Deadlock Avoidance for Free Choice Multi-Reentrant Flow lines: Critical Siphons & Critical Subsystems [1]

P. Ballal*, F. Lewis*, *Fellow IEEE*, J. Mireles**, *Member IEEE*, Jr., K. Sreenath*
*Automation & Robotics Research Institute, University of Texas at Arlington,
7300 Jack Newell Blvd. S., Fort Worth, TX 76118-7115, USA.
** Instituto de Ingeniería y Tecnología de la Universidad Autónoma de Ciudad Juárez,
Ave. del Charro 610 Nte. Juárez, Mexico.

***Abstract*— This paper has two contributions. First, we present an analysis of deadlock avoidance for a generalized case of Multi Reentrant Flow Line systems (MRF) called the Free Choice Multi Reentrant Flow Line systems (FMRF). In FMRF, some tasks have multiple resource choices; hence routing decisions have to be made and current results in deadlock avoidance do not hold. This analysis is based on the so-called Circular Waits (CW) of the resources in the system. For FMRF, the well known notions of Critical Siphons and Critical Subsystems must be generalized and we redefine these objects for such systems. Our second contribution provides a matrix formulation that efficiently computes the objects required for deadlock avoidance. A MAXWIP dispatching policy is formulated for deadlock avoidance in FMRF systems. According to this policy, deadlock in FMRF is avoided by limiting the work in progress (WIP) in the critical subsystems of each CW. A main contribution of this paper is a matrix formulation for direct and efficient computation of Petri Net objects.**

***Index Terms*—Deadlock Avoidance, Petri nets, Discrete Event Systems, MRF, FMRF, Intelligent Control.**

Notations

DES discrete event systems
MRF multi-reentrant flow lines
FMRF free choice multi-reentrant flow lines
CW circular wait
CB circular blocking
PN Petri Nets
WIP work in progress

I. Introduction

Resource assignment and task sequencing play important roles in applications involving mobile wireless sensor networks, manufacturing systems, and other decision resource systems. But, the use of shared resources in these discrete event systems (DES) creates major problems while sequencing tasks. If the assignment of the resources is not correctly made, serious problems might arise. Such problems include blocking and system deadlock [3, 9, 10, 17], which are dangerous situations that eventually stop all the activity in the flow line involved.

Several existing approaches in the literature are designed for the case of Multi Reentrant Flow Line systems (MRF) [3, 7, 10, 17], where resources are shared and can perform more than one task. Analysis of deadlock avoidance is well understood for MRF. An important aspect in deadlock avoidance strategies in MRF is the concept of a Circular Wait (CW) [16] among the resources. It has been shown in [3, 10, 16, 17] that deadlock occurs in MRF when blocking develops in a CW.

The analysis of shared resources becomes even harder when there are multiple choices of resources for a given task. This means routing decisions have to be made [14]. These systems are a generalized case of MRF systems called the Free Choice Multi-Reentrant Flow Lines (FMRF). In FMRF, more than one resource can perform same task in the system. In other words, some tasks in FMRF may not have predetermined resources assigned. This is the case, for instance, in wireless sensor networks [11]. In the case of FMRF, the known methods of deadlock avoidance provided for MRF do not work.

For analysis, modeling and control of DES, Petri Nets (PN) [13, 15] have been extensively used. Though the PN framework offers rigorous ground for theoretical analysis, it is very inconvenient for actual computational analysis of the practical DES, causing problems of computational complexity in scheduling and deadlock avoidance. PN theory does not provide efficient computational means for computing objects such as circular waits, siphons, etc. Matrix techniques that exploit the PN structure can be used to alleviate this. Recently, a matrix-based discrete event controller has been proposed, proving to be very efficient in computing PN objects and in sequencing tasks in manufacturing environments as well as sensor networks [1, 4, 5, 6, 10, 11, 12].

PN objects such as Circular Waits, Critical Siphons, and Critical Subsystems have to be computed to avoid deadlock in MRF. Due to the existence in FMRF of decision places, which are followed by two or more decision branches, these objects cannot be used for deadlock avoidance there without redefinition,

In this paper we show how these objects can be computed for FMRF. The key issue is that it is necessary to redefine, or generalize the definition of, critical subsystems for FMRF.

1 Supported by ARO grant DAAD 19-02-1-0366; ARO grant ARO W91NF-05-1-0314; NSF grant IIS-0326505; NSF grant CNS-0421282, Singapore SERC TSRP grant 0421120028, NI Lead User grant, Texas ARP grant 14-748779.

Also, we provide a matrix formulation which can efficiently compute these objects. This matrix formulation allows fast and efficient numerical computation techniques to be applied to PN analysis. We show how a MAXWIP dispatching policy can be formulated for FMRF to avoid blocking phenomena. Under this policy, deadlock in FMRF can be avoided by limiting the work in progress (WIP) in the Critical Subsystems of each CW.

The rest of the paper is organized as follows. In Section II we describe the properties that characterize FMRF systems using Petri Nets. In Section III we show the correlation between circular waits and structures referred to as Critical Siphons and Critical Subsystems. We generalize the notion of critical siphon to the case of FMRF. Section IV presents a matrix formulation over an or/and algebra that makes it efficient and direct to compute the Petri Net objects needed for deadlock avoidance. In Section V we illustrate the new notions by computing them for two FMRF examples, one of which admits deadlock, and one of which does not.

II. PETRI NET ANALYSIS

A. Petri Nets

A Petri net (PN) a bipartite digraph (*P, T, I, O*), where *P* is the set of places, *T* is a set of transitions, *I* is the set of input arcs from places to transitions, and *O* is the set of outputs arcs from transitions to places. One can represent *I* as an input incidence matrix which has $I(i,j)=1$ if there is an input arc from place *j* to transition *i* and *O* as an output incidence matrix which has $O(i,j)=1$ if there is an output arc from transition *i* to place *j*. The incidence matrix is defined as

$$W=O-I \qquad (1)$$

Given a node *v* (either transition or place), we define $\bullet v$ as the pre-set of *v* (set of nodes with arcs to *v*) and $v\bullet$ as the post-set of *v* (set of nodes with arcs from *v*). Similarly, for a set of nodes $S=\{v_i\}$, define $\bullet S=\{\bullet v_i\}$ and $S\bullet=\{v_i\bullet\}$.

The following assumptions allow one to represent a discrete event system by a Petri Net:

1. There are no machine failures.
2. No pre-emption. A resource cannot be removed from a job until it is complete.
3. Mutual exclusion. A single resource can be used for only one job at a time.
4. Hold while waiting. A process holds the resources already allocated to it until it has all the resources required to perform a job.

B. Reentrant Flow Lines

A special case of PN is the multiple reentrant flow-line system (MRF), see Figure 1. For MRF systems, we partition the set of places, $P=J\cup R\cup PI\cup PO$, with the places in *J, R, PI, PO* representing respectively, the jobs performed, the availability of resources, input of parts, and output of products. Each part path starts with a *PI*-place and terminates with a *PO*-place. We denote the set of job places *J* for part type *j* as J_j so that $J=\cup_j J_j$. Let the set of transitions along part path *j* be $x_{j1}, x_{j2},\ldots,x_{jLj}$, with x_{j1} and x_{jLj} being the initial and terminal transitions respectively.

R(p) is the set of resources needed by job *p*. For any resource $r\in R$ define the jobs performed by *r* as $J(r)$. We partition the resource set *R* as R_s and R_{ns}, with R_s being the set of shared resources, i.e. those needed for more than one job, and R_{ns} being the set of non-shared resources. Then, $|J(r)|=1$ if $r\in R_{ns}$ and $|J(r)|>1$ if $r\in R_s$, with $|S|$ denoting the cardinality of a set S (i.e. the number of elements).

More general than MRF are the free-choice multiple-reentrant flow-line systems (FMRF), which have no predetermined resource allocation for the jobs. That is, several different resources may be capable and available to perform a specific job. Then, routing decisions may be required along the part path about which resources to use for the next job, see Figure 2.

We formally define FMRF systems as a class of systems satisfying the following properties (ϕ being the empty set):

Properties of FMRF

1. $p\in P, \bullet p\cap p\bullet=\phi$
2. on part path j, $x_{j1}\bullet\cap P\setminus J=\phi$ and $\bullet x_{jLj}\cap P\setminus J=\phi$
3. $\forall p\in J,\ \bullet\bullet p\cap R=p\bullet\bullet\cap R:=R(p)$ with $|R(p)|=1$
4. $p\in J,\ |p\bullet|\geq 1$
5. $\forall p_i,\ \ p_k\in J_j,\ \ i\neq k,\ \ p_i\neq p_k$
6. $\forall p_i\in J_j,\ \ p_k\in J_l,\ \ j\neq l,\ \ p_{ji}\bullet\cap p_{lk}\bullet=\phi$
7. $R_s\neq\phi$

This means that there are: (1) no self loops, (2) each part path has a well-defined beginning and an end, (3) every job requires only one resource with no two consecutive jobs using the same resource, (4) there may be some jobs that can be done by different resources (i.e. routing decisions may have to be made), (5) there are no part path loops, (6) for any two distinct jobs on different part paths there is no assembly, i.e. two part paths cannot merge into one, (7) there are shared resources.

