

공정 교체 시간을 고려한 배치작업의 일정관리 Incompatible Batch Scheduling with Setup Time

김주일, 이영훈

연세대학교 산업시스템공학과
서울시 서대문구 신촌동 134

bluefrog@yonsei.ac.kr, youngh@yonsei.ac.kr

Abstract

본 논문은 작업이 공정시간 등의 특성으로 인해 Family로 구분되어지고 동일한 Family 내의 작업끼리만 배치로 구성할 수 있는 제조환경에서의 스케줄링에 관한 연구이다. 이때 배치의 최대크기는 정해져 있어 그 이하의 수로 구성된 작업들로 하나의 배치를 구성할 수 있다. 동일 Family 내의 작업은 작업시간이 동일하고 배치로 구성된 작업이 진행될 때 수에 관계없이 하나의 작업에 소요되는 시간과 동일한 시간이 소요된다. 또한 다른 Family와의 작업이 이어질 때는 교체시간이 발생한다. 목적함수는 전체지연 시간을 최소화하는 것이며 납기는 동일한 Family에 있는 작업일지라도 각 작업별로 주어진다. 이 목적함수의 최적해를 구할 수 있는 동적계획법을 제시하고 이를 응용한 발견적 기법을 개발, 적용하여 각각 성능비교를 실시하였다.

1. 서론

배치 스케줄링은 제조공장의 설비가 자동화 및 고도화되면서 동시에 여러 개의 작업을 진행할 수 있는 환경에서 배치의 구성과정을 스케줄링에 포함시켜 고려하는 일정계획을 말한다. 배치로 구성될 수 있는 작업은 동일한 성격을 가지고 있는 경우가 대부분이며 때로는 어떤 작업도 배치로 구성될 수 있는 경우도 있다. 본 논문은 작업이 공정시간 등의 특성으로 인해 Family로 구분되어지고 동일한 Family 내의 작업끼리만 배치로 구성할 수 있는 경우(Incompatible)의 스케줄링에 관한 연구이며 이때 배치의 최대크기는 정해져 있어 그 이하의 수로 구성된 작업들로 하나의 배치를 구성할 수 있다. 동일 Family 내의 작업은 작업시간이 동일하고 배치로 구성된 작업이 진행될 때 수에 관계없이 하나의 작업에 소요되는 시간과 동일한 시간이 소요된다. 목적함수는 전체지연시간(Total Tardiness)을 최소화하는 것이며 납기는 동일한 Family에 있는 작업일지라도 각 작업별로 주어진다.

이와 같은 형태의 작업환경은 반도체 제조공정에 찾을 수 있다. 반도체 제조공정은 제조환경에 따른 복잡한 공정으로 인하여 정밀한 일정관리가 필요하다. 특히 웨이퍼 Fabrication 공정은 수백가지 이상의 공정으로 구성되어 반도체 제조공정 중에서 가장 복잡하다고 알려져 있으며 공정마다 각기 다른 작업시간을 갖고 있다. 공정의 종류가 변할 때는 설비세척과 공구의 교체로 인하여 일정시간의 교체시간이 발생한다. 웨이퍼 Fabrication 공정 중 확산공정은 일정 수의 웨이퍼가 실린더 리액터와 함

께 들어가 밀폐되어져 캐리어가스와 함께 가열된다 웨이퍼는 자재특성에 따른 표준 로트 크기가 주어져 있으며 일반적으로 6 개에서 12 개의 로트로 구성된다. 이들은 하나의 배치를 이룬 후에는 마지막 공정까지 함께 작업이 진행된다. 공정의 화학적인 특성으로 하나의 배치를 구성하는 작업들은 동일한 작업시간을 가져야만 한다. 동일한 작업시간을 이루는 작업들을 하나의 Family를 구성한다. 이 공정의 효율적인 스케줄링이 필요한 이유는 이들의 작업시간이 일반적으로 다른 공정에 비하여 긴 특성 때문이다. 반도체 산업은 대체적으로 적은 재공으로 짧은 시간 내에 납기를 맞추는 형태의 제조환경이 요구된다.

