

< 논문 >

# 반도체 웨이퍼 팹의 에이전트 기반 스케줄링 방법

윤 현 중\*

(2005년 3월 17일 접수, 2005년 10월 16일 심사완료)

## Agent-Based Scheduling for Semiconductor Wafer Fabrication Facilities

Hyun Joong Yoon

**Key Words :** Scheduling (스케줄링), semiconductor wafer fabrication (반도체 제조 시스템), agent system (에이전트 시스템), hard temporal constraints (경성 시간제약)

### Abstract

This paper proposes an agent-based scheduling method for semiconductor wafer fabrication facilities with hard inter-operation temporal constraints. The scheduling problem is to find the feasible schedules that guarantee both logical and temporal correctness. A proposed multi-agent based architecture is composed of scheduling agents, workcell agents, and machine agents. A scheduling agent computes optimal schedules through bidding mechanisms with a subset or entire set of the workcell agents. A dynamic planning-based approach is adopted for the scheduling mechanism so that the dynamic behaviors such as aperiodic job arrivals and reconfiguration can be taken into consideration.

### 1. 서 론

웨이퍼 제조 공정은 웨이퍼에 집적회로를 생성하는 과정으로서, 하나의 집적회로를 만들기 위해서 일반적으로 수 백 단계의 제조 공정을 거쳐야 한다. 이와 같은 웨이퍼 제조를 위한 설비를 웨이퍼 팹 (wafer fab)이라 한다. 웨이퍼 팹은 단위 공정을 수행하는 워크셀 (workcell)들로 구성되어 있으며, 각 워크셀은 단위 공정을 수행하기 위한 하나 이상의 기계들로 이루어져 있다.

웨이퍼 팹의 고급 스케줄링 문제에서는 시간제약을 고려해야 하는 경우가 생기는데, 기존 연구에서는 시간제약으로 주로 납입기한 (due date)만을 고려해 왔다. 예를 들면, Lu 와 Kumar<sup>(1)</sup>는 납입기한과 버퍼의 우선 순위를 고려한 스케줄링 기법들을 분석하고, 이들이 지연시간 (tardiness)의 평균 및 분산에 미치는 효과를 조사하였다. Kim *et al.*<sup>(2)</sup>과 Kim *et al.*<sup>(3)</sup>는 완성된 웨이퍼의 지연시간을 줄이는 할당규칙들을 제안하였고, Mason *et*

al.<sup>(4)</sup>는 가중치를 고려한 지연시간을 줄이기 위한 재스케줄링 (rescheduling) 방법들을 평가하였다.

웨이퍼 팹에서의 시간제약 조건에는 납입기한 외에도, 연속된 두 개 이상의 공정 사이에도 존재할 수 있다. 즉, 어떤 공정은 선행 공정이 끝난 후 정해진 시간 안에 공정을 시작해야 하는데,<sup>(5)</sup> 이와 같은 공정간 시간제약은 주로 제조 공정의 기술적인 문제에 기인한다. 예를 들면, 노 (furnace)에서의 공정은 세척 공정 후 2 시간 내에 시작되어야 하는데,<sup>(6)</sup> 이러한 시간제약을 만족하지 못 할 경우 웨이퍼는 세척 공정을 다시 거쳐야 된다. 또, 노에서의 공정을 마친 웨이퍼는 그 다음 공정을 특정 시간 내에 시작해야 하는데, 시간제약을 만족하지 못한 웨이퍼는 노에서 공정을 다시 거쳐야 한다. 이와 같이 공정간 시간제약을 만족하지 못해 선행 공정을 다시 거쳐야 하는 현상은, 웨이퍼 흐름을 불안정하게 만들어 생산성 저하를 야기할 수 있다.

시간제약이 있는 생산시스템 스케줄링 문제는, 실시간 시스템 (real-time system)의 스케줄링 문제로 고려될 수 있다. 실시간 시스템이란 시스템의 정확성 (correctness)이 논리적 정확성과 시간적 정확성으로 표현되는 시스템을 의미한다.<sup>(7)</sup> 생산시스템의 스케줄링 측면에서 보면, 논리적 정확

\* 회원, 대전시 유성구 구성동 373-1 한국과학기술원 기계기술연구소  
E-mail : yunkji@kaist.ac.kr  
TEL : (042)869-5229

성은 유한 자원의 용량 제약 조건 및 공정들의 선후관계 조건 등에 관련이 있으며, 시간적 정확성은 공정간 시간제약이나 납입기한과 관련이 있다. 실시간 시스템은 경성 마감시간 (hard deadline)을 갖는 시스템과 연성 마감시간 (soft deadline)을 갖는 시스템으로 나뉜다. 경성 마감시간을 갖는 시스템에서는 시간제약을 반드시 만족해야 하는 반면, 연성 마감시간을 갖는 시스템에서는 시간제약이 중요하기는 하지만 반드시 만족하지 않아도 된다.<sup>(7)</sup> 또, 실시간 시스템의 작업 (job)은 작업이 도착하는 시각에 따라 주기적, 비주기적, 특별적 (sporadic) 작업으로 나뉜다. 주기적 작업은 작업이 고정된 주기로 시스템에 도착하는 작업을, 비주기적 작업은 불규칙적으로 시스템에 도착하는 작업을 의미한다. 특별적 작업이란 비주기적이지만 두 개의 연속된 작업들 간에 최소 시간 간격이 존재하는 작업을 의미한다.

실시간 시스템의 스케줄링 문제는 주어진 논리적 및 시간적 제약조건을 만족시키도록 유한 자원과 시간을 할당하는 문제이다. 실시간 시스템의 스케줄링 기법은 주로 컴퓨터 과학과 경영 과학 분야에서 연구되어왔다. Ramamritham 과 Stankovic<sup>(8)</sup>은 실시간 시스템의 스케줄링 기법을 정적 스케줄링과 동적 스케줄링으로 나누고, 컴퓨터 과학 분야에서는 주로 동적 스케줄링에, 경영 과학 분야에서는 주로 정적 스케줄링에 관심을 두고 있다고 밝혔다.

