

유연생산 시스템 스케줄링 분석을 위한 추이적 행렬을 이용한 페트리 넷의 분할

Slices Method of Petri Nets Using the Transitive Matrix for Scheduling Analysis in FMS

송 유진, 김 종욱, 이 종근
(You-Jin Song, Jong-Wuk Kim, and Jong-Kun Lee)

Abstract : We focus on the slicing off some sub-nets using the transitive matrix. Control flows in the Petri nets is done based on the token flows. One control flow explains the independent tokens status and if the token-in divides into several tokens after firing a transition then the control flow divides to several flows, as well. Accordingly, we define that the basic unit of concurrency (short BUC) is a set of the executed control flows based on the behavioral properties in the net. The BUC is S-invariant which has one control flow. We show the usefulness of transitive matrix to slice off some subnets from the original net based on BUC-through on an example.

Keywords : basic unit concurrency, FMS, slices, scheduling, time Petri nets, transitive matrix

I. 서론

페트리 넷은 많은 시스템의 모델링에 사용되는 그래픽형 이면서 수학적 모델링 도구이다. 그래픽형으로 페트리 넷은 플로우 차트나, 블록 다이어그램이나, 네트워크 형태로 표현되어 사용되고 있다. 무엇보다도 이러한 모델링에서 토큰의 흐름을 통하여 생동적이고 병행적인 시스템의 흐름을 시뮬레이션 할 수 있는 장점을 갖는다. 수리적인 면으로는, 상태방정식이나, 산술적 방정식 또는 기타 다른 수리적인 형태를 이용하여 시스템의 상태를 검증하거나 분석할 수가 있다. 페트리 넷의 장점은 페트리 넷이 가지는 독특한 성질, 즉 도달성(reachability), 생존성(boundedness)과 안전성(liveness) 등을 이용하여 시스템을 분석, 검증이 가능하다. 이러한 페트리 넷에 트랜지션과 플레이스, 점화시간이라는 개념을 부가하여서 처리 시간을 분석하는 형태를 타임 페트리 넷(Time Petri nets)이라고 부르며, 이는 스케줄링 분석을 시작으로 각종 병행 처리나, 동시 처리성이 있는 모델링 분석 및 검증에 적극적으로 활용되고 있다.

그러나, 동시성과 병행성을 갖는 시스템을 페트리 넷으로 분석 할 경우 상태공간이 상대적으로 폭발적으로 증대되는 경우가 발생하게 되어 이러한 시스템 모델링을 분석하는데 어렵다. 따라서, 많은 학자들이 이러한 경우를 해결하기 위하여 페트리 넷의 축소방식이나, 수학적 축소방식과 모듈화적 형태 등으로 연구를 하고 있다. 이러한 연구의 가장 기본적인 공통점은 기존 모델의 기본적인 성질들을 변형하지 아니하고 여러 개의 분석이 가능한 형태로 모델을 분할시켜서 분석과 검증을 시행하는 것이다.

본 연구에서는 제어적 독립성에 근거하여 구분되어 질 수

있는 모듈라 페트리 넷의 개념에서 이를 추이적 행렬식을 이용하여 간단하게 그리고 쉽게 모델을 분할 할 수 있는 분할 알고리즘을 처음으로 제시한다. 본 연구에서의 분할이라 함은 기본적인 모델에서 트랜지션이나 플레이스를 감축시키거나 합병하는 것이 아니라, 제어가 독립성을 갖는 모델로 분리 하는 것이므로 합병이나, 감축하는 기법보다 간편하다. 추이적 행렬이란 페트리 넷의 플레이스와 트랜지션간의 관계를 표시한 행렬로써 페트리 넷의 성질 분석에 활용 되었다. 본 연구에서는 추이적 행렬에서 제어적 독립성을 찾는 알고리즘을 제시하고 이의 활용성과 검증을 위하여 유연생산 시스템에서의 스케줄링 문제를 분석한다.

본 연구의 구성은 2장에서 사용되어질 페트리 넷의 기본적인 정의들과 추이적 행렬을 정의하고 3장에서는 추이적 행렬을 이용한 모델 분할을 위한 알고리즘을 제시한다, 4장에서 유연생산 시스템의 스케줄링 분석에 기 제시한 알고리즘을 적용하여 분석한다. 마지막으로 결론과 앞으로의 연구 과제에 대하여 5장에서 다룬다.

1. 관련 연구

페트리 넷의 분할 관련 연구는 변형적 연구(transformation), 즉 축소나 일정한 형식화하는 연구가 Berthelot[20]를 중심으로 진행되었고, 도달성트리를 이용하는 연구로는 Valmari [21]가, 그리고 수리적인 알고리즘을 이용하여 Memmi[24]가 시행하여 왔다. 한편, 분석할 경우에 발생할 수 있는 상태 폭발을 최소화 하고자 하는 일명 모듈화 페트리 넷 연구는 Notomi[22]를 중심으로 시행되고 있다. 그러나, 이러한 모듈화 연구는 기존의 합성, 수학적 그리고 도달성 분석보다는 효율적이기는 하나, 작성자의 임의의 기준에 의하여 시스템이 분할되어 다양한 모델링 편의와 작성이 어렵다는 단점이 있다. 따라서, 이러한 기준을 설정하기 위하여 시스템 내에서 발생할 수 있는 구조적 병행성을 제시하여 분석하는 기준을 [14][23]에 제시하였다.

