Deadlock Analysis of Petri Nets Based on the Resource Share Places Relationship

Sanghwan Kim1), **Sangho Lee**1),
Computer Engineering Dept.1)
Chungbuk National University
Gaesin-dong 48, Cheungju
Chungbuk, Korea
sanghwan@dhc.co.kr, shlee@chungbuk.ac.kr,

Jongkun Lee2)[a]
LIS/Computer Engineering Dept.2)
Changwon National University
Salim-dong 9, Changwon,
Kyungnam, Korea
jklee@changwon.ac.kr

Abstract: The Deadlock problem in FMS, occurred by the relationship between more than two operations based on the resource share machines or robots, and buffers. Since a deadlock is a condition in which the excessive demand for the resources being used by others causes activities to stop, it is very important to detect and prevent a deadlock. In this paper, we study the problem of deadlock detection and avoidance by using the PN model based on the relationship of the resource share places for FMS (Flexible Management System). Also, we propose a deadlock detection conditions after analyzed the Petri Net used the transitive matrix. For presenting the results, the suggested deadlock detection and avoidance algorithms were also adapted to an illustrated model.

Keywords: avoidance, deadlock, Petri-nets, recovery net, transitive matrix.

Lee Jong-kun is a professor and Ph.D. supervisor at Changwon National University(CNU), Korea. He is also director of NEXUS center of CNU, and chairman of Kyungnam Smart home technique Forum in Kyungnam. Currently, his research interests include concurrent model and algorithm, Scheduling analysis in FMS, Mobile security and Petri nets theory.

Lee Sang-ho is a professor and Ph.D. supervisor at Chungbuk National University, Korea. Now, he is also chairman of Chungbuk e-Learning Innovation Forum. Currently, his research interests include Network Security, Privacy in Ubiquitous, and Petri net theory.

Kim Sang-hwan is a Ph.D. candidate at Chungbuk National University, Korea. He is also general manager of Daihan Calsonic Co., manufacturing air-condition system of Nissan Motor Company in Korea. His research interests include Petri net, Semantic Grid and workflow theory.

1. Introduction

"Deadlock status" is a status that is stopped a flow of a marking due to a wait for a resource and its confirmation and prevention is an important problem in a system. While deadlock problem is one of the important properties for analyzing the Petri nets, at the same time, it also can be one of the critical points in the scheduling problem of FMS (flexible Management System). Deadlock usually appears in contain subsystem which is run in parallel and resources share places. Therefore, it is important to develop efficient resource allocation policies which improve and optimize the system performances such as use of resource while preventing deadlock situation[16]. Most of technical literatures have been used Petri nets to derive deadlock prevention and avoidance algorithms [1-3, 6, 9-11, 13-19].

Deadlock avoidance methods are in many cases related to a set of sequential process sharing common resources of [15]. Xing et al. proposed a necessary and sufficient liveness condition based on the deadlock structure [16]. Moreover, J. park and S.A. Reveliotis [19] present deadlock avoidance policies for system with multiple resource acquisition and flexible routing, called conjunctive/disjunctive resource allocation system. Control places were also used for siphon control in [3], [17] and [18]. However, while the proposed studies have been achieved in an application of the DES (Discrete Event System) to need a real time process, most of the studies didn't show an ease understanding and application of a system included many processes.

The method for deadlock prevention introduced in this paper is based on the structural property that is considered to be based on Petri nets such as transition and place invariant. If we are able to explain the behavioral properties using by the transitive matrix, it will be easy to analyze the deadlock status in system. Since transitive matrix can explain all relations between the place and transitions in Petri nets, we have reported an algorithm [5, 11] to analyze scheduling in FMS using the transitive matrix after slicing resource share environment.

The object of this paper is to propose a method to analyze the deadlock and to avoid problem in Petri nets using the transitive matrix. Since this method may detect the deadlock status and avoid schedule in the

[a] Corresponding author

transitive matrix directly, it's easy to find the deadlock property.

This paper is organized as follows; some definitions of Petri Nets and transitive matrix are given in section 2. Deadlock detect conditions are presented in section 3. In section 4, we show a deadlock avoidance policy, and show an illustration model with one example in section 5.

2. Petri nets

A. Petri nets

In this section certain terms which are often used in the latter part of this paper are [4, 7, 8].