According to property 3, $R(p)=\bullet\bullet p\cap R=p\bullet\bullet\cap R$, with the cardinality $|R(p)|=1$; Under the foregoing assumption, one has $J(r)=r\bullet\bullet\cap J=\bullet\bullet r\cap J$.

Property 4 distinguishes MRF from FMRF systems. A FMRF system can have $|p\bullet|>1$ for some $p\in J$, so that routing decisions are needed. We call such job places 'decision places'. In MRF, one has $|p\bullet|=1\ \ \forall p\in J$. That is, MRF are a special class of FMRF. A transition $x\in p\bullet$ is said to be a posterior transition of *p*. A decision place has multiple posterior transitions, i.e. $|p\bullet|>1$. The resources used by decision places are called decision resources.

Figure 1 shows a sample MRF, while Figure 2 shows a sample FMRF. In Figure 1 the part paths are independent and neither split nor recombine. R1 is a shared resource along a single part path, and R2 is a shared resource between two part paths. In Figure 2 the part paths split at decision places. The decision places are B1 and B2, each followed by two transitions. A choice is required there to decide which resource (R1 or R2, respectively R3 or R4) to use for the next job.

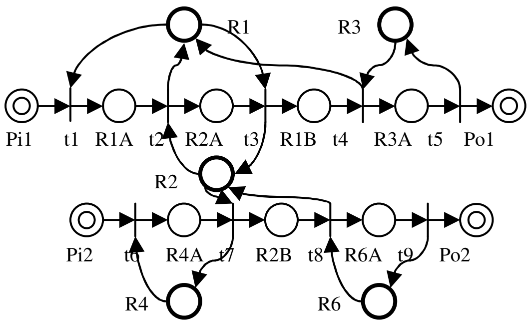

Figure 1. MRF system

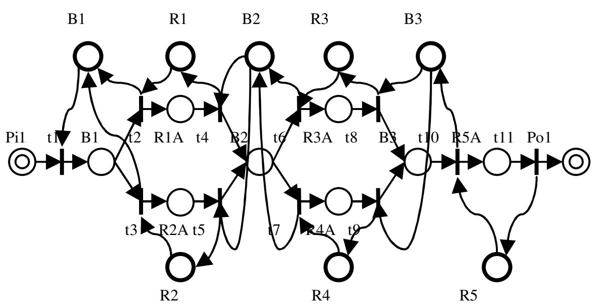

Figure 2. FMRF system

C. Circular Wait (CW)

The following background is taken from [3, 10, 16]. We say resource r_i waits for resource r_j (denoted $r_i \rightarrow r_j$) if the availability of r_j is an immediate prerequisite for the release of r_i, i.e., $\bullet r_i \cap r_j \bullet \neq \phi$. A wait relation digraph is defined as $G_w = (R, A)$ where R is the set of nodes and $A = \{a_{ij}\}$ is the set of edges with a_{ij} drawn if $r_i \rightarrow r_j$ (i.e. each a_{ij} represents a transition in $\bullet r_i \cap r_j \bullet$). In G_w, define an R-path between r_i and r_k as a set of R-places such that $r_i \rightarrow r_j \rightarrow \ldots \rightarrow r_k$. Then r_i is said to wait over an R-path for r_k, denoted $r_i \mapsto r_k$, if there is an R-path between r_i and r_k. A circular wait (CW) is a set of resources $C \subset R$, with $|C|>1$, such that for any ordered pair $\{r_i, r_j\} \subset C, r_i \mapsto r_j$. A CW always contains at least one shared resource.

The simplest CW is a set of resources $C \subset R$, such that for some appropriate re-labeling, one has $r_1 \rightarrow r_2 \rightarrow \ldots \rightarrow r_q \rightarrow r_1$, with $r_i \neq r_j$ for $i \neq j, 1 \leq i, j \leq q$. This will be referred to as a simple circular wait. A simple circular wait is a simple circuit in the graph and is a CW not containing any other CW. Consider a wait relation graph G_w and a CW $C \subset G_w$. Then for every $r \in C$, there exists at least one simple circular wait $\sigma \subset C$ such that $r \in \sigma$. A CW is a strongly connected sub graph in the digraph G_w and can be obtained by taking unions of non-disjoint simple CW.

Given a CW $C = \{r_i\}$, one can partition the set of transitions $\bullet r_i$ as $\bullet r_i = \bullet r_{io} \cup \bullet r_{i+}$, where $\bullet r_{io} = \{x \in T \mid \bullet x \cap C \neq \phi\}$, the set of input transitions of r_i with input arcs from some other $r_j \in C$, and $\bullet r_{i+} = \{x \in T \mid \bullet x \cap C = \phi\}$, the set of input transitions of r_i with no input arcs from any other $r_j \in C$. We loosely say that the transitions $\bullet r_{io}$ are 'in the CW C'.

The job set of CW $C = \{r_i\}$ is given by $J(C) = \bigcup_{i=1}^{n} J(r_i)$. Partition this as $J(C) = J(C)_+ \cup J(C)_0$, where $J(C)_0 = \{p \in J(C) \mid p\bullet \in \bullet r_{io}, r_i \in C\}$ and $J(C)_+ = \{p \in J(C) \mid p\bullet \in \bullet r_{i+}, r_i \in C\}$.

Note that for FMRF, one may have $J(C)_+ \cap J(C)_0 \neq \phi$. In fact, if $p \in J(C)$ is a decision place with C a simple CW, i.e. $|p\bullet| > 1$ so that it has more than one posterior transition, then, $p \in J(C)_+$ and $p \in J(C)_0$. This is precisely what distinguishes deadlock avoidance in MRF from deadlock avoidance in FMRF systems.

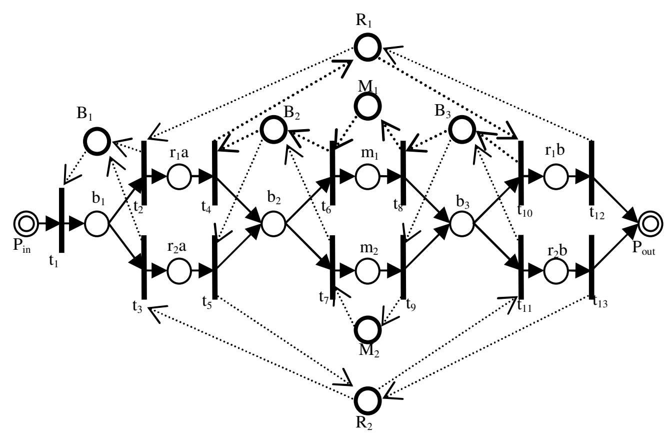

Figure 3. FMRF system

Figure 3 shows a FMRF system. Here, *R1-B2-M1-B3* is a *CW* C. Here, *r1a*, *m1*, *b2* and *b3* are in $J(C)_0$ and *r1b* is in $J(C)_+$. *b2* and *b3* are decision places and *B2* and *B3* are the respective decision resources. In FMRF systems, due to their construction, the decision places (*b2* and *b3* in this example) are also a part of $J(C)_+$.

D. Marking, Place Vector

A place $p \in P = J \cup R \cup PI \cup PO$ is said to be marked when it contains a token, which depending on the place containing it indicates an ongoing job, the existence of an available resource, a part in, or a product out. In a FMRF, the initial marking vector denoted as m_0 assigns tokens only to R and PI-places. It is assumed that the PO-places are always empty.

Given $p \in P$, $m(p)$ denotes the marking of p, i.e. the number of tokens in p. Given a set of places S, $m(S)$ denotes the number of tokens in S. A set of places is said to be unmarked or empty if none of its places has any tokens.

Given the set of places P, the PN place vector, or p-vector, $\vec{p}$ has dimension of $|P|$, and one element corresponding to each place. Let the set of all places be $P = \{p_1, p_2, ..., p_Q\}$. Then the place vector has Q elements. Any set of m places $\{p_{i_k} \in P \mid k = 1,2,...,m\}$ can be represented as a $|P|$-vector $\vec{p}$ having m entries $\vec{p}_{i_k} = 1$ and zero entries otherwise. The PN marking vector $m(\vec{p}) = [m(p_1) \quad m(p_2) \cdots]^T \in N^{|P|}$, with the natural numbers $N = \{0, 1, 2...\}$, gives the number of tokens in each place.

Define the PN transition vector, or x-vector, x to have dimension of $|T|$, and one element x_i corresponding to each transition. Let the set of all transitions be $T = \{x_1, x_2, ...x_L\}$. Then the transition vector has L elements. A set of m transitions $\{x_{i_k} \in T \mid k = 1,2,...,m\}$ can be represented as an L-vector x having m entries $x_{i_k} = 1$ and zero entries otherwise. The well-known PN marking transition equation [15],

$$m^+(\vec{p}) = m(\vec{p}) + W^T x \tag{2}$$

gives the new marking vector $m^+(\vec{p})$ in terms of the previous marking vector $m(\vec{p})$ and the transitions that have fired appearing as 1's in x.

