Journal of the Society of Korea Industrial and Systems Engineering
Vol, 27, No. 1, pp.47－50, March 2004.

구간 공정 시간을 갖는 작업들의 일괄처리 일정계획문제

오 세 호

청주대학교 산업정보시스템공학과

A Batch Scheduling Problem for Jobs with Interval-typed Processing Time

Se-Ho Oh

Department of Industrial & Information Systems Engineering, Chongju University

This paper deals with the problem of batching and scheduling of jobs whose processing times are different respectively. But, they are given as not the exact value but the range from the lower limits to the upper, which makes it possible to group jobs into batches. The grouping of jobs is desirable because of the capability of the batch processor to accommodate several jobs at once. The time required to process the jobs in any batch depends on their lower limit processing times. Once processing is initiated on a batch processor, the batch cannot be interrupted, nor can other jobs be started. And all jobs are assumed to be simultaneously available. This paper develops the model to describe these situation and a heuristic method to minimize its total tardiness.

Keywords : tardiness, batching, batch processor

1. 서 론

여러 개의 작업들이 기계의 제한된 능력의 범위 내에서 동시에 처리되는 일정계획문제를 일괄처리 문제라고 한다. 이러한 문제들의 중요한 특징인 일괄처리 공정은 제빵 작업, 도자기 열처리 공정, 반도체 제조과정, 유기물 숙성 공정 등에서 그 예를 찾아 볼 수 있다.[1][3][4] 지금까지 제시된 모형들과 해법들은 일괄처리 과정의 특성이 반영된 것이라고 볼 수 있다. Lee et al [5] 과 Candru et al [2]은 동시에 처리되는 작업들의 공정시간들 중에서 가장 큰 값이 일괄처리 공정시간이 되는 모형을 제시하였다. 본 연구에서는 단위 작업들의 공정시간은 다른데 제품의 품질특성을 만족시키기 위해서는 하한 값과 상한 값 범위 안에서 처리되어야 하는 조건을 충족시켜야 하는 모형을 제시하였다. 발효 숙성공정이나 열처리 공정 등에서 이러한 예를 찾아 볼 수 있을 것이다. 이렇듯 처리시간은 다르지만 일정 범위 안에서 유연하게 처리될 수 있다는 점에 착안하여 단위 작업들을 낱개로 처리하는 대신 몇 개씩 묶어 일괄 처리하는 접근 방법을 모색하였다. 본 연구에서 다루고자 하는 모형의 개별 작업들은 도착시간과 납기일이 주어져 있다. 모든 작업들은 시작 시점에 준비 가능한 것으로 가정하고 총납기지연의 최소화를 의사결정기준으로 하는 발견적 기법을 제시하고자 한다. 전형적인 일괄처리 문제들은 작업 능력이 자재취급에서의 단위적재처럼 해석될 수 있기 때문에 근본적으로는 적재문제(packing problem)와 같이 NP-hard 라고 보면 보다 구조적으로 복잡한 본 모형의 경우에도 NP-hard임은 자명하다.

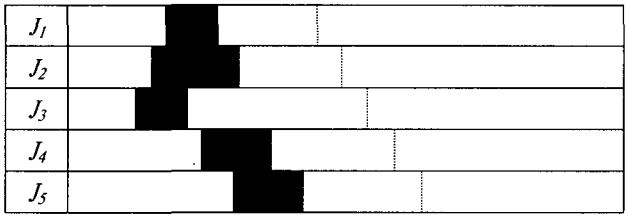
2. 문제의 정의

일괄처리 공정이 오븐을 통해 이루어지고 있다. 오븐은 처리능력 범위 내에서 여러 개의 작업을 동시에 처

리할 수 있다. 그런데 본 연구에서 다루고자 하는 문제는 고전적인 일괄처리작업 일정계획문제와는 달리 단위 작업들의 공정시간이 서로 다르다. 그렇지만 정해진 시간에 정확하게 작업이 끝나는 것이 아니라 일정 범위 안에서 이루어지기만 하면 된다. 즉 각 단위 작업들의 공정시간이 상하한 값 안에만 들어오면 원하는 가공이 이루어진다고 본다. 모형화를 위한 기본 가정들은 다음과 같다.

