

자원 공유 플레이스의 관계행렬을 이용한 효율적인 교착상태 확인 정책

김중욱, 차정원, 이종근
창원대학교 컴퓨터공학과
{jwkim,jcha,jklee}@changwon.ac.kr

EFFICIENT DEADLOCK DETECTION POLICY BASED ON THE RESOURCE SHARE PLACES IN PETRI NETS

Jongwoog Kim, Jeongwon Cha, Jongkun Lee
Computer Engineering Dept. Changwon National University
{jwkim,jwcha,jklee}@changwon.ac.kr

요 약

여러 개의 작업이 동시에 작동 할 때 서로 다른 작업에서 특정한 로봇이나 장치를 사용하기 위하여 상대방의 작업이 끝나기를 기다리는 상태를 교착상태라 하며 생산공정에서는 효율성과 경제성 제고에 가장 중요한 문제 중 하나이다. 지금까지 이러한 교착상태 확인은 도달성(reachability)기법이나 서브넷(Subnet)으로 분리하여 분석하는 기법 등이 가장 많이 사용 되었으나, 분석에 필요한 시간과 노력, 복잡성과 효율성에 있어서 효과적이지 못한 단점을 가지고 있다. 본 연구는 페트리넷(Petri Net)에서 모든 플레이스(Place) 간의 관계를 나타내는 추이(Transitive)행렬을 이용하여 교착상태가 나타날 가능성이 있는 자원공유 플레이스(Place)를 분석하여 교착상태를 검증하는 정책을 제시한다. 이를 위하여 교착상태를 확인할 수 있는 자원공유 플레이스의 관계를 정의하고, FMS(Flexible Manufacturing System) 모델을 이용하여 제안한 알고리즘의 성과를 검증한다.

1. 서 론

유연생산시스템(FMS)은 machine, 이동로봇, 그리고 원활한 작업진행을 위한 자원의 배분을 수행하는 여러 절차로 구성된 파트 등을 이용하여 여러 종류의 아이템을 생산하는 생산 시스템이다. 특히 교착상태는 자원을 기다리며 공정이 중단된 상태므로 FMS의 스케줄링 문제에 있어 매우 중요한 포인트가 되었다. 이런 이유로 교착상태를 회피할 수 있는 효과적인 FMS 제어정책이 필요하다[17]. 시스템에서 교착상태의 방지 및 확신은 매우 중요한 문제이기 때문에 교착상태 문제를 해결하기 위해 분석 및 회피방법이 여러 연구자에 의해 제안된 결과[1-3,5,6,13-19] graph model, finite state machines model 과 페트리넷 모델(PetriNet model)을 기반으로 하는 몇 가지 기법이 채택되었다. 페트리넷 분석모델에서는 siphon 분석과 [5,13,15,17,20] 도달성 그래프(reachability

graph) 분석방법이 사용되었다. 특히 siphon 분석기법은 초기 마킹(marking)과 연관된 아크(arc)를 초기 모델에 추가하는 제어 플레이스 추가를 통해 구현되었는데 이러한 초기모드에서 siphon 특성을 분석하고 찾기 위한 노력이 필요하다. 도달성 그래프 기법은 만약 초기 모델이 크다면 도달성 그래프는 초기 모델보다 더 복잡하고 큰 모델을 획득하게 된다. 인접행렬을 이용한 알고리즘으로 DAPN[21]이 있는데, 이는 인접 행렬을 작성하고 작업별로 행렬을 분리하여 토큰의 흐름을 분석하여 교착상태를 확인하는 기법으로 알고리즘이 복잡한 단점을 가진다. 추이행렬은 페트리넷에서의 플레이스와 트랜지션(transition)의 모든 관계를 쉽게 표현할 수 있으므로 본 논문에서는 교착상태를 쉽게 탐지하기 위하여 추이행렬기반의 페트리넷의 자원공유 플레이스와 일반적 플레이스 간의 구조적 속성을 정의한 후 교착상태 문제를 분석하는 기법을 제안한다.

페트리넷의 정의와 추이행렬의 요약이 2장에 기술되었고, 교착 상태를 찾기 위한 예제가 3장에 기술되었다. 4장에서 사례모델을 통하여 제안 알고리즘을 검증하였다.

2. 페트리넷

페트리넷은 일반적으로 시스템의 모델링에 쓰이며, 다음과 같이 정의할 수 있다 [4,7,8,11,12].

