대한산업공학회/한국경영과학회 2002 춘계공동학술대회
한국과학기술원(KAIST) 2002년 5월 3일~4일

제한된 시간변동을 갖는 시간제약 이산사건시스템의 스케줄링 분석

Discrete Event System with Bounded Random Time Variation

김자희, 이태억
한국과학기술원 산업공학과

Abstract

We discuss scheduling analysis for a discrete event system with time windows of which firing or holding time delays are subject to random variation within some finite range. To do this, we propose a modified p-time Petri net, named p+-time Petri net. We develop a condition for which a synchronized transition does not have a dead token, that is, the firing epochs do not violate the time window constraints. We propose a method of computing the feasible range of the token sojourn time at each place based on a time difference graph. We also discuss an application for analyzing wafer residency times within the process chambers for a dual-armed cluster tool for chemical vapor deposition.

Keywords: cluster tool, feasible scheduling, time constraint, time variation, p^+-time Petri net

1. 서론

최근 반도체 업계에서는 여러 개의 공정모듈과 운송모듈, 로드락이 하나의 장비에 통합되어 있는 클러스터장비의 사용이 증가되고 있다. 반도체의 일부 공정에서는 공정이 끝난 웨이퍼가 공정모듈에 오래 머무를 경우 웨이퍼를 폐기처분한다. 이 때, 공정이 끝난 웨이퍼가 공정모듈에 머물 수 있는 최대시간을 "최대체재시간"이라고 하고, 공정이 끝난 웨이퍼가 최대체재시간 이상 공정모듈에 머물 수 없다는 제약을 "체재시간제약"이라고 한다(Rostami, Hamidzadeh, and Camporese 2000). 이런 종류의 공정을 위해서 공정개발자가 공정을 수행하기 전에 체재시간제약을 만족시키는 스케줄이 가능한지 여부를 판단하는 방법이 연구되었다(Shin등 2001; Kim등 2002). 그러나 이런 연구들은 공정 및 작업시간의 변동을 고려하지 않았기 때문에 이론적으로 체재시간제약을 만족하도록 개발된 공정도 양산 중 체재시간제약을 만족시키지 못 하는 경우가 발생한다. 본 논문에서는 작업시간과 대기전략이 주어질 때, 빈번한 시간변동에서 스케줄이 적당한지를 판단하기 위하여 p^+-time Petri net을 제안하고 이를 이용하여 이산사건시스템을 모형화하고, 분석하는 방법을 제시한다.

2. p^+-time Petri net

시간변동에 있는 이산사건 시스템에 체재시간제약여부를 분석하기 위하여 각 place에 반드시 머물러야 하는 최소시간의 변동범위와 최대한 머물 수 있는 시간상한을 갖는 p^+-time Petri net를 다음과 같이 정의한다.

p^+-time Petri net의 정의 : p^+-time Petri net은 $P^+TPN=(P, T, I, O, M_0, \tau, \delta)$로 정의된다:

(1) P, T, I, O, M_0은 기존의 Petri net의 place, transition, input function, output function, initial marking과 동일하게 정의된다(Murata 1989; Peterson 1981).

(2) $\tau: P \rightarrow X$ 는 place에 token이 반드시 머물러야 하는 최소시간의 변동폭을 정의한다. Place P_{ij}에 들어온 token은 $\tau(P_{ij})=\tau_{ij}$의 시간이 지나야 가용(available)하며 τ_{ij}는 $a_{ij} \le \tau_{ij} \le b_{ij}$인 분포는 알려져 있지 않은 확률변수이다.

(3) $\delta: P \rightarrow Q$는 place P_{ij}에서 token이 반드시 머물러야 하는 시간 τ_{ij} 이 후에 token이 추가로 머물 수 있는 최대 허용 시간을 정의한다.

본 논문에서는 Petri net의 특수한 종류로 각 place가 오직 한 개의 input transition과 한 개의 output transition을 갖는 event graph만을 다룬다. Event graph는 decision이 전부 결정된 시스템의 형태를 모형화한다. Transition 를 T_j로 직접 연결하는 place가 있을 경우 T_i는 T_j의 선행 transition이며, 연결하는 place를 P_{ij}로, 이렇게 직접 연결되어 있는 transition간 관계들의 집합을 $E=\{(i,j) | T_i$에서 T_j를 연결하는 *place* P_{ij}가 존재$\}$로 표현하자. Transition T_i에서 T_j로 연결하는 place가 하나 이상일 경우도 있으나 표기의 편의상 본 논문에서는 특별히 명시하지 않는 한, 두 transition들 사이에 두 개 이상의 place가 없다고 가정하자. 본 논문의 결과들은 별도의 place가 있는 경우에 대해 간단하게 확장될 수 있다.

