

# 도착시간과 제약시간이 존재하는 배치생산공정 일정계획

김영옥, 조문수\*  
숭실대학교

woogie@ssu.ac.kr, jmsu@ssu.ac.kr\*

## Scheduling of Batch Manufacturing Process with Arrival Time and Time Constraint

Kim Young Wook, Cho Moon Soo\*  
Soongsil Univ.

### 요 약

본 논문은 반도체 제조공정의 Queue Time 이라는 제한 시간 내 진행하여야 하는 제약 시간과 작업의 도착시간, Batch 생산이라는 특성을 반영한 Scheduling Problem 에 대해 Heuristic Algorithm 의 한계점을 보완한 Meta Heuristic Algorithm 중 하나인 Genetic Algorithm 을 제안한다. 기존 Heuristic 방법으로 구현한 Rule Based Priority Scheduling, 수리모형을 풀어 최적해를 구하는 MIP 와 비교하여 제안한 방법이 우수한 성능을 보임을 확인하였다.

### I. 서 론

공정 Scheduling 이란, 단기간 내에 수행해야 하는 작업 계획을 말하며, 설비 가동률 및 생산성 극대화를 목적으로 한다. 이를 위해서 설비에 진행되는 작업들을 적절히 배치해 유휴시간을 최소화해야 하고, 각 설비에 작업을 적절히 분배해야 설비 가동률 및 생산성 극대화를 위한 총 작업시간 최소화 목적을 달성할 수 있다.

Scheduling 문제에 대해 여러 가지 제조 특성을 반영해 최적해를 찾는 수많은 연구가 이루어지고 있지만, 대부분 개별생산공정(DPM: discrete processing machine) 최적화에 대한 연구이며, 배치 생산공정(BPM: batch processing machine)에 대한 연구는 많이 부족하다. 여기서 개별생산공정은 웨이퍼를 하나씩 낱개로 생산하는 공정을 말하고 배치 생산공정은 여러 개의 제품을 묶어 배치(Batch)로 구성하고, 구성된 배치를 동시에 처리하는 공정을 말하며 반도체 생산공정에서 주로 쓰이는 용어이다.[1]

Scheduling 예제 중 하나인 Job-Shop Scheduling Problem(JSSP)의 최적해를 찾기 위해 사용되는 기법 중 수리모형의 경우 문제의 크기가 커짐에 따라 계산 시간이 비약적으로 증가해 실제 현장에서 사용은 불가능하며, 규칙 기반의 우선순위 할당 방식의 경우 구현이 간단하고 계산 시간이 오래 걸리지 않으나 최적해를 보장하지는 못한다.

따라서 본 연구에서는 반도체 제조공정의 Queue Time 이라는 제약 시간 특성과 작업의 예상 도착시간을 반영한 Batch 단위 생산공정의 특성을 JSSP 에 반영하여 Meta Heuristic Algorithm 인 Genetic Algorithm(GA)을

사용하여 효율적인 일정 계획 수립을 도모하고, Rule-Based Priority Scheduling 방식과 비교한다.

### II. 관련연구

공정 Scheduling 문제에 대한 연구는 지금까지도 많은 연구자들에 의해서 진행되고 있다. Job Shop 의 경우 단일 설비에서 다품종 제품을 생산하는 환경에 대한 일정 계획 문제이며, Flow Shop 의 경우 여러 공정을 순서대로 진행하여 제품을 생산하는 환경에 대한 Scheduling 문제이다. 반도체 제조업의 경우 Job Shop 과 Flow Shop 의 특성을 모두 가지고 있으며, 반도체 제조 관련 특성을 반영한 연구 또한 활발히 이루어지고 있다.

최근 3 년간 이루어진 Scheduling 문제에 대한 연구를 보면 다음과 같다. 먼저 Joo(2019)는 병렬 설비를 위한 주기적 일정 계획 문제를 해결하기 위해 도착시간이 존재하는 여러 작업들에 대해서 총 작업 완료 시간을 최소화하기 위한 일정 계획 알고리즘을 제안하였다. ERD(Earliest Ready Date)규칙에 따라 생산 순서를 결정하였으며, Makespan 을 최소화하는 일정 계획을 수립한다.[2] Han *et al.*(2019)는 대기시간을 허용하지 않는 두 단계 조립 시스템에서 총 작업시간 최소화를 위한 일정 계획을 연구하였다.[3] List Scheduling Algorithm 과 Modified NEH Algorithm 그리고 Simulated Annealing Algorithm 을 사용하여 최적화하였으며, 세 가지 Algorithm 간의 성능 비교를 하였다. Kang *et al.*(2019)는 JobShop 공정을 Genetic Algorithm 을 사용하여 최적화하였으며, 설비의 생산능력과 리드타임의 특성을 반영하는 문제를 제시하고 있다.[4] 위 논문들의 경우 개별생산공정에 대한 연구이다.