가정 1 : 오븐에서 일괄처리하기 위해 투입되는 작업의 개수는 오븐의 능력을 넘을 수 없다.
가정 2 : 일단 열처리 공정이 시작되면 새로운 작업이 추가될 수도 없고 제거될 수도 없다.
가정 3 : 각 단위 작업들의 처리는 최대, 최소 값의 범위 내에서 이루어진다.
가정 4 : 시작시점에 모든 작업들은 처리가 가능하다.
가정 5 : 주어진 납기를 지키지 못하면 납기지연이 발생하며 각 작업들의 납기지연의 비중은 동일하고 총 납기지연은 각 작업들의 납기지연의 합으로 산정된다.

<그림 1>은 5개의 단위작업을 시작 시점에 준비되었다고 가정하고 처리가능범위(빗금), 납기 시점(점선)을 나타낸 것이다.



<그림 1> 5개의 단위 작업의 예

위의 상황에서 오븐이 한번에 3개까지 처리할 수 있다면 J_1, J_2, J_3 를 한 묶음으로 J_1, J_2, J_4 와 J_2, J_4, J_5 를 각각 한 묶음으로 일괄 처리하는 방안이 가능하다. J_1, J_2, J_3 를 선택했다면 다음 단계에서는 J_4, J_5 이 일괄 처리되고 끝난다. 만약 J_1, J_2, J_4 를 선택한다면 다음 단계에서는 J_3 가 그 다음 단계에서는 J_5 가 선택되어 처리된다.

3. 해법의 개발

해법의 개발에 필요한 표기와 용어에 대한 정의는 다음과 같다.

T : (k-1)번째 일괄처리 작업후의 시점
J_i ($i = 1, 2,\ldots, N$) : 단위 작업(job)
LT_i, UT_i : J_i 의 열처리 시간의 하한, 상한값
DT_i : J_i 의 납기
MDT_i : 에서의 J_i 의 수정납기(modified due date)
C : 오븐의 일괄처리 능력
TD_T : 시점 T 까지의 총납기지연시간
BP_j : 일괄공정시간(batch processing time)
B_T^j ($j = 1, 2, \ldots, q$) : 시점 T 에서의 일괄처리(batching)가 가능한 작업들의 묶음들의 집합
FN : 처리된 작업들의 집합
R_T : T 에서 묶음 대상이 되는 작업들의 집합
PS_j : 묶음여유(batch slack)

납기지연(tardiness)은 작업의 완료시간과 비례하지 않기 때문에 납기지연 최소화 문제의 해법을 개발하기가 어렵다. 즉 조합최적화 문제의 일반적 접근 방법에 의존해야 아는 경우가 대부분이다. 전술했듯이 이러한 문제들은 NP-hard 이기 때문에 효율적인 최적 해법을 개발하기가 불가능하다.

아무튼 이러한 납기지연 문제에 대한 이론적 접근방법으로는 이웃짝교환 분석(adjacent pairwise interchange analysis)인데 본 연구에서 제시된 모형의 경우에는 작업과 작업간의 교환 분석보다는 작업 묶음과 묶음간의 분석이 요구된다. 그렇지만 이것도 작업들 간의 분석에 바탕을 둘 수 있기 때문에 단위 작업들 간의 관계부터 살펴보자.

정의 : 두 모수 집합 m_i,와 n_j가 $m_i < m_j$ 이면 $n_i \le n_j$ 를 의미한다면 두 모수 집합은 agreeable하다고 한다.

두 모수 집합을 공정시간과 납기라고 한다면 다음의 정리가 성립한다.

정리 1. 어떠한 단위 작업 쌍에 대해서도 공정시간들과 납기들이 agreeable이면 총납기지연은 SPT 순서(shortest processing time sequencing) 혹은 EDD 순서 (earliest due date sequencing)에 의해 최소화된다.

증명.[1].

본 연구에서 제시된 모형에서 단위 작업의 공정시간으로 하한값(LT_i)을 잡으면 다음의 따름정리가 성립한다.

따름정리 2. LT_i 와 DT_i가 agreeable 이면 총납기지연은 SPT 순서(shortest processing time sequencing) 혹은 EDD 순서 (earliest due date sequencing)에 의해 최소화된다.