이러한 분석 기준도 모델링에 활용하는데 복잡하고 시스

접수일자 : 2001. 12. 6., 수정완료 : 2002. 1. 16.

송유진, 김종욱, 이종근 : 창원대학교 컴퓨터공학과(syj@sarim.changwon.ac.kr/jwkim@3lit.co.kr/jklee@sarim.chanwon.ac.kr)

※ 본 연구는 2000년도 창원대 학술진흥재단 연구교수 공모과제로 일부 지원 받아 수행하였다.

템마다 다양하게 기준이 제기되어져야 하는 단점이 있다. 따라서, 이러한 구조적 병행성을 수학적으로 표현하여 이를 통하여 분할 가능한 구조를 찾는다면 효율성이 있을 것이다. 본 연구는 이러한 관점에서 구조적 병행성이 상태불변이라는 점을 활용하여 추이적 행렬식을 이용하여 시스템 모델을 분할하는 알고리즘을 제시하고자 한다. 본 연구팀은 그 동안 패트리 넬 unfolding 개념을 이용한 스케줄링 분석 연구[16][18][19]를 꾸준히 시행하여, 그 동안 연구되어진 다른 스케줄링 연구 방법[5]-[9][15]과 적용의 차이를 두었다. 기존의 연구 방법에서는 공정의 처리시간과 공유자원의 작업 시간을 중심으로 전체적으로 스케줄링을 분석하므로 많은 시간이 소요되는 단점이 있었다. 또한 알고리즘을 시뮬레이션을 이용한 형태로 결과를 산출하므로 정확한 알고리즘의 복잡도에 대하여 분석이 어려운 단점이 있다. 본 연구팀에서 제시한 unfolding으로 분석하는 기법은 전체적인 가능 스케줄링 파악과 분석은 용이하나, 어떻게 모델을 분할하는가에 대한 연구가 요구되어 왔다[17]. 또한 패트리 넬의 분할을 위하여 추이적 행렬을 이용한 연구도 수행하여 왔다[25].

II. 타임패트리 넬과 추이적 행렬

1. 타임 패트리 넬과 모델 분할

이 절에서는 일반적인 타임 패트리 넬의 성질들을 서술하여 본다[10]-[13].

TPN은 $\langle P, T, E, S, Mo, \tau \rangle$ 인 6가지 튜플로 구성되며, $P = \{p_1, p_2, \dots, p_n\}$, $|P| \neq 0$, P는 플레이스의 유한집합, $T = \{t_1, t_2, \dots, t_m\}$, $|T| \neq 0$, T는 트랜지션의 유한집합, $P \cap T = \emptyset$, $E : P \times T \rightarrow N$, E는 입력함수, $S : T \times P \rightarrow N$, S는 출력함수(N:양의 정수집합), $Mo \in M = \{M | M : P \rightarrow N\}$, Mo는 초기토큰상태, τ 는 시간함수: $\tau : T \rightarrow Q^* \times (Q^* \cup \infty)$ (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_i의 수를 $\#(P_i, O(t_j))$ 로 표시한다.

패트리 넬의 행렬식 C는 $C = \langle P, T, B^-, B^+ \rangle$, 여기서 P, T는 각각 플레이스와 트랜지션의 집합을 나타낸다. B⁻와 B⁺는 m열과 n행의 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 $x^* = \{y \in P \cup T; (x, y) \in E\}$, 그리고 후행집합은 $x_* = \{y \in P \cup T; (y, x) \in E\}$ 이다.

트랜지션 T와 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'를 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 $M | \sigma > M' | \sigma'$ 로도 표현한다.

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

한 트랜지션 t는 만일 다음의 성질을 만족한다면 N에서

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

정의 1 : 사건 및 상태불변 : 행 벡터 $x^T = (x_1, x_2, \dots, x_n) \geq 0$ 는 $A x^T = \Delta M = 0$ 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.

패트리 넬에서의 제어 흐름은 토큰의 흐름에 의하여 기인한다고 정의할 수가 있다. 즉 하나의 제어적 흐름은 독립적인 토큰의 상태를 표현한다고 할 수가 있으며 또한 트랜지션의 점화에 따라 토큰이 여러 개로 분할되면 제어흐름도 같은 수만큼 분할되어진다. 따라서, 이러한 개념을 이용하여 상태 불변을 만족하는 플레이스의 집합과 관련된 트랜지션의 집합으로 구성된 서브 넬으로 분할할 수가 있는데 이를 모델 분할이라고 칭한다.

