공통납기에 대한 완료시간의 W.M.A.D. 최소화에 관한 연구
—Minimizing the Weighted Mean Absolute Deviation of Completion Times about a Common Due Date—

吳 明 鎭*
崔 鍾 德**

Abstract

This paper studies a single machine scheduling problem in which all jobs have the common due date and penalties are assessed for jobs at different rates. The scheduling objective is to minimize the weighted mean absolute deviations(WMAD). This problem may provide greater flexibility in achieving scheduling objectives than the mean absolute deviation (MAD) problem. We propose three heuristic solution methods based on several dominance conditions. Numerical examples are presented. This article extends the results to the problem to the problem of scheduling n-jobs on m-parallel identical processors in order to minimize the weighted mean absoulte deviation.

1. 서 론

단일기계 일정계획문제(Single Machine Scheduling Problem)는 단일의 생산설비에서 많은 량의 job을 처리하는 문제이고 각 job에는 처리시간과 납기가 있어, 수행측도 적합한 job의 처리순서를 구하는 문제이다.

공통납기(common due date)를 갖는 jobs의 완료시간의 WMAD(Weighted Mean Absolute Deviation)를 최소화하는 일정획문제는 편차에 대한 수행측도로써 많은 응용에서 중요하다.

이 문제는 Just-In-Time 생산방식을 채택하고 있는 자동차 조립공장에서 납기와 관련해 납기차 문제의 적정 재고관리 문제 및 사용자의 시간에 맞추어 부품이 공급되는 MRP(Materials Requirements Planning)에 응용될 수 있으며, Computer System의 화일조직에 있어서 on-line system에 의한 반응시간의 편차를 최소로 하여 이용자에게 일양반응시간을 제공하는 써비스문제, 부패하기 쉬운 상품의 생산시스템 및 제조조립공장에서 순서문제에 중요하게 응용될 수 있다.

종래의 연구에서는 수행측도가 tardiness에 기준을 둔 average tardiness, maximum tardiness, number of tardy jobs와 관련하여 대부분 연구되어 왔고 그러한 수행측도는 job이 조기완료되는 비용은 무시해 왔다. Tardiness와 earliness의 취급을 달리하는 가장 중요한 이유중의 하나는 tardiness와 earliness에 할당되는 비용이 본질적으로 다르기 때문이다. 납기는 보통 외부적인 제한이기 때문에 tardiness는 명백하지만 earliness 비용은 간접적 특성을 갖는다. 조기완료 jobs은 자본과 부족한 면적과 관계되고 일반적으로 자원배당과 이용은 최적보다 낮게 갖는 것을 가리킨다. 그래서 earliness 비용은 비효율적인 비용이다.

Earliness와 tardiness 둘다 불합리한 실제적인 입장에서 Sidney[10]는 제한된 자원 projects 문제와 부패하기 쉬운 상품의 생산에 대해 언급했고, Kanet[7]는 Jobs의 penalty를 같은 비율로 두고, 납기 $d \geq MS$(MS는 jobs의 makespan)인 경우에 대한 분산의 합을 최소화하는 알고리즘(algorithm)을 제시하였고, Hall[6]은 Kanet의 모순되는 점을 보완하여 해의 확장성을 나타내고, parallel identical machines의 경우로 확장했다. 또 Eilon and Chowdhury[4]는 완료시간분산을 최소화하는 문제는 V형(V-shaped)이 되어야 하는 것을 보여주고 Bagchi, Sullivan, and Chang[1,2]은 Kanet의 single machine 문제가 2-parallel machine의 mean flow

* 경남전문대학 공업경영과 부교수
**동아대학교 대학원 산업공학과 박사과정
접수 1990년 4월 25일

time 문제에 동등함을 나타내고 또 MSD(Mean Squared Deviation) 문제가 완료시간 문제와 동등함을 나타낸다. Sundararaghavan and Ahmed[11]는 m-parallel identical machines으로 확장하고, d≤MS인 경우에 대한 근사적 방법을 개발하였다. 따라서 본 연구에서는 모든 jobs이 뚜렷한 다른 weights를 갖고 공통납기를 갖는 경우에 job의 완료시간의 W.M.A.D를 최소화하는 최적스케줄을 구하는 모형을 정식화하고 이문제에 대한 우월한 특성을 분석하여 3가지의 발견적 기법(heuristic methods)을 제시한다. 또 수치 시험을 통해 해의 분석과 유효성을 보이고 이 문제를 multiple machines인 경우로 확장하고자 한다.