배치 생산 공과 관련된 일정 계획 문제의 연구는 아직 많이 이루어지지 않았다. Koo(2014)는 평균 납기 지연 최소화를 위한 배치 생산공정의 실시간 로딩 전략에 대해 LBT 방식을 제안하였으며, 다중 설비가 아닌 단일 설비에 대한 배치 구성 전략을 이야기하고 있다.

앞서 살펴본 논문들의 경우 다양한 제조 산업의 특성들을 적용하여 최적화하였지만, 배치 생산공정의 특성을 반영한 연구는 매우 적은 편임을 알 수 있다. 기존 연구 중 Koo(2014)만이 배치 생산환경을 다루고 있지만, 단일 공정이며, 설비 또한 단일 설비 환경에서의 최적화이기 때문에 실제 현장의 상황과는 거리가 멀다. 본 논문에서는 제약시간과 도착시간이 존재하는 배치생산공정에 대한 총생산완료시간과 제약시간초과의 최소화 문제를 풀어본다.

### III. 문제정의

본 연구에서 사용될 구성 요소로는 작업과 설비가 있으며, 작업의 정보에는 도착시간과 잔여 제약 시간이 존재한다. 설비에는 배치의 크기, 그리고 작업시간이 있다. 본 연구의 Scheduling 문제는 다음과 같은 특징들을 갖는다.

- 1) 각 작업은 서로 다른 도착시간 및 제약 시간을 가지고 있다.
- 2) 도착시간이 존재하는 작업은 해당 시간 경과 이후 진행 가능하다.
- 3) 도착시간이 서로 다른 작업이 배치로 구성될 경우 해당 배치는 가장 늦게 도착하는 작업을 기준으로 시작 가능하다.
- 4) 기계 별 서로 다른 배치의 크기를 갖는다.

본 연구의 목적함수는 총작업완료시간과 작업 별 잔여 제약시간에 가중치를 부여하여 목적함수의 값을 최소화 가 목표로 식(1)과 같이 표현된다.

$$\text{Minimize } Z = w_f MS - w_g QD_{(1)}$$

Table1. Notation  $w_1MS-w_2QD$

Notation	Description
j	Job
m	Machine
b	Bucket(Batch Sequence)
$P_M$	Machine m에서 진행하는 Job j의 Process Time
$BS_M$	Machine m의 BatchSize
$QT_j$	Job j의 잔여 제약시간
$AT_j$	Job j의 도착시간
$X_{jmb}$	Job의 할당여부 (할당시 1 아닐경우 0)
$L_{mb}$	Machine m의 Bucket b에서의 시작시간
$QD_m$	Machine m에 할당된 Job의 잔여 제약시간 합
$MS_m$	Machine m의 총 작업완료시간

<Table 1>은 본 연구에서 사용되는 변수들에 대한 정의이다. 사용되는 Data는 Sampling된 Data로 Job과 Machine의 개수를 변경하며 실험한다.

### IV. 유전 알고리즘

최적화에 사용할 알고리즘은 Meta Heuristic Algorithm 중 하나인 Genetic Algorithm(GA)이다. GA는 자연세계의 진화 과정을 모방한 최적해를 탐색하는 기법

중 하나이다. 해를 염색체의 묶음인 유전자로 표현하며, 적합도 함수를 통해 해의 적합성을 알 수 있다. 적합도 함수란 염색체가 얼마나 최적해에 가까운지 평가하는 함수이다.

초기 유전자 집단은 Random 하게 작업을 배치한다. 유전자 내 염색체 표현은 <Figure 1>과 같이  $\sum_m BS_m * b$ 의 크기를 갖는 배열에 작업을 할당하여 표현한다.

Figure 1. 염색체의 표현

M1,b1	M1,b1	M1,b1	M2,b1	M2,b1	M1,b2	M1,b2	M1,b2	M2,b1	M2,b1
J1	J8	J2	J6	J3	J7	J9	J10	J4	J5

적합도는 식(1)로 평가하며, 적합도가 높을수록 선택될 확률이 높은 룰렛 휠 방식으로 2개의 집합을 선택하여 부모 1,2로 선정한다. 이외의 유전자들은 삭제 후 Random Sampling을 통해 새로운 유전자로 대체한다. <Figure 2>와 같이 선택된 2개의 유전자를 부분일치 교배하여 특정 위치의 작업을 교환하고, <Figure 3>과 같이 작업이 중복되는 부분은 재조정한다.

Figure 2. partially mapped crossover

J1	J8	J2	J6	J3	J7	J9	J10	J4	J5
J3	J7	J1	J6	J2	J10	J9	J4	J8	J5

Figure 3. legalize the offspring

J8	J7	J1	J6	J2	J10	J9	J2	J4	J5
J1	J8	J2	J6	J3	J7	J9	J4	J10	J5

이후 일정 확률로 염색체의 일부분을 다른 염색체와 치환한다. 위 과정을 차례대로 반복하며 적합도 평가 기준을 만족하거나 일정 횟수를 초과할 경우 마지막 유전자 집합 중 적합도가 가장 높은 유전자를 해로 나타내며 Algorithm을 종료한다.