Definition 1: Let $PN = (PR, T, I, O, M_0)$ be a Petri nets,

PR = P $\cup$ R, where,

$P = \{p_1, p_2, \ldots\ldots, p_n\}$: a finite set of Work Places $(n \geq 0)$

$R = \{r_1, r_2, \ldots\ldots, r_m\}$: a finite set of Resources Sharing Places $(m \geq 0)$

$T = \{t_1, t_2, \ldots\ldots\ldots, t_k\}$: a finite set of Transitions $(k \geq 0)$

$P \cap R = \phi \quad P \cap T = \phi \quad R \cap T = \phi$

$\forall p \in P \qquad |\bullet p| = 1 \ \ and \ \ |p \bullet| = 1, \ \forall r \in R \qquad |\bullet r| \geq 1 \ \ and \ \ |r \bullet| \geq 1$

I : $\{PR\} \times T \to N$, I is an Input Function

O: $T \times \{PR\} \to N$, O is an Output Function

C(pr,t) = O(pr,t) – I(pr,t), C is the incidence matrix

$P \neq 0 \quad and \quad R \neq 0 \ \ and \quad T \neq 0$

$M_0 \in M = \{M | M : \{PR\} \to N\}$, M_0 is the initial marking

N : Set of nonnegative integers

Definition 2 [14]: A transition $t_j \in T$ in PN enabled if M $\geq$ I($\cdot$,t) and may fire yielding the marking M'=M+C($\cdot$,t). We write M[s> M' to denote that the enable sequence of transitions s may fire at M yielding M'. Finally, we denote so the sequence of null length.

B. Transitive matrix

We recall now some basic definitions of transitive matrix which are as used in [4, 5 , 11].

Definition 3: Invariant matrix

The matrix of PN structure, M_{PN} is M_{PN} = <PR', T, B^-, B^+>, where PR':P $\cup$ R is a set of working places and resource sharing places, and T is a set of transitions, respectively. B^- and B^+ are matrices of m rows by n columns defined by

B^- =[I,j] = #(PR_i, I(t_j)), matrix of input function,

B^+ =[I,j] = #(PR_i, O(t_j)), matrix of output function

Definition 4: S-Invariant

A column-vector x^T=(x_1,x_2,…,x_n) $\geq$ 0 of the homogeneous equation A x^T=ΔM =0 is called a T-invariant, where x^T is x's transpose. An integer solution y = (y1,y2,…,ym)^T of the transposed homogeneous equation Ay = 0 is called a S-invariant.

Definition 5: Transitive matrix

Let B^- and B^+ be an input function matrices and output function matrix of m rows by n columns in Petri nets, respectively.

The place and transition transitive matrix are as follows:

$M_{PR} = B^-(B^+)^T$: place transitive matrix

$M_T = (B^+)^T B^-$: transition transitive matrix

Definition 6: Let M_{PR} be the labeled place transitive matrix:

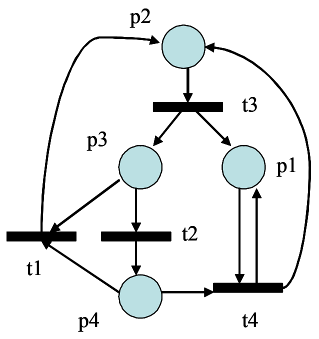
$$M_{PR} = B^- \, diag(t_1, t_2, \ldots, t_n)(B^+)^T ,$$

where $t_i (i = 1,2,...,n)$ is :

$$|t_i| = \begin{cases} 1 \text{ fire } t_i \\ 0 \text{ not fire } t_i \end{cases}$$

The elements of M_{PR} describe the directly transferring relation that is from one place to another place through one or more transitions.

Example:



$$M_{PR} = \begin{array}{cccc|c} p1 & p2 & p3 & p4 & \\ \hline t_4/2 & t_4/2 & 0 & 0 & p1 \\ t3 & 0 & t3 & 0 & p2 \\ 0 & t_1/2 & 0 & t_2 & p3 \\ t_4/2 & t_1/2, t_4/2 & 0 & 0 & p4 \end{array}$$

a) Example of Petri net **b) M_{PR} of Example**

Through introducing the m×m place transitive matrix, we can evaluate transition enabling firing, and calculate quantity and sequence of transition enabling firing.

Definition 7: Let a reachable marking $M_R(K+1)$ from $M(k)$ be an m-vector of nonnegative integer. The transformation defined by

$$\mathrm{M_R}(\mathrm{k}+1)^{\mathrm{T}} = \mathrm{M}(\mathrm{k})^{\mathrm{T}} M_{PR}$$

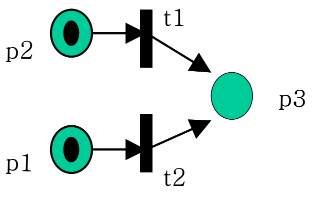
The elements of M_{PR} describe the direct transferring relation that is from one place to another place through one or more transitions.