2. 본 연구의 모형화

2.1 기호 설명

각 기호에 대한 설명은 다음과 같다.

N : n개 독립 jobs의 집합. N = {1, 2, …, n}
P_j : Job j의 processing time
C_j : Job j의 completion time
d : common due date
E : earliness의 jobs 집합
T : tardiness의 jobs 집합
W_j : job j의 *weighting factor*
MS : $\sum_{j\in N} P_j$ (jobs의 makespan)
[j] : E에서 j번째 job
[i] : T에서 i번째 job
S : 집합 N에서 임의의 n jobs의 순서
U : 처리순서가 미정인 jobs의 집합

2.2 전제조건과 목적함수

모형의 설정에 있어서 다음과 같은 조건을 전제로 한다.

(i) 각 job의 준비시간은 job의 가공순서에 무관계(가공시간에 포함).
(ii) 기계는 동시에 2개 이상의 jobs을 처리할 수 없다.
(iii) 일단 어떤 job이 가공을 시작하면 완료될 때까지 다른 job을 처리할 수 없다.
(iv) 모든 jobs은 언제든지 가공 가능한 상태이다.
(v) d는 d전에 job을 스케줄링하는데 충분한 freedom을 줄 수 있게 충분히 크다. (즉, d≥MS)

본 연구의 목적함수는 공통납기에 대한 jobs의 완료시간의 W.M.A.D를 최소화한다. 완료되는 job에 대해 매단위 시간 비용은 다르기 때문에 W.M.A.D문제는 M.A.D(Mean Absolute Deviation) 문제보다 스케줄링 목적을 달성하는데 더 큰 유연성을 제공할 수 있다.

구하고자 하는 스케줄은 다음과 같은 목적함수를 최소화하는 schedule S를 찾는 것이다.

$$Z(S)=\frac{\sum_{j\in N} W_j \mid C_j-d \mid}{\sum_{j\in N} W_j} \qquad \cdots(1)$$

목적함수 (1)은 다음과 같이 다시 표현할 수 있다.

$$\frac{\sum_{j\in N} W_j[(j-1)P_{[j]}+jP_{[j]}]}{\sum_{j\in N} W_j} \qquad \cdots(2)$$

하나의 순서 목적함수는 한 집합이 job parameters로 결정되는 두개의 같은 집합의 쌍의 합으로 모형화할 수 있다.

3. 발견적 기법의 특성 및 고찰

3.1 우월 특성(Dominance Properties)

WMAD 문제는 weighted mean flow time(WMF) 측도의 two-parallel machine 문제와 유사하다. 즉 스케줄 S의 E에서 job 순서는 두 기계의 하나에 고려될 수 있고 flow-time 측도하에서 역 순서에서 관찰되어진다. T에서 job 순서는 나머지 기계에 순서대로 표현할 수 있다. WMAD 문제는 최소한 NP-complete로 알려진 WMF 문제만큼 hard 문제이다[10].

WMAD 문제에 대한 우월한 특성을 다음과 같이 주어진다.

[Proposition 1(P1)]

최적 스케줄에서 어떤 두개 jobs 사이에 idle time은 존재하지 않는다.

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

만일 idle time이 d 이전(이후)에 일어났다고 하면 스케줄에서 idle time 앞으로(뒤로) jobs를 이동함으로써 idle time은 제거된다. 모든 jobs의 완료시간은 d에 가까이 접근함으로써 S 스케줄은 개량되어진다.