Petri net을 이용하여 클러스터장비를 모형화한다면 place P_{ij}를 공정모듈 혹은 로봇이 웨이퍼를 갖고 작업 중이거나 작업 후 대기상태라고 할 수 있고 P_{ij}의 input transition T_i은 공정모듈에 웨이퍼가 들어와 공정을 시작하는 사건이나, 로봇의 작업이 시작하는 사건, output transition T_j은 공정모듈에서 웨이퍼가 나가는 사건 혹은 다음 로봇작업이 시작하는 사건이라고 정의할 수 있다. Place P_{ij}에 token이 있다는 뜻은 공정모듈 혹은 로봇이 웨이퍼를 갖고 있다는 의미이다. 웨이퍼는 최소한 공정시간만큼은 머물러야 하므로 token이 place에 머물러

야 하는 최소한의 시간에 관한 모형화가 필요하다. 그러나 실제 공정시간에는 오차가 있으므로 이를 모형화하기 위해서는 token이 place에 머물러야하는 시간을 상수가 아닌 확률변수로 주어야 한다. 그러나 대부분의 기술자들이 정확한 작업시간의 분포를 알지 못 하고 각 작업시간의 변동의 범위에 관한 정보만을 갖고 있다. 이러한 경우를 모형화하기 위해 P^+TPN은 token이 머물러야 하는 시간이 상수인 p-timed Petri net(Khansa, Aygalinc, and Denat 1996)을 확장하여 token이 최소한 머물러야 하는 시간 X를 $[a_{ij}, b_{ij}](0 \le a_{ij} \le b_{ij} < \infty)$사이의 분포를 알 수 없는 확률변수로 정의한다. 클러스터장비를 사용하는 공정 중 화학박막증착과 같은 공정은 공정이 끝난 웨이퍼가 공정모듈에 머물 수 있는 시간에 제한이 있다. 이를 위하여 p-time Petri net에서는 token이 place에 머물 수 있는 시간의 최대값을 정의한다. 그러나 p-time Petri net은 token이 place에 머물러야 하는 시간이 상수이므로 token이 머물 수 있는 시간의 최대값도 상수로 정의되지만 P^+TPN는 token이 머물러야 하는 시간이 계속 변화하여 알 수 없으므로 $\delta(P_{ij}) = \delta_{ij}$는 τ_{ij}만큼의 시간이 지나 token이 가용해진 후 머물 수 있는 시간으로 정의한다.

P^+TPN의 동적인 움직임을 정의하기 위해서는 시간에 따른 각 place내 token 수의 변화에 관한 법칙이 필요하다. 어떤 place의 token 수가 변화되기 위해서는 직접 연결된 transition이 fire되어야 하고, transition이 fire하기 위해서는 그 transition은 enable되어야 한다. 만일 어떤 transition T_j의 각 input place P_{ij}의 가용한 token 수가 P_{ij}에서 T_j로 향한 arc의 weight보다 크거나 같다면 그 T_j는 "enable"되어 있다고 한다. Enable된 transition은 그 즉시 "fire"한다. 즉, input place의 token을 P_{ij}에서 T_j로 향한 arc의 weight 수만큼 줄인다. 그리고 T_j에서 T_j의 각 output place P_{jk}로 향한 arc의 weight수만큼의 token을 추가한다.

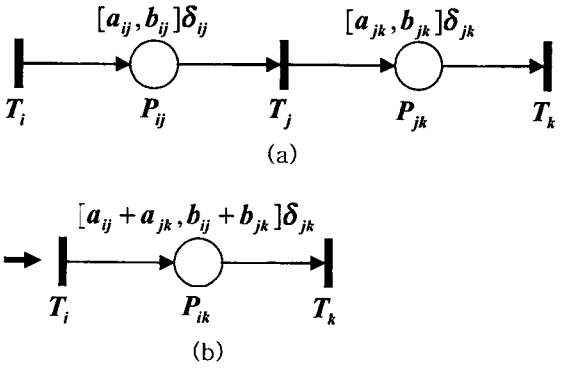

그림 1. 감소법칙.

예를 들어 그림 1(a)와 같은 p$^+$-time event graph이 있다고 하자. Transition T_i가 n번째 fire한 시점을 x_i^n이라고 하고 μ_{ij}는 place P_{ij}의 초기 token의 수라고 하면 transition T_j는 정의에 의해 $x_i^{n-\mu_{ij}} + a_{ij}$와 $x_i^{n-\mu_{ij}} + b_{ij}$ 사이에 enable되기 때문에 각각에 대해 token이 머물 수 있는 최대치는 $x_i^{n-\mu_{ij}} + a_{ij} + \delta_{ij}$와 $x_i^{n-\mu_{ij}} + b_{ij} + \delta_{ij}$가 된다. 그러므로 transition T_j에 다른 input place가 있다고 하더라도 transition T_j의 fire하는 시점은 식 (1)을 만족해야한다.

$$x_i^{n-\mu_{ij}} + a_{ij} \le x_j^n \le x_i^{n-\mu_{ij}} + b_{ij} + \delta_{ij} \quad (1)$$

이 조건을 만족하지 않을 때 token이 죽는다(token death)라고 하고 체재시간 제약을 위반하며 시스템 진행은 중단된다. p$^+$-time event graph가 live하기 위해서는 deadlock뿐만 아니라 이러한 token death가 없이 각 transition이 무한히 반복하여 firing될 수 있어야한다.

만일 transition T_j에 place P_{ij} 외에 다른 input place가 없다면 transition T_j의 fire시점은 오직 place P_{ij}의 token이 가용해지는 시점만 관계가 있다. 그러므로 Sloan과 Buy(Sloan and Buy 1989)의 감소법칙을 변형하여 1과 같이 p$^+$-time event graph의 감소법칙을 만들어 단순화할 수 있다.

