

처리속도가 가변적인 작업들의 일괄처리 일정계획문제

-A Batch Processing Problem for Jobs with Variable Processing Speed -

오 세 호 *

Oh Se Ho

이 근 부 *

Lee Keun Bu

Abstract

This paper deals with a batch processor model in which the batch processing speed depends on the jobs assigned to the batch. Each job is completed by accomplishing its required job processing quantity which is calculated as job speed product processing time. Its speed is given as not the exact value but the range. Thus the batch sets are constructed by the jobs which hold the speed in common. And the batch sets are processed as soon as possible. We developed the model to described the problem situation and adopt the total tardiness as the decision criterion.

Keywords : Batch Processor, Processing Speed, Tardiness

1. 서 론

본 연구에서는 몇 개의 작업들이 묶여서 배치 프로세서(batch processor)로 처리되는 상황을 다루고자 한다. 유기물 발효 공정, 도자기 열처리 공정, 반도체 제조과정, PCB(printed circuit board) 검사 공정 등에서 그 예를 찾아볼 수 있다[2, 6, 7]. 도자기

* 청주대학교 이공대학 산업공학과 교수

2006년 7월접수; 2006년 8월 수정본 접수; 2006년 8월 게재확정

열처리 과정처럼 일괄처리 시간이 미리 정해지는 모형이 대부분이지만 일괄처리 묶음 집합을 구성하고 있는 작업들의 공정시간에 따라 결정되는 좀 더 복잡한 문제도 제시되었다. 이러한 문제는 burn-in 모형이라고 일컬어지고 있는데 Lee et al. [8]과 Chandru et al. [3]등에 의해서 연구되었다. 처리시간이 서로 다른 여러 종류의 회로기판(circuit board)이 동일한 오븐(oven)에서 처리되는데 오븐의 처리 능력 범위 내에서 몇 개씩 묶여서 동시에 이루어진다. 이 때의 오븐의 처리시간은 같이 묶여있는 기판들의 처리시간들 중에서 가장 큰 값과 같다. 또한 공정시간이 제한적이긴 하지만 상황에 따라 변하는 모형에 대한 연구도 이루어 졌다[4, 5]. 본 연구에서는 전형적인 burn-in 모형의 변형으로서, 일괄처리의 처리 속도를 일정 범위 내에서 자유롭게 조절함으로써 작업들의 공정시간을 단축시킬 수 있는 모형을 제시하였다. 각 작업들은 요구 공정량과 작업속도, 납기로 정의되는데 공정량과 작업속도가 공정시간을 대신한다. 즉 공정시간은 공정 요구량을 작업속도로 나누어 계산한다. 작업속도는 일정 범위 안에서 선택이 가능하다. 작업속도를 공유할 수 있는 작업들끼리 묶음 집합을 이루게 함으로써 작업완료시간의 단축을 기대할 수 있을 것이다. 따라서 총납기지연을 최소화할 수 있는 방안을 모색하고자 한다.

2. 문제의 정의

일괄 공정은 속도의 조절이 가능한 배치 프로세서에 의해 이루어진다. 단위 작업들도 공정 속도를 조절할 수 있다. 동일한 작업속도로 가공이 가능한 작업들로 묶음을 구성하면 배치 처리가 가능하다. 각 작업들은 작업량, 작업속도, 납기로 정의된다. 실제 작업들의 작업량은 작업속도와 배치 공정시간이 곱해져 산정되는데 정해진 작업량을 충족시켜야 한다. 그러므로 동일한 묶음에 속한 작업들의 작업완료 시간은 요구 작업량에 따라 산출된다.

작업 $J_i(a_i, b_i, q_i, d_i)$ 의 성취 정도와 처리시간과의 관계는 다음 식으로 표현된다.

$$q_i(t) = pt \quad \text{where} \quad a_i \leq p \leq b_i \quad (1)$$

모형 수립을 위해 다음의 가정을 전제로 한다.

모형 수립을 위한 기본 가정

- (1) 시작 시점에 모든 작업들은 처리가 가능하다.
- (2) 배치 프로세서의 능력(capacity) 범위 안에서 단위 작업들이 투입된다.
- (3) 일단 일괄처리공정이 시작되면 새로운 작업이 추가될 수 없고 제거될 수도 없다.
- (4) 배치 프로세서의 처리속도는 묶음 집합을 이루는 각 작업들의 가능 작업 속도에 따라 조절되며 가능한 한 빠른 속도가 선택된다.