[*Proposition 2(P2)*]

납기전에 작업시작된 job 스케줄은 납기후에 완료되면 최적이 될 수 없다. 즉, 모든 최적 스케줄에서 하나의 job 완료 시간과 납기는 일치해야 한다.

증명. d 전에 job K는 가공 시작하여 d 이후에 완료되는 스케줄 S를 생각하자.

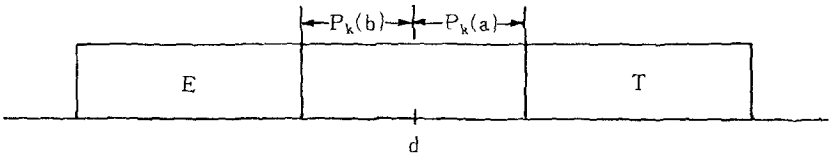

Figure 1. Schedules S

P(b)는 d 전에 job k가 가공되는 시간 양이고 P(a)는 d 이후에 job k가 가공되는 시간 양이다.

분명하게 $\sum_{j\in E} W_j \leq \sum_{j\in T} W_j + W_k$ or $\sum_{j\in E} W_j > \sum_{j\in T} W_j + W_k$이다.

먼저 $\sum_{j\in E} W_j \leq \sum_{j\in T} W_j + W_k$를 생각하자.

그러면 모든 jobs는 d=C_k까지 왼쪽으로 이동될 수 있다. 이 이동에 의한 목적 함수값의 변화는,

$$Z(S)-Z(S') = \sum_{j\in E} W_j P_k(b) + (\sum_{j\in T} W_j + W_k) P_k(a) - [\sum_{j\in E} W_j (P_k(b)+P_k(a))]$$

$$= (\sum_{j\in T} W_j + W_k - \sum_{j\in E} W_j) P_k(a) \geq 0$$

다음은 $\sum_{j\in E} W_j > \sum_{j\in T} W_j + W_k$를 생각하자.

그러면 모든 jobs는 d=C_k-P_k까지 이른쪽으로 이동될 수 있다. 이 이동에 의한 목적 함수값의 변화는

$$Z(S)-Z(S'') = \sum_{j\in E} W_j P_k(b) + (\sum_{j\in T} W_j + W_k) P_k(a) - [(\sum_{j\in T} W_j + W_k)(P_k(a)+P_k(b))]$$

$$= (\sum_{j\in E} W_j - \sum_{j\in T} W_j - W_k) P_k(b) > 0$$

따라서 목적 함수값은 그러한 이동에 의해 증가되지 않는다.

[Proposition 3(P3)]

만일 E에서 Jobs이 WLPT(Weighted Largest Processing Time) 순서이면 스케줄은 우월하다.

$$\frac{P[1]}{W[1]} \geq \frac{P[2]}{W[2]} \geq \cdots\cdots \frac{P[m_1]}{W[m_1]}$$

역시 T에서 Jobs이 WSPT(Weighted Shortest Processing Time) 순서에 있으면 우월하다.

$$\frac{P(m_2)}{W(m_2)} \leq \cdots\cdots \leq \frac{P[2]}{W[2]} \leq \frac{P(1)}{W(1)}$$

증명. Weighted V-shape가 아닌 스케줄 S를 생각하자.

위치 i와 i+1에서 jobs의 쌍이 존재한다면

(a) $\frac{P_i}{W_i} < \frac{P_{i+1}}{W_{i+1}}$, $i \in E$

(b) $\frac{P}{W} > \frac{P}{W}$, $i \in T$

jobs i와 i+1을 순서에서 교체한 새로운 스케줄은 S'라 하면 S'의 목적값이 S보다 더 작은 것을 보이면 충분하다.