정의 2 : 패트리 넬 모델 분할[23] : 패트리 넬 $N = (P, T, F, M)$ 에서, 상태불변으로 생성된 모델 분할의 집합을 Slice Set = $\{P_Slice_i | i=1, \dots, N\}$ 라 하면, 패트리 넬의 모델분할은 $Slice_i | i=1, \dots, n$ 이며 $Slice_i = (P_i, T_i, F_i, M_i, \tau_i)$ 는 다음의 조건을 만족한다.

$$P_i = P_Slice_i,$$

$$T_i = \{t \in T | s \in P_i, (s, t) \in F \text{ or } (t, s) \in F\},$$

$$F_i = \{(p, t) \in F, (t, p) \in F | p \in P_i, t \in T_i\},$$

$$\forall \tau \in \tau_i, \tau_i(t) = \tau(t) \text{ and } \forall p \in M_i, M_i(p) = M(p).$$

모델 분할은 트랜지션의 점화에 의하여 상태 불변을 기준으로 여러 개의 서브 넬으로 나뉘어 질 수 있으며, 이러한 모델 분할에서의 트랜지션의 행동 점화 조건은 다음과 같이 정의 할 수 있다[23]:

- 트랜지션 t_i 가 공유되어 있지 못한 경우 : 단지 트랜지션 마킹 $m(t_i)$ 만을 만족한다.
- 트랜지션 t_i 가 여러 분할에 공유되었을 경우 : 트랜지션 t_i 에 관계되어진 모든 마킹이 만족되어야 한다.

2. 추이적 행렬

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

정의 3[13] : 플레이스 행렬식 B_P와 트랜지션의 행렬식 B_T은 각각 다음과 같이 표현한다

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

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

정의 4[17] : L_{BP} 을 추이적 인지 행렬(the labeled place transitive matrix)이라 하면 :

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

여기서, $t_i (i=1, 2, \dots, n)$ 는 다음을 표기한다:

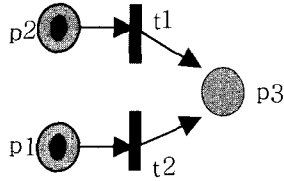
$$|t_i| = \begin{cases} 1 & \text{은 } t_i \text{ 점화} \\ 0 & \text{은 } t_i \text{ 점화불능} \end{cases}$$

L_{BP} 의 요소는 하나 이상의 트랜지션에서 하나 이상의 플

레이스로 집화되어지는 직접적 관계성을 표시 하고 있다.

정의 5[13] : L_{BP}^* 를 $m \times m$ 형 추이적 행렬이라 하자. 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.

예제 : 예를 들어 다음과 같은 간단한 넷의 경우를 생각해 보자:



이 넷은 간단하게 두 개의 트랜지션 t1과 t2를 가지고 3개의 플레이스 p1, p2, 그리고 p3가 있는데 그 중에서 p1과 p2가 초기 토큰을 가지고 있다. 따라서, 트랜지션 t1과 t2는 접화가 가능한 상태이다. 추이적 행렬 L_{BP}^* 은:

$$L_{BP}^* = \begin{bmatrix} 0 & 0 & t_1 \\ 0 & 0 & t_2 \\ 0 & 0 & 0 \end{bmatrix}$$

여기서, 플레이스 p1은 트랜지션 t2의 집화를 통하여 플레이스 p3으로 토큰을 보내며 플레이스 p2는 트랜지션 t1의 집화를 통하여 플레이스 p3에 토큰을 보낸다. 또한 역으로 플레이스 p3는 플레이스 p1과 p2로부터 트랜지션 t2와 t1를 통하여 집화가 가능하다는 것을 나타낸다.

III. 분할 모델 알고리즘

1. 병행적 기본단위(BUC)

패트리 넷에서의 제어 흐름은 토큰의 흐름에 의하여 기인한다고 정의할 수가 있다. 하나의 제어적 흐름은 독립적인 토큰의 상태를 표현한다고 할 수가 있으며 또한 트랜지션의 집화에 따라 토큰이 여러 개로 분할되면 제어흐름도 같은 수만큼 분할되어 진다. 따라서, 이러한 기본적인 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 분할한다.

정의 1 : 병행적 기본 단위(BUC) : BUC는 패트리 넷 모델에서 하나의 자원 공유(상태불변)에 관련되어진 서브 넷이다. BUC의 구성은 다음과 같다 :

- 1) 자원공유 플레이스
 - 2) 관련 트랜지션의 집합
 - 3) 자원공유 플레이스와 관련되어진 플레이스의 집합
- 따라서, BUC는 자원공유 수만큼 발생하게 된다.

모델은 두개의 자원공유 : M1과 M2를 갖는 시스템이며, 두 개의 BUC를 구할 수가 있다.

2. 분할 알고리즘

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

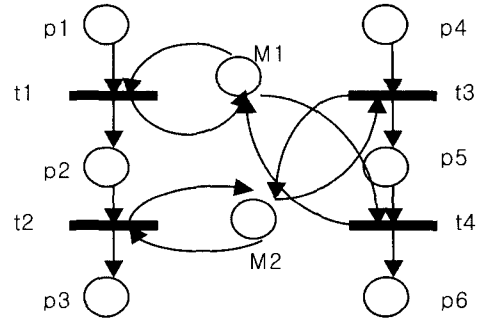


그림 1. BUC를 위한 패트리 넷 모델.
Fig. 1. The Petri net model for BUC.

표 1. 그림 1의 추이적 행렬식.