### V. 실험결과

본 연구에서 제안한 GA의 성능 검증을 위한 실험이 수행되며, Rule Based Priority Scheduling(RBPS) 및 ortools를 이용한 MIP(Mixed Integer Programming)와 비교한다. 성능 검증 시 사용되는 문제의 사이즈는 <Table 2>와 같이 설정하여 실험한다.

Table2. Size of Problem

Problem Sequence	Jobs	Machines
1	9	2
2	20	3
3	50	4
4	70	5
5	90	6

GA는 10개의 해집단을 가지며, 교차율과 돌연변이율은 각각 0.7과 0.3으로 고정하였고, 500세대에 도달하며 종료하였다. 모든 실험은 3.60 GHz AMD Ryzen 5 3600 6-Core Processor, 16GB RAM의 PC에서 Python 개발 환경으로 실험되었다.

문제들에 대해서 OR-tools MIP Solver를 사용하여 3장에서 제시한 수리 모형에 대한 최적해와 RBPS 및 GA의 성능을 비교하였다. MIP의 경우 1800초의 계산

시간제한을 두며 시간 내 최적해를 찾지 못할 경우 근사 최적해를 이용해 다른 Algorithm 해들의 성능을 비교하도록 하였다. 각 실험 데이터들에 대해서 RPR(Relative Performance Ratio)를 계산하였다. 여기서, RPR은 식(2)와 같이 정의된다.

$$RPR(\%) = \frac{Z_a - Z_{Min}}{Z_{Min}} \times 100 \quad (2)$$

$Z_{Min}$ 은 3개의 성능 비교 Algorithm 중 목적함수 값이 가장 작은 값을 의미하며,  $Z_a$ 는 평가대상 Algorithm의 목적함수 값이다.

작은 사이즈의 문제에서는 MIP로 최적해 도출이 가능하나 문제의 사이즈가 커질수록 제한 시간 내 최적해 도출이 불가능함을 알 수 있었다. RBPS의 경우 수행 횟수와 상관없이 일정한 결과를 도출하였고, GA의 경우 문제의 사이즈와 상관없이 우수한 수행 속도를 보여주며 최적해 도출은 불가능하였지만, 최적해에 근사한 결과를 도출하는 것을 알 수 있다.

## VI. 결론 및 추후연구

알려진 바와 같이 Meta Heuristic Algorithm의 경우 문제의 복잡도가 커지게 될 경우 전역 최적해를 구하는 MIP 모델에 비해 해를 찾는 시간이 상당히 짧고, RBPS의 경우에는 우선순위를 위한 규칙을 모두 탐색하며, 조건 문과 완전히 일치하지 않을 경우 문제 해결이 불가능해 제안한 알고리즘의 성능이 속도나, 최적해 산출에서 더 뛰어날 것으로 보인다.

연구의 한계점으로는 Sampling된 Data를 사용하였기 때문에 실제 제조 공정에서 비슷한 유형의 문제를 해결하기에는 어려움이 있다. Sampling Data가 아닌 실제 제조공정의 Data를 통해 실험해 볼 필요가 있으며, 추가로 확률적인 부분들인 설비 고장이나 작업자 개입 등을 고려하여 좀 더 현장과 유사한 데이터 환경에서 연구를 진행해 볼 필요가 있다. 본 논문에서 제안한 알고리즘과 추후 연구 과제는 반도체 제조업이 외에도 다양한 제조업의 생산 스케줄링에 도움이 될 것으로 기대한다.

## 참 고 문 헌

- [1] Pyung-Hoi Koo, "A Real-Time Loading Strategy of Batch Processing Machines for Average Tardiness Minimization," *Journal of the Korean Institute of Industrial Engineers*, Vol.40, No.2, pp. 215-222, April, 2014.
- [2] Un-Gi Joo, "Periodic Scheduling Problem on Parallel Machines," *Journal of Convergence for Information Technology*, vol.9, No.12, pp.124-132, 2019.
- [3] Jun-Hee Han, Bong-Joo Jeong, Ju-Yong Lee, and Sang-Oh Shim, "A Scheduling Problem to Minimize Total Completion Time in a No-Wait Two-Stage Assembly-Type Flow Shop," *經營科學*, 第36卷, 第4號, pp.1-12, 12月, 2019.
- [4] In-Wook Kang, Ho-Yeon Kang, and Jong-Yeon Lee, "Genetic Algorithm을 이용한 Job-Shop 공정의 생산 스케줄링 최적화 분석," *Journal of the Korean Institute of Industrial Engineers*, pp.2681-2688, Nov, 2019