

반도체 생산 배취공정에서의 배취 크기의 결정

Batch Sizing Heuristic for Batch Processing Workstations in Semiconductor Manufacturing

천길웅*, 홍유신*

Kil-Woong Chun*, Yushin Hong*

Abstract

Semiconductor manufacturing line includes several batch processes which are to be controlled effectively to enhance the productivity of the line. The key problem in batch processes is a dynamic batch sizing problem which determines number of lots processed simultaneously in a single batch. The batch sizing problem in semiconductor manufacturing has to consider delay of lots, setup cost of the process, machine utilization and so on. However, an optimal solution cannot be attainable due to dynamic arrival pattern of lots, and difficulties in forecasting future arrival times of lots at the process. This paper proposes an efficient batch sizing heuristic, which considers delay cost, setup cost, and effect of the forecast errors in determining the batch size dynamically. Extensive numerical experiments through simulation are carried out to investigate the effectiveness of the proposed heuristic in four key performance criteria: average delay, variance of delay, average lot size and total cost. The results show that the proposed heuristic works effectively and efficiently.

1. 서론

21세기를 이끌어 갈 대표적인 산업으로 알려져 있는 반도체 산업은 "산업의 쌀"이라고 불릴 정도로 중요한 위치를 차지하고 있는

산업으로서 국제사회에서 가장 경쟁이 치열한 대표적 산업이라 할 수 있다. 따라서 국내는 물론 해외의 모든 반도체 생산업체들은 국제경쟁력의 향상을 위하여 엄청난 노력이 경주되고 있다. 그러나 국내의 반도체 생산

* 포항공과대학교 산업공학과

에 있어서 생산관리 및 생산일정계획의 효율화를 통한 생산성 향상을 위한 노력은 아직까지는 미미한 실정이다.

반도체 생산공정은 수십개의 공정을 반복적으로 거치는 매우 복잡한 Flow를 흐르게 되며, 하나의 Lot이 투입되어 완성되기까지는 일반적으로 두달여의 기간이 소요된다. 한 개의 제품이 공정에 투입되어 완제품으로서 생산될 때까지의 기간을 생산주기(Cycle Time)라 할때 모든 생산공정에서와 마찬가지로 반도체 산업에서도 이러한 생산주기는 고객을 만족시키고 생산성을 향상하기 위한 가장 중요한 관리대상이다. 특히 반도체 생산의 경우에는 생산주기가 길어지게 되면 생산성이 저하되고 시장의 변화에 민첩하게 대처할수 있는 능력을 떨어뜨리며, 약속된 납기일(delivery date)를 어기는 결과를 초래할 뿐만 아니라 반도체 생산의 가장 중요한 성능지표인 생산수율(yield)에 치명적인 영향을 주는 문제점을 안고 있다.

일반적으로 반도체 생산공정에서는 실제 가공시간보다는 가공을 위한 준비 및 대기시간이 훨씬 큰 비율을 차지하고 있다. 이러한 과도한 준비시간 및 대기시간을 줄이기 위하여는 주요한 애로공정들이 효율적으로 관리되어야 한다. 반도체 공정에서의 대표적인 애로공정으로는 Reentrant Flow를 갖는 Wafer Fabrication 공정의 Photo와 배취공정(Batch Process)인 Ion Implantation, Test의 Burn-in Oven등을 들수 있다. 특히 배취공정에서는 Dynamic하게 도착하는 여러개의 Lot들을 한 개의 배취로 묶어 가공함에 따라 Lot들의 대기시간이 길어지게 된다.

현실적으로 반도체의 생산공정은 고도의

기능을 가진 설비들로 구성되어 있음에 따라 공정중의 Lot의 위치, Lot의 가공시간, 대기 중인 Lot의 수, 설비의 고장 및 수리시간 등 공정과 관련된 거의 모든 데이터가 On-Line으로 수집되는 기능을 가지고 있다. 그러나, 지금까지는 효율적인 일정계획 및 공정관리를 위한 기법들이 개발되어 있지 못함에 따라 수집되는 데이터들을 생산관리 및 일정계획등을 위하여 효과적으로 이용하지 못하고 있다.

본 연구에서는 반도체 생산공정의 배취공정에서의 최적 배취의 크기(Batch Size)를 결정하기 위한 기존의 기법들의 조사 및 분석을 통하여 보다 효율적이고 현실성있는 새로운 최적 배취의 크기를 결정할수 있는 기법을 제시한다. 또한 기존의 기법과 함께 제시된 기법을 배취공정에 응용하여 여러가지 성능지표에 대한 결과를 비교분석하여 공정의 특성에 따른 적절한 기법을 선택할수 있게 함으로써 국내 반도체 산업의 생산성 향상에 기여하고자 한다.

배취공정에서의 최적 배취의 크기를 결정하는 문제는 오랫동안 많은 연구가 수행되어 왔으나 대부분의 연구에서는 잘 알려진 바와 같이 Makespan의 최소화, 납기의 만족, 혹은 평균 생산주기의 최소화 등과 같은 단일 성능지표에 따른 일반적인 해를 제시하는데 중점을 두어왔다. Ikura와 Gimple[10]은 뒤에 도착한 작업이 더 늦은 납기일을 가지며 가공시간이 동일할 때 모든 작업이 각각의 납기일까지 마칠수 있는 일정의 존재 여부를 결정하는 알고리즘을 제시하였다. Ahmadi *et al.*[1]는 배취설비에서 모든 작업이 동일한 가공시간을 갖는다는 가정하에 평균 생산주

기와 Makespan을 최소화하는 문제에서 Polynomial-Time 알고리즘과 함께 휴리스틱을 제시하였다. Lee *et al.*[11]는 단일 배취설비에서 가공시간이 상수이고 뒤에 도착한 작업이 더 늦은 납기일을 갖는 경우에 최대 지연시간(Maximum Tardiness)와 지연된 작업의 수를 최소화하는 알고리즘을 개발하였다. Chandru *et al.*[4,5]는 배취설비에서 Makespan을 최소화하는 문제를 다루면서 Branch and Bound 알고리즘과 휴리스틱을 제시하였다. 이들의 휴리스틱은 수치실험을 통하여 적절한 CPU시간 내에 좋은 해를 구할 수 있음을 보였다. 한편, Uzsoy[15]는 단일 배취설비에서 작업들이 각각 다른 가중치를 갖는 경우에 Total Completion Time을 최소화하는 문제를 다루었다. 그러나 위의 논문들은 모든 작업이 가공 가능한 상태에 있다는 가정하에서 특정한 한가지의 성능지표만을 최적화하는 기법들로서 여러 형태의 성능지표가 동시에 고려되어야 하는 반도체 생산공정의 배취공정에 적용하기에는 문제점을 안고 있다.