Definition 8: Let M_{PR} be the labeled place transitive matrix. If a transition t_k appears s times in the same column of M_{PR}, then we will replace t_k in M_{PR} by t_k / s .

Definition 9: Weighted Place Transitive Matrix M_{PR}.

Let M_{PR} be the $m \times m$ weighted place transitive matrix. If a transition t_k appears s times in the same column of M_{PR}, then we will replace t_k by t_k / s in M_{PR}.
It is represented a control flow to a column direction of a transitive matrix. Therefore a control flow can be known through a value of $\sum (\frac{t_k}{s})_i$, where i is a number of places.

Example:



$$M_{PR} = \begin{bmatrix} 0 & 0 & t_1 \\ 0 & 0 & t_2 \\ 0 & 0 & 0 \end{bmatrix}$$

If this example net is able to be explained like as $M(k) = [P_1(k), P_2(k), P_3(k)]^T$, then we will obtain $\mathrm{M_R} = [0,0,t_1(P_1(k)) + t_2(P_2(k))]^T$.

In this example, the next marking may have maximum two tokens from transition t1 and t2: like as M_{PR} = [0, 0, 2].

Definition 10: Relation condition in column

Let $\#(pr_i, O(t_k))$ be a number of output value from p_i to transition t_k, and $\#(pr_i, E(t_k))$ be a number of input value from t_k to transition p_i.

A formal definition of the relation conditions in column are as follows.

1) $\#(pr_i, O(t_k)) = \#(pr_i, E(t_k)) \Rightarrow \sum \frac{t_{k_i}}{s} = 1$: It is maintained a control scope under a firing of the token t_k.

2) $\#(pr_i, O(t_k)) > \#(pr_i, E(t_k)) \Rightarrow \sum \frac{t_{k_i}}{s} > 1$: It is expanded a control scope under a firing of the token t_k.

3) $\#(pr_i, O(t_k)) < \#(pr_i, E(t_k)) \Rightarrow \sum \frac{t_{k_i}}{s} < 1$: It is reduced a control scope under a firing of the token t_k.

It is represented a firing possibility of each transition to a row in the transitive matrix. Also, a token exist in the resource common place of a row direction.

And we can know whether a firing possible or not through a value of $\sum \frac{t_{k_i}}{s}$ because the number of transitions is represented a row direction of a resource common place is the number of transitions is inputted at the place. So, if a value of $\sum (\frac{t_{k_i}}{s})_i$ is greater than 1, it is possible to firing.

Definition 11: Relation condition in row

Let $\#(pr_i, O(t_k))$ be a number of output value from p_i to transition t_k, and $\#(p_i, E(t_k))$ be a number of input value from t_k to transition p_i.

A formal definition of the relation conditions in column are as follows.

1) $\#(pr_i, O(t_k)) = \#(pr_i, E(t_k)) \Rightarrow \sum \frac{t_{k_i}}{s} \geq 1$

2) $\#(pr_i, O(t_k)) > \#(pr_i, E(t_k)) \Rightarrow \sum \frac{t_{k_i}}{s} \geq 1$

3) $\#(pr_i, O(t_k)) < \#(pr_i, E(t_k)) \Rightarrow \sum \frac{t_{k_i}}{s} \geq 1$

Therefore, we can know that the sum of transitions of a row direction of a transitive matrix is 1 in any case. If a value of $\sum (\frac{t_{k_i}}{s})_i$ is smaller than 1, we can know that the Petri net model mistake.

Definition 12: Deadlock relationship in M_{PR}

Let M_{PR} be a transitive matrix of PN = (PR,T,I,O,M), then PN is called a deadlock-free if a relationship in row(column) of resource share place $\sum (\frac{t_{k_i}}{s})_i \geq n,$ where n ≥1 and integer, else the PN is deadlock.

3. Deadlock fine algorithm in Petri net using the transitive matrix

A. Deadlock

The deadlocks are critical system parts for liveness analysis, because transitions may never be enabled again if they contain places of an unmarked deadlock in their input set of places. Hence, the liveness property means that every process can always be finished and that It is possible to start new process in the system to be executed [16].