페트리넷 PN 은 5-튜플로 구성된다

PN = (P, T, I, O, Mo)
 P = {p₁, p₂, ..., p_n} : 플레이스의 유한 집합,
 T = {t₁, t₂, ..., t_k} : 트랜지션의 유한 집합,
 I(t_j ∈ T → I(t_j) ∈ P) : 트랜지션의 입력 함수,
 O(t_j ∈ T → O(t_j) ∈ P) : 트랜지션의 출력 함수,
 Mo: P → {0, 1, 2, 3, ...} : 초기 마킹 P ∩ T = ∅
 트랜지션 t ∈ T 는 M에서 점화가능하며 이를 M[t> 로 표현한다, iff ∀p ∈ •t: M(p) > 0 가 된다. 만일 M[t> 이면 트랜지션 t 는 점화가능하며, 새로운 마킹 M'을 가지며, M[t> M', with M'(p) = M(p) - 1 로 표시된다, 만일 p ∈ •t; M'(p) = M(p) + 1 만일 p ∈ t; 그렇지 않으면 M(p) = M'(p), for ∀ p ∈ P.
 초기 마킹 Mo 에서 도달 가능한 마킹의 집합은 R(PN, Mo) 로 표시된다.

(PN, Mo) 는 4-튜플 PN = <P, T, I, O, M> 를 갖는 페트리넷이라고 하자.

트랜지션 t ∈ T는 ∀M ∈ R(PN, Mo), ∃M' ∈ R(PN, M), M'[t> 이면 마킹 Mo로부터 생존(live) 가능하다. PN 이 만일 ∃t ∈ T, Mo[t> 이면 생존 불가하다. 만일 ∀M ∈ R(PN, Mo), ∃t ∈ T, M[t> 이면 (PN, Mo) 는 교착 자유상태(deadlock free)이다. 만일 ∀t ∈ T, ∃M ∈ R(PN, Mo), M[t> 이면 (PN, Mo) 는 부분 생존하다. 만일 ∀t ∈ T, t 가 Mo로부터 생존하면 (PN, Mo) 는 생존한다.

만일 ∃k ∈ N, ∀M ∈ R(PN, Mo), ∀p ∈ P, M(p) ≤ k 이면 (PN, Mo) 유한 한다.

PN 구조의 행렬 C는 C = <P, T, B⁻, B⁺> 이라 하면, 여기서 P = {p₁, p₂, ..., p_m} 는 플레이스의 유한집합, T = {t₁, t₂, ..., t_n} 는 트랜지션의 유한집합, B⁻ 와 B⁺ 는 m 열과 n 행의 행렬로 다음과 같이 정의된다:

B⁻ = [I, j] = # (P_i, I(t_j)), 입력함수 행렬,
 B⁺ = [O, j] = # (P_i, O(t_j)), 출력함수행렬.
 또한, B = B⁺ - B⁻ 는 유사(incidence)행렬이라 한다.

2.1 추이적 행렬

추이 행렬에 대한 일반 정의를 정리해 보면 다음과 같다[7-9].

동차방정식 A x^T = ΔM = 0 에서 행 벡터(vecor) x^T = (x₁, x₂, ..., x_n) ≥ 0 는 T-상태불변(T-invariant)라 하며, 여기서 x^T 는 x'의 변형이다.

하나의 정수 값 y = (y₁, y₂, ..., y_m)^T 가 변형동차방정식에서 존재한다면 Ay = 0 는 S-상태불변(S-invariant)라 한다.

이 식을 활용하여 플레이스 추이 행렬을 다음과 같이 얻을 수 있다 :

$$B = B(B^+)^T: \text{플레이스 추이행렬}$$

이제 L_{BP} 를 추이 행렬이라고 하면:

$$L_{BP} = B^- \text{diag}(t_1, t_2, \dots, t_n)(B^+)^T,$$

여기서 t_i (i = 1, 2, ..., n) 는 :

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

L_{BP} 의 요소는 하나의 플레이스에서 다른 플레이스와의 관계를 트랜지션을 이용하여 직접적으로 표현 되어 진다.

L_{BP}^{*} 를 라벨추이행렬이라 할 때, 만일 어떤 트랜지션 t_k 가 같은 행에서 s 번 나타나게 된다면 우리는 t_k / s 로 표시한다.