Table 1. The Transitive Matrix of Fig. 1.

	p1	p2	p3	p4	p5	p6	M1	M2
L_{BP}^*	0	t1/2	0	0	0	0	t1/2	0
	0	0	t2/2	0	0	0	0	t2/2
	0	0	0	0	0	0	0	0
	0	0	0	0	t3/2	0	0	t3/2
	0	0	0	0	0	t4/2	t4/2	0
	0	0	0	0	0	0	0	0
	0	t1/2	0	0	0	t4/2	t1/2	0
	0	0	t2/2	0	t3/2	0	t4/2	0
	0	0	t2/2	0	t3/2	0	0	t2/2
	0	0	0	0	0	0	0	t3/2

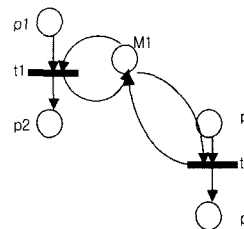


그림 2. M1의 BUC.
Fig. 2. BUC of M1.

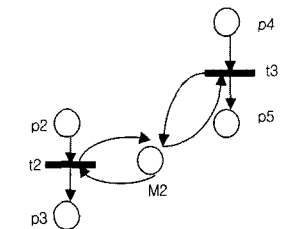


그림 3. M2의 BUC.
Fig. 3. BUC of M2.

- 1) 추이적 행렬 L_{BP}^* 를 작성한다.
- 2) 자원공유 플레이스를 선택한다
- 3) 선택된 자원공유 플레이스의 열에 있는 트랜지션을 모두 선택한다. 합이 1이 되는 트랜지션과 플레이스를 선택한다.
- 4) 행에서 선택된 트랜지션의 합이 1이 되는 트랜지션과 관련 플레이스를 선택한다.
- 5) 만일 트랜지션의 합이 1이 안되면 해당 트랜지션이 1이 되는 플레이스를 찾는다. 3)으로 간다.
- 6) 다음의 자원공유 플레이스를 선택하여 3)-5)를 자원공유 플레이스가 없을 때까지 반복한다.

분할 알고리즘을 이용하여 분할되어진 모델 분할은 기존의 패트리 넷 축소나 변형 기법과는 달리 감축되거나 합성되어지는 플레이스나 트랜지션이 없이 모든 플레이스와 트랜지션이 존재하므로 다음과 같은 성질을 통하여 기존의 성질 유지성을 검증할 수가 있다.

정의 2[23]: 행동일치(Behavioral Equivalent) : P와 Q를 두 개의 패트리 넷이라 하자. 만일 두 개의 넷이 갖는 도달성 그래프가 1:1 대응이 가능하다면 이들 P와 Q는 행동일치한다고 하며 이를 P~Q로 표시한다.

정리 1: N을 타임 패트리 넷이라 하고, S를 분할 알고리즘을 이용하여 얻은 모델 분할의 집합이라고 하자. 만일 모델 분할 집합 S가 행동 점화 조건을 만족한다면 N과 S는 행동일치 한다(N~S).

증명 : 만일 모델 분할의 집합 S가 행동점화 조건을 만족하지 않는다고 하자. 따라서, 공유되어진 트랜지션 t의 마킹이 공유되어진 모델 분할에서 만족하지 않을 경우, 제어의 흐름은 만족하지 않는 모델 분할에로 흘러가지 못한다. 따라서, 이러한 경우에는 분할 알고리즘에서 제시한 구조적 병행성을 만족하지 아니 하므로 발생할 수가 없다. 따라서, 모든 모델 분할은 트랜지션이 공유되었을 경우 제어의 흐름이 분할되어진다. 따라서, 원래의 넷 N에서 분할되어진 모델 분할의 집합 S에서 행동점화 조건을 만족하지 않는 모델 분할은 존재하지 않는다. 따라서, 항상 N과 S는 행동 일치한다. ■

IV. 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 net으로 되어 있다. 이와 같은 유연생산시스템에 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 스케줄링 분석을 하므로 그 실용성을 검증하고자 한다.

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 종합화하기 위한 기본적인 산술식에 대하여 생각하여 본다.

모델 분할 S는 다음과 같이 3개의 관련된 함수 f(S), g(ti) 그리고 h(ti)를 갖는데, 이를 통하여 가장 효율적인 스케줄링 시간(makespan time)을 산출한다. 함수 f(S)는 초기 마킹 m₀에서부터 트랜지션 ti를 지나 도달 마킹까지의 전체 예상가능시간을 나타낸다. 이러한 전체예상가능시간 함수 f(S)는 다시 두 개의 함수 h(ti)와 g(ti)로 구성되는데, 예상 도달 시간 함수 h(ti)는 트랜지션 ti까지의 예상도달시간으로 트랜지션 ti는 포함되지 아니한다, 한편 예상가능 시간함수 g(ti)는 트랜지션 ti에서부터 도달 마킹까지의 예상가능시간을 나타낸다. 따라서, 이러한 함수들의 관련식은 다음과 같이 종합화할 수가 있다:

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

따라서, 모델 분할에서의 전체예상가능시간 함수 f(S)는 다음과 같이 변형될 수가 있다.

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

가능 스케줄 S는 여러 개가 발생 가능하므로, 최적화되어진 스케줄은 전체예상가능시간 함수 f(S)의 최소화를 통하여 구할 수가 있다.