정적 스케줄링 기술은 주기적 작업을 갖는 실시간 시스템에 적용이 가능하며, 오프라인 (off-line)에서 실행 가능성 (feasibility)을 분석한다. 경영 과학 분야에서는 비체류 (no-wait) 스케줄링 문제에 이와 같은 정적 스케줄링 기술을 주로 사용하고 있다. 비체류 스케줄링 문제란 철강 제조 공정과 같이 연속 공정으로 이루어진 생산 시스템을 의미한다. Hall 과 Sriskandarajah<sup>(9)</sup>는 비체류 스케줄링에 관한 기존 연구 방법들을 비교, 분석하였다. 특히, 비체류 생산시스템의 한 종류인 호이스트 (hoist) 생산 시스템에의 스케줄링에 관한 많은 연구가 있었다. 대부분의 호이스트 스케줄링 기법은 분기한정법 (branch-and-bound) 방법을 이용하여 오프라인에서 주기적 스케줄을 구하는데 초점을 맞추고 있다.<sup>(10)</sup>

동적 스케줄링 기술은 비주기 작업이나 기계 고장과 같은 시스템의 불확실성을 고려할 수 있다는 장점이 있다. 동적 스케줄링 기술은 크게 동적 계획 (dynamic planning-based) 방법과 동적 할당 (dynamic best effort) 방법으로 나뉜다.<sup>(8)</sup> 동적 계획 방법은 새로운 작업이 시스템에 도착할 때

실행 가능성을 온라인 (on-line)으로 분석하여, 주어진 시간제약을 만족시킬 수 있을때만 작업을 받아들인다. 반면, 동적 할당 방법은 실행 가능성 여부를 확인하지 않고 시간제약을 맞추기 위한 최선의 결정을 내린다. 따라서, 시간제약을 만족시키는 것을 보장할 수 없다. 동적 할당 방법의 대표적인 예로는 EDF (Earliest Deadline First)와 LLF (Least Laxity First)가 있다.<sup>(11)</sup> 결론적으로, 동적 계획 방법은 경성 마감시간이 있는 시스템에 적합하며, 동적 할당 방법은 연성 마감시간이 있는 시스템에 적합하다.

본 논문에서는 경성 공정간 시간제약 (hard inter-operational temporal constraints)이 있는 웨이퍼 맵의 실시간 스케줄링 문제를 다루며, 비주기적 로트<sup>1</sup> (lot) 및 시스템 재구성과 같은 동적인 거동을 고려한다. 이를 위하여 본 논문에서는 동적 계획 접근법을 적용한다. 그러나, 실제로 존재하는 대부분의 실시간 시스템 스케줄링 문제는 높은 계산 복잡도를 갖는다고 알려져 있다. 따라서, 본 논문에서는 실시간 적용성을 만족시키기 위하여 다중 에이전트 개념을 적용한 분산 스케줄링 기법을 이용하며, 다항식 함수 시간을 갖는 스케줄링 기법을 제안하는 것을 목적으로 한다.

## 2. 시간제약 워크셀 그룹

공정간 시간제약은 일반적으로 웨이퍼 맵 전반적으로 주어지기 보다는, 몇 개의 워크셀들의 집합에서 국지적으로 주어진다. 시간제약 워크셀 그룹, 최소 시간제약 워크셀 그룹, 제어가능 워크셀은 다음과 같이 정의된다.

**정의 1:** 시간제약 워크셀 그룹  $G$  는 한 개 이상의 공정간 시간제약이 있는 워크셀들의 집합이다.  $G$  에 포함된 워크셀로부터 외부의 워크셀 사이에, 외부의 워크셀로부터  $G$  에 포함된 워크셀 사이에, 공정간 시간제약이 존재하지 않는다.

**정의 2:** 시간제약 워크셀 그룹  $G$  가 또 다른 시간제약 워크셀 그룹을 포함하지 않을때, 이를 최소 시간제약 워크셀 그룹이라 한다

<sup>1</sup> 웨이퍼 맵에서 로트는 카세트 또는 FOUP (Front Opening Unified Pod)을 의미하며, 하나의 로트는 일반적으로 25 개의 웨이퍼를 포함한다.

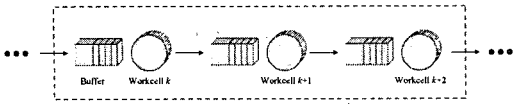


Fig. 1 A temporal constraint workcell group

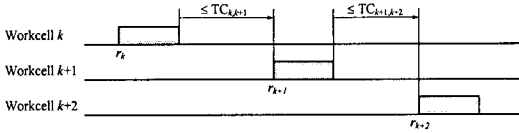


Fig. 2 Hard inter-operation temporal constraints

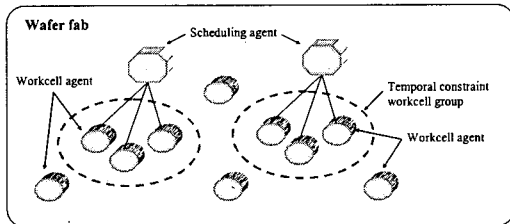


Fig. 3 The scheduling architecture in a wafer fab

정의 3: 시간제약 워크셀 그룹  $G$  와  $G$  에 포함된 워크셀  $g$  에서,  $g$  가  $G$  외부의 워크셀로부터 연결되는 공정 흐름을 가질때,  $g$  는 제어가능 워크셀이라 한다.

Fig. 1 은 워크셀  $k, k+1, k+2$  로 이루어진 시간제약 워크셀 그룹과, 제어가능 워크셀  $k$  를 나타낸다. 워크셀  $k$  와  $k+1$  의 공정간 시간제약과, 워크셀  $k+1$  과  $k+2$  의 공정간 시간제약은 각각  $TC_{k,k+1}$  과  $TC_{k+1,k+2}$  로 나타낸다. 각 워크셀은 도착한 로트들을 저장할 수 있는 버퍼가 있다. Fig. 2 에서 볼 수 있듯이 시간제약 워크셀 그룹의 스케줄링 문제는, 주어진 시간제약을 만족하도록 각 워크셀에서의 로트 투입 시간을 결정하는 문제이다.

