(t) = 0$. $\forall t \in T_2$, $\eta^*(t) = \eta_1(t)$ and $\eta_2(t) = 0$. $\forall t \in T_3$, $\eta^*(t) = \eta_2(t)$ and $\eta_1(t) = 0$. All satisfy $\eta^* = \eta_1 + \eta_2$.

Case T_4 is discussed based on Table 4.

Column 6 '$t \in H$. H unchanged in the union' means if t is in H, H is a (same type) handle to both c_b and their union. H cannot be, say, PP-handle to c_{b1}, but a different type, say TP-handle, to c_{b2} or $c_{b1} \cup c_{b2}$. This is the reason that if any entry in Column 6 is yes (Cases 1–5), $\eta_1 = \eta_2$. Let $N' = I_1 \cap I_2$. t is either in N' or on a handle H to N', where $I_1(I_2)$ is the I-subnet of the SMS built on $c_{b1}(c_{b2})$.

The values for the rest entries are 'no' (Cases 6–10) and come from the fact that part of a TP- (or PT-) handle may become a TT-handle and part of a PP-handle may become a TP- (or TT-) handle.

Note that if $\eta_1 + \eta_2 \neq \eta^*$, then the SMS for $c_{b1} \cup c_{b2}$ is elementary, but not a basic siphon.

For Cases 1–5, $\eta_1 + \eta_2 = \eta^*$ holds for only Cases 3–5 where $\eta_1 = \eta_2 = 0$ and $\forall t \in T_{b1}' \cap T_{b2}'$ (see Lemma 5 for definition of T_b'), it must be on a TT-handle corresponding to entry '0_{tt}' in the table. In order to be on a TT-handle, $t' \in t\bullet\bullet \cap N'$ (N' defined earlier in this lemma) must be on the same subprocess (Definition 6) as t. This implies that the R (e.g. $R = \{R2, M3, R4\}$) of N^* must extend between two adjacent WP such as WP_{12} and WP_3 in Figure 1.

However, at the terminal r nodes of N' where part of a TP- (and PT-) handle becomes a TT-handle, Cases 11 and 12 apply and $\eta_1 + \eta_2 \neq \eta^*$. Such a problem will not occur *iff* $c_{b1} \cap c_{b2} = \{r\}$. That is it must be an S^4PR where Cases 1, 2, 4 and 5 cannot happen either. Thus, if $\eta^* = \eta_1 + \eta_2$, then $c_{b1} \cap c_{b2} = \{r\}$ and if $c_{b1} \cap c_{b2} = \{r\}$, then $\eta^* = \eta_1 + \eta_2$. □

COROLLARY 2. *For every sub-SCC N^* made of N_1 and N_2 in an S^4PR (defined earlier) corresponding to an SMS S^*, $\eta^* = \eta_1 + \eta_2$ iff $N_1 \cap N_2 = \{r\}$, where $r \in R$ and η^* (η_1, η_2) is the η value for S^* (S_1, S_2) which is the SMS corresponding to N^* (N_1, N_2).*

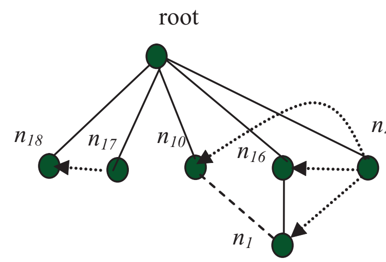

FIGURE 2. The c_b-graph to compute all dependent SMS.

Proof. It follows by a similar reasoning to that for Theorem 1. □

We explain Cases 6, 7 and 12 where $t \in T_4$. Case 6: $\eta_1(t) = 1$. By Lemma 7, t is on a TP-handle [t p t' r] to c_{b1}. Due to the constraint that every state place can use and release exactly one resource, t must be in a PT-handle [r t p t'] to c_{b2}, $\eta_2(t) = -1$. t is also in a TT-handle [t p t'] to $c = c_{b1} \cup c_{b2}$; hence $\eta_1 + \eta_2 = \eta^* = 0$. Case 7: $\eta_2(t) = 0$. t is on a TT-handle [t p t']. t and t' are on c_{b2} and in a PP-circuit [r t p t' r] (must be a circuit) to c_{b1}. Hence $\eta_2 = 0$. t remains to be in a TT-handle to c. Thus we have $\eta_1 + \eta_2 = \eta^* = 0$. As an example, $\eta_{16} + \eta_{18} = \eta_{14}$. Case 12: $\eta_2(t) = 0$. t is on a TT-handle [t p t']. t and t' are on c_{b2} and in a PT-circuit [r t p t'] to c_{b1}. Hence $\eta_2 = -1$. t remains to be in a TT-handle to c. Thus we have $\eta_1 + \eta_2 \neq \eta^*$. Cases 8–11 are similar.