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

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

d(Ai)를 도달 가능시간의 효율 도수라 한다면, 효율 도수는 다음과 같이 산출된다.

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

여기서, d(h(t_i))는 h(m)의 도수, n(h(t_i))는 h(m)의 토큰의 합.

따라서, 가능시간 효율도수 h(m)는 모델분할 되어진 넷에서의 효율적인 스케줄링을 선정하는데 사용되어지는 한 요소가 될 수가 있다.

2. 적용사례

이 절에서는 그림 4.1에서 보이는 것과 같은 사례를 제시한 알고리즘을 가지고 분석하여 보고자 한다. 그림 4의 넷에서는 2개의 기기 R1과 R2에 의하여 2개의 작업이 수행되어 지는데, 각 작업은 모두 3개씩의 공정을 갖는다. 따라서, 작업 1은 트랜지션 t1, t3 그리고 t5에 의하여 이루어지는데 총 도달시간은 각각 2, 7 그리고 1의 점화시간이 소요된다. 한편 작업 2의 경우에는 트랜지션 t2, t4 그리고 t6에 의하여 수행되어지는데 점화 시간은 각각 3, 2, 그리고 4의 시간이 소요된다. 이 사례는 [9]에서 응용한다.

먼저, 그림 2의 추이적 행렬식은 다음의 표 2의 행렬로 나타낼 수 있다. 이 행렬의 순서는 앞에서부터 p1, p4, p2, R1, p5, R2, p3, p6으로 되어있다.

이 행렬에서, 초기 토큰은 R1과 R2에 존재한다. 따라서 R1의 행과 열을 선택한다. 그리고 관련되어지는 트랜지션과 플레이스들을 선택하면 트랜지션 t1, t3, t4와 t2이고 플레이스는 p1, p2, p3, p4, p5와 p6이 선택된다. 이제 이들을 서로 연결시키면 그림 5와 같이 R1에 관한 모델분할을 얻게 된다. 같은

표 2. 예의 추이적 행렬식.

Table 2. Transitive matrix of example net.

$r_{mp} =$	0	$t_3/2$	0	$t_3/2$	0	0	0	0
	0	0	0	0	0	$t_5/3$	0	0
	0	0	0	$t_4/3$	$t_4/3$	0	0	$t_4/3$
	t_1	$t_3/2$	t_1	$t_4/3$	$t_4/3$	0	t_2	$t_4/3$
	0	0	0	0	0	$t_5/3$	0	0
	0	0	0	0	0	$t_6/2$	0	0
						$t_5/3$		
	0	0	0	$t_4/3$	$t_4/3$	0	0	$t_4/3$
	0	0	0	0	0	$t_6/2$	0	0

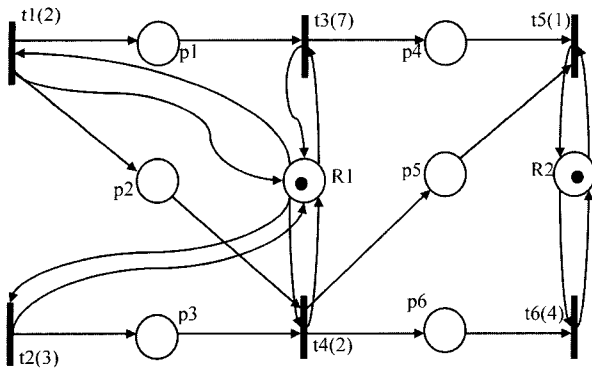


그림 4. 두 개의 기기를 공유한 모델의 페트리 넷.
Fig. 4. A example Petri nets which has two resource constraints.

방법으로 R2의 경우를 모델 분할하여 보면 그림 6과 같다.

만일 초기 토큰의 위치를 달리 한다면, 새로운 형태의 모델 분할을 얻을 수가 있다. 이것은 유연생산 시스템에서 사이클 형태의 작업인 경우에서 최적의 사이클 공정을 찾는 경우에 적용하는데 사용될 것으로 예견된다. 따라서, 이러한 연구는 추후에 적용 연구하고자 한다. 모델 분할을 시키는 순서 즉 전체 공정에 영향을 가장 많이 미치는 작업부터 먼저 모델 분할을 시켜서 이에 대한 공정을 분석함이 타당하다고 생각된다. R1의 경우 전체 공정시간은 14이고 R2는 5가 되므로 우선 R1부터 스케줄링을 분석하기로 한다.