(5) 주어진 납기를 지키지 못하면 납기 지연이 발생하고 총 납기지연은 각 작업들의 납기지연의 합으로 산정된다.

묶음의 처리속도는 배치 처리 묶음집합의 구성에 따라 달라진다. 가정(4)에 의해 같은 집합에 속한 각각의 작업들은 동일한 속도로 처리된다.

$J_1 (3, 5, 12, 5)$, $J_2 (2, 4, 8, 4)$, $J_3 (5, 6, 12, 6)$, $J_4 (3, 6, 18, 5)$, $J_5 (5, 7, 15, 8)$ 로 주어진 5개의 단위작업들의 경우, 묶는 방법에 따라 어떤 상황이 전개되는지 알아보자. 배치 프로세서의 처리능력을 3이라 했을 때, (J_1, J_2) , (J_3, J_4, J_5) 를 일괄 처리한 결과와 (J_1, J_2, J_4) , (J_3, J_5) 를 일괄처리한 결과는 각각 <그림 1>, <그림 2>과 같다. 전자는 총지연이 1이고 후자는 3이 발생한다.

	1	2	3	4	5	6	7	8
J_1								
J_2								
J_3								
J_4								
J_5								

굵은실선 : 시작지점, 점선 : 납기, 음영 : 공정

<그림 1> (J_1, J_2) , (J_3, J_4, J_5) 를 일괄처리한 결과

	1	2	3	4	5	6	7	8	
J_1									
J_2									
J_3									
J_4									
J_5									

굵은실선 : 시작지점, 점선 : 납기, 음영 : 공정

<그림 2> (J_1, J_2, J_4) , (J_3, J_5) 를 일괄처리한 결과

3. 해 법

앞으로의 논리 전개에 필요한 표기와 용어 정의는 다음과 같다.

표기와 용어 정의

T_k : k번째 배치 처리 후의 시점

$J_i (a_i, b_i, q_i, d_i) (i = 1, 2, \dots, n)$: 단위 작업(job)

a_i : J_i 의 최저 공정 속도

b_i : J_i 의 최고 공정 속도

q_i : J_i 의 공정 요구양(required processing quantity)

d_i : J_i 의 납기

C : 배치 프로세서의 일괄처리 능력

TD_t : 시점 t 까지의 총납기지연시간

BP_r : 묶음 r 의 일괄처리공정시간

B_t^i ($j = 1, 2, \dots, r$) : 시점 t 에서의 일괄처리가 가능한 작업들의 집합

N : 모든 작업들의 집합

FN : 처리된 작업들의 집합

R_t : T_k 에서 묶음 대상이 되는 작업들의 집합

BS_j : 배치 여유(batch slack)

납기지연은 작업의 완료시간과 비례하지 않기 때문에 가장 기본 모형인 단일기계 일정계획 문제에서도 납기지연을 최소화시킬 수 있는 해법을 개발하기가 쉽지 않다. 근본적으로 일괄처리 문제들은 적재문제(packing problem)와 동일한 복잡도를 갖는다고 보면 NP-hard 임이 자명하다. 본 연구에서는 납기지연 문제의 이론적 접근 방법인 이웃쌍교환 분석(adjacent pairwise interchange analysis)으로 출발하여 묶음 작업(batch jobs)들 간의 교환에 따른 납기지연을 비교하고자 한다.

정의 : 두 모수 집합의 원소 u_i, u_j 와 v_i, v_j 가 $u_i < u_j$ 이면 $v_i \leq v_j$ 를 의미한다면 두 모수 집합은 agreeable하다고 한다.

