병렬 기계에 있어서 공통 납기 결정과 일정 계획
—Common Due-Data Determination and Sequencing on Parallel Processors—

오 명 진*
이 상 도**

Abstract

This paper considers scheduling a set of n-jobs on m-paraller identical processors in which all jobs have the common due date.

The objective of the problem is to minimize the weighted mean absolute deviation of job completion times about such common due dates under the assumption that each job has a different weight, and to determine the optimal value of a common due date. We propose four heuristic solution methods based on several dominance conditions, and its solution procedure is illustrated with numerical examples.

The performance comparison is made among four heuristic scheduling procedures.

1. 서 론

스케듈링 이론은 1950년대 2기계 플로우샵(flow shop) 문제에 대한 Johnson과 단일 기계의 최대납기차 최소화 문제에 대한 Jackson의 성공으로 발단하여 최근 스케듈링 문제의 관심이 점점 고조되어 그 이론적 연구도 광범위하게 행하여져 많은 이론적 해법을 주고 있다.

그러나 실제의 생산 현장에서는 수많은 종류와 수량의 생산설비가 있으며 처리해야 할 가공의 종류도 많고 또한 시간적으로 동태성이 있어 작업시간도 확률적으로 변동한다. 이러한 현실 상황하에서 일정계획의 최적화를 해석적으로 행하는 것은 불가능에 가까운 것이다. 그러므로 일정계획에 있어서 시뮬레이션이나 발견적 기법 등이 많이 사용되고 있다. 또 지금까지는 단일기계에 대한 문제나 다수기계에 대한 문제라 할지라도 시간에 관한 문제만을 주로 연구했는데 실제로는 각 상황에 따라 비중이 다르므로 각각의 비용이 다른 특정한 경우에 대해서 연구가 필요하다.

공통납기(common due date)를 갖는 작업들(jobs)의 완료시간의 WMAD(Weighted Mean Absolute Deviation)를 최소화하는 일정계획 문제는 편차에 대한 수행척도로써 컴퓨터 시스템의 화일조직에 있어서 온라인 시스템의 반응시간의 편차를 최소화하는 문제, 적정재고관리문제 및 부품이 공급되는 MRP(Materials Requirements Planning)에 응용될 수 있다. 특히 이 문제는 JIT(Just-In-Time) 생산방식을 채택하고 있는 자동차 조립공장에 있어서는 매우 중요하다[3].

종래의 연구에서는 납기는 보통 외부적인 제한이기 때문에 납기지연 비용은 명백하지만 조기완료 비용은 간접적 특성을 가지므로 비효율적인 비용이라 하여 작업의 조기 완료 비용은 무시해 왔다. 그러나 실제적인 입장에서는 조기 완료 비용은 무시할 수 없다. 왜냐하면 조기완료된 작업들은 보다 더 많은 재고비용을 초래할 수 있고 제품이 진부성을 갖는 경우는 주문 취소로 생산환경의 변화요인으로 과다 재고보유의 위험을 증가시킨다.

이러한 두 비용의 불합리한 현실적 상황하에서 Sidney[11]는 제한된 자원 프로젝트 문제와 부패하기 쉬운 상품을 생산할 경우에 대해서 다루었고, Kanet[8]는 작업들이 가중치(벌과금)를 같은 비율로 두고 MAL(Mean Absolute Lateness)의 합을 최소화하는 알고리즘을 제시하였다. Hall[7]은 동일한 가중치가 부과되는 상황하에서 단일기계와 복수기계의 완료시간 편차를 최소화하는 문제를 다루었으며, Kanet[8]의 보순되는 점

---

*경남전문대학 공업경영과 부교수
**동아대학교 산업공학과 교수
접수 : 1991. 4. 27.

을 보완하여 해의 확장성을 나타내었다. Eilon and Chowdhury[4]는 완료시간 분산을 최소화하는 문제는 V형이 되어야 하는 것을 보여주고 Sundararaghavan and Ahmed[12]는 m-병렬기계문제로 확장하여 근사적인 방법을 취급하였다. Emmons[6]는 조기완료와 납기지연에 따라서 서로 다른 벌과금이 부과되는 경우의 공통납기를 갖는 복수기계 스케듈링 문제를 다루었다.