보조정리 1: $T_i \rightarrow P_{ij} \rightarrow T_j \rightarrow P_{jk} \rightarrow T_k$와 같이 연결되어 있을 때에 다음 조건을 만족하면 place P_{ij}과 P_{jk}를 하나의 place P_{ik}로 통합할 수 있다.

(1) Place P_{jk}는 초기에는 token이 없어야 한다.

(2) Transition T_j의 input place는 오직 P_{ij}뿐이다.

(3) Transition T_j의 output place는 오직 P_{jk}뿐이다.

이 경우 통합된 place P_{ik}와 관련된 $t_{ik}, \delta_{ik}, \mu_{ik}$는 아래와 같이 정의된다.

(1) $\tau_{ik} = \tau_{ij} + \tau_{jk} \in [a_{ij} + a_{jk}, b_{ij} + b_{jk}]$

(2) $\delta_{ik} = \delta_{ij}$

(3) $\mu_{ik} = \mu_{ij}$

보조정리 1에서는 직렬로 연결된 transition들이 어떻게 통합될 수 있는지를 보여주고 있다. 만일 transition의 input place가 하나 이상인 경우 그 transition은 place들의 token 흐름을 동기화(synchronized)한다. 만일 그림 2와 같이 transition T_k에서 place P_{ik}와 P_{jk}가 동기 된다고 하면 이 transition이 enable하는 시점은 이 두 개의 place에 token이 들어온 시점에 의해 결정이 되고, p$^+$-time event graph의 경우 보조정리 2와 같이 T_k의 선행 transition들이 fire한 시점에 의해 결정된다.

보조정리 2: 동기하는 transition T_j가 n번째 fire하는 시점(x_j^n)은 이 transition의 선행 transition T_i들이 $n - \mu_{ij}$번 째 fire한 시점($x_i^{n-\mu_{ij}}$)과 이 두 transition을 잇는 place P_{ij}가 가용해 지는 데 걸리는 시간(τ_{ij})에 대해 다음의 관계를 가져야한다.

$$\max_{\forall i,(i,j)\in E}\{x_i^{n-\mu_{ij}} + a_{ij}\} \le x_j^n \le \max_{\forall i,(i,j)\in E}\{x_i^{n-\mu_{ij}} + b_{ij}\} \quad (2)$$

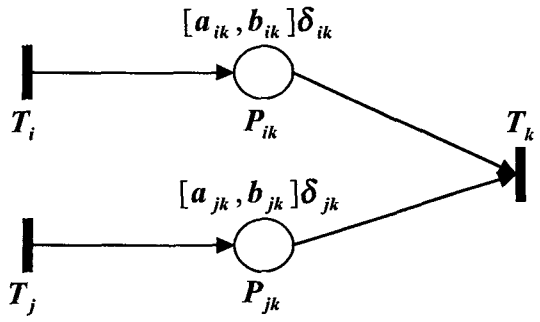

그림 2. 동기하는 transition T_k.

p-time Petri net의 경우 동기화하는 transition에서는 하나의 input place의 token에게 허용한 시간 이상 다른 input place의 token이 가용해지지 않아서 token이 죽는 경우가 발생한다. p-time Petri net의 최소 체재시간은 상수이므로 만일 fire 가능범위가 존재하면 스케줄이 가능하다. 그러나 p$^+$-time event graph에서는 식 (2)를 만족시키는 x_j^n의 값이 존재한다고 하더라도 place들에서 token이 가용해지기까지 걸리는 시간의 길이에 따라 체재시간제약을 만족시키지 않을 수 있다. 그러므로 p$^+$-time event graph의 liveness의 정의는 다음과 같다.

p$^+$-time event graph의 liveness 정의: 만일 어떤 p$^+$-time event graph의 모든 동기하는 transition의 input place에서 token이 죽는 경우가 결코 발생하지 않는다면 이 event graph는 live하다.

그러므로 p$^+$-time event graph가 live한지 여부를 판단하기 위해서는 임의의 동기하는 transition T_j의 input place P_{ij}에서 token이 최소한 머물러야 하는 시간 τ_{ij}가 어떤 값을 갖더라도 token이 죽지 않을 조건을 알아야한다.

보조정리 3: p$^+$-time event graph의 모든 동기하는 transition T_k에서 token이 언제나 죽지 않기 위해서는 임의의 input place의 token이 늦게 가용해지더라도 일찍 가용해진 token이 죽기 전에는 가용해져야한다. 즉, 임의의 firing 횟수 n과 T_k의 input place P_{ik}, P_{jk}에 대해 식 (3)의 조건을 만족시켜야 한다.

$$x_i^{n-\mu_{ik}} + b_{ik} \le x_j^{n-\mu_{ik}} + a_{jk} + \delta_{jk} \quad (3)$$

단, $(i,k),(j,k) \in E, i \neq j$.

이로부터 다음의 정리를 얻을 수 있다.

정리: 만일 deadlock이 없는 p$^+$-time event graph의 동기하는 모든 transition 에서 식 (3)의 조건을 만족하면 live하다.

