

추이적 행렬을 이용한 유연생산시스템의 순환 스케줄링 분석

이종근*

A Study of Cyclic Scheduling Analysis in FMS Based on the Transitive Matrix

Jongkun LEE

Abstract

The analysis of the cyclic scheduling problem in FMS using the transitive matrix has been proposed. Since the transitive matrix may explain all the relations between the places and transitions, we propose an algorithm to get good solution after slicing off some subnets from the original net based on machines operations. For analyzing the schedule problem, we considered two time functions such as produce time and waiting time using the P-invariant. In addition, we are shown the effectiveness of proposed algorithm after comparing with unfolding algorithms.

Key Words : BUC, Cyclic scheduling problem, FMS, Petri nets, Transitive matrix

* 창원대학교 컴퓨터공학과

1. 서론

유연생산시스템에서의 주요한 과제중의 하나는 경제성을 추구하는 최적화의 작업 환경을 구축할 수 있는 스케줄링을 구축하는 것이다. 이러한 스케줄링의 분석을 위하여 많은 학자들이 패트리 넬을 이용하여 스케줄링 알고리즘을 제시하였었다[1-7,9-11,20,21]. 특히 순환 알고리즘이란, 한정된 환경에서의 반복적 일정을 제시함으로써 생산의 최적화를 꾀하며 작업 시간의 단축을 꾀하는 작업 형태의 하나이다[11]. 이러한 순환 스케줄링 분석 알고리즘은 크게 세 가지의 형태로 구분이 가능하다. Hillion [3]은 작업의 수행 가능성 비율을 기초로 한 발견적 방법(heuristic algorithm) 모델을 제시하였고 Korbaa [6,7]는 기존의 작업 일정을 다시 재그룹화 하여서 작업 환경 WIP (Work in Process)을 고려한 순환 알고리즘을 제시하였다. Unfolding 분석 알고리즘[9,10,11,12]은 본 연구팀과 EC Lill의 연구팀이 공동으로 제시한 분석 알고리즘으로 패트리 넬의 모델을 자원 공유의 기기를 중심으로 여러 개의 서브 넬으로 (BUC: Basic unit Control) 분리하여서 각각의 서브 넬을 unfolding 기법으로 분석하여 최적의 스케줄링을 얻은 후 이를 종합화 하는 기법이다. 특히 추이적 행렬식[8,17,18,19]에서 BUC를 구하였으며, 자원공유 분석을 위한 unfolding 알고리즘을 제시하였다.

Unfolding 알고리즘에서는 자원공유별로 BUC를 분리하여서 기기작업시간 중심으로 작업시간과 최소대기 시간을 기준으로 분리된 BUC를 다시 unfolding 기법을 적용하여서 최적치를 구하는 방법을 선택하였다. 따라서, 최소대기 시간의 경우 충분한 사이클작업 시간을 반영하지 못하는 단점이 있으며, unfolding을 통하여서 실행 가능한 스케줄을 구하는 것 역시 위의 단점을 보완하기 위한 보안적인 방법으로 간주된다. 그러나, 최소 대기 시간을

활용함으로써 항상 최적치의 결과를 얻지 못하는 단점이 있다.

본 연구에서는 이러한 단점들을 보완하는 P-invariant를 이용한 대기시간을 산출하여서 전체사이클 작업시간을 대기함수로 활용하며, 또한 치환을 통하여 최적치를 구하는 기법을 새로이 제안한다. Unfolding 알고리즘에서는 추이적 행렬, unfolding 분석과 분할, 치환분석과 종합 같이 4단계의 과정을 추이적 행렬을 이용한 분할을 통하여 그 결과를 치환을 통하여 시행 가능한 모든 스케줄을 분석 및 종합할 수가 있어, 분석 과정을 3단계로 단순화 하여 분석 시간이 절약되고 알고리즘의 자동화에 있어서도 용이한 장점이 있다.

본 연구의 구성은 2장에서 사용되어질 패트리 넬의 기본적인 정의들과 추이적 행렬을 정의하고 3장에서는 추이적 행렬을 이용한 모델 분할을 위한 알고리즘을 제시한다, 4장에서는 추이적 행렬을 이용한 모델 분할을 이용하여 유연생산 시스템의 스케줄링 분석하고 unfolding 분석기법과 비교 분석한다. 마지막으로 결론과 앞으로의 연구과제에 대하여 5장에서 다룬다.

2. 타임패트리 넬과 추이적 행렬

2.1 타임 패트리 넬과 모델 분할

이 절에서는 일반적인 타임 패트리 넬의 성질들을 서술하여 본다 [1-4,10-13,16,17].

TPN은 $\langle P, T, E, S, Mo, \tau \rangle$ 인 6가지 튜플로 구성되며, $P = \{p_1, p_2, \dots, p_n\}$, $|P| \neq 0$, P 는 플레이스의 유한집합, $T = \{t_1, t_2, \dots, t_m\}$, $|T| \neq 0$, T 는 트랜지션의 유한집합, $P \cap T = \emptyset$, $E: P \times T \rightarrow N$, E 는 입력 함수,

$S: T \times P \rightarrow N$, S 는 출력함수(N : 양의 정수집합), $Mo \in M = \{M | M: P \rightarrow N\}$, Mo 는 초기토큰상태,

τ 는 시간함수: $\tau:T \rightarrow Q^*x(Q^*U^\infty)$ (Q^* 는 양의 정수집합)

트랜지션 t_j 의 입력 플레이스의 P_i 수를 $\#(P_i, I(t_j))$ 로 표시하고, 또한 트랜지션 t_j 의 출력 플레이스 P_i 의 수를 $\#(P_i, O(t_j))$ 로 표시한다.