따라서 본 연구에서는 각 작업들은 공통납기를 갖는 상황하에서 모든 작업들이 뚜렷한 서로 다른 가중치를 갖는 경우에 병렬기계에 있어서 작업의 완료시간의 WMAD를 최소화 하는 최적 스케듈의 모형을 정식화 하고 이 문제에 대한 우월 특성을 기초로 하여 4가지의 발견적기법(heuristic method)을 제시하고 또한 최적공통납기를 결정한다. 이 4가지 발견적 기법에 대한 수치예를 제시하고 수치실험을 통하여 해의 분석과 유효성을 보이고자 한다.

2. 모형화

2.1 기호 설명

각 기호에 대한 설명은 다음과 같다.

N : n개 독립 작업들의 집합, N={1, 2, ……, n}

M : m개 병렬기계의 집합, M={1, 2, ……, m}

P_j : 작업 j의 가공시간

C_j : 작업 j의 완료시간

d : 공통납기

B : 조기완료된 작업 집합, $\{j \mid C_j \leq d\}$

A : 납기지연된 작업 집합, $\{j \mid C_j > d\}$

W_j : 작업 j의 가중치

MS : $\Sigma_j \in_N P_j$(job의 makespan)

S : 집합 N에서 임의의 n 작업들의 순서

U : 처리순서가 미정인 작업들의 집합

$|B|$: B 집합에서 작업의 갯수

$|A|$: A 집합에서 작업의 갯수

[j] : 집합 B에서의 j번째 집합

(j) : 집합 A에서의 j번째 집합

F : 전체 작업의 완료시간

2.2 가정

본 연구의 모형에 대한 가정은 다음과 같다.

1) 모든 작업들의 작업시간은 미리 정해진다.
2) 모든 작업은 병렬 복수기계로 가공한다.
3) 각 작업은 준비시간과 대기시간은 고려하지 않는다. (즉, 가공시간에 포함)
4) 작업은 기계 한대에 하나씩 처리한다.
5) 작업들 간에는 상호독립이다.
6) 일단 어떤 작업이 가공을 시작하면 가공이 완료될 때까지 다른 작업을 가공할 수 없다.
7) 각 작업은 공통 납기일을 갖는다.

2.3 목적 함수

본 연구의 목적함수는 병렬기계에 있어서 공통납기에 대한 작업들의 WMAD를 최소화한다. 목적식은 다음을 최소화하는 스케듈 S를 구하는 것이다.

$$Z(S)=\frac{\Sigma_i \in_M [\Sigma_j \in_N W_j \mid C_j - d \mid]}{\Sigma_j \in_N W_j} \tag{1}$$

목적함수 (1)은 다음과 같이 다시 표현할 수 있다.

$$Z(S)=\frac{\Sigma_i \in_M [\Sigma_j \in_N W_{ji}((j-1)P_{[ji]}+jP_{(ji)})]}{\Sigma_j \in_N W_j} \qquad (2)$$

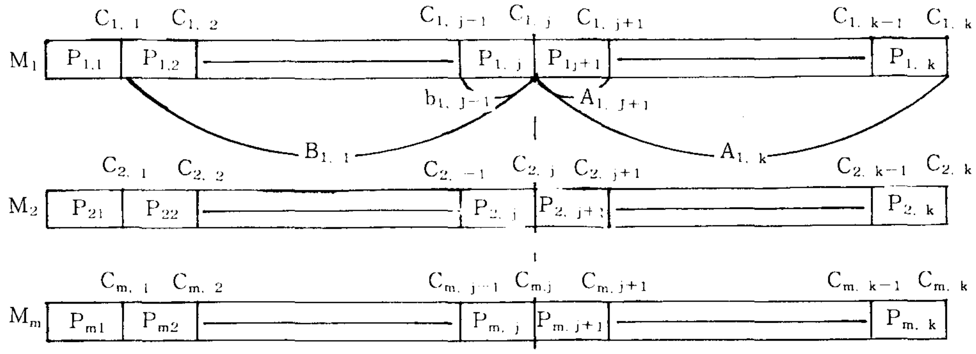


Fig. 1 schedule of model

순서목적함수는 한 집합이 작업 파라메터들로 결정되는 두개의 같은 집합의 쌍의 합으로 모형화 한 수 있다 (Fig. 1 참조).