(a)에 대해

$$Z(S)-Z(S') = (\sum_{k=1}^{i-1} W_k)P_i + (\sum_{k=1}^{i} W_k)P_{i+1} - [(\sum_{k=1}^{i-1} W_k)P_{i+1} + (\sum_{k=1}^{i-1} W_k + W_{i+1})P_i]$$

$$= W_i P_{i+1} - W_{i+1} P_i > 0$$

(b)에 대해

$$Z(S)-Z(S') = \sum_{k=1}^{n} W_k)P_i + (\sum_{k=i+1}^{n} W_k)P_{i+1} - [(\sum_{k=1}^{n} W_k)P_{i+1} + (W_i + \sum_{k=i+2}^{n} W_k)P_i]$$

$$W_{i+1} P_i - W_i P_{i+1} > 0$$

따라서 jobs i와 i+1을 교체함으로써 목적 함수값은 감소된다.

[Proposition 4(P4)]

jobs의 순서중에서 P_j/W_j가 가장 큰 job을 첫번째 가공해야 한다.

증명. (P3)은 만족하고 (P4)는 만족하지 않는 스케줄 S를 생각하자. 그러면 P_k/W_k가 가장 큰 job은 순서에서 마지막 위치에 있어야 한다. S에서 첫번째와 마지막 jobs를 교체함으로써 스케줄이 개량되는 것을 볼 수 있다.

WMAD 문제가 NP-complete 문제임에도 불구하고 최적 스케줄은 다음과 같은 3 경우에서는 polynomial time에 찾을 수 있다.

(1) 모든 weights가 동등하다.
(2) early와 late 완료 jobs가 다른 weights를 갖는다.
(3) 모든 processing times가 동일하다.

3.2 발견적 기법

Heuristic solution methods는 스케줄의 우월한 특성을 기초로하여 나타내어지고 모든 가능한 스케줄의 고려없이 좋은 해를 찾을 수 있다.

여기서 목적함수의 Z에 대해 최소 납기로서 d^*를 정의할 수 있고 unconstrained 문제는 최소한 d^*이거나 보다 큰 공통 납기를 가져야 한다.

WMAD 문제에 대한 효율적인 3가지 발견적 기법들을 나타내면 다음과 같다.

〈발견적 기법 1(H1)〉

Step 1) n jobs를 P_k/W_k 순으로 나열하고 P_k/W_k가 가장 큰 job k를 선택하여 E에 할당한다.

Step 2) U에서 다음으로 P_k/W_k가 가장 큰 job을 선택한다.

Step 3) early set(E)의 $\sum_{j\in E} W_j$와 late set(T)의 $\sum_{j\in T} W_j$에 대한 각각 배치된 jobs에 대한 weighting의 총량을 계산

Step 4) 만일 $\sum_{j\in E} W_j < \sum_{j\in T} W_j + W_k$이면 early set(E)에서 마지막 위치에 job k를 할당하고 만일 $\sum_{j\in E} W_j > \sum_{j\in T} W_j + W_k$이면 late set(T)에서 첫번째 위치에 할당한다.

Step 5) $U \neq \phi$이면 Step 2)로 가고, $U = \phi$이면 stop.

〈발견적 기법 2(H2)〉

Step 1) $E_i \leftarrow \phi$, i=1, …, m, $Ti \leftarrow \phi$, i=1, …, m

Step 2) $U \neq \phi$이면 U에서 P_k/W_k가 가장 큰 job k를 선택해 E에서 마지막 위치에 Job k를 정하고

Step 3) 만일 $U \neq \phi$이면 U에서 P_k/W_k가 가장 큰 job k를 선택해 T에서 첫번째 위치에 job k를 정한다.

Step 4) $S \leftarrow (E, T)$

Step 5) $U \neq \phi$이면 Step 2)로 가고, $U = \phi$이면 Stop.

〈발견적 기법 3(H3)〉

Step 1) $E = \phi$, $T = \phi$

Step 2) n Jobs를 P_k/W_k 순으로 나열한다.

Step 3) P_k/W_k가 가장 큰 job k를 선택하여 E의 마지막에 놓는다.