패트리 넬의 행렬식 C 는 $C = \langle P, T, B^-, B^+ \rangle$, 여기서 P, T 는 각각 플레이스와 트랜지션의 집합을 나타낸다. B^- 와 B^+ 는 m 열과 n 행의 행렬식을 의미하는데, 다음과 같이 정리된다.

$$B^- = [i, j] = \#(P_i, I(t_j)), \text{ 입력함수의 행렬,}$$

$$B^+ = [i, j] = \#(P_i, O(t_j)), \text{ 출력함수의 행렬.}$$

$x \in P \cup T$ 인 노드의 선행집합은 $x^* = \{y \in P \cup T; (x, y) \in F\}$, 그리고 후행집합은 $^*x = \{y \in P \cup T; (y, x) \in F\}$ 이다.

트랜지션 T 와 마킹열 σ 에 대하여 $\sigma \in T^*$ 이고 $T' \subset T$ 이면, $\sigma|_{T'} \in T'^*$ 는 $T \setminus T'$ 와 같이 전체 트랜지션 집합에서 T' 을 제외한 부분집합을 얻게 된다.

만일 마킹열 $\sigma, \sigma' \in T^*$ 이고, 만일 σ' 가 σ 의 선행 마킹열이면 $\sigma' \leq \sigma$ 로 표시한다. 즉, 만일 $\sigma'' \in T^*$ 이면, $\sigma', \sigma'' = \sigma$ 이다. 만일 $\sigma \in T^*$ 이면, $M \rightarrow M'$ 로 표시하며 $M[\sigma > M']$ 로도 표현한다.

패트리 넬 N 은 다음의 성질들을 만족하면 보존성이 있다: 만일 $k \in N$ 이고 모든 플레이스 $p_i \in P$ 에서 모든 도달가능한 마킹 M 에서, $M \leq R(N)$, $M(p_i) \in k$ (만일 $k=1$ 이면 넬의 마킹은 안전하다고 한다) 이다. 여기서, $R(N)$ 은 N 의 도달 가능한 마킹이다.

한 트랜지션 t 는 만일 다음의 성질을 만족한다면 N 에서 생존성이 있다: 만일 모든 마킹 $M \leq R(N)$ 에서, $M' \leq R(N)$ 이면 마킹 M' 은 트랜지션 t 에서 점화 가능하다. 만일 $\forall t \in T$ 가 생존성이 있다면, 타임 패트리 넬 N 은 생존성이

있다.

(정의. 2.1) 사건 및 상태불변

행 벡터 $x^T = (x_1, x_2, \dots, x_n) \geq 0$ 는 $Ax^T = \Delta M = 0$ 의 동차방정식으로 사건 불변(T-invariant) 이라 한다, 여기서 x^T 는 x' 의 변형이다. $Ay = 0$ 의 변형적 동차방정식이 갖는 정수형 해 $y = (y_1, y_2, \dots, y_m)^T$ 는 상태 불변(S-invariant)이다.

일반적인 상태불변은 음의 값으로 표현되기도 하며 여러 상태 불변들이 합해져 하나의 상태불변을 이루기도 한다.

2.2 추이적 행렬

추이적 행렬이란 플레이스와 트랜지션간의 관계를 행렬로 표시함으로 초기 토큰을 가진 플레이스를 통하여 마킹의 흐름을 파악 할 수 있는 행렬이다. 앞 절에서 제시한 입력함수의 행렬과 출력함수의 행렬을 이용하여 플레이스 행렬식과 트랜지션 행렬식을 정의 할 수가 있다[13,17].

(정의. 2.2) [13]: 플레이스 행렬식 B_P 와 트랜지션의 행렬식 B_T 은 각각 다음과 같이 표현한다

$$B_P = B^-(B^+)^T$$

$$B_T = (B^+)^T B^-$$

(정의 2.3):[17] L_{BP} 을 추이적 인지 행렬(the labeled place transitive matrix) 이라 하면 :

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

L_{BP} 의 요소는 하나 이상의 트랜지션에서 하나 이상의 플레이스로 점화되어지는 직접적 관계성을 표시 하고 있다.

(정의 2.4)[13]: L_{BP}^* 를 추이적 행렬이라 한다. 만일 트랜지션 t_k 가 L_{BP} 에서 같은 열 행렬에 s 번 나타난다면, 이러한 t_k 를 L_{BP} 에서는

t_k/s 로 표시한다. 기타의 경우에는, $L_{BP}^*[i, j]$ 는 L_{BP} 와 같은 표현을 갖는다.