In the literature [14], we can find two reasons have been identified as necessary conditions for the occurrence of deadlock: A resource cannot be used by two or more processes simultaneously or when a resource is being used, it is not released unless the process using it finishes with it. In this time, we can summary the deadlock definition like as follows:

Definition 13 [5]: Dead node

1) A transition of net will be dead at a marking M if it's not enabled at any marking reachable from M.
2) A place of a net will be dead at a marking M if it's not marked at any marking reachable from M.
Dead node is a node which has more than one dead transition or dead place.

Definition 14: Deadlock net
If net has more than one dead node, then this net has deadlock status.

Example:

For example, we consider one model like as follows:

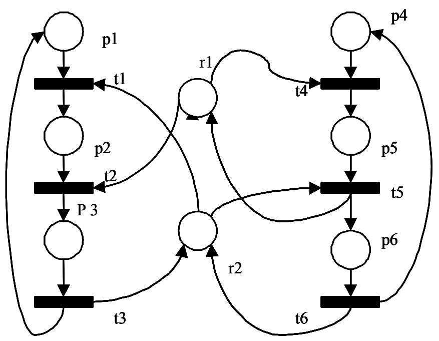
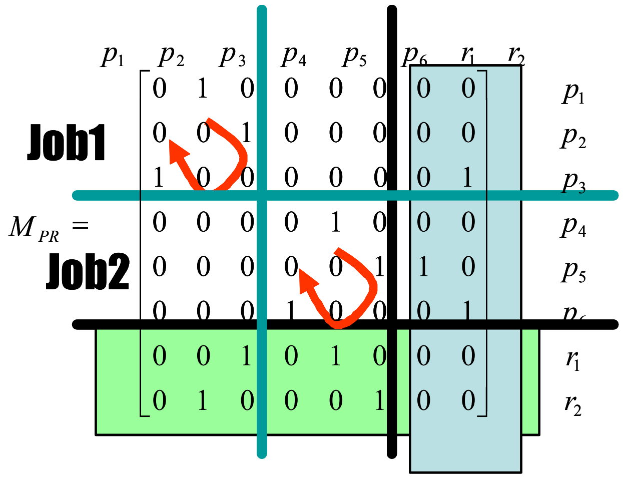


Figure 3.1 Example Petri Nets

In this net, two transitions t2 and t5 are deadlock statuses by the place r1 and r2.

Also based on the unfolding net such as figure 3.2, two deadlock transitions may be found.

Table 3.1 M_{PR} of figure 3.1



$$M_{PR} = \begin{array}{c|cccccccc|} & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & r_1 & r_2 \\ p_1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ p_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ p_3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ p_4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ p_5 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ p_6 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ r_1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ r_2 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}$$

In this M_{PR}, through definition 9 and 10, we can obtain a relationship like as follow table M_{PR}'.

Table 3.2 M'_{PR} of table 1

$M_{PR}'=$

p1	p2	p3	p4	p5	p6	r1	r2		Σ
0	1/2	0	0	0	0	0	0	**P1**	1/2
0	0	1/2	0	0	0	0	0	**P2**	1/2
1	0	0	0	0	0	0	1	**P3**	2
0	0	0	0	1/2	0	0	0	**P4**	1/2
0	0	0	0	0	1/2	1/2	0	**P5**	1
0	0	0	1	0	0	0	1	**P6**	2
0	0	1/2	0	1/2	0	0	0	**r1**	1
0	1/2	0	0	0	1/2	1/2	0	**r2**	3/2
1	1	1	1	1	1	1	2		Σ

In this table M_{PR}, we can find two jobs like as J1: p1-p2-p3 and J2:p4-p5-p6, and this relationship is explained one cycle. Also, we can find some relationship between the resource share places r1 and r2:

Case of r1:

-relation condition in row : 1/2+1/2+1 = 2

-relation condition in column : 1/2+1/2=1

Case of r2:

-relation condition in row : 1/2+1/2+1/2 = 3/2

-relation condition in column: 1+1=2

In r2, relation condition of row is not equal of 1, this means that the number of token is more less than out and it's may be occurred a deadlock status. In r1, it has all conditions of row and column are satisfied deadlock-free. Finally, this net is deadlock.