A p-invariant is defined as a set of resources and places that is in the null space of W. i.e.

$$Wp=0. \tag{3}$$

Note that if p is a p-invariant, then

$$p^T m^+(\vec{p}) = p^T m(\vec{p}) + p^T W^T x = p^T m(\vec{p})$$

so that the number of tokens in a p-invariant is constant. One type of p-invariant is given by any resource plus all of its jobs, $r \cup J(r)$.

E. Circular Blocking

A circular blocking CB is a circular wait that is empty and will always remain so [3, 10, 16]. That is, for a CW $C = \{r_i\}$:

1. $m(C)=0$, and
2. no tokens will ever be added to C.

In this situation, one is said to have deadlock, where the resources in the CW are waiting for each other and will never again become available. If a resource on a given part path is involved in a CB, then all downstream activity along that part path will eventually end. That is, after some time, the downstream jobs on that part path will never again be performed. Let $\overline{C} = \{C_i\}$ be a set of disjoint CW. Then, $\overline{C}$ is said to be in CB if each CW C_i is in CB.

F. Siphons

The analysis of CB and deadlock can be carried out formally using the notion of siphon. A siphon is a set of places having the property that its input transition set is contained in its output transition set i.e.

$$\bullet S \subset S \bullet \tag{4}$$

A siphon has the key property that, once it is unmarked, it remains so.

A minimal siphon of a CW C is the smallest siphon containing the CW. Define a critical siphon for a CW C as a smallest siphon which has the property that a CW is a CB if and only if the critical siphon is empty. It is shown in [10, 12] that a minimal siphon in MRF is a critical siphon.

For MRF systems, for every CW C, the set of places $\hat{S}_c$ defined by

$$\hat{S}_c = C \cup J(C)_+ \tag{5}$$

is a minimal siphon as well as a critical siphon, where $J(C)_+$ was defined earlier [10, 16].

In the case of FMRF systems, this object is still a minimal siphon, but it cannot be used in deadlock analysis due to decision places. Hence it is not a critical siphon as is now shown.

Lemma 1: For FMRF systems, $\hat{S}_c$ is a minimal siphon for the CW C.

Proof: To prove that $\hat{S}_c$ is a siphon, we need to show $\bullet \hat{S}_c \subseteq \hat{S}_c \bullet$, i.e., every transition having an output place in $\hat{S}_c$ has an input place in $\hat{S}_c$. It is known that, for every $r_i \in C$, there exists $r_j \in C, i \neq j$, such that $\bullet r_i \cap r_j \bullet \neq \phi$. So, if, $r_i \in C \cap R_{ns} \cap \bar{B}_r$ with B_r being the decision resources, then $|\bullet r_i| = 1$, and there exists some $r_j \in C, i \neq j$, such that $\bullet r_i \in \{r_j \bullet\}$. On the other hand, $r_i \in C \cap (R_s \cup B_r)$, then for every $x \in \{\bullet r_i\}$, either $x \in \{r_j \bullet\}$, for some $r_j \in C, i \neq j$, or $x \in p\bullet$, for some $p \in J(C)_+$. Moreover, $\forall p \in J(C)_+$, $\bullet p \in \{r \bullet\}$ for some $r \in C$ by Property 3. This proves that $\bullet \hat{S}_c \subseteq \hat{S}_c \bullet$, therefore by construction, S_c is a siphon.

The minimality of $\hat{S}_c$ is obvious from the way it is constructed in FMRF systems. If any $p \in J(C)_+$ is left out, then there will be some $r_k \in C \cap R_s$ with $\{\bullet r_k\} \not\subset \hat{S}_c \bullet$. Similarly if one of the resources in C is not considered, $r_i \in C$ is left out, then there will be at least one $r_j \in C$ with $\{\bullet r_j\} \not\subset \hat{S}_c \bullet$. Both the cases violate the definition of siphon. ■

Lemma 2: Let C be a simple CW and $\hat{S}_c$ contain a decision place whose associated decision resource is not a shared resource, then $\hat{S}_c$ contains a p-invariant and can never be empty.

Proof: Let C be a simple CW and $\hat{S}_c$ be defined by (5). Suppose $\hat{S}_c$ contains a decision place p with an associated resource place r_p that is not shared, i.e. $J(r_p) = p$. Then $C \cap \left|\bullet \bullet r_p\right| = 1$ and $\left|p \bullet\right| > 1$, so that $p \in J(C)_0$ and $p \in J(C)_+$. Therefore, $p \in \hat{S}_c$. Moreover, $r_p \in \hat{S}_c$ so that the p-invariant $r_p \cup J(r_p)$ is in $\hat{S}_c$. ■

III. Critical Siphons and Deadlock Analysis for FMRF Systems

Lemma 1 shows that $\hat{S}_c$ given by (5) is indeed a minimal siphon, even for FMRF, but Lemma 2 shows that in some situations for FMRF, $\hat{S}_c$ can never be empty. Since deadlock can still occur in such situations, it is necessary to provide an alternative formula for critical siphons for FMRF.

A. *FMRF Critical Siphons*

Let C_s be the set of all simple CWs in a FMRF system. Let p be a decision place with resource place r_p=$R(p)$. Define the set of all the simple CWs in C_s containing the decision resource r_p as

$$C_p = \{C \in C_s \mid r_p \in C\} \qquad (6)$$

Given a CW C, let the set of its decision places be

$$J_{dec}(C) = \{p \in J(C) \mid \left|p \bullet\right| > 1\} \qquad (7)$$

Define,

$$C_p(C) = \{C_p \mid p \in J_{dec}(C)\} \qquad (8)$$

which is the key to deadlock analysis in FMRF systems.

We need to recursively compute $C_p(C)$, as each individual simple CW in $C_p(C)$ defined in (8) may contain more decision resources, and so on as shown in Figure 4. Hence we need an algorithm to calculate $C_p(C)$. Algorithm 1 shows a modified version of the binary tree algorithm [2].

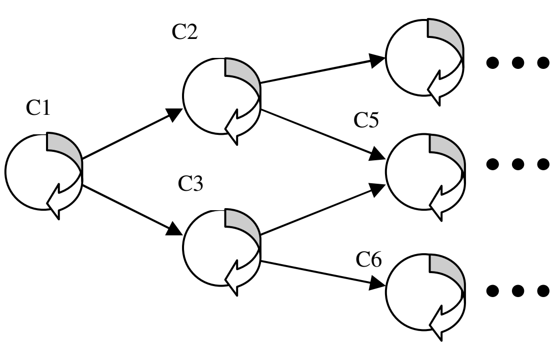

Figure 4. Tree representation of simple CW dependence through decision places

Algorithm 1: Compute $C_p(C)$ for a CW C

Given a CW C,
1. Calculate $C_p(C)$ using Equation 8
2. $x = 1$
3. $y = |C_p(C)|$
4. If $x \le y$
    If any $r_p \in (C_x \subset C_p(C)) \notin C_p(C) \setminus C_x$
    $C_p(C) = [\ C_p(C)\ C_p(C_x)\ ]$
    $x = x + 1$
    $y = |C_p(C)|$
    Go to 4
    else
    $x = x + 1$
    Go to 4
   else
5. Return $C_p(C)$ ■

Note that this algorithm computes the set $C_p(C)$ for a single CW. Also, when this algorithm terminates, $C_p(C)$=$C_p(C_p(C))$. This algorithm terminates in one of the two ways:

a) $J(C_p(C))_+ \cap J(C_p(C))_0 = \phi$ or

b) $J(C_p(C))_+ \cap J(C_p(C))_0 \ne \phi$.

In the discussions ahead, we will prove that in case (b), $C_p(C)$ can never be in CB and hence no CW in $C_p(C)$ can be in CB.

Define a relation $C \sim \tilde{C}$, if there exists a resource $r \in C \cap \tilde{C}$. Note that "$\sim$" satisfies the following properties:

1. $\tilde{C} \sim \tilde{C}$ (Reflexivity)
2. If $C \sim \tilde{C}$ then $\tilde{C} \sim C$ (Symmetry)
3. If $C \sim \tilde{C}$ and $\tilde{C} \sim \bar{C}$ then it is not guaranteed that $C \sim \bar{C}$ (Not Transitive)

Therefore, "$\sim$" is not an equivalence relation. Define a second relation $C \approx \tilde{C}$ if there exists a set of CW C_i such that $C \sim C_1 \sim C_2 \ldots \sim C_n \sim \tilde{C}$. Note that,

1. $C \sim C$ (Reflexivity)
2. If $C \sim \bar{C}$ then $\bar{C} \sim C$ (Symmetry)
3. If $C \approx \tilde{C}$ then $C \approx \tilde{C}$ (Transitive)

Therefore the relation "$\approx$" is an equivalence relation [8] and partitions the set of all the CW into disjoint equivalence classes $K(C)$ ={ $\bar{C} : \bar{C} \approx C$ }, i.e. $K(C)$=$K(\bar{C})$.

Corollary 1: For any simple CW $\bar{C} \in C_p(C)$, $C_p(C)$ = $C_p(\bar{C})$.