또한 배취공정을 모형화함에 있어 대기이론(Queuing Theory)을 적용한 연구들로서 Lot들의 도착은 포아송 분포를 따르고 가공시간이 지수분포를 가지는 배취공정을 다룬 논문을 볼 수 있다. Neuts[13]는 가공을 위한 최소 배취의 크기(Minimum Batch Size)를 정한 후 대기중인 Lot의 수가 최소 배취의 크기와 같거나 혹은 최소 배취의 크기보다 크게 되면 가공을 시작하는 MBS를 제안하였다. Medhi[12]는 같은 가정하에서 대기시간의 분포를 유도하였으며, Deb과 Serfozo[6]는 동적 계획법을 이용하여 평균 비용을 최소화하는 최적 최소 배취의 크기를 결정하는 방법을

제시하였으나 최적이지 아닌 최소 배취의 크기에 의하여 가공되는 경우와 평균 비용면에서의 차이에 대한 적절한 분석이 결여되어 있다. 이러한 논문들은 대기이론을 이용한 모형화를 위하여 포아송 도착과 지수분포를 갖는 가공 시간을 가정하였으나 일반적으로 반도체 생산공정의 경우 완벽한 Shop Floor Control 시스템을 갖추고 있어 공정내의 모든 Lot들의 위치를 추적할 수 있음을 고려할 때 Lot들의 도착시각이 예측 가능하고 동시에 배취의 가공시간이 일정함으로 위의 연구 결과를 반도체 생산공정의 모형화에 이용하는 것은 적합치 못하다고 판단된다.

반도체 생산공정의 배취공정의 특성을 고려하여 배취의 크기를 결정하는 연구로 먼저 Glassey와 Weng[8]을 들 수 있다. 이들은 정해진 배취의 가공시간을 계획기간으로 하여 계획기간내에 도착하는 모든 Lot들의 도착시각을 정확히 예측할 수 있다는 가정 아래 모든 Lot들의 평균 지연시간(Average Delay)을 최소화하는 Dynamic Batching Heuristic(DBH)를 제안하였다. 이들은 시뮬레이션을 통하여 DBH가 안정적이고, 중간범위의 Traffic Intensity에서 MBS보다 우월함과 동시에 도착시각의 분산이 커질 때 더 효율적이라 주장하고 있다. 그러나 DBH의 경우 가공이 시작되어 끝날 때까지 도착하는 Lot들의 대기시간을 고려하지 않는 문제점을 가지고 있다. 또한 Fowler *et al.*[7]는 단일품목 및 다품목일 경우에 대하여 배취의 크기를 결정하기 위한 Next Arrival Control Heuristic(NACH)을 제안하였다. NACH는 DBH의 변형으로서 매 Lot의 도착시마다 다음 Lot만의 도착시각을 예측할 수 있다는 가정하에 가공을 시작할지

혹은 다음 Lot의 도착을 기다릴지 여부를 결정하게 된다. NACH는 Lot의 도착시점을 정확히 예측치 못할 경우에 예측오차의 영향을 줄일수 있다는 장점을 갖고 있는 반면에, 단지 다음에 도착할 Lot만을 고려하기 때문에 근시안적인 결정이 이루어지는 위험을 안고 있다. 즉 몇 개의 Lot을 더 기다림으로써 지연시간을 감소시킬수 있는 상황에서도 단지 현시점이 다음 Lot의 도착시점보다는 지연시간을 감소시킨다는 이유로 현시점에서 가공을 시작함으로써 전체적인 지연시간을 증가시키는 경우가 발생하게 된다. Weng과 Leachman[16]이 제시한 Minimum Cost Rate (MCR)은 잘 알려진 Silver-Meal Heuristic의 개념을 도입하여 평균 지연시간을 최소화하는 배취의 크기를 결정하는 기법으로 위의 두 기법들과 비교하여 볼때 보다 효율적이라 할수 있다. MCR의 경우 Lot들의 도착시각을 정확히 예측할수 있다고 가정함에 따라 DBH, NACH에 비하여 상대적으로 먼 시점까지 많은 Lot들의 도착시각까지 예측되어야 한다. 그러나 Lot들의 도착시각을 정확히 예측할수 없는 경우 예측기간이 길어짐에 따라 예측능력이 떨어지게 되고 예측 오차가 커지게 된다. 즉 예측치에 오차가 발생할때 현시점에서 Lot들의 도착시각이 멀어질수록 오차의 영향이 커짐에 따라 오히려 예측기간이 짧은 NACH보다 좋지 않은 결과를 보일 수 있다.

위의 세 기법은 매우 단순하고 실용적이라는 장점을 가지고 있으나, 예측치의 오차의 영향에 대한 분석 및 최적 배취의 크기를 결정하는데 가장 중요한 요소인 준비비용(Setup Cost)에 대한 검토가 전혀 고려되지 않았다는 문제점을 갖고 있다. 본 논문에서는 예

측치의 오차에 대한 영향을 분석하여 오차에 대한 보정방법을 제시하며, Lot들의 가공 지연에 따른 지연비용과 준비비용을 동시에 고려한 전체적인 비용의 최소화를 도모하는 배취의 크기를 결정하는 기법을 제시하고자 한다. 2장에서는 배취공정의 모형을 설명하고 3장에서는 최적 배취의 크기를 결정하는 알고리즘을 제안하였다. 4장에서는 수치실험을 통하여 예측 오차 및 지연비용과 준비비용에 따른 여러 성능지표들의 비교 분석을 통하여 배취공정의 특성에 따른 효율적인 기법을 제시하고 5장에서 결론 및 추후 연구 방향에 대하여 언급하고자 한다.

2. 배취공정의 모형

본 논문의 대상인 배취공정은 단일설비로 구성되어 있으며 고장이 발생하지 않는다. 가공대상 Lot들은 단일 종류의 Lot들로 모든 Lot들의 배취공정에 도착 시각은 예측할수 있으며 가공을 위하여 Lot들이 대기하게 되는 설비의 Queue의 용량은 무한하다. 배취의 용량(Batch Capacity)은 제한되어 있고 가공시간은 배취의 크기와 관계없이 일정하되 매 가공시마다 일정한 준비비용이 발생한다.

본 모형의 설명을 위하여 공정과 관련된 도수, 관련 비용과 필요한 변수들은 아래와 같이 정의된다.