증명. 각 작업들을 LT_i 에 완료하는 대안이 그렇지 않은 대안을 지배한다. 그러므로 LT_i를 공정시간으로 잡으

면 정리1.에 의해 성립한다.

일괄처리기(batch processor)인 오븐의 처리 용량이 C라면 공정들을 낱개로 처리하는 것보다 가능한 한 묶음으로 처리하는 것이 후속되는 작업들의 납기지연을 줄일 가능성을 높일 것이다. 따라서 다음 정리가 성립한다.

정리 3. 일괄처리기의 용량이 C라 할 때 어떠한 단위작업 쌍에 대해서도 LT_i 와 DT_i가 agreeable이면 순서결정기준이 총납기지연 최소화라 할 때 SPT 순서(shortest processing time sequencing) 혹은 EDD 순서 (earliest due date sequencing)에 의해 결정된 해는 적어도 묶음처리할 수 있는 경우의 해에 의해 지배된다.

증명. C=1인 경우;따름정리 2.에 의해 자명하다.
$C \geq 2$인 경우;EDD순서로 나열된 작업을 $J_1, J_2,...,J_n$ 이라 하자. J_i (i = 1, 2,..., n-1)와 묶여 처리되면서 일괄처리 공정시간이 LT_i를 넘지 않는 작업을 찾는다. 존재하지 않으면 C=1인 경우에 해당된다. 존재하면 J_k ($k \geq i$, k=2,..., n) 라 하고, 납기를 악화시키지 않고 J_i 와 묶어 처리할 수 있다. 동시에 J_k 에 의해 발생될 납기지연이 없어진다. 그러므로 일괄처리하는 해에 의해 지배된다.

위의 정리는 모든 작업 쌍에 대해서 LT_i 와 DT_i가 agreeable이어야 적용할 수 있지만 후속 묶음작업들을 가려내는 시점에서 부분적으로 응용할 수 있다.

실제로 J_i와 J_j간에 식 (1)이 성립할 때는 J_j가 선행하게 된다.

$$T + p_i > DT_i \quad \cdots\cdots (1)$$

위 식은 T에 좌우되기 때문에 두 작업들 간의 선행관계가 확정적이지 못하다. 즉 T시점에서의 선행관계가 최적해와 다르게 나타날 수 있다. 이와 같이 최적해에서와 같은 선행관계를 보장할 수 없지만 T시점에서 고려되는 작업들 간의 선행관계를 얻어낼 수 있으므로 발견적 해법을 구현하는데 이용할 수 있다.

발견적해법의 절차

단계 0 : 초기화 T = 0 , FN = ∅
단계 1 : $kard(FN) = N$ 이면 종료
MDT_i = max { $T + LT_i$, DT_i } $\forall i \notin FN$
올림차순으로 정리하여 kC개를 R_T에 넣음
B^j_T (j = 1, 2,....,q) 생성
단계 2 : B^j_T (j = 1, 2,....,q) 중 묶음
여유(BS_j)를 최소로 하는 것을 선택
Tie가 발생하면 일괄처리 공정 시간
(BP_j)이 작은 것을 선택
T 수정
TD_T 수정
FN 수정하고 단계 1로

해의 탐구의 정확성과 효율성은 B^j_T 의 생성과 선택기준에 달려 있다고 볼 수 있다. 이 B^j_T 는 일괄처리 공정능력의 정수배(k) 내에서 작업들(R_T)을 선택한 다음 이들간의 묶음 가능한 집합들로 만들어 진다. B^j_T의 생성절차와 선택 기준은 다음과 같다.

B^j_T 의 생성 절차

단계 1 : R_T 의 작업들을 SPT 규칙에 의해 재배열한다.
단계 2 : 일괄 처리 가능한 작업들로 용량의 범위내에서 묶어 B^j_T (j =1,2,....,q)를 생성한다.