Proof: The algorithm merely computes the set $C_p(C)$=$K(C)$. Therefore, $C_p(C)$ = $C_p(\bar{C})$. ■

Lemma 3: $C_p(C)$ is a CB if and only if:

1. $J(C_p(C))_+ \cap J(C_p(C))_0 = \phi$
2. $m(C_p(C))$ = 0 and
3. for each $r_i \in C_p(C), \forall p \in J(r_i)$ with $m(p) \ne 0$,

$p \in J(C_p(C))_0$

Proof: Necessity: Let $C_p(C)$ be in CB i.e it is empty. This means that $m(C_p(C))=0$. Let $p \in J(C_p(C))$ and suppose $p \in J(C_p(C))_+$. Then either $p\bullet \cap r\bullet = \phi$ or there exists a resource $r \notin C_p(C)$ such that $p\bullet \cap r\bullet \neq \phi$. This means that a transition $t \in p\bullet$ may fire and put a token in *R(p)* so that $C_p(C)$ does not remain empty. Therefore, all the marked jobs are in $J(C_p(C))_0$ and not in $J(C_p(C))_+$ and conditions 1 and 3 hold.

Sufficiency: Conditions 1 and 3 of Lemma 3 imply that any $r_i \in C_p(C)$ can get a token if and only if some $r_j \in C_p(C)$ with $j \neq i$, can get a token. However by condition 2 of Lemma 3, all the R-places in $C_p(C)$ are empty, and hence none of them can ever get any token. In other words, $C_p(C)$ is in CB. ■

Lemma 4: For MRF, C is in CB if and only if:

1. $m(C_p(C)) = 0$ and
2. for each $r_i \in C_p(C), \forall p \in J(r_i)$ with $m(p) \neq 0$, $p \in J(C_p(C))_0$

Proof: In MRF, $C_p(C) = C$ and condition 1 of Lemma 3 always holds. ■

Given a CW C define the object,

$$S_c = C_p(C) \cup J(C_p(C))_+ \quad (9)$$

The next results show that S_c is a critical siphon for C for FMRF systems under a certain condition.

Lemma 5: For any CW C, S_c defined in (9) is a minimal siphon for $C_p(C)$.

Proof: One needs to show that $\bullet S_c \subseteq S_c \bullet$, i.e., every transition having an output place in S_c has an input place in S_c. By construction, $\bullet S_c \subset \bullet C_p(C)$, i.e., the output places for these transitions are resources in $C_p(C)$. It is known that, for every $r_i \in C_p(C)$, there exists $r_j \in C_p(C), i \neq j$, such that $\bullet r_i \cap r_j \bullet \neq \phi$. So, if $r_i \in C_p(C) \cap R_{ns}$, there exists some $r_j \in C_p(C), i \neq j$, such that $\bullet r_i \in \{r_j \bullet\}$. On the other hand, $r_i \in C_p(C) \cap R_s$, then for every $x \in \{\bullet r_i\}$, either $x \in \{r_j \bullet\}$, for some $r_j \in C_p(C), i \neq j$, or $x \in p\bullet$, for some $p \in J(C_p(C))+$. Moreover, $\forall p \in J(C_p(C))_+, \bullet p \in \{r\bullet\}$ for some $r \in C_p(C)$ by Property 3. This proves that $\bullet S_c \subseteq S_c \bullet$, therefore by construction, S_c is a siphon. Minimality follows by the way S_c is constructed. ■

Note, for MRF, $C_p(C) = C$, $S_c = \hat{S}_c$ and Lemma 1 is recovered.

Lemma 6: If $J(C_p(C))_+ \cap J(C_p(C))_0 \neq \phi$, then S_c contains a p-invariant and can never be empty.

Proof: If $J(C_p(C))_+ \cap J(C_p(C))_0 \neq \phi$, it means that there exists a decision place p such that $p \in J(C_p(C))_+$ and $p \in J(C_p(C))_0$. Also, by definition of S_c, $p \in S_c$. If r_p is the corresponding resource of p, $r_p \in S_c$, so the p-invariant $r_p \cup J(r_p)$ is also in S_c. Hence it can never be empty. ■

The next result shows that S_c is a critical siphon for $C_p(C)$.

Theorem 1: Given $C_p(C)$ and let $J(C_p(C))_+ \cap J(C_p(C))_0 = \phi$. Then $C_p(C)$ is in CB if and only if S_c is empty.

Proof. Necessity: If $C_p(C)$ is in CB, Lemma 3 shows that $m(C_p(C))=0$ and $m(p) \neq 0$ only for $p \in J(C_p(C))_0$. Suppose S_c is not empty. Then there is a place p' such that $p' \in J(C_p(C))_+$ and $m(p') \neq 0$. This is a contradiction.

Sufficiency: S_c is a siphon and by definition, once it is empty, it will remain so. Thus, when $m(C_p(C))=0$, with all the tokens lost to some $p \in J(r_i)$, with $m(p) \neq 0$ means that $p \notin J(C_p(C))_+$ and hence, $p \in J(C_p(C))_0$. Since $J(C_p(C))_+ \cap J(C_p(C))_0 = \phi$, by Lemma 3 $C_p(C)$ is in CB. ■

Theorem 2: Let $J(C_p(C))_+ \cap J(C_p(C))_0 \neq \phi$. Then $C_p(C)$ can never be in CB.

Proof: For $C_p(C)$ to be in CB, $m(C_p(C))=0$ and $m(p) \neq 0$ only with $p \in J(C_p(C))_0$. Suppose also $p \in J(C_p(C))_+$. This implies $\bullet(p\bullet) \cap R \notin C_p(C)$. Thus, this token can be fired. Hence, deadlock does not occur. ■

There exists a relationship between Lemma 6 and Theorem 2. When $J(C_p(C))_+ \cap J(C_p(C))_0 \neq \phi$, it means S_c has a p-invariant and hence, $C_p(C)$ can never be in CB.

Lemma 7: Given a CW $C=\{r_i\}$. Then

(i) $J(C)_0 \subset J(C_p(C))_0$

(ii)If $J(C_p(C))_+ \cap J(C_p(C))_0 = \phi$,then $J(C)_+ \setminus J(C)_0 \subset J(C_p(C))_+$

Proof: (i) Given $C=\{r_i\}$. Then by definition $C \subset C_p(C)$. Now, let there be place p' such that $p' \in J(C)_0$ and $p' \notin J(C_p(C))_0$. In this case, $\bullet(p'\bullet) \cap C \neq \phi$ and $\bullet(p'\bullet) \cap C_p(C) = \phi$. This means that $C \not\subset C_p(C)$, which is a contradiction. Hence, $J(C)_0 = \{p \in J(C) \mid p\bullet \in \bullet r_{io}, r_i \in C\}$ $\subset \{p \in J(C_p(C) \mid p \in r_{j0}, r_j \in C_p(C)\} \subset J(C_p(C))_0$

(ii) In FMRF, if C is a CW with decision place p then, $|p\bullet| > 1$, $p \in J(C)_0$ and also possibly $p \in J(C)_+$ as stated earlier. Hence the set $\{J(C)_+ \setminus J(C)_0\}$ strictly consists of job places belonging to $J(C)_+$ without decision places. Let $\{J(C)_+ \setminus J(C)_0\}$ be denoted by $J(C)_{s+}$. Now by definition, $J(C) = J(C)_+ \cup J(C)_0$ and $J(C_p(C))=J(C_p(C))_+ \cup J(C_p(C))_0$. By (i), $J(C)_0 \subset J(C_p(C))_0$. Since $C \subset C_p(C)$, $J(C) \subset J(C_p(C))$. Therefore, $J(C) \setminus J(C)_0 \subset J(C_p(C)) \setminus J(C_p(C))_0$. The term $J(C) \setminus J(C)_0$ is equal to $J(C)_+ \setminus J(C)_0$ i.e. $J(C)_{s+}$. Also, the term $J(C_p(C)) \setminus J(C_p(C))_0$ is $J(C_p(C))_+$ as $J(C_p(C))_+ \cap J(C_p(C))_0 = \phi$. Thus, $J(C)_+ \setminus J(C)_0 \subset J(C_p(C))_+$. ■

The next results show that S_c is a critical siphon for C.

Theorem 3: If $J(C_p(C))_+ \cap J(C_p(C))_0 = \phi$ then CW C is in CB if and only if $C_p(C)$ is in CB.
Proof: Necessity: Suppose C be in CB and $C_p(C)$ is not in CB.
Case 1: If C does not contain decision places then $C=C_p(C)$ and the result follows.
Case 2: If C is in CB, then $m(C)=0$. Consider a resource $r \in C$ and its job place p such that r is dead i.e. $m(r)=0$ for all future firings. Since r is dead, $p\bullet \in \bullet r$ and hence, $\bullet(p\bullet) \cap R \in \bullet\bullet r \cap R$ must all be dead for C to be in CB. $C_p(C)$ and C are related to each other by their decision resources. So, iterating this process in a PN from r backwards to a path of transitions and R-places which include all resources of $C_p(C)$, must be dead. For C to remain in CB, no token should be added to C and hence no token should be added to $C_p(C)$. Thus, all the jobs in $C_p(C)$ must be in $J(C_p(C))_0$. The condition $J(C_p(C))_+ \cap J(C_p(C))_0 = \phi$ ensures that when $C_p(C)$ is in CB, all the tokens are in $J(C_p(C))_0$ and not $J(Cp(C))_+$. Hence C is in CB.