3. 클러스터장비의 p$^+$-time event graph 모형

최근 20년간 반도체산업에서 클러스터장비는 식각공정(etching)부터 화학박막증착공정까지 다양한 공정으로 그 사용이 증가되고 있다. 클러스터 장비는 그림 3과 같이 일반적으로 공정을 수행하는 복수개의 공정모듈(process module, PM)과 외부와의 입출력을 담당하는 로드락(loadlock)이 웨이퍼의 이송을 담당하는 운송모듈(transfer module, TM)에 방사형으로 연결되어 있다. 공정할 웨이퍼가 들어있는 카세트가 로드락에 들어오면 TM의 로봇은 웨이퍼를 하나씩 공정할 공정모듈로 이송하고, 공정모듈에서 공정이 끝나면 다시 로봇이 로드락의 카세트로 이송한다. 카세트 안의 모든 웨이퍼의 공정이 끝나면 로드락은 카세트를 외부로 반납한다.

TM 중앙의 로봇은 초기에는 한 팔 로봇을 많이 사용했으나 최근에는 생산성을 향상시키기 위해서 두 개의 팔을 갖는 로봇의 사용이 증가되고 있다. 이와 같은 양팔 로봇 클러스터장비의 경우 두 개의 로봇 팔이 180°로 연결되어 있기 때문에 두 팔을 같이 이동하고, 한 번에 하나의 집는 작업이나 놓는 작업만 할 수 있다. 하나의 팔을 버퍼로 사용하는 양팔 로봇 클러스터장비의 가장 효율적인 로봇운영방식인 교환작업이란 그림 3과 같이 공정을 수행할 웨이퍼를 한쪽 팔에 든 로봇이 비어있는 또 다른 팔로 공정이 끝난 웨이퍼를 꺼낸 후에 180° 돌아 새 웨이퍼를 공정에 넣는 일련의 작업이다(Venkatesh등 1997). 이후 방금 꺼낸 웨이퍼를 다음 공정에서 교환하는 작업이 반복되기 때문에 모든 공정모듈에서 교환작업을 할 경우 로봇작업순서는 웨이퍼흐름에 따라 결정이 된다.

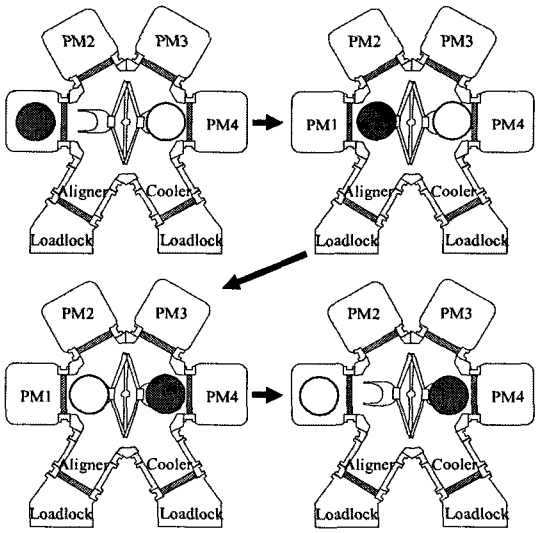

그림 3. 교환작업.

클러스터장비는 생산성을 높이기 위하여 같은 공정을 하나 이상에서 수행하도록 하는 병렬공정모듈이 존재하므로 다양한 종류의 웨이퍼흐름이 가능하다. 먼저 웨이퍼흐름 중 로드락을 나온 웨이퍼가 두 대의 공정모듈에서 공정을 하고 다시 로드락으로 반환되는 웨이퍼흐름 (1,1)의 경우를 Kim 등(Kim등 2002)의 방법론에 따라 각 공정의 시작이라는 event를 transition으로 공정 중 및 공정 후 다음 작업 전의 대기를 하나의 place로 모형화가 가능하다. 웨이퍼흐름 (1,1)은 로드락을 나온 웨이퍼가 두 개의 공정모듈에서 두 종류의 공정을 모두 거친 후에 로드락으로 반환된다. 이 때 가장 효율적이 로봇작업은

로드락에서 웨이퍼를 꺼내($T_1, P_{1,2}$) 첫 번째 공정을 의한 공정모듈로 이송($T_2, P_{2,3}$) 및 교환작업 ($T_3, P_{3,4}$)을 하고 교환된 웨이퍼를 두 번째에도 이송($T_4, P_{4,5}$), 교환작업($T_5, P_{5,6}$)을 한 후에 최종 교환된 웨이퍼를 들고 로드락으로 이송($T_6, P_{6,7}$), 반환($T_7, P_{7,1}$)하는 작업을 반복하는 것이다. 공정시간과 로봇의 작업시간이 표 1과 같다면 웨이퍼흐름 (1,1)의 p$^+$-time event graph은 그림 4(a)이고, 보조정리 1을 적용하여 단순화하면 그림 4(b)가 된다.