Step 4) 다음으로 P_k/W_k가 가장 큰 job k를 선택하여 E의 마지막에 놓고 $Z(B) = \sum_{j\in N} |C_j - d|$를 계산하고 T의 첫번째에 놓고 $Z(A) = \sum_{j\in N} |C_j - d|$를 한다.

Step 5) 만일 $Z(B) \leq Z(A)$이면 E의 마지막에 할당하고
$Z(B) > Z(A)$이면 T의 첫번째에 할당한다.

Step 6) $U \neq \phi$이면 Step 4)로 가고, $U = \phi$이면 Stop.

3.3 수치예제

10-jobs에 대해 제안된 발견적 기법의 응용을 예증하기 위해 Table 1에서 data들을 제공한다.

Table 1. Example Data

	1	2	3	4	5	6	7	8	9	10
P	21	18	42	29	15	48	2	15	37	46
W	17	14	4	11	8	10	13	12	1	16
P_i/W_j	1.24	1.28	10.5	2.64	1.89	4.8	0.15	1.25	37	2.89

$d = \sum_{j\in N} P_j = 273$

(H1)

Table 2. Step-to-step Results from Heuristic 1

Step	K*	$\sum_{j\in N} W_j$	$\sum_{j\in T} W_j+W_k$	Early	Tardy
1	9	0	1	9	
2	3	1	4	9-3	
3	6	5	10	9-3-6	
4	10	15	11	9-3-6-10	
5	4	15	27	9-3-6-10	4
6	5	31	19	9-3-6-10	5-4
7	2	31	33	9-3-6-10-2	5-4
8	8	45	31	9-3-6-10-2	8-5-4
9	1	45	48	9-3-6-10-2-1	8-5-4
10	7	62	44	9-3-6-10-2-1	7-8-5-4

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

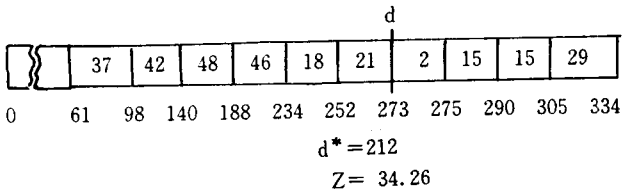

Figure 2. The schedule generated by (H1)

(H2)

Table 3. Step-to-Step Results from Heuristic 2

Step	K*	Assignment(E/T)	Early	Tardy
1	9	E	9	
2	3	T	9	3
3	6	E	9-6	3
4	10	T	9-6	10-3
5	4	E	9-6-4	10-3
6	5	T	9-6-4	5-10-3
7	2	E	9-6-4-2	5-10-3
8	8	T	9-6-4-2	8-5-10-3
9	1	E	9-6-4-2-1	8-5-10-3
10	7	T	9-6-4-2-1	7-8-5-10-3

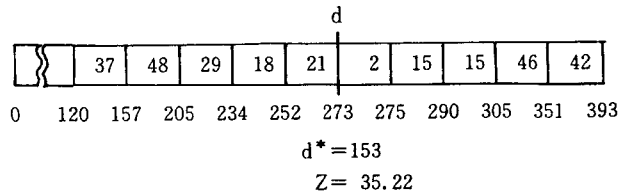

Figure 3. The schedule generated by (H2)

(H3)

Table 4. Step-to-Step Results from Heuristic 3

Step	K*	Z(B)	Z(A)	Early	Tardy
1	9	0	0	9	
2	3	42	42	9-3	
3	6	138	90	9-3	6
4	10	182	182	9-3-10	6
5	4	269	240	9-3-10	4-6
6	5	285	285	9-3-10-5	4-6
7	2	357	249	9-3-10-5	2-4-6
8	8	399	399	9-3-10-5-8	2-4-6
9	1	504	483	9-3-10-5-8	1-2-4-6
10	7	493	493	9-3-10-5-8-7	1-2-4-6