### 3. 에이전트기반 스케줄링 시스템

비메모리 반도체 시장 요구가 커짐에 따라, 반도체 산업의 생산 패러다임은 대량 생산에서 대량 구매로, 예측기반 생산에서 주문기반 생산으로 진화하고 있다. 따라서, 차세대 반도체 팩은 보다 지능적이고 민첩해져야 한다. 보다 효율적인 반도체 팩의 스케줄링 시스템을 위하여, 분산 환경의 에이전트 기반 접근법을 제시한 연구들이 최근 이루어지고 있다. Mönch *et al.*<sup>(12)</sup> 및 Mönch 와 Stehli<sup>(13)</sup> 는 웨이퍼 팩의 공정 제어를 위한 에이전트 기반 제어 시스템과 온톨로지 (ontology)를 제시하였으

며, 홀로닉<sup>2</sup> (holonic) 생산 시스템의 레퍼런스 아키텍처인 PROSA (Production Resource Order Staff Architecture)<sup>(14)</sup>를 기반으로 계층적 구조를 갖는 아키텍처를 제시하였다. Yu 와 Huang<sup>(15)</sup>은 반도체 주조 팩 (foundry fab)에서 고객 주문 처리과정을 에이전트 개념을 이용하여 모델링하고, 고객 주문 처리과정을 4 단계의 프로세스로 구분하였다. 각 프로세스들은 에이전트를 가지고 있으며, 에이전트 사이에 통신을 위한 일반 메시지 전송 플랫폼을 제시하였다. Cheng *et al.*<sup>(16)</sup>은 반도체 산업의 홀로닉 생산 실행 시스템 개발을 위한 시스템적 접근 방법을 제안하였다.

본 단원에서는 웨이퍼 팩을 위한 지능적 다중 에이전트를 이용한 스케줄링 방법을 제시한다. 본 논문에서 에이전트는 자기 자신의 상태와 행동 및 제어 스레드 (thread)를 갖고, 동시에 복잡한 문제를 해결하기 위해 다른 개체들과 통신하고 상호작용하는 능력을 가진 소프트웨어 개체로 정의된다. 제안된 다중 에이전트 기반 스케줄링 시스템은 서비스 에이전트와 자원 에이전트로 이루어진다. 서비스 에이전트에는 스케줄링 에이전트가 해당하며, 이는 스케줄 생성 외에 중개자<sup>(17)</sup> (mediator) 역할을 한다. 스케줄링 에이전트는 공정간 시간제약이 있는 워크셀들을 시간제약 워크셀 그룹으로 묶고, 그들을 모니터링하고 관리하는 역할을 한다. 자원 에이전트에는 워크셀 에이전트와 기계 에이전트가 해당된다. 워크셀 에이전트들은 스케줄링 에이전트가 시간제약을 만족하는 스케줄을 계산할 수 있도록, 입찰 (bidding) 메커니즘을 이용하여 스케줄링 에이전트와 협업한다. 기계 에이전트는 워크셀 에이전트에 의해 할당된 로트의 공정을 수행한다. 기계에서 이루어지는 로트의 공정은 일반적으로 하나 이상의 하위공정으로 이루어져 있으며, Yoon<sup>(18)</sup>은 단일 기계에서 시간제약이 있는 비주기적 스케줄링 방법을 제안하였다. Fig. 3 은 웨이퍼 팩에서 다중 에이전트 개념을 이용한 스케줄링 시스템 아키텍처를 보여준다. Fig. 3 에는 두 개의 시간제약 워크셀 그룹이 있으며, 각 시간제약 워크셀 그룹은 스케줄링 에이전트에 의해 관리된다.

<sup>2</sup> Wyns<sup>(14)</sup>은 홀로닉 생산시스템을 “홀론 (holon)이라 불리는 자율적이고 상호 작용하는 에이전트들로 이루어진 분산 생산시스템”이라 정의했다.

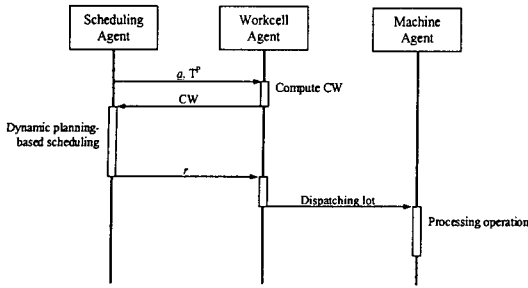


Fig. 4 A sequence diagram for scheduling mechanism

### 4. 스케줄링 메카니즘

Fig. 4 는 제어가능 워크셀의 버퍼에 있는 새로운 로트의 스케줄링 과정을 시퀀스 다이어그램으로 나타낸 것이다. 스케줄링 메카니즘은 다음의 네 단계로 이루어져 있다.

**단계 1:** 스케줄링 에이전트는 각 워크셀에 로트의 정보를 보낸다. 로트의 정보는 로트가 각 워크셀에 투입될 예상시각의 최하 한계치  $a$  과 각 워크셀에서 가공 시간  $T^p$  를 포함한다.

**단계 2:** 각 워크셀은 로트의 가공이 실제로 시작될 수 있는 시간 간격들을 표현하는 시간제약 집합 (temporal constraint set)  $CW$  를 계산한다. 계산된  $CW$  는 스케줄링 에이전트로 보내진다.

**단계 3:** 스케줄링 에이전트는 논리적, 시간적 정확성을 만족시키는 최적해를 구하고, 각 워크셀에 로트를 투입하는 시각  $r$  을 계산한 후, 계산된 시각에 따라 로트를 워크셀 에이전트로 할당한다.

**단계 4:** 각 워크셀은 스케줄링 에이전트로부터 받은 스케줄에 따라 로트들을 해당 기계로 할당한다.

각 단계에서 계산된 스케줄은 이미 투입된 로트에 대해 결정된 스케줄을 변경하지 않는다. 또 본 논문에서는 배치(batch) 공정은 고려하지 않는다.

시간간격의 표현 방법으로 Nissanke<sup>(7)</sup>에 의해 제시된 정의를 따른다. 즉,

$$t \in [t_1, t_2] \text{ if and only if } t_1 \leq t \text{ and } t < t_2 \quad (1)$$

이다. 시간제약 집합은 시간간격들의 집합이며, 이는 Dechter *et al.*<sup>(19)</sup>의 정의를 따른다. 이때, 시간제약 집합은 오름차순으로 정렬된 서로소 집합이라 가정한다. 즉,

$$C = \{[L_1, U_1], [L_2, U_2], \dots, [L_m, U_m]\} \quad (2)$$

이며,  $L_1 < U_1 < \dots < L_m < U_m$  의 관계를 갖는다.

**정의 4<sup>(19)</sup>:**  $T = \{I_1, I_2, \dots, I_i\}$ 와  $S = \{J_1, J_2, \dots, J_m\}$ 를 시간제약 집합이라고 하면,

(1)  $T$ 와  $S$ 의 교집합  $T \cap S$ 은 두 집합 모두에 의해 허용되는 시각만을 포함한다.

(2)  $T$ 와  $S$ 의 합집합  $T \cup S$ 은 두 집합 중 어찌 하나에 의해 허용되는 시각만을 포함한다.