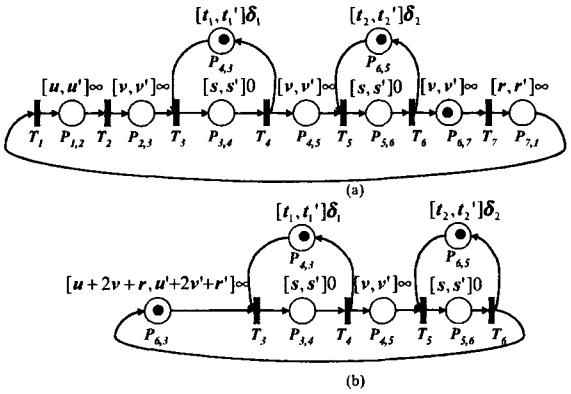

그림 4. 웨이퍼흐름 (1,1)의 p$^+$-time event graph.

실질적으로 클러스터장비의 작업시간은 일정하지 않으므로 각 작업시간을 상수가 아니라 표 1과 같이 정의한다. 예를 들어 로드락에서 웨이퍼를 집는 작업은 아무리 빨라도 u 이전에는 끝날 수 없지만, 통신장애나 aligning의 재시도 등에 의해 경우에 따라 u보다 큰 u'의 시간이 걸릴 수도 있다. 대부분의 로봇작업은 작업이 끝난 후에 웨이퍼를 로봇 팔 위에 오랫동안 들고 있어도 무한히 들고 있을 수 있다. 그러나 공정이 끝난 웨이퍼를 공정모듈 내에서 너무 오래 기다리게 할 경우 잔여열이나 가스에 의해 웨이퍼가 손상되므로 공정 후 최대체재시간 δ_1 혹은 δ_2 이상 머무를 수 없다. 만일 교환작업 도중 대기가 허용되지 않는다면 최소교환작업이 끝나면 곧장 공정모듈에 웨이퍼가 들어가므로 교환작업의 최대허용시간은 0이다.

그림 4의 동기하는 transition은 T_3, T_5 이므로 동기하는 transition T_3, T_5 모두에 대해 token이 언제나 live해야 이 스케줄이 모든 경우에 대해 체재시간제약을 만족한다고 할 수 있다.

예를 들어 첫 번째 공정에서 공정이 끝난 웨이퍼가 최대체재시간의 범위 δ_1 이상 머물지 않기 위해서는 transition T_3의 fire 횟수에 관계없이 token이 죽지 않아야 한다. 보조정리 3의 결과를 이용하면 임의의 fire 횟수 n에서 token이 죽지 않기 위한 조건은 $x_6^{n-1}+u'+2v'+r' \le x_4^{n-1}+t_1+\delta_1$이므로 δ_1은 $x_6^{n-1}+u'+2v'+r'-(x_4^{n-1}+t_1)$보다 커야 한다. $u'+2v'+r', t_1$은 상수이므로 최소 δ_1은 $x_6^{n-1}-x_4^{n-1}$의 최대값에 의해 결정되므로 두 transition이 fire하는 시간차의 최대값을 구하는 방법이 추가로 필요하다.

4. 시간차 그래프

3장의 클러스터장비의 예에서 알 수 있듯이 두 firing 시점간의 차의 최대값($\max\{x_i^m - x_j^n\}$)들의 관계를 아는 것이 필요하다. 주기적인 작업을 모형화한 cyclic p$^+$-time event graph의 경우에는 $\max\{x_i^m - x_j^n\}$와 $\max\{x_k^l - x_j^n\}$의 관계는 $\max\{x_i^{m-1} - x_j^{n-1}\}$와 $\max\{x_k^{l-1} - x_j^{n-1}\}$의 관계와 같다. 즉, firing 회수와 무관하게 시점 차에 관한 관계 그래프를 그릴 수 있다. 본 논문에서는 이러한 그래프를 시간차 그래프(time difference graph)로 명명하고 p$^+$-time event graph에서 시간차 그래프를 유도하는 방법을 제안하고 이 시간차 그래프를 이용하여 token이 가용해진 이후 place에 머무는 최대시간을 구하는 방법을 제시한다. 먼저 시간차 그래프를 정의하면 다음과 같다.

시간차 그래프의 정의 : 시간차 그래프는 방향성이 있는 그래프로 노드 Y에서 노드 Z로 향하는 화살표가 있으면 노드 Y를 노드 Z의 부모, Z를 노드 Y의 자식이라고 부른다. 그리고 화살표의 값의 부모노드의 값이 자식노드의 값보다 얼마나 큰지를 표현한다. 이 그래프의 노드는 노드의 종류와 차이를 알고 싶은 시점에 해당하는 두 transition의 번호를 포함한다. 노드의 종류는 자식노드의 중 어떤 값을 선택하는가에 따라 나뉜다.

(1) 자식 노드의 값에 연결하는 화살표 값을 더한 값

	작업시간의 변동범위	최대허용시간
로드락에서 웨이퍼를 집음($P_{1,2}$)	$[u, u']$	∞
웨이퍼의 이송($P_{2,3}, P_{4,5}, P_{6,7}$)	$[v, v']$	∞
교환작업($P_{3,4}, P_{5,6}$)	$[s, s']$	0
로드락에 웨이퍼를 반환($P_{7,1}$)	$[r, r']$	∞
첫 번째 공정($P_{4,3}$)	$[t_1, t_1']$	δ_1
두 번째 공정($P_{6,5}$)	$[t_2, t_2']$	δ_2

표 1. 각 작업시간의 변동범위와 최대허용시간.