6. COMPUTATION OF DEPENDENT SIPHONS OF S^4PR

Recall that each SMS can be constructed from a sub-SCC. After locating all c_b, we may condense the resource subnet N_u by building a graph with a set of nodes and edges (Figure 2). Each node represents a c_b or a dummy node (e.g. root). Output nodes (e.g. for S_{18}, S_{17}, S_{10}, S_{16} and S_4) of a dummy node (root) represent those c_b sharing the same set of resource places ($\{p_{21}\}$). There is a dashed and directed arc from n_h to n_k to indicate $R_{bh} \supset R_{bk}$ (Rule 4 in Definition 16) implying the constraint that η_h and η_k never appear together in (or touched by) any η. The problem then is equivalent to finding all connected components in the graph while observing the constraint and others mentioned later. For instance, path n_{10}-root-n_{16} in Figure 2 is a connected component and indicates a dependent siphon whose $\eta = \eta_{10} + \eta_{16}$.

DEFINITION 17. *A c_b-graph $\hat{G}$ is a graph built using the following rules*: (1) *any node is either a dummy node or a node n_i for c_{bi}*; (2) *$\forall c_{bi}$, there is a unique $n_i \in \hat{G}$*; (3) *$\forall$ (c_{bi}, c_{bj}), $R_{bi} \cap R_{bj} \neq \Phi$ and neither $R_{b1} \subseteq R_{b2}$ nor $R_{b2} \subseteq R_{b1}$, there is a solid edge between n_i and n_j*, (4) *if $R_{bi} \supset R_{bj}$, the edge between*

n_i and n_j is dashed and directed from n_i to n_j; (5) $\forall$ (c_{b1}, $c_{b2}, \ldots, c_{bk}$), *if* $R_{b1} \cap R_{b2} \cdots \cap R_{bk} \neq \Phi$, *then* $n_1, n_2, \ldots, n_k$ *are output nodes of a dummy node*; (6) $\forall(c_{b1}, c_{b2}, \ldots, c_{bk})$, *if* $n_1{-}d_1{-}n_2{-}d_2 \ldots {-}d_k{-}n_k$, *where* d_i, $1 \leq i \leq k$, *is a null or dummy node*, *and* $R_{b1} \cup R_{b2} \cdots \cup R_{bk} = R_{bq}$, $q \notin \{1, 2, \ldots k\}$ *then add a dashed edge between* n_1 *and* n_k.

Condition 6 is to avoid generating a dependent siphon which actually is a basic siphon. We call such a generation *nonunique*:

We construct a dependent siphon using Theorem 2 based on the following:

DEFINITION 18. *An SMS S is said to be generated by* c_{b1}, $c_{b2}, \ldots, c_{bk}$ *if S is created by the handle-construction process upon a strongly connected resource subnet containing* c_{b1}, $c_{b2}, \ldots, c_{bk}$. *It is called a unique generation if* (1) *no proper subset of* c_{b1}, $c_{b2}, \ldots, c_{bk}$ *generates the same S and* (2) *S is not a basic siphon.*

Counter examples: (1) the union of the set of resource places in the c_b for S_1, S_{10}, S_{16} equals that for S_4; i.e. $R_{b1} \cup R_{b10} \cup R_{b16} = R_{b4}$. The union of these c_b generates S_4—neither new nor dependent and is not a unique generation. In column 3 of Table 2, S_{18} (one of S_1, S_{10} and S_{16}) and S_{17} (S_4) never appear together. To avoid such, we create a dashed edge between n_{18} (one of n_1, n_{10} and n_{16}) and n_{17} (n_4) to forbid it. In the graph, a path (n_{10}-root-n_{16}-n_1) covers the three nodes for S_1, S_{10}, S_{16} and would normally create a new dependent siphon. To avoid such, we create a dashed edge between n_1 and n_{10} to forbid it.

Further, a circuit with all edges solid may not exist in the graph. Otherwise, a new c_b and hence S can be formed from the set of resources in the circuit and hence it is not a unique generation. Thus, combinations of c_b on the circuit should be avoided. We may break the circuit by making one edge dashed (found by locating a tree from the solid subgraph). In any case, the presence of a dashed edge forbids the associated combination.

THEOREM 2. *For an* S^4PR (*defined earlier*), *the unique generation by* c_{b1}, $c_{b2}, \ldots, c_{bk}$ *creates a strongly dependent siphon* S^* *and* $\eta^* = \eta_1 + \eta_2 + \cdots + \eta_k$.