Sufficiency: Let $C_p(C)$ be in CB. Then $m(C_p(C))=0$ and hence $m(C)=0$. Also by Lemma 3, all marked jobs are in $J(C_p(C))_0$. and not in $J(C_p(C))_+$ and $J(C_p(C))_0 \cap J(C_p(C))_+ = \phi$. By Lemma 7 $J(C)_0 \subset J(C_p(C))_0$ and $J(C)_{s+} \subset J(C_p(C))_+$ where $J(C)_{s+} = \{J(C)_+ \setminus J(C)_0\}$. Now, $J(C)_{s+} \cap J(C_p(C))_0 = \phi$. Hence for all resources r_i in C and $p \in J(r_i)$ having $m(p) \neq 0$, $p \in J(C)_0$. Therefore, $r_i \in C$ can get a token if and only if some other $r_j \in C_p(C)$ has a token. Hence C is in CB. ■

Corollary 2: Given a CW C, and suppose $J(C_p(C))_0 \cap J(C_p(C))_+ = \phi$, then C is in CB if and only if S_c is empty.
Proof: By Theorem 3, C is in CB if and only if $C_p(C)$ is in CB. Then Theorem 1 completes the proof. ■

Corollary 3: Given $J(C_p(C))_0 \cap J(C_p(C))_+ = \phi$, $C_p(C)$ is in CB if and only if all the simple CWs in $C_p(C)$ are in CB.
Proof: Necessity: Suppose $C_p(C)$ is in CB. Let there be a simple CW $C_1 \subset C_p(C)$ such that it is not in CB. It means either $m(C_1) \neq 0$ or token can be added to C_1. Since $C_1 \subset C_p(C)$, $m(C_p(C)) \neq 0$ or tokens can be added to $C_p(C)$. Hence, $C_p(C)$ is no longer in CB.

Sufficiency: If all the simple CWs in $C_p(C)$ are in CB, it means $m(C_i)=0$ and no tokens will ever be added to C_i. Since $C_p(C)=\bigcup C_i$, $m(C_p(C))=0$ and $m(J(C_p(C)_0)) \neq \phi$. Hence by definition of CB, $C_p(C)$ is in CB, provided $J(C_p(C))_0 \cap J(C_p(C))_+ = \phi$. ■

Corollary 4: Given a CW C, and suppose $J(C_p(C))_0 \cap J(C_p(C))_+ = \phi$, then any CW $\bar{C} \in C_p(C)$ is in CB if C is in CB.
Proof: By Theorem 3, C is in CB if and only if $C_p(C)$ is in CB. Then Corollary 3 completes the proof. ■

B. FMRF Critical Subsystems

For better perspective on achieving dispatching policies with deadlock avoidance, the concept of critical subsystems [10] is introduced. To avoid deadlock, work in progress (WIP) must be limited within certain critical subsystems that are constructed by using critical siphons, called MAXWIP.

A critical subsystem for a CW C in MRF is defined as the set of J-places $J(C)_0$. In case of FMRF, Critical Subsystem for a CW C can be defined if $J(C_p(C))_0 \cap J(C_p(C))_+ = \phi$ and in this case it is defined as the set of J-places $J(C_p(C))_0$. Define the set $S_c^* = S_c \cup J(C_p(C))_0$ called the support of the binary p-invariant that minimally covers S_c.

Lemma 8: In FMRF, a simple CW C is not in deadlock if and only if one has $m(J(C_p(C))_0 < m(S_c^*)$.

Proof: There is CB in a system if and only if S_c is empty (by Theorem 1). Since $S_c^* = S_c \cup J(C_p(C))_0$ is the support of the binary p-invariant, the total number of its tokens is conserved. Thus, S_c will be empty if and only if all the tokens are in $J(C_p(C))_0$ i.e. the critical subsystem. ■

The condition $m(S_c^*)$ can be given in terms of the initial marking of $C_p(C)$, since for FMRF, the initial marking assigns tokens only to R and PI places.

Note that one can now avoid deadlock of a simple CW C by ensuring that the marking of its critical subsystem S_c^* is less than the total marking of the initial resources in $C_p(C)$. To be able to analyze FMRF systems (and for MRF systems) and all its possible deadlock structures, we need to identify the set of all CWs i.e simple CWs and CWs composed of union of non-disjoint simple CWs (unions through shared resources among simple CWs) [9, 11]. Let us denote this set as CW_r.

Lemma 9: There is no CB in the FMRF system if for every $C \in CW_r$, the set $C_p(C)$ is not in CB.

Proof: By Theorem 1, when $C_p(C)$ is not in CB, it implies that S_c is not empty. Then Lemma 8 completes the proof. ■

Lemma 10: In FMRF, there is no CB in the system if and only if for every $C \in CW_r$, one has $m(J(C_p(C))_0$ less than the initial marking of $C_p(C)$. ■

These results show that we can avoid deadlock by MAXWIP, i.e. limiting WIP in all the critical subsystems.

IV. Matrix Computation of Petri Net Objects

PN provide great pictorial insight and mathematical techniques for analysis, but they have had the deficiency of not providing an efficient computational framework for simple computer-based analysis. We now introduce a matrix framework for computing structural objects of a PN that corrects this deficiency.

Commensurate with the partitioning $P = J \cup R \cup PI \cup PO$ with *J, R, PI*, and *PO* being the sets of job places, resource places, input places, and output places respectively, partition the place vector as

$$\vec{p} = \begin{bmatrix} v \\ r \\ pi \\ po \end{bmatrix}$$

with $v \in J,\, r \in R,\, pi \in PI,\, po \in PO$.

Similarly, partition the input incidence matrix as $I = [F_v \quad F_r \quad F_i \quad F_o]$. Note that there are no input arcs to transitions from the places in *PO*, so that $F_o = 0$, a matrix of zeros. Partition the output incidence matrix as $O = [S_v^T \quad S_r^T \quad S_i^T \quad S_o^T]$. Note that there are no output arcs from transitions to the places in *PI*, so that $S_i = 0$. These sub-matrices are Boolean matrices having entries of 0 or 1.

A. Or/And Algebra For Computing PN Objects

Given Boolean matrices $A = [a_{ij}]$ and $B = [b_{ij}]$, define a logical or/and matrix algebra wherein addition operations are replaced by logical 'or' and multiplication operations by logical 'and'. That is, the matrix product is defined by $C = A \otimes B$ with

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee (a_{i3} \wedge b_{3j}) \vee \cdots$$

with $\wedge$ denoting logical 'and' and $\vee$ denoting logical 'or'. The matrix sum is defined by $C = A \oplus B$ with

$$c_{ij} = a_{ij} \vee b_{ij}\,.$$

Note that these matrix products are easily performed using standard software programs including MATLAB®, etc.

Then one has computational methods for computing PN objects based on the following result.

Lemma 11: Matrix computation of PN pre and post sets for places.

1. Let *v* represent a set of job places. Then
   a. $v\bullet$ is represented by the vector $F_v \otimes v$
   b. $\bullet v$ is represented by the vector $S_v^T \otimes v$
2. Let *r* represent a set of resource places. Then
   a. $r\bullet$ is represented by the vector $F_r \otimes r$
   b. $\bullet r$ is represented by the vector $S_r^T \otimes r$

Proof:

1. a: Any set of job places $\{j_{i_k} \in J \mid k = 1,2,...,m\}$ is represented as a vector *v* having *m* entries $v_{i_k} = 1$ and zero entries otherwise. Then, by definition $v\bullet = \{j_{i_k}\bullet\} = \{x_l \in T \mid I(l,i_k) = 1, some\ i_k\}$ $= \{x_l \in T \mid F_v(l,i_k) = 1, some\ i_k\}$. Moreover, $x = F_v \otimes v = \{x_l\}$ with

$$x_l = \vee_i [F_v(l,i) \wedge v(i)] = \vee_{i_k} [F_v(l,i_k) \wedge v(i_k)]$$

which is equal to 1 if and only if $F_v(l,i_k) = 1$ for some i_k.

1. b: Similarly, by definition , $\bullet v = \{\bullet j_{i_k}\} = \{x_l \in T \mid O(l,i_k) = 1, some\ i_k\}$ $= \{x_l \in T \mid S_v^T(l,i_k) = 1, some\ i_k\}$. Also, $x = S_v^T \otimes v = \{x_l\}$ with

$$x_l = \vee_i [S_v^T(l,i) \wedge v(i)] = \vee_{i_k} [S_v^T(l,i_k) \wedge v(i_k)]$$

which is equal to 1 if and only if $S_v^T(l,i_k) = 1$ for some i_k.