식 (1)에서 d는 d 이전에 작업들을 일정하는데 있어서 충분한 자유를 줄 수 있도록 큰 값을 가진다. 즉 $d \geq MS$이다.

WMAD 문제는 W_j가 1인 경우에는 MAD(Mean Absolute Deviation) 문제가 되므로 MAD보다 각 작업에 대해 매단위 시간비용이 다르기 때문에 더 큰 유연성을 제공할 수 있다.

이들의 공통납기일과 전체 작업의 완료시간은 다음과 같다.

$$d^* = \max_i \{\Sigma_j \in_B P_{[ji]}\}$$
$$F = d^* + \max_i \{\Sigma_j \in_A P_{(ji)}\}$$

3. 발견적 기법의 특성과 해법

3.1 우월특성(Dominance Properties)

WMAD 문제는 NP-complete로 알려진 WMF(Weighted Mean flow time) 문제만큼 최소한 hard 문제로써 NP-complete 문제이다[11]. 다음과 같은 경우에는 다항시간(polynomial time)으로 찾을 수 있다.

1) 각 작업들에 대한 가중치가 동일하다.

2) 조기완료와 납기지연된 작업이 서로 다른 가중치를 갖는다.

3) 각 작업들에 대한 가중치는 서로 다르나, 가공시산이 동일하다.

공통납기를 고려한 병렬복수기계의 일정계획에 대한 WMAD 문제를 최소화하는 해법의 개발을 위한 우월한 특성은 다음과 같은 성질로 주어진다.

[정의 1]

최적납기 d^*는 다음과 같다.

$$0 \leq d^* \leq F$$

[정리 1]

최적스케쥴에서 어떤 두개 작업들 사이에 여유시간(idle time)은 존재하지 않는다.

증명 : d 이전 혹은 d 이후에 idle time이 삽입된 어떤 스케쥴 S를 생각하자.

만일 여유시간이 d 이전(이후)에 일어났다고 하면 스케듈에서 여유시간 앞으로(뒤로) 이동함으로써 이유시간은 제거된다. 모든 작업들의 완료시간은 d에 가까이 접근하므로써 S스케듈은 개량되어진다.

[정리 2]
어떤 특정한 작업순서 S에 대해서 납기전에 작업시작된 잡스케듈은 납기후에 완료되면 최적이 될 수 없다. 즉, 모든 최적스케듈에서 하나의 작업 완료시간과 납기는 일치해야 한다.

증명 : 吳明鎭, 崔鍾德 [1]을 참조.

[정리 3]
만일 B에서 작업들이 WLPT(Weighted Largest Processing Time) 순서이면 스케듈은 우월하다.

$$\frac{P_{[m,\ 1]}}{W_{[m,\ 1]}} \geq \frac{P_{[m,\ 2]}}{W_{[m,\ 2]}} \geq \cdots\cdots \geq \frac{P_{[m,\ j]}}{W_{[m,\ j]}}$$

역시 A에서 작업들이 WSPT(Weighted Shortest Processing Time) 순서에 있으면 스케듈은 우월하다.

$$\frac{P_{(m,\ j+1)}}{W_{(m,\ j+1)}} \leq \cdots\cdots \leq \frac{P_{(m,\ k-1)}}{W_{(m,\ k-1)}} \leq \frac{P_{(m,\ k)}}{W_{(m,\ k)}}$$

증명 : 吳明鎭, 崔鍾德 [1]을 참조.

[정리 4]
작업들의 순서중에서 P_j/W_j가 가장 큰 작업을 첫번째 가공해야 한다.

증명 : [정리 3]을 만족하고 [정리 4]는 만족하지 않는 스케듈 S를 생각하자.
그러면 P_k/W_k가 가장 큰 Job은 순서에서 마지막 위치에 있어야 한다. S에서 첫번째와 마지막 작업들을 교체함으로써 스케듈이 개량되는 것을 볼 수 있다.

가중치들이 같은 경우는(즉 W=1) LPT 순서를 갖는 m−jobs의 i번째 집합을 Si로 정의하면 즉, $S_1=\{1, 2, \cdots\cdots, m\}$, $S_2=\{m+1, m+2, \cdots\cdots, 2m\}$, ……, 등 따라서 $B=S_1 \cup S_3 \cup S_5$, ……, $A=S_2 \cup S_4$, ……으로 나타낼 수 있다.