추이 행렬에서 열 벡터는 플레이스 p_i에게 행 벡터의 플레이스에서 토큰이 입력되는 관계를 나타낸 것이며 행 벡터는 플레이스 p_i에서 열 벡터의 플레이스로 토큰이 출력되는 관계를 나타내는 것이다. 지금의 추이 행렬은 기본 페트리넷으로 나타낸 모델에만 적용되므로 모든 토큰의 기본 개수는 1이며 복수개의 토큰이나 아크는 고려하지 않는다.

추이적 행렬의 행 방향은 제어흐름을

표현한다. 따라서 $\Sigma(t_k/s_i)$ 의 값에 따라 제어흐름을 알 수 있다.

만약 $\#(p_i, O(t_k)) < \#(p_i, E(t_k))$ 이면, 해당 플레이스의 병행적인 흐름이 유지되거나 확장되지 않고, 축소됨을 뜻한다.

이를 유형별로 정리하면 다음과 같다.

각각의 $\Sigma(t_k/s_i)$ 에 대해,

(1) $\#(p_i, O(t_k)) = \#(p_i, E(t_k))$ 인 경우, : 토큰 tk의 점화로 인한 제어 영역이 그대로 유지된다.

(2) $\#(p_i, O(t_k)) < \#(p_i, E(t_k))$ 인 경우, $\Sigma(t_k/s_i) > 1$: 토큰 tk의 점화로 인한 제어 영역이 넓어지게 된다.

(3) $\#(p_i, O(t_k)) > \#(p_i, E(t_k))$ 인 경우, $\Sigma(t_k/s_i) < 1$: 토큰 tk의 점화로 인한 제어 영역이 좁아지게 된다.

추이적 행렬의 열방향은 각 트랜지션의 점화 가능성을 표현한다. 또한, 열방향의 자원 공유 플레이스에는 토큰이 존재한다.

그리고 자원 공유 플레이스의 열방향에 표현된 트랜지션의 수는 그 플레이스로 입력되는 트랜지션의 수를 말하므로 $\Sigma(t_k/s_i)$ 의 값에 따라 해당 트랜지션의 점화 여부를 알 수 있는데, $\Sigma(t_k/s_i)$ 의 값이 1이면 점화가 가능하다.

이를 유형별로 정리하면 다음과 같다.

각각의 $\Sigma(t_k/s_i)$ 에 대해,

(1) $\#(p_i, O(t_k)) < \#(p_i, E(t_k))$ 인 경우 $\Sigma(t_k/s_i) = 1$

(2) $\#(p_i, O(t_k)) < \#(p_i, E(t_k))$ 인 경우 $\Sigma(t_k/s_i) = 1$

(3) $\#(p_i, O(t_k)) > \#(p_i, E(t_k))$ 인 경우 $\Sigma(t_k/s_i) = 1$

따라서, 추이적 행렬의 열 방향의 트랜지션의 합은 어떤 경우라도 1이 됨을 알 수 있다.

2.2 교착상태

교착 상태란 시스템의 흐름에서 서로가 대기상태가 되어 마킹의 흐름이 정지 될 경우 이를 교착상태라 한다. 또한 이러한 상태는 주로 자원공유 플레이스에서 많이 발생 되는 성질이 있다. 즉 자원공유 플레이스에서의 입출력 토큰의 수가

음수이거나 정수가 되지 못하면 자력으로 점화가 불가하므로 교착상태 가능성이 있다.

성질1:교착 노드

1) 마킹 M 으로부터 어떠한 트랜지션으로 점화가능하지 못하다면, 그 트랜지션은 교착 트랜지션이라 한다.

2) 마킹 M 으로부터 어떠한 플레이스로 마킹이 전이가 되지 못하면 그러한 플레이스를 교착플레이스라 한다.

성질2:교착 상태

페트리넷에서 교착 노드가 하나 이상 나타나면 마킹 M 은 교착상태라 한다.

성질 3: $Rc(p_i)$ 를 행의 플레이스 p_i 의 토큰 수의 합이고, $Rr(p_i)$ 를 열 플레이스 p_i 의 토큰 수의 합이라고 하면:

$$Rc(p_i) = \sum_{i=1}^n (p_i) = \sum_{i=1}^n (f \bullet p_i),$$

$$Rr(p_i) = \sum_{i=1}^n (p_i) = \sum_{i=1}^n (f \bullet p_i),$$

여기서 n 은 플레이스의 수,

f 는 L_{BP}^* 의 함수

만일 $\exists t$ 이면 $f=1$ 아니면 $f=0$.