C : 설비의 배취 용량

P : 배취의 가공시간

C_D : 단위 Lot당 단위 지연시간당 비용

C_S : 매 가공시의 준비비용

t_0 : 대기중인 Lot의 수가 하나 이상이면서 설비의 휴지가 시작되는 시각

- q : t_0 에 Queue에 대기중인 Lot의 수
 T_i : t_0 부터 i 번째 도착하는 Lot의 실제 도착시간
 t_i : t_0 부터 i 번째 도착하는 Lot의 예측 도착시간 ($t_i = E[T_i]$)

배취공정에 도착하는 Lot의 수의 시간에 따른 변화를 그림으로 표시하면 그림 1과 같다. t_0 에 q 개의 Lot이 Queue에 대기중이며 시간이 경과함에 따라 지속적으로 Lot들이 도착하게 되고 가공의 시작이 늦어질수록 대기중인 Lot들의 지연시간은 길어지게 되는 반면에, 한번 가공시의 배취의 크기가 커짐에 따라 준비비용을 줄이는 효과를 얻을 수 있다. 즉, 준비비용이 미미한 경우에는 준비비용을 고려하지 않은 기존의 기법들을 이용할 수 있지만 준비비용이 커짐에 따라 이러한 준비비용을 고려한 최적 배취의 크기를 결정하는 기법이 필요하게 된다. 또한 모든 Lot들의 도

착 시간을 정확하게 예측할 수 있고 계획기간(Planning Horizon)이 한정된 경우에는 동적 계획법을 이용하여 총 비용을 최소화하는 가공 시작 시점들을 결정할 수 있으나 장기간의 계획기간내에 도착하는 모든 Lot들의 도착 시간을 정확하게 예측할 수도 없을 뿐만 아니라 문제가 지나치게 커짐에 따라 엄청난 계산시간이 소요되게 된다.

앞에서 설명된 기존의 기법들은 배취공정에 도착하는 Lot들의 도착 시간을 정확히 예측할 수 있다고 가정하고 있다. 그러나 반도체 생산공정과 같은 복잡한 공정에서 배취공정에 도착되는 Lot의 도착 시간들을 정확히 예측한다는 것은 현실적이지 못한 가정이라 할 수 있다. 실제 현장에서는 여러가지 불확실한 상황을 줄이기 위하여 설비 점검등을 하고 있지만 설비 자체가 정교하고 고도의 정밀도를 요구하므로 설비의 고장 및 수리 등

대기중인 Lot의 수

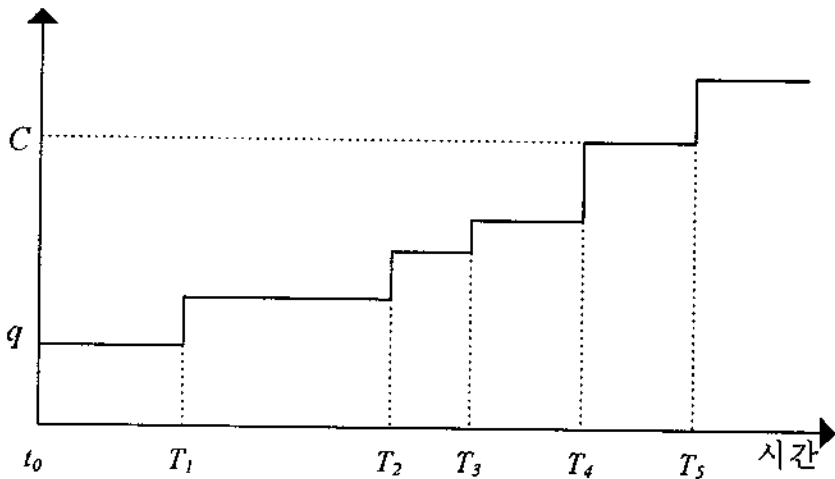


그림 1. 시간에 따른 대기 Lot의 수의 변화

으로 인한 불확실성이 항시 존재하며 Reentrant Flow, Rework 등의 복잡한 공정흐름 때문에 Lot의 도착 시각을 예측하는데 있어서 오차는 피할수 없게 된다. 본 연구에서는 예측 오차는 예측기간(예측시점으로부터 예측 도착 시각까지의 경과시간)이 길어질수록, 즉 예측 시점에서 예측된 도착 시각이 멀어질수록 예측 오차가 커진다고 가정하였다.

위의 변수의 정의에서 T_i 는 아래의 식 (1)로 표시된다.

$$T_i = t_i + \varepsilon_i, \quad \varepsilon_i \sim Normal\{0, k(t_i - t_0)^2\} \quad (1)$$

식 (1)에서 예측오차 ε_i 는 평균이 0이고 분산이 예측기간의 제곱에 비례하며 서로 독립인 정규분포를 갖는다고 가정하였으며 k 는 분산의 정도를 나타내는 상수이다. Glassey와 Weng[8]은 예측 오차가 Lot들의 도착 간격의 분산에 비례한다고 가정하였으나 본 논문에서와 같이 예측기간의 제곱에 비례한다는 가정이 보다 타당성이 있다고 인정된다. 이와 같은 가정하에서는 예측된 Lot의 도착 시각이 t_0 에서 멀어짐에 따라 예측 오차가 커지게 되고 특히 예측기간이 가장 긴 MCR의 경우 예측치의 오차가 커짐에 따라 부정적인 영향을 미칠수 있는 반면, 다음 Lot만의 도착 시각만을 이용하는 NACH는 예측 오차의 영향을 적게 받는 장점을 가질수 있다.

3. 배취의 크기 결정 알고리즘

앞 절에서 설명된 배취공정에 있어서 지연비용과 준비비용을 포함한 총 비용의 단위시간당 평균비용은 아래와 같이 구할수 있다.

즉 t_0 에서부터 i 번째 도착하는 Lot까지를 하나의 Batch로 묶어 가공을 시작할때 한 생산 주기의 단위시간당 평균비용($Cost_i$)은 아래의 식 (2)로 주어진다.

$$Cost_i = \frac{C_s + C_d(q(t_i - t_0)) + \sum_{i | t_i < t_0} (t_i - t_0) + \sum_{i | t_i > t_0 + P} (t_i + P - t_0)}{t_i + P - t_0} \quad (2)$$

식 (2)에서 분모는 하나의 배취가 가공되는 주기를 나타내며, 분자의 첫번째 항은 준비비용을 의미하고 두번째 항은 가공중인 Lot들의 지연시간과 가공중에 설비에 도착되는 Lot들이 가공이 끝날때까지의 지연시간에 따른 지연비용을 나타낸다.

먼저 t_0 에 $q \geq C$ 이면 바로 가공을 시작하게 된다. 그러나 $q < C$ 일때는 $i=0$ 에서 $i=(C-q)$ 까지의 모든 i 에 대하여 Lot들의 도착 시각을 예측하여 식 (2)에 주어진 평균비용을 구한 후 이중 최소의 평균비용을 가지는 i^* 를 결정, $(q+i^*)$ 를 하나의 배취로 하여 가공을 시작한다. 만약 Lot들의 도착 시각이 정확하게 예측될수 있다면 위의 기법은 적절한 기법이라 할수 있으나 예측된 도착 시각에 오차가 발생되는 경우에는 이에 대한 보정이 필요하게 된다. 즉 t_0 에서의 평균비용을 비교한 결과 $i^*=0$ 이면 바로 q 를 하나의 배취로 하여 가공을 시작하지만 $i^*>0$ 인 경우에는 다음 Lot의 도착 시각인 T_1 까지 가공을 미루게 된다. 시간이 경과하여 하나의 Lot이 도착하는 시점인 T_1 이 되면 T_1 을 새로운 t_0 로 재 정의한후 앞으로 도착되는 Lot들의 도착 시각을 다시 예측하여 새로운 평균비용을 계산, 비교를 함으로써 새로운 i^* 를 결정하게 된다.