2. a: Any set of resources places $\{r_{i_k} \in R \mid k = 1,2,...,m\}$ is represented as a vector *r* having *m* entries $r_{i_k} = 1$ and zero entries otherwise. Then, by definition $r\bullet = \{r_{i_k}\bullet\} = \{x_l \in T \mid I(l,i_k) = 1, some\ i_k\}$ $= \{x_l \in T \mid F_r(l,i_k) = 1, some\ i_k\}$. Moreover, $x = F_r \otimes r = \{x_l\}$ with

$$x_l = \vee_i [F_r(l,i) \wedge r(i)] = \vee_{i_k} [F_r(l,i_k) \wedge r(i_k)]$$

2. b: Similarly, by definition, $\bullet r = \{\bullet r_{i_k}\} = \{x_l \in T \mid O(l,i_k) = 1, some\ i_k\}$ $= \{x_l \in T \mid S_r^T(l,i_k) = 1, some\ i_k\}$. Also, $x = S_r^T \otimes r = \{x_l\}$ with

$$x_l = \vee_i [S_r^T(l,i) \wedge r(i)] = \vee_{i_k} [S_r^T(l,i_k) \wedge r(i_k)]$$

which is equal to 1 if and only if $S_r^T(l,i_k) = 1$ for some i_k. ■

Lemma 12: Matrix computation of PN pre and post sets for transitions:

1. Let *x* represent a set of transitions. Then
   a. $x\bullet \cap J$ is represented by the vector $S_v \otimes x$
   b. $x\bullet \cap R$ is represented by the vector $S_r \otimes x$

c. $\bullet x \cap J$ is represented by the vector $F_v^T \otimes x$

d. $\bullet x \cap R$ is represented by the vector $F_r^T \otimes x$

Proof:

Post sets of transitions are sets of job places and resources, i.e. $x\bullet = \{J, R\}$.

1. a. Any set of transitions $\{t_l \in T \mid l = 1,2,...,n\}$ is represented as a vector x having n entries $x_l = 1$ and zero entries otherwise. Then, by definition $x\bullet \cap J = \{t_l \bullet\} \cap J = \{j_{ik} \in J \mid O(i_k, l) = 1, some \quad l\}$ $= \{j_{ik} \in J \mid S_v(i_k, l) = 1, some \quad l\}$. Moreover, $S_v \otimes x = \{j_{ik}\}$ where

$$j_{ik} = \bigcup_i \{S_v(i_k, l) \otimes x(i)\} = \bigcup_l \{S_v(i_k, l) \otimes x_l\}$$

which is equal to 1 if and only if $S_v(i_k, l) = 1$ for some l.

1. b. Any set of transitions $\{t_l \in T \mid l = 1,2,...,n\}$ is represented as a vector x having n entries $x_l = 1$ and zero entries otherwise. Then, by definition $x\bullet \cap R = \{t_l \bullet\} \cap R = \{r_{ik} \in R \mid O(i_k, l) = 1, some \quad l\}$ $= \{r_{ik} \in R \mid S_r(i_k, l) = 1, some \quad l\}$. Moreover, $S_r \otimes x = \{r_{ik}\}$ where

$$r_{ik} = \bigcup_i \{S_r(i_k, l) \otimes x(i)\} = \bigcup_l \{S_r(i_k, l) \otimes x_l\}$$

which is equal to 1 if and only if $S_r(i_k, l) = 1$ for some l.

1. c. Any set of transitions $\{t_l \in T \mid l = 1,2,...,n\}$ is represented as a vector x having n entries $x_l = 1$ and zero entries otherwise. Then, by definition $\bullet x \cap J = \{\bullet t_l\} \cap J = \{j_{ik} \in J \mid I(i_k, l) = 1, some \quad l\}$ $= \{j_{ik} \in J \mid F_v^T(i_k, l) = 1, some \quad l\}$. Moreover, $F_v^T \otimes x = \{j_{ik}\}$ where

$$j_{ik} = \bigcup_i \{F_v^T(i_k, l) \otimes x(i)\} = \bigcup_l \{F_v^T(i_k, l) \otimes x_l\}$$

which is equal to 1 if and only if $F_v^T(i_k, l) = 1$ for some l.

1. d. Any set of transitions $\{t_l \in T \mid l = 1,2,...,n\}$ is represented as a vector x having n entries $x_l = 1$ and zero entries otherwise. Then, by definition $\bullet x \cap R = \{\bullet t_l\} \cap R = \{r_{ik} \in R \mid I(i_k, l) = 1, some \quad l\}$ $= \{r_{ik} \in R \mid F_r^T(i_k, l) = 1, some \quad l\}$. Moreover, $F_r^T \otimes x = \{r_{ik}\}$ where

$$r_{ik} = \bigcup_i \{F_r^T(i_k, l) \otimes x(i)\} = \bigcup_l \{F_r^T(i_k, l) \otimes x_l\}$$

which is equal to 1 if and only if $F_r^T(i_k, l) = 1$ for some l.

■

B. Circular Waits In Matrix Form

A wait relation digraph for MRF [12] is defined as

$$W_r = (S_r \otimes F_r)^T \qquad (10)$$

Each 'one' in the elements w_{ij} of W_r, represents that the digraph has an arc from resource i to resource j. CWs appear as loops in this digraph. Simple CW appear as simple loops e.g. not containing any smaller loops. W_r is used to compute all the simple CWs in MRF systems using a binary string algebra approach used by [12] which gives an output matrix CW_r^*. Each CW is represented as a row in the matrix CW_r^*. In this matrix CW_r^* each entry of 'one' in position (i,j) means that each resource j is included in the i^{th} simple CW.

However, due to the complexity of the Free-Choice extension of the MRF systems, and due to the diversity of loop paths that a set of resources contained in a simple CW might have, we need to identify not only the resources that compose each simple CW, but also the transitions that link them. This will give us specific information needed to locate siphons needed for constructions of our deadlock policy for FMRF systems. We define W_t (by duality of W_r) as

$$W_t = (F_r \otimes S_r)^T \qquad (11)$$

This is a digraph of transitions. That is, W_t is a digraph having arcs from transition t_i to transition t_j (by 'bypassing' a resource $\bullet t_j \cap t_i \bullet$.). Each 'one' in the elements w_{ij} of W_t, represents that the digraph has an arc from transition i to transition j. Then, one can identify loops among transitions by using string algebra as above which gives the output matrix CW_t^*. We loosely say t is in CW if $t \in CW_t^*$.

But, even if we calculate two outcomes from the string algebra, transition loops and resource loops, we will not be able to identify which set of transition loops correspond to which set of resource loops (due to the behavior of the algorithm.). To do this, define,

$$W = \begin{bmatrix} O_r & S_r \\ F_r & O_t \end{bmatrix} \qquad (12)$$

where O_r is a zero-matrix having nxn elements, O_t is a zero-matrix having mxm elements, n be the number of resources or rows (column) of S_r (F_r), and m be the number of transitions or rows (columns) of F_r (S_r). This is a digraph of transitions T and resources R.

For example, this digraph can be obtained by erasing all the jobs places from Figure 3, and keeping the resources, the transitions, and all the links between them, as shown in Figure 5.

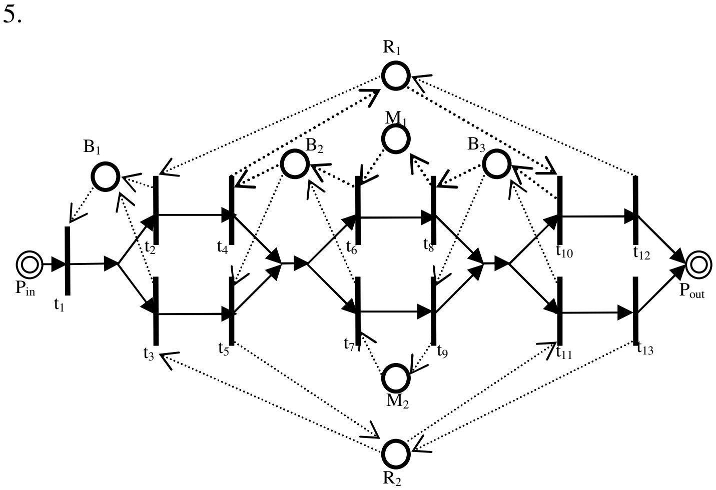

Figure 5. Graphical representation of digraph matrix W

Note that, using this digraph matrix W with the binary string algebra algorithm given in [12], we get the set C_w = [CW_r^* CW_t^*], where CW_r^* is the set of simple CW of resources and CW_t^* is a set of simple CW of transitions.

In order to find the complete set of simple CWs and the union of non-disjoint simple CWs, we use the Gurel algorithm [10] which gives a matrix G. G provides the set of composed CWs (rows) from unions of simple CWs (columns) i.e. an entry of '1' in every (i, j) position implies j^{th} simple CW is included in the i^{th} composed CW. Then, we can calculate the set of loop resources CW_r and loop transitions CW_t using the following constructions,

$$CW_r = G^T \otimes CW_r^* \quad (13)$$

$$CW_t = G^T \otimes CW_t^* \quad (14)$$

Note that * denotes a simple CW, while CW_r and CW_t refer to all the CWs (i.e. simple and their unions).

C. Matrix Algorithm for Computing $C_p(C)$

Property 4 of FMRF, i.e $p \in J$, $|p\bullet| \geq 1$, is what differentiates FMRF from MRF systems. We apply this property of FMRF to compute $C_p(C)$ using matrices. To do so, we must first compute C_p and $J_{dec}(C)$ for each CW as required to run Algorithm 1. The next algorithm shows a routine which can easily be implemented using tools such as MATLAB®, to compute C_p and $J_{dec}(C)$.