Proof: Prove by induction. We first prove the case for $k = 2$. By Theorem 1, $\eta^* = \eta_1 + \eta_2$.

Now assume it holds for $n = k - 1$. Replace c_{b1} by c_{bk} and c_{b2} by c_{bi}, $i \in \{1, 2, \ldots k - 1\}$ in the proof of Theorem 1 (i.e. n_k is an output node of n_i), we again have $\eta^* = \eta^{k-1*} + \eta_k = \eta_1 + \eta_2 + \cdots + \eta_k$. It holds for $n = k$ because for each case in the proof of Theorem 1, exactly two c_b (c_{bk} and c_{bi}) are involved due to the constraint that every state place can use and release exactly one resource. Otherwise, if n_k is an output node of another n_j, $j \in \{1, 2, \ldots k - 1\}$ and $j \neq i$, then a circuit with all edges solid exists containing n_i, n_j, n_k in the graph—forbidden. It is strongly dependent since all the coefficients in the expression for η^* are positive (see Definition 15). □

Starting from the root, perform a breadth-first search. Each time we reach a new node n_d from n_c, we create a new $\eta = \eta_v + \eta_d$ for each η_v in list l_c which includes all η touching (Definition 14) η_c. In the case of a dashed edge between n_d and a traced node n_t, delete all creations where η_v touches both η_c and η_t corresponding to a path from n_t to n_c.

The above method generates all dependent siphons based on the idea that any dependent siphon corresponds to a sub-SCC made of multiple interconnected c_b in the resource subnet (see Algorithm 1). The above method constructs a graph where each c_b corresponds to a node and each connected component μ (with no dashed arcs between nodes in μ) in the graph corresponds to a sub-SCC and a dependent siphon. Conversely, each sub-SCC made of multiple interconnected c_b corresponds to a μ. Thus, it generates all dependent siphons since each of them can be computed using Algorithm 2 from the corresponding sub-SCC.

Note that in the worst case, the number β of dependent siphons is $O(2^h)$, where h is the number of nodes or basic siphons in the above graph. If h is $O(\log_2|R|^f)$ (f is a constant), then $\beta = O(|R|^f)$ is polynomial. And the above method takes a polynomial time (unlike other methods) with respect to the size of the resource subnet.

Finally, we have the following important result.

THEOREM 3. *For an* S^4PR, *the set of elementary siphons equals that of basic siphons.*

Proof. The set of all basic siphons plus those constructed from multiple interconnected c_b form the set of all SMS. First, we show that any basic siphon is an elementary one. Suppose the η for any basic siphon can be a linear sum of those of other basic siphons, i.e. η is dependent. This implies the SMS constructed from a c_b equals that from a number of other c_b—contradiction. Next, we show that any elementary siphon S is a basic one. Assume contrarily that S corresponds to a sub-SCC made of multiple c_b, by Theorem 2, S is dependent—contradiction. □

7. CONCLUSION

We have proposed an algebraic approach to compute SMS by computing the new characteristic T-vector ζ for a c_b as a linear sum of ζ_r, $r \in c_b$. For S^4PR, a special subclass of S^3PR, the sets of elementary and basic siphons are identical and can be computed without the knowledge of all SMS. An elementary siphon includes all resource places in the circuit plus all input state places of transitions with positive components in ζ. We have also proposed to build a graph to construct characteristic T-vectors η for all dependent siphons without their computations.

Because only one resource is used in each job stage and the processes are modeled using state machines in S^3PR, its modeling power is limited. It cannot model iteration statements (loop) in each sequential process (SP) and the relationships of synchronization and communication among SP. At any state of a process, it cannot use multi-sets of resources. Future work will extend the technique to more complex nets than S^3PR and S^4PR.

REFERENCES

[1] Barkaoui, K. and Abdallah, I. B. (1996) An efficient deadlock avoidance control policy in FMS using structural analysis of Petri nets. In *Proc. IEEE Int. Conf. on System, Man, and Cybernetics*, San Antonio, TX, pp. 525–530. IEEE, Piscataway, NJ.

[2] Banaszak, Z. and Krogh, B. H. (1990) Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows. *IEEE Trans. Robot. Automat.*, **6**, 724–734.

[3] Lautenbach, K. and Ridder, H. (1996) *The linear algebra of deadlock avoidance—a Petri net approach*. University of Koblenz, Research Report, Institute for Computer Science, Koblenz, Germany.

[4] Hsien, F. S. and Chang, S. C. (1994) Dispatching-driven deadlock avoidance controller synthesis for flexible manufacturing systems. *IEEE Trans. Robot. Automat.*, **10**, 196–209.