이와 같이 Lot들의 도착시간을 다시 예측할 때 예측기간이 줄어들어 따라 예측 오차를 줄이는 효과를 기대할 수 있다. 물론 계산량이 증가되는 문제점은 있으나 식 (2)의 평균 비용을 계산하는데 소요되는 시간은 무시할 수준임을 고려할 때 계산시간에는 별 문제점은 없게 된다(그림 2 참조).

공을 시작하고 알고리즘을 종료함

단계 2 : 모든 $i(0 \leq i \leq C-q)$ 에 대하여 Lot들의 도착 시간을 예측하고 식 (2)를 이용하여 계산된 $Cost_i$ 를 최소화하는 i^* 를 결정함.

단계 3 : 만약 $i^* > 0$ 이면

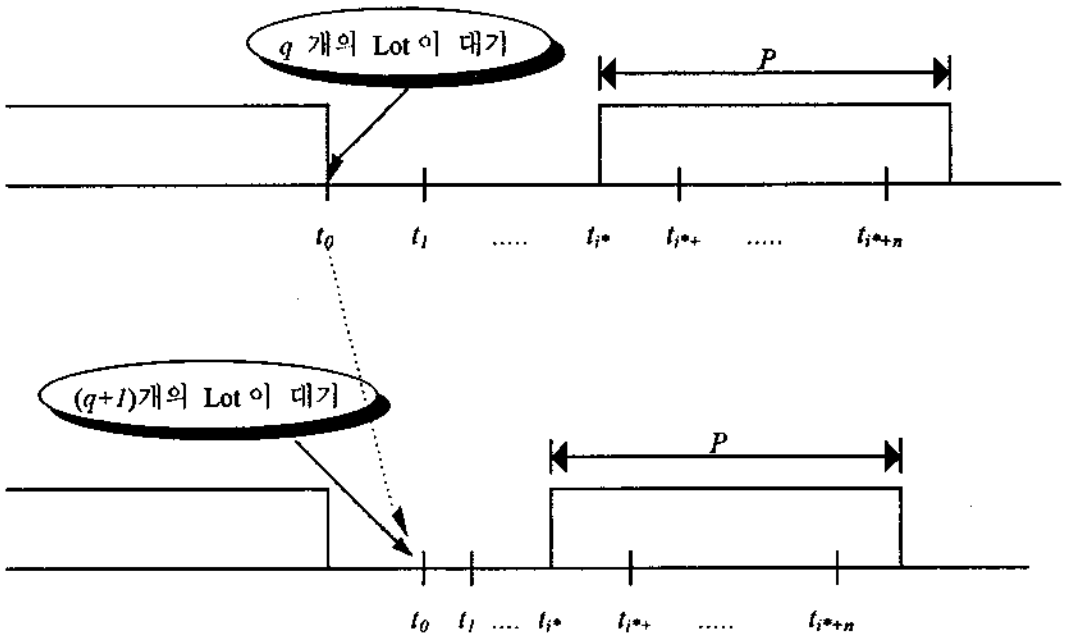


그림 2. 가공 시작 시점의 보정

위에서 제안된 알고리즘을 MMCR (Modified MCR)이라 정의하였으며 MMCR 알고리즘은 설비의 가공이 종료될 때 마다 만약 $q > 0$ 이면 바로 가동되고, $q = 0$ 이면 다음 Lot이 도착되는 시점에서 가동을 시작한다.

MMCR 알고리즘

단계 1 : 만약 $q \geq C$ 이면

C 개의 Lot을 한 배취로 하여 가

다음 Lot이 도착할때까지 기다리다가 다음 Lot이 도착하면

$t_0 = T_i$, $q = q + 1$ 로 놓고 단계 1로 감

아니면

q 개의 Lot을 한 배취로 하여 가공을 시작하고 알고리즘을 종료함

위의 알고리즘을 설명하면 다음과 같다. 먼저 단계 1에서는 설비의 가공이 끝났을 때 대기중인 Lot의 수가 배취의 용량보다 크거나 같을 때 ($q \geq C$)에는 바로 새로운 배취의 가공을 시작한다. 그러나 대기중인 Lot의 수가 배취의 용량보다 적을 경우 ($q < C$)에는 단계 2로 가 앞으로 도착될 $(C-q)$ 개의 Lot들의 도착시간을 예측하고 이에 따른 단위시간당 평균 비용을 계산하여 가장 적은 평균비용을 발생시키는 Lot(i^* 번째 도착하는 Lot)의 도착시점을 새로운 배취의 가공시작시점으로 결정하고 단계 3으로 간다. 단계 3에서는 단계 2에서 결정된 $i^* > 0$ 일 때는 다음 Lot이 도착하면 대기중인 Lot의 수(q) 및 기준시점(t_0)를 Update하고 단계 1로 가며, $i^* = 0$ 이면, 즉 바로 가공을 시작하는 것이 최적이면 바로 가공을 시작하게 된다.

4. 수치 실험

기존의 대표적인 알고리즘인 DBH, NACH, MCR과 함께 본 논문에서 제안한 MMCR 알고리즘의 실행에 따른 주요 성능지표들에 대한 비교 및 평가를 위하여 시뮬레이션을 이용한 수치실험을 수행하였다. 시뮬레이션 프로그램은 C언어로 작성되어 아래와 같이 정의되는 TI (Traffic Intensity)에 따라 각 알고리즘에 대하여 각각 100,000 Time Unit의 시뮬레이션을 실시하였으며 비교 대상 성능지표로는 평균 지연시간, 지연시간의 분산, 평균 배취의 크기, 총비용을 선택하여 그 결과를 제시하였다. 일반적으로 지연시간의 분산에 대하여는 많은 연구에서 언급되고 있지 않으나 반도체 생산공정의 생산성 평가에 가

장중요한 지표인 수율에 치명적인 영향을 주는 지표임을 고려할때 반드시 비교 평가되어야 할 중요한 성능지표이다. 위의 성능지표 이외에도 설비이용율(Utilization)등도 관심대상이지만 이는 평균 배취의 크기 및 평균 지연시간으로부터 쉽게 계산될수 있으므로 표에는 생략하였다.

$$TI = \frac{\text{Arrival Rate}(\lambda)}{\text{Service Rate} \times \text{Batch Capacity}}$$