Algorithm 2:

```
Calculating Cp using matrices
   for i=1:|(CWr)|
    if max(CWr(i,:)*Sr*Fv)>1   [(i,:) implies entire i^th row]
      Cp(i,:)=CWr(i,:)
    end
  end

Calculating Jdec(C)
   for i=1:|(CWr)|
     if max(CWr(i,:)*Sr*Fv)>1
      Jdec(i,:)=CWr(i,:)*Sr*Fv
     end
   end
      for i=1:|Jdec|
         for j=1:|Jdec|
           if Jdec(i,j)=1
              then Jdec(i,j)=0
           end
           if Jdec(i,j)>1
             then Jdec(i,j)=1
          end
        end
     end

Finding initial Cp(C)
   D=Cp*Sr*Fv
   m=1
   for i=1:|(CWr)|
      if max(and(D(i,:),Jdec(i,:)))==1
         Cp(C)((m,:))=CWr(i,:)
         m=m+1
      end
   end
```
■

After computing C_p, $J_{dec}(C)$ and the initial $C_p(C)$ using Algorithm 2, we use Algorithm 1 to compute $C_p(C)$ recursively. Note that Algorithm 1 only computes the set $C_p(C)$ for a CW C under consideration. Then, this set is converted to a single row by using union of all the rows of $C_p(C)$. The same procedure is followed for all the other CWs in the system using a simple 'for loop' to get the complete set $C_p(C)$ for all the CWs in the system.

D. Critical Siphons And Critical Subsystems In Matrix Form

In this section we use matrices to compute the PN objects i.e. Critical Siphons and Critical Subsystems required in Section III for deadlock analysis in FMRF systems.

$\bullet C$ *and* $C\bullet$ are the set of input and output transitions from a CW C. Once CW_r is constructed, we use Algorithm 2 and then Algorithm 1 to construct $C_p(C)$ for each $C \in CW_r$. Let $\bullet C_p(C)$ and $C_p(C)\bullet$ be the set of input and output transitions from $C_p(C)$. In matrix formulation, it is denoted as ${}_dC_p(C)$ and $C_p(C)_d$ respectively. It is computed as:

$${}_dC_p(C) = C_p(C) \otimes S_r \quad (15)$$

$$C_p(C)_d = C_p(C) \otimes F_r^T \quad (16)$$

To find the set of transitions $CW_t(C_p(C))$ between resources in the set $C_p(C)$, we use,

$$CW_t(C_p(C)) = {}_d C_p(C) \otimes C_p(C)_d \quad (17)$$

The critical subsystems $J(C_p(C))_0$ for a given CW_r are given by,

$$J(C_p(C))_0 = CW_t(C_p(C)) \otimes F_v \quad (18)$$

The siphon job sets $J(C_p(C))_+$ are given by,

$$J(C_p(C))_+ = {}_d C_p(C) \otimes (\overline{{}_d C_p(C) \otimes C_p(C)_d}) \otimes F_v \quad (19)$$

Now we have all the machinery to compute using matrices the objects required for deadlock avoidance using Theorem 3 and Lemma 10.

E. MAXWIP Dispatching Policy For Deadlock Avoidance In FMRF

Lemmas 8, 9, and 10 formulate the MAXWIP dispatching policy for deadlock avoidance in FMRF systems. According to this policy, CB in FMRF is avoided by limiting WIP in the critical subsystems of each CW C. This policy can be implemented as a real-time deadlock avoidance control scheme by controlling the firing of the precedence transitions of the critical subsystems.

V. EXAMPLES

We now illustrate the new notions developed in this paper, as well as the power of the matrix formulation, by computing them for two example FMRF. In the first example, deadlock can occur, while a small change in the structure yields Example 2, where deadlock can never occur.

Example 1: Consider Figure 3. This particular system contains three decision resources B1, B2 and B3. According to Section IV, the relevant matrices for calculating the siphon jobs and critical subsystems are F_r, S_r, F_v and S_v . F_v is a matrix of jobs required to fire transitions. S_v is a matrix of jobs started on firing of transitions. F_r is a matrix of resources required to fire transitions and S_r is a matrix of resources released when transitions are fired. These matrices for the system in Figure 3 are given by,

$$F_r = \begin{array}{c} \begin{array}{ccccccc} R1 & R2 & M1 & M2 & B1 & B2 & B3 \end{array} \\ \left[\begin{array}{ccccccc} 0&0&0&0&1&0&0\\ 1&0&0&0&0&0&0\\ 0&1&0&0&0&0&0\\ 0&0&0&0&0&1&0\\ 0&0&0&0&0&1&0\\ 0&0&1&0&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1\\ 1&0&0&0&0&0&0\\ 0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{array}\right] \end{array} \begin{array}{l} \\ t1\\t2\\t3\\t4\\t5\\t6\\t7\\t8\\t9\\t10\\t11\\t12\\t13 \end{array} \qquad S_r' = \begin{array}{c} \begin{array}{ccccccc} R1 & R2 & M1 & M2 & B1 & B2 & B3 \end{array} \\ \left[\begin{array}{ccccccc} 0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0\\ 1&0&0&0&1&0&0\\ 0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0\\ 0&0&0&0&0&1&0\\ 0&0&1&0&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1\\ 1&0&0&0&0&0&0\\ 0&1&0&0&0&0&0 \end{array}\right] \end{array} \begin{array}{l} \\ t1\\t2\\t3\\t4\\t5\\t6\\t7\\t8\\t9\\t10\\t11\\t12\\t13 \end{array}$$

$$F_v = \begin{array}{c} \begin{array}{ccccccccc} b1 & r1a & r2a & b2 & m1 & m2 & b3 & r1b & r2b \end{array} \\ \left[\begin{array}{ccccccccc} 0&0&0&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&1 \end{array}\right] \end{array} \begin{array}{l} \\ t1\\t2\\t3\\t4\\t5\\t6\\t7\\t8\\t9\\t10\\t11\\t12\\t13 \end{array} \qquad S_v' = \begin{array}{c} \begin{array}{ccccccccc} b1 & r1a & r2a & b2 & m1 & m2 & b3 & r1b & r2b \end{array} \\ \left[\begin{array}{ccccccccc} 1&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{array}\right] \end{array} \begin{array}{l} \\ t1\\t2\\t3\\t4\\t5\\t6\\t7\\t8\\t9\\t10\\t11\\t12\\t13 \end{array}$$

The set of CWs in this system using the binary string algebra [11] is given by:

$$CW_r = \begin{array}{c} \begin{array}{ccccccc} R1 & R2 & M1 & M2 & B1 & B2 & B3 \end{array} \\ \left[\begin{array}{ccccccc} 1&0&1&0&0&1&1\\ 1&0&0&1&0&1&1\\ 0&1&1&0&0&1&1\\ 0&1&0&1&0&1&1\\ 1&0&1&1&0&1&1\\ 1&1&1&0&0&1&1\\ 1&1&1&1&0&1&1\\ 1&1&0&1&0&1&1\\ 0&1&1&1&0&1&1 \end{array}\right] \end{array}$$

Each row in this set represents a CW e.g. the first row of this set represents the CW R1-M1-B1-B3. Using Algorithm 2 and 1, the set $C_p(C)$ for this system is computed as:

$$C_p(C) = \begin{array}{c} \begin{array}{ccccccc} R1 & R2 & M1 & M2 & B1 & B2 & B3 \end{array} \\ \left[\begin{array}{ccccccc} 1&1&1&1&0&1&1\\ 1&1&1&1&0&1&1\\ 1&1&1&1&0&1&1\\ 1&1&1&1&0&1&1\\ 1&1&1&1&0&1&1\\ 1&1&1&1&0&1&1\\ 1&1&1&1&0&1&1\\ 1&1&1&1&0&1&1\\ 1&1&1&1&0&1&1 \end{array}\right] \end{array}$$

We observe that in this case, each row is the same, which indicates that each CW in CW_r is related by the decision resources B2 and B3. This is why we end up with the same resources in each row of $C_p(C)$. In fact, all the CWs in CW_r are in the same equivalence class $C_p(C)$.

The siphon jobs and critical subsystems using Equations (18) and (19) are given by

$$J(C_p(C))_+ = \begin{array}{c} \begin{array}{ccccccccc} b1 & r1a & r2a & b2 & m1 & m2 & b3 & r1b & r2b \end{array} \\ \left[\begin{array}{ccccccccc} 0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&1&1 \end{array}\right] \end{array}$$

$$J(C_p(C))_0 = \begin{array}{c} \begin{array}{ccccccccc} b1 & r1a & r2a & b2 & m1 & m2 & b3 & r1b & r2b \end{array} \\ \left[\begin{array}{ccccccccc} 0&1&1&1&1&1&1&0&0\\ 0&1&1&1&1&1&1&0&0\\ 0&1&1&1&1&1&1&0&0\\ 0&1&1&1&1&1&1&0&0\\ 0&1&1&1&1&1&1&0&0\\ 0&1&1&1&1&1&1&0&0\\ 0&1&1&1&1&1&1&0&0\\ 0&1&1&1&1&1&1&0&0\\ 0&1&1&1&1&1&1&0&0 \end{array}\right] \end{array}$$

In this case, $J(C_p(C))_0 \cap J(C_p(C))_+ = \phi$. Hence, if all the jobs in $J(C_p(C))_0$ i.e. *r1a*, *r2a*, *b2*, *m1*, *m2*, and *b3* are marked, then according to Theorem 3, $C_p(C)$ is in CB and deadlock will occur.