위의 정의대로 모든 단위 작업 쌍에 대해서 공정 시간과 납기가 agreeable이면 총 납기지연은 SPT 순서(shortest processing time sequencing) 혹은 EDD 순서(earliest due date sequencing)에 의해 최소화된다. 따라서 배치 프로세서의 처리 용량이 1이고 모든 단위 작업 쌍에 대해서 q_i/b_i 와 d_j 가 agreeable인 특별한 경우에는 총 납기지연을 SPT 순서 혹은 EDD 순서에 의해 최소화시킬 수 있을 것이다. 만약 모든 단위 작업 쌍에 대해서 q_i/b_i 와 d_j 가 agreeable인 상황에서 일괄처리가 가능하다면 낱개로 처리하는 것보다 가능한 한 묶음으로 처리하는 것이 후속되는 작업들의 납기지연을 줄일 가능성을 높여 줄 것이다. 즉 일괄처리가 낱개로 처리하는 대안을 지배한다. 이와 같은 지배성은 모든 단위 작업 쌍에 대해서 q_i/b_i 와 d_j 가 agreeable일 때만 성립되기 때문에 본 모형에 적용시키기에는 무리가 따른다.

다음으로 이러한 규칙들보다는 약하지만 진행 시점 T 에 의해 순서가 결정되는 규칙을 고려해 보자.

정리 1. 시점 T 에서 단위 작업 J_i 와 J_j 의 순서를 결정하려 할 때, 다음 식이 성립하는 경우를 제외하고는 납기가 이른 작업을 선행 시킨다.

$$T + \max\{q_j/b_i, q_j/b_j\} > \max\{d_i, d_j\} \quad (2)$$

위의 정리를 통해서 두 작업간의 순서는 어느 진행 시점에서 결정되느냐에 따라 달라진다. 결국 위의 규칙은 최적조건의 충분조건이 아닌 필요조건이라고 할 수 있다. 그러나 최적 선행관계를 보장할 수 없더라도 진행 시점 T 에서 고려되는 작업들(candidates)간의 선행관계를 파악할 수 있기 때문에 배치 집합을 생성하는데 이용될 수 있다.

해법의 절차를 약술하면 다음과 같다.

해법의 절차

단계 0: $FN = \emptyset$. $T_0 = 0$. $k = 0$

단계 1: $kard(FN) = n$ 이면 종료.

집합 $N-FN$ 을 대상으로 순서결정규칙에 의해 R_t 결정

단계 2: R_t 를 대상으로 묶음 B_t^j ($j = 1, 2, \dots, r$)를 생성하여 배치여유가 가장 작은 것(B_t^s)을 선택

단계 3: 일괄공정시간 산정과 처리된 작업 수정.

$$BP_s = \max\{q_j/b_i, j \in B_t^s\}$$

$$k = k + 1$$

$$FN = FN - B_t^s$$

$$T_k = T_k + BP_s$$

k번째 일괄처리가 이루어진 직후인 T_k 시점에서 단위 작업들은 다음의 규칙들에 의해서 배열된다.

순서 결정 규칙

- (1) SPT
- (2) EDD
- (3) 수정 납기(modified due date)
- (4) 작업속도 가변성(variability of processing speed)

첫 번째 SPT 규칙을 적용할 경우, 고려되는 각 작업들의 공정시간을 q_i/a_i 과 q_i/b_i 사이의 어떤 값으로 고정시킬 것인지 결정하여야 한다. 본 연구에서는 q_i/b_i 를 채택하였는데 납기가 늦게 잡혀있다면 탄력성 있게 좀더 크게 잡는 방법 등에 대한 분석이 요구된다. 두 번째 EDD 규칙은 기존의 방법과 동일하다. 세 번째로 수정납기 규칙(MDD)은 다음 식에 의해 구해진 납기의 올림차순으로 정리한다.

$$md_i = \max\{T_k + q_j/b_i, d_i\}, \forall j \in R_t \quad (3)$$

이 방법은 최적 선행관계를 보장할 수 없지만 진행 시점 T_k 에서 고려되는 작업들

(Rt)간의 선행관계를 파악할 수 있기 때문에 배치 집합을 생성시키는데 이용하였다. 네 번째 가변성(VPS)은 단일기계일정계획에서 사용되는 긴급률과 흡사하다. 단위 작업들은 최대 작업 속도로 진행될 때와 최소의 작업 속도로 진행될 때에 걸리는 시간적 차이가 발생하는데 이들 간의 차이가 작을수록 빨리 처리하는 것이 유리할 수 있다. 그만큼 다른 작업들과 묶음을 형성할 여유가 작아진다고 볼 수 있다. 이러한 개념을 정량화하기 위해 가변성을 정의하여 긴급률 규칙의 가중치로 사용한다. 따라서 식 (4)의 값이 작은 순으로 우선순위를 정한다.