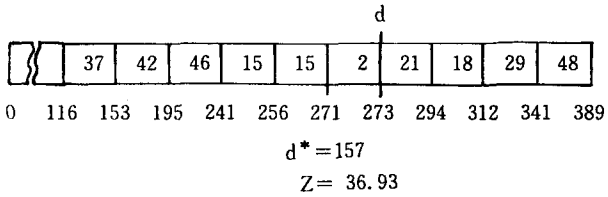

Figure 4. The schedule dgenerated by (H3)

주어진 문제에 대해 발견적 기법들을 적용하여 스케줄을 상호 비교하여 그 유효성을 평가하고자 한다. 평가에 사용한 문제를 편의상 다음과 같이 분류하여 10개 다른 job 집합(6, 7, 8, 9, 10, 15, 20, 30, 40, 50)을 사용하여 평가하였다.

Table. 5

Data set	1	1	1	1	1	1
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)

각각 일양분포에서 랜덤으로 열거하였고 Weight factor는 processing time에 독립이다. 6개 data sets에서 결과는 Table 6에 나타내고 60개 결과는 Table 7에 나타낸다.

Table 6. comparison of Heuristics

	DS_1	DS_2	DS_3	DS_4	DS_5	DS_6
H1	5	7	7	7	6	7
H2	4	1	1	2	4	0
H3	1	2	2	1	0	3

Table 7. Results of 60 test problems

	6	7	8	9	10	15	20	30	40	50	계
H1	2	3	2	4	6	3	3	5	6	5	39
H2	3	1	1	1	0	2	3	1	0	0	12
H3	1	2	3	1	0	1	0	0	0	1	9

Table 7에서 보면 우수 스케줄들 생성하는 빈도면에서 발견적 기법 1이 다른 기법보다 우수하며 job 수가 증가함에 따라 현저하게 나타난다.

발견적 기법 1은 weight factor를 다른 기법보다 더 강조하였다.

4. Parallel Identical Machines

m-processors가 있고 각 processor i(i=1, ···, m)에 대해 E_i, T_i, $[j_i]$, (j_i)를 정의한다. 그러면 single machine 목적함수(2)를 간단하게 수정해서 다음과 같이 multi-machine 목적함수를 나타낼 수 있다.

$$\frac{\sum_{i\in m}\left[\sum_{j\in N} W_{ji}((j-1)P_{[ji]}+jP_{(ji)})\right]}{\sum_{j\in N} W_j} \qquad \cdots(3)$$

weights가 같은 경우는 (즉 W=1) LPT 순서를 갖는 m-jobs의 i번째 집합을 S_i로 정의하면 즉, $S_1=\{1, 2, \cdots, m\}$, $S_2=\{m+1, m+2, \cdots, 2m\}$, ··· 등, 따라서 $E=S_1 U\ S_3 U\ S_5, \cdots$, $T=S_2 U\ S_4 U, \cdots$, 으로 나타낼 수 있다.

그러나 weights가 다른 경우는 다른 선택이 필요하다. 즉, weight factors에 따라 largest-to-smallest로 만나야 한다.

[Propostion 5(P5)]

m-jobs의 두 집합의 요소6의 쌍을 나타내고 각 쌍을 합할 때 m 합의 가장 큰 것을 최소화 하려면 최적쌍은 largest-to-smallest으로 만나야 한다.

증명. 한 집합 S_k에서 a_1와 $a_2(a_1>a_2)$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 관계가 된다.

$$\max\{a_1+b_1,\ a_2+b_2\} > \max\{a_1+b_2,\ a_2+b_2\}$$

따라서 쌍을 교체함으로써 단지 목적함수값은 증가되지 않는다.

[Algorithm for Minimal Weighted Mean Absolute Deviation]

Step 1) n-jobs를 P_k/W_k 순으로 나열한다.

Step 2) P_k/W_k가 가장 큰 job를 m개 선택하여 각 processor에 하나씩 할당한다.