본 논문에서는 이동연산자  $\oplus$ 와  $EARLIEST(\cdot)$  및  $LATEST(\cdot)$ 를 다음과 같이 정의한다.

**정의 5:**  $C = \{[L_1, U_1], [L_2, U_2], \dots, [L_m, U_m]\}$ 를 시간제약 집합이라 하면,

(1) 이동연산자  $\oplus$ 는 다음과 같이 정의된다.

$$C \oplus [a, b] = \{[L_1 + a, U_1 + b], [L_2 + a, U_2 + b], \dots, [L_m + a, U_m + b]\} \quad (3)$$

(2)  $EARLIEST(C)$ 는  $L_1$ 이고  $LATEST(C)$ 는  $U_m$ 이다.

최소 시간제약 워크셀 그룹의 제어가능 워크셀의 버퍼에 있는 새로운 로트의  $\lambda$  번째 공정을  $O_\lambda$  ( $\lambda = 1, 2, \dots, A$ )라 하고,  $W_\lambda$ 를  $O_\lambda$ 가 수행되는 워크셀이라 할 때, 로트가  $W_\lambda$ 에 도착하는 시각의 최소값  $a_\lambda$ 는 다음의 식으로 구한다.

$$a_1 = (\text{current time}),$$

$$a_\lambda = a_1 + \sum_{k=1}^{\lambda-1} T_k^p \quad (\lambda = 2, 3, \dots, A) \quad (4)$$

여기서  $T_k^p$ 는  $O_k$ 의 가공시간을 의미한다.

스케줄링 메카니즘의 두 번째 단계인 시간제약 집합  $CW$ 는 각 워크셀 에이전트에 의해 다음의 절차로 계산된다.  $W_\lambda$ 가  $\mu(\lambda)$ 개의 동일한 병렬 기계들의 집합  $\{M_{(\lambda,1)}, M_{(\lambda,2)}, \dots, M_{(\lambda,\mu(\lambda))}\}$ 을 가지고 있다고 하면,  $CM_{(\lambda,i)}$ 는  $M_{(\lambda,i)}$ 에서  $O_\lambda$ 가 시작될 수 있는 시간제약 집합을 의미한다. 즉,  $CM_{(\lambda,i)}$ 는 다음의 식으로 부터 구해진다.

$$CM_{(\lambda,i)} = \{[a_i, b_{(\lambda,i)}]\} \cap (CB_{(\lambda,i)} \oplus [0, -T_\lambda^p]) \quad (5)$$

여기서  $CB_{(\lambda,i)}$ 는  $M_{(\lambda,i)}$ 에 이미 할당된 로트들의 공정 불력을 나타내는 시간제약 집합의 여집합이다. 즉, 새로운 로트의  $O_\lambda$ 가 수행될 수 있는 시간제약 집합을 의미한다. 또,  $CM_{(\lambda,i)}$ 의 최소 시각은  $a_i$ 보다

크고,  $CM_{(\lambda,i)}$ 의 최대 시각은  $b_{(\lambda,i)}$ 보다 작거나 같다. 일반적인 경우  $b_{(\lambda,i)}$ 값은 무한대의 값을 갖는다. 그러나,  $M_{(\lambda,i)}$ 가 유지, 보수나 제거될 계획이 있다면  $b_{(\lambda,i)}$ 값은 유한한 값을 가질 수 있다.  $W_\lambda$ 에서  $O_\lambda$ 의 시작 가능 시간간격을 의미하는  $CW_\lambda$ 는 다음과 같이 정의된다.

$$CW_\lambda = \bigcup_{i=1-\mu(\lambda)} CM_{(\lambda,i)}. \quad (6)$$

스케줄링 메카니즘의 세번째 단계에서, 스케줄링 에이전트는 각 워크셀 에이전트로부터 받은  $CW$ 를 이용하여 스케줄을 계산한다. 시간제약 워크셀 그룹에서 스케줄링 문제는 일반적인 시간제약 만족문제<sup>(19)</sup> (general temporal constraint satisfaction problem)의 특수한 경우로 볼 수 있다. 즉,  $O_\lambda$ 를  $W_\lambda$ 에 투입하는 시각을  $r_\lambda$ 라 할 때  $A$ 개의 변수  $r_\lambda$  ( $\lambda = 1, \dots, A$ )를 결정하는 문제이며,  $i < j$ 를 만족하는 모든  $i, j$  쌍 ( $i, j = 1, 2, \dots, A$ )에 대하여 다음과 같은 복수형의 시간제약 조건을 갖는다.

$$T_i^P + T_{i,j}^{TR} \leq r_j - r_i \leq T_i^P + TC_{i,j} \quad (7)$$

또,  $CW_\lambda = \{[L_{\lambda,1}^W, U_{\lambda,1}^W], \dots, [L_{\lambda,\varphi(\lambda)}^W, U_{\lambda,\varphi(\lambda)}^W]\}$ 라 할 때,  $r_\lambda$  ( $i = 1, 2, \dots, A$ )는 다음과 같은 단일형의 시간제약 조건들의 집합을 갖는다.

$$(L_{i,1}^W \leq r_i < U_{i,1}^W) \vee \dots \vee (L_{i,\varphi(i)}^W \leq r_i < U_{i,\varphi(i)}^W), \quad (8)$$

식 (7)의 복수형 시간제약 조건은 공정 선후관계 조건 및 로트의 시간제약을 의미한다.  $T_{i,j}^{TR}$ 와  $TC_{i,j}$ 는 각각 로트를  $W_i$ 에서  $W_j$ 로 이송하는데 소요되는 시간과, 공정  $i$ 와  $j$  사이의 경성 시간제약을 의미한다. 식 (8)의 단일형 시간제약 조건들의 집합은 워크셀에서 새로 투입되는 로트의 공정 블록이 기존의 공정 블록과 겹치지 않는 조건을 의미한다. 일반적인 시간제약 만족 문제의 해의 존재 여부를 결정하는 것은 NP-hard로 알려져 있으며, 해를 구하는 계산 복잡도는  $O(n^3 \cdot k)$ 로 알려져 있다.<sup>(19)</sup> 여기서  $n, e, k$ 는 각각 변수의 개수, 복수형 및 단일형 시간제약 조건 집합들의 개수, 제약조건 집합 원소 개수의 최대값이다.