그러면, $Dr(p_i) = Rr(p_i) - Rc(p_i)$ 은 플레이스 P_i 의 교착상태 조건이라 한다.

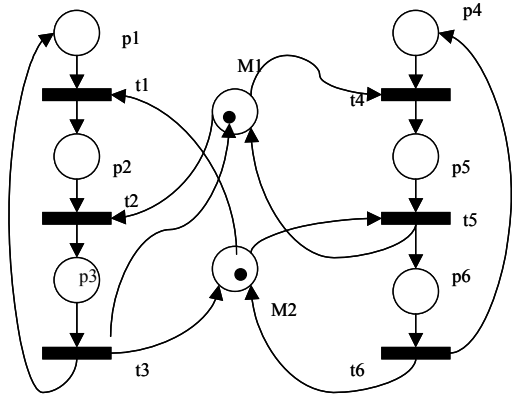
여기서, $Dr(p_i) = k$, 만일 $k \geq 0$ 이고 k 가 정수이면 이 PN 은 교착자유상태이고 그렇지 않으면 교착상태이다.

증명: 추이적 행렬은 기본 페트리넷을 기초로 작성 되어지는(입력과 출력이 각각 1이며 플레이스의 토큰도 1이어야 한다) 플레이스 행렬이므로 모든 행과 열의 합이 정수이어야 하며, 음수가 되면 점화가 불가하게 된다. 따라서, 정수이며 음수가 아닌 토큰의 합이어야 자유교착상태라 하겠다.

3. 추이 행렬을 이용한 교착 상태

3.1 교착상태 확인

교착상태는 자원공유 플레이스에서 발생하게 된다. 그림 3.1 과 같은 예에서 처럼, 두 개의 트랜지션 t_2 과 t_5 는 플레이스 M_1 과 M_2 에서 교착 상태가 발생된다. 즉 자원 공유플레이스 M_1 과 M_2 가 각각 트랜지션 t_1 과 t_4 에서 점화를 하면 트랜지션 t_2 와 t_5 에서 M_1 과 M_2 의 토큰을 기다리는 교착상태가 발생하게 된다. 그러나, M_1 이 t_4 에서 점화를 하지 않고 t_1 이 점화 완료 될 때까지 기다린다면 t_2 는 점화가 가능하다. FMS 에서는 동시처리를 원칙으로 고려하므로 대기시간의 발생도 교착상태 발생으로 본다.



(그림 3.1) 예제 페트리넷

	P1	p2	p3	p4	p5	p6	M1	M2	
L_{BP}^*	0	$t1/2$	0	0	0	0	0	0	P1
	0	0	$t2/2$	0	0	0	0	0	P2
	$t3$	0	0	0	0	0	$t3$	$t3$	P3
	0	0	0	0	$t4/2$	0	0	0	P4
	0	0	0	0	0	$t5/2$	$t5/2$	0	P5
	0	0	0	$t6$	0	0	0	$t6$	P6
	0	0	$t2/2$	0	$t4/2$	0	0	0	M1
	0	$t1/2$	0	0	0	$t5/2$	$t5/2$	0	M2

- 1) 자원공유 플레이스 M_1 :
 $Rr(M_1) : 1 + 1/2 + 1/2 = 2,$
 $Rc(M_1) : 1/2 + 1/2 = 1$
 $Dr(M_1) = 2 - 1 = 1$
- 2) 자원공유 플레이스 M_2 :
 $Rr(M_2) : 1 + 1 = 2,$
 $Rc(M_2) : 1/2 + 1/2 + 1/2 = 1(1/2)$
 $Dr(M_2) = 2 - 1(1/2) = 1/2$

3.2 교착자유상태

이 절에서는 교착자유상태의 예를 살펴본다 (그림 3.2).