본 수치실험에서 가공시간(P)은 40 Time Unit로, 배취의 용량(C)은 20 Lot으로 정하였으며, Lot들의 도착시간의 간격은 TI 가 변함에 따라 식 (3)으로부터 도착율(Arrival Rate, λ)을 결정한후 $[0, 2/\lambda]$ 의 Uniform 분포에 의하여 생성되었다. 예측 오차가 성능지표들에 미치는 영향을 비교, 분석하기 위하여는 Lot들의 도착 시각을 정확히 예측할수 있는 경우($k=0$)와 예측 도착 시각이 오차를 포함하는 ($k=0.4$), 즉 식 (1)의 ϵ_i 가 평균이 0이고 분산이 $0.4(t_i - t_0)^2$ 인 정규분포를 가질 경우로 구분하였으며 지연비용과 준비비용의 영향을 비교를 위하여는 하나의 Lot의 단위시간당 지연비용(C_D)는 1로 정하고, 3수준의 준비비용($C_S = 50, 150, 300$)에 대하여 수치실험을 수행하였다. 한편 첨부된 표들에서 본 논문에서 제안된 MMCR을 준비비용이 50인 경우에는 MMCR1로, 150인 경우에는 MMCR2로, 300인 경우에는 MMCR3로 표시하였다.

수치실험을 통하여 얻어진 각각의 알고리즘들의 성능지표를 보여주는 실험결과치들에 대한 비교 분석을 위하여 평균 지연시간 및 평균 배취의 크기에 대하여는 유의수준 5%의 t -test를, 지연시간의 분산에 대하여는 유

의수준 5%의 *F-test*를 통한 통계적 검정을 실시하였다. 첨부된 표들에서 각각의 성능지표에 대하여 가장 좋은 성능을 보이는 결과치는 Bold체로 나타내었으며 가장 좋은 성능을 보이는 결과치와 비교하여 통계적으로 유의차가 없는 결과치에 대하여 *표시를 하였다.

• Lot들의 도착 시각을 정확히 예측할수 있는 경우($k=0$)

Lot들의 도착 시각을 정확히 예측할수 있는 경우의 수치실험을 통한 성능지표의 결과치는 [표 1]에서 [표 6]까지 주어져 있다. [표 1]에서 평균 지연시간의 결과를 보면 $TI=0.9$ 를 제외한 모든 TI 에서 MCR이 우월함을 나타내고 있다. 물론 $TI=0.9$ 인 경우에는 MMCR1이 조금 우월한 것으로 나타나 있으나 이 경우에는 모든 알고리즘의 결과치 사이에 통계적인 유의차가 없음을 볼때 큰 의미를 가진다고는 할수 없다. 평균 지연시간에 있어서 MCR이 상대적으로 좋은 결과를 주는 이유는 DBH나 NACH보다 상대적으로 많은 Lot들의 도착 시각을 정확히 예측할수 있고, 준비비용에 대한 고려가 없기 때문이다. 그러나 MCR은 TI 가 낮은 경우(0.1-0.3)에는 DBH나 NACH와 통계적으로 유의차가 없으며 TI 가 중간수준 이상인 경우(0.4이상)에는 MMCR1과 통계적인 유의차가 없음을 보여주고 있다. 한편 준비비용이 커짐에 따라 평균 지연시간은 증가하게 되고 특히 TI 가 낮은 경우의 평균 지연시간에 있어서 기존의 세 알고리즘과 MMCR사이에는 엄청난 차이를 보이고 있지만 TI 가 점차 증가함에 따라 그 차이는 급격히 줄어드는 현상을 볼수 있다.

표 1. 평균 지연시간($k=0$)

TI	DBH	NACH	MCR	MMCR1	MMCR2	MMCR3
0.1	12.530	12.406*	11.957	13.946	28.070	62.741
0.2	15.883*	15.826*	15.547	15.925	20.679	36.608
0.3	17.100*	17.126*	16.897	17.076*	19.192	27.832
0.4	17.866	17.786*	17.548	17.677*	18.653	23.636
0.5	18.262	18.315	18.081	18.120*	18.647	21.719
0.6	18.656	18.628	18.349	18.369*	18.779	20.489
0.7	18.959	18.806	18.613	18.620*	18.773*	19.832
0.8	19.250	19.058	18.892	18.915*	19.028*	19.461
0.9	20.172*	20.193*	20.122*	20.052	20.062*	20.211*

[표 2]에는 각각의 알고리즘에 따른 Lot들의 지연시간의 분산을 보여주고 있다. 일반적인 다른 제품과는 달리 반도체 생산에 있어서는 Lot들의 생산주기가 지나치게 길어지게 되면 생산수율에 치명적인 영향을 주기 때문에 평균 지연시간과 함께 지연시간의 분포도 매우 중요한 관리대상이 된다. [표 2]에서 보는 바와 같이 NACH가 모든 TI 수준에서 가장 우월함을 보여주고 있으나 MCR과의 통계적인 유의차는 거의 없는 것으로 나타나고 있다. 이는 모든 Lot들의 도착 시각을 정확히 예측할수 있다는 가정에 기인된 것으로 만약 Lot들의 도착 시각을 정확히 예측할수 없는 경우에는 다른 결과를 보여주고 있음을 뒤에서 설명되어 있다. 지연시간의 분산의 경우에도 평균 지연시간의 경우와 같이 준비비용이 커짐에 따라 분산이 증가하는 것을 볼수 있으며 이러한 각 알고리즘에 따른 지연시간의 분산의 차이는 TI 가 증가함에 따라 감소함을 볼수 있다.

[표 3]의 평균배위의 크기에 있어서 기존의 세 알고리즘에서의 결과치는 유사하나 준

표 2. 지연시간의 분산($k=0$)

TI	DBH	NACH	MCR	MMCR1	MMCR2	MMCR3
0.1	150.59*	146.66	149.62*	217.30	615.56	2236.59
0.2	161.57	153.81	156.50*	180.50	309.63	771.05
0.3	160.17	151.22	156.52	168.76	234.46	450.85
0.4	161.41	149.11	152.96*	160.96	196.58	322.23
0.5	158.29	147.10	149.61*	155.66	177.81	259.83
0.6	158.02	146.74	148.06*	152.15	169.06	215.18
0.7	157.27	144.91	145.95*	148.31	156.79	184.27
0.8	154.56	143.01	143.21*	144.55*	150.82	163.99
0.9	151.72	147.04	146.60*	146.42	148.44*	154.39

표 3. 평균 배치의 크기($k=0$)

TI	DBH	NACH	MCR	MMCR1	MMCR2	MMCR3
0.1	2.369	2.381	2.402	2.868	4.486	8.138
0.2	4.560	4.505	4.480	4.743	6.032	9.401
0.3	6.661	6.537	6.513	6.709	7.709	10.567
0.4	8.834	8.577	8.551	8.726	9.429	11.741
0.5	10.907	10.622	10.584	10.752	11.285	13.239
0.6	13.080	12.666	12.650	12.767	13.282	14.685
0.7	15.141	14.645	14.602	14.712	15.039	16.067
0.8	17.113	16.592	16.535	16.606	16.862	17.468
0.9	18.725	18.470	18.342	18.390	18.500	18.812

비비용을 고려하게 되면 준비비용이 커짐에 따라 낮은 TI수준에서는 평균 배치의 크기가 커지게 되며 TI가 커질수록 알고리즘간의 결과치에서의 차이는 감소하여 TI=0.9에서는 거의 차이가 없음을 볼수 있다. 평균 배치의 크기가 크다는 것은 설비의 가동시 설비이용율이 커지는 장점을 가지는 반면에 설비의 후지시간이 길어짐과 동시에 Lot들의 대기시간이 길어지게 됨으로 배치의 크기를 결정하는데 있어서 설비이용률과 Lot들의 대기시간이 동시에 고려되어야 한다.