스케줄링 에이전트는 새로 투입된 로트의 최적 스케줄을 찾기 위해 앞서 설명한 시간제약 만족 문제를 푼다. 즉, 제안된 스케줄링 알고리즘은 최소 시간제약 워크셀 그룹에서 마지막 공정을 마친

는 시각을 최소화 하는 스케줄을 다항식 함수 단계 내에 계산한다. 제안된 스케줄링 알고리즘은 *FEASIBLE\_SPACE*와 *OPTIMAL\_SCHED*로 이루어져 있다. *FEASIBLE\_SPACE*는 새로 투입되는 로트에 대해  $r_\lambda$ 의 시간제약 집합인  $CS_\lambda$  ( $\lambda = 1, 2, \dots, A$ )를 구한다. 즉,  $CS_\lambda$ 는  $r_\lambda$ 가 실행될 수 있는 시간간격들의 집합이다. *OPTIMAL\_SCHED*는 로트의 마지막 공정을 끝내는 시각이 최소로 하도록 하는  $r_\lambda$  ( $\lambda = 1, 2, \dots, A$ )를 계산한다. *FEASIBLE\_SPACE*의 의사코드는 다음과 같다.

#### FEASIBLE\_SPACE

Input  $CW_\lambda$  ( $\lambda = 1, 2, \dots, A$ )

Output  $CS_\lambda$  ( $\lambda = 1, 2, \dots, A$ )

```

1   $\lambda \leftarrow 1$ 
2   $CS_1 \leftarrow CW_1$ 
3  while  $\lambda < A$ 
4      do  $\lambda \leftarrow \lambda + 1$ 
5      compute  $CS_\lambda$ 

```

*FEASIBLE\_SPACE*는  $CS_1$ 를  $CW_1$ 로 설정하는 것으로 시작하여, 두번째 공정부터  $A$ 번째 공정까지  $CS_\lambda$ 를 다음 식을 이용하여 순차적으로 계산한다.

$$CS_\lambda = \bigcap_{k=1-\lambda-1} \{CS_k \oplus [T_k^P + T^{TR}, T_k^P + CT_{k,\lambda}]\} \cap CW_\lambda \quad (9)$$

우측항에서 전반부는 공정 선후조건과 공정간 시간제약을 의미하며, 후반부의  $CW_\lambda$ 는 새로 투입되는 로트의 공정 블록이 이전에 투입된 로트의 공정 블록과 겹치지 않는 조건을 의미한다.

또, *OPTIMAL\_SCHED*의 의사코드는 다음과 같다.

#### OPTIMAL\_SCHED

Input  $CS_\lambda$  ( $\lambda = 1, 2, \dots, A$ )

Output  $r_\lambda$  ( $\lambda = 1, 2, \dots, A$ )

```

1   $\lambda \leftarrow A$ 
2   $r_\lambda \leftarrow \text{EARLIEST}(CS_\lambda)$ 
3  while  $\lambda > 1$ 
4      do  $\lambda \leftarrow \lambda - 1$ 
5       $r_\lambda \leftarrow \text{LATEST}(\{[0, r_{\lambda+1} - T^{TR} - T_\lambda^P]\} \cap CS_\lambda)$ 

```

*OPTIMAL\_SCHED*는 마지막 공정이 끝나는 시각을 최소화 하도록 하면서,  $A-1$ 번째 공정부터 첫번째 공정까지  $r_\lambda$ 를 역순으로 계산한다. 각  $r_\lambda$ 는 각 버퍼에 체류 시간을 최소화 하도록 결정된다. 즉, *FEASIBLE\_SCHED*에서 구해진 실행가능 해 공간 중에서 *OPTIMAL\_SCHED*는 마지막 공정의 끝나는 시각을 최소화 하는 하나의 스케줄을 결정한다.

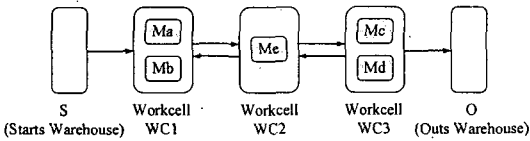


Fig. 5 Workcell layout

Table 1 Mapping between processes and machines

Machine	Process	Description	Processing Time
Ma & Mb	S1 & S5	Diffusion	S1 = 75 mins/lot S5 = 85 mins/lot
Mc & Md	S2 & S4	Ion Implantation	S2 = 30 mins/lot S4 = 50 mins/lot
Me	S3 & S6	Lithography	S3 = 55 mins/lot S6 = 10 mins/lot

Table 2 Computing time

Number of Lots	Computation Time	Std. Devn. of Computation Times	Average Computation Time Per Lot
10	0.0280 s	0.0052 s	0.0028 s
25	0.0675 s	0.0097 s	0.0027 s
50	0.1550 s	0.0226 s	0.0031 s
75	0.2470 s	0.0258 s	0.0033 s
100	0.3106 s	0.0244 s	0.0031 s

결론적으로, 스케줄링 에이전트가 새로 투입되는 로트에 대하여, 각 공정을 시작하는 시각  $r$  를 계산하는 의사코드는 다음과 같다.

```

Input  $CW_\lambda (\lambda = 1, 2, \dots, A)$ 
Output  $r_\lambda (\lambda = 1, 2, \dots, A)$ 
1 call FEASIBLE_SPACE
2 call OPTIMAL_SCHED
3 for every  $\lambda$  for  $\lambda = 1, 2, \dots, A$ 
4 do return  $r_\lambda$ 
    
```

제안된 스케줄링 알고리즘의 계산 복잡도는 다음과 같이 구할 수 있다. 각  $l$  과  $m$  의 원소를 개수 갖는 두 개의 시간제약 집합의 교집합을 계산하는 과정은  $O(l+m)$ 라 알려져 있다.<sup>(20)</sup> FEASIBLE\_SPACE 의 3-5 줄의 루프는  $A-1$  번 수행되고,  $CW_\lambda$  원소의 최대값을  $L$  이라 할 때, 마지막 5 번째 줄의 복잡도는  $O(A^2 \cdot L)$ 이므로, FEASIBLE\_SPACE 의 계산 복잡도는  $O(A^3 \cdot L)$ 이다. 마찬가지로 OPTIMAL\_SCHED 의 계산 복잡도를 구하면  $O(A^2 \cdot L)$ 이다. 따라서, 제안된 스케줄링 알고리즘의 계산 복잡도는  $O(A^3 \cdot L)$ 이다.