그러나 가중치가 다른 경우는 다른 선택이 필요하다. 즉 가중치 요인에 따라 최대−최소(largest-to-smallest) 짝으로 만나야 한다.

[정리 5]
m-jobs의 두 집합의 요소의 쌍을 나타내고 각 쌍을 합할 때 m합의 가장 큰 것을 최소화 하려면 최적쌍은 최대−최소(largest-to-smallest)의 짝으로 만나야 한다.

증명 : 한 집합 S_k에서 a_1과 $a_2(a_1>a_2)$가 다른 한 집합 S_{k+1}에 있는 b_1과 $b_2(b_1>b_2)$에 대해 쌍으로 만난다고 할 때 분명하게 다음과 같은 관계로 된다.
$\max\{a_1+b_1,\ a_2+b_2\} > \max\{a_1+b_2,\ a_2+b_2\}$ 따라서 쌍을 교체하므로써 단지 목적함수 값은 증가되지 않는다.

3.2 발견적 기법의 해법

작업들의 완료시간의 WMAD를 최소화하는 문제에 대한 최적스케듈을 얻는 것은 이론적으로는 가능하지만 실제에서는 계산시간이 과다하게 소요되므로 짧은 시간안에 비교적 쉽게 실현가능한 스케듈을 얻는 4가지 발견적 기법들을 제시한다.

〈발견적 기법 1(H1)〉
Step 1) n개의 작업들을 P_k/W_k의 내림차순으로 나열한다.
Step 2) P_k/W_k가 가장 큰 작업 m개를 선택하여 조기 완료된 작업 집합 B의 각 기계에 하나씩 할당한다.
Step 3) 만일 U≠0이면 U에서 P_k/W_k가 다음으로 가장 큰 작업 m개를 선택하여 지연된 집합 A의 첫번째 위치에 각 작업들을 최대−최소로 짝을 지워(largest-to-smallest) 하나씩 할당한다.

Step 4) $S \leftarrow (B, A)$

Step 5) $U \neq 0$이면 Step 2)로 간다.
$U = 0$이면 Step 6)으로 간다.

Step 6) 최적납기 $d^* = \max_i \{\Sigma_j \in_{B_i} P_{[ji]}\}$를 구한다.

Step 7) stop

〈발견적 기법 2(H2)〉

Step 1) n개의 작업들을 P_k/W_k의 내림차순으로 나열한다.

Step 2) P_k/W_k가 가장 큰 작업들을 m개 선택하여 각 기계의 조기 완료된 작업 집합 B에 하나씩 할당한다.

Step 3) 다음으로 P_k/W_k가 가장 큰 작업 K를 m개 선택하여

$\sum_{i=1}^{m} \sum_{j \in B} W_{ji}$와 $\sum_{i=1}^{m} \sum_{j \in A} W_{ji}$를 계산한다.

Step 4) a) $\sum_{i=1}^{m} \sum_{j \in B} W_{ji} < \sum_{i=1}^{m} \sum_{j \in A} W_{ji} + W_k$이면

각 기계의 조기완료 된 작업 집합 B에서 마지막 위치에 최대—최소로 짝을 지워(largest-to-smallest) 하나씩 할당한다.

b) $\sum_{i=1}^{m} \sum_{j \in B} W_{ji} > \sum_{i=1}^{m} \sum_{j \in A} W_{ji} + W_k$이면

각 기계의 지연된 작업 집합 A에서 첫번째 위치에 최대—최소로 짝을 지워 하나씩 할당한다.

Step 5) $U \neq 0$이면 Step 3)으로 가고
$U = 0$이면 Step 6)으로 간다.

Step 6) 최적납기 $d^* = \max_i \{\Sigma_j \in_{B_i} P_{[ji]}\}$를 구한다.

Step 7) stop

〈발견적 기법 3(H3)〉

Step 1) n개의 작업들을 P_k/W_k의 내림차순으로 나열한다.

Step 2) P_k/W_k가 가장 큰 작업 m개를 선택하여 조기 완리된 작업 집합 B의 각 기계에 하나씩 할당한다.

Step 3) $\alpha_t = 2$, $\beta_t = 1$로 초기화한다.