1) 기기 R1에서의 효율적 스케줄링

R1모델분할에서는 적용 가능한 5개의 스케줄링을 생각할 수가 있다:

즉,

- S1: t2 → t1 → t3 → t4: 14
- S2: t2 → t1 → t4 → t3: 14
- S3: t1 → t2 → t3 → t4: 14
- S4: t1 → t2 → t4 → t3: 14
- S5: t1 → t3 → t2 → t4: 14

R1의 경우에는 초기 토큰이 R1에 있으므로 점화 가능한 트랜지션은 t1,t2,t3,t4이지만 t4는 반드시 t1과 t2가 점화되어야 가능하고 t3의 경우는 t1이후의 작업 공정이므로, 여기서는 먼저 t1과 t2만 점화가 가능하다. 따라서 총 5개의 스케줄링이 생성된다. 이 경우에서 예상가능시간은 공교롭게도 공히 14로 동일하다. 이 의미는 이 스케줄링에서는 어느 작

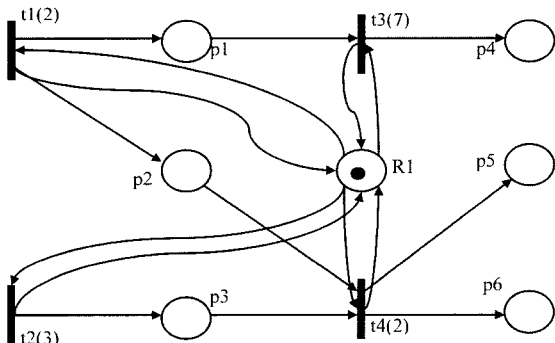


그림 5. R1의 서브 넷.
Fig. 5. A Subnet R1.

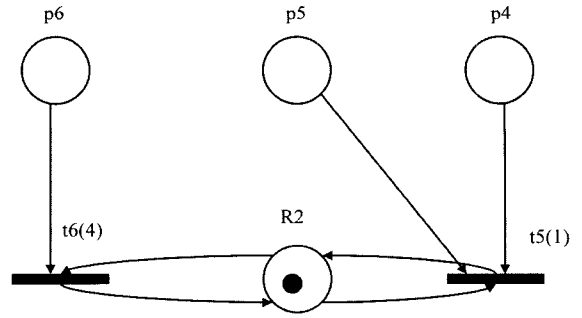


그림 6. R2의 서브 넷.
Fig. 6. A Subnet R2.

업을 먼저 수행하여도 시간에는 영향을 미치지 아니한다는 의미가 된다.

2) 기기 R2의 경우

기기R1의 경우와 같이 우리는 모델 분할을 그림 7과 같이 얻을 수가 있다.

여기서는 두 개의 스케줄링을 고려하여 생각하여보면 다음과 같다

- S1: t6 → t5 : 5
- S2: t5 → t6 : 5

여기서, 두 개의 가능한 스케줄링이 모두 같은 값을 보유하므로 이는 두개의 스케줄링 모두 다 최적의 스케줄링 가능성을 의미하므로 앞서 얻은 결과에 이 두 개의 결과를 조합하여 스케줄링을 종합한다.

3. 최적 스케줄링

이제, 두 개의 모델분할에서 구한 각각의 효율적 스케줄링을 적용해 보고자 한다. R1에서는 모두 동일한 시간 값을 나타내었고, R2에서 얻은 결과를 적용하여 최적치를 구하면 15의 값이 최적치가 되어 두 개의 효율성 있는 스케줄링 : t1 → t2 → t4 → t3 → t6 → t5와 t2 → t1 → t4 → t3 → t6 → t5를 발견할 수가 있다. 이를 간트 차트로 나타내면 다음의 그림 8과 같다.

- S1S1 : t1 → t2 → t4 → t3 → t5 → t6 : 19
- S1S2 : t1 → t2 → t4 → t3 → t6 → t5 : 15
- S2S1 : t2 → t1 → t4 → t3 → t6 → t5 : 15
- S2S2 : t2 → t1 → t4 → t3 → t5 → t6 : 19

3. 최적치에 대한 검증

새로운 효율치의 제시에 대하여서는 그 구한 값에 대한 검증이 필요하다. 본 연구에서는 최적치를 구하는데 있어서

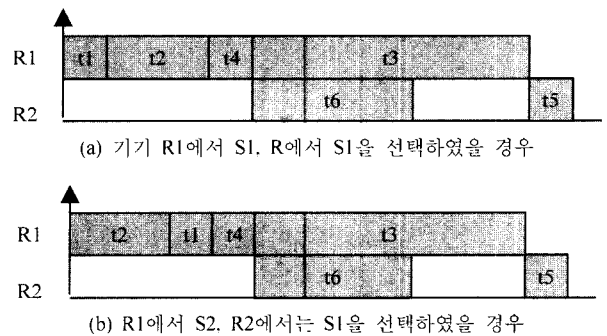


그림 7. 기기를 기준으로 한 최적 스케줄링.
Fig. 7. Optimal schedule based on machine in I date.

소요되어지는 시간적 성능을 비교하기보다는 가장 최적치 인가에 대하여 검증하고자 한다. 이 예는 간단한 예이므로 예상 가능한 모든 스케줄링을 선정하여 최적치를 계산하여 보고자 한다. 사례의 밑에서는 총 10개의 가능 스케줄링이 있으며 이에 소요되어지는 예상시간은 다음의 그림 8과 같다. 이 스케줄들에서, 우리는 두 개의 스케줄을 선택할 수가 있다. 즉 (4)와 (8)이 최소의 가능시간 임을 발견할 수가 있다. 또한 여기서, 우리는 스케줄링 $t_4 \rightarrow t_3$ 과 $t_6 \rightarrow t_5$ 는 불변임을 발견할 수가 있다. 즉 어떤 공정이 있다 하더라도, 이 공정을 통과한다면 그것은 가장 최소의 작업시간을 얻을 수가 있다는 것이다.