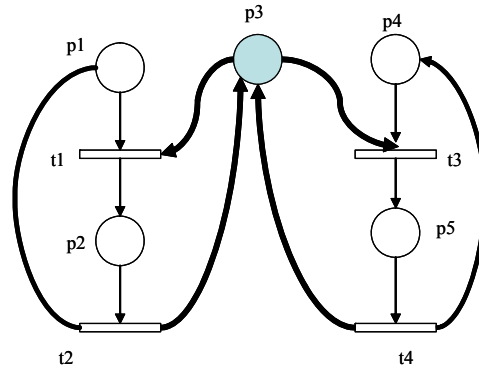


그림 3.2 교착자유상태 모델

표 3.2 그림 3.2의 M_{PR}

	P1	p2	p3	p4	p5	
L_{BP}^*	0	$t1/2$	0	0	0	P1
	$t2$	0	$t2$	0	0	P2
	0	$t1/2$	0	0	$t3/2$	P3
	0	0	0	0	$t3/2$	P4
	0	0	$t4$	$t4$	0	P5

다음과 같이 교착상태 조건식을 자원공유 플레이스 p_3 을 중심으로 정리하면:

$$Rr(p_3) = 1 + 1 = 2,$$

$$Rc(p_3) = 1 + 1 = 2$$

$$Dr(p_3) = 2 - 2 = 0$$

따라서, 교착자유상태이다

3.3 교착상태 감지 알고리즘

앞 절서 보인 교착상태 감지 알고리즘을 정리하면 다음과 같다:

Algorithm: deadlock fine

Input: $N = \langle P, T, F, M \rangle$

Output: N is deadlock free or not

(1) Define L_{BP}^* of a Petri net initial N .

- (2) Find all relation of resource share places in each column. L_{BP}^*
- (3) Find all relation of resource share places in each row L_{BP}^* .
- (4) Calculate $Dr(p_i)$: deadlock condition using the L_{BP}^*
- (5) Repeat (2)-(4) for all resource share places.
- (6) If $\sum_{i=1}^n Dr(p_i) = 0$ then this net is deadlock free, but if not then find Dr have negative places or not, if have negative place then this net N is deadlock.

$$L_{BP}^* = \begin{bmatrix} 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 1 & 1 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

이제 자원공유 플레이스 M1과 M2를 살펴보면:

- 1) 플레이스 M1:
 $Rr(M1) = 1+1=2,$
 $Rc(M1) = 1/2+1/2+1/2=1(1/2)$
 $Dr(M1) = 2 - 1(1/2) = 1/2$
- 2) 플레이스 M2:
 $Rr(M2) = 1+1= 2$
 $Rc(M2) = 1/2+1/2=1$
 $Dr(M2) 2-1= 1$

4. 사례연구

일반적인 FMS 의 사례[20]를 이 절에서는 살펴 봄으로 제안 한 알고리즘을 검증하여 본다 (그림 4.1). 여기에는 2 개의 기기 M1 과 M2 가 존재하며, 한 개의 로봇과 두개의 이동 포트가 있다. 또한 두 개의 작업 Job 1 과 Job 2 가 존재한다. 이러한 작업의 내용을 정리하면 다음과 같다:

- Job 1: {M1, M2}
- Job 2: {M2, M1}

또한 페트리넷 모델링은 다음과 같다, 특히 사례 모델의 추이 행렬은 표 4.1 과 같다:

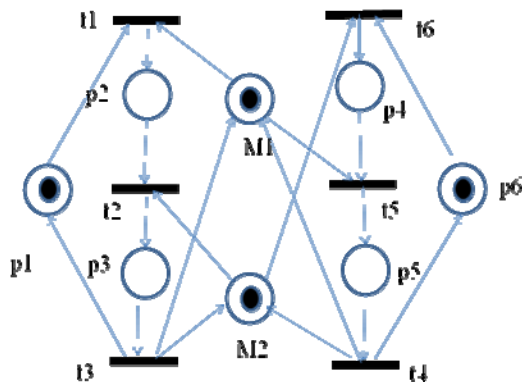


그림 4.1 사례모델

표 4.1 그림 4.1의 L_{BP}^*

플레이스 M1은 M2와 달리 조건식이 정수가 아니므로 전체 PN은 교착상태이다.

5. 결론

본 논문에서 추이 행렬을 기반으로 하여 페트리넷의 교착상태 문제 분석과 추이행렬을 기반으로 하는 자원공유 플레이스의 교착상태 조건을 제안하였다. 교착상태 문제는 충돌 플레이스에 기반한 두개 이상의 트랜지션 관계에서 발생하였다. 그러므로 입력 토큰의 수가 출력 토큰의 수 보다 작으면 공유 플레이스에서 교착상황이 발생하였고, 충돌 플레이스내 입력토큰과 출력토큰의 관계를 이용한 교착상태 조건을 제안하였다. 또한 추이행렬을 이용하여 교착자유상태와 교착 상태를 찾기 위해 몇 개의 예제에 적용하였다. 제안된 교착상태 탐지 기법은 다른 논문에 비해 간단하며 쉽게 적용할 수 있음을 제시하였다. 향후 추이행렬을 이용해 일반적 페트리넷 모델로 확장을 하고, FMS의 cyclic scheduling에서의 교착상태회피로 연구를 확장해 나갈 것이다.