Step 4) $W = \min_t \{\Sigma_j \in_B W_j \ \alpha_t, \ \Sigma_j \in_B W_j \ \beta_t\}$로 둔다.

Step 5) 만일 $\Sigma_j \in_B W_j \alpha_t = W$인 t가 존재하면 각 기계의 조기 완료된 작업 집합의 다음 마지막 위치에 작업 m개를 최대—최소로 짝을 지워 하나씩 할당하고 α_t를 증분한다. 그렇치 않으면 $\Sigma_j \in_A W_j \beta_t = W$를 각 기계에 지연된 작업 집합의 첫번째 위치에 작업 m개를 최대—최소로 짝을 지워 하나씩 할당하고 β_t를 증분한다.

Step 6) 만일 $U = 0$이면 Step 7)로 가고,
그렇지 않으면 Step 4)로 간다.

Step 7) 최적납기 $d^* = \max_i \{\Sigma_j \in_{B_i} P_{[ji]}\}$를 구한다.

Step 8) Stop

〈발견적 기법 4(H4)〉

Step 1) n개의 작업들을 가공시간 P_k/W_k의 내림차순으로 나열한다.

Step 2) $B_i \leftarrow 0$, $i = 1, \cdots\cdots, n$
$A_i \leftarrow 0$, $i = 1, \cdots\cdots, n$

Step 3) P_k/W_k가 가장 큰 작업 m개를 선택하여 조기 완료된 집합 B의 각 기계에 하나씩 할당한다.

Step 4) $U \neq 0$이면 P_k/W_k가 다음으로 가장 큰 작업 m개를 선택하여 이동한다.

$Q \leftarrow \{t \mid t \in s, \ \mid B_t \mid\}$
$R \leftarrow \{t \mid t \in s, \ \mid A_t \mid\}$

Step 5) a) $|B_t|,\ _t\in_Q = |A_t|,\ _t\in_R$이면 $B_t,\ _t\in_R$의 마지막 위치에 작업 m개를 P_j의 최대－최소로 짝을 지워 하나씩 할당한다.

b) $|B_t|\ \ _t\in_Q = |A_t| + 2,\ _t\in_R$이면 $A_t,\ _t\in_R$의 첫번째 위치에 작업 m개를 P_j의 최대－최소로 짝을 지워 하나씩 할당한다.

Step 6) 그 외는 $W = \min\{\Sigma_i \Sigma_j \in_B P_{ji},\ \Sigma_i \Sigma_j \in_A P_{ji}\}$를 선택하여 $W = \Sigma_i \Sigma_j \in_B P_{ji}$이면 B의 마지막에 P_j의 최대－최소로 짝을 지워 하나씩 할당하고 그렇치 않으면 A의 첫번째 위치에 P_j의 최대－최소로 짝을 지워 하나씩 할당한다.

Step 7) U=0이면 최적납기 $d^* = \max_i\{\Sigma_j \in_{Bi} P_{(ji)}\}$를 구하고 Step 8)로 간다. 그렇지 않으면 Step 4)로 간다.

Step 8) Stop

4. 수치예제

13 작업을 3 기계에서 처리하는 병렬 기계 문제에 대한 제안된 발견적 기법을 예증하기 위한 데이터들은 Table 1과 같다.

Table 1 Example Data

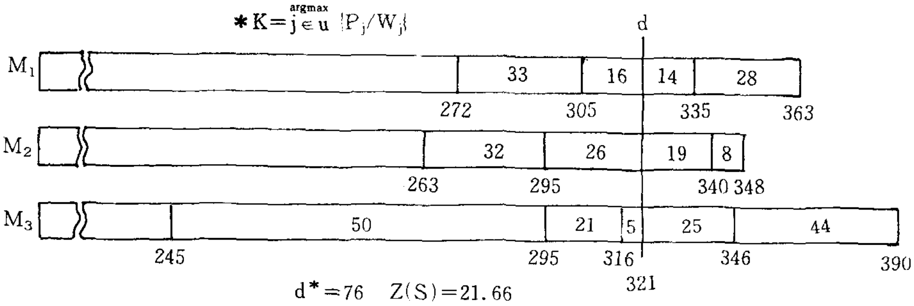
Job	1	2	3	4	5	6	7	8	9	10	11	12	13
P_j	16	19	26	5	21	32	50	44	8	28	14	33	25
Wj	4	7	4	5	3	2	5	6	1	3	6	2	8
P_j/W_j	4	2.71	6.5	1	7	16	10	7.33	8	9.33	2.33	16.5	3.13