$$\frac{q_i/b_i - T_k}{q_i/a_i - q_i/b_i} \tag{4}$$

이상의 방법들을 통해 나열된 작업들은 몇 개의 묶음 집합들을 생성시킨다. 정해진 개수 (r)만큼 B_t^j ($j = 1, 2, \dots, r$)를 만들어 각 묶음 집합들의 배치여유를 계산한다.

$$BS_j = \sum_{i \in B_t^j} \{d_i - BP_j\} \text{ where } BP_j = q_i/b_{min} + T_k, b_{min} = \min\{b_i, i \in B_t^j\} \tag{5}$$

배치여유는 묶음에 소속된 작업들이 BP_j시점에 가공을 시작한다고 가정했을 때 각각의 작업들이 납기에 접근한 정도의 합을 나타낸다. 배치 집합중 여유가 작은 묶음을 선택하여 일괄공정시간을 산정하고 마지막 단계로 진행한다. 이 때 선택된 배치집합이 B_{st}이라면 공정시간은 아래와 같이 계산된다.

$$BP_s = \max\{q_i/b_i, i \in B_t^s\} \tag{6}$$

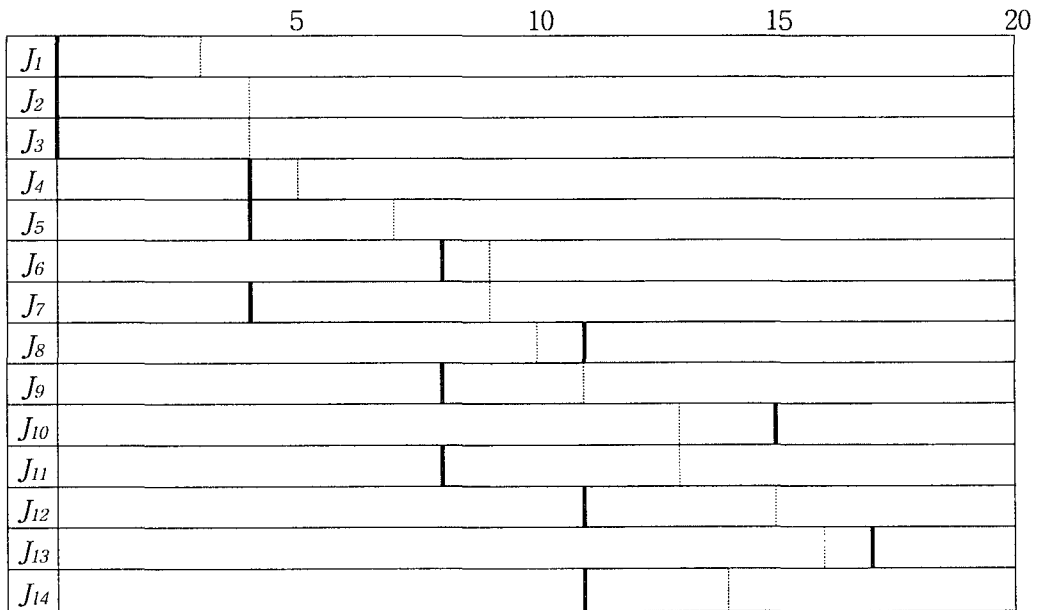
4. 수치 예제

14개의 단위 작업을 일괄 처리하는 문제(<표 1>)의 해를 MDD선택 기준으로 풀어 보면 매회의 배치집합으로 {J1, J2, J3 }, {J4, J5, J7 }, {J6, J9, J11 }, {J8, J12, J14 }, {J10 }, {J13 }이 얻어지고 총 납기지연은 6이 된다. 작업들이 처리되는 과정을 나타내면 <그림 3>와 같다.