3. 분할 모델 알고리즘

패트리 넬에서의 제어 흐름은 토큰의 흐름에 의하여 기인한다고 정의 할 수가 있다. 하나의 제어적 흐름은 독립적인 토큰의 상태를 표현한다고 할 수가 있으며 또한 트랜지션의 점화에 따라 토큰이 여러 개로 분할되면 제어 흐름도 같은 수 만큼 분할 되어 진다. 따라서, 이러한 기본 적인 개념을 이용하여 병행적 기본단위 (the basic unit of concurrency : BUC)라 정의한다. 특히 자원 공유가 많은 유연 생산 시스템의 경우, 이러한 자원공유 모델을 하나의 BUC로 분류하여 분할 한다[10,11,18,19].

(정의 3.1): 병행적 기본 단위(BUC)

BUC 는 패트리 넬 모델에서 하나의 자원 공유(상태불변)에 관련 되어진 서브 넬이다. BUC의 구성은 다음과 같다 :

- 1) 자원공유 플레이스
- 2) 관련 트랜지션의 집합
- 3) 자원공유 플레이스와 관련 되어진 플레이스의 집합

따라서, BUC 는 자원공유수 만큼 발생하게 된다.

(예제) BUC

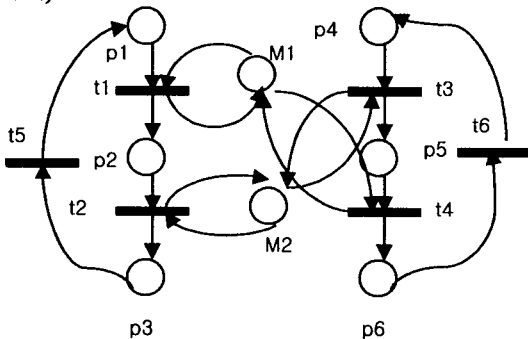
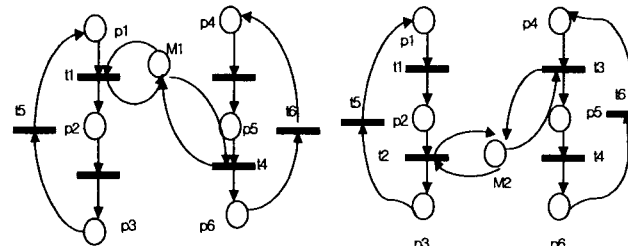


Fig. 1 Example model

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

Fig.2 Transitive matrix

모델은 두개의 자원공유: M1 과 M2 를 갖는 시스템 이며, 두 개의 BUC 를 구 할 수가 있다.



(1) BUC of M1

(2) BUC of M2

Fig.3 BUC

3.1 분할 알고리즘

앞 절에서 우리는 구조적 병행성을 추이적 행렬을 이용하여 서브 넬을 분할하는 개념을 제시하였다. 이 절에서는 이러한 추이적 행렬에서 분할하는 처리 과정을 다음과 같이 정리 한다:

- (1) 추이적 행렬 L_{BP}^* 를 작성한다.
- (2) 자원공유 플레이스를 선택한다
- (3) 열에서 해당 트랜지션과 플레이스를 선택 한다.
- (4) 행에서 선택된 플레이스의 트랜지션과 관

런 플레이스를 선택한다.

(5) (4)를 반복한다, 처음의 자원공유플레이스까지 도착하도록.

(6) (5)에서 초기의 플레이스를 찾지 못하면 중단한다(P-invariant가 없음).

(7) 다음의 자원공유 플레이스를 선택하여 (3)-(6)를 자원공유 플레이스가 없을 때까지 반복한다.

분할 알고리즘을 이용하여 분할 되어진 모델분할은 기존의 패트리 넬 축소나 변형 기법과는 달리 감축되거나 합성 되어지는 플레이스나 트랜지션이 없이 모든 플레이스와 트랜지션이 존재하므로 다음과 같은 성질을 통하여 기존의 성질 유지성을 검증 할 수가 있다.

(정의 3.2) BUC의 합성

$BUC_1 = (P_1, T_1, F_1, M_1, \tau_1), BUC_2 = (P_2, T_2, F_2, M_2, \tau_2), \dots, \text{et } BUC_n = (P_n, T_n, F_n, M_n, \tau_n)$, 를 분할알고리즘으로 얻은 n 개의 BUC 의 패트리 넬이라 하자. BUC의 합은 패트리 넬이다:
 $BUC \cup BUC_2 \cup \dots \cup BUC_n = ((P_1 \cup \dots \cup P_n), (T_1 \cup \dots \cup T_n), (F_1 \cup \dots \cup F_n), (M_1 \cup \dots \cup M_n), (\tau_1 \cup \dots \cup \tau_n))$.

(정리. 3.1) N 을 타임 패트리 넬이라고하고, S를 분할 알고리즘을 이용하여 얻은 BUC의 합성이라고 하자. 그러면, $N \supseteq S$.