$d = MS = \Sigma_j \in_N P_j = 321$

Table 2 Step-to-Step Results from Heuristic 1

Step	K*	기계	할당	순서	
				Early	Late
1	12	M_1	B	12	
2	6	M_2	B	6	
3	7	M_3	B	7	
4	10	M_1	A		10
5	9	M_2	A		9
6	8	M_3	A		8
7	5	M_1	B	12, 1	10
8	3	M_2	B	6, 3	9
9	1	M_3	B	7, 5	8
10	13	M_1	A	12, 1	11, 10
11	2	M_2	A	6, 3	2, 9
12	11	M_3	A	7, 5	13, 8
13	4	M_3	B	7, 5, 4	13, 8

$* K = \underset{j \in u}{\operatorname{argmax}} \{P_j/W_j\}$



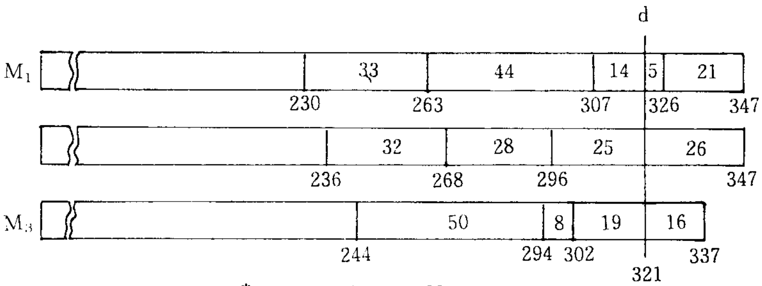
$d^* = 76 \quad Z(S) = 21.66$

Fig. 2 The schedule generated by H1

Table 3 Step-to-Step Results from Heuristics 2

Step	K*	$\sum_{j\in A} W_j$	$\sum_{j\in A} W_j+W_k$	기계	할당	순 서	
						Early	Late
1	12			M_1	B	12	
2	6	0	9	M_2	B	6	
3	7			M_3	B	7	
4	8			M_1	B	12, 8	
5	10	9	10	M_2	B	6, 10	
6	9			M_3	B	7, 9	
7	5			M_1	A	12, 8	5
8	3	19	11	M_2	A	6, 10	3
9	1			M_3	A	7, 9	1
10	13			M_1	B	12, 8, 11	5
11	2	19	32	M_2	B	6, 10, 13	3
12	11			M_3	B	7, 9, 2	1
13	4	40	16	M_1	A	12, 8, 11	4, 5

$*K = \overset{\text{argmax}}{j \in U} \{P_j/W_j\}$



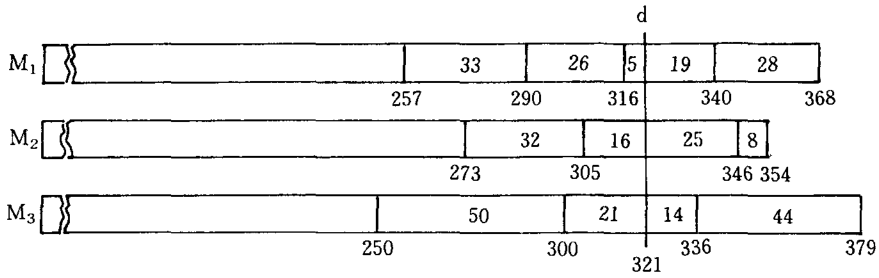
$d^* = 91 \quad Z(S) = 14.39$

Fig. 3 The schedule generated by H2

Table 4 Step-to-Step Results from Heuristic 3

Step	K*	α_t	β_t	기계	할당	순 서	
						Early	Late
1	12	1		M_1	B	12	
2	6	1		M_2	B	6	
3	7	1		M_3	B	7	
4	10	2	1	M_1	A	12	10
5	9	2	1	M_2	A	6	9
6	8	2	1	M_3	A	7	8
7	5	2	2	M_1	B	12, 3	10
8	3	2	2	M_2	B	6, 1	9
9	1	2	2	M_3	B	7, 5	8
10	13	3	2	M_1	A	12, 3	2, 10
11	2	3	2	M_2	A	6, 1	13, 9
12	11	3	2	M_3	A	7, 5	11, 8
13	4	3	2	M_1	A	12, 3, 4	2, 10