5. 시뮬레이션

본 단원에서는 제안된 스케줄링 메커니즘을 5 개의 기계로 구성된 인텔 미니 팸<sup>(21)</sup>을 이용하여

평가하였다. Pa, Pb 의 두가지 종류의 제품용 로트와 TW 의 테스트 로트를 생산하며, 모든 로트는 starts >> S1 >> S2 >> S3 >> S4 >> S5 >> S6 >> outs 의 순서로 공정을 진행한다. 본 논문에서는 배치 작업을 고려하지 않았으므로 시뮬레이션에서도 배치 작업을 고려하지 않았다. Fig. 5 와 Table 1 은 워크셀들의 레이아웃 및 공정의 흐름을 보여준다. 로트의 이송 시스템은 S < WC1 < WC2 < WC3 < O 사이에 루프를 형성하며, 각 루프에서 로트의 이송시간은 4 분으로 가정하였다.

시뮬레이션을 통한 성능 평가를 위해, 마감시간 여유도 (deadline slackness)를 새롭게 정의하였으며, 다음과 같이 정의된 성능지수들을 이용한다.

- 마감시간 여유도는  $T_i^P$  에 대한  $TC_{i,i+1}$  의 비율이다. 즉, (마감시간 여유도) :=  $TC_{i,i+1} / T_i^P$  이다.
- 단위시간당 생산량 (throughput rate)은 특정 시간동안 생산되는 로트의 양이다.
- 사이클 시간 (cycle time)은 마지막 공정을 끝낸 연속된 두 로트의 평균 간격 시간으로서, 단위 시간당 생산량의 역수와 같다.
- 흐름 시간 (flow time)은 특정 로트가 웨이퍼 팸에 투입되어서 마지막 공정을 수행하고 나올때까지 걸리는 시간이다.
- 지연시간 (tardiness)는 특정 로트의 모든 공정 지연시간의 총합이다. 이때, 공정 지연시간은 공정이 시간제약을 초과하여 끝난 시간이다.

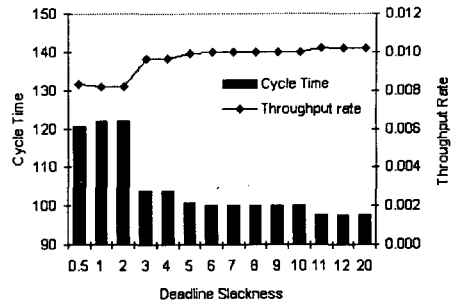
미니 팸에서 마지막 공정을 제외한 모든 공정에 공간 시간제약이 있다고 가정하였다. 따라서, 미니 팸에는 하나의 최소 시간제약 워크셀 그룹이 존재한다. Table 2 는 투입된 로트의 개수에 따른 제안된 스케줄링 알고리즘의 계산 시간을 보여준다. 인텔 모바일 펜티엄 III 700MHz, 256MB SDRAM 환경에서 시뮬레이션을 수행하였으며, 각 계산 시간은 20 번의 시뮬레이션으로부터 구해진 시간들의 평균이다. 각 로트의 스케줄을 계산하기 위한 평균 시간은 약 0.0027~0.0033 초임을 알 수 있다.

다음은 Pa, Pb, TW 가 서로 다른 가공 순서를 갖는다고 가정하였다. 즉, Pa: starts >> S1(Ma) >> S2(Mc) >> S3(Me) >> S4(Md) >> S5(Mb) >> S6(Me) >> outs, Pb: starts >> S2(Mc) >> S1(Ma) >> S3(Me) >> S5(Mb) >> S4(Md) >> S6(Me) >> outs, and TW: starts >> S4(Md) >> S5(Mb) >> S6(Me) >> S1(Ma) >> S2(Mc) >> S3(Me) >> outs 이며, 각 로트 종류의 생산 비율은 1:1:1 로 가정하였다. Fig. 6 은 제안된 스케줄링 알고리즘을 적용했을때 마감시간 여유도와

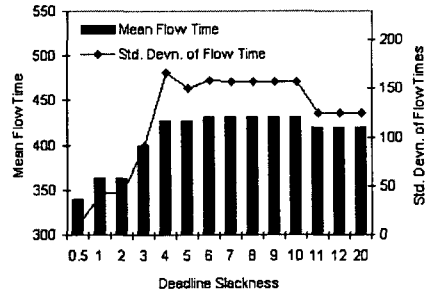
성능지수들의 관계를 보여준다. Fig. 6(a)는 마감시간이 느슨해질수록, 사이클 시간이 감소하다가 마감시간 여유도가 11 이 넘어가면 97.7 분으로 수렴하는 것을 보여준다. 마찬가지로, 단위시간당 생산량은 마감시간이 느슨해 질수록 증가하다가 0.01023 로트/분로 수렴한다.

또, 흐름시간도 웨이퍼 팹에서 중요한 성능 지수 중 하나인데, 생산되는 웨이퍼의 질을 높이고 주문 납기일을 맞추기 위해서는 흐름시간의 평균과 표준편차를 줄이는 것이 유리하다. 그러나, 평균 흐름시간의 지나친 감소는 단위시간당 생산량의 감소를 의미한다.<sup>(22)</sup> 따라서, 흐름시간과 단위시간당 생산량 사이의 보정이 요구된다. Fig. 6(b)는 마감시간 여유도가 증가함에 따라 평균 흐름시간이 증가하는 것을 보여준다. 이는 마감시간이 느슨해 질수록 로트가 워크셀의 버퍼에서 기다리는 시간이 길어지기 때문이다. 다시 말해서, 마감시간 여유도가 작아질 경우 실행 가능한 해를 구하기 위하여 로트의 투입 시각을 늦추어야 하는 상황이 생길 수 있는데, 이러한 경우 기존에 이미 투입된 공정 블록들의 배치에 따라 흐름시간이 줄어들 수 있으며, 일반적으로 투입 시각을 늦출수록 기존 공정 블록들의 밀도가 낮아지기 때문에 흐름시간이 감소하는 추세를 보이는 것이다.