(증명) $S = \langle P_s, T_s, F_s, M_s \rangle$ BUC의 합성이라고 하고, $N = \langle P, T, F, M \rangle$ 을 패트리 넬이라고 하면: 플레이스 p 는 P_s 의 한 원소라면 $\exists t_s \in T_s$, 따라서 $p \in \bullet t_s$ 그리고 BUC의 정의에 의하여, P_s (또는 T_s) 는 분할 알고리즘으로 얻은 플레이스의 집합(또는 트랜지션) 이다, 즉 $P_s \subseteq P$ (ou $T_s \subseteq T$). 따라서, 다음과 같이 정의 할 수가 있다, 만일 $p \in \bullet t_s$ 그러면 $p \in \bullet t \in T$, 또 만일 $p \in t_s \bullet$ 이면 $p \in t \bullet \in T$. 그러므로, $N \supseteq S$.

4. 추이적 행렬을 이용한 유연생산시스템의 스케줄링 분석

유연생산시스템(Flexible Manufacturing System: FMS)이란 다품종 소량 생산을 그 특징으로 하는 제조 시스템이다. 이와 같은 생산 시스템의 구성은 대개 각종 CNC machine, 치공구, 공정간의 이동을 위한 장비들의 종합적인 net 으로 되어 있다. 이와 같은 유연생산 시스템에 있어서 가장 중점적으로 다루는 것은 작업효율의 최적화를 위해 시스템을 구성하는 기기들의 작업 스케줄을 최적화하는 것이다. 특히 유연생산 시스템에서는 작업의 효율성과 경제성을 위하여 자원공유의 현상이 많다. 따라서, 이러한 환경에서의 최적 스케줄링 분석이 절대적으로 필요하다. 이 장에서는 앞서 제시한 추이적 행렬을 이용한 모델 분할 기법을 이용하여 유연생산시스템에서의 순환 스케줄링을 분석하고자 한다.

4.1 최적 스케줄링 분석

스케줄링 문제에서 최적해란 가능한 빠르게 최적의 시간의 작업 흐름을 계산하여 내는 것이 중요하다. 따라서, 이 장에서는 모델분할에서 최적의 예상가능 시간에 의한 스케줄링을 산출하여 이를 종합화 하기 위한 기본적인 산술식에 대하여 생각하여 본다.

모델 분할 S 는 다음과 같이 3 개의 관련된 함수 $f(S), g(t_i)$ 그리고 $h(t_i)$ 를 갖는데, 이를 통하여 가장 효율적인 스케줄링시간(makespan time)을 산출한다. 함수 $f(S)$ 는 초기 마킹 m_0 에서부터 트랜지션 t_i 를 지나 도달 마킹까지의 전체 예상가능시간을 나타낸다. 이러한 전체 예상가능시간 함수 $f(S)$ 는 다시 두 개의 함수 $h(t_i)$ 와 $g(t_i)$ 로 구성되는데, 예상 도달 시간함수 $h(t_i)$ 는 트랜지션 t_i 까지의 예상도달시간으로 트랜지션 t_i 는 포함되지 아니한다, 한편 예상가능 시간함수 $g(t_i)$ 는 트랜지션 t_i

에서부터 도달 마킹까지의 예상가능시간을 나타낸다. 따라서, 이러한 함수들의 관련식은 다음과 같이 종합화 할 수가 있다.

$$f(S) = h(t_i) + g(t_i)$$

따라서, 모델 분할에서의 전체예상가능시간 함수 $f(S)$ 는 다음과 같이 변형 될 수가 있다 [11]:

$$f(S) = \text{Max}_{1 \leq k \leq n} \left\{ \left(\sum_{i=1}^k h(t_i) \right) + \left(\sum_{j=i+1}^n g(t_j) \right)_k \right\}$$

여기서,

- n 은 작업별 트랜지션의 수,
- k 는 각 기기의 자원공유 수

가능 스케줄 S 는 여러 개가 발생 가능하므로, 최적화 되어진 스케줄은 전체예상가능시간 함수 $f(S)$ 의 최소화를 통하여 구할 수가 있다.

$$\text{optimize } S = \min (f(S_i))$$

최적 해를 구하기 위하여서는, 예상가능시간 함수 $h(m)$ 의 효율성을 고려 할 수 있다. 이를 위하여서는 도달 가능시간과 이에 필요한 토큰의 수(자원 공유 수), 즉 트랜지션의 점화에 따른 토큰의 양을 가지고 그 효율성을 산출 할 수가 있다.

$d(m_i)$ 를 도달 가능시간의 효율 도수라 한다면, 효율 도수는 다음과 같이 산출 된다.

$$d(m_i) = \frac{h(m_i)}{n(m_i)}$$

여기서,

- $h(m_i)$ 는 m_i 의 작업가능 시간,
- $n(m_i)$ 는 m_i 의 토큰의 합 (자원공유수).

따라서, 가능시간 효율도수 $d(m_i)$ 는 모델분할 되어진 벨에서의 효율적인 스케줄을 선정하는데 사용 되어지는 한 요소가 될 수가 있다.