$*K = \overset{\text{argmax}}{j \in u} \{P_i/W_i\}$



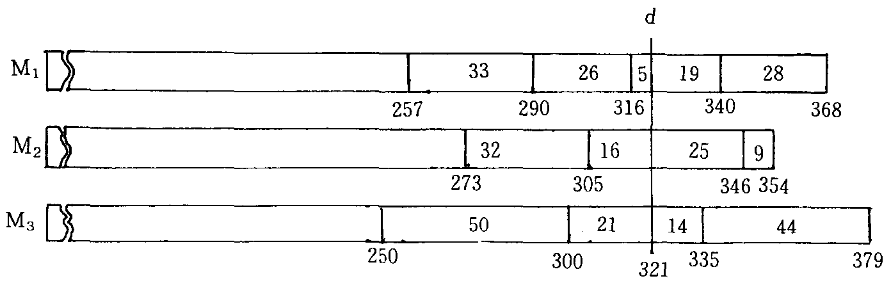
d*=71 Z(S)=20.68

Fig. 4 The schedule generated by H3

Table 5 Step-to-Step Results from Heuristic 4

Step	p*	기계	할당	순서 Early	서 Late
1	7	M_1	B	12	
2	8	M_2	B	6	
3	12	M_3	B	7	
4	6	M_1	A		10
5	10	M_2	A		9
6	3	M_3	A		8
7	13	M_1	B	12, 3	10
8	5	M_2	B	6, 1	9
9	2	M_3	B	7, 5	8
10	1	M_1	A	12, 3	2, 10
11	11	M_2	A	6, 1	13, 9
12	9	M_3	A	7, 5	11, 8
13	4	M_1	B	12, 3, 4	2, 10

$* p = \underset{j \in u}{\text{argmax}} \{P_j/W_j\}$



d*=66 Z(S)=20.32

Fig. 5 The schedule generated by H4

Table 1의 데이터에 대한 수치 예제에서 발견적 기법 2가 Z(S) 값이 14.39이고 d*=91로 다른 기법 보다도 우수한 것을 보여주고 있으나 유효성을 평가하기 위해 Table 6과 같은 데이터를 가지고 각 발견적 기법들에 적용하여 상호 비교하고자 한다.

Table 6 Random data

Data set	1	2	3	4	5	6
Processing time	U(1, 30)	U(1, 30)	U(1, 50)	U(1, 50)	U(1, 100)	U(1, 100)
Weight factor	U(1, 10)	U(1, 20)	U(1, 10)	U(1, 20)	U(1, 10)	U(1, 20)

가중치와 작업 시간은 각각 일양분포에서 랜덤으로 Table 6과 같이 추출한다. 단, 가중치와 작업 시간은 독립이다. 편의상 작업들은 7가지 집합으로 분류하고 기계 댓수는 3가지로 분류하여 평가하였다. 작업 시간과 기계 댓수는 다음과 같다.

작업집합 : 10, 15, 18, 20, 30, 50, 100

기계댓수 : 2, 3, 5

이 데이터 집합에 대한 수치 실험의 결과는 Table 7~9와 같다.

Table 7 Comparison of Heuristics

	DS_1 (30, 10)	DS_2 (30, 20)	DS_3 (50, 10)	DS_4 (50, 20)	DS_5 (100, 10)	DS_6 (100, 20)	계
H	1	0	2	1	0	0	4
H2	11	11	10	11	9	11	63
H3	3	2	0	2	4	4	15
H4	6	8	9	7	8	6	44

Table 8 Results of 126 test problems

	10	15	18	20	30	50	100	계
H1	1	2	1	0	0	0	0	4
H2	14	14	9	13	8	3	2	63
H3	0	0	0	0	2	5	8	15
H4	3	4	6	5	6	12	8	44

Table 9 Comparison of processors

	2	3	5	계
H1	0	3	1	4
H2	17	18	28	63
H3	10	3	2	15
H4	15	18	11	44

Table 8에서 보면 발견적 기법 2가 다른 기법 보다 우수한 스케듈을 생성하는 빈도가 많다. 기계 댓수가 증가하면 현저하게 좋게 나타나고 작업수가 증가하면 단일 기계에서는 발견적 기법 2가 현저하게 증가하지만[1] 병렬 복수 기계에서는 작업수가 증가하면 다른 발견적 기법(발견적 기법 4)이 좋아지는 경향이 있다. 발견적 기법 2는 가중치 요소를 다른 기법 보다 더 강조하였고 발견적 기법 4는 작업 시간요소에 대해 더 강조 하였다.