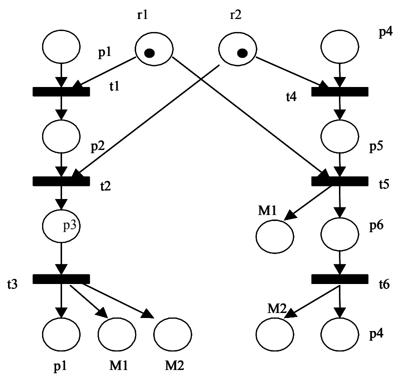


Figure 3.2 Unfolding net of Figure 3.1

By the definition 8, we can consider a reachable marking as follows:
Mo =[1,0,0,1,0,0,1,1], where [p1,p2,p3,p4,p5,p6,r1,r2].

Then, the reachable marking M_1 is :

M_1 = Mo• M_{PR}, then,
M_1 = [0, (p1(t1/2) + r2(t1/2)), r1(t2/2), 0, (p5(t4/2) + r1(t4/2)), r2(t5/2), r2(t5/2), 0].

Since the number of token should be explained by integer value such as equal or greater than zero, we can explain the reachable marking M_1 as follows:

M_1 = [0, 1, 0, 0, 1, 0, 0, 0].

Then next reachable marking is :

$M_2 = M_1 \bullet M_{PR}$
$= [0,0, p2(t2/2),0,0,p5(t5/2),0,0]$
$=[0, 0, 0, 0, 0, 0, 0, 0]$.

In this result, we can find deadlock status occurred in transitions t2 and t5 by the places r1 and r2.

Definition 15: Deadlock in M_{PR}

Let a reachable marking $M_R(K+1)$ from $M(k)$ be an m-vector of nonnegative integer. The transformation defined by $\mathrm{M_R}(\mathrm{k}+1)^{\mathrm{T}} = \mathrm{M}(\mathrm{k})^{\mathrm{T}} M_{PR}$
If $M_R = 0$ then this net is deadlock at marking M_{R-1}.

B. Deadlock Detection Algorithm

We propose a deadlock detection algorithm based on the previous section.
Algorithm: deadlock fine

Input: N = <P,T,F,M>

Output: N is deadlock free or not

(1) Define M_{PR} of a Petri net initial N.

(2) Find all relation of resource share places in each column M_{PR}.

(3) Calculate M_R from the initial marking.

(4) If they have one deadlock node then this net N is deadlock net but if not this net is deadlock free.

4. Deadlock Avoidance Policy

A. Deadlock Avoidance

Since the deadlock status occurs at resource share node, the resource share nodes having deadlock status need a token each node for recovering the deadlock status.

Definition 13: deadlock avoidance policy

Let $\exists pi,pj \in P$ be the deadlock nodes at transition ti and tj, respectively ,where $\exists$ ti,tj $\in$ T, and these places pi and pj are the resource share places, pi $\neq$pj and ti $\neq$tj, $\cdot$ti =pi, $\cdot$tj = pj. Then, add one input arc to pj at transition ti, and add one arc to pj at transition tj.

$ti\cdot = O(pi,ti) \cup O(pj,ti)$
$tj\cdot = O(pj,tj) \cup O(pi,tj)$

Example:

As shown in figure 3.1, we consider a system with 2 machines such as r1 and r2, there are two types jobs such as JOB_1 and JOB_2. Since the location of JOB1 does not overlap with that of JOB2 and the starting points exist in each job, JOB1 and JOB2 can be completed at once. The recovery deadlock net is summarized in figure 4.1 and a transitive matrix of this example in table 4.1.

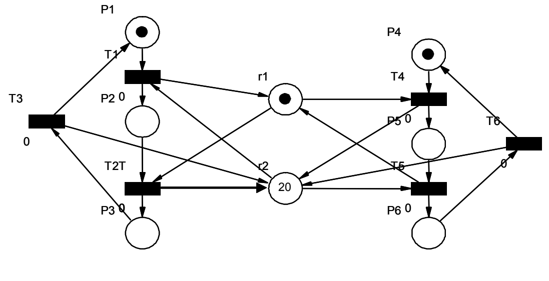


Figure 4.1 Recovery deadlock net

Table 4.1 Transitive matrix M_{PR}' of figure 4.1

M_{PR}'=

p1	p2	p3	p4	p5	p6	r1	r2		Σ
0	1/2	0	0	0	0	1/2	0	**P1**	1
0	0	1/2	0	0	0	0	1/2	**P2**	1/2
1	0	0	0	0	0	0	1	**P3**	2
0	0	0	0	1/2	0	0	1/2	**P4**	1
0	0	0	0	0	1/2	1/2	0	**P5**	1
0	0	0	1	0	0	0	1	**P6**	2
0	0	t2/2	0	1/2	0	0	1/2 1/2	**r1**	2
0	1/2	0	0	0	1/2	1/2 1/2	0	**r2**	2
1	1	1	1	1	1	2	4		**Σ**

In this M_{PR}', we can obtain a reachable marking as follows:

Case of r1:

-relation condition in row: 1/2+1/2+1/2+1/2=2
-relation condition in column: 1/2+1/2+1/2+1/2 = 2

Case of r2:

-relation condition in row:
1/2+1+1/2+1+1/2+1/2=4
-relation condition in column:
1/2+1/2+1/2+1/2 = 2

In this case, places p2, p5, r1 and r2 have a token. We can say that this marking is fire and deadlock free.