표 4. 총비용($k=0$)

TI	DBH	NACH	MCR	MMCR1
0.1	170676	169598	166371	159264
0.2	270216	271040	268826	266431
0.3	369273	371814	368819	368117
0.4	470723	472510	468796	468377
0.5	572134	576557	571160	570283
0.6	676533	679503	671250	670761
0.7	779603	778182	771768	771138
0.8	887484	883481	877976	877672
0.9	1027356	1030010	1027689	1024185

[표 4, 5, 6]에 주어진 총비용을 비교하여 보면 모든 TI 및 3수준의 준비비용에 대하여 MMCR이 우수함을 보여주고 있다. 이는 MMCR이 기존의 알고리즘들의 장점을 택함과 동시에 기존 알고리즘들에서 고려되지 못한 준비비용을 고려한 점을 감안할때 당연한 결과라고 판단된다. 특히 낮은 TI에서는 그 차이가 크게 나타나고 있으며 TI가 점차 증가함에 따라 그 차이가 줄어들고 있음을 볼수 있다. 결론적으로 모든 Lot들의 도착 시각을 정확히 예측할수 있을 때에는 준비비용

의 수준이 바람직한 알고리즘을 선정하는데 중요한 역할을 함을 알수 있다. 즉 TI가 상당히 높은 수준인 경우, 수치실험의 결과에서 보는바와 같이 어느 알고리즘을 선정하더라도 성능지표에는 거의 차이가 없음을 볼수 있다. 그러나 TI가 중간이하의 수준일 때에는 준비비용의 영향이 커지게 되고 TI가 낮아짐에 따라 그 영향은 더욱 커지게 된다. 즉 TI가 중간 이하의 수준일때 준비비용이 작은 경우에는 기존의 MCR을 이용하더라도

표 5. 총비용($k=0$)

TI	DBH	NACH	MCR	MMCR2
0.1	384978	382816	377690	312182
0.2	490977	494407	493481	458449
0.3	594600	601360	599181	580044
0.4	697317	705852	702167	691602
0.5	801737	812310	807708	799834
0.6	906634	917075	909144	905087
0.7	1010976	1017321	1011628	1006808
0.8	1121478	1124815	1119428	1117821
0.9	1267585	1273493	1272871	1267032

표 6. 총비용($k=0$)

TI	DBH	NACH	MCR	NMCR3
0.1	706381	702684	694709	505500
0.2	822088	829524	830486	689631
0.3	932577	945706	944793	843736
0.4	1037162	1055843	1053239	984292
0.5	1146140	1165964	1162606	1111398
0.6	1251785	1273497	1266038	1231498
0.7	1357998	1376061	1371438	1348514
0.8	1472423	1486798	1482681	1466628
0.9	1627913	1638776	1640703	1626359

MMCR과 모든 성능지표에서 큰 차이를 보이지는 않고 있으나 준비비용이 커지게 되면 MMCR의 효과는 더욱 커지게 된다. 다만 준비비용이 커지게 되면 대기시간이 지나치게 길어지는 Lot들의 수가 증가하여 생산수율이 급격히 떨어지게 됨으로 이에 대한 적절한 대안이 고려되어야 한다.

• Lot들의 도착 시각을 정확히 예측할수 없는 경우($k=0.4$)

Lot들의 도착 시각을 정확히 예측할수 없는 경우, 즉 예측 도착 시각이 오차를 갖는 경우의 실험결과는 [표 7]에서 [표 10]까지 주어져 있는바 앞의 오차가 없을 경우의 실험 결과치들과 비교하여 보면 예측치의 오차의 영향으로 전반적인 성능지표들의 값이 떨어짐을 볼수 있다. [표 7]의 평균 지연시간에 있어서는 TI 수준이 낮을 때는 기존의 세 알고리즘간의 통계적인 유의차가 없으나 TI가 중간 수준이상이면 NACH가 가장 우월함을 볼수 있다. 이는 NACH는 바로 다음에 도착되는 Lot만의 도착 시각을 예측하여야 하나 DBH나 MCR은 상대적으로 많은 Lot들의 도착 시각을 예측하여야 함에 따라 예측 오차의 영향이 커지게 되어 나타나는 현상이라 볼수 있다. 따라서 Lot들의 도착 시각을 예측하는데 있어서 예측 오차가 예측기간의 영향을 받게 될때 많은 Lot들의 도착 시각들을 예측하여 배취의 크기를 결정하는 것이 평균 지연시간에 있어서 바람직하지만은 아님을 보여주고 있다. 한편 준비비용이

표 7. 평균 지연시간($k=0.4$)

TI	DBH	NACH	MCR	NMCR1	NMCR2	NMCR3
0.1	12.542*	12.479*	12.150	13.849	27.959	62.814
0.2	15.933*	15.929*	15.898	16.050*	20.591	36.459
0.3	17.423	17.132	17.163*	17.213*	19.182	27.625
0.4	18.017*	17.809	17.907*	17.876*	18.741	23.498
0.5	18.432	18.154	18.390	18.270*	18.839	21.643
0.6	18.807	18.576	18.805	18.665*	18.936	20.423
0.7	19.072	18.777	19.005	18.928*	19.028	19.995
0.8	19.405	19.087	19.363	19.330	19.342	19.664
0.9	20.265*	20.227	20.634	20.477	20.457	20.552

적을 경우인 MMCR1의 평균 지연시간도 가장 짧은 평균 지연시간을 보이는 NACH와 비

교하여 볼때 통계적으로 거의 유의차가 없는 것으로 나타났다. 이는 MMCR이 예측기간이 길어 예측 오차의 영향을 많이 받게 되어 있지만 새로운 Lot이 도착할 때마다 알고리즘을 재가동하여 가공 시작 시점을 보정하여 주기 때문이다. 반면에 준비비용이 커짐에 따라 평균 지연시간은 예측 오차가 없을 경우와 거의 유사한 결과를 보여주고 있다.