다음은 제안된 스케줄링 알고리즘을 FIFO (First-In-First-Out)와 EDF (Earliest Deadline First)의 두 할당규칙과 비교하였다. FIFO는 워크셀 버퍼에 가장 빨리 도착한 로트를 우선적으로 선택하며, EDF는 마감시간이 빠른 로트를 우선적으로 선택한다. 할당규칙을 적용할때 평균 지연시간과 평균 흐름시간이 발산하는 것을 방지하기 위하여, 두 할당규칙과 WIP (Work-In-Process) 규칙을 함께 사용하였다. WIP 규칙은 최소 시간제한 워크셀 그룹 내에 있는 로트의 개수를 미리 설정된 값으로 유지하는 법칙이다. Fig. 7은 WIP 수준이 1 에서 10 으로 증가함에 따라 EDF 를 적용했을때 시뮬레이션 결과를 보여준다. 여기서 마감시간 여유도는 2 로 하였다. WIP 수준이 증가함에 따라 사이클 시간이 감소하다가 수렴하는 것을 볼 수 있는데, 이는 동시에 WIP 수준이 증가함에 따라 단위시간당 생산량이 증가하다가 수렴함을 의미한다. 이러한 수렴점은 해당 시간제한 워크셀 그룹의 최대 허용 용량과 관련이 있다. 그러나, 평균 흐름시간 및 평균 지연시간은 WIP 수준이 증가함에 따라 같이 증가하여 발산한다. Fig. 7에서 볼 수 있듯이 WIP 수준이 4 보다 작을때 평균 지연시간은 0 의 값을 갖는다.



(a)



(b)

Fig. 6 Simulation results of the proposed scheduling algorithm (a) cycle time and throughput, and (b) mean and standard deviation of flow times

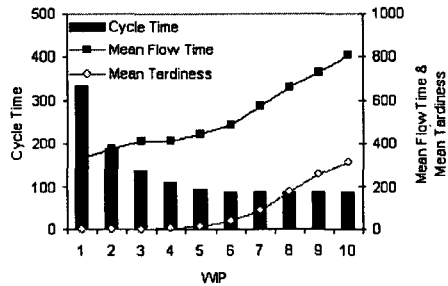


Fig. 7 Simulation results of EDF as increasing WIP level

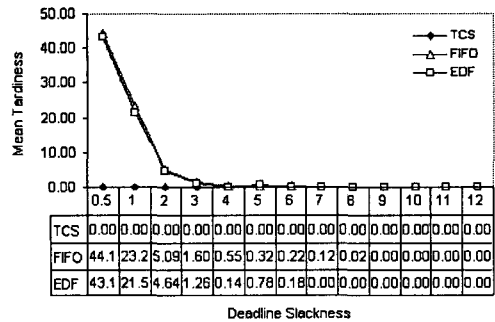


Fig. 8 Comparison of mean tardiness under TCS, FIFO, and EDF

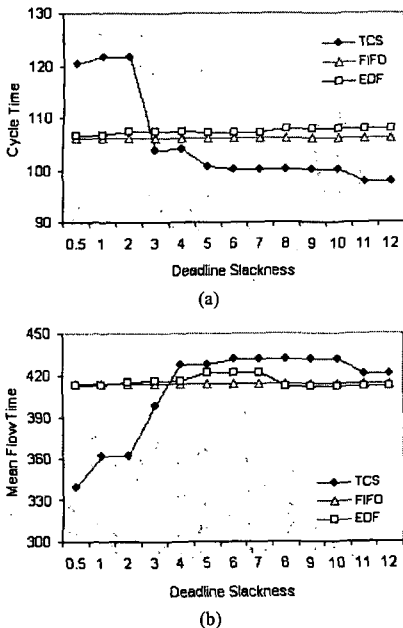


Fig. 9 Comparison of (a) cycle time and (b) mean flow time under TCS, FIFO, and EDF

Fig. 8, 9는 제안된 스케줄링 알고리즘 (TCS)과 WIP 수준을 4로 했을 때 FIFO 및 EDF를 비교한 것이다. Fig. 8은 평균 지연시간을 비교하여 보여주는데, 제안된 스케줄링 알고리즘은 모든 경우에 있어서 시간제약 조건을 만족시키는 것을 알 수 있다. 또, EDF가 FIFO보다 평균 지연시간을 줄이는데 조금 더 효과적임을 알 수 있다. 즉, FIFO는 마감시간 여유도가 9 이상일 때 평균 지연시간이 0가 되는데 비해, EDF는 마감시간 여유도가 7 이상일 때 평균 지연시간이 0가 된다.

Fig. 9는 제안된 알고리즘, FIFO, EDF를 적용했을 때 사이클 시간과 평균 흐름시간을 비교한다. Fig. 9(a)에서 보면 마감시간 여유도가 0.5에서 2 사이의 비교적 적은 값을 가질 때, FIFO와 EDF가 제안된 스케줄링 알고리즘보다 낮은 사이클 시간을 갖는다. 그러나, 이와 같은 낮은 마감시간 여유도 영역에서는 FIFO와 EDF가 0가 아닌 평균 지연시간을 갖는다는 것을 유념해야 한다. 반면, 마감시간 여유도가 3 이상의 값을 가지면 제안된 스케줄링 알고리즘이 FIFO나 EDF보다 좋은 성능을 보여준다. 또, Fig. 9(b)에서 보면 높은 마감시간 여유도 영역에서, FIFO와 EDF가 제안된 스케줄링 알고리즘보다 낮은 평균 흐름시간을 갖는다. 이는 제안된 스케줄링 알고리즘이 새로 투입되는 로트의 마지막 공정이 끝나는 시간을 최소로 하도록 설계되었기 때문이다. 즉, 제안된 스케줄링 알고리즘은 사이클 시간을 줄이거나 단위시간당 생

산량을 높이는데 보다 더 효과적임을 알 수 있다.

## 6. 결론

본 논문은 경성 공정간 시간제약이 있는 웨이퍼 팹의 스케줄링 방법을 다루었다. 먼저 분산 스케줄링 메커니즘을 위한 다중 에이전트 기반의 아키텍처를 제안하였으며, 주어진 시간제약을 만족하는 스케줄을 생성할 수 있는 스케줄링 알고리즘을 제시하였다. EDF와 같은 할당규칙 기반의 접근 방법은 주어진 시간제약을 만족시키기 위하여 우선 순위를 정하지만, 시간제약을 만족시킨다는 보장은 할 수 없다. 이는 도미노 효과로 인하여 로트의 흐름을 불안정하게 만드는 요인이 된다. 따라서 본 논문에서는 동적 계획 접근 방법을 이용하였다. 또, 웨이퍼 팹에서 자원 활용의 유연성을 높이고 실시간 적용성을 높이기 위하여 에이전트 사이의 입찰 메커니즘을 이용하였다. 이를 위하여, 시간제약 워크셀 그룹이라는 개념을 도입하였으며, 다항식 함수의 계산 복잡도를 갖는 입찰 메커니즘 기반의 실시간 스케줄링 알고리즘을 제안하였다. 제안된 알고리즘이 시간제약을 만족하는 스케줄을 생성하는데 계산 속도면에서 충분히 효율적이고, 시스템의 단위시간당 생산량을 줄이는데 효과적임을 시뮬레이션 결과로부터 알 수 있었다.