배치 공정의 일정관리는 작업을 어떻게 배치로 구성할 것인가와 구성되어진 배치들을 어떻게 나열할 것인가의 2 가지의 의사결정을 이루어지며 서로 밀접하게 연결되어있어 문제의 어려움을 더한다. 일반적인 평가지수를 사용하는 정적인 문제에서 Family가 명확하게 나누어지는 경우 위의 2 가지 의사결정은 별도로 나누어 해결될 수 있다(Uzsoy[1]). 본 논문에서는 교체시간을 가진 문제를 정의하고 단일 배치 공정기계에서 고정된 배치크기와 고정된 작업 Family를 가지고 있으며 배치간 교체시간이 있는 문제에 대해 문제의 Polynomial time 알고리즘인 동적계획법을 제시하였다. 또한 Family의 수가 커질수록 동적계획법의 계산시간이 급격히 늘어남에 따라 빠른 시간에 실행해를 구할 수 있으며 동적인 작업환경에서도 적용할 수 있는 발견적 규칙을 제시하여 각각 성능비교를 실시하였다.

배치 스케줄링의 전체지연시간을 최소화하는 문제는 여러 연구자에 의해 광범위하게 이루어졌다.

Du and Leung[2]에서는 일반적인 작업환경에서 이 문제는 NP-hard 문제임을 증명하였다. 그리고 Lawler[3]에는 이 문제에 관해 Pseudo-polynomial 동적계획법을 제시하였다. Mehta and Uzsoy[4]에는 Family 가 나누어진 상황에서 고정된 Family 수와 고정된 배치크기를 가진 문제에 관하여 Polynomial time 에 실행될 수 있는 동적계획법을 제시하였고 발견적 기법을 같이 비교하였다. 평가지수로서 전체가중지연시간에 관한 연구에서 Vepsalainen and Morton[5]에는 Apparent Tardiness Cost(ATC) 기법을 소개하였다. 또한 Chambers[6]에는 배치의 크기를 정해져 있는 문제에서 Decomposition Heuristic(DH)을 개발하였고 ATC 기법을 이 환경에 맞추어 변형하여 비교 분석하였다. 김영대[7]에는 납기에 관한 문제에 관한 기존 발견적 기법과 새로운 발견적 기법을 제시하여 비교 실험하였다.

2. 문제 정의 및 해법

다루고자 하는 문제는 모두 n 개의 작업을 대상으로 한다. 각 작업들은 m 개 Family 로 나누어져 있고 각각의 Family 는 n_j 개의 작업이 존재한다. 즉 작업 총수는 n 이다. 각각의 공정은 B 개의 배치 단위로 동시에 작업이 진행되어진다. 각각의 작업은 모두 납기를 가지고 있으며 j Family 의 i 번째 작업의 납기는 d_{ij} 로 표시한다. Family j 의 배치 B_{kj} 에는 B 개의 작업을 넘게 할당할 수 없다. j Family 의 i 번째 작업의 작업완료시간은 C_{ij} 로 표시하며 지연시간 dms T_{ij} 로 표시하며 목적함수인 전체지연시간은 Family j 의 총지연시간 T_j 의 합으로 구할 수 있다. 수식적 표현은 아래와 같다.

$$T_{ij} = \max(C_{ij} - d_{ij}, 0) \quad \sum T_j = \sum_{j=1}^m \sum_{i=1}^{n_j} T_{ij}, \forall ij$$

각 작업이 진행되어질 때 이전 작업과 Family 의 종류가 다르다면 공정시간에는 일정시간의 교체시간이 추가된다. 이 문제에서 교체시간은 Family 의 종류에 상관없이 일정하다고 가정하였다. 배치를 구성할 때 마지막 배치를 제외한 모든 배치는 배치 최대크기만큼의 작업들로 구성된 최적스케줄이 존재하며 이 내용은 쉽게 증명할 수 있다. 또한 같은 Family 내의 배치들은 납기에 따라 연속적으로 스케줄링 되어진 해 중에서 최적 스케줄이 존재한다. 이를 수식으로 표현하면 다음과 같다.

$$\max_{ij \in B_{kj}} \{d_{ij}\} \leq \max_{ij \in B_{k+1j}} \{d_{ij}\}, \forall i, j, k$$