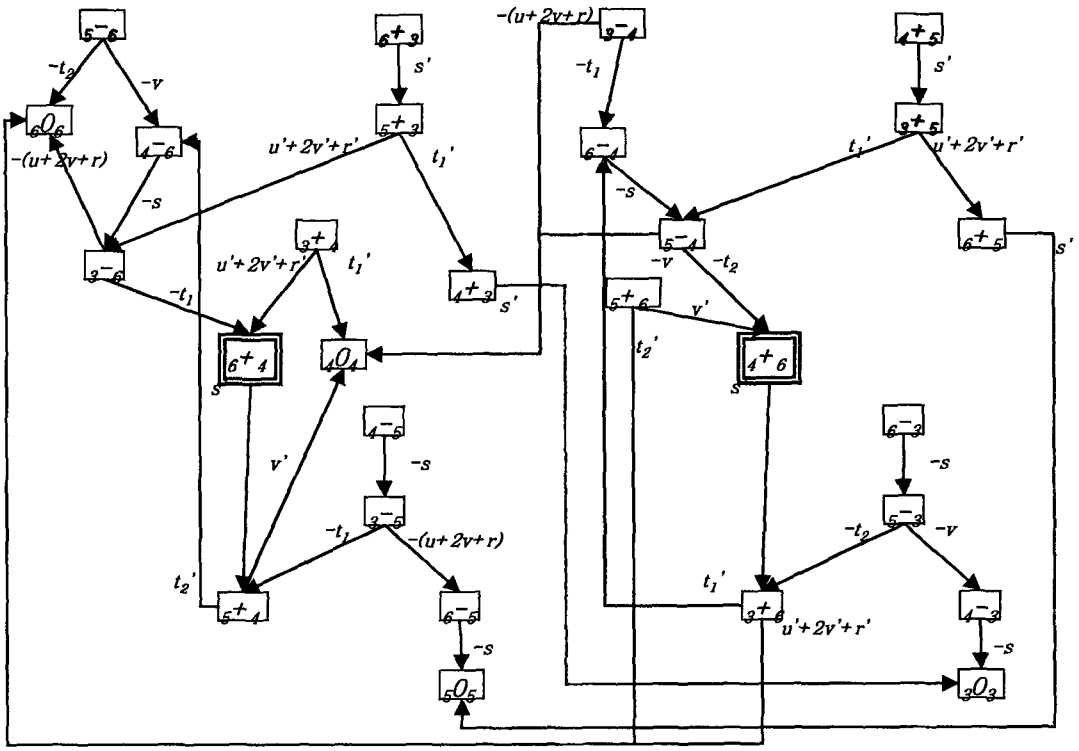

그림 5. 웨이퍼흐름 (1,1)의 시간차 그래프.

중 최대의 값을 가질 경우 '+' 노드라 한다.
(2) 자식 노드의 값에 연결하는 화살표 값을 더한 값 중 최소의 값을 갖는 경우 '-' 노드라 한다.

시간차 그래프의 정의에 따라 생성하는 방법은 다음과 같다. 그래프의 각 노드는 $\max\{x_i^m - x_j^n\}$라는 의미이다. 이 두 시점(x_i^m, x_j^n)간의 차이를 알기 위해서는 이 두 시점사이에 어떤 transition이 언제 fire했는지를 알아야 한다. 만일 $x_i^m \geq x_j^n$일 경우, transition T_j가 n번 째 fire한 후 T_i가 m번 째 fire할 때까지 얼마나 걸렸는지를 알아야 하므로 $\max\{x_i^m - x_j^n\}$는 T_i의 직전에 어떤 transition이 fire했는지를 파악해야 한다. 시간차 그래프의 노드를 시작 transition, 노드의 종류, 종료 transition 순으로 표현하면 $x_i^m \geq x_j^n$인 이 노드는 보조정리 2에 의해 $\max\{x_i^m\} = \max_{\forall k,(k,i)\in E}\{x_i^{m-\mu_{ki}} + b_{ki}\}$이므로 $_i + _j$로 표현된다. 자식노드로는 $(k,i)\in E$인 모든 k에 대해서 $\max\{x_k^{m-\mu_{ki}} - x_j^n\}$에 해당하는 노드로 가는 각 화살표의 값은 b_{ki}가 된다. 반대로 $x_i^m < x_j^n$일 경우 transition T_i가 m번 째 fire한 후 T_j가 n번 째 fire할 때까지 얼마나 걸렸는지를 알아야 하므로 $\max\{x_i^m - x_j^n\}$은 T_j를 기준으로 직전에 어떤 transition이 fire했는지를 파악해야 한다. 보조정리 2에 의해 $\max\{-x_j^n\} = -\min\{x_j^n\} = -\min_{\forall k,(k,j)\in E}\{x_k^{m-\mu_{ki}} + a_{kj}\}$이므로 이 노드의 종류는 '-'이고 $_j - _i$로 표현한다. $(k,j)\in E$의 모든 k에 대해 $\max\{x_i^m - x_k^{n-\mu_{ki}}\}$에 해당하는 노드들이 자식노드가 되며 이 두 노드들 간의 화살표의 값은 a_{kj}가 된다. 즉, 그림 4(b)의 p$^+$-time event graph의 모든 transition의 조합을 노드로 갖는 시간차 그래프를 그리면 그림 5와 같이 28개의 노드를 가진 그래프가 완성된다. 예를 들어 그림 5의 $_5 + _3$노드의 경우 이 노드의 의미는 $\max\{x_5^n - x_3^n\}$이고 자식노드는 T_5의 선행 transition T_4의 n번째 fire한 시점 x_4^n과 x_3^n의 차에 관한 노드($_4 + _3$)와 T_4의 선행 transition T_6의 $n-1$번째 fire한 시점 x_6^{n-1}과 x_3^n의 차에 관한 노드($_6 - _3$)가 된다.