[표 8]에 주어진 지연시간의 분산에 대한 결과치를 보면 모든 TI수준에서 NACH가 지연시간의 분산이 가장 작은 것으로 나타났다. 앞의 오차가 없을 경우에는 MCR도 NACH와 비교하여 볼때 통계적으로 유의차가 없는 것으로 나타났으나 예측 오차가 있을 때의 결과치를 보면 MCR을 적용하였을 때의 지연시간의 분산이 더욱 커지는 것을 볼수 있는데 이는 평균 지연시간의 경우와 같이 예측기간이 늘어남에 따라 예측치의 오차가 커지기 때문이다. 또한 낮은 수준의 TI에서는 준비비용이 커짐에 따라 지연시간의 분산이 현저하게 커지게 되나 TI의 수준이 증가함에 따라 준비비용이 지연시간의 분산에 미치는 영향은 점차 감소하게 된다.

[표 9]의 평균 배취의 크기는 다른 성능지표와는 달리 예측 오차의 영향을 가장 적게 받는 것으로 나타났다. 즉 [표 3]에 주어진 예측 오차가 없을 때의 평균 배취의 크기와 [표 9]의 평균 배취의 크기를 비교하여 보면 모든 알고리즘에 있어서 실험 결과치의 차이가 거의 없음을 볼수 있다.

준비비용의 수준에 따른 총비용은 [표 10, 11, 12]에 주어져 있다. [표 10]의 준비비용이 적을 경우의 총비용을 보면 낮은 TI 수준에서는 MMCR1이 가장 경제적인 반면에 TI

표 8. 지연시간의 분산($k=0.4$)

TI	DBH	NACH	MCR	NMCR1	NMCR2	NMCR3
0.1	153.38*	150.03	153.75*	215.68	615.60	2217.78
0.2	163.41	155.12	163.80	184.95	308.84	768.49
0.3	167.56	152.02	161.28	171.30	235.84	448.46
0.4	164.27	150.45	159.17	165.62	197.77	321.05
0.5	162.44	146.47	155.80	159.22	180.19	258.72
0.6	161.94	146.40	155.72	158.31	170.57	213.65
0.7	159.09	143.85	151.48	152.74	160.60	188.79
0.8	156.18	143.34	148.69	149.33	154.76	168.02
0.9	152.66	147.23	150.06	148.96*	151.48	156.71

표 9. 평균 배취의 크기($k=0.4$)

TI	DBH	NACH	MCR	NMCR1	NMCR2	NMCR3
0.1	2.375	2.382	2.405	2.848	4.476	8.138
0.2	4.554	4.513	4.514	4.747	6.010	9.384
0.3	6.706	6.533	6.536	6.727	7.681	10.500
0.4	8.830	8.592	8.613	8.768	9.429	11.686
0.5	10.955	10.595	10.688	10.793	11.315	13.183
0.6	13.103	12.677	12.794	12.876	13.294	14.628
0.7	15.154	14.611	14.762	14.806	15.123	16.127
0.8	17.128	16.606	16.682	16.714	16.951	17.530
0.9	18.759	18.468	18.474	18.463	18.601	18.867

의 수준이 중간(0.5)이상인 경우에는 MMCR1과 큰차이는 없으나 NACH가 상대적으로 우월하며 TI=0.9인 경우에는 DBH가 가장 경제적인을 보여주고 있다. 이러한 결과는 준비비용이 적을 경우에는 예측오차가 준비비용보다 총 비용에 미치는 영향이 상대적으로 크기 때문에 NACH가 MMCR1보다 우월하나 준비비용이 커지게 됨에 따라 준비비용의 영향이 상대적으로 커지게 되어 MMCR이 NACH보다 경제적인을 볼수 있다([표 11, 12] 참조). 한편 TI=0.9인 경우에는 준비비용의 수준과 관계없이 DBH가 가장 경제적인

것으로 나타나 있는데 이는 수치실험의 결과의 처리과정에서의 통계적인 오차에 기인된 것으로 판단되나 앞으로 좀 더 연구되어야만 명확한 해석이 가능할 것으로 생각된다.

표 10. 총비용($k=0.4$)

TI	DBH	NACH	MCR	NMCR1
0.1	170436	169920	167201	159371
0.2	270869	271880	271514	267597
0.3	373372	371995	372408	369865
0.4	473788	472772	474443	471822
0.5	575904	572841	577751	573582
0.6	680875	677819	683613	678650
0.7	783457	777434	784189	781160
0.8	893578	884555	895062	893539
0.9	1031080	1031264	1049521	1042567

표 11. 총비용($k=0.4$)

TI	DBH	NACH	MCR	NMCR2
0.1	384141	383037	378217	311982
0.2	491932	494846	494471	458460
0.3	597197	601660	601870	580943
0.4	700484	705713	706807	693336
0.5	804507	809194	812019	803732
0.6	910576	915191	918821	909502
0.7	1014635	1017187	1021455	1013778
0.8	1127371	1125689	1135029	1128509
0.9	1270907	1274854	1293044	1282571

종합적으로 준비비용이 무시할 수준인 경우에는 평균 지연시간 및 지연시간의 분산에서 가장 좋은 결과를 보여주는 NACH가 가장 바람직한 알고리즘이나 준비비용이 커짐에 따라 NACH와 비교하여 볼때 MMCR의 상대적인 우월성이 커지게 된다. 물론 앞의 예측 오차가 없을 경우와 같이 TI가 커짐에 따라 모든 성능지표의 결과치들에서 알고리즘

표 12. 총비용($k=0.4$)

TI	DBH	NACH	MCR	NMCR3
0.1	704645	702753	694783	505867
0.2	823495	829362	828925	688725
0.3	932921	946226	946382	843326
0.4	1040479	1055104	1055318	983931
0.5	1147409	1163747	1163450	1111888
0.6	1255128	1271313	1271690	1231914
0.7	1361362	1378790	1377360	1351819
0.8	1478013	1487373	1495128	1472304
0.9	1630585	1640193	1658267	1639363

간의 차이는 점차 줄어들게 되어 아주 높은 수준의 TI에서는 알고리즘간의 차이가 거의 없음을 볼때 어떤 알고리즘을 이용하더라도 유사한 결과를 얻을수 있음을 알수 있다.