5. 결　론

본 연구에서는 병렬 복수 기계에 대한 작업이 공통 납기를 갖는 경우에 WMAD를 최소화 하는 일정계획에 관한 문제를 고찰하였다. 이 문제는 NP-Complete 문제이기 때문에 최적 스케듈을 구하는 것은 이론적으로는 가능하지만 실제로 계산 시간이 과다하게 소요되기 때문에 비교적 짧은 시간으로 간단하게 실행 가능한 4가지 발견적 기법들을 제시하여 비교 분석 하였고 또한 d^*를 구하는 방법을 보여 줌으로써 납기 결정에 중요한 기초 자료를 제공한다.

비교 분석의 결과 가중치 요소에 더 강조한 발견적 기법 2가 우수한 빈도로 나타나고 기계 댓수가 증가하면 현저하게 증가한다. 단일 기계에 있어서는 작업수가 증가하면 현저하게 우수하지만 복수 기계에서는 작업수가 증가하면 작업 시간을 강조한 발견적 기법 4가 우수하게 나타난다.

본 연구에서는 공통납기가 1개 있는 경우이지만, 앞으로 연구 과제로써 공통 납기가 복수개 있는 경우로 확장하여 연구하고자 한다.

참 고 문 헌

1. 吳明鎭 · 崔鍾德, "공통납기에 대한 완료시간의 W. M. A. D. 최소화에 관한 연구," 韓國工業經營學會, 13(21), 143-151, 1990.
2. Bagchi, U., Chang, Y. L., and Sullivan, R. S., "Minimizing Absolute and Squared Deviations of Completion Times with Different Earliness and Tardiness Penalties and a Common Due Data," *Naval Research Logistics Quarterly, 34*, pp. 739-751, 1987.
3. Bagchi, U., Sullivan, R. S., and Chang, Y. L., "Minimizing Mean Absolute Deviation of Completion Times about a Common Dlue Date," *Naval Research Logistics Quarterly, 33*, pp. 227-240, 1986.
4. Baker, K. R., *Introduction to Sequencing and Scheduling*, John Wiley & Sons, New York, 1974.
5. Eilon, S., and Chowdhury, I. G., "Minimizing Waiting Time Variance in the Single Machine Problem," *Management Science, 23*, 567-575, 1977.
6. Emmons, H., "Scheduling to a Common Due Date on Parallel Uniform Processors," *Naval Research Logistics Quarterly, 34*, pp. 803-810, 1987.
7. Hall, N. G., "Single and Multiple-Processor Models for Minimizing Completion Time Variance," *Naval Research Logistics Quarterly, 33*, pp. 49-54, 1986.
8. Kanet, J. J., "Minimizing the Average Deviation of Job Completion Times About a Common Due Date," *Naval Research Logistics Quarterly, 28*, pp. 643-651, 1981.
9. Kanet, J. J., "Minimizing Variation of Flow Time in Single Machine Systems," *Management Science, 27*, pp. 1453-1459, 1981.
10. Panwalkar, S. S., Smith, M. L., and Seidmann, A., "Common Due Date Assignment to Minimize Total Penalty for the One Machine Scheduling Problem," *Operations Research, 30*, pp. 391-399, 1982.
11. Sidney, J. B., "Single Machine Scheduling with Earliness and Tardiness Penalties," *Operations Research, 25(1)*, pp. 62-69, 1977.
12. Sundararaghavan, P. S., and Ahmed, M. U., "Minimizing the Sum of Absolute Lateness in Single Machine and Multimachine Scheduling," *Naval Research Logistics Quarterly, 31*, pp. 325-333, 1984.
13. Szwarc, W., "Single-Machine Scheduling to Minimize Absolute Deviation of Completion Times from a Common Due Data," *Naval Research Logistics Quarterly, 36*, pp. 663-673, 1989.