B. Algorithm for the Avoidance of the Deadlock Status

The procedures for the detection and the avoidance of the deadlock status using the resource status table in this section 3.1 can be illustrated in the following algorithm.

Algorithm:

Input: N = <P,T,F,M>

Output: N is deadlock free or not

(1) Define M_{PR} of a Petri net initial N.

(2) make a marking sequence from the initial place and resource share place.

(3) if it's not satisfy enable then define that deadlock status and deadlock node.

(4) add one arc to input deadlock status node for another resource share place.

(5) verify the marking sequence with modified M_{RP}.

(6) If this marking sequence accepts fire conditions then this net will be avoided deadlock status and stopped, else this is not deadlock status avoidance, repeat (1) to (4) until satisfy (6).

5. Illustration Model

An FMS example introduced in [12] is represented in Fig. 5.1. There are 3 machines (M1, M2 and M3), one robot and two transport devices, also four operations Job 1, Job 2, Job 3 and Job 4.

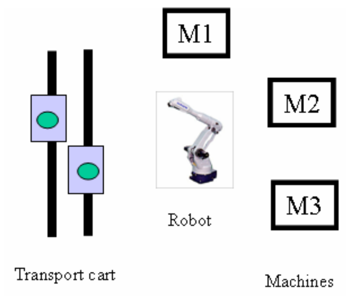


Figure 5.1. System structure

Let’s assume that the production ratio is 1/4, which means that the goal is to manufacture 25% of Job1 1, Job 2, Job 3 and Job 4. We define that the incorporate alternative process plans as follows:

Job 1: {M1, M2, M3}

Job 2: {M2, M1, M3}

Job 3: {M1, M2, M3}

Job 4: {M3, M2, M1}

The PN representation of system is as follows:

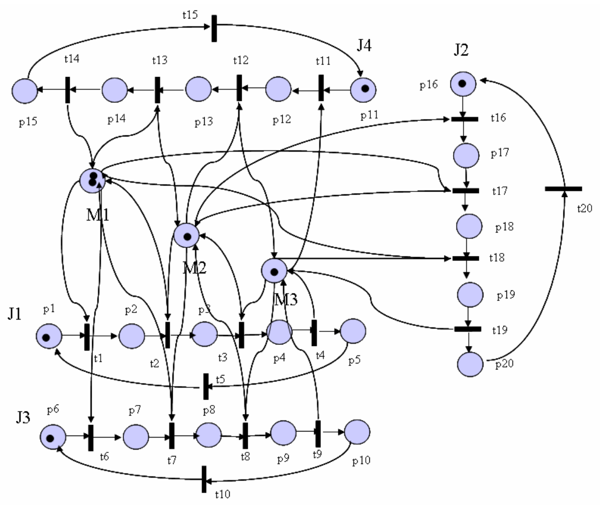


Figure 5.2. Illustrative example: 3 machines and 4 jobs

First, the initial tokens are in M1, M2, M3, p1, p6 and p12. In this case, places p2, p12, p17 have a token and M1, M2 and M3 have not any token for fire. We can say that this marking is dead at M_1.

Based on the avoidance algorithm, we can get an avoidance PN and transitive matrix like as follows:

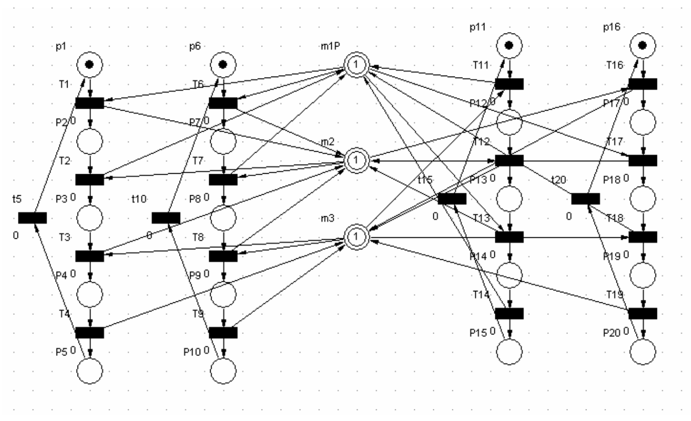


Figure 5.3. Avoidance model of Fig.4.2

Table 5.2. Transitive matrix of Fig.4.3

$$M_{PR}=\begin{bmatrix}
0 & t1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t1/2 & 0 \\
0 & 0 & t2/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t2/2 & 0 & 0 \\
0 & 0 & 0 & t3/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t3/2 & 0 \\
0 & 0 & 0 & 0 & t4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t4 \\
t5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & t6/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t6/2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & t7/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t7/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t8/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t8/2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t9 \\
0 & 0 & 0 & 0 & 0 & t10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t11/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t11/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t15 & 0 & t12/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t12/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t13/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t13/2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t14 & 0 & 0 & 0 & 0 & 0 & t14/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t16/2 & 0 & 0 & 0 & 0 & 0 & t16/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t17/2 & 0 & 0 & 0 & t17/2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t18/2 & 0 & t18/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t19 & 0 & 0 & t19 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & t1/2 & 0 & 0 & 0 & 0 & t6/2 & 0 & 0 & 0 & 0 & 0 & 0 & t13/2 & 0 & 0 & 0 & t17/2 & 0 & 0 & 0 & \begin{matrix} t1/2 \\ t6/2 \\ t17/2 \\ t13/2 \end{matrix} & 0 \\
0 & 0 & t2/2 & 0 & 0 & 0 & 0 & t7/2 & 0 & 0 & 0 & 0 & t12/2 & 0 & 0 & 0 & t16/2 & 0 & 0 & 0 & \begin{matrix} t2/2 \\ t7/2 \end{matrix} & 0 & \begin{matrix} t12/2 \\ t16/2 \end{matrix} \\
0 & 0 & 0 & t3/2 & 0 & 0 & 0 & 0 & t8/2 & 0 & 0 & t11/2 & 0 & 0 & 0 & 0 & 0 & 0 & t18/2 & 0 & \begin{matrix} t11/2 \\ t18/2 \end{matrix} & \begin{matrix} t3/2 \\ t8/2 \end{matrix} & 0
\end{bmatrix}$$

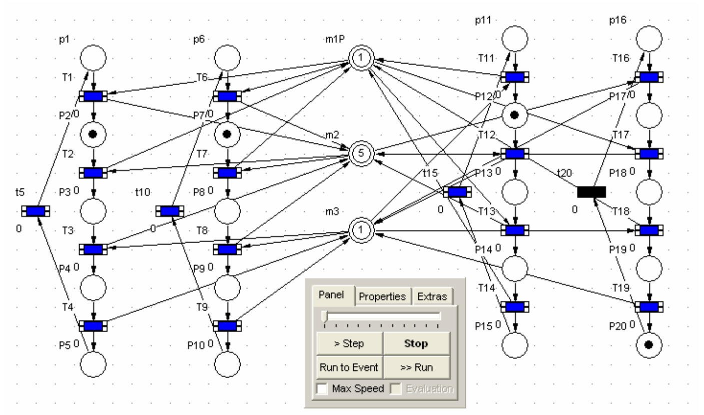


Figure 5.4. Running process of Figure 5.3

In this figure, we can find that modified model has not deadlock status and this model is bounded. Also, we use Visual Object Net ++ (Evaluation Version 1.44.2) to verify the proposed model is not deadlock.

B. Algorithms for the Avoidance of the Deadlock Status

The procedures for the detection and the avoidance of the deadlock status using the resource status table in Chapters 5.1 can be illustrated in the following algorithm.

Algorithm:

Input: N = <P,T,F,M>

Output: N is deadlock free or not

(1) Define M_{PR} of a Petri net initial N.

(2) make a marking sequence from the initial place and resource share place.

(3) if it’s not satisfy enable then define that deadlock status and deadlock node.

(4) add one arc to input deadlock status node for another resource share place.

(5) verify the marking sequence with modified M_{PR}.

(6) If this marking sequence accepts fire conditions then this net will be avoided deadlock status and stopped, else this is not deadlock status avoidance, repeat (1) to (4) until satisfy (6).