5. 결론 및 추후 연구방향

본 논문에서는 반도체 생산공정의 배취공정에서 가공을 위한 배취의 크기를 결정하는 알고리즘에 대한 연구를 수행하였다. 먼저 기존의 대표적인 알고리즘인 DBH, NACH 및 MCR의 장점 및 문제점등을 분석하여 기존의 알고리즘을 개선한 새로운 알고리즘을 제안하였다. 기존의 세 알고리즘중에는 MCR이 전반적으로 가장 우수한 것으로 나타나 있으나 이러한 결과는 모든 Lot들의 도착 시각을 정확히 예측될수 있다는 가정이 선행되어야 한다. 그러나 반도체 생산공정과 같이 복잡한 Lot들의 흐름을 갖는 공정에서 배취공정에 도착하는 Lot들의 도착 시각을 정확히 예측한다는 것은 현실적으로 매우 어려운 과제이다. 만약 Lot들의 도착 시각을 정확히 예측하지 못하고 예측 오차가 발생하는 경우에

는 NACH가 MCR보다 우월함을 보여주고 있다. 또한 기존의 세 알고리즘의 경우에는 단지 Lot들의 평균 지연시간만을 비교 대상 성능지표로 선정하였으나 실제 반도체 생산공정에서는 평균 지연시간과 함께 지연시간의 분산 및 설비용률등도 매우 중요한 성능지표임으로 이에 대한 각각의 알고리즘의 결과치에 대한 비교 평가가 되어야 하며 또한 배취의 크기를 결정하는데 있어서 준비비용의 영향을 고려하지 않을수 없다. 이와 같은 여러가지의 문제점들을 고려하여 본 논문에서는 기존의 MCR과 NACH의 장점을 취합하여 새로운 배취의 크기를 결정하는 알고리즘인 MMCR을 제안하였다.

기존의 알고리즘과 함께 새로 제안된 MMCR의 성능 평가를 위하여 시뮬레이션을 이용한 수치실험을 수행하였다. 수치실험 결과를 비교하기 위한 성능지표로는 평균 지연시간, 지연시간의 분산, 평균 배취의 크기 및 지연비용 및 준비비용을 고려한 총 비용등을 선정하였다. 먼저 Lot들의 도착 시각을 예측하는데 있어서 정확한 예측이 가능한 경우와 예측기간의 영향을 받는 예측 오차가 있을 경우와 함께 준비비용을 세수준으로 나누어 준비비용과 지연비용과의 상대적인 영향을 비교 분석하였다. 먼저 예측 오차가 없고 준비비용이 상대적으로 적을 경우에는 MCR이 우수하다고 나타났으나 준비비용이 커지게 됨에 따라 MMCR의 상대적인 우월성이 증가하게 된다. 그러나 MMCR의 경우 낮은 수준의 TI 에서는 준비비용의 총 비용에 미치는 영향이 매우 크게 됨에 따라 평균 지연시간 및 지연시간의 분산이 지나치게 증가하는 것을 볼수 있어 이에 대한 적절한 Tradeoff가

고려되어야 할것이다. 한편 Lot들의 도착 시각을 정확히 예측하기 어려운 경우에는 평균 지연시간 및 지연시간의 분산만을 고려한다면 NACH가 좋은 결과를 보이고 있으나 준비비용의 상대적인 영향이 커짐에 따라 MMCR의 장점이 더욱 커지게 된다.

본 논문에서는 단일 종류의 Lot만을 가공하는 경우의 배취의 크기를 결정하는 알고리즘을 제안하였으나 여러 종류의 Lot들을 동시에 가공하는 배취공정으로 문제가 확장되면 지연비용 및 준비비용과 함께 종류별 Lot들의 수요에 따라 가공 순서를 결정하는 연구가 수행되어야 한다. 또한 반도체 생산공정은 배취공정뿐만 아니라 하나씩 Lot을 가공하는 공정(Single Lot Processing)등 여러 형태의 공정들로 구성되어 있음에 따라 전체 생산공정의 생산성을 고려한 일정계획을 수립할수 있도록 되어야 함과 동시에 설비의 고장 및 수리, Rework등 공정에서 발생하는 여러 형태의 불확실 요인에 효과적으로 대처할수 있는 기법등을 포함하는 실용적인 기법의 개발을 위한 노력이 지속되어야 할것이다.

참 고 문 헌

- [1] Ahmadi, J. H., Ahmadi, R. H., Dasu, S., and Tang, C. S., "Batching and Scheduling Jobs on Batch and Discrete Processors," *Operations Research*, vol. 40, 750-763, 1992.
- [2] Bitran, G. R., and Tirupati, D., "Planning and Scheduling for Epitaxial Wafer Production," *Operations Research*, vol. 36, no. 1, 34-49, 1988.

- [3] Bitran, G. R., and Tirupati, D., "Development and Implementation of a Scheduling System for a Wafer Fabrication Facility," *Operations Research*, vol. 36, no. 3, 377-395, 1988.
- [4] Chandru, V. Lee, C. Y. and Uzsoy, R., "Minimizing Total Completion Time on Batch Processing Machines," *Int. Journal of Production Research*, vol. 31, 2097-2121, 1993.
- [5] Chandru, V. Lee, C. Y. and Uzsoy, R., "Minimizing Total Completion Time on a Batch Processing Machine with Job Families," *Operations Research Letters*, vol. 13, 61-65, 1993.
- [6] Deb, R. K., and Serfozo, R. F., "Optimal Control of Batch Service Queues," *Advances in Applied Probability*, vol. 5, 340-361, 1973.
- [7] Fowler, J. W., Phillips, D. T., and Hogg, G. L., "Real-Time Control of Multiproduct Bulk Service Semiconductor Manufacturing Process," *IEEE Trans. Semicond. Manufact.*, vol 5, 158-163, 1992.
- [8] Glassey, C. R. and Weng, W., "Dynamic Batching Heuristics for Simultaneous Processing," *IEEE Trans. Semicond. Manufact.*, vol. 4, 77-82, 1991
- [9] Gurnani, H., Anupindi, R., and Akella, R., "Control of Batch Processing Systems in Semiconductor Wafer Fabrication Facilities," *IEEE Trans. Semicond. Manufact.*, vol 5, 319-328, 1992.
- [10] Ikura, Y., and Gimple, M., "Efficient Scheduling Algorithms for a Single Batch Processing Machine," *Operations Research Letters*, vol. 5, 61-65, 1986.
- [11] Lee, C. Y., Uzsoy, R., and Martin-Vega, L. A., "Efficient Algorithms for Scheduling Batch Processing Machines," *Operations Research*, vol. 40, 764-775, 1992.
- [12] Medhi, J., "Waiting Time Distribution in a Poisson Queue with a General Bulk Service Rule," *Management Science*, vol. 21, no. 7, 777-782, Mar., 1975.
- [13] Neuts, M. F., "A General Class of Bulk Queues with Poisson Input," *Annu. Math. Statist.*, vol. 38, 759-770, Mar., 1967.
- [14] Powell, W. B. and Humblet, P., "The Bulk Service Queue with a General Control Strategy: Theoretical Analysis and a New Computational Procedure," *Oper. Res.*, vol. 34, no.2, 267-275, 1986.
- [15] Uzsoy, R., "Scheduling Batch Processing Machines with Nonidentical Job Sizes," *International Journal of Production Research*, vol. 32, 1615-1635, 1994.
- [16] Weng, W. and Leachman, R. C., "An Improved Methodology for Real-Time Production Decisions at Batch-Process Work Stations," *IEEE Trans. Semicond. Manufact.*, vol 6, 219-225, 1993.