따라서, 우리는 제안되어진 알고리즘이 최적치를 얻을 수가 있다고 주장할 수가 있다. 그러나, 이것은 아주 단순한 사례이며, 이 사례를 응용한 [9]에서는 마크드 그래프를 이용한 LFT 알고리즘을 이용하여 한 개의 스케줄링 $t_1 \rightarrow t_2 \rightarrow t_4 \rightarrow t_3 \rightarrow t_6 \rightarrow t_5$ 을 발견한 것에 비하면 여기서는 최적 가능한 모든 스케줄링을 발견한 차이점이 있다.

지금까지 우리는 한 과정의 작업 공정을 표시하였다. 그러나 이를 두 개의 사이클로 나타낸다면, 우리는 그림 9와 같이 보다 효율화되어진 스케줄링을 표현할 수가 있다. 즉 반복하면서 총 15의 작업시간에서 1의 시간이 줄어든 14의 시간

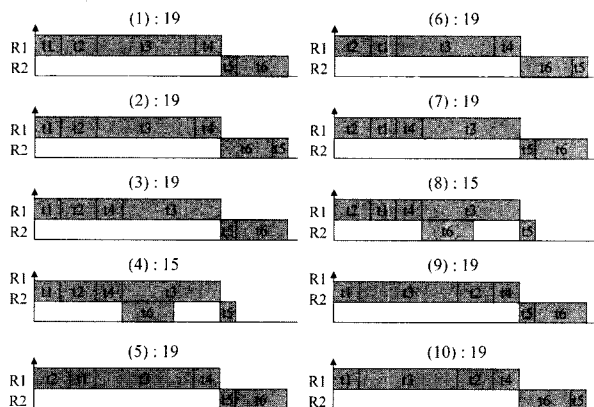


그림 8. 가능한 스케줄링.
Fig. 8. Possible schedule.

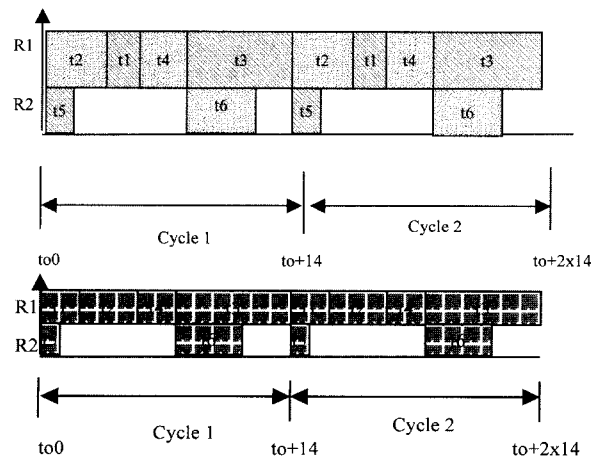


그림 9. 2사이클의 스케줄링.
Fig. 9. Schedule of 2 cycles.

으로 작업 수행이 가능한 스케줄링이 나타나게 되는 것이다.

V. 결론 및 앞으로의 연구 방향

이 연구에서, 우리는 먼저 구조적 병행성을 갖는 부분을 분할시켜서 분할되어진 모델 분할을 가지고 효율적인 스케줄링을 구하는 분석 기법을 제시하였다. 자원의 공유되어진 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 패트리 넷의 플레이스와 트랜지션 간의 관계를 표시하고 있는 추이적 행렬식을 이용한 알고리즘을 제시함으로써 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 모델 분할을 구할 수 있음으로써 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 하며, 특히 효율성 검증을 위하여 다른 알고리즘들과 성능 분석을 통한 검증도 하고자 한다.

알림

본 연구를 위하여 많은 의견을 제시, 교환한 프랑스 릴공대(Ecole Centrale de Lille)의 오르지 꼬르바 교수와 장클라우드 장띠나 교수에게 감사를 드린다.

참고문헌

- [1] B. Berthomier and M. Diaz, "Modeling and verification of time dependent systems using time petri nets," *IEEE Tr. Soft Eng.*, 17, pp. 259-273, 1991.
- [2] J. Carlier, P. Chretienne, "Timed Petri nets schedules," in G. Rozenberg, ed., advanced in PN'1988, vol. 340 of LNCS, pp. 62-84, Springer-Verlag, Berlin, Germany, 1988.
- [3] M. Holiday and M. Vernon, "A generalized TPN model for performance analysis," *IEEE. Tran. On s/w*, se-13, pp. 1297-1310, 1987.
- [4] J. Lee, "Time composition problem in time Petri nets," In : Proceeding CIS'97, pp. 507-512, 1997.
- [5] W. Zuberek and W. Kubah, "Throughput analysis of manufacturing cells using timed Petri nets," In : Proceeding ICSYMC 1993, pp. 1328-1333, 1993.
- [6] W. Zuberek, "Schedules of flexible manufacturing cells and their timed colored Petri net models," In : Proceeding ICSYMC 1995, pp. 2142-2147, 1995.
- [7] O Korbbaa, H. Camus, and J-C Gentina, "FMS cyclic scheduling with overlapping production cycles," *In Proceeding ICATPN'97*, pp. 35-52, 1997.