시간차 그래프는 역화살표 방향으로만 영향을 받는다. 예를 들어 첫 번째 공정의 공정 후 최대체재시간은 $_6 + _4$의 값에 $u' + 2v' + r' - t_1$을 더한 값이므로 $_6 + _4$ 및 $_6 + _4$의 자손들의 값만 영향을 미친다. 즉, 그림 5의 회색 노드만이 첫 번째 및 두 번째 공정의 공정 후 최대체재시간에 영향을 미치므로 실제 분석을 할 경우에는 간소화된 시간차 그래프를 이용하는 것이 유리하다.

시간차 그래프는 min-max graph의 형태를 가지므로 Schwiegelshohn과 Thiele의 알고리즘을 이용하여 값을 구할 수 있다(Schwiegelshohn and Thiele 1999). 이 알고리즘은 초기에 '-'노드의 값을 0으로 정하고, 그 가정 하에서 '+' 노드의 값을 구한다. 그리고 이 변화된 '+' 노드의 값을 기준으로 '-'노드의 값을 구한다. 이와 같이 '+' 노드의 값을 기준으로 '-' 노드의 값을 구하고 '-'노드의 값을 기준으로 '+' 노드의 값을 구하는 작업을 더 이상의 값의 변화가 없을 때까지 반복한다.

예를 들어 3장의 클러스터장비의 예에서 각 작업시간이 $u=13, u'=22, v=2, v'=3, s=23, s'=25, r=13, r'=15, t_1=t_2=120, t_1'=t_2'=130$라고 하자. 이 때, 만일 공정개발자가 첫 번째 공정에서 공정이 끝난 후에 웨이퍼가 머무는 시간이 20이하여야 한다고 했을 때, 이 제약을 어기는 경우가 발생하는지 여부를 시간차 그래프를 이용하여 분석하자. 먼저 그림 5의 시간차 그래프에서 필요한 노드만을 추출하면 그림 6(a)와 같고, bipartite graph를 만들면 그림 6(b)와 같다. Schwiegelshohn과 Thiele의 알고리즘에 의하면 $_6+_4$, $_4+_4$, $_6+_6$은 0라고 가정하고, 이 값들을 기준으로 $_4-_6$의 값을 계산해야 하는데, 이 값은 $\min\{0-53, 0-143\}=-143$이 된다. 다시 이 값들을 기준으로 $_6+_4$값을 구하면 $\max\{0+28, -143+155\}=28$이 된다. 이를 반복하면 최종적으로 $_4-_6$은 -53, $_6+_4$은 102에서 안정화된다. 정리 2에 의해 가능한 최대체재시간은 $_6+_4-87$로 15이므로 표 2의 20보다 작아 이 클러스터장비의 첫 번째 공정에서는 체재시간제약을 언제나 만족한다.

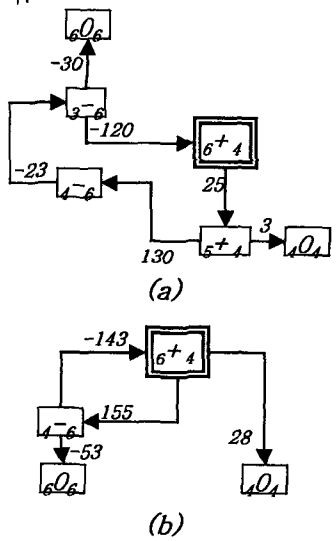

그림 6. 단순화한 시간차 그래프.