2.1. 동적계획법 (DP)

각 Family 의 작업들은 납기가 작은 순서로 배치를 최대크기에 맞게 채워서 배치를 구성할 수 있다. 하나의 Family j 에는 총 k_j 개의 배치가 존재하게 된다. 부분적으로 이루어진 스케줄에는 Family j 의 첫번째부터 w_j 까지 배치들로 구성되어진다. $f(w_1, w_2, \dots, w_m, T, q)$ 은 Family j 의 첫번째부터 w_j 개의 배치까지 스케줄링이 이루어졌을 때 전체 납기지연시간의 최소값이며 T 는 현재까지 이루어진 Family 교체회수이고 q 는 직전에 할당되어진 작업의 Family 이다. 우리가 최종적으로 구하고자 하는 값은 $f(k_1, k_2, \dots, k_m, T, q)$ 이며 T 와 q 는 고정되어있지

않다.

Boundary Condition: $f(0, 0, \dots, 0, 0, *) = 0$

$$f(w_1, w_2, \dots, w_m, T, q) = \infty \text{ if any } w_j > k_j \text{ or any } w_j < 0 \\ \text{ or } T < 0 \text{ or } T > \sum k_i - 1.$$

Recursive Equation:

$$f(w_1, \dots, w_q, \dots, w_m, T, q) = \\ \min_q \{f(w_1, \dots, w_{q-1}, \dots, w_m, T^*, q^*) +$$

$$\sum_{iq \in B_{wq}} \min(0, C_{iq} - d_{iq})\}$$

where if $q = q^*$ then $T^* = T$

else $T^* = T - 1$

작업완료시간 C_{iq} 는 아래와 같이 계산한다.

$$C_{iq} = \sum_{k=1}^m w_k p_k + s \cdot T$$

2.2. Batch Apparent Tardiness Cost with Setup Time (BATCS)

Mehta and Uzsoy[4]는 Vepsalainen and Morton[5]에 의해 제시된 Apparent Tardiness Cost 기법을 배치 상황에 맞추어 변형시킨 BATC(Batch Apparent Tardiness Cost)기법을 소개하였다. 본 논문의 교체시간을 고려한 경우, BATC 를 아래와 같이 변형하여 BATCS 라 하고 이에 대한 인덱스를 아래와 같이 구한다.

$$BATS(B_{ij}, Q) = \frac{1}{p_j} \exp \left\{ - \frac{\sum_{ij \in B_{ij}} \max(d_{ij} - s^* - p_j - t, 0)}{k(p+s)} \right\}$$

여기에서 사용된 k 는 스케일링 파라미터이며 s 는 교체시간이며 s^* 는 직전에 할당되어진 작업의 Family q^* 와 현재 할당할 대상인 작업의 Family j 가 동일하다면 값을 가지지 않고 다르다면 교체시간 s 의 값을 가진다. 전체적인 기법적용순서는 다음과 같다.

Step1: 각 Family 의 작업들을 납기가 작은 순서로 나열한다

Step2: 작업들을 정해진 배치 크기만큼 채워서 배치를 구성한다.

Step3: 작업 가능한 배치에 대해 BATCS 인덱스를 구하고 그 중 가장 큰 값을 갖는 배치를 스케줄링한다.

Step4: 모든 배치가 스케줄링이 되지 않았다면 Step 3 부터 다시 시행한다.

스케일링 파라미터 값인 k 는 0.5 에서 2 가지 값을 사용하여 구한 후 목적함수값 중에서 최소납기지연시간을 선택한 k 를 최적 BATCS 값(BEST BATCS)으로 사용하였다.