<표 1> 수치예제

	a_i	b_i	q_i	d_i		a_i	b_i	q_i	d_i
1	4	6	15	3	8	5	6	24	14
2	3	5	10	4	9	4	7	20	11
3	5	6	20	4	10	8	9	18	16
4	3	6	12	5	11	7	10	10	13
5	5	7	12	7	12	6	9	15	15
6	7	9	21	9	13	5	7	18	21
7	6	8	24	9	14	4	6	18	14

$C=3, |R_t|=6$



굵은실선 : 시작지점, 점선 : 납기, 음영 : 공정

<그림 3> 수치예제의 결과

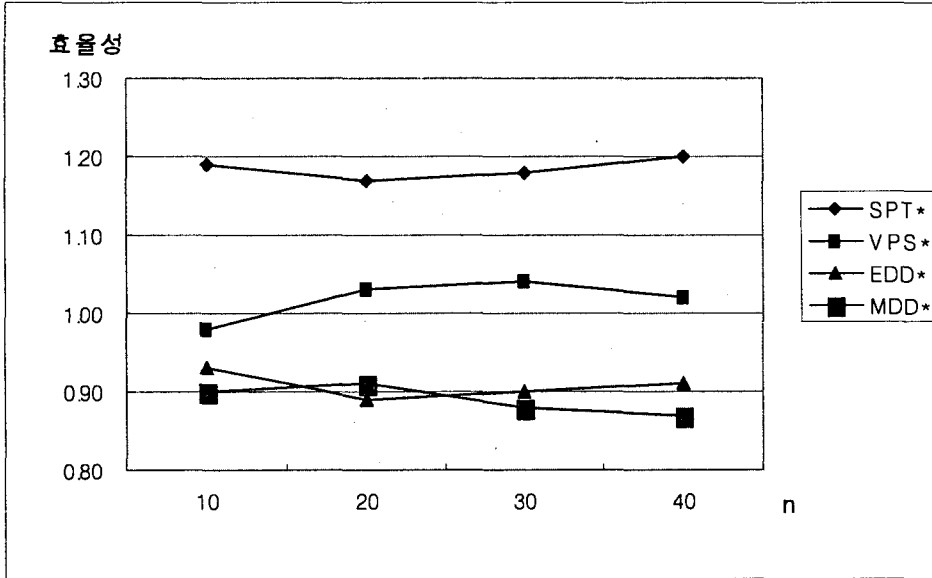
앞 절에서 제시한 순서결정규칙들을 비교하기 위해 단위 작업들의 개수가 각각 10, 20, 30, 40으로 구성된 문제를 20개씩 만든 다음 각각의 규칙들을 적용시켜 총 납기지연시간을 구하였다. 매회 묶음집합을 만들기 위해 고려되는 작업들(R_t)의 개수를 크게 잡으면 보다 나은 해를 얻을 수 있는 반면 계산량은 증가할 것이다. 따라서 R_t 의 개수와 최적해의 접근근성에 대한 연구가 필요하다. 본 연구에서는 R_t 를 최대 2C로 잡고 순서결정규칙을 적용하였을 때의 총납기지연시간을 계산하여 상대적 효율성을 비교하였다. 각 문제에 각각의 규칙으로 적용시켜 총납기지연과 총납기지연의 평균을 구한 다음 총납기지연을 평균으로 재산정한 값들을 <표 2>와 같이 얻었다.

<표 2> 효율성 산정치

n	SPT*	VPS*	EDD*	MDD*
10	1.19	0.98	0.93	0.90
20	1.17	1.03	0.89	0.91
30	1.18	1.04	0.90	0.88
40	1.20	1.02	0.91	0.87

$$MDD^* = 4MDT / (MDD + EDD + VPS + SPT)$$

효율성 산정치들을 그림으로 나타내면 <그림 4>와 같다.



<그림 4> 순서 결정 규칙들 간의 효율성 비교

상대적이긴 하지만 수치들이 작으면 작을수록 납기지연이 작은 해를 구할 수 있다는 것을 의미한다. SPT규칙과 EDD규칙의 성능은 수정납기규칙과 가변성규칙에 비해 효율성이 떨어진다. SPT규칙은 납기시점은 고려하지 않고 단위공정시간의 하한 값만 반영하였고 EDD규칙은 공정의 가변성을 전혀 고려를 하지 않았기 때문인 것으로 보인다. 한 결정기준을 이용하기 때문이다. 특히 SPT규칙은 공정시간의 하한값만을 단순 비교했기 때문에 효율이 현저히 떨어진다고 볼 수 있다. 그러므로 상한값과 하한값을 모두 고려한 기준식의 개발이 필요하다. 가변성규칙의 경우도 납기를 고려하지 못한 단점을 보완할 수 있도록 개선되어야 할 것이다.