그러므로 첫 번째 공정에서의 공정 후 체재시간이 최대가 되는 경우는 transition T_6이 fire한 후 53초 후에 T_4가 fire하고, T_6 이 다시 fire할 때까지 155초가 걸릴 때이다. 이를 간트챠트로 그리면 그림 7과 같다. 즉, terminal node $_60_6$를 기준으로 $_6+_4$의 값을 계산하므로 그림 7의 transition T_6이 $n-1$번째 fire한 후를 기준으로 각 place들이 어떤 값을 가지며 가용해 질 때 $P_{4,3}$의 token이 가용해 진 후 가장 오래 place에 머물게 되는지를 보이고 있다. $_60_6$의 부모노드가 $_4-_6$이므로 T_6이 $n-1$번째 fire한 시점 x_6^{n-1}과 T_4의 n번째 fire시점 x_4^n의 차는 최소가 되어야 하므로 $P_{6,3}$과 $P_{3,4}$에 들어온 token은 가능한 빨리 가용해 져야 한다. 그리고 $_4-_6$의 부모노드는 $_6+_4$이므로 T_4의 n번째 fire시점 x_4^n과 T_6의 n번째 fire시점 x_6^n의 차는 최대가 되어야 하므로 place $P_{6,5}$와 $P_{5,6}$은 가능한 늦게 가용해 져야 한다. 마지막으로 x_6^{n-1}로 시작한 경로가 x_3^{n+1}에서 만나기 위해서는 x_4^n는 $P_{4,3}$이라는 place를 x_6^n은 $P_{6,3}$이라는 place를 거쳐야 한다. 그러므로 $P_{4,3}$는 가능한 빨리, $P_{6,3}$는 가능한 늦게 가용해 져야하며, 그때, 최대체재시간은 $P_{4,3}$이 가용해 진 후, transition T_3이 $n+1$번째 fire할 때(x_3^{n+1})까지의 차가 된다.

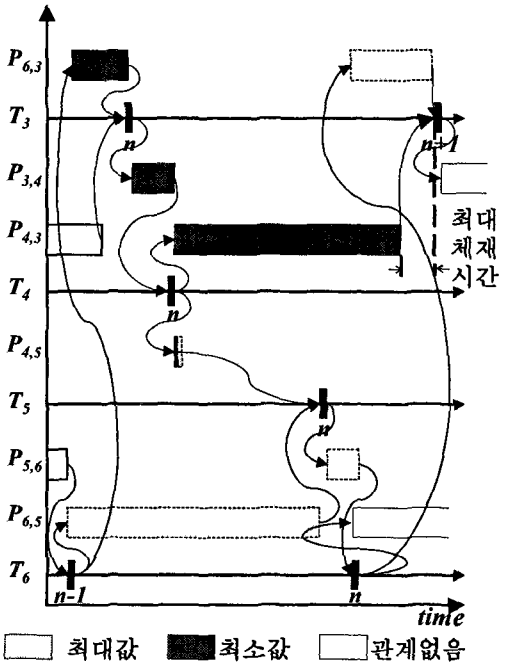

그림 7. 간트챠트.

5.결론

본 논문에서는 시간제약이 있고 작업시간이 일정하지 않은 시스템을 모형화하고 분석할 수 있는 p^+-time Petri net을 제안하였다. 이 net에 대해 transition의 enabling rule과 firing rule, Petri net의 liveness를 정의하였다. 그리고 liveness여부를 판단하기 위해 min-max graph형태의 시간차 그래프를 정의하고 p^+-time event graph에서 시간차 그래프를 찾는 방법을 제안하였다. 그리고 화학박막증착용 양팔 클러스터장비의 예를 모형화하고 분석하였다.

앞으로 제한된 시간변동을 갖는 시간제약이 있는 이산사건시스템의 성질을 좀 더 잘 분석하기 위하여 p^+-time Petri net의 다양한 분석방법을 개발하는 것이 필요하다. 그리고 p^+-time event graph으로 모형화된 시스템과 기존의 확률적인 모형과의 관계에 관한 분석과 클러스터장비의 대기전략에 따른 p^+-time event graph의 모형화 및 분석 등의 추가연구가 필요하다.

References

Khansa W., P. Aygalinc, and J.-P. Denat (1996), Structural analysis of p-time Petri nets, *In Proceedings of the CESA '96, Imacs Multiconference IEEE-SMC*, 127-136.

Kim J.-H, T.-E. Lee, H.-Y. Lee, and D.B. Park (2002), Scheduling of Dual-Armed Cluster Tools with Time Constraints, *In Proceedings of the MASM 2002, International Conference on Modeling and Analysis of Semiconductor Manufacturing.*

Murata T. (1989), Petri nets: Properties, analysis and application, *In Proceedings of the IEEE*, 77(4), 541-580.

Peterson J.L. (1981), *Petri Net Theory and the Modeling of Systems,* Prentice-Hall, Englewood Cliffs, N.J.

Rostami S., B. Hamidzadeh, and D. Camporese (2000), An Optimal Scheduling Technique for Dual-arm Robots in Cluster Tools with Residency Constraints, *In Proceedings of the 39th IEEE Conference on Decision and Control* 3459-3464.

Schwiegelshohn U. and L. Thiele (1999), Dynamic Min-Max Problems, *Discrete Event Dynamic Systems: Theory and Applications,* 9, 111-134.

Shin Y.-H., T.-E. Lee, J.-H. Kim, and H.-Y. Lee (2001), Modeling and implementing a real-time scheduler for dual-armed cluster tools, *Computers in Industry,* 45(1), 13-27.

Sloan R.H. and U. Buy (1996), Reduction rules for time Petri nets, *Acta Informatica,* 33, 687-706.

Venkatesh S., R. Davenport, P. Foxhoven, and Jaim Nulman (1997), A Steady-State Throughput Analysis of Cluster Tools: Dual-Blade Versus Single-Blade Robot, *IEEE Transactions on Semiconductor Manufacturing*, 10(4), 418-424.