Step 3) 다음으로 P_k/Wk가 가장 큰 job k를 m개 선택하고 $\sum_{i=1}^{m}\sum_{j\in E} W_{ji}$와 $\sum_{i=1}^{m}\sum_{j\in T} W_{ji}$를 계산한다.

Step 4) a) $\sum_{i=1}^{m}\sum_{j\in E} W_{ji} < \sum_{i=1}^{m}\sum_{j\in T} W_{ji}+W_k$이면 각 processor의 early set(E)에서 마지막 위치에 largest-to-smallest으로 위치를 할당하고

b) $\sum_{i=1}^{m}\sum_{j\in E} W_{ji} > \sum_{i=1}^{m}\sum_{j\in T} W_{ji}+W_k$이면 각 processor의 late set(T)에서 첫번째 위치에 largest-to-smallest으로 위치를 할당한다.

Step 5) $U\neq\phi$이면 Step 3)으로 가고, $U=\phi$이면 Stop.

Single machine 문제를 간단하게 수정해서 multiple machines 문제의 알고리즘을 만들었다. 이때 최소납기 d^*는 다음과 같다.

$$d^*=\max_i\left\{\sum_{j\in E} P_{[ji]}\right\}$$

5. 결 론

jobs의 완료시간의 W.M.A.D.를 최소화하는 스케줄에 관한 문제를 고찰하였다. 이 문제에 대한 최적의 스케줄을 얻는 것은 이론적으로는 가능하지만 실제에서는 계산 시간이 과다하게 소요되므로 짧은 시간안에 비교적 쉽게 실현 가능한 스케줄을 얻는 발견적 기법들을 제시하였다. 발견적 기법들에서 등호가 되는 경우에는 job을 E 집합에 넣고 있으나 임의적으로 선택하면 현재의 목적값보다 더욱 개량될 수 있다. Single machine을

간단하게 수정하여 multiple machines으로 확장하는 것을 보였고 또한 최소납기를 보여줌으로써 납기 결정에 중요한 기초가 된다.

본 연구를 기초로 하여 다수의 공통 납기를 갖는 일정 계획 문제는 앞으로 중요한 연구 과제가 될수 있다.

참 고 문 헌

[1] 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 Date", Naval Research Logistics Quarterly, 34, pp. 739~751, 1987.

[2] Bagchi, U., Sullivan, R. S., and Chang, Y. L., "Minimizing Mean Absolute Deviation of Completion Times about a Common Due Date", Naval Research Logistics Quarterly. 33, pp. 227~240, 1986.

[3] Baker, K. R., Introduction to Sequencing and Scheduling, John Wiley & Sons, New York. 1974.

[4] Eilon, S., and Chowdhury, I. G., "Minimizing Waiting Time Variance in the Single Machine Problem", Management Science, 23, pp. 567~575, 1977.

[5] Emmons, H., "Scheduling to a Common Due Date on Parallel Uniform Processors", Naval Research Logistics Quarterly, 34, pp. 803~810, 1987.

[6] Hall, N. G., "Single and Multiple-Processor Models for Minimizing Completion Time Variance", Naval Research Logistics Quarterly, 33, pp. 49~54, 1986.

[7] 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.

[8] Kanet, J. J., "*Minimizing Variation of Flow Time in Single Machine Systems*", *Management Science*. 27, pp. 1453~1459, 1981.

[9] Panwalkar, S. S., Smith, M. L., and Seidmann, A., "Common Due Date Assignment to Minimize Total Penalty for Penalty for the One Machine Scheduling Problem", Operations Research, 30, pp. 391~399, 1982.

[10] Sidney, J. B., "Single Machine Scheduling with Earliness and Tardiness Penalties", Operations Research, 25, No. 1, pp. 62~69, 1977.

[11] 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.

[12] Szwarc, W., "Single-Machine Scheduling to Minimize Absolute Deviation of Completion Times from a Common Due. Date", Naval Research Logistics Quarterly. 36, pp. 663~673, 1989.