Example 2: Figure 6 shows a small change in the PN structure of Figure 3. A new resource R3 is added to the system which performs the task *r3a*, i.e. R2 is not used for *r2a*.

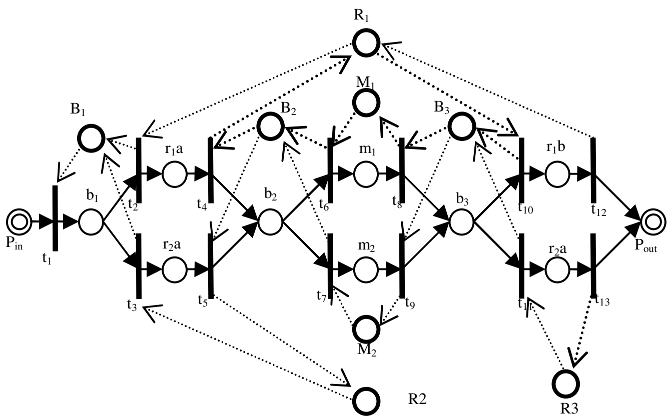

Figure 6. Case where $J(C_p(C))_0 \cap J(C_p(C))_+ \neq \phi$

Here, only the F_r and S_r matrices change. The matrices F_v and S_v remain the same. This is because a new resource is added to the system, but the tasks and their transitions remain the same. The addition of a new resource results in an additional column in the F_r and S_r^T matrices.

$$F_r = \begin{array}{c} \begin{array}{cccccccc} R1 & R2 & R3 & M1 & M2 & B1 & B2 & B3 \end{array} \\ \begin{bmatrix} 0&0&0&0&0&1&0&0 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{bmatrix} \end{array} \begin{array}{l} t1 \\ t2 \\ t3 \\ t4 \\ t5 \\ t6 \\ t7 \\ t8 \\ t9 \\ t10 \\ t11 \\ t12 \\ t13 \end{array} \quad S_r' = \begin{array}{c} \begin{array}{cccccccc} R1 & R2 & R3 & M1 & M2 & B1 & B2 & B3 \end{array} \\ \begin{bmatrix} 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0&0 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \end{bmatrix} \end{array} \begin{array}{l} t1 \\ t2 \\ t3 \\ t4 \\ t5 \\ t6 \\ t7 \\ t8 \\ t9 \\ t10 \\ t11 \\ t12 \\ t13 \end{array}$$

The corresponding CW_r and $C_p(C)$ matrices are given by:

$$CW_r = \begin{array}{c} \begin{array}{cccccccc} R1 & R2 & R3 & M1 & M2 & B1 & B2 & B3 \end{array} \\ \begin{bmatrix} 1&0&0&1&0&0&1&1 \\ 1&0&0&0&1&0&1&1 \\ 1&0&0&1&1&0&1&1 \end{bmatrix} \end{array} \qquad C_p(C) = \begin{array}{c} \begin{array}{cccccccc} R1 & R2 & R3 & M1 & M2 & B1 & B2 & B3 \end{array} \\ \begin{bmatrix} 1&0&0&1&1&0&1&1 \\ 1&0&0&1&1&0&1&1 \\ 1&0&0&1&1&0&1&1 \end{bmatrix} \end{array}$$

The siphon jobs and critical subsystem using Equations (18) and (19) are given by:

$$J(C_p(C))_+ = \begin{array}{c} \begin{array}{ccccccccc} b1 & r1a & r2a & b2 & m1 & m2 & b3 & r1b & r2b \end{array} \\ \begin{bmatrix} 0&0&0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&1&1&0 \end{bmatrix} \end{array}$$

$$J(C_p(C))_0 = \begin{array}{c} \begin{array}{ccccccccc} b1 & r1a & r2a & b2 & m1 & m2 & b3 & r1b & r2b \end{array} \\ \begin{bmatrix} 0&1&0&1&1&1&1&0&0 \\ 0&1&0&1&1&1&1&0&0 \\ 0&1&0&1&1&1&1&0&0 \end{bmatrix} \end{array}$$

We see that $J(C_p(C))_0 \cap J(C_p(C))_+ \neq \phi$. Here, $R3 \notin C_p(C)$ and therefore, the task $b3 \in J(C_p(C))_0$ and $b3 \in J(C_p(C))_+$. Hence, according to Theorem 2, deadlock can never occur.

VI. Conclusions

A theory for deadlock avoidance in Free Choice Multi-Reentrant Flow Lines (FMRF) is provided that extends known results for MRF. It is shown that the occurrence of decision places, which are followed by two or more resource paths, means that the usual notion of critical siphon used in MRF does not apply for FMRF. Therefore, we define a new notion of critical siphon for FMRF that is tied to deadlock avoidance. A MAXWIP dispatching policy is formulated for deadlock avoidance in FMRF systems. According to this policy, it has been shown that the work in progress (WIP) must be limited in the critical subsystems to avoid deadlock. A matrix formulation that efficiently computes the various PN objects required for deadlock avoidance in FMRF has been provided. This matrix formulation allows.fast and efficient numerical computation techniques to be applied to PN analysis.

References

[1] Ballal P., Giordano V., Lewis F., "Deadlock free dynamic resource assignment in multi-robot systems with multiple missions: a matrix-based approach," Proc. Mediterranean Conf. Control & Automation, Ancona, Italy, June 2006.

[2] Corman, T., Leisenson, C., and Rivest R., “Introduction to Algorithms,” Prentice Hall of India, 2001.

[3] Ezpeleta, J.; Colom, J.M.; Martinez, J., “A Petri net based deadlock prevention policy for flexible manufacturing systems,” IEEE Transaction on Robotics and Automation, Volume 11, Issue 2, April 1995 Page(s):173 – 184.

[4] Giordano V., Lewis F., Mireles J., Turchiano B., “Coordination control policy for mobile sensor networks with shared heterogeneous resources,” Proceedings of the IEEE International Conference on Control and Automation, Budapest, June 2005.

[5] Giordano V., Ballal P., Lewis F., Turchiano B., Zhang J. B., “Supervisory control of mobile sensor networks: math formulation, simulation, implementation,” IEEE Transactions on Systems, Man and Cybernetics, Part B, Volume 36, Issue 4, Aug. 2006 Page(s):806 – 819.

[6] Giordano V., Lewis F., Turchaino B., Ballal P., Yeshala V., “Matrix computational framework for discrete event control of wireless sensor networks with some mobile agents,” Proc. Mediterranean Conf. Control & Automation, Limassol, Cyprus, June 2005.

[7] Gurel, A., Bogdan S., Lewis F., “ Matrix Approach to Deadlock-Free Dispatching in Multi-Class Finite Buffer Flowlines,” IEEE Transactions on Automatic Control, Volume 45, Issue 11, Nov 2000 Page(s): 2086-2090.

[8] I. N. Herstein., “Topics in Algebra,” Blaisdell Publishing Co., 1964.

[9] Lawley, M. “Integrating Routing Flexibility and Algebraic Deadlock Avoidance Policies in Automated Manufacturing Systems,” International Journal of Production Research, 38(13); 2931-2950, 2000.

[10] Lewis F., Gurel A., Bogdan S., Docanalp A. Pastravanu O., “Analysis of deadlock and circular waits using a matrix model for flexible manufacturing systems,” Automatica, vol.34, no. 9, September 1998.

[11] Lewis F., "Wireless sensor networks," Smart environments:Technologies, Protocols, and Applications, ed. D. J. Cook and S. K.Das, John Wiley, New York, 2004.
[12] Mireles J., Lewis F., Gurel A., "Implementation of a deadlock avoidance policy for multipart reentrant flow lines using a matrix-based discrete event controller," Proceedings of the International symposium on advances in robot dynamics and control, New Orleans, November 2002.
[13] Murata, T. "Petri Nets: Properties, Analysis and Applications," Proceedings of the IEEE, vol.77, no.4, April 1989, pp.541-80.
[14] Ozmutlu, S. And Harmonosky, C.M., "A real time methodology for minimizing mean flowtime in FMSs with routing flexibility: Threshold based Alternate Routing," European Journal of Operational Research, 166 (2): 369-384, 2005.
[15] Peterson, J. L., "Petri Net Theory and the Modeling of Systems," Prentice-Hall, Englewood Cliffs, NJ, 1981.
[16] Wysk, R.A.; Yang, N.S.; Joshi, S.; "Detection of deadlocks in flexible manufacturing cells," IEEE Transactions on Robotics and Automation, vol.:7 , Issue: 6 , December 1991.
[17] Xing K.Y., Hu B.S., Chen H.X., "Deadlock Avoidance Policy for Petri-net Modeling of Flexible Manufacturing Systems With Shared Resources," IEEE Transactions on Automatic Control, Volume 41, Feb 1991 Page(s): 289-295.