참고문헌

- [1]Corbett JC, "Evaluating Deadlock detection methods for concurrent software," International Journal of IEEE Tr. Sof. Eng. 22(3), 1996
- [2] J.Park, S.A.Reveliotis, "Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible manufacturing," IEEE Tr. Robotics and Automat., Vol.46, pp.1572-1583,2001
- [3]Damasceno BC. And Xie X., "Petri nets and deadlock-free scheduling of multiple-resource operations," In Proc. of International Conference of IEEE SMC'99 ,878-883,1999
- [4]Desel J. And Esparza J., "Free Choice Petri nets," Cambridge univ. Press,1995
- [5]Ezpleta J.,Colom JM, Martinez J., "A Petri net based deadlock prevention policy for flexible manufacturing systems," International Journal of IEEE Tr. Robotics and Automat., 11(2),173-184,1995
- [6] A.Giua,et etc., "Observer-Based state-feedback control of timed Petri nets with deadlock recovery," In: IEEE Tr. on Automatic control, Vol. 49, No. 1, pp. 17-29,2004
- [7]Liu J.,Itoh Y., Miyazawa I., Seikiguchi T., "A Research on Petri nets Properties using Transitive matrix," In Proc. of 1999 International Conference on IEEE SMC99,888-893,1999
- [8] Lee J., Korbaa O., "Scheduling analysis of FMS:An unfolding time Petri nets approach," mathematics and Computer simulation, 70, pp. 419-432,2006
- [9] Lee J., "Deadlock find algorithm using the Transitive Matrix," In Proc. of CIE'04,2004
- [10] F.basil, A.Giua, C.Seatzu., "Observer-based state-feedback control of time Petri nets with deadlock recovery: theory and implementation," In Proc. Of Symp. On Discrete events in Industrial and manufacturing systems, CESA2003,2003
- [11]Murata T., "Petri Nets: Properties, Analysis an Applications," In Proc. of the IEEE, 77(4), 541-580,1989
- [12]Peterson JL, Petri Net Theory and the Modeling of Systems, Englewood Cliffs, NJ : Prentice-Hall, Inc.,1987
- [13] Li ZW, Zhao M, "On Controllability of Dependent Siphon for Deadlock prevention in Generalized Petri nets," IEEE Tr. SMC-A, vol.38,no.2,pp.369-384,2008
- [14]Han Z., Lee G.,"Application of Petri nets for deadlock analysis and avoidance in flexible manufacturing systems," Int J. Adv. Manu. Tech., vol .25, pp.735-742,2005
- [15]Xiong HH, Zhou MC, "Deadlock free scheduling of an automated manufacturing system based on Petri nets," In Proc. of the 1997 International Conference on IEEE ICRA'97, 945-950,1997
- [16]Yoon H. and Lee D., "Deadlock-Free Scheduling for Automated Manufacturing cells," In Proc. of International Conference on Control, Automation Robotics and Vision 2000,2000
- [17] Uzam M. and Zhou MC, "An Iterative Synthesis Approach to Petri net-based deadlock prevention Policy for Flexible manufacturing System," IEEE Tr. SMC-A, vol. 37, no.3, pp.362-371,2007
- [18] Li ZW, Hu HS,Wang AR, "Design of Liveness-Enforcing Supervisors for Flexible manufacturing Systems Using Petri nets," IEEE Tr. SMC-C,vol.37,no.4,pp.517-525,2007
- [19] Z.W.Li, M.C.Zhou, "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.38-51,2004
- [20]Jeng ND, and DiCesare F., "Synthesis using resource Control Nets for Modeling Shated-resource Systems," IEEE Tr. On Robotics and Automation, Vol. 11, no. 3, pp.317-327,1995
- [21]송유진,이종근,"DAPN과 인접행렬을 이용한 교착상태 회피에 대한 연구",한국시물레이션학회 논문지,제15권,1호,pp.1-10,2006