각 작업의 흐름과 시간을 이용하여 치차의 최적화 수량을 산출 할 수가 있다[7].

$$WIP = \sum_{\text{pallets type } i} \left| \frac{\sum \text{Operating time OS to be carried by } i}{CT} \right|_i$$

4.2 적용사례

본 장에서는 분석을 위하여 일반적으로 많이 적용되어진 예제[14]를 살펴 본다(그림 1). 두 개의 생산품목 (P_1 와 P_2) 은 2 개의 기기: M_1 와 M_2 을 통하여서 작업이 수행된다. P_1 는 3 개의 작업이 수행되는데: $M_1(5 \text{ t.u.}$: 기본시간), $M_2(5 \text{ t.u.})$ 와 $M_1(1 \text{ t.u.})$ P_2 는 세 개의 작업 수행 : $M_2(3 \text{ t.u.})$, $M_1(4 \text{ t.u.})$ 과 $M_2(2 \text{ t.u.})$ 로 구성되어진다. 여기서, 두 개의 품목을 위한 치차는 연합적인 것으로 가정한다. 각 치차는 부품을 하나씩 만 운반이 가능하다고 가정한다. 각 작업의 작업 수행 순서는 OS1 과 OS2 로 표현 한다.

예제의 사이클 타임 CT 는 $CT = 10$ 이며, WIP 의 값은: 에서

$$WIP = \left\lceil \frac{11+9}{10} \right\rceil = 2$$

즉 최적의 필요한 치차의 수는 2 이 되며, 이는 최적화의 기준이 된다.

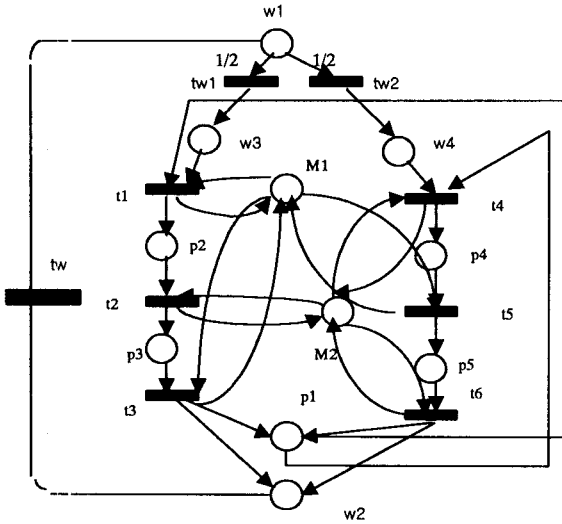


Fig. 4 : Example model

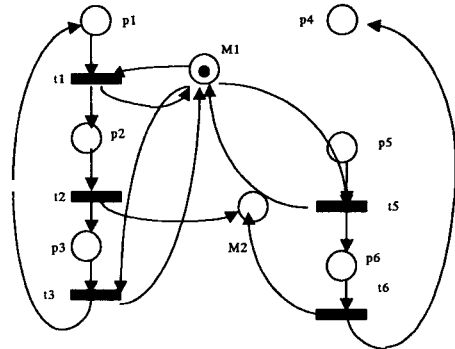


Fig 5 : BUC of M1

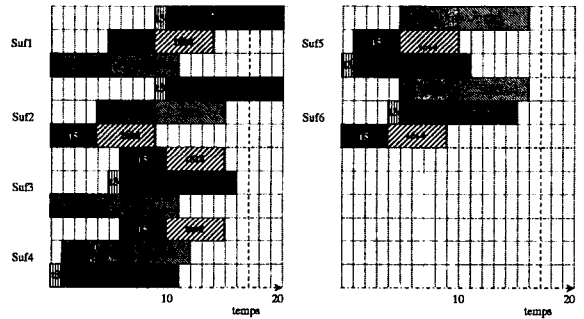


Fig 6 : Results scheduling of M1

1) M1의 BUC

M1의 BUC는 Fig. 5과 같이 얻을 수가 있으며, 6개의 가능스케줄을 구할 수가 있으며 이들의 최적의 전체예상시간 함수를 구하면 Fig. 6과 같은 결과를 얻을 수가 있다. 구하여진 결과 중에서 Suf4가 15로 가장 최적의 스케줄을 알 수가 있다.

Tab. 2: Fig.5의 MILN

$L_{BP} =$	0	t1/3	0	t4/3	0	0	0	0	0	t1/3	t4/3
	0	0	t2/2	0	0	0	0	0	0	0	t2/2
	t3/2	0	0	0	0	0	t3/2	0	0	t3/2	0
	0	0	0	0	t5/2	0	0	0	0	t5/2	0
	t6/2	0	0	0	0	0	t6/2	0	0	0	t6/2
	0	0	0	0	0	0	0	tw1	tw2	0	0
	0	0	0	0	0	0	tw	0	0	0	0
	0	t1/3	0	0	0	0	0	0	0	t1/3	0
	0	0	0	t4/3	0	0	0	0	0	0	t4/3
										t1/3	
	t3/2	t1/3	0	0	t5/2	0	t3/2	0	0	t3/2	0
										t5/2	
										t2/2	
	t6/2	0	t2/2	t4/3	0	0	t6/2	0	0	0	t4/3
										t6/2	

2) M2의 BUC

M2의 BUC는 Fig. 7과 같이 얻을 수가 있으며, M1과같이 6개의 가능 스케줄을 구할 수가 있으며 이들의 최적의 전체예상시간 함수를 구하면 Fig. 8과 같은 결과를 얻을 수가 있다. 결과 중에서 Suf2, Suf4, Suf5와 Suf6 이 16 ut 로 최적치임을 알 수가 있다.