2.3. Decomposition Heuristic (DH)

Decomposition Heuristic 은 Chamber[6]에 의해 단일공정에 전체가중지연시간을 최소화하는 문제에 적용되었다. 이 기법은 전체 문제를 부분적으로 나누어 각각의 부분적인 최적값을 여러 번 구하여 이를 조정함으로써 한정된 시간 안에 최적해에 가까운 값을 구하고자 하는 방법이다. 전체문제에서 λ 개 만큼의 배치만으로 부분 최적값을 구한 후 이 중 이 최적해의 스케줄 중에 f 개의 배치를 전체 스케줄에 할당을 한 후 다시 남은 배치들로 모든 작업의 할당이 끝날 때 까지 반복하는 과정을 한다. 모든 배치가 전체 스케줄에 할당이 되면 기법 적용 전의 스케줄의 값과 비교하여 값이 크다면 이전 스케줄을 최종 스케줄로 확정하고 작다면 이 전체 스케줄을 다시 처음부터 DH 기법을 적용하여 개선된 해가 나오지 않을 때 까지 반복한다.

동적계획법은 문제의 크기가 커질수록 State 수가 급격히 늘어나 계산시간이 크게 증가하는 데 비해 이 기법은 한계가 정해진 State 수를 사용함으로써 문제의 크기가 커질수록 동적계획법보다 빠른 시간안에 최적해에 가까운 값을 산출하게 된다. 또한 이 기법의 계산의 정확성과 계산시간은 각각의 부분 동적계획법을 적용하는 λ 와 f 값의 크기에 따라 달라진다. 즉 λ 를 크게하면 보다 최적해에 가까운 값을 산출하지만 계산시간이 증가하게 되고 f 값을 크게하면 보다 작은 분할회수로 빠르게 값을 구할 수 있지만 최적해와의 차이는 커지게 된다.

3. Computational Experiment

발견적 기법의 결과값을 비교하기 위하여 각각의 실험배경에 사용되는 예제는 랜덤하게 생성하여 비교하였다. <표 1>은 이 실험을 위한 결과치를 실험 모형의 실험설계이다.

< 표 1 > 실험설계

Parameter	사용된 값
Family 의 작업 수 (n_j)	30, 40, 50, 60
Family 의 수 (m)	3, 4
배치크기 (B)	4, 8
Family 작업시간 (P_j)	2 with probability 0.2 4 with probability 0.2 10 with probability 0.3 16 with probability 0.2 20 with probability 0.1
교체시간	$\bar{p} * S$ $S = 0.5, 1.5$
작업별 납기 (d_{ij})	$d_{ij} = Uniform \left[\mu - \frac{\mu R}{2}, \mu + \frac{\mu R}{2} \right]$ $\mu = C * (1-T)$ $C = n_j * m * E[P]/B$ $E[P] = \text{Expected processing time}$ $T = 0.3, 0.6$ $R = 0.5, 1.5$

실험에 사용되어진 파라미터의 경우수는

Family 의 수, Family 의 작업 수, 배치의 크기, 납기 관련 생성 파라미터 T, R , 교체시간 파라미터 s 에 따라 <표 1>과 같이 모두 128 가지의 경우가 나온다. 평가지수인 전체지연시간의 변화는 발견적 기법의 값을 최적값인 동적계획법의 값에 대한 비율로 계산하였다. 이 모형의 결과는 <표 2>와 같다. 128 가지의 경우의 결과에서 직접적으로 관련되는 Family 의 수, 작업 수, 배치의 크기에 따른 16 가지의 경우로 나누었다. 이 실험은 모두 C 언어로 프로그래밍되어졌고 PentiumIII 500MHz PC 에서 실험하였다.

< 표 2 > 발견적 기법의 평가

	Best BATC	DH 6,3	DH 12,6	DH 24,12
3*30*4	2.00	1.76	1.27	1.00
3*40*4	2.10	2.02	1.68	1.53
3*50*4	2.63	3.11	1.71	1.50
3*60*4	3.72	2.36	2.21	1.80
3*30*8	1.26	1.27	1.00	1.00
3*40*8	2.17	1.69	1.17	1.00
3*50*8	2.44	1.98	1.45	1.13
3*60*8	2.54	1.98	1.63	1.16
4*30*4	2.68	2.14	1.73	1.39
4*40*4	3.32	2.77	1.89	1.73
4*50*4	3.45	2.92	2.05	1.96
4*60*4	3.51	3.16	2.40	2.22
4*30*8	1.72	1.44	1.13	1.00
4*40*8	1.90	1.54	1.29	1.11
4*50*8	2.28	1.95	1.54	1.33
4*60*8	2.46	2.10	1.68	1.34
평균	2.51	2.14	1.61	1.39