B^j_T 선택 기준을 설정하기 위해 식(2)와 같이 묶음여유(batch slack)를 정의하였다.

$$BS_i = \sum_{i \in B_T} \{DT_i - BP_j\}$$

where $BP_j = max\{LT_i, i \in B^j_T\}$ ····· (2)

이 식에서 BP_j는 일괄처리 공정시간(batch processing time)을 의미한다. 묶음 여유 값은 각 묶음들을 T 시점에 가공한다고 가정하였을 때 이 묶음에 속해있는 각각의 작업들이 납기에 접근한 정도를 나타낸다.

4. 수치예제

다음과 같이 12개의 단위작업을 일괄처리하는 문제를 앞절의 방법으로 해를 구해보자

<표 1> 수치예제

		납기(DTi)	LTi	UTi	
1		3	2	3	
2		5	3	5	
3		10	1	2	
4		4	2	4	
5		7	1	3	
6		10	4	6	오븐의 처리능력(C) = 3
7		7	4	7	
8		14	5	8	
9		8	3	5	
10		11	2	4	
11		14	3	5	
12		14	6	9	

풀이 결과는 다음 <표2>와 같다.

<표 2> 수치예제의 풀이과정

회	단계	B_T^j	BS_j	BP_j	TD	T
	0	$\varnothing$			0	0
1	1	{1,4,2}{1,4,5} {1,4,9}{1,2,5} {1,2,9}{1,5,9}				
	2	B_T^1={1,4,2}선택	BS_1=3	BP_1=3	0	3
2	1	{5,9,10}{5,3,10} {7,9,6}{7,9,10}				
	2	B_T^3={7,9,6}선택	BS_3=4	BP_3=4	0	7
3	1	{5,3,10}{5,10,11} {11, 8}{8,12}				
	2	B_T^1={5,3,10}선택	BS_1=1	BP_1=2	2	9
4	1	{11,8}{8,12}				
	2	B_T^1={11,8}선택	BS_1=0	BP_1=5	2	14
5	1	{12}				
	2	B_T^1={12}선택	BS_1=-6	BP_1=6	8	20

해법을 적용한 결과는 {1,4,2} {7,9,6} {5,3,10} {11,8} {12}로 묶음집합을 구성하여 처리하면 총 납기지연은 8이 된다.

5. 결 론

본 연구에서는 총납기 지연을 최소화시키는 일괄처리 문제를 다루었다. 각각의 단위작업들이 서로 다른 공정시간을 갖는 문제를 일괄처리 접근방법으로 해법의 개발을 시도하였다.

연구의 결과 다음과 같은 활용 방안과 추후 연구 방향을 요약해 볼 수 있을 것이다.

첫째, 공정시간이 일정한 범위로 주어지는 모형에 활용된다.

둘째, 총납기지연뿐만 아니라 일괄처리 횟수나 총 작업소요시간에 비례하는 비용 함수식을 다루는 모형으로 확장시키면 다각도로 목적함수에서의 해법과 해의 의미부여가 가능하다.

셋째, 일괄처리는 묶음집합을 생성하고 선택하는 절차를 좀더 정교하게 개선해 나간다면 보다 효율적인 해법을 만들 수가 있고

넷째, 최적해법의 개발을 위한 이론적인 단서 제공하여 최적해법의 개발도 가능하다고 본다.

참고문헌

[1] K. R. Baker, *Elements of Sequencing and Scheduling*, Amos Tuck School of Business Administration, Dartmouth College, Hanover., 1998

[2] V. Chandru, C.Y. Lee and R. Uzsoy, "Minimizing Total Completion Time on Batch Processing Machine," *International Journal of Production Research* 31, pp2097~2122, 1993.

[3] G. R. Dobson and R. S. Nambimadom, "The batch loading and scheduling problem," *Oper.Res.*, 49, pp.52~65, 2001.

[4] D. S. Hochbaum and D. Landy, "Scheduling semiconductor burn-in operations to minimize total flowtime," *Oper. Res.*, 45, pp.874~885, 1997.

[5] C. Y. Lee, R. Uzsoy and L. A. Martin-Vega, "Efficient algorithms for scheduling semiconductor burn-in operations," *Oper. Res.*, 40, pp.764~775, 1992.

[6] S. Webster and K. R. Baker, "Scheduling groups of jobs on a single machine", *Oper. Res.*, 43, pp.692~703, 1995