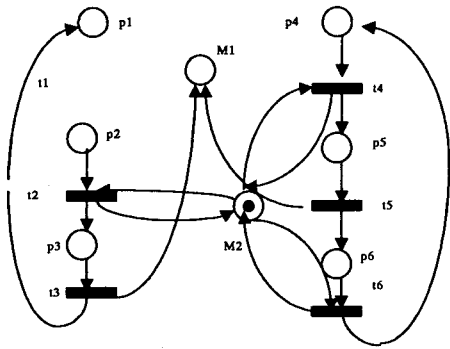


Fig 7 : BUC of M2

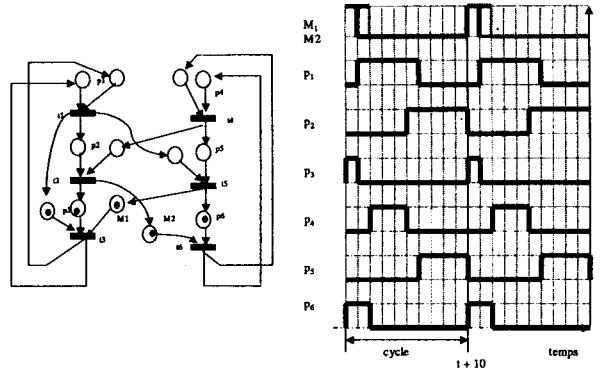


Fig 9 Results scheduling

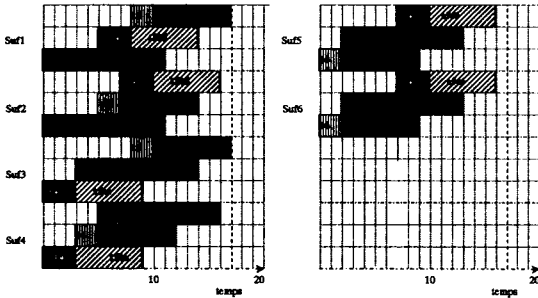


Fig. 8 Results scheduling of M2

4) 순환 스케줄링 결과

이제 BUC에서 얻은 결과들을 종합하여 보면 다음과 같은 결과를 얻게 된다.

- Bs1: $F((Suf_4)_{M1}, (Suf_2)_{M2}) : t_1 t_3 t_5 t_2 t_6 t_4 = 26$
- Bs2: $F((Suf_4)_{M1}, (Suf_4)_{M2}) : t_1 t_3 t_5 t_4 t_6 t_2 = 26$
- Bs3: $F((Suf_4)_{M1}, (Suf_5)_{M2}) : t_1 t_3 t_5 t_6 t_2 t_4 = 26$
- Bs4: $F((Suf_4)_{M1}, (Suf_6)_{M2}) : t_1 t_3 t_5 t_6 t_4 t_2 = 26$

위의 결과에서 4개의 스케줄 모두가 다 최적이지만, Bs1과 Bs3은 치차의 수가 4개가 되어서 기준치에서 벗어나고 Bs2의 경우는 기존 프로세스의 법칙에 어긋나므로 결국 Bs4만이 최적치로 남게 된다.

Bs4의 스케줄을 나타내 보면 Fig. 9와 같다.

4.3. 순환 스케줄 분석 알고리즘 성능분석

이 절에서는 앞 장에서 제시한 P-invariant를 이용한 순환 스케줄링 기법과 Unfolding 알고리즘의 결과와 비교하여 제시한 알고리즘의 효율성을 확인하고자 한다.

패트리 넷의 Unfolding은 패트리 넷의 마킹 관계를 모두 나타냄으로 패트리 넷의 성질들을 분석하는데 사용되어 왔다. Unfolding 알고리즘[19]은 이러한 unfolding의 성질들을 이용하여 유연생산시스템에서의 스케줄링을 분석하는데 처음으로 활용하였다.

[19]에서 제시된 알고리즘은 BUC(서브 넷)를 분리하는 과정과 unfolding을 이용한 BUC를 분석하는 과정으로 구성 되어 있다.

Unfolding 넷 UTPN은 최적의 작업 시간 계산을 위하여 함수 f (UTPN)를 제시하고 있는데, 함수 f (UTPN)는 초기 마킹에서부터 목적 마킹까지 통과 되어지는 트랜지션 t_i 의 모든 작업 시간의 합 함수이다. 여기서, 함수 f (UTPN)는 트랜지션 t_i 의 작업시간 함수인 $h(t_i)$ 와 다음 트랜지션의 작업시간(여기서는 최소대기시간) $g(t_i)$ 로 구성 되어지는데, 함수 f 의 전체 작업시간은 다음과 같은 산술식으로 얻어진다.

$$f(UTPN) = \text{Max}_{1 \leq j \leq n} \{ (\sum_{i=1}^j h(t_i)) + g(t_j) \}$$

여기서,

$h(t_i)$ 트랜지션 t_i 의 작업시간

$g(t_i)$ 는 트랜지션 transition t_i 의 최소 대기 시간

모델 UTPN의 최적 함수 $f(UTPN)$ 의 최적치로서 얻을 수가 있다. 다시 말하면, 만일 가장 짧은 unfolding 벨이 바로 최적의 스케줄링 벨이라고 할 수가 있다.