< 표 3 > 계산시간 (단위: 초)

	DP	BEST·BATC	DH 6,3	DH 12,6	DH 24,12
3*30*4	83.83	0.01<	0.01<	0.00	0.34
3*40*4	85.67	0.01<	0.04	0.09	0.39
3*50*4	86.44	0.01<	0.05	0.12	0.44
3*60*4	88.09	0.01<	0.05	0.12	0.45
3*30*8	82.71	0.01<	0.01<	0.04	0.32
3*40*8	82.78	0.01<	0.01<	0.03	0.37
3*50*8	83.65	0.01<	0.01	0.05	0.39
3*60*8	84.15	0.01<	0.01	0.08	0.40
4*30*4	92.92	0.01<	0.06	0.29	0.45
4*40*4	103.28	0.01<	0.07	0.35	0.49
4*50*4	211.19	0.01<	0.10	0.36	0.53
4*60*4	486.26	0.01<	0.12	0.39	0.77
4*30*8	79.83	0.01<	0.05	0.08	0.38
4*40*8	81.96	0.01<	0.05	0.16	0.38
4*50*8	83.93	0.01<	0.06	0.28	0.42
4*60*8	88.23	0.01<	0.07	0.27	0.43
평균	119.06	0.01<	0.06	0.17	0.43

*0.01<: 0.01 초 이하의 시간

이상의 결과에서 동적 계획법은 문제의 크기가 커질 때 계산 시간이 비약적으로 상승하나 발견적 기법은 일정시간 안에 결과를 찾아내는 것을 확

인할 수 있다.

4. 결론

반도체 환경에 적합하며 교체시간을 고려한 상황에서 전체지연시간을 최소화하는 것을 목적으로 한 문제에서 최적값을 고려한 동적계획법을 개발하였고 이를 응용한 Decomposition heuristic 과 배치상황을 고려한 BATCS 의 발견적 기법을 제시, 성능을 비교하였다. 동적계획법은 최적값을 찾아주나 Family 의 수나 작업의 크기 등 문제 크기가 커지면 비약적으로 계산시간이 증가함을 알 수 있었으며 발견적 기법은 비교적 빠른 시간에 적당한 실행가능해를 찾을 수 있었다.

이후에는 적용범위로는 교체시간이 각 Family 에 따라 달라지는 경우에 대한 문제의 적용을 고려할 수 있다. 이러한 문제는 교체시간의 변동으로 인한 동적계획법의 State 수 증가로 인한 계산시간의 급격한 한계로 인하여 최적해를 찾는 방법보다는 보다 효율적으로 적당한 실행가능해를 찾을 수 있는 방법이 모색되어야 할 것이다.

Reference

- [1] Uzsoy, R. (1995) Scheduling batch processing machines with incompatible job families. International Journal of Production Research, 33, 2685-2708
- [2] Du, J. and Leung, J.Y.T. (1990) Minimizing total tardiness on one machine is NP-hard. Mathematics of Operations Research, 15, 83-495.
- [3] Lawler, E.L. (1977) A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness. Annals of Discrete Mathematics, 1, 331-342.
- [4] Sanjay V. Mehta, Reha Uzsoy, (1998) Minimizing total tardiness on a batch processing machine with incompatible job families. IIE Transactions 30, 165-178.
- [5] Vepsalainen, A. and Morton, T.E. (1987) Priority rules and lead time estimation for job shop scheduling with weighted tardiness costs. Management Science, 33, 1036-1047.
- [6] Chambers, R.J., Carraway, R.L., Lowe, T.J. and Morin, T.L. (1991) Dominance and decomposition heuristic for single machine scheduling. Operations Research, 39, 639-647.
- [7] Yeong-Dae Kim, Jung-Ug Kim, Seung-Kil Lim, and Hong-Bae Jun, (1998) Due date based scheduling and control policies in a multiproduct semiconductor wafer fabrication facility. IEEE Transactions on semiconductor manufacturing, Vol. 11, No. 1, February 155-164