특히, 본 논문은 시간제약 워크셀 그룹에서 시간제약을 만족하는 스케줄을 구하는데 초점을 맞추고 있다. 시간제약 워크셀 그룹에 포함되지 않은 워크셀들은 기존의 할당규칙과 같은 다양한 스케줄링 기법으로 제어될 수 있다. 예를 들면, Kim et al.<sup>(2)</sup>과 Kim et al.<sup>(3)</sup>에 의해 제안된 방법은 주문의 납기 기한을 맞추는데 효율적이며, Yoon과 Lee<sup>(22)</sup> 및 Lu et al.<sup>(23)</sup>에 의해 제안된 방법은 흐름 시간의 표준편차를 줄이는데 효과적으로 이용될 수 있다. 이와 같이, 제안된 스케줄링 방법은 기존의 할당규칙 기반의 스케줄링 시스템에 손쉬운 적용이 가능하다는 장점이 있다. 또한, 새로운 공정이 투입될 수 있는 시간 간격을 의미하는 시간제약 집합 CM을 변경함으로써, 재구성 생산시스템이나 기계 유지보수 스케줄에도 적용이 가능하다.

## 후기

이 논문은 한국과학재단의 해외 Post-doc. 연구 지원에 의하여 연구되었음



## 참고문헌

- (1) Lu, S. H. and Kumar, P. R., 1991, "Distributed Scheduling Based on Due Dates and Buffer Priorities," *IEEE Transactions on Automatic Control*, Vol. 36, No. 12, pp. 1406~1416.
- (2) Kim, Y. D., Kim, J. U., Lim, S. K. and Jun, H. B., 1998, "Due-Date Based Scheduling and Control Policies in a Multiproduct Semiconductor Wafer Fabrication Facility," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 11, No. 1, pp. 155~164.
- (3) Kim, Y. D., Kim, J. G., Choi, B. and Kim, H. U., 2001, "Production Scheduling in a Semiconductor Wafer Fabrication Facility Producing Multiple Product Types with Distinct Due Dates," *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 5, pp. 589~598.
- (4) Mason, S. J., Jin, S. and Wessels, C. M., 2004, "Rescheduling Strategies for Minimizing Total Weighted Tardiness in Complex Job Shop," *International Journal of Production Research*, Vol. 42, No. 3, pp. 613~628.
- (5) Watts, H., 2000, "Improving Fab Performance," *Future Fab International*, Vol. 9.
- (6) Robinson, J. K., 1998, "Capacity Planning in a Semiconductor Wafer Fabrication Facility with Time Constraints between Process Steps," Ph. D. Dissertation, Department of Mechanical and Industrial Engineering, University of Massachusetts, Amherst.
- (7) Nissanke, N., 1997, *Realtime System*, Prentice Hall, London.
- (8) Ramamritham, K. and Stankovic, J.A., 1994, "Scheduling Algorithm and Operating Systems Support for Real-Time Systems," *Proceedings of the IEEE*, Vol. 82, No. 1, pp. 55~67.
- (9) Hall, N. G. and Sriskandarajah, C., 1996, "A Survey of Machine Scheduling Problems with Blocking and No-Wait in Process," *Operations Research*, Vol. 44, No. 3, pp. 510~525.
- (10) Chen, H., Chu, C. and Proth, J. M., 1998, "Cyclic Scheduling of a Hoist with Time Window Constraints," *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 1, pp. 144~152.
- (11) Stankovic, J. A., Spuri, M., Ramamritham, K. and Buttazo, G. C., 1998, *Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms*, Kluwer Academic Publishers, Norwell, MA.
- (12) Mönch, L., Stehli, M. and Zimmerman, J., 2003, "FABMAS: An Agent-Based System for Production Control of Semiconductor Manufacturing Processes," in *Proceedings of the First International Conference on Industrial Application of Holonic and Multi-Agent Systems*, Prague, Czech Republic.
- (13) Mönch, L. and Stehli, M., 2003, "An Ontology for Production Control of Semiconductor Manufacturing Processes," in *Proceedings of the First German Conference on Multiagent System Technologies*, Erfurt, Germany.
- (14) Wyns, J., 1999, "Reference Architecture for Holonic Manufacturing Systems; The Key to Support Evolution and Reconfiguration," Ph. D. Dissertation, Department of Mechanical Engineering, Katholieke Universiteit Leuven, Belgium.
- (15) Yu, C. Y. and Huang, H. P., 2001, "Development of the Order Fulfillment Process in the Foundry Fab by Applying Distributed Multi-Agents on a Generic Message-Passing Platform," *IEEE/ASME Transactions on Mechatronics*, Vol. 6, No. 4, pp. 387~398.
- (16) Cheng, F. T., Chang, C. F. and Wu, S. L., 2004, "Development of Holonic Manufacturing Execution Systems," *Journal of Intelligent Manufacturing*, Vol. 15, pp. 253~267.
- (17) Shen, W., 2002, "Distributed Manufacturing Scheduling Using Intelligent Agents," *IEEE Intelligent Systems*, Vol. 17, No. 1, pp. 88~94.
- (18) Yoon, H. J., 2004, "Real-Time Scheduling of Semiconductor Integrated Single-Wafer Processing Tools," Ph. D. Dissertation, Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea.
- (19) Dechter, R., Meiri, I. and Pearl, J., 1991, "Temporal Constraint Network," *Artificial Intelligence*, Vol. 49, pp. 61~95.
- (20) Schwalb, E. and Dechter, R., 1997, "Processing Disjunction in Temporal Constraint Networks," *Artificial Intelligence*, Vol. 93, pp. 29~61.
- (21) Kemp, K., Intel Five-Machine Six Step Mini-Fab Description, *Intel/ASU Report* [on-line].
- (22) Yoon, H. J. and Lee, D. Y., 2000, "A Control Method to Reduce the Standard Deviation of Flow Time in Wafer Fabrication," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 13, No. 3, pp. 389~392.
- (23) Lu, S. H., Ramaswamy, D. and Kumar, P. R., 1994, "Efficient Scheduling Policies to Reduce Mean and Variance of Cycle-Time in Semiconductor Manufacturing Plants," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 7, No. 3, pp. 374~388.