6. Conclusions

In this paper, we focused on the avoidance policy of the deadlock problem in Petri nets using the transitive matrix. The transitive matrix explained all relationship between the places and transitions in the net. By these relationships, we can find deadlock status based on the relation between the resource share places. Finally, it can be said that this method is very easy to find the deadlock status and modify the deadlock status in the net. In this work, we showed very simple example for guaranty about to find and to avoidance deadlock in ordinarily Petri nets. In near future the studies to find and avoidance the deadlock problem in General Petri nets using the transitive matrix and to deadlock avoidance problem on the cyclic scheduling in FMS will be published.

REFERENCES

1. CORBETT, J.C. (1996), **Evaluating Deadlock detection methods for concurrent software**, IEEE tr. Sof. Eng. Vol. 22(3), pp. 161-180.
2. DAMASCENO, BC. and XIE, X.(1999), **Petri nets and deadlock-free scheduling of multiple-resource operations**, In IEEE SMC'99, pp. 878-883.
3. EZPLETA, J., COLOM, J.M., MARTINEZ, J. (1995), **A Petri net based deadlock prevention policy for flexible manufacturing systems**, IEEE tr. Robotics and Automat., Vol. 11, no. 2, pp. 173-184.
4. LIU, J., ITOH, Y., MIYAZAWA, I., SEIKIGUCHI, T., (1999). **A Research on Petri nets Properties using Transitive matrix**, In: Proceeding IEEE SMC99, pp. 888-893.
5. LEE, J., KORBAA, O. (2006). **Scheduling analysis of FMS**: An unfolding time Petri nets approach, mathematics and Computer simulation, 70, pp. 419-432.
6. MELZER, S. and ROMER, S. (1997). **Deadlock checking using net Unfoldings**, In Proc. of the Conf. on Computer-Aided Verfication (CAV'97).
7. MURATA, T. (1989). **Petri Nets: Properties, Analysis an Applications**, Proceedings of the IEEE, 77(4), IEEE, USA, pp. 541-580.
8. PETERSON, J.L.V. (1981), **Petri Net Theory and the Modeling of Systems**, Englewood Cliffs, NJ: Prentice-Hall, Inc.
9. SHATZ, S.M., TU, S., MURATA, T. (1996). **An application of Petri net reduction for Ada Tasking Deadlock Analysis**, In: IEEE Trans. on Parallel and Distributed Systems, Vol. 7, No. 12, pp. 1307-1322.
10. XIONG, H. H., ZHOU, M.C. (1997), **Deadlock free scheduling of an automated manufacturing system based on Petri nets**, In IEEE ICRA'97, pp. 945-950.
11. LEE, J. (2004). **Deadlock find algorithm using the Transitive Matrix**, In: Proceeding CIE'04.
12. A. GIUA, et elc., (2004), **Observer-Based state-feedback control of timed Petri nets with deadlock recovery**, In: IEEE Trans. on Automatic control, Vol. 49, No. 1, pp. 17-29.
13. X. KE-YI, HU BS, CHEN HX (1996), **Deadlock Avoidance Policy for Petri net Modeling of Flexible Manufacturing Systems with Shared resources**, In: IEEE Trans. on Automatic control, Vol. 41, No. 2, pp. 289-295.

14. BASIL, F., GIUA, A. SEATZU, C. (2003), **Observer-based state-feedback control of time Petri nets with deadlock recovery: theory and implementation**, In Proceed. Of Symp. On Discrete events in Industrial and manufacturing systems, CESA 2003.

15. BANAZAK, Z. A. and KROGH, B. H. (1990), **Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows**, IEEE tr. Robotics and Automat., Vol. 6, no. 6, pp. 724-734.

16. XING, K., HU, B. AND CHEN, H. (1995), **Deadlock avoidance policy for flexible manufacturing systems**, Petri Nets in Flexible and Agile Automation, Kluwer, Boston, MA, pp. 239-263.

17. BARKAOUI, K., ABDALLAH, I.B. (1995), **Deadlock avoidance in FMS base don structure theory of Petri nets**, In proc. IEEE symp. Emerging Technologies for factory Automation, pp. 499-510.

18. LI, Z. W., ZHOU, M. C. (2004), **Elementary siphons of Petri Nets and their application to deadlock prevention in flexible manufacturing system**, IEEE tr. on system, Man and Cybern., part A, Vol. 34, pp. 38-51.

19. PARK, J., REVELIOTIS, S. A. (2001), **Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible manufacturing**, IEEE tr. Robotics and Automat., Vol. 46, pp. 1572-1583.