- [8] H. Ohi, H. Camus, E. Castelain, and JC Gentina, "Petri nets modelling of ratio-driven FMS and implication on the WIP for cyclic schedules," In : *Proceeding SMC'95*, pp. 3081-3086, 1995.
- [9] P. Richard, Scheduling Timed Marked Graphs with Resources : a serial method, accepted in proceeding INCOM'98, 1998.
- [10] T. Murata, Petri Nets : Properties, Analysis an Applications, *Proceedings of the IEEE*, 77(4), pp. 541-580, April, 1989.
- [11] DY Lee, F. DiCesare, "Petri net-based heuristic scheduling for flexible manufacturing," In : *Petri Nets in Flexible and Agile Automation*, ed. Zhou MC, Kluwer Aca. Pub., USA, pp. 149-187, 1995.
- [12] E. Best, L. Cherkasova, J. Desel, and J. Esparza, "Characterization of home states in free choice systems," *Hildesheimer Informatik-Berichte 9/90*, Universitat Hildesheim, 1990.
- [13] Jinghong LIU, Y. Itoh, I. Miyazawa, T. Seikiguchi, "A research on Petri nets properties using transitive matrix," in *proceeding IEEE SMC99*, pp. 888-893, 1999.
- [14] WJ Lee, S. D. Cha, YR Kwon, "Integration and analysis of use cases using modular Petri nets in requirements engineering," *IEEE Trans. On Software Eng.*, vol. 24, no. 12, pp. 1115-1130, 1998.
- [15] S. Julia, R. Valette, M. Tazza, "Computing a feasible schedule under a set of cyclic constraints," *2nd International Conference on Industrial Automation*, Nancy 7-9, June, pp. 141-146, 1995.
- [16] 이종근, "타임 페트리 넷 unfolding을 이용한 FMS의 스케줄링 분석," 제어 · 자동화 · 시스템공학논문지, 제7권, 제4호, pp. 344-350, 2001.
- [17] Jong K. Lee, O. Korbaa, JC. Gentina, "Slice analysis method of Petri nets in FMS using the transitive matrix," *will be appeared in proceeding INCOM2001*, Austria, Sep., 2001.
- [18] Jong K. Lee, O. Korebaa, JC. Gentina, "Modeling and analysis of cycle schedule using Petri nets unfolding," *will be appeared in proceeding IEEE, SMC 2001*, USA, Oct., 2001.
- [19] C. S. Bae, J. W. Kim, Jong K. Lee, "Modelinsation et analyse des systemes flexibles de production manufacturiere: ordonnancement de l'en-cours sous contrainte de ratio," *Proceeding CIFA2000*, Lille, 5-8 July, 2000, pp. 790-795.
- [20] Gerard Berthelot, "Transformation and decompositions of nets," *Advances in Petri Nets*, pp. 359-376, 1986.
- [21] Antti Valmari, "The state explosion problem," *Petri Nets*, pp. 429-528, 1996.
- [22] Masato Notomi and Tadao Murata, "Hierarchical reach ability graph of bounded Petri nets for concurrent-software analysis," *IEEE Transaction on Software Engineering*, vol. 20, no. 5, 1994.
- [23] 이우진, 김홍남, 권용래, "Petri Net Slice를 이용한 Petri Net 모델의 합성적 분석," 한국 정보 처리학회 추계 학술 발표 논문집, 제6권, 제2호, 1999.
- [24] G. Memmi et G. Roucairol, "Linear algebra in net theory," *LNCS*, no. 84, SpringerVerlag, Hamburg, FRG, 1979.
- [25] Y. Song and J. Lee, "A study on the subnet of Petri net models using transitive matrix," In : *Proceeding IEEE-SMC '2000*, vol. 4, pp. 3122-3127, 2000.



송 유 진

1969년 6월 10일생. 1992년 창원대학교 컴퓨터공학과 졸업(학사). 1995년 창원대학교 대학원 컴퓨터공학과 졸업(석사). 2001년 창원대학교 대학원 컴퓨터공학과 박사 수료. 1995년~현재 창원대학교 강사. 관심분야는 페트리넷, 성능분석, 정보보호, 스케줄링 연구.



김 종 욱

1967년 8월 5일생. 1992년 경상대학교 전자계산학과 졸업(학사). 2001년 창원대학교 대학원 컴퓨터공학과 석사 수료. 현재 삼일 정보 기술(주) 대표이사. 관심분야는 페트리넷, 성능분석, 정보보호, 스케줄링 연구.



이 종 근

1952년 4월 28일생. 1974년 숭실대학교 전자계산학과 졸업(학사). 1977년 고려대학교 경영대학원 경영학과 졸업(석사). 1986년 숭실대학교 대학원 전자계산학과 졸업(석사). 1990년 Univ. de Montpellier II 전산과 박사 수료. 2002년 Ecole Centrale Paris 전산학박사. 1983년~현재 창원대학교 컴퓨터공학과 교수, 창원대학교 전산소장, 자연대부학장 역임. 관심분야는 페트리 넷, 성능분석, 정보보호, 스케줄링 연구.