4. 결론

본 연구는 공정 속도가 서로 다른 작업들이 일괄 처리되는 모형을 다루었다. 각각의 작업들은 특성상 일정 범위 내에서 공정의 속도를 조절할 수 있다. 조절 가능한 범위 내에서 속도를 공유할 수 있는 작업들을 묶어서 처리함으로써 총납기지연을 최소화시키는 발견적 해법을 개발하였다.

연구의 결과와 추후 연구 방향을 요약하면 다음과 같다.

첫째, 해법의 효율성은 배치 집합의 생성과 선택 방법에 의해 좌우되는데 이 과정에서 사용된 순서결정규칙에 대해 비교한 결과 EDD와 MDD규칙이 상대적으로 우수한 것으로 나타났다.

둘째, 속도의 조절은 납기의 충족과 비용간의 절충으로 이루어진다고 볼 수 있기 때문에 속도 관리에 대한 융통성을 확보할 필요가 있다. 따라서 일괄 공정속도에 대한 민감도 분석과 아울러 속도와 비용에 대한 항목을 목적함수에 추가시킨 모형에 대한 연구가 필요하다.

셋째, 발효 공정, 식물 재배과 같이 공정 속도의 완급을 조절할 수 있는 작업들로 구성된 시스템 관리에 활용될 수 있을 것이다.

마지막으로 총지연작업수, 총소요시간 등 다른 의사결정기준에 대한 분석이 요구된다.

5. 참 고 문 헌

- [1] Ahmed, I., and W.W. Fisher, "Due Date Assignment, Job Order Release and Sequencing," *Decision Sciences*, 23, 1992, pp.633~644.
- [2] K. R. Baker, *Elements of Sequencing and Scheduling*, Amos Tuck School of Business Administration, Dartmouth College, Hanover., 1998.
- [3] V. Chandru, C.Y. Lee and R. Uzsoy, "Minimizing Total Completion Time on Batch Processing Machine," *International Journal of Production Research* 31, 1993, pp.2097~2122.
- [4] T.C.E. Cheng, Q. Ding and B.M.T. Lin, "A concise survey of scheduling with time-dependent processing times," *European Journal of Operational Research*, 152, 2004, pp.1-13.
- [5] T.C.E. Cheng, Y. He, H. Hoogeveen, Min Ji and G. J. Woeginger, "Scheduling with step-improving processing times," *Operations Research Letters*, 34, 2006, pp.37-40.
- [6] G. R. Dobson and R. S. Nambimadom, "The Batch Loading and Scheduling Problem," *Oper. Res.*, 49, 2001, pp.52~65.
- [7] D. S. Hochbaum and D. Landy, "Scheduling semiconductor burn-in operations to minimize total flowtime," *Oper. Res.*, 45, 1997, pp.874~885.
- [8] C. Y. Lee, R. Uzsoy and L. A. Martin-Vega, "Efficient algorithms for scheduling semiconductor burn-in operations," *Oper. Res.*, 40, 1992, pp.764~775.
- [9] S. Webster and K. R. Baker, "Scheduling groups of jobs on a single machine", *Oper. Res.*, 43, 1995, pp.692~703.

저 자 소 개

오 세 호 : 현재 청주대학교 IT학부 교수로 재직 중이며 서울대에서 석사, 박사 학위를 취득하였으며 주요 관심분야는 OR, Scheduling 등이다.

이 근 부 : 현재 청주대학교 IT학부 교수로 재직 중이며 동아대에서 공학박사 학위를 취득하였으며 주요 관심분야는 작업방법의 최적화, CTD, 안전공학 등이다.

저 자 주 소

오 세 호 : 충북 청주시 상당구 내덕동 청주대학교 이공대학 산업공학과

이 근 부 : 충북 청주시 상당구 내덕동 청주대학교 이공대학 산업공학과