$$\text{최적 UTPN} = \min (f(UTPN_i))$$

BUC의 작업 수순이 최적 스케줄링을 구성하는데 가장 중요한 요소이다. Unfolding 알고리즘에서는 각 기기의 작업시간 효율도를 자원 공유수에 대비하여 그 작업 빈도율을 계산하여 이를 근거로 작업의 수순을 정한다. 즉 $d(m_i)$ 를 기기의 작업시간 효율도라고 하면,

$$d(m_i) = \frac{\varphi(m_i)}{\gamma(m_i)}$$

여기서,

$\varphi(m_i)$ 는 기기 i 의 종합작업시간,

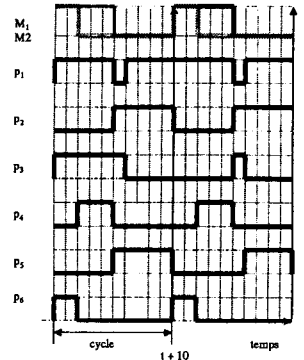
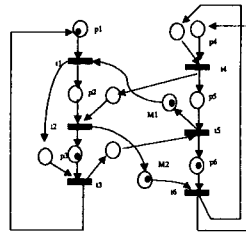
$\gamma(m_i)$ 는 기기 i 의 자원공유수.

따라서, 작업시간 효율도 $d(m_i)$ 가 높은 BUC부터 스케줄 분석을 하게 된다.

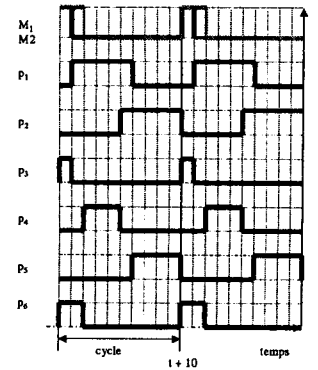
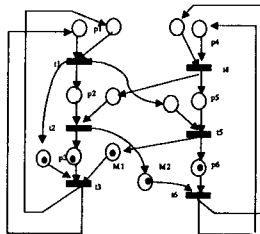
이러한 unfolding 알고리즘을 종합 정리하면:

- 1) 추이적 행렬을 작성한다.
- 2) 추이적 행렬에서 BUC를 분리한다,
- 3) 자원분산의 비율을 계산하여 BUC의 수순을 정한다,
- 4) 순서에 의하여 Unfolding 함수를 이용하여 BUC의 스케줄링을 분석한다.

이러한 unfolding 알고리즘을 그림 4의 모델에 적용하면 최종적으로 그림 10의 순환 스케줄 2개를 얻을 수가 있다.



(a)



(b)

Fig 10 : Results scheduling

위의 결과를 비교하여 보면, unfolding의 경우에는 두 개의 결과를 얻는데 반하여, 제시한 알고리즘에서는 오히려 한 개의 결과를 얻는 결과를 보였으며 같은 결과를 얻었다. 특히 치차의 경우, 최적의 값을 같이 얻지는 못하였다. 그러나, 예제의 경우, 계산상으로는 2개의 치차가 필요한 것으로 산출 되었으나, 실제적으로는 3개의 치차가 운행되어야만 하는 문제점이 있으며, 3개의 치차가 최적치로 판명되었다 [7,11].

5. 결론 및 앞으로의 연구 방향

이 연구에서, 우리는 먼저 구조적 병행성을 갖는 부분을 분할시켜서 분할되어진 모델분할을 가지고 효율적인 스케줄링을 구하는 분석 기법을 제시하였다. 자원의 공유 되어진 스케줄링 분석은 조합 최적화 문제로써 NP-hard 문제이다. 즉 자원 공유의 수가 많거나 복잡하여 질 경우 복잡도가 지수적으로 증가하는 문제가 있다. 따라서, 모델을 자원공유 단위로 모델분할을 하여 복잡도의 증가를 최소화 하였다. 또한 모델의 분할에 있어서 페트리 넷의 플레이스와 트랜지션간의 관계를 표시하고 있는 추이적 행렬식을 이용한 알고리즘을 제시함으로 손쉬운 분석모델을 제시하였다. 제시된 모델은 먼저 분할하고 이를 분석하고 그리고 그 결과들을 종합하는 3단계의 분석 과정을 갖는데, 그 과정이 다른 모델보다 아주 단순하다. 특히 계산방법에 있어서 그래픽 응용보다 쉬운 산술식을 이용하였고 또한 초기 토큰의 위치를 바꿈에 따라 새로운 모델분할을 구하여 종합적인 스케줄링 분석 모델의 가능성이 있다. 따라서, 앞으로의 연구에서는 사이클적인 스케줄링 분석에 그 응용성을 증명하고자 하며, 또한 간편하게 분할이 가능한 추이적 행렬식의 향상을 위하여 연구하고자 한다. 특히 다양한 시스템의 형태에도 적용이 가능한 알고리즘의 개발을 위하여 다양한 모델들에게 적용하고자 한다.