[5] Xing, K. Y., Hu, B. S. and Chen, H. X. (1996) Deadlock avoidance policy for Petri-net modeling of flexible manufacturing systems with shared resources. *IEEE Trans. Automat. Contr.*, **42**, 289–296.

[6] Park, J. and Reveliotis, S. A. (2001) Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. *IEEE Trans. Automat. Contr.*, **46**, 1572–1583.

[7] Viswanadham, N., Narahari, Y. and Johnson, T. (1990) Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models. *IEEE Trans. Robot. Automat.*, **6**, 713–723.

[8] Huang, Y. S., Jeng, M., Xie, X. L. and Chung, S. L. (2001) A deadlock prevention policy for flexible manufacturing systems using siphons. In *Proc. IEEE Int. Conf. on Robotics Automation*, Seoul, Korea, May 21–26, pp. 541–546, IEEE, Piscataway, NJ.

[9] Barkaoui, K. and Abdallah, I. B. (1995) A deadlock prevention method for a class of FMS. In *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, Vancouver, BC, Canada, October 22–25, pp. 4119–4124, IEEE, Piscataway, NJ.

[10] Ezpeleta, J., Colom, J. M. and Martinez, J. (1995) A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Trans. Robot. Automat.*, **11**(2), 173–184.

[11] Huang, Y. S., Jeng, M. D., Xie, X. L. and Chung, S. L. (2001) Deadlock prevention policy based on Petri nets and siphons. *Int. J. Prod. Res.*, **39**(3), 283–305.

[12] Fanti, M. P. and Zhou, M. C. (2004) Deadlock control methods in automated manufacturing systems. *IEEE Trans. Syst., Man, Cybern., A: Syst. Humans*, **34**(1), 5–22.

[13] Zhou, M. C. and Fanti, M. P. (2005) *Deadlock Resolution in Computer Integrated Systems*. Marcel Dekker, New York, NY.

[14] Zhou, M. C. and DiCesare, F. (1993) *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Kluwer Academic Publishers, Boston, MA.

[15] Zhou, M. C., DiCesare, F. and Desrochers, A. A. (1992) A hybrid methodology for synthesis of Petri nets for manufacturing systems. *IEEE Trans. Robot. Automat.*, **8**(3), 350–361.

[16] Iordache, M. V., Moody, J. O. and Antsaklis, P. J. (2001) A method for the synthesis liveness enforcing supervisors in Petri nets. In *Proc. Amer. Control. Conf.* Arlington, Virginia, June 25–27, pp. 4943–4948. IEEE, Piscataway, NJ.

[17] Tricas, F. and Martinez, J. (1995) An extension of the liveness of concurrent sequential processes competing for shared resources. In *Proc. IEEE Int. Conf. on System, Man, and Cybernetics*, Vancouver, BC, Canada, October 22–25, pp. 4119–4124. IEEE, Piscataway, NJ.

[18] Li, Z. and Zhou, M. (2004) Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems. *IEEE Trans. Syst., Man, Cybern., A: Syst. Humans*, **34**(1), 38–51.

[19] Ezpeleta, J., Couvreur, J. M. and Silva, M. (1993) A new technique for finding a generating family of siphons, traps, and st-components: application to colored Petri nets. In Rozenberg, G. (ed.), *Advances in Petri Nets*, **674**, pp. 126–147. Springer-Verlag, New York.

[20] Lautenbach, K. (1987) Linear algebraic calculation of deadlocks and traps. In Voss, Genrich and Rozemberg (eds), *Concurrency and Nets*, pp. 315–336. Springer-Verlag, Berlin, Germany.

[21] Chao, D. Y. and Nicdao, J. A. (2001) Liveness for synchronized choice Petri nets. *Comput. J.*, **44**(2), 124–136.

[22] Chu, F. and Xie, X. L. (1997) Deadlock analysis of Petri nets using siphons and mathematical programming. *IEEE Trans. Robot. Automat.*, **13**(6), 793–804.

[23] Tarjan, R. (1972) Depth-first search and linear graph algorithms. *SIAM J. Comput.*, **1**, 146–160.

[24] Johnson, D. B. (1975) Finding all the elementary circuits of a directed graph. *SIAM J. Comput.*, **4**, 77–84.

[25] Cormen, T. H., Leiserson, C. E. and Rivest, R. L. (1992) *Introduction to Algorithms*. The MIT Press/MacGraw-Hill Book Company, Cambridge, MA.

[26] Li, Z. and Zhou, M. (2006) Clarifications on the definitions of elementary siphons in Petri nets. *IEEE Trans. Syst., Man, Cybern., A: Syst. Humans*, to appear.