알 림

본 연구를 위하여 많은 의견을 교환한 프랑스 릴공대(Ecolecentrale de Lille)의 오르지 꼬르바 교수, 임 파스칼교수와 장클라우드 장띠나 교수에게 감사를 드린다. 또한 본 연구는 2001년도 창원대 학술진흥재단 공모과제로 일부 지원 받아 수행 하였음.

참고문헌

- [1] Berthomier B., Diaz M., "Modeling and Verification of Time Dependent Systems using Time Petri nets," IEEE Tr. Soft Eng., 17, 1991, pp. 259-273
- [2] Best E, Cherkasova L., Desel D. and Esparza J., " Characterization of Home States in Free Choice Systems", Hildesheimer Informatik-Berichte 9/90, Universitat Hildesheim, 1990
- [3] Carlier J., Chretienne P., "Timed Petri nets Schedules, in G. Rozenberg, ed., advanced in PN'1988, vol.340 of LNCS, pp.62-84, Springer-Verlag, Berlin, Germany, 1988
- [4] Holiday M., Vernon M., "A Generalized TPN Model for Performance Analysis," IEEE. Tran. On s/w, se-13, 1987, pp.1297-1310
- [5] Julia S., Valette R., Tazza M., "Computing a feasible Schedule Under A Set of Cyclic Constraints", 2nd International Conference on Industrial Automation, Nancy 7-9 juin 1995, p.141-146
- [6] Korbaa O., Camus H., Gentina J.-C., "FMS Cyclic Scheduling with Overlapping production cycles", in proceeding ICATPN'97, 1997, pp.35-52
- [7] Korbaa O., "Commande cyclique des systemes flexibles de production manufacturiere a l'aide des reseaux de Petri: de la planification a l'ordonnancement des regimes transtaires", Thèse de doctorat USTL 1, juillet 1998
- [8] Lee DY, DiCesare F., "Petri Net-based heuristic Scheduling for Flexible Manufacturing," In: Petri Nets in Flexible and Agile Automation, ed. Zhou MC, Kluwer Aca. Pub., USA 1995, pp.149-187
- [9] Lee J., Korbaa O., Gentina J.C., "Slice Analysis Method of Petri nets in FMS Using the Transitive Matrix", will be appeared in proceeding INCOM2001, Austria, Sep., 2001
- [10] Lee J., Korbaa O., Gentina J.C., " Modeling

- and Analysis of Cycle Schedule using Petri nets unfolding”, will be appeared in proceeding IEEE,SMC 2001,USA, Oct.,2001
- [11] Lee J.,“Une méthode d’analyse d’ordinnancement des systems flexibles de production manufacturière utilisant le dépliage des réseaux de Petri”, Thèse de doctorat, EC Paris, mars 2002
- [12] 이종근,“타임 패트리 넷 unfolding을 이용한 FMS의 스케줄링 분석”, 제어자동화시스템공학논문지, 제7권, 제4호,2001,pp.344-350
- [13] Liu J., Itoh Y., Miyazawa I., Seikiguchi T.,, “A Research on Petri nets Properties using Transitive matrix”, in proceeding IEEE SMC99,1999,pp.888-893
- [14] Murata M., Petri Nets: Properties, Analysis an Applications, Proceedings of the IEEE, 77(4), April 1989, pp. 541-580.
- [15] Ohl H., Camus H., Castelain E. and Gentina JC, “Petri nets Modelling of Ratio-driven FMS and Implication on the WIP for Cyclic Schedules, In:Proceeding SMC’95, 1995, pp.3081-3086
- [16] Richard P.,” Scheduling timed marked graphs with resources : a serial method”, accepted in proceeding INCOM’98, 1998
- [17] Song Y., Lee J, ”A study on the subnet of Petri net models using Transitive Matrix”, In:Proceeding IEEE-SMC’2000,vol.4,pp.3122-3127,2000
- [18] 송유진,이종근,“추이적 행렬을 이용한 패트리 넷모델의 분석방법에 대한 연구”,시물레이션학회논문지, 제10권,1호,pp.13-24,2001
- [19] 송유진,김종욱,이종근,“ 유연생산시스템 스케줄링 분석을 위한 추이적 행렬을 이용한 패트리 넷 분할”, 제어시스템공학회논문지, 게재승인, 2002
- [20] Zuberek W., Kubah W., “Throughput Analysis of Manufacturing Cells Using Timed Petri nets,” In:Proceeding ICSYMC 1993, 1993, pp.1328-1333
- [21] Zuberek W., ”Schedules of Flexible Manufacturing Cells and their Timed Colored Petri net Models,” In:Proceeding ICSYMC 1995, 1995, pp.2142-2147

● 저자소개 ●



이종근 (e-mail : jklee@sarim.changwon.ac.kr)

- 1974 숭실대학교 전자계산학과 졸업
- 1977 고려대학교 경영대학원 경영학과 졸업
- 1986 숭실대학교 대학원 전자계산학과 졸업
- 1990 프랑스 몽펠리에 대학교 전산과 박사수료
- 2002 프랑스 에콜 상뜨랄 빠리 컴퓨터공학 박사

1983~현재 창원대학교 컴퓨터공학과 교수

관심분야: 패트리 넬, 모델링 성능 분석, 유연생산시스템의 스케줄링
분석, 정보보호